



HAL
open science

Simulation interactive d'objets déformables pour la conception de simulateurs d'apprentissage aux gestes médicaux-chirurgicaux

Florence Zara

► To cite this version:

Florence Zara. Simulation interactive d'objets déformables pour la conception de simulateurs d'apprentissage aux gestes médicaux-chirurgicaux. Synthèse d'image et réalité virtuelle [cs.GR]. Université Claude Bernard Lyon1, 2020. tel-02977482

HAL Id: tel-02977482

<https://hal.science/tel-02977482v1>

Submitted on 5 Nov 2020

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

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



N° d'ordre NNT : 026 2020

**HABILITATION A DIRIGER DES RECHERCHES DE
L'UNIVERSITÉ DE LYON**

Opérée au sein de :
l'Université Claude Bernard Lyon 1
Préparée au :
**Laboratoire d'InfoRmatique en Images
et Système d'information (LIRIS)**

Soutenue publiquement le 19 octobre 2020 par
Florence ZARA

**Simulation interactive d'objets déformables pour la
conception de simulateurs d'apprentissage aux gestes
médicaux-chirurgicaux**

Devant le jury composé de :

M. Stéphane Cotin	Directeur de Recherche Inria Nancy - Grand Est	Rapporteur
M. Hervé Delingette	Directeur de Recherche Inria Sophia Antipolis - Méditerranée	Rapporteur
M. Bernhard Thomaszewski	Associate Professor Université de Montréal – Dept. I.R.O.	Rapporteur invité
M. François Faure	Professeur des Universités Univ. Grenoble Alpes – Inria Grenoble - Rhône-Alpes – LJK	Président
Mme Marie-Paule Cani	Professeur des Universités École Polytechnique – LIX	Examinatrice
Mme Maud Marchal	Maître de Conférences HDR INSA Rennes – IRISA	Examinatrice
M. Tanneguy Redarce	Professeur des Universités INSA Lyon – Laboratoire Ampère	Examinateur
M. Fabrice Jaillet	Maître de Conférences HDR Université Lyon 1 – LIRIS	Examinateur

A David.

A mes enfants, Lucie, Charlotte et Célian.

Pour la rédaction en musique, Placebo et Dionysos.

Remerciements

De coutume, je rédige ces remerciements quelques jours après avoir soutenu mon Habilitation à Diriger les Recherches.

Je tiens tout d'abord à remercier l'ensemble de mon jury de soutenance et en premier lieu mes trois rapporteurs, M. Bernhard Thomaszewski, M. Stéphane Cotin et M. Hervé Delingette, à qui j'ai donné beaucoup de travail en rapportant ce manuscrit plus long que la normale. Je remercie ensuite mon président de jury, M. François Faure, qui m'a initié à la simulation par modèles physiques en étant mon co-encadrant de doctorat ; Mme Marie-Paule Cani, qui fût la responsable d'une de mes équipes de recherche lors de mon doctorat (il y a déjà 20 ans...) ; Mme Maud Marchal avec qui c'est toujours un plaisir d'échanger. Je remercie chaleureusement, M. Fabrice Jaillet avec qui je travaille au quotidien. Je remercie très vivement M. Tanneguy Redarce qui m'a permis de débiter dans mon domaine actuel de recherche, en me proposant de travailler sur le développement du simulateur d'accouchement et en co-animant un des thèmes du GdR STIC-Santé.

Je remercie ensuite les membres de l'équipe ORIGAMI et plus particulièrement, M. Nicolas Bonneel, M. David Cœurjolly, M. Guillaume Damiand et Mme Julie Digne pour leurs remarques constructives lors de répétitions de soutenance d'HDR. Ils m'ont ainsi permis d'améliorer la présentation de mon travail, ainsi que mon projet de recherche.

Je remercie M. Emmanuel Promayon, pour le travail effectué en collaboration depuis de nombreuses années et son soutien amical et technique lors de la mise en place des deux visio-conférences nécessaires à ma soutenance d'HDR et pour une ultime répétition.

Je remercie à nouveau M. Guillaume Damiand, pour notre collaboration depuis de nombreuses années et pour celles à venir.

Je remercie M. Frédéric Chambat, pour le cours particulier qu'il m'a fait autour de la simulation me permettant de me lancer dans la rédaction de la première partie de ce manuscrit.

Je remercie mon conjoint, David Mayet, pour son soutien durant ces années de rédaction, mais également durant la dernière ligne droite en me permettant de travailler à toute heure en s'occupant du reste et plus particulièrement de trois enfants...

Avant propos

Au bout de quelques années de recherches, de plusieurs co-encadrements de doctorats qui ont été soutenus, d'écoutes de soutenances d'Habilitation à Diriger des Recherches (HDR), je me suis lancée à rédiger la mienne. Encore faut-il savoir en quoi cela consiste.

Arrêté du 23 novembre 1988

Article 1 : *"L'Habilitation à Diriger des Recherches sanctionne la reconnaissance du haut niveau scientifique du candidat, du caractère original de sa démarche dans un domaine de la science, de son aptitude à maîtriser une stratégie de recherche dans un domaine scientifique ou technologique suffisamment large et de sa capacité à encadrer de jeunes chercheurs".*

Article 4 : *"Le dossier de candidature comprend soit un ou plusieurs ouvrages publiés ou dactylographiés, soit un dossier de travaux, accompagnés d'une synthèse de l'activité scientifique du candidat permettant de faire apparaître son expérience dans l'animation d'une recherche.*

Dans le domaine plus particulier de l'Informatique, il existe des manuscrits qui correspondent simplement à des compilations d'articles, d'autres qui ne font que quelques pages, d'autres plus conséquents qui ressemblent à des manuscrits de doctorat. Il n'y a donc pas à ma connaissance de règles clairement établies pour rédiger un manuscrit d'HDR.

Pour ma part, j'ai choisi de mettre à profit la rédaction de mon HDR pour prendre du recul sur les années écoulées depuis mon recrutement en septembre 2005 en tant que maître de conférences à l'Université Lyon 1 au sein du LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information), mais également pour rédiger et valider les connaissances acquises au fil des années dans le domaine de la Mécanique. Ces connaissances sont en effet nécessaires dans mon domaine de recherche qui est la simulation réaliste par modèles physiques avec des applications directes dans le domaine médical.

Mon souhait est de pouvoir transmettre ce savoir auprès des étudiants qui débutent dans cette thématique de recherche. En effet, il existe de nombreux ouvrages en Mécanique mais il n'est pas forcément aisé de trouver la démarche globale qui nous intéresse pour la réalisation de simulations numériques de matériaux dits déformables, et encore moins de trouver les détails des différentes équations que tout le monde connaît, mais dont nous ne savons pas toujours d'où elles proviennent, surtout quand nous n'avons pas abordé ces connaissances durant notre cursus universitaire initial.

Dans ce sens, la première partie de mon HDR rassemble les connaissances nécessaires en Mécanique des Milieux Continus pour aborder la simulation du comportement d'objets déformables soumis à des contraintes, mais également en théorie des plaques minces, et cette première partie se termine par la présentation d'un modèle physique usuellement employé en Informatique Graphique qui est le système discret des masses-ressorts.

Table des matières

Introduction	1
I Notions préliminaires	13
1 Un peu de mécanique des milieux continus	15
1.1 Introduction	16
1.2 Cadre de la dynamique newtonienne	17
1.3 Vecteur de traction et tenseur des contraintes	20
1.4 Identification de l'équation à résoudre	22
1.5 Cas des objets élastiques : déformation et déplacement	23
1.6 Loi de comportement de l'objet	31
1.7 Energie de déformation élastique	40
1.8 Retour sur l'équation de la dynamique à résoudre	42
1.9 Discrétisation du problème continu	44
1.10 Méthode des éléments finis	48
1.11 Résumé des notions de linéarité ou non-linéarité	62
1.12 Conclusion	64
2 Modèle mécanique d'une plaque mince	67
2.1 Introduction	68
2.2 Cadre théorique de l'étude d'une plaque mince	69
2.3 Cinématique d'une plaque mince	72
2.4 Déformation d'une plaque mince	74
2.5 Tenseur de contraintes et loi de comportement	76
2.6 Énergie de déformation d'une plaque mince	79
2.7 Équation du mouvement d'une plaque mince	81
2.8 Conclusion	82
3 Modèle physique des masses-ressorts	85
3.1 Introduction	86
3.2 Présentation du système discret des masses-ressorts	87
3.3 Méthode d'intégration d'Euler implicite	90
3.4 Limitation naturelle du coefficient de Poisson	97
3.5 Instabilité naturelle en compression	100
3.6 Conclusion	103

II	Contributions	107
4	Apport du calcul formel pour la simulation physique	109
4.1	Contexte de ce travail de recherche	110
4.2	Notions préliminaires sur le modèle masse-tenseurs	112
4.3	Approche adoptée	114
4.4	Génération des équations	115
4.5	Un peu de détails techniques	122
4.6	Adaptation dynamique selon critères	127
4.7	Évaluation de notre approche	129
4.8	Discussion	133
5	Apport d'un modèle topologique pour la simulation physique	139
5.1	Contexte de ce travail de recherche	140
5.2	Notions préliminaires en topologie	141
5.3	Changements topologiques durant la simulation	144
5.4	Modèle unifié combinant topologie et physique	146
5.5	Discussion	156
6	Apport d'un modèle surfacique pour la simulation physique	163
6.1	Contexte de ce travail de recherche	164
6.2	Approches existantes sur les structures fines	165
6.3	Modèle d'une plaque mince par éléments finis CST-DKT	167
6.4	Formulation co-rotationnelle pour les éléments CST-DKT	183
6.5	Contrainte de préservation de volume	187
6.6	Simulation parallèle par coloriage de graphes	191
6.7	Évaluation du modèle CST-DKT et de sa parallélisation	194
6.8	Discussion	198
7	Amélioration du modèle physique des masses-ressorts	205
7.1	Contexte de ce travail de recherche	206
7.2	Travaux existants sur les modèles masses-ressorts	208
7.3	Proposition d'un système hybride	215
7.4	Formulation des constantes de raideur des ressorts	217
7.5	Formulation des forces de correction	219
7.6	Validation de notre système hybride	222
7.7	Proposition de forces non-linéaires pour les ressorts	229
7.8	Discussion	232
8	Vers la réalisation d'un simulateur d'accouchement	239
8.1	Contexte de ce travail de recherche	240
8.2	Pourquoi un simulateur d'accouchement ?	242
8.3	Avant tout, il faut créer les maillages	247
8.4	Comment simuler numériquement un accouchement ?	248
8.5	Couplage avec un dispositif physique	261
8.6	Discussion	264

9 Conclusion et perspectives	273
9.1 Bilan des contributions	274
9.2 Poursuite des travaux de recherche	279
9.3 Vers la réalisation d'un simulateur de ponction	284
9.4 Vers l'usage de l'Intelligence Artificielle ?	286
9.5 Bilan de la rédaction de mon HDR	287
III Annexe	291
A Rapport d'activités	293
B Liste des publications	307

Liste des symboles

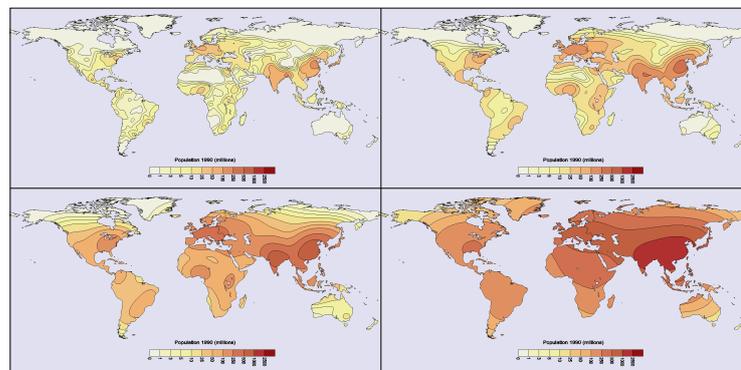
ϵ^{cd}	Tenseur de déformation de Cauchy-Green droit (appelé tenseur de dilatation)
ϵ^{cg}	Tenseur de déformation de Cauchy-Green gauche (appelé tenseur de dilatation)
ϵ^g	Tenseur de déformation de Green-Lagrange
λ	Premier coefficient de Lamé
\mathbb{F}	Gradient de la déformation
\mathbb{U}	Gradient du déplacement
μ	Second coefficient de Lamé appelé module de cisaillement - unité de mesure : le Pascal (Pa)
ν	Coefficient de Poisson du matériau - sans unité
Ω_t	Etat déformé qui correspond à l'état courant au temps t
Ω_0	Etat de référence qui correspond à l'état non-déformé au temps t_0
\vec{f}_{ext}	Forces extérieures volumiques appliquées sur l'objet - unité de mesure : $N.m^{-3}$
\vec{T}	Forces de contacts ou de tractions appliquées sur l'objet
∂D_t	Surface du domaine d'un matériau considéré à l'instant t
Φ	Base de fonctions connues indépendantes (dans le cadre de la MEF)
ϕ	Déformation pour passer de l'état de référence à l'état déformé
ρ	Masse volumique de l'objet
$\overset{\equiv}{C}$	Tenseur de rigidité
$\overset{\equiv}{D}$	Matrice de compliance
$\bar{\epsilon}$	Tenseur de déformation
$\bar{\sigma}$	Tenseur de contraintes de Cauchy dont les composantes sont homogènes à une pression - unité de mesure : le Pascal (Pa) ou $N.m^{-2}$
a	Vecteur des paramètres de l'approximation de l'interpolation (cadre de la MEF)
X	Position initiale d'un point matériel de l'objet - unité de mesure : le mètre (m)
x	Position courante au temps t d'un point matériel de l'objet

\vec{a}	Accélération de l'objet - unité de mesure : $m.s^{-2}$
\vec{F}	Forces appliquées sur l'objet - unité de mesure : le Newton (N) ou $Kg.m.s^{-2}$
\vec{g}	Force de la gravité - unité de mesure : $m.s^{-2}$
\vec{n}	Normale à la surface
\vec{u}	Déplacement d'un point matériel de l'objet
\vec{v}	Vitesse de l'objet - unité de mesure : $m.s^{-1}$
\vec{x}	Position de l'objet - unité de mesure : m
d	Dimension de l'objet ($d = 3$ en 3 dimensions)
D_t	Domaine d'un matériau considéré à l'instant t
ds	Élément de surface
dV	Volume élémentaire à l'instant t_0
dv	Volume élémentaire au cours du mouvement
E	Module de Young du matériau - unité de mesure : le Pascal (Pa)
I_{e_1}, I_{e_2}	Invariants du tenseur de déformation
k	Raideur d'un ressort - unité de mesure : $N.m^{-1}$ ou $Pa.m$
m	Masse de l'objet - unité de mesure : le kilogramme (Kg)
t	Temps - unité de mesure : la seconde (s)
V	Volume de l'objet - unité de mesure : m^3
W	Travail d'une force (énergie fournie par la force) - unité de mesure : le Joule (J)
$W(\epsilon)$	Énergie de déformation - unité de mesure : le Joule (J).
W_e	Densité d'énergie élastique - unité de mesure : $J.m^{-3}$

Introduction

Mon parcours de recherche ¹

Mon premier pas dans le monde de la recherche a été effectué en 1999 durant mon stage de maîtrise (IUP MAI - Mathématiques Appliquées et Industrielles) avec Jean-Marc Vincent au sein du projet APACHE du Laboratoire de Modélisation et de Calcul (LMC) de Grenoble. Ce travail de recherche a continué durant mon stage de DEA (Diplôme d'Études Approfondies) en 2000. L'objectif était de proposer dans un premier temps une nouvelle façon de calculer des cartes pour des géographes, en faisant appel à des notions de mathématiques appliquées avec une interpolation de Hermite, et dans un second temps de faire appel au calcul parallèle pour accélérer les temps de calculs sur une grappe de processeurs. Ce travail a permis de mettre au point un serveur interactif de cartes géographiques.

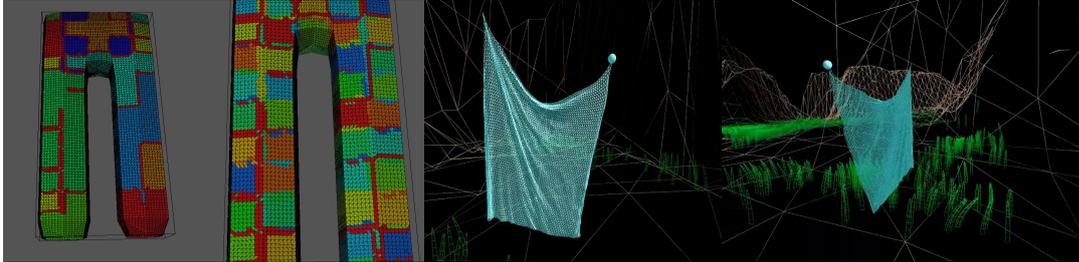


Travail issu de mon DEA. Potentiels de la population mondiale en 1990 en fonction du rayon d'interaction $R = 250, 500, 1000$ et 2000 km (respectivement de gauche à droite et de haut en bas).

J'ai continué mon travail de recherche dans le monde du parallélisme avec un doctorat de 2000 à 2003 à Grenoble dans le laboratoire ID-IMAG (Informatique et Distribution) au sein du projet INRIA-APACHE en collaboration avec l'équipe iMAGIS de l'INRIA Montbonnot. Mon doctorat a été encadré par Jean-Marc Vincent et François Faure avec l'aide précieuse de Thierry Gauthier et de Bruno Raffin, tous deux chercheurs au sein du projet INRIA-MOAI. L'idée était d'apporter les approches du monde du parallélisme dans le monde de l'informatique graphique. A ce moment-là, il n'y avait pas encore la disponibilité grand public des cartes graphiques que nous connaissons désormais. J'ai

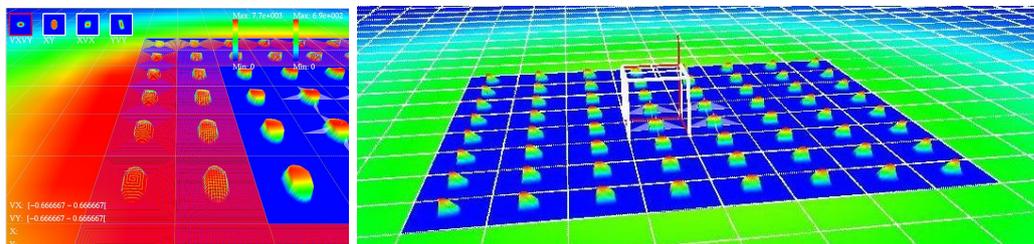
1. Les années ont passé, les laboratoires, les équipes et les projets ont souvent changé de noms entre temps (même parfois plusieurs fois...), mais pour être cohérente j'utilise les noms employés à l'époque.

mis en place des algorithmes parallèles pour la simulation de textiles modélisés par un système masses-ressorts avec la réalisation d'un couplage entre une simulation parallèle et une visualisation multi-écrans. Ce fut mon premier pas dans le monde de l'animation par modèles physiques et de l'informatique graphique.



Travail issu de mon doctorat. Partitionnement d'un pantalon et images issues de la "d mo valley".

Ensuite, changement de cap pour aller   Strasbourg de 2003   2005 au sein de l' quipe IGG (Informatique G om trique et Graphique) du LSIIT (Laboratoire des Sciences de l'Image, de l'Informatique et de la T l d tection). Je me suis attaqu e   des probl mes de simulation num rique de plus grande envergure, avec la simulation de faisceaux de particules ou de plasma pour le projet INRIA-CALVI. J'ai ainsi d couvert le monde de la physique mais  galement le monde de la visualisation en prenant part   l'encadrement du doctorat de Matthieu Haefele qui  t  dirig e par Jean-Michel Dischler (LSIIT-IGG) et Eric Sonnendr cker (IRMA - Institut de Recherche Math matique Avanc e). Ce doctorat a permis la mise en place d'outils permettant de visualiser une grande masse de donn es 4D. Les outils d velopp s s'appuyaient sur des algorithmes permettant de compresser de grands volumes de donn es multi-dimensionnelles que nous avons   visualiser.



Travail issu de mon ATER. Visualisation multi-r solution de l'espace d'int gration (v_x, v_y) (  gauche), activation de la loupe et d'un pointeur de valeurs (  droite).

En septembre 2005, j'ai  t  recrut e en tant que ma tre de conf rences au sein de l' quipe SAARA (Simulation, Analyse, Animation pour la R alit  Augment e) du LIRIS (Laboratoire d'InfoRmatique en Images et Syst mes d'information)   Lyon. Et depuis le 1er janvier 2020, je fais partie de l' quipe ORIGAMI regroupant 23 permanents du LIRIS autour des th matiques de l'Informatique Graphique. Mes recherches sont d sormais ax es sur la mod lisation et la simulation d'objets d formables et plus particuli rement de tissus mous. Ces recherches m'ont permis d'acqu rir de nouvelles connaissances en M canique des Milieux Continus. A noter que durant l'ann e universitaire 2014-2015, j'ai b n fici  de 6 mois de CRCT (Cong s pour Recherches ou Conversions Th matiques) d livr s par l'Universit  Lyon 1 et de 6 mois de d l gation CNRS me permettant ainsi de consolider une collaboration scientifique avec l' quipe GMCAO (Gestes M dico-Chirurgicaux Assist s par ordinateur) du laboratoire TIMC-IMAG de Grenoble.

Direction de mes recherches actuelles

Simulation du comportement des organes. De nombreux travaux s'attellent à la simulation de tissus mous aussi bien dans le domaine de la mécanique que dans celui de l'informatique graphique. L'objectif des chercheurs en mécanique réside dans la compréhension fine du comportement d'un ou plusieurs organes en interaction afin d'en améliorer notre connaissance. La Fig. 1 illustre la chaîne de traitements alors adoptée. Ainsi, à partir de connaissances anatomiques et d'images médicales (de type IRM ou scanner), un modèle géométrique des organes est élaboré à la précision des images médicales. En parallèle, la loi de comportement de l'organe ainsi que ses paramètres mécaniques modélisant son comportement physique sont déduits à partir d'expérimentations effectuées sur celui-ci. Enfin, après la définition des conditions limites, les Équations aux Dérivées Partielles régissant le comportement de l'organe au cours du temps sont généralement résolues par la méthode des éléments finis. Au final, cette chaîne de traitement permet l'obtention d'une simulation mécanique précise mais avec un temps de calcul généralement important pouvant aller jusqu'à plusieurs dizaines d'heures selon la complexité de la simulation.

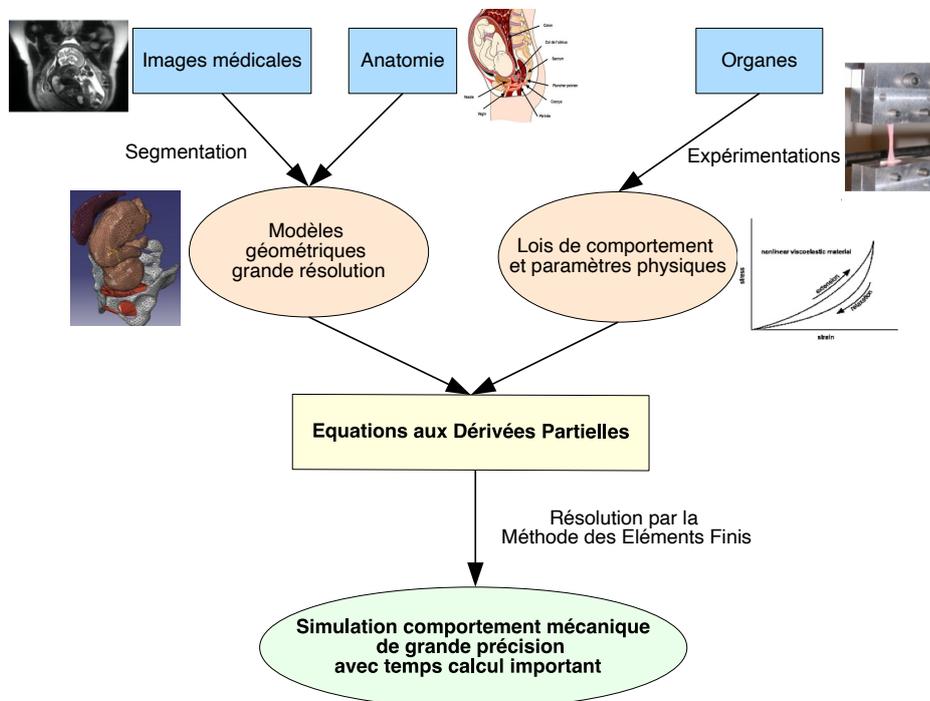


Figure 1 – Chaîne de traitement de la réalisation d'une simulation médicale dans le domaine de la mécanique permettant l'obtention d'une simulation de référence.

Pour notre part, l'objectif que nous nous sommes fixés réside dans l'intégration de ces simulations numériques au sein de simulateurs pour l'apprentissage de gestes médicaux.

Contexte médical. L'apprentissage réalisé directement auprès du patient pose des problèmes médicaux et éthiques avec l'exposition du patient à des risques qui sont directement liés à l'enseignement. C'est pourquoi, la formation médicale repose majoritairement sur l'observation plutôt que sur de la pratique. Or la dextérité nécessaire au maniement des outils chirurgicaux nécessite de la pratique ainsi que la compréhension et l'appropriation du geste à réaliser. L'emploi de systèmes d'entraînement basés sur de la simulation peut alors constituer une réponse à ce problème de formation. Ils répondent également aux

préconisations de la Haute Autorité de la Santé qui recommande dans sa lettre n° 30 de avril-juin 2012 : “Jamais la première fois sur le patient ! ”. De plus, l’enseignement *in situ* ne peut couvrir l’ensemble des situations possibles, alors que l’emploi de simulateurs permet la couverture d’un ensemble de situations plus larges, mais surtout permet de les cibler du point de vue de leurs pertinences pour la formation. L’efficacité de tels outils de formation passe ainsi par le bien fondé sur le plan pédagogique des situations proposées et par le réalisme adéquat de la simulation avec une immersion de l’apprenti dans le scénario proposé. Ces outils d’apprentissage peuvent ainsi se baser sur une simulation numérique couplée à un dispositif physique sur lequel l’apprenti retrouve les sensations perçues dans la réalité. L’idée est alors d’apporter par le biais de la simulation numérique des informations au dispositif physique (comme par exemple la position des organes sur lesquels ils interagissent ou encore les forces exercées sur eux), ainsi qu’un rendu visuel des mécanismes physiologiques engendrés lors du geste, permettant d’améliorer leur compréhension par l’apprenti.

Vers la conception de simulateurs médicaux. Nous visons ainsi la conception de simulateurs d’apprentissage de gestes médicaux-chirurgicaux qui combinent (voir Fig. 2) :

- Un dispositif physique permettant la manipulation par le médecin de l’instrument médical usuel et restituant les sensations ressenties lors du geste.
- Une simulation numérique du comportement des organes en interaction entre eux et avec les instruments médicaux, permettant un rendu visuel de la réalité ou au contraire permettant de voir des zones usuellement cachées, et permettant le calcul des informations nécessaires pour le dispositif physique.
- Un composant pédagogique avec différents scénarios pertinents pour la formation.



Figure 2 – Deux simulateurs combinant un dispositif physique et une simulation numérique : le Lap MentorTM de Sionix (à gauche) et le VirTeaSy Implant Pro de Didhaptic (à droite).

L’enjeu de ces nouveaux types de simulateurs médicaux est :

- d’assurer un apprentissage du geste médical sans risque pour le patient dans le cadre de la formation médicale initiale et continue ;
- d’accélérer le processus d’apprentissage en multipliant et ciblant les situations rencontrées, et en améliorant la connaissance du geste et du raisonnement associé ;
- d’élaborer à moyen terme de nouveaux gestes médicaux ainsi que leurs méthodes d’évaluation associées ;
- et de proposer à plus long terme des outils de planification pré-opératoire et d’entraînement aux opérations à risques dites “patient spécifique”, c’est-à-dire directement basées sur les données médicales d’un patient.

Verrous scientifiques à lever. La conception de ces simulateurs requière une approche pluridisciplinaire nécessitant des compétences² :

- en didactique, pour réaliser des simulateurs ayant un apport significatif pour l'apprentissage, en identifiant les paramètres à simuler et en mettant en place des scénarios d'apprentissage pertinents ;
- en informatique et mécanique, pour réaliser une simulation interactive avec un degré de précision adéquate pour l'apprentissage ;
- en mécatronique, pour concevoir une interface haptique adaptée aux gestes visés et restituant les sensations usuellement ressenties par le médecin lors de ces gestes ;
- et enfin cliniques, pour analyser et valider les différentes parties du simulateur.

Le challenge que nous essayons ainsi de relever en informatique consiste à proposer des simulations restituant un comportement réaliste des organes, en temps interactif, tout en assurant la stabilité nécessaire à un dispositif interactif. C'est-à-dire que nous ne recherchons pas une précision absolue du comportement biomécanique des tissus mous, mais à partir d'une connaissance théorique nous essayons de simplifier les simulations pour diminuer leurs temps de calcul tout en conservant un comportement global pertinent pour une intégration dans un simulateur d'apprentissage de gestes médicaux.

Contributions. La complexité de la simulation dépend notamment du modèle physique employé pour simuler le comportement mécanique de l'objet avec une loi de comportement plus ou moins complexe, de la discrétisation de l'objet en un plus ou moins grand nombre d'éléments, de la subdivision des éléments en un certain nombre de nœuds, et des méthodes numériques employées durant la simulation.

Pour diminuer le temps de calcul des simulations, nous avons ainsi travaillé au fil des années sur ces différents aspects avec :

- la proposition de **modèles permettant de gérer des objets que nous appelons mixtes**, c'est-à-dire des objets décomposés en éléments de types différents. Un élément est alors défini par sa topologie, sa géométrie et son comportement mécanique reproduit par l'emploi d'un modèle physique ;
- l'apport d'un **modèle topologique complexe** utilisé pour gérer ces objets mixtes. Ce modèle permet de modifier de manière optimale l'objet durant la simulation avec des changements au niveau de sa topologie (découpe, modification de la discrétisation de l'objet avec le raffinement ou le dé-raffinement d'une partie de l'objet), de sa géométrie (modification de la subdivision des éléments de l'objet) ou encore au niveau de son comportement mécanique (modification du modèle physique ou de la loi de comportement employés pour modéliser les éléments) ;
- la réalisation d'une chaîne de traitement permettant de **générer automatiquement les équations relatives aux forces exercées** sur un élément donné de l'objet en fonction de la définition de cet élément ;
- la mise en place d'un **modèle physique surfacique** permettant de gagner en temps de calcul par rapport à un modèle physique volumique ;

2. F. Zara, L. Vadcard, T. Redarce. Développement de nouvelles technologies pour la formation aux gestes médico-chirurgicaux. Annales des Mines. Réalités Industrielles. Connaissances et systèmes technologiques pour la santé. 2014.

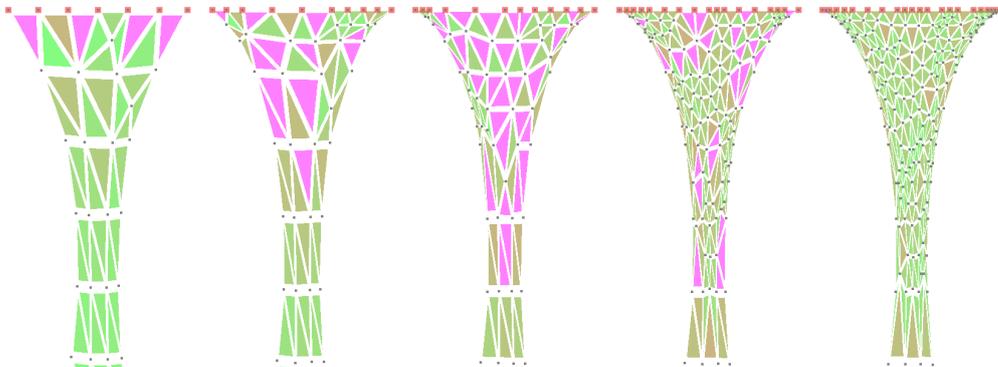
-
- l'**amélioration du modèle physique des masses-ressorts** permettant son emploi pour la simulation du comportement déformable d'objets définis par leur module de Young et coefficient de Poisson, ainsi qu'une formulation non-linéaire des forces élastiques.
 - la **parallélisation des traitements** effectués durant la simulation pour gagner encore un peu plus en temps de calculs.

L'idée sous-jacente est ensuite de jouer dynamiquement sur ces différents paramètres définissant la complexité en calcul de la simulation, par l'emploi de critères mis en place en fonction de l'application visée. Ces critères peuvent être des critères basés sur le temps de simulation, des critères géométriques pour assurer la qualité du modèle géométrique de l'objet, des critères mécaniques (contraintes ou déformations des éléments), des critères sur la convergence des méthodes, ou encore des critères de précision de calculs.

Encadrement doctoral

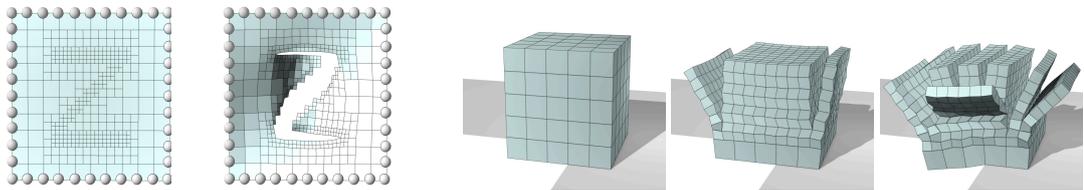
Ces travaux ont été réalisés durant le co-encadrement de 6 **doctorats** financées par le ministère (bourse ministérielle, et projet IDEFI-SAMSEI - Stratégies d'Apprentissage des Métiers de Santé en Environnement Immersif) et par la région Rhône-Alpes (cluster ISLE - Informatique Signal Logiciel Embrqué, PRRH - Programme de Recherche de la Région Rhône-Alpes en Hadronthérapie, LabEx PRIMES - Physique, Radiobiologie, Imagerie Médicale et Simulation), et durant des encadrements de stages de Master Recherche financés par le projet SAGA (Simulateurs pour l'Apprentissage des Gestes de l'Accouchement) de l'ANR MN (Modèles Numériques) 2012. Je donne ici un aperçu de ces travaux.

- Durant le doctorat de **Xavier Faure** (2010-2014) que j'ai co-encadré avec Fabrice Jaillet et Jean-Michel Moreau (LIRIS-SAARA), nous avons travaillé à la fois sur le modèle géométrique discrétisant l'objet en éléments de différents types et sur le modèle physique employé avec l'utilisation de différentes lois de comportement décrivant le comportement mécanique de l'objet. Ce travail de doctorat a abouti au développement d'un environnement permettant la gestion d'objets mixtes du point de vue loi de comportement et topologie/géométrie des éléments constituant l'objet, avec une évolution possible de ces éléments au cours de la simulation en fonction de critères spécifiques. Ce travail a mis en avant l'apport du calcul formel en simulation mécanique afin de générer automatiquement les formulations des forces appliquées sur les nœuds d'un élément ainsi que leurs dérivées en fonction de la topologie, la géométrie et de la loi de comportement de l'élément.



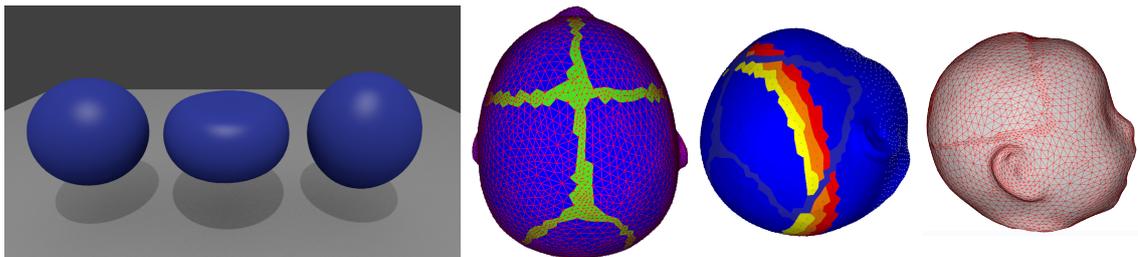
Doctorat de Xavier Faure. Raffinement au cours de la simulation en 2D.

• Durant le doctorat de **Elsa Fléchon** (2011-2014) que j'ai co-encadré avec Fabrice Jaillet avec l'aide de Guillaume Damiand (LIRIS-M2Disco), nous avons travaillé à intégrer des informations physiques au sein d'un modèle topologique. Ce doctorat a abouti à la proposition d'une structure de données permettant de décomposer un objet en cellules de type différent intégrant des informations relatives au modèle physique employé pour la simulation. Il est ainsi possible de combiner au sein d'un même objet, des cellules de topologie et de modèle physique ou de loi de comportement différents. De plus, le modèle topologique employé sous-jacent permet de modifier la topologie des objets au cours de leur simulation. La gestion dynamique de ces changements topologiques permet de raffiner ou de rendre plus grossier certaines zones du modèle géométrique décrivant l'objet simulé afin de concentrer les calculs aux endroits les plus pertinents (par exemple dans les zones de contact). Ce travail m'a permis de découvrir le monde de la topologie tout en mettant en avant ses bénéfices pour la simulation physique.



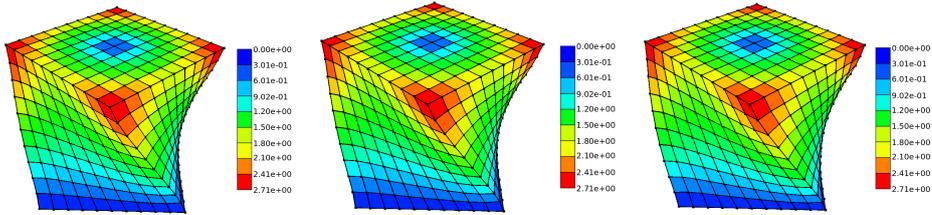
Doctorat de Elsa Fléchon. Modifications topologiques (raffinement et découpe) d'un objet 2D et d'un objet 3D au cours de leur simulation.

• Durant le doctorat de **Mathieu Bailet** (2011-2014) que j'ai co-encadré avec Emmanuel Promayon (laboratoire TIMC-IMAG de Grenoble), un nouveau modèle physique a été proposé permettant une simulation efficace d'objets incompressibles. Ce modèle est basé sur le modèle physique surfacique des coques auquel une contrainte de préservation de volume peut être intégrée. Ce modèle a été employé pour simuler la déformation subie par la tête du fœtus durant un accouchement. L'emploi d'un modèle surfacique et non volumique permet une diminution des temps de calculs de manière significative.



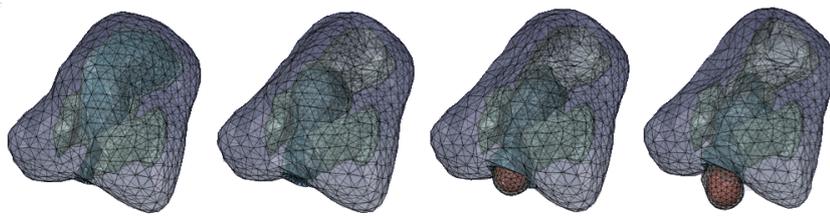
Doctorat de Mathieu Bailet. Pression appliquée sur une balle et sur la tête d'un fœtus.

• Durant le doctorat de **Karolina Golec** (2014-2017) que j'ai co-encadré avec Guillaume Damiand, Stéphane Nicolle (équipe Biomécanique des chocs du LBMC) en collaboration avec Jean-François Palierne (laboratoire de physique de l'ENS de Lyon), nous avons proposé une extension du système des masses-ressorts. Le souhait était d'utiliser ce modèle physique dans le cadre de la simulation du comportement non-linéaire de tissus mous.



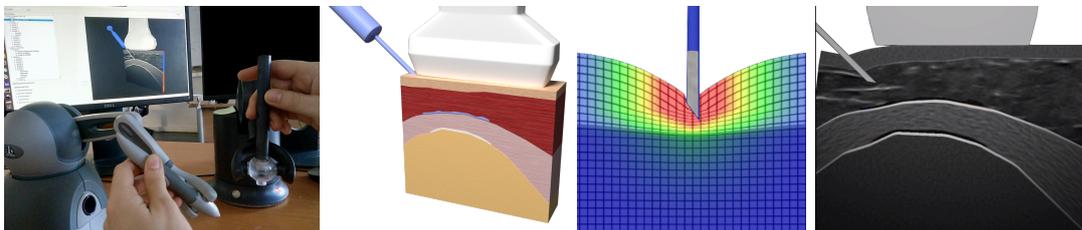
Doctorat de Karolina Golec. Test de torsion en grande déformation.

- Dans un contexte médical, j'ai co-encadré avec Behzad Shariat et Tanneguy Redarce (laboratoire Ampère de l'INSA Lyon) le doctorat de **Romain Buttin** (2007-2010) sur la modélisation et la simulation de la descente du fœtus au cours de l'accouchement. Ce doctorat nous a permis de comprendre le mécanisme complexe de l'accouchement et de valider la faisabilité d'un simulateur médical des gestes de l'accouchement couplant une simulation numérique à un dispositif physique. Le dispositif physique (BirthSIM) est notamment basé sur des mannequins usuellement employés par les sages-femmes pour leur apprentissage. Ce travail a abouti à la simulation de la descente du fœtus sans trajectoire imposée, ce qui constituait une première mondiale.



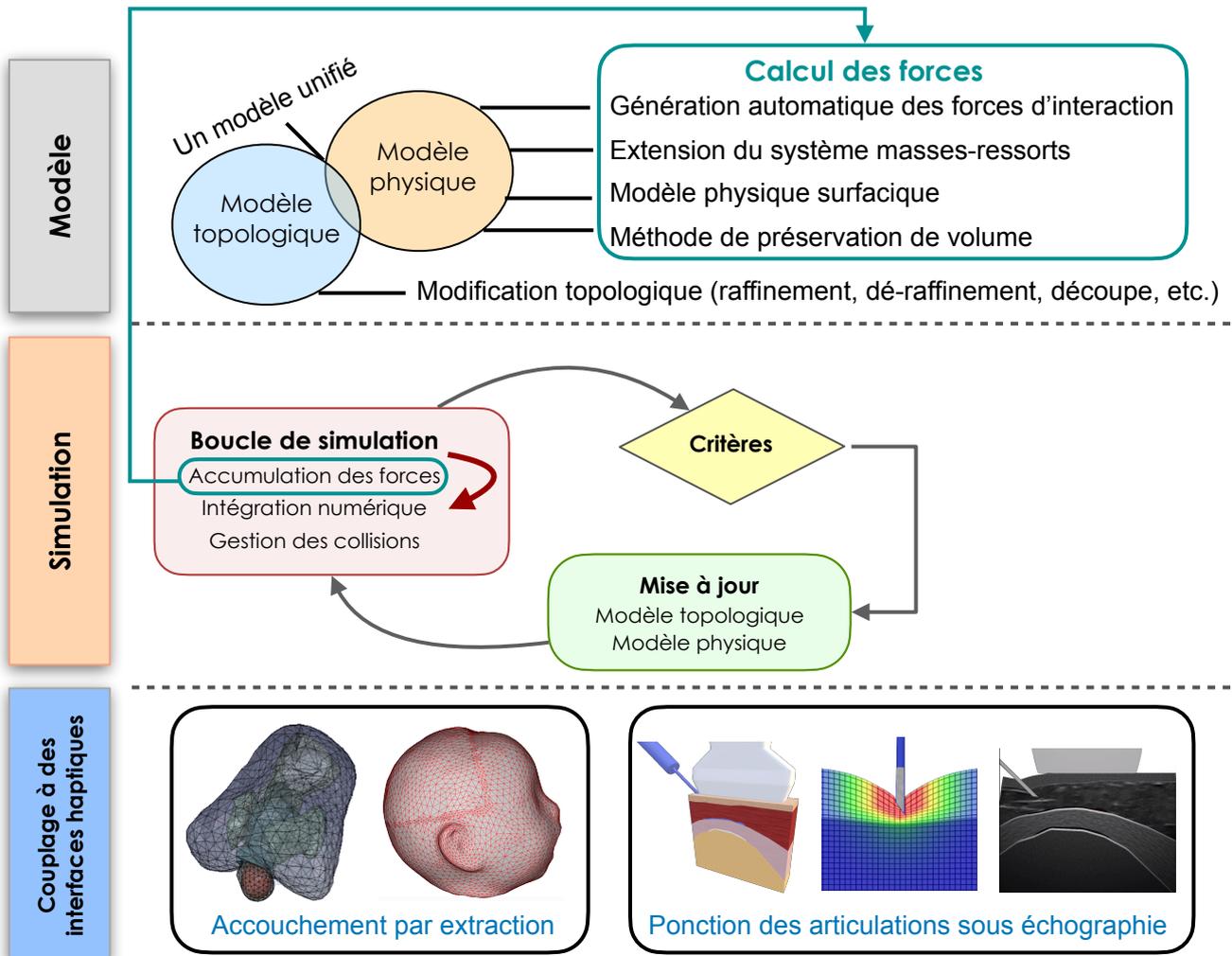
Doctorat de Romain Buttin. Simulation de la descente foetale durant l'accouchement.

- Enfin, le doctorat de **Charles Barnouin** (2016-2020) que j'ai co-encadré avec Fabrice Jaillet, rentre dans le cadre de la conception d'un simulateur de ponction sous échographie. Celui-ci a nécessité la réalisation d'une simulation numérique temps réel de l'insertion d'une aiguille dans une articulation couplée à deux interfaces haptiques représentant une sonde échographique et une aiguille. Je ne détaillerai pas dans ce manuscrit ces travaux de recherche qui se sont concentrés sur le calcul en temps réel de l'image échographique correspondant au rendu de la scène durant le geste de ponction. Je mentionnerai juste les objectifs de ce projet dans le chapitre 9.



Doctorat de Charles Barnouin. Simulateur d'entraînement pour le geste de la ponction sous échographie couplant deux interfaces haptiques et une simulation 3D avec un rendu ultra-sonore de la scène.

La Fig. 3 illustre notre stratégie de recherche et les contributions apportées dans le domaine de la simulation par modèles physiques.



Simulateurs pour l'apprentissage de gestes médicaux-chirurgicaux

Figure 3 – Stratégie adoptée pour élaborer des simulations interactives de tissus mous pour la conception de simulateurs pour l'apprentissage de gestes médicaux-chirurgicaux.

Développement logiciel

Ces travaux de recherche ont été réalisés sous différents environnements de simulation : en employant le progiciel Abaqus (développé par Dassault Systèmes) afin de vérifier la faisabilité d'une simulation ; en rajoutant de nouvelles briques dans la librairie Open Source SOFA développée au sein de la communauté d'informatique graphique ; ou encore en développant nos propres environnements de simulation (librairie TopoSIM et librairie NExT). Ces environnements de simulation peuvent être connectés ensemble grâce à la librairie CamiTK (Computer Assisted Medical Intervention Tool Kit) développée par le laboratoire TIMC-IMAG, facilitant ainsi leur comparaison et validation. A court terme, notre objectif est de combiner ces différentes approches au sein d'un même environnement de simulation et de valider l'ensemble de notre travail dans le cadre de la réalisation d'un simulateur pour l'apprentissage des gestes médicaux de l'accouchement.

Recherche en simulation médicale

Comme tout chercheur travaillant dans le domaine de la simulation médicale, j'ai appris à jongler entre différentes disciplines et acteurs pour mener à bien ces travaux de recherche. En effet, nous avons vu que la première étape consiste à obtenir des images médicales auprès de médecins afin de construire le modèle géométrique des organes. J'ai ainsi tissé des collaborations avec différents hôpitaux pour obtenir ces données médicales (groupe hospitalier Cochin-Saint Vincent de Paul de Paris, maternité Port Royal, centre hospitalier et maternité Lyon Sud des Hospices Civiles de Lyon). Ces acteurs médicaux nous aident également à comprendre les mécanismes physiologiques et anatomiques que nous souhaitons modéliser et à trouver des critères pertinents de validation. Puis, les compétences requises tombent dans le domaine de la Mécanique des Milieux Continus afin de caractériser le comportement mécanique des organes et appréhender les méthodes permettant de résoudre les équations de la mécanique régissant le comportement de l'objet simulé. Ensuite, nous pouvons retourner dans notre domaine de recherche initial qui est l'informatique, pour appréhender les différents modèles physiques proposés en Informatique Graphique et bénéficier des avancées en calcul parallèle avec notamment l'emploi des cartes graphiques et des architectures multi-cœurs, permettant de réduire les temps de calculs des algorithmes de simulation. Enfin, quand notre objectif concerne la conception de simulateurs d'apprentissage, il est important d'établir les éléments clés du métier à transposer dans la simulation ainsi que les scénarios à mettre en place pour garantir un apprentissage adéquat du geste médical visé. Ceci s'effectue en étroite collaboration avec des chercheurs en Science de l'Éducation.

Pour mieux appréhender ces différents aspects, et pour rencontrer facilement les acteurs majeurs de ces thématiques de recherche en France, j'ai animé de 2007 à 2014 avec Tanneguy Redarce (laboratoire Ampère de l'INSA de Lyon) et durant quelques années avec Lucile Vadcard (Laboratoire des Sciences de l'Éducation de l'Université de Grenoble), le thème F "Apprentissage et assistance aux gestes médicaux-chirurgicaux" du GdR STIC Santé. J'ai ainsi organisé une dizaine de journées thématiques abordant les thèmes nécessaires à la conception de simulateurs de gestes médicaux. L'animation de ce thème s'est terminée en 2014 par la mise en place d'une école d'été sur l'ensemble de ces thématiques, permettant d'apporter aux jeunes doctorants les connaissances nécessaires pour appréhender sereinement cette thématique de recherche. Les cours ont notamment été assurés par les acteurs français majeurs du domaine de la simulation médicale couvrant la chaîne de traitement complète de la réalisation de ces simulations médicales.

Encore quelques années de recherche assurées

Le challenge que j'ai énoncé précédemment concerne la réalisation de simulations interactives pour la conception de simulateurs pour l'apprentissage des gestes médicaux-chirurgicaux. Les verrous scientifiques à lever concernent l'obtention du temps interactif tout en garantissant une simulation stable pour reproduire le comportement mécanique des tissus mous en interaction entre eux et/ou avec des instruments médicaux. L'objectif est alors de réussir à restituer les sensations ressenties par le praticien, avec la réalisation d'une simulation globalement réaliste plutôt qu'une simulation fine de chacun des organes.

Ce challenge constitue une première étape vers le but recherché par tous les chercheurs en simulation médicale. En effet, notre Saint Graal concerne la conception de simulations dites "patient spécifique" permettant la planification et l'entraînement aux gestes d'une opération médicale délicate d'un patient en particulier. Il s'agit par exemple de s'entraîner à la résection d'une tumeur difficilement atteignable, de vérifier la faisabilité d'un accouchement par voie basse, ou encore de mesurer l'endommagement des tissus et ligaments du plancher pelvien d'une parturiente suite à un accouchement par voie naturelle.

Pour cela, à partir des données spécifiques d'un patient (images médicales, paramètres d'élasticité des tissus, volume de la respiration mesurée par un spiromètre, etc.), il s'agit de construire rapidement le modèle géométrique des organes concernés par l'acte médical, et d'élaborer leur modèle physique intégrant les paramètres mécaniques mesurés pour réaliser une simulation précise de l'acte médical visé. Au temps d'exécution, la précision du comportement des organes constitue alors une contrainte importante de la simulation qui pouvait être légèrement relâchée dans le cadre de simulateurs d'apprentissage.

Enfin le verrou scientifique inhérent à la simulation médicale concerne la phase de validation. Il est effectivement difficile de comparer directement le résultat d'une simulation numérique à la réalité quand il s'agit de la simulation d'une opération médicale, puisqu'il faudrait réussir à produire des données relatives à la même opération médicale. Comment par exemple valider la simulation d'un accouchement quand on ne peut pas avoir une IRM dynamique de cet accouchement ? Nous nous basons alors sur d'autres simulations établies comme étant des "gold standards" à atteindre. Ces simulations sont basées sur une modélisation la plus précise possible des organes visés et sur une méthode de résolution robuste. Mais ces standards comportent nécessairement des erreurs de précision car ils sont eux-mêmes des simulations qui sont basées sur des images médicales produites à une certaine résolution, et ces simulations sont également basées sur des mesures réalisées sur les tissus mous avec une erreur due à l'expérimentation, etc. Il s'agit donc de savoir quantifier l'erreur produite sur toute la chaîne de traitement afin de fournir l'erreur finale de la simulation et effectuer des compromis en conséquence.

Organisation du manuscrit

Ce manuscrit se poursuit par 9 chapitres divisés en 2 parties :

- La partie **I** rassemble les **connaissances théoriques** permettant d’appréhender le domaine scientifique de la simulation d’objets déformables. Dans ce sens, le chapitre 1 présente les bases de Mécaniques des Milieux Continus nécessaires à la modélisation des matériaux déformables. Ce chapitre se termine par la présentation de la Méthode des Éléments Finis usuellement employée pour résoudre le problème de la dynamique newtonienne établissant le déplacement d’un matériau au cours du temps. Puis le chapitre 2 présente le cas particulier de la modélisation des plaques minces sous les hypothèses de petites déformations et petits déplacements. Enfin, le chapitre 3 présente le modèle discret des masses-ressorts qui a été proposé dans le domaine de l’informatique graphique en alternative à la méthode des éléments finis.
- Dans la partie **II**, les chapitres 4 à 8 présentent les **travaux que j’ai réalisés et co-encadrés** en revenant ainsi sur chacune des doctorats brièvement cités dans le contexte particulier de la simulation interactive d’objets déformables. Nous verrons ainsi quelles stratégies peuvent être mises en place pour réduire le temps de calcul d’une simulation afin d’obtenir un temps d’exécution compatible avec l’emploi d’un dispositif haptique permettant d’interagir avec les objets simulés. Nous verrons ainsi que cela passe par la proposition de nouveaux modèles physiques capables de simuler le comportement de l’objet mais à moindre coût de calcul que l’approche de la MEF, ou encore par la parallélisation des algorithmes de simulation. Plus précisément au travers ces chapitres, nous verrons que j’ai travaillé à la fois sur des modèles physiques directement basés sur la MMC comme l’approche masse-tenseur (autre façon de résoudre les équations de la MMC) ou encore sur le modèle élément fini des coques permettant la modélisation du comportement d’une plaque mince, mais également sur le modèle discret des masses-ressorts en proposant une extension de ce modèle permettant d’aller au-delà de ses limitations naturelles.
- La partie **II** se termine par le chapitre 9 qui dresse un **bilan des contributions apportées** et ouvre les **perspectives de recherche** issues de ces travaux.

Ces travaux de recherche menés en même temps que mes enseignements et mes responsabilités administratives, ont donné lieu à des collaborations scientifiques au niveau national, à de l’animation scientifique principalement réalisée au sein du GdR STIC Santé, à des encadrements d’étudiants et d’ingénieurs, et à des publications scientifiques. Ceci est présenté en Annexe de ce manuscrit dans la partie **III** : le chapitre **A** présente mon **rapport d’activités** et le chapitre **B** donne la **liste de mes publications**.

Partie I

Notions préliminaires

Cette partie n'a pas vocation à présenter les contributions relatives à mes travaux de recherche, mais à expliquer comment sont caractérisés les matériaux dits déformables et comment ils sont modélisés pour simuler leurs comportements. Je rassemble ainsi dans cette partie les notions de mécanique acquises au fil des années, qui sont souvent décimées dans la littérature, mais qui sont nécessaires à la compréhension des travaux que j'ai effectués dans le domaine de la simulation physique d'objets déformables.

Comme il existe déjà de nombreux "State-of-the-Art Reports (STARs)" dans le domaine de l'Informatique Graphique, je ne fournis pas dans cette partie un état de l'art de tous les modèles proposés dans le domaine de l'animation par modèles physiques, mais je me concentre sur les modèles employés au cours de mes travaux.

Cette première partie se décompose en 3 chapitres.

- Le chapitre 1 concerne le domaine de la Mécanique des Milieux Continus. Il donne les bases nécessaires à la compréhension des grandeurs employées en simulation des matériaux dans le cadre de l'étude des objets déformables.
- Le chapitre 2 présente le modèle mécanique d'une plaque mince sous les hypothèses de petites déformations et de petits déplacements.
- Le chapitre 3 présente le modèle physique des masses-ressorts employé en animation pour modéliser le comportement d'un objet déformable.

Chapitre 1

Un peu de mécanique des milieux continus

Ce chapitre présente les notions de mécanique des milieux continus nécessaires à la compréhension du comportement des objets déformables. En effet, ces notions sont souvent éparpillées dans la littérature et les notations employées en mécanique diffèrent de celles usuellement employées par les mathématiciens et informaticiens. C'est pourquoi, je souhaitais rassembler ces connaissances dans ce chapitre, pour ensuite aborder sereinement mes travaux de recherche sur la modélisation des objets déformables.

Sommaire

1.1	Introduction	16
1.2	Cadre de la dynamique newtonienne	17
1.3	Vecteur de traction et tenseur des contraintes	20
1.4	Identification de l'équation à résoudre	22
1.5	Cas des objets élastiques : déformation et déplacement	23
1.6	Loi de comportement de l'objet	31
1.7	Energie de déformation élastique	40
1.8	Retour sur l'équation de la dynamique à résoudre	42
1.9	Discrétisation du problème continu	44
1.10	Méthode des éléments finis	48
1.11	Résumé des notions de linéarité ou non-linéarité	62
1.12	Conclusion	64

1.1 Introduction

Les personnes travaillant dans le domaine de l'animation par modèles physiques ont déjà lu à maintes reprises des chapitres de manuscrits de thèse présentant les notions de mécanique nécessaires à la modélisation du comportement des objets déformables.

Je peux citer ceux de Aurélien [Deram \(2012\)](#), Mathieu [Bailet \(2014\)](#) et Vincent [Majorczyk \(2015\)](#). Il existe également de nombreuses références académiques sur le domaine de la Mécanique des Milieux Continus. Parmi ceux qui m'ont le plus aidée à comprendre ces notions, je peux citer le livre de [Coirier and Nadot-Martin \(2013\)](#) et celui de [Belytschko et al. \(2000\)](#), et comme supports d'enseignement ceux de Christian [Olagnon \(2010\)](#), de Nicolas [Moës \(2011\)](#), de Samuel [Forest et al. \(2010\)](#), de Helmut [Klöcker \(2003\)](#) et de [Sifakis and Barbic \(2012\)](#), ou encore ceux de Hervé [Oudin \(2008\)](#) et Yves [Debard \(2011\)](#) sur la méthode des éléments finis.

En voici encore un... En effet, un de mes souhaits en rédigeant mon Habilitation à Diriger les Recherches, était de mettre par écrit les notions acquises au cours de plusieurs années de recherche. N'ayant en effet jamais trouvé de support présentant l'ensemble de ces notions, je souhaitais les regrouper en espérant que cela puisse aider les chercheurs qui débutent dans ce domaine. Je présente ainsi ici les notions théoriques de mécanique nécessaires à la modélisation et la simulation physique d'objets déformables.

Organisation de ce chapitre : Ce chapitre s'articule de la façon suivante.

- La section [1.2](#) présente la dynamique newtonienne avec le rappel du principe fondamental de la dynamique. Nous aborderons ensuite la notion de contraintes appliquées sur l'objet en définissant dans la section [1.3](#) le vecteur de tractions en fonction du tenseur de contraintes du matériau. Ces notions permettront de définir dans la section [1.4](#), l'équation régissant le mouvement de l'objet.
- La section [1.5](#) se focalisera ensuite sur le cas des objets élastiques. Nous définirons les notions de déformation et de déplacement. Nous verrons notamment que le gradient mathématique de ces grandeurs physiques permet d'obtenir des indications sur la variation de ces grandeurs dans toutes les directions de l'espace. Nous verrons ensuite des tenseurs de déformations plus complexes.
- Dans la section [1.6](#), nous verrons ensuite que la loi de comportement d'un matériau permet de relier les contraintes appliquées sur l'objet à la déformation subie par celui-ci. Quelques exemples seront donnés pour illustrer le cas de l'élasticité isotrope.
- Dans la section [1.7](#), nous aborderons le calcul de l'énergie de déformation élastique.
- Dans la section [1.8](#), nous reviendrons sur l'équation à résoudre pour définir le mouvement de l'objet au cours du temps. Puis, dans la section [1.9](#), nous verrons comment est établie la formulation discrète du problème initial.
- Dans la section [1.10](#), nous présentons la méthode des éléments finis (MEF) classiquement utilisée pour la résolution du système discret.
- Enfin, dans la section [1.11](#), nous résumons les notions de linéarité et de non-linéarité.

1.2 Cadre de la dynamique newtonienne

Nous cherchons à reproduire le mouvement d'un objet en calculant l'évolution de sa position au cours du temps. Pour cela, le temps qui est une grandeur continue doit être discrétisé en différents instants t . Soit h le pas de temps de cette discrétisation. Ainsi, si nous considérons que t_0 est le temps initial de la simulation, le temps suivant est $t_1 = t_0 + h$, puis le temps encore suivant est $t_2 = t_1 + h = t_0 + 2h$, etc.

Pour calculer la position d'un objet, nous nous appuyons sur la mécanique newtonienne basée sur les lois de Newton qui sont toujours vraies et qui s'appliquent à tout élément solide. La première loi de Newton dit que le mouvement d'un objet est provoqué par les forces qui lui sont appliquées. La seconde loi de Newton, plus connue sous le nom de "Principe Fondamental de la Dynamique", dit que la somme des forces qui sont exercées sur un objet est égale à la masse de l'objet multipliée par son accélération. Si nous nous plaçons dans l'espace 3D, cette seconde loi se formule de la manière suivante :

$$\vec{F}(t) = m \vec{a}(t) \quad (1.1)$$

avec m la masse (en Kg) de l'objet qui n'évolue pas dans le temps, $\vec{F}(t) \in \mathbb{R}^3$ le vecteur des forces (en N) appliquées sur l'objet à l'instant t , $\vec{a}(t) \in \mathbb{R}^3$ le vecteur correspondant à son accélération (en $m s^{-2}$) à l'instant t . Le vecteur des forces et de l'accélération sont définis à chaque instant t car ils évoluent au cours du temps. Ainsi, connaissant les forces qui sont appliquées sur un objet, nous en déduisons le vecteur accélération de cet objet, la masse étant un paramètre connu de la simulation. Soit :

$$\vec{a}(t) = \frac{\vec{F}(t)}{m}.$$

A partir de là, il suffit d'intégrer par rapport au temps le vecteur accélération pour obtenir le vecteur vitesse de l'objet, puisque l'accélération est la dérivée par rapport au temps de la vitesse. Soit :

$$\vec{v}(t) = \int_t \vec{a}(t) dt \quad \text{avec} \quad \vec{a}(t) = \frac{d}{dt} \vec{v}(t).$$

De la même façon, une fois que nous avons le vecteur vitesse, il suffit de l'intégrer par rapport au temps pour obtenir le vecteur position de l'objet, puisque la vitesse est la dérivée par rapport au temps de la position. Soit :

$$\vec{x}(t) = \int_t \vec{v}(t) dt \quad \text{avec} \quad \vec{v}(t) = \frac{d}{dt} \vec{x}(t).$$

Comme ces intégrales n'ont que très rarement des solutions analytiques, un schéma d'intégration numérique doit généralement être utilisé pour calculer les vecteurs vitesse et position en calculant une approximation des intégrales des vecteurs accélération et vitesse. Le choix du schéma d'intégration employé nécessite alors un compromis entre la précision de son résultat (en choisissant une approximation ayant une erreur plus ou moins grande) et le temps de calcul nécessaire à l'emploi de ce schéma d'intégration numérique.

En animation physique, les schémas d'intégration numérique qui sont privilégiés sont le schéma d'Euler semi-implicite (nécessitant des petits pas de temps pour ne pas faire une erreur trop importante) et le schéma d'Euler implicite ([Baraff and Witkin \(1998\)](#)) (dit

inconditionnellement stable, c'est-à-dire donnant une approximation numérique ayant la même erreur quelque soit le pas de temps, mais nécessitant la résolution d'un système linéaire, c'est-à-dire beaucoup plus de calculs que la première méthode). Pour le schéma d'Euler semi-implicite, les calculs sont très simples avec :

$$\begin{cases} \vec{v}(t+h) &= \vec{v}(t) + h \vec{a}(t) \\ \vec{x}(t+h) &= \vec{x}(t) + h \vec{v}(t+h) \end{cases}$$

Au final, la boucle de simulation, c'est-à-dire l'enchaînement des calculs permettant l'obtention de la position d'un objet à chaque instant t , est décrite par les étapes suivantes :

1. Calcul de l'ensemble des forces $\vec{F}(t)$ qui sont exercées sur l'objet.
2. Calcul de l'accélération $\vec{a}(t)$ de l'objet en divisant les forces $\vec{F}(t)$ par la masse m .
3. Calcul des nouvelles vitesse $\vec{v}(t+h)$ et position $\vec{x}(t+h)$ de l'objet par intégration numérique des vecteurs accélération et vitesse précédents.
4. Affichage de l'objet à l'écran en fonction de la nouvelle position $\vec{x}(t+h)$.
5. Puis itération en reprenant l'étape 1 pour obtenir les nouvelles forces $\vec{F}(t+h)$.

A noter que nous considérerons ici l'analyse du mouvement d'objets déformables, c'est-à-dire qu'à la différence des objets rigides, la distance entre deux points quelconques de l'objet peut varier au cours du temps. Ce changement est fonction du matériau considéré. En effet, les mêmes efforts appliqués sur des objets de même géométrie mais de matériaux différents (par exemple caoutchouc et acier) ne conduisent pas à la même déformation (c'est-à-dire au même changement dans la forme des objets). Nous aurons ainsi besoin de définir des relations (appelées lois de comportement) permettant de relier les efforts appliqués à la déformation induite et ceci en fonction du matériau considéré.

L'enjeu de la simulation physique d'objets déformables réside ensuite dans le calcul des forces appliquées sur l'objet simulé. Ces forces sont dites externes quand elles sont relatives aux forces extérieures appliquées sur l'objet (comme la gravité ou des forces de contacts issues de l'interaction avec un autre objet). Ces forces sont dites internes quand elles sont relatives au modèle physique employé pour modéliser le comportement physique de l'objet. Cette modélisation physique de l'objet est usuellement basée sur sa décomposition en éléments. Le modèle physique permet ensuite de déterminer les interactions qu'il y a entre ces éléments, c'est-à-dire de calculer les forces entre ces éléments. Ce calcul est alors plus ou moins complexe et gourmand en temps d'exécution, selon le modèle physique employé ainsi que la décomposition adoptée en un plus ou moins grand nombre d'éléments.

Loi fondamentale de la dynamique. En Mécanique des Milieux Continus, nous appliquons le principe fondamental de la dynamique à tout élément de volume. Nous définissons ainsi un volume élémentaire dV centré autour de la position géométrique $\mathbf{X} \in \mathbb{R}^3$ à un instant donné t_0 dans un espace affine euclidien. Nous considérons ensuite que toutes les particules constituant le matériau qui sont présentes à l'instant t_0 dans le volume élémentaire dV restent dans le même volume dv au cours du mouvement de l'objet. Le centre de gravité $\mathbf{x}(t) \in \mathbb{R}^3$ de l'élément de volume dv caractérisera alors la position du point matériel aux instants t ultérieurs. Le volume élémentaire dV doit être représentatif du matériau et doit rester suffisamment petit devant les dimensions de l'objet étudié.

Soit D_t le domaine du matériau considéré à l'instant t et ∂D_t sa surface. Le volume de l'objet est alors défini par $V = \int_{D_t} dv$. Ensuite, si nous supposons que la mesure masse est continue par rapport au volume de l'objet ayant une vitesse $\vec{v}(t) \in \mathbb{R}^3$ à l'instant t et en notant $\rho = m/V$ la masse volumique de l'objet (soit $m = \rho V$), nous pouvons écrire :

$$m \vec{a}(t) = m \frac{d}{dt} \vec{v}(t) = \rho V \frac{d}{dt} \vec{v}(t) = \frac{d}{dt} \int_{D_t} \rho \vec{v}(t) dv.$$

Le principe fondamental de la dynamique peut alors s'écrire de la façon suivante :

$$\frac{d}{dt} \int_{D_t} \rho \vec{v}(t) dv = \int_{D_t} \rho \vec{f}_{ext} dv + \int_{\partial D_t} \vec{T} ds \quad (1.2)$$

avec $\vec{f}_{ext} \in \mathbb{R}^3$ les forces extérieures volumiques appliquées sur l'objet et $\vec{T} \in \mathbb{R}^3$ les forces appliquées sur la surface de l'objet. Ces forces de contacts ou tractions appliquées à la surface correspondent à celles qui sont notamment appliquées sur l'objet pour effectuer des tests de poussée ou de cisaillement. Au final, l'enjeu réside dans le calcul de l'ensemble des forces appliquées sur l'objet dont la réaction va dépendre du matériau de l'objet. Il est important de noter que \vec{f}_{ext} correspond à des forces volumiques qui s'exercent sur la totalité de l'objet c'est-à-dire que \vec{f}_{ext} est un vecteur de densité de force par unité de volume, tandis que \vec{T} correspond aux forces appliquées sur le bord externe de l'objet c'est-à-dire que \vec{T} correspond à des forces appliquées par unité de surface.

Force de gravité. Comme force volumique considérons l'exemple classique de la densité des forces volumiques exercées par la gravité sur un milieu continu. Soit $\|\vec{g}\| = 9,81 \text{ ms}^{-2}$ la constante de gravité. La force de gravité appliquée sur l'objet est alors définie par :

$$\int_{D_t} \rho \vec{g} dv = \rho V \vec{g} = m \vec{g}.$$

Conservation de la masse. Il y a un certain nombre d'équations de bilan à respecter lors d'un mouvement dans un milieu continu. La masse de tout domaine matériel reste notamment constante pendant le mouvement de ce domaine induisant :

$$\frac{d}{dt} m = \frac{d}{dt} \int_{D_t} \rho dv = 0.$$

En utilisant les formules de la dérivée particulaire d'une intégrale de volume, nous obtenons^{1, 2} :

$$\int_{D_t} \left(\frac{\partial}{\partial t} \rho + \text{div}(\rho \vec{v}) \right) dv = 0 \quad \text{ou} \quad \int_{D_t} \left(\frac{\partial}{\partial t} \rho + \rho \text{div} \vec{v} \right) dv = 0.$$

Nous pouvons alors en déduire les **formes locales de la conservation de la masse** :

$$\frac{\partial}{\partial t} \rho + \text{div}(\rho \vec{v}) = 0 \quad \text{et} \quad \frac{\partial}{\partial t} \rho + \rho \text{div} \vec{v} = 0.$$

1. La dérivée particulaire du volume élémentaire dv est égale à $\frac{d}{dt}(dv) = \text{div}(\vec{v}) dv$. Toutes les explications se trouvent dans [Coirier and Nadot-Martin \(2013\)](#) et [Moës \(2011\)](#).

2. Pour rappel, soit \vec{u} un vecteur de composantes (u_1, u_2, u_3) , $\text{div} \vec{u} = \frac{\partial u_1}{\partial x} + \frac{\partial u_2}{\partial y} + \frac{\partial u_3}{\partial z}$ (un scalaire).

1.3 Vecteur de traction et tenseur des contraintes

Concentrons nous maintenant sur les contraintes appliquées sur le bord externe de l'objet. Elles sont caractérisées en un point de l'objet par un vecteur de contraintes (ou vecteur de tractions) qui est défini pour une unité de surface ds autour de ce point par

$$\vec{T} = \bar{\bar{\sigma}} \cdot \vec{n}, \quad T_i = \sum_{j=1}^3 \sigma_{ij} n_j \quad (1.3)$$

avec $\vec{n} \in \mathbb{R}^3$ la normale à l'élément de surface ds et $\bar{\bar{\sigma}}$ le tenseur de contrainte de Cauchy (représenté dans une certaine base) qui est d'ordre 2³.

Le tenseur $\bar{\bar{\sigma}}$ décrit l'état de contrainte en tout point de l'objet et dans toutes les directions. Ses composantes sont homogènes à une pression (en Pa) puisqu'elles correspondent à une force exercée sur une unité de surface autour d'un point (N.m^{-2}). En observant l'état d'équilibre en rotation visible sur la Fig. 1.1, nous pouvons voir que ce tenseur est symétrique avec $\bar{\bar{\sigma}} = \bar{\bar{\sigma}}^T \Leftrightarrow \sigma_{ij} = \sigma_{ji}$. Les termes diagonaux du tenseur de contrainte correspondent à des tractions et les termes non diagonaux à du cisaillement. Notons que ces composantes ont des valeurs positives dans le cas de la traction, c'est-à-dire quand nous tirons sur l'objet. Au final, ce tenseur est de la forme :

$$\bar{\bar{\sigma}} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}. \quad (1.4)$$

Ainsi par symétrie seulement 6 composantes sont nécessaires à sa définition. Nous pouvons alors employer la notation de Voigt⁴ consistant à créer un vecteur à partir de la matrice symétrique en supprimant ses doublons :

$$\bar{\bar{\sigma}} = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{pmatrix} = \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix}. \quad (1.5)$$

La Fig. 1.1 illustre la signification de ces composantes.

- La composante σ_{12} (ou σ_{xy}) correspond à la composante du tenseur $\bar{\bar{\sigma}}$ en considérant la composante dans la direction de x_2 (ou y) de la contrainte appliquée à la face perpendiculaire à la direction x_1 (ou x).
- Les composantes d'une ligne ou d'une colonne représentent les composantes du vecteur de contraintes s'exerçant sur une face perpendiculaire à un des axes. Par exemple, les composantes de la première ligne ($\sigma_{11}, \sigma_{12}, \sigma_{13}$) sont celles du vecteur de contrainte s'exerçant sur la facette perpendiculaire à la direction x_1 (ou x).

3. Noter que le nombre de barres mises sur le tenseur indique l'ordre du tenseur.

4. Pour retenir cette notation, nous mettons tout d'abord les éléments diagonaux dans l'ordre des indices, puis les éléments non diagonaux. Pour ces derniers en \mathbb{R}^3 , à la place $i + 3$ il n'y a pas d'indice i .

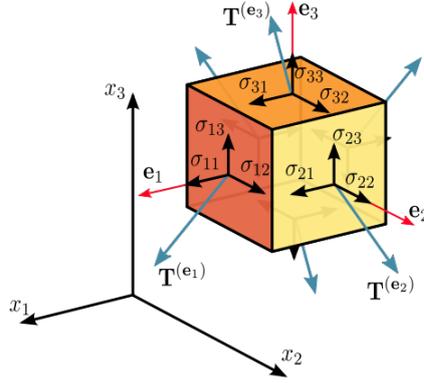


Figure 1.1 – Tenseur de contrainte pour un élément 3D de matière [Wikipédia (2016)].

Pour définir l'ensemble des tractions appliquées sur un élément de matière, il faut ainsi considérer les contraintes appliquées sur chacune de ses faces et dans chaque direction. Considérons par exemple, la face perpendiculaire à la direction x d'un élément de matière de volume $dv = dx dy dz$ et regardons ce qui se passe dans cette direction uniquement. La surface de cette face est définie par $ds = dy dz$. Si nous appliquons une force dans la direction x , nous devons considérer ce qui se passe sur la face positionnée en x , mais également la traction induite sur la face positionnée en $(x - dx)$, c'est-à-dire que nous allons considérer la composante σ_{11} (ou σ_{xx}) en deux points de l'espace (x, y, z) et $(x - dx, y, z)$. En considérant le repère local $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$ de l'élément de matière, nous avons alors comme traction appliquée à la surface ds :

$$\begin{aligned} \vec{T} ds &= \sigma_{11}(x, y, z) dy dz \vec{e}_1 - \sigma_{11}(x - dx, y, z) dy dz \vec{e}_1 \\ &= (\sigma_{11}(x, y, z) - \sigma_{11}(x - dx, y, z)) dy dz \vec{e}_1 \\ &= \frac{\partial \sigma_{11}}{\partial x} dx dy dz \vec{e}_1 \\ &= \frac{\partial \sigma_{11}}{\partial x} dv \vec{e}_1 \end{aligned}$$

Ainsi en considérant toutes les directions pour cette face perpendiculaire à la direction x , nous obtenons comme traction :

$$\vec{T} ds = \frac{\partial \sigma_{11}}{\partial x} dv \vec{e}_1 + \frac{\partial \sigma_{12}}{\partial y} dv \vec{e}_2 + \frac{\partial \sigma_{13}}{\partial z} dv \vec{e}_3$$

Si nous considérons maintenant toutes les faces de notre élément de matière, l'ensemble des forces de traction qui lui sont appliquées peut alors s'exprimer comme la somme entre les forces appliquées sur la face perpendiculaire à la direction x , les forces appliquées sur la face perpendiculaire à la direction y , et celles appliquées à la face perpendiculaire à la direction z . Nous avons ainsi ⁵ :

$$\begin{aligned} \vec{T} ds &= \frac{\partial \sigma_{11}}{\partial x} dv \vec{e}_1 + \frac{\partial \sigma_{12}}{\partial y} dv \vec{e}_2 + \frac{\partial \sigma_{13}}{\partial z} dv \vec{e}_3 \\ &+ \frac{\partial \sigma_{21}}{\partial x} dv \vec{e}_1 + \frac{\partial \sigma_{22}}{\partial y} dv \vec{e}_2 + \frac{\partial \sigma_{23}}{\partial z} dv \vec{e}_3 \\ &+ \frac{\partial \sigma_{31}}{\partial x} dv \vec{e}_1 + \frac{\partial \sigma_{32}}{\partial y} dv \vec{e}_2 + \frac{\partial \sigma_{33}}{\partial z} dv \vec{e}_3 \\ &= \vec{div} \bar{\sigma} dv. \end{aligned} \tag{1.6}$$

5. Petit rappel d'analyse vectorielle : si nous considérons \mathbb{A} un tenseur d'ordre 2,
 $\vec{div} \mathbb{A} = \left(\frac{\partial A_{11}}{\partial x} + \frac{\partial A_{12}}{\partial y} + \frac{\partial A_{13}}{\partial z} \right) \vec{e}_1 + \left(\frac{\partial A_{21}}{\partial x} + \frac{\partial A_{22}}{\partial y} + \frac{\partial A_{23}}{\partial z} \right) \vec{e}_2 + \left(\frac{\partial A_{31}}{\partial x} + \frac{\partial A_{32}}{\partial y} + \frac{\partial A_{33}}{\partial z} \right) \vec{e}_3.$

Nous avons alors défini la somme des tractions appliquées sur un élément en fonction du tenseur de contrainte et du volume de l'élément considéré. A noter que si nous considérons le cas particulier illustré par la Fig. 1.2, où la même force de pression est appliquée sur chacune des faces de notre élément et que ces forces sont perpendiculaires aux faces (c'est-à-dire que nous n'avons pas de cisaillement), le tenseur de contrainte se résume à :

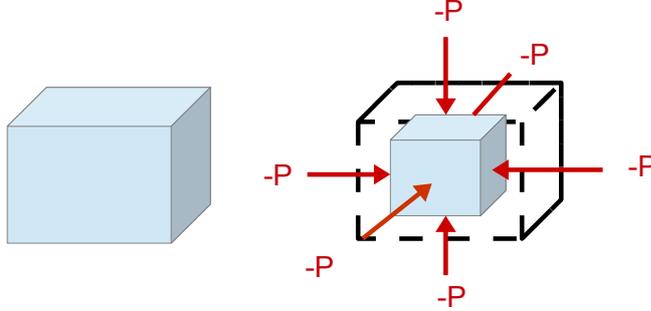


Figure 1.2 – Pression homogène appliquée sur un élément de matière.

$$\bar{\sigma} = \begin{pmatrix} -P & 0 & 0 \\ 0 & -P & 0 \\ 0 & 0 & -P \end{pmatrix} = \begin{pmatrix} \sigma_0 & 0 & 0 \\ 0 & \sigma_0 & 0 \\ 0 & 0 & \sigma_0 \end{pmatrix}. \quad (1.7)$$

Dans ce cas particulier, nous avons alors⁶ :

$$\overrightarrow{div} \bar{\sigma} = \frac{\partial \sigma_0}{\partial x} \vec{e}_1 + \frac{\partial \sigma_0}{\partial y} \vec{e}_2 + \frac{\partial \sigma_0}{\partial z} \vec{e}_3 = \overrightarrow{grad} \sigma_0 = \nabla \sigma_0. \quad (1.8)$$

1.4 Identification de l'équation à résoudre

En nous replaçant dans le cas général et en considérant un élément de volume dv situé au bord externe de l'objet (c'est-à-dire soumis aux forces de tractions), l'équation (1.2) peut s'écrire de la façon suivante pour cet élément de volume :

$$\rho dv \frac{d}{dt} \vec{v}(t) = \rho \overrightarrow{f_{ext}} dv + \overrightarrow{div} \bar{\sigma} dv.$$

Après simplification par le volume dv et en supposant que le vecteur $\overrightarrow{f_{ext}}$ est connu (constituant les données du problème), nous obtenons pour cet élément de volume :

$$\rho \frac{d}{dt} \vec{v}(t) = \rho \overrightarrow{f_{ext}} + \overrightarrow{div} \bar{\sigma}. \quad (1.9)$$

Ainsi si cet élément de volume n'est soumis qu'à la force volumique de la gravité, l'équation suivante régit le mouvement de l'objet :

$$\rho \frac{d}{dt} \vec{v}(t) = \rho \vec{g} + \overrightarrow{div} \bar{\sigma}. \quad (1.10)$$

Le mouvement est ainsi décrit par une équation vectorielle qui s'ajoute à l'équation scalaire de conservation de la masse, soit un total de $3 + 1 = 4$ équations. Les inconnues

6. Autre rappel d'analyse vectorielle : pour une fonction f , $\overrightarrow{grad} f = \nabla f = \frac{\partial f}{\partial x} \vec{e}_1 + \frac{\partial f}{\partial y} \vec{e}_2 + \frac{\partial f}{\partial z} \vec{e}_3$.

du problème sont la masse volumique ρ (le volume de l'objet peut changer au cours du mouvement si nous ne considérons pas un objet incompressible), le vecteur vitesse \vec{v} et le tenseur de contraintes $\bar{\sigma}$. Nous avons ainsi un total de $1 + 3 + 6 = 10$ inconnues. Il manque donc 6 équations pour résoudre le problème. Pour les obtenir, il faut relier le vecteur vitesse \vec{v} au tenseur des contraintes $\bar{\sigma}$ pour chaque point matériel.

De manière plus générale, il faut réussir à faire le lien entre les contraintes modélisées par le tenseur $\bar{\sigma}$ en chaque point matériel \mathbf{X} (représenté par un élément de volume élémentaire) et la transformation $\phi(\mathbf{X}, t)$. Nous verrons que c'est la loi de comportement qui permettra de faire ce lien, sachant qu'elle est définie pour un matériau donné (c'est-à-dire qu'il n'existe pas de loi de comportement universelle).

1.5 Cas des objets élastiques : déformation et déplacement

Nous analysons le cas particulier des objets élastiques qui sont caractérisés par le fait qu'ils retournent à leur forme initiale une fois que nous relâchons la **contrainte**. Pour caractériser la **déformation** d'un objet élastique, nous allons considérer que cet objet est composé d'un **ensemble de points** et nous allons observer au cours du temps la différence de **déplacement** qu'il y a entre un point de l'objet et un autre point.

En effet, imaginez dans votre main un objet déformable (une éponge par exemple). Vous la déplacez sans lui appliquer de contrainte sur ces points (une simple translation, c'est-à-dire un déplacement rigide). Cet objet n'est pas déformé. Maintenant si vous appuyez dessus, vous appliquez une contrainte qui crée une déformation de l'objet. Cette déformation est caractérisée par le fait que les points de deux bords de l'objet se sont rapprochés. Il y a donc eu une différence de déplacement entre des points de l'objet. Ensuite, cet état déformé est maintenu tant que vous ne relâchez pas la pression que vous appliquez sur cet objet. Vous pouvez déplacer cet objet en maintenant cet **état déformé** en effectuant une simple translation. Par contre, si vous relâchez cette pression, l'objet revient à son **état initial** (dans notre cas uniquement soumis à la force de gravité).

Lien entre déformation et déplacement de l'objet. Nous considérons l'objet au travers de deux états (ou configurations) : l'**état de référence** noté Ω_0 (qui correspond à l'état non-déformé au temps t_0 ou configuration initiale), et un **état déformé** noté Ω_t (qui correspond à l'état courant au temps t). La déformation est une relation entre deux configurations. Nous nous intéressons ainsi à la **déformation** ϕ permettant de passer de l'état de référence de l'objet à son état déformé. Elle permet ainsi de passer de la *position initiale* $\mathbf{X} \in \mathbb{R}^d$ d'un point matériel de l'objet de dimension d à sa *position courante* $\mathbf{x} \in \mathbb{R}^d$ au temps t avec $\mathbf{x} = \phi(\mathbf{X}, t)$. La position \mathbf{X} est prise dans l'état de référence et est constante dans le temps. Tandis que la position \mathbf{x} est prise dans l'état déformé et évolue dans le temps. Cette transformation est réversible dans le cas des objets élastiques avec $\mathbf{X} = \phi^{-1}(\mathbf{x}, t)$. En effet, $\phi(\mathbf{X}, t)$ est alors bijective et de classe C^1 .

Pour décrire le mouvement, nous pouvons choisir soit le couple (\mathbf{x}, t) appelé variables d'Euler, soit le couple (\mathbf{X}, t) appelé variables de Lagrange. La connaissance de la transformation ϕ ou de son inverse ϕ^{-1} définit alors complètement le mouvement. La Fig. 1.3 illustre ces notations en considérant $d = 3$.

Intéressons nous au **déplacement** des points **entre l'état de référence et l'état déformé**. Ce déplacement est défini par $\vec{u} = \mathbf{x} - \mathbf{X}$ pour un point donné de l'objet. Dans l'espace 3D, nous avons ainsi $u_i = x_i - X_i$ (pour $i = 1, 2, 3$) en considérant une position courante $\mathbf{x} = (x_1, x_2, x_3)^T$ et une position initiale $\mathbf{X} = (X_1, X_2, X_3)^T$. Nous avons ainsi $\phi(\mathbf{X}, t) = \mathbf{x} = \mathbf{X} + \vec{u}$ comme relation directe entre la transformation et le déplacement.

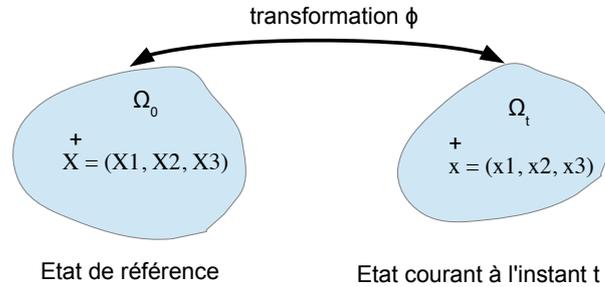
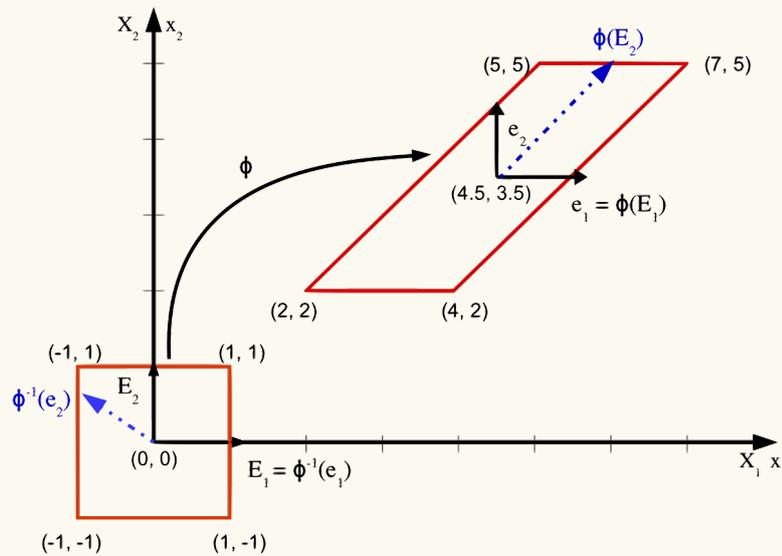


Figure 1.3 – Illustration de la transformation ϕ entre l'état de référence Ω_0 et l'état déformé Ω_t .

Exemple⁷. Pour illustrer ces notions, considérons un domaine 2D qui se déforme au cours du temps. La configuration de référence (carré) et celle déformée à l'instant $t = 1$ (parallélogramme) sont présentées sur la figure ci-dessous.



La transformation $\mathbf{x} = \phi(\mathbf{X}, t)$ qui permet de passer de la position initiale $\mathbf{X} = (X_1, X_2)^T$ (prise dans l'état de référence) à la position courante $\mathbf{x} = (x_1, x_2)^T$ (prise dans l'état déformé à l'instant t), s'écrit

$$\begin{cases} x_1 &= \frac{1}{4}(18t + 4X_1 + 6tX_2) \\ x_2 &= \frac{1}{4}(14t + (4 + 2t)X_2) \end{cases} \quad (1.11)$$

Les coordonnées spatiales (ou position courante) sont ainsi définies en fonction des coordonnées matérielles (ou position initiale) et du temps t .

Pour $t = 0$, nous avons bien $x_1 = X_1$ et $x_2 = X_2$.

7. Cet exemple est issu du cours de Mécanique des Milieux Continus de Nicolas Moës (2011).

Par ailleurs, nous pouvons observer que le point $\mathbf{X} = (0, 0)^T$ pris dans l'état de référence est transformé pour $t = 1$ en $\mathbf{x} = (4.5, 3.5)^T$ dans l'état déformé.

Le déplacement est défini dans cet exemple par

$$\begin{cases} u_1 &= x_1 - X_1 = \frac{1}{4}(18t + 6t X_2) \\ u_2 &= x_2 - X_2 = \frac{1}{4}(14t + 2t X_2) \end{cases} \quad (1.12)$$

Gradient des grandeurs physiques. Le gradient d'une quantité donne la façon dont cette quantité varie dans l'espace dans toutes les directions. Le **gradient du déplacement** est défini par $\mathbb{U} = \overline{\text{grad}} \vec{u} = \nabla \vec{u} = \partial \vec{u} / \partial \mathbf{X} \in \mathbb{R}^d \times \mathbb{R}^d$. Il va ainsi nous donner une indication sur la façon dont le déplacement entre des points de l'objet varie dans l'espace dans toutes les directions. Nous allons le relier au **gradient de la déformation** qui est défini par $\mathbb{F} = \nabla \phi(\mathbf{X}, t) = \partial \phi(\mathbf{X}, t) / \partial \mathbf{X} = \partial \mathbf{x} / \partial \mathbf{X} \in \mathbb{R}^d \times \mathbb{R}^d$, mesurant pour chaque direction de l'espace la variation du point \mathbf{x} par rapport à la variation subie par le point \mathbf{X} . \mathbb{F} représente ainsi la jacobienne du mouvement défini par la déformation ϕ . Dans l'espace 3D, nous avons $F_{ij} = \partial x_i / \partial X_j$ pour $i, j = 1, 2, 3$. et les composantes sont sans dimension correspondant à la dérivée d'une distance par rapport aux directions de l'espace.

Quelques exemples simples de champs de déformation⁸.

- Dans le cas où un objet subit une translation de vecteur \mathbf{w} entre l'état de référence et l'état déformé, nous avons :

$$\mathbf{x} = \phi(\mathbf{X}, \mathbf{t}) = \mathbf{X} + \mathbf{w} \quad ; \quad \mathbb{F} = \partial \phi(\mathbf{X}, t) / \partial \mathbf{X} = \mathbb{1}.$$

- Dans le cas où un objet subit un changement d'échelle d'un facteur γ entre l'état de référence et l'état déformé, nous avons :

$$\mathbf{x} = \phi(\mathbf{X}, \mathbf{t}) = \gamma \mathbf{X} \quad ; \quad \mathbb{F} = \partial \phi(\mathbf{X}, t) / \partial \mathbf{X} = \gamma \mathbb{1}.$$

- Dans le cas où un objet 2D subit un changement d'échelle d'un facteur 0.7 le long de l'axe horizontal et d'un facteur 2 dans le sens vertical, nous avons :

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \phi(\mathbf{X}, \mathbf{t}) = \phi \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} 0.7 X_1 \\ 2 X_2 \end{pmatrix} \quad ;$$

$$\mathbb{F} = \begin{pmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} \end{pmatrix} = \begin{pmatrix} 0.7 & 0 \\ 0 & 2 \end{pmatrix}.$$

Autrement dit \mathbb{F} mesure dans les trois directions de l'espace, les variations de longueurs de l'objet entre l'état déformé et l'état de référence. Pour cela, nous observons la déformation entre deux points infiniment proches se trouvant initialement à proximité du point \mathbf{X} . Cette déformation est donc caractérisée en comparant le vecteur \overrightarrow{dx} entre ces deux points pris dans l'état déformé par rapport au vecteur \overrightarrow{dX} entre ces deux points pris dans leur

8. Ces exemples sont issus du cours de [Sifakis and Barbic \(2012\)](#).

configuration initiale avec $\mathbb{F} = \overrightarrow{dx}/\overrightarrow{dX}$. D'un point de vue physique, \overrightarrow{dX} donne la direction d'une ligne tracée sur Ω_0 passant par \mathbf{X} qui se transforme dans la configuration actuelle Ω_t en une courbe dont la tangente au point \mathbf{x} est \overrightarrow{dX} .

Suite de l'exemple. Dans notre cas le gradient du déplacement (1.12) est défini par

$$\mathbb{U} = \begin{bmatrix} \frac{\partial u_1}{\partial X_1} & \frac{\partial u_1}{\partial X_2} \\ \frac{\partial u_2}{\partial X_1} & \frac{\partial u_2}{\partial X_2} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 3t \\ 0 & t \end{bmatrix}. \quad (1.13)$$

Le gradient de la déformation donnée par la relation (1.11) est défini par

$$\mathbb{F} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 3t \\ 0 & 2+t \end{bmatrix}.$$

Dans cet exemple, le gradient de la déformation \mathbb{F} est dit uniforme car il ne dépend pas de la position initiale (X_1, X_2) . Mais dans le cas général, il est courant que celui-ci dépende à la fois de la position initiale et du temps.

Les vecteurs \overrightarrow{E}_1 et \overrightarrow{E}_2 placés initialement selon les axes (X_1, X_2) sont transformés à l'instant $t = 1$ en $\mathbb{F} \cdot \overrightarrow{E}_1$ et $\mathbb{F} \cdot \overrightarrow{E}_2$ (en considérant la relation $\overrightarrow{dx} = \mathbb{F} \cdot \overrightarrow{dX}$) avec

$$\mathbb{F} \cdot \overrightarrow{E}_1 = \frac{1}{2} \begin{bmatrix} 2 & 3 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathbb{F} \cdot \overrightarrow{E}_2 = \frac{1}{2} \begin{bmatrix} 2 & 3 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}. \quad (1.14)$$

Ainsi, le vecteur \overrightarrow{E}_1 initialement parallèle à l'axe 1 reste parallèle à cet axe et ne change pas de taille. Par contre, le vecteur \overrightarrow{E}_2 initialement parallèle à l'axe 2 tourne de 45 degrés et voit sa taille multipliée par $3/\sqrt{2}$.

Si nous considérons au temps $t = 1$, les vecteurs \overrightarrow{e}_1 (parallèle à l'axe 1) et \overrightarrow{e}_2 (parallèle à l'axe 2), nous pouvons retrouver leurs orientations dans l'état de référence (en considérant la relation $\overrightarrow{dX} = \mathbb{F}^{-1} \cdot \overrightarrow{dx}$) avec

$$\mathbb{F}^{-1} = \frac{1}{2+t} \begin{bmatrix} 2+t & -3t \\ 0 & 2 \end{bmatrix},$$

nous donnant comme relations

$$\mathbb{F}^{-1} \cdot \overrightarrow{e}_1 = \frac{1}{3} \begin{bmatrix} 3 & -3 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathbb{F}^{-1} \cdot \overrightarrow{e}_2 = \frac{1}{3} \begin{bmatrix} 3 & -3 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ \frac{2}{3} \end{bmatrix}.$$

Nous pouvons également étudier la variation relative d'un volume dV de type parallélépipède formé par les trois vecteurs $\overrightarrow{dX}_1, \overrightarrow{dX}_2, \overrightarrow{dX}_3$. Après déformation, ce volume se transforme en dv qui est formé par les trois vecteurs $\overrightarrow{dx}_1, \overrightarrow{dx}_2, \overrightarrow{dx}_3$. Nous avons alors :

$$\frac{dv}{dV} = \frac{\overrightarrow{dx}_1 \cdot (\overrightarrow{dx}_2 \wedge \overrightarrow{dx}_3)}{\overrightarrow{dX}_1 \cdot (\overrightarrow{dX}_2 \wedge \overrightarrow{dX}_3)} = \frac{(\mathbb{F} \cdot \overrightarrow{dX}_1) \cdot ((\mathbb{F} \cdot \overrightarrow{dX}_2) \wedge (\mathbb{F} \cdot \overrightarrow{dX}_3))}{\overrightarrow{dX}_1 \cdot (\overrightarrow{dX}_2 \wedge \overrightarrow{dX}_3)}$$

$$\Rightarrow \frac{dv}{dV} = \frac{\det(\mathbb{F}) (\overrightarrow{dX}_1 \cdot (\overrightarrow{dX}_2 \wedge \overrightarrow{dX}_3))}{\overrightarrow{dX}_1 \cdot (\overrightarrow{dX}_2 \wedge \overrightarrow{dX}_3)} = \det(\mathbb{F}) = J$$

avec $J = \det(\mathbb{F})$ le jacobien de la transformation ϕ (c'est-à-dire le déterminant du gradient de la déformation ϕ défini par \mathbb{F}). A noter que selon la conservation de la masse, nous avons la relation $dm = \rho dv = \rho_0 dV$ donnant $dv/dV = \rho_0/\rho$. Ce qui amène au final $\rho_0 = J \rho$ avec ρ_0 la masse volumique de l'élément de volume dV sur la configuration de référence Ω_0 .

Il est également intéressant de relier le gradient de déplacement au gradient de déformation. Partons soit de la définition de la déformation ou soit de la définition du déplacement :

$$\begin{aligned} \mathbb{F} &= \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial \phi(\mathbf{X}, t)}{\partial \mathbf{X}} = \nabla \phi(\mathbf{X}, t) = \nabla(\mathbf{X} + \vec{u}) \\ &= \nabla \mathbf{X} + \nabla \vec{u} \\ &= \mathbb{1} + \mathbb{U} \\ \mathbb{U} &= \frac{\partial \vec{u}}{\partial \mathbf{X}} = \nabla \vec{u} = \nabla(\mathbf{x} - \mathbf{X}) = \nabla(\phi(\mathbf{X}, t) - \mathbf{X}) \\ &= \nabla \phi(\mathbf{X}, t) - \nabla \mathbf{X} \\ &= \mathbb{F} - \mathbb{1} \end{aligned}$$

Nous avons ainsi comme relations $\mathbb{F} = \mathbb{1} + \mathbb{U}$ ou $\mathbb{U} = \mathbb{F} - \mathbb{1}$ qui se notent ainsi dans l'espace 3D pour $i, j = 1, 2, 3$:

$$\frac{\partial x_i}{\partial X_j} = F_{ij} = \delta_{ij} + u_{ij}, \quad \frac{\partial u_i}{\partial X_j} = F_{ij} - \delta_{ij} \quad (1.15)$$

avec δ_{ij} le symbole de Kronecker défini par $\delta_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$

Suite de l'exemple. Nous retrouvons bien ces relations dans notre exemple avec

$$\begin{aligned} \mathbb{1} + \mathbb{U} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 & 3t \\ 0 & t \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 3t \\ 0 & 2+t \end{bmatrix} = \mathbb{F}. \\ \mathbb{F} - \mathbb{1} &= \frac{1}{2} \begin{bmatrix} 2 & 3t \\ 0 & 2+t \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 3t \\ 0 & t \end{bmatrix} = \mathbb{U}. \end{aligned}$$

Dans certains cas, la définition de \mathbb{F} basée sur l'observation de la variation du vecteur défini entre deux points, ne permet pas de représenter exactement l'état de déformation. En effet, alors que nous traitons un solide indéformable, nous pouvons obtenir un gradient \mathbb{F} différent de l'identité pour certains déplacements. Ce cas arrive notamment dans le cadre d'une rotation autour d'un axe. C'est pourquoi, des tenseurs plus complexes sont employés pour décrire de manière plus précise la déformation et ainsi obtenir une relation plus précise entre la déformation et le déplacement.

Tenseur de déformation. Pour quantifier la déformation, nous allons définir un tenseur de déformation. Ce tenseur noté $\bar{\epsilon}$ est sans dimension puisque la déformation n'a pas d'unité. C'est un tenseur symétrique (avec $\epsilon_{ij} = \epsilon_{ji}$) d'ordre 2 qui est donc diagonalisable dans une base orthonormée. Les valeurs propres sont alors appelées directions principales de

déformation. De la même façon que pour le tenseur de contrainte, ce tenseur possède 9 composantes dans l'espace $3D$ réduit par symétrie à 6 composants pour le définir avec :

$$\bar{\bar{\epsilon}} = \begin{pmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{yx} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{pmatrix} = \begin{pmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{pmatrix} = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{yz} \\ \epsilon_{xz} \\ \epsilon_{xy} \end{pmatrix} = \begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{13} \\ \epsilon_{12} \end{pmatrix}. \quad (1.16)$$

Le tenseur de déformation doit être nul pour un mouvement de corps rigide (puisque n'y a pas de variation entre deux points de l'objet lors de son mouvement) et en $\mathbb{F} = \mathbb{I}$ (c'est-à-dire dans la configuration initiale). Nous allons voir dans la suite les mesures de déformation les plus utilisées dans notre domaine.

Le **tenseur de déformation de Cauchy-Green droit** se calcule à partir du carré du vecteur \vec{dx} avec⁹ :

$$\begin{aligned} (\vec{dx})^2 &= \vec{dx} \cdot \vec{dx} = (\mathbb{F} \cdot \vec{dX}) \cdot (\mathbb{F} \cdot \vec{dX}) \\ &= (\mathbb{F}^T \mathbb{F} \cdot \vec{dX}) \cdot \vec{dX} \\ &= \vec{dX} \cdot (\mathbb{F}^T \mathbb{F} \cdot \vec{dX}) \end{aligned}$$

Le tenseur de Cauchy-Green droit est défini par $\epsilon^{cd} = \mathbb{F}^T \mathbb{F}$. Nous obtenons ainsi bien $\epsilon^{cd} = \mathbb{I}$ quand il n'y a pas de variation de distance, car celle-ci n'est plus influencée par la rotation. Il définit une métrique sur la configuration de référence Ω_0 à partir de la métrique canonique sur la configuration actuelle Ω_t .

Suite de l'exemple. Pour $t = 1$, le tenseur de Cauchy-Green droit est défini dans notre cas par

$$\epsilon^{cd} = \mathbb{F}^T \mathbb{F} = \frac{1}{2} \begin{bmatrix} 2 & 0 \\ 3 & 3 \end{bmatrix} \cdot \frac{1}{2} \begin{bmatrix} 2 & 3 \\ 0 & 3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 3 \\ 3 & 9 \end{bmatrix}.$$

Le **tenseur de déformation de Cauchy-Green gauche** se calcule de la même façon à partir du carré du vecteur \vec{dX} avec¹⁰ :

$$\begin{aligned} (\vec{dX})^2 &= \vec{dX} \cdot \vec{dX} = (\mathbb{F}^{-1} \cdot \vec{dx}) \cdot (\mathbb{F}^{-1} \cdot \vec{dx}) \\ &= ((\mathbb{F}^{-1})^T \mathbb{F}^{-1} \cdot \vec{dx}) \cdot \vec{dx} \\ &= \vec{dx} \cdot ((\mathbb{F}^{-1})^T \mathbb{F}^{-1} \cdot \vec{dx}) \\ &= \vec{dx} \cdot ((\mathbb{F}^T)^{-1} \mathbb{F}^{-1} \cdot \vec{dx}) \\ &= \vec{dx} \cdot (\mathbb{F} \mathbb{F}^T)^{-1} \cdot \vec{dx} \end{aligned}$$

Le tenseur de Cauchy-Green gauche est ainsi défini par $\epsilon^{cg} = \mathbb{F} \mathbb{F}^T$. Il définit une métrique sur la configuration actuelle Ω_t à partir de la métrique canonique initiale sur Ω_0 . Les tenseurs de Cauchy-Green ϵ^{cg} et ϵ^{cd} sont des tenseurs symétriques ($(\epsilon^{cg})^T = \epsilon^{cg}$ et $(\epsilon^{cd})^T = \epsilon^{cd}$) définis positifs et sont appelés tenseurs des dilatations car ils permettent de mesurer l'allongement de la ligne \vec{dX} . Par contre, il faut bien noter que $\epsilon^{cg} \neq (\epsilon^{cd})^T$.

9. A noter que nous avons utilisé la formule $\vec{v} \cdot \mathbb{A} \vec{u} = \mathbb{A}^T \vec{v} \cdot \vec{u}$ de l'algèbre tensorielle avec \vec{u}, \vec{v} des vecteurs et \mathbb{A} un tenseur d'ordre 2, en considérant dans notre cas $\vec{v} = \mathbb{F} \cdot \vec{dX}$, $\mathbb{A} = \mathbb{F}$ et $\vec{u} = \vec{dX}$.

10. Petits rappels d'algèbre linéaire : $(\mathbb{A}^{-1})^T = (\mathbb{A}^T)^{-1}$ et $\mathbb{B}^{-1} \mathbb{A}^{-1} = (\mathbb{A} \mathbb{B})^{-1}$.

Le **tenseur de déformation de Green-Lagrange** est calculé à partir de la différence entre les carrés de la configuration initiale et de la configuration déformée avec :

$$\begin{aligned}
 (\vec{dx})^2 - (\vec{dX})^2 &= \vec{dX} \cdot (\epsilon^{cd} \cdot \vec{dX}) - (\vec{dX} \cdot \vec{dX}) \\
 &= \vec{dX} \cdot (\epsilon^{cd} \vec{dX} - \vec{dX}) \\
 &= \vec{dX} \cdot (\epsilon^{cd} - \mathbb{1}) \cdot \vec{dX} \\
 &= 2 \vec{dX} \cdot \left(\frac{1}{2} (\epsilon^{cd} - \mathbb{1}) \cdot \vec{dX} \right)
 \end{aligned}$$

Le tenseur de Green-Lagrange est défini par $\epsilon^g = \frac{1}{2}(\epsilon^{cd} - \mathbb{1}) = \frac{1}{2}(\mathbb{F}^T \mathbb{F} - \mathbb{1})$ donnant la relation

$$\frac{1}{2} [(\vec{dx})^2 - (\vec{dX})^2] = \vec{dX} \cdot \epsilon^g \cdot \vec{dX}.$$

A noter qu'il est ainsi bien nul pour $\mathbb{F} = \mathbb{1}$ c'est-à-dire dans la configuration initiale. Si nous décomposons les vecteurs \vec{dX} et \vec{dx} selon leurs normales et orientations, nous avons $\vec{dX} = dL \vec{N}$ et $\vec{dx} = dl \vec{n}$ avec dL et dl les normes des vecteurs considérés, \vec{N} et \vec{n} leurs vecteurs élémentaires. Nous obtenons ainsi :

$$\begin{aligned}
 \frac{1}{2} [(dl \vec{n} \cdot dl \vec{n}) - (dL \vec{N} \cdot dL \vec{N})] &= dL \vec{N} \cdot \epsilon^g \cdot dL \vec{N} \\
 \Rightarrow \frac{1}{2} [dl^2 - dL^2] &= dL \vec{N} \cdot \epsilon^g \cdot dL \vec{N} \\
 \Rightarrow \frac{1}{2} \frac{dl^2 - dL^2}{dL^2} &= \vec{N} \cdot \epsilon^g \cdot \vec{N}.
 \end{aligned}$$

Si nous considérons le vecteur \vec{N} selon la direction x dans le repère initial, nous avons

$$\frac{1}{2} \frac{dl^2 - dL^2}{dL^2} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_{11}^g & \epsilon_{12}^g & \epsilon_{13}^g \\ \epsilon_{21}^g & \epsilon_{22}^g & \epsilon_{23}^g \\ \epsilon_{31}^g & \epsilon_{32}^g & \epsilon_{33}^g \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \epsilon_{11}^g.$$

De même nous obtenons selon les deux autres directions

$$\begin{aligned}
 \frac{1}{2} \frac{dl^2 - dL^2}{dL^2} &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_{11}^g & \epsilon_{12}^g & \epsilon_{13}^g \\ \epsilon_{21}^g & \epsilon_{22}^g & \epsilon_{23}^g \\ \epsilon_{31}^g & \epsilon_{32}^g & \epsilon_{33}^g \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \epsilon_{22}^g, \\
 \frac{1}{2} \frac{dl^2 - dL^2}{dL^2} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \epsilon_{11}^g & \epsilon_{12}^g & \epsilon_{13}^g \\ \epsilon_{21}^g & \epsilon_{22}^g & \epsilon_{23}^g \\ \epsilon_{31}^g & \epsilon_{32}^g & \epsilon_{33}^g \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \epsilon_{33}^g.
 \end{aligned}$$

Les éléments diagonaux du tenseur de déformation de Green-Lagrange donnent ainsi les changements relatifs de longueur des vecteurs élémentaires parallèles aux directions x, y, z de l'état de référence (en considérant la différence entre les carrés de la configuration initiale et de la configuration déformée). Ce tenseur de déformation appelé **tenseur de Green-Lagrange ou de Green ou de Green-Saint-Venant** est largement employé dans la pratique pour quantifier l'intensité de la direction de déformation.

Suite de l'exemple. Pour $t = 1$, le tenseur de Green-Lagrange est défini dans notre cas par

$$\epsilon^g = \frac{1}{2}(\epsilon^{cd} - \mathbb{1}) = \frac{1}{2} \frac{1}{2} \begin{bmatrix} 2 & 3 \\ 3 & 9 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 0 & 3 \\ 3 & 7 \end{bmatrix}.$$

La composante ϵ_{11}^g est ainsi nulle indiquant qu'un vecteur unitaire placé selon l'axe 1 dans l'état de référence ne voit pas sa taille évoluer à $t = 1$. Ceci est en accord avec le vecteur $\mathbb{F} \cdot \vec{E}_1$ obtenu dans la relation (1.14) ayant une norme égale à celle de \vec{E}_1 .

Considérons désormais le vecteur \vec{E}_2 de l'état de référence. Sa norme initiale est égale à 1 (soit $dL = 1$). Selon la relation (1.14), il s'est transformé dans l'état déformé à $t = 1$ en $[1.5 \ 1.5]^T$ (soit $dl = 3/\sqrt{2}$). Nous avons ainsi

$$\frac{1}{2} \frac{dl^2 - dL^2}{dL^2} = \frac{1}{2} \frac{\frac{9}{2} - 1}{1} = \frac{7}{4} = \vec{E}_2 \cdot \epsilon^g \cdot \vec{E}_2 = \epsilon_{22}^g.$$

Ceci est cohérent avec la valeur $7/4$ calculée précédemment pour la composante ϵ_{22}^g .

Pour obtenir la formulation du tenseur de Green-Lagrange en fonction du déplacement, nous utilisons la relation $\mathbb{F} = \mathbb{1} + \mathbb{U}$:

$$\begin{aligned} \epsilon^g &= \frac{1}{2}(\epsilon^{cd} - \mathbb{1}) = \frac{1}{2}(\mathbb{F}^T \mathbb{F} - \mathbb{1}) \\ &= \frac{1}{2}((\mathbb{1} + \mathbb{U})^T \cdot (\mathbb{1} + \mathbb{U}) - \mathbb{1}) \\ &= \frac{1}{2}((\mathbb{1} + \mathbb{U})^T + (\mathbb{1} + \mathbb{U})^T \cdot \mathbb{U} - \mathbb{1}) \\ &= \frac{1}{2}(\mathbb{U}^T + \mathbb{U} + \mathbb{U}^T \cdot \mathbb{U}) \end{aligned}$$

Nous obtenons au final (pour $i, j = 1, 2, 3$) :

$$\epsilon^g = \frac{1}{2}(\mathbb{U}^T \cdot \mathbb{U}) + \frac{1}{2}(\mathbb{U}^T + \mathbb{U}), \quad \epsilon_{ij}^g = \frac{1}{2} \sum_{k=1}^3 \left(\frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right) + \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} \right). \quad (1.17)$$

Ce tenseur présente un terme non-linéaire $\frac{1}{2}(\mathbb{U}^T \cdot \mathbb{U})$ et un terme linéaire $\frac{1}{2}(\mathbb{U}^T + \mathbb{U})$. Dans le cadre de **petites perturbations avec de petits déplacements**, nous pouvons considérer uniquement la partie linéaire de ce tenseur définissant ainsi une relation linéaire (c'est-à-dire une relation d'ordre 1) entre le déplacement et les déformations. Nous parlons alors de "linéarité géométrique". Le tenseur linéarisé est alors défini par :

$$\epsilon^g = \frac{1}{2}(\mathbb{U}^T + \mathbb{U}), \quad \epsilon_{ij}^g = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} \right) \text{ pour } i, j = 1, 2, 3. \quad (1.18)$$

Nous pouvons remarquer que les termes diagonaux de ce tenseur linéarisé représentent l'allongement relatif dans la direction \vec{e}_i avec :

$$\epsilon_{ii}^g = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_i} + \frac{\partial u_i}{\partial X_i} \right) = \frac{\partial u_i}{\partial X_i}.$$

Suite de l'exemple. Dans notre cas, si nous prenons la formulation du tenseur de Green-Lagrange en fonction du déplacement, nous obtenons :

$$\begin{aligned}\epsilon_{11}^g &= \frac{1}{2} \left(\frac{\partial u_1}{\partial X_1} \frac{\partial u_1}{\partial X_1} \right) + \frac{1}{2} \left(\frac{\partial u_2}{\partial X_1} \frac{\partial u_2}{\partial X_1} \right) + \frac{1}{2} \left(\frac{\partial u_1}{\partial X_1} + \frac{\partial u_1}{\partial X_1} \right), \\ \epsilon_{12}^g &= \frac{1}{2} \left(\frac{\partial u_1}{\partial X_1} \frac{\partial u_1}{\partial X_2} \right) + \frac{1}{2} \left(\frac{\partial u_2}{\partial X_1} \frac{\partial u_2}{\partial X_2} \right) + \frac{1}{2} \left(\frac{\partial u_1}{\partial X_2} + \frac{\partial u_2}{\partial X_1} \right), \\ \epsilon_{21}^g &= \frac{1}{2} \left(\frac{\partial u_1}{\partial X_2} \frac{\partial u_1}{\partial X_1} \right) + \frac{1}{2} \left(\frac{\partial u_2}{\partial X_2} \frac{\partial u_2}{\partial X_1} \right) + \frac{1}{2} \left(\frac{\partial u_2}{\partial X_1} + \frac{\partial u_1}{\partial X_2} \right), \\ \epsilon_{22}^g &= \frac{1}{2} \left(\frac{\partial u_1}{\partial X_2} \frac{\partial u_1}{\partial X_2} \right) + \frac{1}{2} \left(\frac{\partial u_2}{\partial X_2} \frac{\partial u_2}{\partial X_2} \right) + \frac{1}{2} \left(\frac{\partial u_2}{\partial X_2} + \frac{\partial u_2}{\partial X_2} \right).\end{aligned}$$

Ce qui nous donne :

$$\epsilon^g = \begin{bmatrix} 0 & 3t/4 \\ 3t/4 & t/2 + 5t^2/4 \end{bmatrix}.$$

Pour $t = 1$, nous retrouvons le tenseur ϵ^g calculé précédemment en utilisant la formulation en fonction du gradient de la déformation.

Le tenseur de Green-Lagrange linéarisé est défini dans notre cas par

$$\epsilon^g = \begin{bmatrix} \frac{1}{2} \left(\frac{\partial u_1}{\partial X_1} + \frac{\partial u_1}{\partial X_1} \right) & \frac{1}{2} \left(\frac{\partial u_1}{\partial X_2} + \frac{\partial u_2}{\partial X_1} \right) \\ \frac{1}{2} \left(\frac{\partial u_2}{\partial X_1} + \frac{\partial u_1}{\partial X_2} \right) & \frac{1}{2} \left(\frac{\partial u_2}{\partial X_2} + \frac{\partial u_2}{\partial X_2} \right) \end{bmatrix} = \begin{bmatrix} 0 & 3t/4 \\ 3t/4 & t/2 \end{bmatrix}.$$

Nous pouvons noter que le tenseur de déformation ne prend en compte que les aspects géométriques de la déformation et ne tient pas compte des aspects mécaniques, puisque sa définition ne comporte aucune caractéristique mécanique du matériau considéré.

1.6 Loi de comportement de l'objet

Il nous reste maintenant à faire le lien entre les contraintes appliquées sur l'objet, qui sont modélisées par le tenseur de contraintes $\bar{\sigma}$, et la déformation engendrée issue de cette contrainte, qui est modélisée par le tenseur de déformation $\bar{\epsilon}$. C'est la loi de comportement du matériau qui donne ce lien. Cette relation est décrite de manière générale par

$$\bar{\sigma} = \frac{\partial W(\bar{\epsilon})}{\partial \bar{\epsilon}} \quad (1.19)$$

où $W(\bar{\epsilon})$ correspond à l'énergie de déformation. Cette relation n'est pas universelle mais va dépendre du matériau. Elle est mise en avant en observant la réponse du matériau à des sollicitations diverses telles que des essais uni-axiaux (traction, cisaillement).

Trois classes de comportement mécanique se distinguent principalement ¹¹ :

- **L'élasticité.** Le comportement élastique linéarisé uni-axial est symbolisé par un ressort dont l'allongement Δl est proportionnel à la charge appliquée \vec{F} (en traction

11. Je reprends ici les explications du cours de [Forest et al. \(2010\)](#) sur les lois de comportement.

ou compression). Ceci est valable dans un certain domaine de charge dit domaine d'élasticité. Le facteur de proportionnalité est la souplesse (ou compliance) $1/k$ avec k la raideur du ressort (en N.m^{-1} ou Pa.m). Nous avons alors comme relation linéaire entre ces quantités $\vec{F} = k \Delta l$.

- **La viscosité.** Le comportement visqueux uni-axial est celui d'un amortisseur ou d'un piston dont la réponse dépend de la vitesse de sollicitation. Dans le cas linéaire nous avons comme relation $\vec{F} = \eta \dot{\Delta l}$ avec η le coefficient de viscosité de l'amortisseur (en $\text{N.m}^{-1}.\text{s}$ ou Pa.m.s). Dans ce contexte, plus la vitesse de déformation est rapide plus la force est grande.
- **La plasticité.** La déformation plastique est associée à de la déformation élastique. Au départ, le matériau commence par se déformer de manière réversible, c'est-à-dire qu'il va reprendre sa forme initiale une fois que les sollicitations s'arrêtent (déformation élastique). Puis au delà d'un certain seuil de contrainte, la déformation est définitive c'est-à-dire que le matériau reste déformé une fois les sollicitations terminées. Par exemple un ressort va se déformer définitivement lorsque la force de traction dépasse sa capacité d'allongement élastique. Le comportement plastique uni-axial peut se modéliser comme un frottement solide avec une résistance à la déformation au départ. Le modèle correspond à l'ajout d'une force constante \vec{F}_0 lorsque le glissement a lieu avec $\vec{F} = \vec{F}_0(\dot{\Delta l})$.

En combinant ces comportements mécaniques élémentaires, nous obtenons des comportements plus complexes comme la viscoélasticité, l'élastoplasticité, la viscoplasticité et l'élastoviscoplasticité. Dans le domaine de la simulation d'organes, nous avons vu que nous nous concentrons essentiellement sur le comportement élastique et visqueux des matériaux. C'est-à-dire que dans notre cas, l'objet retourne à sa forme initiale une fois les contraintes appliquées sur celui-ci relâchées. Par contre différentes lois sont alors mises en avant selon la façon dont l'objet retourne à cet état d'équilibre. La Fig. 1.4 illustre trois comportements : (a) élastique linéaire, (b) élastique non linéaire (ou hyper-élastique) et (c) visco-élastique non linéaire. Dans le cas linéaire, nous avons logiquement une relation linéaire (une droite) entre la contrainte appliquée et la déformation induite, tandis que dans le cas non linéaire nous obtenons une courbe entre ces deux quantités.

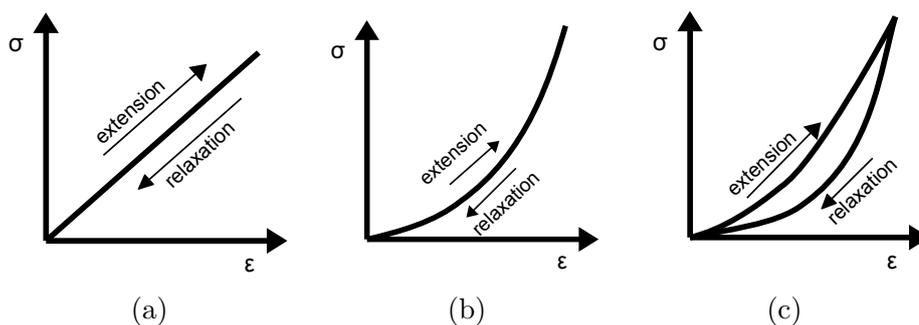


Figure 1.4 – Courbe contrainte-déformation pour différentes lois de comportement $\bar{\sigma} = f(\epsilon)$.

La Fig. 1.5 illustre la réponse d'un matériau aux sollicitations avec une courbe contrainte-déformation. Nous avons schématisé un comportement élastique linéaire en petites déformations puis une phase de plasticité quand la limite élastique du matériau est atteinte et que les sollicitations deviennent trop importantes. Puis si les sollicitations continuent, la limite de rupture peut ensuite être atteinte.

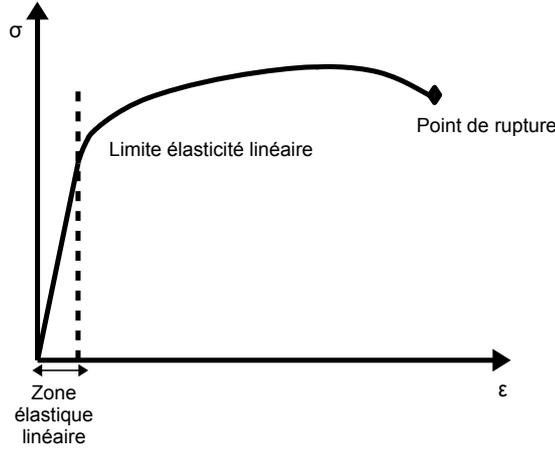


Figure 1.5 – Représentation de la courbe contrainte-déformation d'un matériau.

Cas de l'élasticité linéaire

Concentrons-nous sur le cas de l'élasticité linéaire en nous plaçant en petites déformations. Nous avons alors une loi de comportement linéaire avec une relation d'ordre 1 entre les quantités $\bar{\sigma}$ et $\bar{\epsilon}$. Nous parlons alors de "linéarité physique ou matérielle". Cette relation est ainsi du type :

$$\bar{\sigma} = \overset{\equiv}{\overset{\equiv}{\overset{\equiv}{C}}} : \bar{\epsilon}, \quad \sigma_{ij} = \sum_{k=1}^3 \sum_{l=1}^3 C_{ijkl} \epsilon_{kl}, \quad i, j = 1, 2, 3,$$

où $\overset{\equiv}{\overset{\equiv}{\overset{\equiv}{C}}}$ est un tenseur d'ordre 4 (avec $3^4 = 81$ termes) qui spécifie les coefficients élastiques du matériau, c'est-à-dire que ces coefficients indiquent la façon dont le matériau se déforme dans une direction donnée pour une contrainte appliquée sur celui-ci. Les composants de ce tenseur ont la même dimension qu'une pression puisque que ceux de $\bar{\epsilon}$ sont sans dimension.

Comme $\bar{\epsilon}$ et $\bar{\sigma}$ sont des tenseurs symétriques, nous avons une symétrie en i, j avec $C_{ijkl} = C_{jikl}$ et une symétrie en k, l avec $C_{ijkl} = C_{ijlk}$ et nous avons également $C_{ijkl} = C_{klij}$. Au final, nous obtenons 21 termes pour décrire le comportement du matériau. Pour simplifier l'écriture, la convention de sommation d'Einstein¹² (ou convention de l'indice muet répété) peut être employée. La relation précédente s'écrit ainsi de la façon suivante¹³ :

$$\bar{\sigma} = \overset{\equiv}{\overset{\equiv}{\overset{\equiv}{C}}} : \bar{\epsilon}, \quad \sigma_{ij} = C_{ijkl} \epsilon_{kl}, \quad i, j = 1, 2, 3. \quad (1.20)$$

Du fait de ces propriétés de symétrie, le tenseur $\overset{\equiv}{\overset{\equiv}{\overset{\equiv}{C}}}$ peut être représenté sous la forme d'une matrice 6×6 où les directions représentent les directions de déformation. Autrement dit, la liaison linéaire ente les tenseurs symétriques $\bar{\sigma}$ et $\bar{\epsilon}$ n'ayant chacun que 6 composantes indépendantes peut être réalisée à l'aide de 36 termes seulement. En adoptant la notation

12. La convention de sommation d'Einstein spécifie que quand un indice apparaît deux fois dans la même expression, nous lui faisons prendre les valeurs 1, 2, 3 et nous sommons.

13. Notons que dans cette expression, les indices i, j ont une valeur fixée (1, 2 ou 3) alors que les indices k et l prennent les 3 valeurs (1, 2 et 3). Nous parlons alors d'indice franc pour i et j et d'indice muet pour k et l correspondant à l'indice répété sur lequel la sommation est effectuée.

de Voigt pour simplifier l'écriture, nous obtenons :

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix} = \begin{pmatrix} C_{1111} & C_{1122} & C_{1133} & C_{1123} & C_{1113} & C_{1112} \\ C_{2211} & C_{2222} & C_{2233} & C_{2223} & C_{2213} & C_{2212} \\ C_{3311} & C_{3322} & C_{3333} & C_{3323} & C_{3313} & C_{3312} \\ C_{2311} & C_{2322} & C_{2333} & C_{2323} & C_{2313} & C_{2312} \\ C_{1311} & C_{1322} & C_{1333} & C_{1323} & C_{1313} & C_{1312} \\ C_{1211} & C_{1222} & C_{1233} & C_{1223} & C_{1213} & C_{1212} \end{pmatrix} \begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{13} \\ \epsilon_{12} \end{pmatrix}. \quad (1.21)$$

Nous pouvons également présenter cette relation sous sa **forme inversée**, c'est-à-dire en présentant les déformations en fonction des contraintes, avec

$$\bar{\bar{\bar{\epsilon}}} = \bar{\bar{\bar{D}}} : \bar{\bar{\sigma}}, \quad \epsilon_{ij} = D_{ijkl} \sigma_{kl}, \quad i, j = 1, 2, 3 \quad (1.22)$$

où le tenseur $\bar{\bar{\bar{D}}}$ d'ordre 4 possède les mêmes symétries que $\bar{\bar{C}}$ (i.e. $D_{ijkl} = D_{jikl} = D_{ijlk} = D_{klij}$), ce qui nous donne en notation de Voigt la relation suivante :

$$\begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{13} \\ \epsilon_{12} \end{pmatrix} = \begin{pmatrix} D_{1111} & D_{1122} & D_{1133} & D_{1123} & D_{1113} & D_{1112} \\ D_{2211} & D_{2222} & D_{2233} & D_{2223} & D_{2213} & D_{2212} \\ D_{3311} & D_{3322} & D_{3333} & D_{3323} & D_{3313} & D_{3312} \\ D_{2311} & D_{2322} & D_{2333} & D_{2323} & D_{2313} & D_{2312} \\ D_{1311} & D_{1322} & D_{1333} & D_{1323} & D_{1313} & D_{1312} \\ D_{1211} & D_{1222} & D_{1233} & D_{1223} & D_{1213} & D_{1212} \end{pmatrix} \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix}.$$

Le tenseur $\bar{\bar{\bar{C}}}$ permettant de passer des déformations aux contraintes est appelé tenseur de rigidité, et le tenseur $\bar{\bar{\bar{D}}}$ permettant de passer des contraintes aux déformations est appelée matrice de compliance.

Formulation de la loi de Hooke en fonction de λ et μ . Dans le cas d'un **matériau élastique isotrope**, c'est-à-dire dont le comportement est le même dans toutes les directions, seules deux constantes indépendantes sont nécessaires (au lieu de 21!). La relation (1.21) se présente alors sous la forme suivante

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix} = \begin{pmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\mu \end{pmatrix} \begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{13} \\ \epsilon_{12} \end{pmatrix}. \quad (1.23)$$

Nous obtenons ainsi les équations dites de Lamé :

$$\begin{cases} \sigma_{11} = \lambda(\epsilon_{11} + \epsilon_{22} + \epsilon_{33}) + 2\mu \epsilon_{11} \\ \sigma_{22} = \lambda(\epsilon_{11} + \epsilon_{22} + \epsilon_{33}) + 2\mu \epsilon_{22} \\ \sigma_{33} = \lambda(\epsilon_{11} + \epsilon_{22} + \epsilon_{33}) + 2\mu \epsilon_{33} \\ \sigma_{23} = 2\mu \epsilon_{23} \\ \sigma_{13} = 2\mu \epsilon_{13} \\ \sigma_{12} = 2\mu \epsilon_{12} \end{cases} \quad (1.24)$$

Ces équations peuvent être écrites de manière générale (appelée **loi de Hooke**) par

$$\bar{\sigma} = \lambda \text{Tr}(\bar{\epsilon}) \mathbb{1} + 2 \mu \bar{\epsilon}, \quad \sigma_{ij} = \lambda \epsilon_{kk} \delta_{ij} + 2 \mu \epsilon_{ij} \text{ pour } i, j = 1, 2, 3. \quad (1.25)$$

Le tenseur \bar{C} de la relation (1.20) est alors défini de la façon suivante :

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + 2 \mu \delta_{ik} \delta_{jl}.$$

Paramètres du matériau. λ et μ sont deux paramètres matériaux scalaires appelés **coefficients de Lamé**. Ils sont définis en fonction du **module de Young** E (ou module d'élasticité) du matériau et de son **coefficient de Poisson** ν avec

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

E est exprimé en unité de pression, tandis que ν est sans unité.

Trace du tenseur de déformation. Dans le cas où nous considérons le tenseur de déformation de Green-Lagrange linéarisé, la trace de ce tenseur présent dans la relation (1.25) est¹⁴ :

$$\begin{aligned} \text{Tr}(\bar{\epsilon}) &= \epsilon_{11}^g + \epsilon_{22}^g + \epsilon_{33}^g \\ &= \frac{\partial u_1}{\partial X_1} + \frac{\partial u_2}{\partial X_2} + \frac{\partial u_3}{\partial X_3} \\ &= \frac{\partial V}{V} \\ &= \text{div } \vec{u} \end{aligned}$$

Elle représente ainsi le changement de volume de l'objet avec la somme des déformations subies dans les trois directions. $\text{div } \vec{u}$ peut alors être interprété comme la dilatation volumique relative dans l'hypothèse des petites transformations.

Formulation de la loi de Hooke en fonction de E et ν . Réciproquement, nous pouvons exprimer E et ν en fonction de λ et μ avec

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

Nous pouvons alors écrire la loi de Hooke en fonction de E et ν , donnant la relation suivante entre les tenseurs $\bar{\sigma}$, \bar{C} et $\bar{\epsilon}$:

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix} = \frac{E}{(1 + \nu)(1 - 2\nu)} \begin{pmatrix} 1 - \nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1 - \nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1 - \nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 - 2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 - 2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 - 2\nu \end{pmatrix} \begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{13} \\ \epsilon_{12} \end{pmatrix}.$$

14. En désignant par div l'opérateur divergence par rapport aux variables de Lagrange X_1, X_2, X_3 . Il est important de noter que l'opérateur divergence sur un tenseur d'ordre 2 donne un vecteur ($\text{div } \mathbb{A}$), tandis que sur un vecteur nous obtenons un scalaire ($\text{div } \vec{u}$).

Nous obtenons alors comme équations

$$\left\{ \begin{array}{l} \sigma_{11} = \frac{E}{1+\nu} \left[\epsilon_{11} + \frac{\nu}{1-2\nu} (\epsilon_{11} + \epsilon_{22} + \epsilon_{33}) \right] \\ \sigma_{22} = \frac{E}{1+\nu} \left[\epsilon_{22} + \frac{\nu}{1-2\nu} (\epsilon_{11} + \epsilon_{22} + \epsilon_{33}) \right] \\ \sigma_{33} = \frac{E}{1+\nu} \left[\epsilon_{33} + \frac{\nu}{1-2\nu} (\epsilon_{11} + \epsilon_{22} + \epsilon_{33}) \right] \\ \sigma_{23} = \frac{E}{1+\nu} \epsilon_{23} \\ \sigma_{13} = \frac{E}{1+\nu} \epsilon_{13} \\ \sigma_{12} = \frac{E}{1+\nu} \epsilon_{12} \end{array} \right. \quad (1.26)$$

Nous pouvons les écrire de manière générale sous la forme :

$$\bar{\bar{\sigma}} = \frac{E}{1+\nu} \left(\bar{\bar{\epsilon}} + \frac{\nu}{1-2\nu} \text{Tr}(\bar{\bar{\epsilon}}) \mathbb{1} \right), \quad \sigma_{ij} = \frac{E}{1+\nu} \left(\epsilon_{ij} + \frac{\nu}{1-2\nu} \epsilon_{kk} \delta_{ij} \right) \quad \text{pour } i, j = 1, 2, 3.$$

Le tenseur $\bar{\bar{\bar{C}}}$ de la relation (1.20) est alors défini par :

$$C_{ijkl} = \frac{E}{1+\nu} \left(\delta_{ik} \delta_{jl} + \frac{\nu}{1-2\nu} \delta_{ij} \delta_{kl} \right).$$

Formulation inversée de la loi de Hooke en fonction de de E et ν . Cette relation peut être présentée sous sa forme inversée avec comme relation entre $\bar{\bar{\sigma}}$, $\bar{\bar{\bar{C}}}$ et $\bar{\bar{\epsilon}}$:

$$\begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{13} \\ \epsilon_{12} \end{pmatrix} = \frac{1}{E} \begin{pmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1+\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1+\nu \end{pmatrix} \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix}.$$

Nous obtenons alors les équations appelées **relations de Young**

$$\left\{ \begin{array}{l} \epsilon_{11} = \frac{1}{E} [\sigma_{11} - \nu(\sigma_{22} + \sigma_{33})] \\ \epsilon_{22} = \frac{1}{E} [\sigma_{22} - \nu(\sigma_{11} + \sigma_{33})] \\ \epsilon_{33} = \frac{1}{E} [\sigma_{33} - \nu(\sigma_{11} + \sigma_{22})] \\ \epsilon_{23} = \frac{1+\nu}{E} \sigma_{23} \\ \epsilon_{13} = \frac{1+\nu}{E} \sigma_{13} \\ \epsilon_{12} = \frac{1+\nu}{E} \sigma_{12} \end{array} \right.$$

Elles s'écrivent de manière générale sous la forme :

$$\bar{\bar{\bar{\epsilon}}} = \frac{1+\nu}{E} \bar{\bar{\bar{\sigma}}} - \frac{\nu}{E} \text{Tr}(\bar{\bar{\bar{\sigma}}}) \mathbb{1}, \quad \epsilon_{ij} = \frac{1+\nu}{E} \sigma_{ij} - \frac{\nu}{E} \sigma_{kk} \delta_{ij} \text{ pour } i, j = 1, 2, 3. \quad (1.27)$$

Au final, le tenseur $\bar{\bar{\bar{D}}}$ de la relation (1.22) est défini par :

$$D_{ijkl} = \frac{1+\nu}{E} \delta_{ik} \delta_{jl} - \frac{\nu}{E} \delta_{ij} \delta_{kl}.$$

Formulation inversée de la loi de Hooke en fonction λ et μ . En fonction des coefficients de Lamé, cette forme inversée s'écrit :

$$\begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{13} \\ \epsilon_{12} \end{pmatrix} = \frac{\lambda}{2\mu(3\lambda+2\mu)} \begin{pmatrix} -2(\lambda+\mu) & -1 & -1 & 0 & 0 & 0 \\ -1 & -2(\lambda+\mu) & -1 & 0 & 0 & 0 \\ -1 & -1 & -2(\lambda+\mu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{3\lambda+2\mu}{\lambda} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{3\lambda+2\mu}{\lambda} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{3\lambda+2\mu}{\lambda} \end{pmatrix} \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix}.$$

Ceci donne comme équations :

$$\begin{cases} \epsilon_{11} = \frac{1}{2\mu} \sigma_{11} - \frac{\lambda}{2\mu(3\lambda+2\mu)} (\sigma_{11} + \sigma_{22} + \sigma_{33}) \\ \epsilon_{22} = \frac{1}{2\mu} \sigma_{22} - \frac{\lambda}{2\mu(3\lambda+2\mu)} (\sigma_{11} + \sigma_{22} + \sigma_{33}) \\ \epsilon_{33} = \frac{1}{2\mu} \sigma_{33} - \frac{\lambda}{2\mu(3\lambda+2\mu)} (\sigma_{11} + \sigma_{22} + \sigma_{33}) \\ \epsilon_{23} = \frac{1}{2\mu} \sigma_{23} \\ \epsilon_{13} = \frac{1}{2\mu} \sigma_{13} \\ \epsilon_{12} = \frac{1}{2\mu} \sigma_{12} \end{cases}$$

La formulation générale est alors donnée par :

$$\bar{\bar{\bar{\epsilon}}} = \frac{1}{2\mu} \bar{\bar{\bar{\sigma}}} - \frac{\lambda}{2\mu(3\lambda+2\mu)} \text{Tr}(\bar{\bar{\bar{\sigma}}}) \mathbb{1}, \quad \epsilon_{ij} = \frac{1}{2\mu} \sigma_{ij} - \frac{\lambda}{2\mu(3\lambda+2\mu)} \sigma_{kk} \delta_{ij}, \text{ pour } i, j = 1, 2, 3.$$

Le tenseur $\bar{\bar{\bar{D}}}$ de la relation (1.22) est ainsi défini par :

$$D_{ijkl} = \frac{1}{2\mu} \delta_{ik} \delta_{jl} - \frac{\lambda}{2\mu(3\lambda+2\mu)} \delta_{ij} \delta_{kl}.$$

Exemples de comportement élastique isotrope

Pour illustrer cette section, voici quelques exemples simples de comportements élastiques isotropes qui mettent en avant les formulations des paramètres mécaniques des matériaux (module de Young, coefficient de Poisson et module de cisaillement).

Cas simple d'une déformation par traction. Considérons un élément de matière soumis à une traction définie par un tenseur de contraintes uniaxial, c'est-à-dire un tenseur $\bar{\sigma}$ dont deux de ses valeurs propres sont nulles. La Fig. 1.6 illustre le cas d'une sollicitation selon l'axe x . Comme la contrainte est définie par le rapport entre la force F exercée sur la surface divisée par l'aire ds de celle-ci, cette valeur propre non nulle est égale à $\sigma_x = F/ds$. Le tenseur de contrainte correspondant est alors défini par :

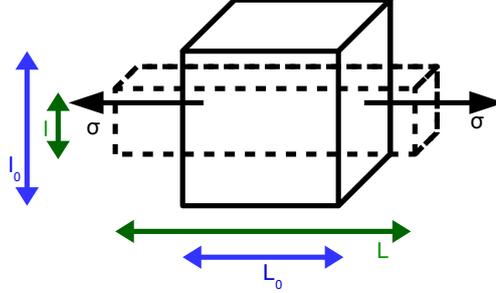


Figure 1.6 – Element de matière sollicité selon l'axe x par un tenseur de contraintes uniaxial.

$$\bar{\sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} = \begin{pmatrix} \sigma_x & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \sigma_x \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Dans le cas d'un élément de matière ayant un comportement élastique isotrope, nous pouvons considérer la relation (1.27) pour observer la déformation induite par cette contrainte en fonction des coefficients E et ν . Nous obtenons alors comme relations non nulles :

$$\begin{cases} \epsilon_{11} = \frac{1+\nu}{E} \sigma_{11} - \frac{\nu}{E} (\sigma_{11} + \sigma_{22} + \sigma_{33}) = \frac{1+\nu}{E} \sigma_x - \frac{\nu}{E} \sigma_x = \frac{\sigma_x}{E}, \\ \epsilon_{22} = \frac{1+\nu}{E} \sigma_{22} - \frac{\nu}{E} (\sigma_{11} + \sigma_{22} + \sigma_{33}) = -\frac{\nu}{E} \sigma_x, \\ \epsilon_{33} = \frac{1+\nu}{E} \sigma_{33} - \frac{\nu}{E} (\sigma_{11} + \sigma_{22} + \sigma_{33}) = -\frac{\nu}{E} \sigma_x. \end{cases}$$

Au final, le tenseur de déformation est défini par :

$$\bar{\epsilon} = \begin{pmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{pmatrix} = \begin{pmatrix} \frac{\sigma_x}{E} & 0 & 0 \\ 0 & -\frac{\nu}{E} \sigma_x & 0 \\ 0 & 0 & -\frac{\nu}{E} \sigma_x \end{pmatrix} = \frac{1}{E} \begin{pmatrix} \sigma_x \\ -\nu \sigma_x \\ -\nu \sigma_x \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (1.28)$$

Ainsi la déformation ϵ_{11} selon l'axe x est égale à la contrainte imposée σ_x divisée par le module de Young E . Autrement dit, le module de Young est défini par le rapport entre la contrainte et la déformation induite par cette sollicitation avec $E = \bar{\sigma}/\bar{\epsilon}$. La loi de

comportement de Hooke s'exprime alors sous la forme $\bar{\sigma} = E \bar{\epsilon}$.

Notons également que cette déformation selon l'axe x est positive. Elle correspond donc à un **étirement** selon l'axe x . Par contre, les déformations selon les axes y et z sont négatives et correspondent donc à un **rétrécissement** de la section selon les axes y et z , appelé **effet Poisson**. Nous pouvons observer ces déformations sur la Fig. 1.6.

Observons plus précisément la déformation selon l'axe y . Des relations $\epsilon_{22} = -\nu \sigma_x/E$ et $\epsilon_{11} = \sigma_x/E$, nous pouvons en déduire que le coefficient de Poisson est défini par¹⁵ :

$$\nu = -\frac{E \epsilon_{22}}{\sigma_x} = -\frac{\epsilon_{22}}{\epsilon_{11}} = -\frac{(l - l_0)/l_0}{(L - L_0)/L_0} = \frac{(l_0 - l)/l_0}{(L - L_0)/L_0},$$

où L, l représentent les longueurs de l'élément selon les axes x et y avant déformation, et L_0, l_0 leurs longueurs selon ces axes quand la contrainte est appliquée. Le coefficient de Poisson correspond ainsi au rapport entre le rétrécissement dans une direction perpendiculaire à l'effort subi et l'allongement dans la direction de l'effort.

Ce coefficient de Poisson est compris entre -1 et $0,5$. Si il est égal à 0 , la section n'est pas réduite. Par contre, pour une valeur de $0,5$ le volume est préservé lors de l'étirement (valeur employée dans le cas des matériaux incompressibles). Enfin, les valeurs expérimentales obtenues pour un matériau quelconque sont souvent voisines de $0,3$.

Cas simple d'une déformation par cisaillement. Considérons désormais que notre élément de matière est soumis à un tenseur de contraintes de cisaillement simple, c'est-à-dire un tenseur $\bar{\sigma}$ dont l'une des contraintes principales est nulle et les deux autres sont opposées. Un tenseur de contrainte correspondant peut être alors défini par :

$$\bar{\sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} = \begin{pmatrix} 0 & \tau & 0 \\ \tau & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \tau \end{pmatrix}.$$

Nous considérons de nouveau la relation (1.27) pour observer la déformation induite par cette contrainte en fonction des coefficients E et ν . En rappelant que le coefficient de Lamé μ est défini par $\mu = \frac{E}{2(1+\nu)}$, nous obtenons :

$$\bar{\epsilon} = \begin{pmatrix} 0 & \frac{(1+\nu)}{E} \sigma_{12} & 0 \\ \frac{(1+\nu)}{E} \sigma_{21} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{(1+\nu)}{E} \tau & 0 \\ \frac{(1+\nu)}{E} \tau & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{\tau}{2\mu} & 0 \\ \frac{\tau}{2\mu} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{\tau}{2\mu} \end{pmatrix}.$$

Nous pouvons observer qu'un cisaillement pure ne provoque aucune élongation, mais un changement d'angle entre les vecteurs des axes x et y avec $2 \epsilon_{12} = \tau/\mu$. Le coefficient de Lamé μ (auss appelé module de cisaillement) joue alors un rôle de raideur de cisaillement.

15. A noter qu'il en serait de même en observant la déformation ϵ_{33} selon l'axe z avec $\nu = -\epsilon_{33}/\epsilon_{11}$.

1.7 Energie de déformation élastique

Voyons pour finir comment s'établit l'expression de l'énergie de déformation en fonction des contraintes et déformations du matériau.

Le travail d'une force correspond à l'énergie fournie par cette force quand son point d'application se déplace, c'est-à-dire quand l'objet subissant la force se déplace ou se déforme. Le travail est exprimé en joules (J), et il est généralement noté W pour le terme anglais *Work*. Pour un petit déplacement $d\vec{u}$ du point d'application de la force, le travail élémentaire de la force \vec{F} est défini par¹⁶ :

$$dW = \vec{F}(\vec{u}) \cdot d\vec{u}.$$

Nous en déduisons que le travail de la force pour un déplacement imposé fini est égal à la circulation du point d'application de la force le long du trajet C (correspondant à un déplacement imposé final u_{final}) avec :

$$W = \int_C \vec{F}(\vec{u}) \cdot d\vec{u} = \int_0^{u_{final}} \vec{F}(\vec{u}) \cdot d\vec{u}.$$

Considérons maintenant un solide élastique isotrope ayant atteint son point d'équilibre après avoir été soumis à l'application de charges complexes. Nous notons $\vec{f}_{ext} \in \mathbb{R}^3$ les forces extérieures volumiques appliquées sur le domaine D_t de l'objet et $\vec{T} \in \mathbb{R}^3$ les forces appliquées sur la surface ∂D_t de l'objet au temps t .

A partir de ce point d'équilibre, nous incrémentons ensuite les forces en notant $d\vec{u}$ l'incrément des déplacements au point d'application de chacune des forces. En reprenant l'équation (1.2) du principe fondamental de la dynamique (établissant la somme des forces appliquées sur l'objet), nous en déduisons que l'énergie fournie par les forces appliquées à partir de cet état d'équilibre est définie par :

$$\begin{aligned} dW &= \int_{D_t} \rho \vec{f}_{ext} \cdot d\vec{u} dv + \int_{\partial D_t} \vec{T} \cdot d\vec{u} ds \\ &= \int_{D_t} \rho \sum_{j=1}^3 F_j du_j dv + \int_{\partial D_t} \sum_{j=1}^3 T_j du_j ds \end{aligned}$$

Nous pouvons ensuite faire apparaître le tenseur de contraintes $\bar{\sigma}$ en remplaçant les tractions \vec{T} par la projection des contraintes sur la normale \vec{n} à l'élément de surface ds considéré. Nous avons alors :

$$\int_{\partial D_t} \sum_{j=1}^3 T_j du_j ds = \int_{\partial D_t} \sum_{i=1}^3 \sum_{j=1}^3 \sigma_{ij} n_i du_j ds = \int_{\partial D_t} \sum_{i=1}^3 \sum_{j=1}^3 \sigma_{ij} du_j n_i ds.$$

En utilisant la convention de sommation d'Einstein, nous obtenons ainsi :

$$\begin{aligned} dW &= \sum_{j=1}^3 \int_{D_t} \rho F_j du_j dv + \sum_{i=1}^3 \sum_{j=1}^3 \int_{\partial D_t} \sigma_{ij} du_j n_i ds \\ &= \int_{D_t} \rho F_j du_j dv + \int_{\partial D_t} \sigma_{ij} du_j n_i ds. \end{aligned}$$

16. Je reprends ici en partie les explications du cours de Klöcker (2003) sur la formulation de l'énergie.

Le théorème de flux-divergence (ou théorème de Green-Ostrogradski) permet ensuite de transformer l'intégrale sur la surface ∂D_t en une intégrale sur le volume D_t du matériau. Nous avons ainsi

$$\int_{\partial D_t} \sigma_{ij} du_j n_i ds = \int_{D_t} \frac{\partial(\sigma_{ij} du_j)}{\partial x_i} dv = \int_{D_t} \left(\frac{\partial \sigma_{ij}}{\partial x_i} du_j + \sigma_{ij} \frac{\partial du_j}{\partial x_i} \right) dv$$

donnant comme formulation de l'énergie

$$dW = \int_{D_t} \left(\rho F_j + \frac{\partial \sigma_{ij}}{\partial x_i} \right) du_j dv + \int_{D_t} \sigma_{ij} \frac{\partial du_j}{\partial x_i} dv.$$

Rappelons que selon l'équation (1.9), nous avons $\rho \frac{d}{dt} \vec{v}(t) = \rho \vec{f}_{ext} + \vec{div} \bar{\sigma}$. Ainsi à l'état d'équilibre (où l'accélération de l'objet est par définition nulle), la première intégrale de volume de la formulation de l'énergie est nulle amenant la relation suivante :

$$dW = \int_{D_t} \sigma_{ij} \frac{\partial du_j}{\partial x_i} dv.$$

Dans le cas de petites déformations, nous pouvons considérer le tenseur de déformation de Green-Lagrange linéarisé (noté ϵ^g avec $\epsilon_{ij}^g = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$). En rappelant que le tenseur de contraintes est symétrique (*i.e.* $\sigma_{ij} = \sigma_{ji}$), nous avons alors :

$$\begin{aligned} dW &= \int_{D_t} \frac{1}{2} \left(\sigma_{ij} \frac{\partial du_j}{\partial x_i} + \sigma_{ji} \frac{\partial du_i}{\partial x_j} \right) dv \\ &= \int_{D_t} \sigma_{ij} \frac{1}{2} \left(\frac{\partial du_j}{\partial x_i} + \frac{\partial du_i}{\partial x_j} \right) dv \\ &= \int_{D_t} \sigma_{ij} d\epsilon_{ij}^g dv. \end{aligned}$$

Le travail de la force ou l'énergie de déformation élastique (exprimée en J) s'écrit ainsi

$$W = \int_{D_t} \int_0^\epsilon \sigma_{ij} d\epsilon_{ij}^g dv$$

avec une densité d'énergie élastique (ou énergie élastique par unité de volume) exprimée en J.m^{-3} qui est définie par :

$$W_e = \int_0^\epsilon \sigma_{ij} d\epsilon_{ij}^g = \int_0^\epsilon \bar{\sigma} : d\epsilon^g.$$

Dans le cas où les contraintes sont linéairement proportionnelles aux déformations (avec $\bar{\sigma} = \bar{\bar{C}} : \epsilon^g$), nous avons :

$$\begin{cases} W = \int_{D_t} \left(\int_0^\epsilon \bar{\bar{C}} : \epsilon^g : d\epsilon^g \right) dv = \frac{1}{2} \int_{D_t} (\bar{\bar{C}} : \epsilon^{g2}) dv \\ W_e = \int_0^\epsilon (\bar{\bar{C}} : \epsilon^g) : d\epsilon^g = \frac{1}{2} (\bar{\bar{C}} : \epsilon^{g2}) \end{cases}$$

L'énergie de déformation élastique s'exprime ainsi par

$$W = \frac{1}{2} \int_{D_t} (\bar{\bar{C}} : \epsilon^g) dv = \frac{1}{2} \int_{D_t} \sum_{i=1}^3 \sum_{j=1}^3 \sigma_{ij} \epsilon_{ij}^g dv \quad (1.29)$$

et la densité d'énergie élastique par

$$W_e = \frac{1}{2} (\bar{\sigma} : \epsilon^g) = \frac{1}{2} \sum_{i=1}^3 \sum_{j=1}^3 \sigma_{ij} \epsilon_{ij}^g. \quad (1.30)$$

Dans le cas d'un **matériau élastique isotrope** dont le comportement suit la loi de Hooke, les contraintes peuvent être exprimées en fonction des déformations et en fonction des coefficients de Lamé avec $\sigma_{ij} = \lambda \epsilon_{kk}^g \delta_{ij} + 2 \mu \epsilon_{ij}^g$ pour $i, j = 1, 2, 3$. L'énergie de déformation élastique par unité de volume est alors définie par

$$\begin{aligned} W_e &= \left(\mu + \frac{\lambda}{2} \right) \left(\sum_{i=1}^3 \epsilon_{ii} \right)^2 - 2 \mu \frac{1}{2} \sum_{i=1}^3 \sum_{j=1}^3 (\epsilon_{ii} \epsilon_{jj} - \epsilon_{ij} \epsilon_{ij}) \\ &= \left(\mu + \frac{\lambda}{2} \right) I_{e_1}^2 - 2 \mu I_{e_2} \end{aligned}$$

avec I_{e_1}, I_{e_2} les invariants du tenseur de déformation définis par :

$$\begin{cases} I_{e_1} &= Tr(\bar{\epsilon}) = \sum_{i=1}^3 \epsilon_{ii} = \epsilon_{11} + \epsilon_{22} + \epsilon_{33}, \\ I_{e_2} &= \frac{1}{2} \sum_{i=1}^3 \sum_{j=1}^3 (\epsilon_{ii} \epsilon_{jj} - \epsilon_{ij} \epsilon_{ij}) = \epsilon_{11} \epsilon_{22} + \epsilon_{22} \epsilon_{33} + \epsilon_{33} \epsilon_{11} - \epsilon_{12}^2 - \epsilon_{23}^2 - \epsilon_{31}^2. \end{cases}$$

Cette expression peut également s'écrire sous la forme suivante :

$$\begin{aligned} W_e &= \left(\mu + \frac{\lambda}{2} \right) (\epsilon_{11}^2 + \epsilon_{22}^2 + \epsilon_{33}^2) + \lambda (\epsilon_{11} \epsilon_{22} + \epsilon_{22} \epsilon_{33} + \epsilon_{33} \epsilon_{11}) + 2\mu (\epsilon_{11}^2 + \epsilon_{22}^2 + \epsilon_{33}^2) \\ &= \frac{\lambda}{2} Tr(\bar{\epsilon})^2 + \mu \sum_{i=1}^3 \sum_{j=1}^3 \epsilon_{ij} \epsilon_{ij}. \end{aligned}$$

avec $Tr(\bar{\epsilon})^2 = \epsilon_{11}^2 + \epsilon_{22}^2 + \epsilon_{33}^2 + 2 \epsilon_{11} \epsilon_{22} + 2 \epsilon_{22} \epsilon_{33} + 2 \epsilon_{33} \epsilon_{11}$.

1.8 Retour sur l'équation de la dynamique à résoudre

Rappelons que notre objectif était de résoudre l'équation (1.9) gouvernant le mouvement de l'objet. Pour cela, nous avons besoin de relier la vitesse $\vec{v}(t)$ de l'objet au tenseur de contraintes $\bar{\sigma}$. Comme illustré par la Fig. 1.7, ce lien va être effectué en exprimant la vitesse et le tenseur de contraintes en fonction du déplacement $\vec{u}(t)$.

- Pour la vitesse, cette relation est immédiate puisque la vitesse correspond à la dérivée par rapport au temps du déplacement avec $\vec{v}(t) = d\vec{u}(t)/dt$.
- Pour le tenseur de contraintes, nous allons utiliser la loi de comportement de l'objet puisqu'elle relie le tenseur de contraintes $\bar{\sigma}$ à la déformation $\bar{\epsilon}$. Ensuite il suffit d'exprimer la déformation en fonction du déplacement $\vec{u}(t)$, pour relier $\bar{\sigma}$ à $\vec{u}(t)$.

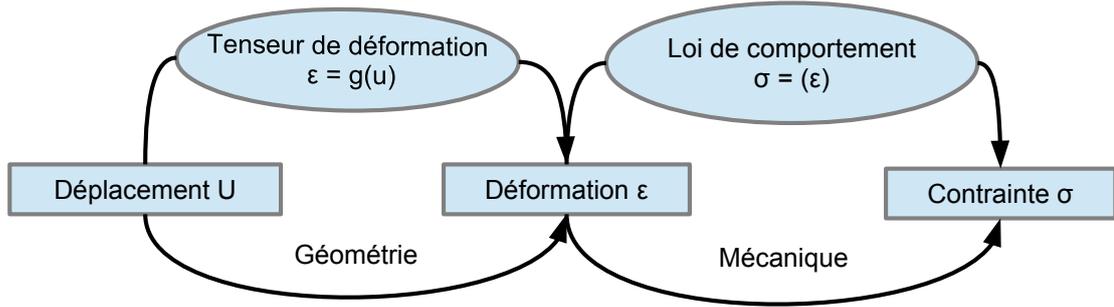


Figure 1.7 – Récapitulatif des relations existantes entre le déplacement $\vec{u}(t)$ de l'objet, sa déformation (tenseur $\bar{\epsilon}$), et les contraintes (tenseur $\bar{\sigma}$) appliquées sur celui-ci.

Pour définir la formulation de l'équation du mouvement de l'objet, nous devons ainsi choisir le tenseur de déformation et la loi de comportement associés à l'objet que nous souhaitons simuler.

Cas d'un matériau élastique isotrope

Si nous considérons un matériau élastique isotrope dont le comportement est régi par la loi de Hooke définie en fonction des coefficients de Lamé par $\bar{\sigma} = \lambda \text{Tr}(\bar{\epsilon}) \mathbb{1} + 2 \mu \bar{\epsilon}$ et en considérant le tenseur de déformation de Green-Lagrange linéarisé défini par $\epsilon^g = \frac{1}{2}(\mathbb{U}^T + \mathbb{U}) = \frac{1}{2}(\nabla \vec{u}^T + \nabla \vec{u})$, nous obtenons¹⁷ :

$$\begin{aligned}
 \overrightarrow{\text{div}} \bar{\sigma} &= \overrightarrow{\text{div}} (\lambda \text{Tr}(\epsilon^g) \mathbb{1} + 2 \mu \epsilon^g) \\
 &= \lambda \overrightarrow{\text{div}} (\text{Tr}(\epsilon^g) \mathbb{1}) + 2 \mu \overrightarrow{\text{div}} (\epsilon^g) \\
 &= \lambda \nabla (\text{Tr}(\epsilon^g)) + 2 \mu \overrightarrow{\text{div}} \left(\frac{1}{2} (\nabla \vec{u}^T + \nabla \vec{u}) \right) \\
 &= \lambda \nabla (\text{Tr}(\epsilon^g)) + \mu \overrightarrow{\text{div}} (\nabla \vec{u}^T) + \mu \overrightarrow{\text{div}} (\nabla \vec{u}) \\
 &= \lambda \nabla (\text{div} \vec{u}) + \mu \nabla (\text{div} \vec{u}) + \mu \Delta \vec{u} \\
 &= (\lambda + \mu) \nabla (\text{div} \vec{u}) + \mu \Delta \vec{u} \\
 &= (\lambda + \mu) \nabla (\text{div} \vec{u}) + \mu \left(\nabla (\text{div} \vec{u}) - \overrightarrow{\text{rot}} \overrightarrow{\text{rot}} \vec{u} \right) \\
 &= (\lambda + 2 \mu) \nabla (\text{div} \vec{u}) - \mu \overrightarrow{\text{rot}} \overrightarrow{\text{rot}} \vec{u}
 \end{aligned}$$

L'équation du mouvement peut alors s'écrire de la manière suivante (appelée **équation de Navier de l'élasticité linéaire isotrope**) :

$$\rho \frac{d^2 \vec{u}(t)}{dt^2} = \rho \overrightarrow{f_{ext}}(t) + (\lambda + \mu) \nabla (\text{div} \vec{u}(t)) + \mu \Delta \vec{u}(t) \quad (1.31)$$

$$= \rho \overrightarrow{f_{ext}}(t) + (\lambda + 2 \mu) \nabla (\text{div} \vec{u}(t)) - \mu \overrightarrow{\text{rot}} \overrightarrow{\text{rot}} \vec{u}(t). \quad (1.32)$$

où $\overrightarrow{f_{ext}}(t)$ incorpore la force de la gravité \vec{g} .

17. Pour rappel, la trace du tenseur de Green-Lagrange linéarisé est définie par $\text{Tr}(\epsilon^g) = \text{div} \vec{u}$. Au niveau mathématiques, nous avons : $\overrightarrow{\text{div}} \nabla \vec{a} = \Delta \vec{a} = \nabla (\text{div} \vec{a}) - \overrightarrow{\text{rot}} \overrightarrow{\text{rot}} \vec{a}$ où $\overrightarrow{\text{rot}} \vec{a} = \nabla \wedge \vec{a}$. Le laplacien est défini par $\Delta f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$ représentant ainsi la courbure dans les différentes directions de l'espace.

L'équation du mouvement est ensuite résolue pour obtenir le déplacement $\vec{u}(t)$ de l'objet à chaque pas de temps. Comme nous ne pouvons que rarement résoudre cette équation de manière analytique (c'est-à-dire qu'il est rare de trouver une solution formelle au problème), une méthode d'approximation numérique est alors employée pour résoudre cette équation de manière discrète. Pour cela, le problème mathématique initialement défini sur un milieu continu est transformé en un problème discret (équation matricielle) de dimension finie. Puis, la méthode des éléments finis (MEF) est généralement employée pour la résolution numérique du problème discret. La discrétisation de l'équation initiale repose alors sur la définition d'un maillage représentant l'objet, c'est-à-dire que l'objet (ou de manière plus générale le domaine) est discrétisé en un ensemble d'éléments et les équations sont résolues sur chacun des nœuds de ces éléments. Nous obtenons alors une solution approchée de l'équation initiale définissant le déplacement de l'objet.

1.9 Discrétisation du problème continu

Détaillons maintenant l'obtention de la formulation discrète du problème de la dynamique newtonienne. L'écriture de cette formulation passe par la reformulation de l'équation du mouvement en équations différentielles ou intégrales. Cette formulation intégrale peut être obtenue de deux façons différentes et équivalentes : soit en utilisant la méthode des résidus pondérés (qui introduit une notion d'erreur), soit en définissant des formulations variationnelles du problème. Nous allons présenter ces deux types d'approches ainsi que la discrétisation du problème mécanique qui en découle¹⁸.

Méthode des résidus pondérés. Reprenons l'équation de la dynamique newtonienne décrivant le mouvement de tout objet déformable. Nous avons vu qu'à l'intérieur du domaine D_t que constitue l'objet au temps t , nous avons :

$$\forall M \in D_t, \rho \frac{d^2 \vec{u}(t)}{dt^2} = \rho \overrightarrow{f_{ext}} + \overrightarrow{div} \overline{\sigma}.$$

Cette équation dite locale est complétée par des équations définies sur la surface ∂D_t de l'objet appelées *conditions limites*. Dans notre cas, elles sont de deux types avec :

- Les conditions limites en déplacement : $\forall M \in \Gamma_u, \vec{u}(t) = \vec{u}_d(t)$ où Γ_u représente une partie de la surface ∂D_t de l'objet sur laquelle le déplacement \vec{u}_d est imposé.
- Les conditions limites en contraintes : $\forall M \in \Gamma_\sigma, \overline{\sigma} \cdot \vec{n} = \vec{T}_d$ où Γ_σ représente une partie de la surface ∂D_t de l'objet sur laquelle les vecteurs de contraintes \vec{T}_d (et non le tenseur de contraintes) sont imposés. Si la surface est libre, \vec{T}_d est nul.
- A noter que nous avons $\Gamma_u \cup \Gamma_\sigma = \partial D_t$ et $\Gamma_u \cap \Gamma_\sigma = \emptyset$, c'est-à-dire que, pour une partie donnée de la surface, nous contraignons soit le déplacement de la surface, soit les vecteurs de contraintes appliqués sur elle, mais jamais les deux en même temps.

Ce problème mécanique défini par une équation locale et des conditions limites doit ainsi être résolu pour obtenir le champ vectoriel $\vec{u}(t)$ définissant le déplacement de l'objet. Formulons-le de manière plus générale pour présenter la méthode des résidus pondérés.

Pour un point $M = (x, y, z)$ au temps t , nous considérons un système qui est défini à la fois par des équations différentielles établies sur le domaine D_t et par des équations

18. Je reprends ici l'ensemble des explications proposées dans le cours de Hervé Oudin (2008).

établies sur sa frontière ∂D_t , avec :

$$\begin{aligned} \forall M \in D_t, \quad L(u) &= f(M, t) && \text{équation locale} \\ \forall M \in \partial D_t, \quad C(u) &= e(M, t) && \text{conditions aux limites} \end{aligned}$$

Nous appelons alors *résidu* l'erreur commise pour une approximation u^* du champ scalaire u qui satisfasse les conditions. Pour simplifier, si nous considérons que l'approximation u^* choisie satisfait les conditions aux limites avec $C(u^*) = C(u)$, nous pouvons considérer uniquement l'équation locale pour définir le résidu. Nous avons alors :

$$\forall M \in D_t, \quad R(u^*) = L(u^*) - L(u) = L(u^*) - f(M, t).$$

Nous définissons ensuite un ensemble de *fonctions dites de pondération (ou de test)* $\psi_i(M)$ définies sur le domaine D_t . La méthode des résidus pondérés consiste en effet à annuler l'erreur commise sur le résidu en le pondérant sur le domaine par un nombre fini de fonctions $\psi_i(M)$. Nous obtenons alors les **formulations intégrales** suivantes définies pour les différentes fonctions de pondération :

$$\forall \psi_i(M), \quad \int_{D_t} \psi_i(M) R(u^*) \, dv = 0.$$

Ainsi, au lieu de résoudre l'équation mathématique $R(u) = 0$, nous considérons un problème équivalent défini par

$$\forall \psi, \quad \int_{D_t} \psi R(u) \, dv = 0. \tag{1.33}$$

Pour obtenir une approximation numérique de la solution du problème (qui ne se résout pas analytiquement), nous restreignons alors le nombre de fonctions de pondérations. Nous définissons ainsi un système matriciel d'ordre n puisqu'en définissant n fonctions de pondération, nous définissons n équations de la forme :

$$\begin{aligned} \forall i \in [1, n], \quad \int_{D_t} \psi_i(M) R(u^*) \, dv &= 0 \\ \Rightarrow \int_{D_t} \psi_i(M) (L(u^*) - f(M, t)) \, dv &= 0 \\ \Rightarrow \int_{D_t} \psi_i(M) L(u^*) \, dv - \int_{D_t} \psi_i(M) f(M, t) \, dv &= 0. \end{aligned}$$

Le nombre n de fonctions de pondération correspond généralement au nombre de paramètres de l'approximation u^* (permettant ainsi l'obtention du même nombre d'équations que de paramètres). L'approximation u^* est en effet définie en fonction de *fonctions de forme* $N_i(M)$ (constituant une base de fonctions pour construire la solution) et de paramètres notés $q_i(t)$ définissant les paramètres de l'approximation (constituant les participations des différentes fonctions de forme dans la solution du problème).

L'**approximation u^* de la solution** est alors de la forme :

$$\begin{aligned} u^*(M, t) = \sum_{i=1}^n N_i(M) q_i(t) &= \sum_{i=1}^n N_i(x, y, z) q_i(t) = (N_1(x, y, z), N_2(x, y, z), \dots, N_n(x, y, z)) \begin{pmatrix} q_1(t) \\ q_2(t) \\ \vdots \\ q_n(t) \end{pmatrix} \\ &= \mathbf{N}(x, y, z)^T \mathbf{q}(t) = \mathbf{N}(M)^T \mathbf{q}(t) \end{aligned}$$

conduisant au système constitué des n équations suivantes :

$$\begin{aligned} \forall i \in [1, n], \quad & \int_{D_t} \psi_i(M) L \left(\sum_{i=1}^n N_i(M) q_i(t) \right) dv - \int_{D_t} \psi_i(M) f(M, t) dv = 0 \\ \Rightarrow \quad & \int_{D_t} \psi_i(M) L \left(\sum_{i=1}^n N_i(M) q_i(t) \right) dv = \int_{D_t} \psi_i(M) f(M, t) dv. \end{aligned}$$

Ainsi en définissant la matrice K (de taille $n \times n$) et le vecteur \mathbf{F} (de taille n) par

$$K = \int_{D_t} \psi(M) L \left(\mathbf{N}(M)^T \right) dv, \quad \mathbf{F} = \int_{D_t} \psi(M) f(M, t) dv,$$

nous obtenons l'équation matricielle $K \mathbf{q}(t) = \mathbf{F}$ dont la solution $\mathbf{q}(t)$ fournit les paramètres de l'approximation.

Dans le cadre de l'équation du mouvement, le **résidu** est défini par

$$\forall M \in D_t, R(u^*) = \rho \frac{d^2 \vec{u}^*(t)}{dt^2} - \rho \vec{f}_{ext} - \overrightarrow{div} \bar{\sigma}$$

conduisant à la **formulation intégrale** suivante du problème :

$$\forall \psi_i(M), \int_{D_t} \psi_i(M) \rho \frac{d^2 \vec{u}^*(t)}{dt^2} dv = \int_{D_t} \psi_i(M) \left(\rho \vec{f}_{ext} + \overrightarrow{div} \bar{\sigma} \right) dv.$$

Le problème est ensuite résolu en restreignant à n le nombre de fonctions de pondération ψ_i et en construisant une approximation de la solution de la forme $\vec{u}^* = \mathbf{N}(M)^T \mathbf{q}(t)$ pour un paramètre $\mathbf{q}(t)$ et en considérant n fonctions de forme N_i , pour $i \in [1, n]$.

Au final, nous devons résoudre le système $K \mathbf{q}(t) = \mathbf{F}$ où :

$$K = \int_{D_t} \psi(M) \rho \frac{d^2}{dt^2} \left(\mathbf{N}(M)^T \right) dv, \quad \mathbf{F} = \int_{D_t} \psi(M) \left(\rho \vec{f}_{ext} + \overrightarrow{div} \bar{\sigma} \right) dv.$$

En introduisant une notion d'erreur sur les équations et en utilisant des fonctions de pondération $\psi_i(M)$ et des fonctions de forme $\mathbf{N}(M)$, la méthode des résidus pondérés permet ainsi de reformuler le problème initialement défini sur un milieu continu en un problème discret en passant par une reformulation en intégrales des équations.

Deux approches sont ensuite adoptées pour définir les fonctions de pondération :

- Nous pouvons employer des fonctions de Dirac comme fonctions de pondérations, annulant ainsi l'erreur d'approximation en un nombre fini de points du domaine.
- Nous pouvons employer les mêmes fonctions pour les fonctions de forme et les fonctions de pondération (méthode de Galerkin). De plus, si les opérateurs employés sont symétriques, nous obtenons alors des matrices symétriques.

D'autres méthodes sont utilisées pour construire une approximation de la solution. Nous présentons dans la suite une approche basée sur la formulation variationnelle de l'équation. Dans le cours de Hervé [Oudin \(2008\)](#), vous pouvez également retrouver l'approche basée sur le *principe des travaux virtuels* qui amène à la même formulation du système matricielle que celui que nous allons maintenant présenter.

Formulation variationnelle. Repartons de nouveau de l'équation générale décrivant le mouvement d'un objet déformable selon la dynamique newtonienne. Nous avons

$$\forall M \in D_t, \quad \rho \frac{d^2 \vec{u}(t)}{dt^2} - \overrightarrow{\text{div}} \bar{\sigma} - \rho \overrightarrow{f_{ext}} = 0.$$

Cette équation est équivalente à : $\forall \vec{\psi}, \quad \int_{D_t} \vec{\psi} \cdot \left(\rho \frac{d^2 \vec{u}(t)}{dt^2} - \overrightarrow{\text{div}} \bar{\sigma} - \rho \overrightarrow{f_{ext}} \right) dv = 0.$

Par ailleurs, en utilisant les relations $\bar{\sigma} : \overline{\text{grad}} \vec{u} = \text{div}(\bar{\sigma} \vec{u}) - \vec{u} \cdot \overrightarrow{\text{div}} \bar{\sigma}^T$ et $\bar{\sigma} : \overline{\text{grad}}^T \vec{u} = \text{div}(\bar{\sigma} \vec{u}) - \vec{u} \cdot \overrightarrow{\text{div}} \bar{\sigma}$ et en sachant que le tenseur de contraintes $\bar{\sigma}$ est symétrique (*i.e.* $\bar{\sigma} = \bar{\sigma}^T$), nous pouvons établir la relation $\bar{\sigma} : \overline{\text{grad}} \vec{\psi} = \text{div}(\bar{\sigma} \vec{\psi}) - \vec{\psi} \cdot \overrightarrow{\text{div}} \bar{\sigma}.$

Nous avons alors¹⁹ :

$$\begin{aligned} \forall \vec{\psi}, \quad & \int_{D_t} \left(\vec{\psi} \cdot \rho \frac{d^2 \vec{u}(t)}{dt^2} + \bar{\sigma} : \overline{\text{grad}} \vec{\psi} - \text{div}(\bar{\sigma} \vec{\psi}) - \psi \cdot \rho \overrightarrow{f_{ext}} \right) dv = 0 \\ \Rightarrow & \int_{D_t} \left(\vec{\psi} \cdot \rho \frac{d^2 \vec{u}(t)}{dt^2} + \bar{\sigma} : \overline{\text{grad}} \vec{\psi} - \psi \cdot \rho \overrightarrow{f_{ext}} \right) dv - \int_{D_t} \text{div}(\bar{\sigma} \vec{\psi}) dv = 0 \\ \Rightarrow & \int_{D_t} \left(\vec{\psi} \cdot \rho \frac{d^2 \vec{u}(t)}{dt^2} + \bar{\sigma} : \overline{\text{grad}} \vec{\psi} - \psi \cdot \rho \overrightarrow{f_{ext}} \right) dv - \int_{\partial D_t} \vec{\psi} \cdot \bar{\sigma} \cdot \vec{n} ds = 0. \end{aligned}$$

Nous pouvons ensuite introduire les conditions limites appliquées à la frontière ∂D_t de l'objet. Nous avons vu qu'elles étaient de deux types : en déplacement sur la partie Γ_u de la surface avec $\forall M \in \Gamma_u, \vec{u}(t) = \vec{u}_d(t)$ et en contraintes sur la partie Γ_σ de la surface avec $\forall M \in \Gamma_\sigma, \bar{\sigma} \cdot \vec{n} = \vec{T}_d.$ Nous obtenons ainsi :

$$\int_{D_t} \left(\vec{\psi} \cdot \rho \frac{d^2 \vec{u}(t)}{dt^2} + \bar{\sigma} : \overline{\text{grad}} \vec{\psi} - \psi \cdot \rho \overrightarrow{f_{ext}} \right) dv - \int_{\Gamma_u} \vec{\psi} \cdot \bar{\sigma} \cdot \vec{n} ds - \int_{\Gamma_\sigma} \vec{\psi} \cdot \vec{T}_d ds = 0.$$

De plus, en utilisant des fonctions de pondération qui sont à valeur nulle sur la frontière Γ_u de l'objet avec $\forall M \in \Gamma_u, \vec{\psi}(M) = \vec{0}$ induisant $\int_{\Gamma_u} \vec{\psi} \cdot \bar{\sigma} \cdot \vec{n} ds = 0,$ nous obtenons la **formulation variationnelle** suivante :

$$\int_{D_t} \left(\vec{\psi} \cdot \rho \frac{d^2 \vec{u}(t)}{dt^2} \right) dv + \int_{D_t} \left(\bar{\sigma} : \overline{\text{grad}} \vec{\psi} \right) dv - \int_{D_t} \left(\psi \cdot \rho \overrightarrow{f_{ext}} \right) dv = \int_{\Gamma_\sigma} \left(\vec{\psi} \cdot \vec{T}_d \right) ds.$$

avec $\forall M \in \Gamma_u, \vec{u}(t) = \vec{u}_d(t)$ et $\vec{\psi}(M) = \vec{0}.$

Cette formulation variationnelle est équivalente au système d'équations aux dérivées partielles, c'est-à-dire que la solution exacte de cette formulation intégrale est la même que celle du système initial. Mais comme il est rare de trouver une solution formelle, la solution recherchée est mise comme précédemment sous la forme d'une combinaison linéaire de n fonctions dites *fonctions de forme*. L'idée est ensuite d'*affaiblir* une des formes intégrales en ne la satisfaisant que pour n *fonctions de pondération*. A noter que si nous construisons une solution satisfaisant toutes les conditions limites nous retombons sur la méthode des résidus pondérés. Mais en pratique, nous construisons une approximation \vec{u}^* satisfaisant les conditions limites en déplacement avec $\forall M \in \Gamma_u, \vec{u}^* = \vec{u}_d.$

19. En utilisant le théorème de flux-divergence (ou de Green-Ostrogradski) : $\int_{D_t} \text{div} \vec{A} dv = \int_{\partial D_t} \vec{A} \cdot \vec{n} ds.$

Pour simplifier, nous considérons des conditions aux limites sur Γ_u telles que $\bar{u}^* = \bar{0}$ et nous employons la méthode de Galerkin pour le choix des fonctions de pondération, c'est-à-dire que nous utilisons les mêmes fonctions pour les fonctions de forme et pour les fonctions de pondération. L'approximation construite a ainsi la forme :

$$\mathbf{u}^*(M, t) = \mathbf{N}(M) \mathbf{q}(t)$$

avec $\mathbf{N}(M)$ la matrice construite à partir des fonctions de forme et $\mathbf{q}(t)$ le vecteur des paramètres de l'approximation. La forme matricielle des fonctions de pondération est alors définie par :

$$\psi(M) = \mathbf{N}(M) \delta \mathbf{q}$$

Si nous reprenons la formulation intégrale, nous devons exprimer $\bar{\sigma} : \overline{\text{grad}} \vec{\psi}$. Pour cela nous utilisons les formes matricielles associées aux lois de comportement avec $\sigma(M) = \mathbf{C}(M) \epsilon(M)$ et aux relations entre déformations et déplacements avec $\epsilon(M) = \mathbf{L} \mathbf{u}(M)$ où \mathbf{L} est la matrice d'opérateurs différentiels correspondant à l'expression du gradient symétrique du champ de déplacements. Nous avons ainsi :

$$\begin{aligned} \bar{\sigma}^* : \overline{\text{grad}} \vec{\psi} &= \psi(M)^T \mathbf{L}^T \underbrace{\mathbf{C}(M) \mathbf{L} \mathbf{u}^*(M)}_{\sigma^*(M)} \\ &= \underbrace{(\mathbf{N}(M) \delta \mathbf{q})^T}_{\psi(M)} \mathbf{L}^T \mathbf{C}(M) \mathbf{L} \underbrace{\mathbf{N}(M) \mathbf{q}(t)}_{u^*} \\ &= \delta \mathbf{q}^T (\mathbf{L} \mathbf{N}(M))^T \mathbf{C}(M) \mathbf{L} \mathbf{N}(M) \mathbf{q}(t). \end{aligned}$$

En définissant $\mathbf{B}(M) = \mathbf{L} \mathbf{N}(M)$, nous obtenons

$$\bar{\sigma}^* : \overline{\text{grad}} \vec{\psi} = \delta \mathbf{q}^T \mathbf{B}(M)^T \mathbf{C}(M) \mathbf{B}(M) \mathbf{q}(t)$$

que nous reportons dans la formulation variationnelle :

$$\forall \delta \mathbf{q}^T, \delta \mathbf{q}^T \int_{D_t} \left(\mathbf{N}^T \rho \mathbf{N} \ddot{\mathbf{q}} + \mathbf{B}^T \mathbf{C} \mathbf{B} \mathbf{q} - \mathbf{N}^T \rho \mathbf{f}_{ext} \right) dv = \delta \mathbf{q}^T \int_{\Gamma_\sigma} \mathbf{N}^T \mathbf{T}_d ds.$$

Au final nous obtenons l'équation matricielle $\mathbf{M} \ddot{\mathbf{q}} + \mathbf{K} \mathbf{q} = \mathbf{F}$ avec

$$\mathbf{M} = \int_{D_t} \mathbf{N}^T \rho \mathbf{N} dv, \quad \mathbf{K} = \int_{D_t} \mathbf{B}^T \mathbf{C} \mathbf{B} dv, \quad \mathbf{F} = \int_{D_t} \mathbf{N}^T \rho \mathbf{f}_{ext} dv + \int_{\Gamma_\sigma} \mathbf{N}^T \mathbf{T}_d ds,$$

où \mathbf{N} correspond à la matrice des fonctions de forme, \mathbf{C} (issue de la loi de comportement) est la matrice faisant le lien entre les contraintes et les déformations, $\mathbf{B} = \mathbf{L} \mathbf{N}$ où \mathbf{L} est la matrice des opérateurs différentiels permettant d'exprimer les déformations en fonction des déplacements.

1.10 Méthode des éléments finis

Nous venons de présenter comment était établie la formulation discrète du problème de la dynamique newtonienne. Reste maintenant à résoudre le système discret obtenu afin de définir le déplacement de l'objet considéré. La méthode des éléments finis (MEF)²⁰ est alors adoptée pour cette résolution. Cette méthode repose sur la discrétisation du domaine continu en sous domaines et dans la définition d'une approximation de la solution en chacun des points de ces sous domaines. Ces solutions dites élémentaires obtenues sur chacun des sous domaines sont ensuite assemblées pour établir la solution complète du problème.

20. Je reprends encore ici les explications et illustrations du cours de Hervé Oudin (2008) sur la MEF.

Discretisation géométrique du domaine La méthode des éléments finis repose sur la discrétisation du modèle continu en sous domaines qui sont appelés *éléments finis*. Ces éléments sont de forme géométrique simple. En 2D, nous utilisons des triangles et des quads, et des tétraèdres ou des hexaèdres en 3D. L'union de ces éléments constitue le maillage de l'objet. Les maillages constitués du même type d'élément (c'est-à-dire d'éléments de même géométrie) sont généralement privilégiés. Il est possible d'employer des maillages non réguliers, c'est-à-dire présentant des éléments de tailles variables. Il est ainsi usuel de raffiner les zones d'intérêt et d'utiliser des éléments plus grossiers dans les zones de moindre intérêt ou dans les zones où la solution varie peu. Par contre, il faut veiller à ce que les éléments soient faiblement distordus pour faciliter la convergence des méthodes numériques employées. L'approximation de la solution est ensuite calculée en des points donnés du maillage appelés nœuds, permettant d'obtenir par interpolation la solution en chacun des points du domaine. En fonction de l'interpolation utilisée, nous pouvons considérer comme nœuds uniquement les sommets des éléments ou ajouter d'autres nœuds comme les points du milieu de chacune des arêtes des éléments.

La génération du maillage de l'objet constitue ainsi une phase importante de l'emploi de la méthode par éléments finis pour l'obtention de résultats appropriés. Il s'agit en effet de trouver le bon compromis entre la taille des éléments finis (générant un plus ou moins grand nombre de nœuds) et la modélisation géométrique idéale de l'objet simulé.

Approximation nodale. Comme mentionné précédemment, après avoir décomposé le domaine initial D_t en un ensemble de sous domaines D_e de forme géométrique simple, la méthode des éléments finis est basée sur la construction pour chacun de ces sous domaines d'une approximation \mathbf{u}^* du champ des variables \mathbf{u} . Cette approximation est construite uniquement à partir des valeurs approchées obtenues en chacun des nœuds des éléments considérés. Nous parlons alors d'approximation nodale.

Si nous considérons le sous domaine élémentaire D_e , l'approximation nodale est définie par

$$\forall M \in D_e, \mathbf{u}^*(M) = \mathbf{N}(M) \mathbf{u}_n \quad (1.34)$$

avec $\mathbf{u}^*(M)$ la valeur de la fonction approchée en tout point M de l'élément D_e , \mathbf{u}_n les variables nodales relatives aux nœuds d'interpolation de l'élément et \mathbf{N} la matrice ligne des fonctions d'interpolation de l'élément.

Si nous notons M_i les nœuds de l'élément pour lesquels l'approximation \mathbf{u}^* est identifiée par rapport à la valeur du champ de variables \mathbf{u} , nous avons

$$\forall M_i, \begin{cases} \mathbf{u}^*(M_i) = \mathbf{u}_i \\ N_j(M_i) = \delta_{ij} \end{cases} \quad (1.35)$$

L'interpolation nodale est alors construite à partir d'une approximation générale :

$$\forall M, \mathbf{u}^*(M) = \Phi(M) \mathbf{a} \quad (1.36)$$

avec Φ une base de fonctions connues indépendantes (par exemple une base polynomiale) et \mathbf{a} le vecteur des paramètres de l'approximation (sans signification physique). Ainsi en identifiant aux nœuds l'approximation \mathbf{u}^* à la valeur du champ de variables \mathbf{u} , nous pouvons exprimer les paramètres \mathbf{a} en fonction des variables nodales \mathbf{u}_n avec

$$\mathbf{u}_n = \mathbf{u}^*(M_n) = \Phi(M_n) \mathbf{a}.$$

Par inversion nous obtenons $\mathbf{a} = \mathbf{T} \mathbf{u}_n$. Si nous récapitulons, nous avons comme relations $\mathbf{u}^*(M) = \mathbf{N}(M) \mathbf{u}_n = \Phi(M) \mathbf{a} = \Phi(M) \mathbf{T} \mathbf{u}_n$ donnant la matrice des fonctions d'interpolation $\mathbf{N}(M) = \Phi(M) \mathbf{T}$.

Comme bases polynomiales complètes, nous pouvons considérer en 2D la base linéaire $(1, x, y)$ avec 3 variables ou la base quadratique $(1, x, y, x^2, xy, y^2)$ avec 6 variables. En 3D, il y a la base linéaire $(1, x, y, z)$ avec 4 variables ou avec 10 variables la base quadratique $(1, x, y, z, x^2, xy, y^2, xz, z^2, yz)$. Pour utiliser ce type de bases, le nombre de termes de la base polynomiale complète doit être égal au nombre de variables nodales à identifier.

Fonctions d'interpolation sur un élément triangulaire en 2D. Illustrons ces notions en considérant un triangle en 2D défini par 3 nœuds de coordonnées (x_1, y_1) , (x_2, y_2) et (x_3, y_3) et en utilisant la base polynomiale linéaire $\Phi(M) = \Phi(x, y) = (1, x, y)$ pour construire les fonctions d'interpolation de cet élément triangulaire.

Nous recherchons ainsi une approximation polynomiale linéaire à 3 paramètres de la forme $\mathbf{u}^*(M) = \mathbf{u}^*(x, y) = \Phi(M) \mathbf{a}$ conduisant à la solution approchée linéaire :

$$\mathbf{u}^*(x, y) = (1 \ x \ y) \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix} = \mathbf{a}_1 + \mathbf{a}_2 x + \mathbf{a}_3 y.$$

Nous identifions ensuite le champ de déplacement au niveau des valeurs nodales \mathbf{u}_i avec $\mathbf{u}^*(x_i, y_i) = \mathbf{u}_i$. Nous obtenons alors le système matriciel suivant :

$$\mathbf{u}_n = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{u}^*(x_1, y_1) \\ \mathbf{u}^*(x_2, y_2) \\ \mathbf{u}^*(x_3, y_3) \end{pmatrix} = \begin{pmatrix} \Phi(x_1, y_1) \\ \Phi(x_2, y_2) \\ \Phi(x_3, y_3) \end{pmatrix} \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix} = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix} \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix}.$$

La relation inverse est de la forme :

$$\mathbf{a} = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix} = \frac{1}{2A} \underbrace{\begin{pmatrix} \delta_{23} & \delta_{31} & \delta_{12} \\ y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{pmatrix}}_{\mathbf{T}} \underbrace{\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix}}_{\mathbf{u}_n} \quad (1.37)$$

avec A l'aire du triangle et $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$ et $\delta_{ij} = x_i y_j - x_j y_i$. Si nous reportons ce résultat dans l'approximation, nous obtenons :

$$\begin{aligned} \mathbf{u}^*(x, y) &= \underbrace{(1 \ x \ y)}_{\Phi(M)} \frac{1}{2A} \underbrace{\begin{pmatrix} \delta_{23} & \delta_{31} & \delta_{12} \\ y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{pmatrix}}_{\mathbf{T}} \underbrace{\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix}}_{\mathbf{u}_n} \\ &= \frac{1}{2A} \underbrace{\left(\delta_{23} + y_{23} x + x_{32} y \quad \delta_{31} + y_{31} x + x_{13} y \quad \delta_{12} + y_{12} x + x_{21} y \right)}_{\mathbf{N}(x, y)} \underbrace{\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix}}_{\mathbf{u}_n} \\ &= \begin{pmatrix} N_1(x, y) & N_2(x, y) & N_3(x, y) \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix} \end{aligned}$$

La solution approchée est ainsi de la forme :

$$\mathbf{u}^*(x, y) = \mathbf{u}_1 N_1(x, y) + \mathbf{u}_2 N_2(x, y) + \mathbf{u}_3 N_3(x, y)$$

avec $N_i(x, y)$ les fonctions d'interpolation attachées aux nœuds (x_i, y_i) du triangle ayant comme valeurs nodales \mathbf{u}_i et qui sont définies par : $N_i(x, y) = \frac{1}{2A} (\delta_{jk} + y_{jk} x - x_{jk} y)$.

Si nous les présentons sous une forme matricielle, nous avons

$$\begin{pmatrix} N_1(x, y) \\ N_2(x, y) \\ N_3(x, y) \end{pmatrix} = \frac{1}{2A} \begin{bmatrix} y_{23} & x_{32} & \delta_{23} \\ y_{31} & x_{13} & \delta_{31} \\ y_{12} & x_{21} & \delta_{12} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

donnant comme relation inverse :

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{pmatrix} N_1(x, y) \\ N_2(x, y) \\ N_3(x, y) \end{pmatrix},$$

A noter que la somme des $N_i(x, y)$ vaut ainsi 1, et que nous avons comme relations : $x = N_1(x, y) x_1 + N_2(x, y) x_2 + N_3(x, y) x_3$ et $y = N_1(x, y) y_1 + N_2(x, y) y_2 + N_3(x, y) y_3$.

L'approximation nodale est généralement construite sur des éléments dits de référence ayant une forme géométrique simple. Puis, une transformation géométrique est utilisée pour passer de l'élément de référence à l'élément dit réel, c'est-à-dire à l'élément quelconque du domaine discrétisé. Pour cela, il faut bien entendu que l'élément de référence et l'élément réel soient de même nature. A noter qu'il existe deux types d'élément de référence. Les éléments de type Lagrange qui disposent d'une variable nodale par nœuds de l'élément. Il faut ainsi augmenter le nombre de nœuds pour augmenter le nombre de variables nodales. Les éléments de type Hermite dont le nombre de variables nodales est augmenté en considérant également les valeurs des dérivées du champ aux nœuds.

Triangle T3 ou CST (*Constant Strain Triangle*)²¹. Comme élément de référence pour un triangle 2D, nous pouvons considérer un triangle rectangle défini dans le repère de référence $\mathbf{s} = (s, t)$ par les 3 nœuds de coordonnées $(0, 0)$, $(1, 0)$, $(0, 1)$.

Comme valeurs respectives de ces nœuds, nous considérons les valeurs $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$. Nous faisons ainsi correspondre les nœuds $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ de l'élément réel défini précédemment avec les 3 nœuds respectifs de cet élément de référence. Le point (s, t) sur l'élément de référence est ainsi le correspondant du point (x, y) de l'élément réel.

Construisons les fonctions d'interpolation de cet élément de référence en utilisant, de la même façon que pour l'élément réel, la base polynomiale linéaire $(1, s, t)$. Nous avons vu que la solution approchée était de la forme $\mathbf{u}^*(M) = \mathbf{u}^*(s, t) = \mathbf{a}_1 + \mathbf{a}_2 s + \mathbf{a}_3 t$.

En identifiant le champ de déplacement aux 3 valeurs nodales du triangle, c'est-à-dire en considérant que $u^*(0, 0) = \mathbf{u}_1$, $u^*(1, 0) = \mathbf{u}_2$ et $u^*(0, 1) = \mathbf{u}_3$, nous obtenons :

$$\mathbf{a}_1 = \mathbf{u}_1, \mathbf{a}_2 = \mathbf{u}_2 - \mathbf{u}_1 \text{ et } \mathbf{a}_3 = \mathbf{u}_3 - \mathbf{u}_1.$$

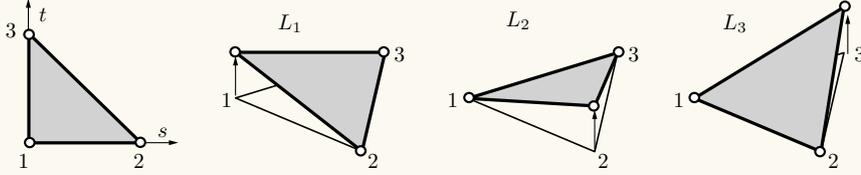
21. Je présente en détails ce type d'élément de référence appelé T3 ou CST (*Constant Strain Triangle*) que nous utiliserons au chapitre 2 dans la modélisation des plaques fines.

La solution approchée est alors de la forme :

$$\mathbf{u}^*(M) = \mathbf{u}^*(s, t) = (1 \ s \ t) \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 - \mathbf{u}_1 \\ \mathbf{u}_3 - \mathbf{u}_1 \end{pmatrix} = \underbrace{(1 - s - t \ s \ t)}_{\mathbf{N}(s,t)} \underbrace{\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix}}_{\mathbf{u}_n}.$$

Nous obtenons ainsi les fonctions d'interpolation linéaires représentées ci-dessous :

$$N_1(s, t) = L_1 = 1 - s - t, \quad N_2(s, t) = L_2 = s, \quad N_3(s, t) = L_3 = t.$$



A noter que l'expression de ces fonctions d'interpolation est plus simple sur l'élément de référence que sur l'élément réel et ne dépend pas des coordonnées de l'élément réel.

La somme de ces fonctions d'interpolation vaut ainsi toujours 1 et nous pouvons également vérifier la relation $N_j(M_i) = \delta_{ij}$ avec

$$\begin{cases} N_1(0, 0) = 1, & N_1(0, 1) = 0, & N_1(1, 0) = 0 \\ N_2(0, 0) = 0, & N_2(0, 1) = 0, & N_2(1, 0) = 1 \\ N_3(0, 0) = 0, & N_3(0, 1) = 1, & N_3(1, 0) = 0 \end{cases}$$

Quantités élémentaires. La méthode des éléments finis repose sur la discrétisation du domaine en éléments. Il est alors nécessaire de définir pour chacun de ces éléments les formes matricielles et vectorielles issues de la formulation intégrale du problème de la dynamique newtonienne. Pour présenter cette démarche, prenons comme point de départ la forme intégrale de notre problème issue du Principe des Travaux Virtuels (PTV).

Étant donné un système matériel D_t , l'idée du PTV consiste à définir en fonction d'un déplacement virtuel $\delta\vec{u}$, le travail virtuel des quantités d'accélération δA et le travail virtuel des efforts δT . Sachant que nous avons à tout instant $\delta A = \delta T$, nous avons :

$$\forall \delta\vec{u}, \underbrace{\int_{D_t} (\rho \delta\vec{u} \cdot \ddot{\vec{u}}) dv}_{\delta A} = - \underbrace{\int_{D_t} (\bar{\sigma} : \overline{\text{grad}} \delta\vec{u}) dv}_{\delta T} + \underbrace{\int_{D_t} (\delta\vec{u} \cdot \rho \vec{f}_{ext}) dv + \int_{\partial D_t} (\delta\vec{u} \cdot \vec{T}) ds}_{\delta T}.$$

Ensuite pour chacun des éléments D_e du domaine discrétisé, nous utilisons l'approximation nodale $\mathbf{u}^*(M)$ pour exprimer le champ de déplacement \vec{u} et le champ des déplacements virtuels $\delta\vec{u}$. Cette approximation étant définie pour tout point M de D_e par $\mathbf{N}(M) \mathbf{u}_n$, nous obtenons pour le premier terme de l'égalité précédente :

$$\int_{D_e} (\rho \delta\vec{u}(M) \cdot \ddot{\vec{u}}(M)) dv = \delta\mathbf{u}_n^T \int_{D_e} (\mathbf{N}(M)^T \rho \mathbf{N}(M)) dv \ddot{\mathbf{u}}_n = \delta\mathbf{u}_n^T M_e \ddot{\mathbf{u}}_n$$

avec \mathbf{M}_e la **matrice élémentaire des masses** définie par

$$\mathbf{M}_e = \int_{D_e} \left(\mathbf{N}(M)^T \rho \mathbf{N}(M) \right) dv.$$

Par ailleurs, le **tenseur des déformations** peut être exprimé par

$$\epsilon(M) = \mathbf{L} \vec{u}(M) = \mathbf{L} \mathbf{N}(M) \mathbf{u}_n = \mathbf{B}(M) \mathbf{u}_n.$$

où \mathbf{L} est la matrice d'opérateurs différentiels correspondant à l'expression du gradient symétrique du champ de déplacements et \mathbf{B} la matrice d'opérateurs différentiels appliquées aux fonctions d'interpolations \mathbf{N} .

La loi de comportement permet par ailleurs de relier le tenseur de contraintes avec le tenseur des déformations avec

$$\sigma(M) = \mathbf{C}(M) \epsilon(M) = \mathbf{C}(M) \mathbf{B}(M) \mathbf{u}_n.$$

Etant données ces relations, nous pouvons écrire :

$$\begin{aligned} \int_{D_e} \left(\bar{\sigma} : \overline{\text{grad}} \delta \vec{u} \right) dv &= \int_{D_e} \left(\bar{\sigma} : \overline{\delta \epsilon} \right) dv \\ &= \delta \mathbf{u}_n^T \int_{D_e} \left(\mathbf{B}(M)^T \mathbf{C}(M) \mathbf{B}(M) \right) dv \mathbf{u}_n \\ &= \delta \mathbf{u}_n^T \mathbf{K}_e \mathbf{u}_n \end{aligned}$$

avec \mathbf{K}_e la **matrice élémentaire des raideurs** définie par

$$\mathbf{K}_e = \int_{D_e} \left(\mathbf{B}(M)^T \mathbf{C}(M) \mathbf{B}(M) \right) dv.$$

Enfin nous avons également :

$$\int_{D_e} \left(\delta \vec{u} \cdot \rho \overrightarrow{f_{d_{ext}}} \right) dv + \int_{\partial D_e} \left(\delta \vec{u} \cdot \overrightarrow{T_d} \right) ds = \delta \mathbf{u}_n^T \mathbf{F}_{de}$$

avec $\mathbf{F}_{de} = \int_{D_e} \left(\mathbf{N}(M)^T \rho \mathbf{f}_{d_{ext}} \right) dv + \int_{\partial D_e} \left(\mathbf{N}(M)^T \mathbf{T}_d \right) ds$

où $\overrightarrow{f_{d_{ext}}}$ et $\overrightarrow{T_d}$ représentent les forces extérieures et le vecteur de contraintes pour l'élément D_e qui sont ensuite exprimés dans une base cohérente avec le même choix de la discrétisation que $\delta \vec{u}$ pour devenir respectivement $\mathbf{f}_{d_{ext}}$ et \mathbf{T}_d .

Au final, si nous reprenons la formulation intégrale issue du PTV, nous obtenons l'équation matricielle suivante pour chacun des éléments D_e du domaine discrétisé :

$$\begin{aligned} \forall D_e, \quad \delta \mathbf{u}_n^T \mathbf{M}_e \ddot{\mathbf{u}}_n + \delta \mathbf{u}_n^T \mathbf{K}_e \mathbf{u}_n &= \delta \mathbf{u}_n^T \mathbf{F}_{de} \\ \Rightarrow \mathbf{M}_e \ddot{\mathbf{u}}_n + \mathbf{K}_e \mathbf{u}_n &= \mathbf{F}_{de} \end{aligned}$$

Matrice $\mathbf{B}(M)$ pour un élément triangulaire 2D. Reprenons notre élément réel triangulaire 2D. Le champ de déplacement est défini dans le plan par le vecteur $(u \ v)^T$.

Le tenseur des déformations s'écrit alors :

$$\epsilon(M) = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2 \epsilon_{xy} \end{pmatrix} = \mathbf{L} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{pmatrix} = \begin{pmatrix} u_{,x} \\ v_{,y} \\ u_{,y} + v_{,x} \end{pmatrix}$$

où \mathbf{L} est la matrice d'opérateurs différentiels. En prenant l'approximation nodale du champ de déplacement définie par $\mathbf{u}^*(M) = \mathbf{N}(M) \mathbf{u}_n$, nous avons :

$$\begin{cases} u(x, y) = u_1 N_1(x, y) + u_2 N_2(x, y) + u_3 N_3(x, y) \\ v(x, y) = v_1 N_1(x, y) + v_2 N_2(x, y) + v_3 N_3(x, y) \end{cases}$$

$$\Rightarrow \begin{cases} u_{,x} = u_1 N_{1,x} + u_2 N_{2,x} + u_3 N_{3,x} \\ v_{,y} = v_1 N_{1,y} + v_2 N_{2,y} + v_3 N_{3,y} \\ u_{,y} + v_{,x} = u_1 N_{1,y} + v_1 N_{1,x} + u_2 N_{2,y} + v_2 N_{2,x} + u_3 N_{3,y} + v_3 N_{3,x} \end{cases}$$

Le tenseur des déformations s'exprime alors sous la forme suivante :

$$\epsilon(M) = \underbrace{\begin{bmatrix} N_{1,x} & 0 & N_{2,x} & 0 & N_{3,x} & 0 \\ 0 & N_{1,y} & 0 & N_{2,y} & 0 & N_{3,y} \\ N_{1,y} & N_{1,x} & N_{2,y} & N_{2,x} & N_{3,y} & N_{3,x} \end{bmatrix}}_{\mathbf{L} \mathbf{N}(x,y)} \underbrace{\begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{pmatrix}}_{\mathbf{u}_n} = \mathbf{B}(M) \mathbf{u}_n.$$

Pour construire la matrice $\mathbf{B}(M) = \mathbf{L} \mathbf{N}(x, y)$, nous devons ainsi calculer les dérivées premières des fonctions d'interpolation $\mathbf{N}(M)$ par rapport aux coordonnées (x, y) .

En considérant les fonctions $N_i(x, y)$ associées au triangle, nous obtenons :

$$\begin{cases} \frac{\partial}{\partial x} N_1(x, y) = N_{1,x} = \frac{1}{2A} y_{23} & ; & \frac{\partial}{\partial y} N_1(x, y) = N_{1,y} = \frac{1}{2A} x_{32} \\ \frac{\partial}{\partial x} N_2(x, y) = N_{2,x} = \frac{1}{2A} y_{31} & ; & \frac{\partial}{\partial y} N_2(x, y) = N_{2,y} = \frac{1}{2A} x_{13} \\ \frac{\partial}{\partial x} N_3(x, y) = N_{3,x} = \frac{1}{2A} y_{12} & ; & \frac{\partial}{\partial y} N_3(x, y) = N_{3,y} = \frac{1}{2A} x_{21} \end{cases}$$

$$\Rightarrow \frac{\partial}{\partial x} N_i(x, y) = N_{i,x} = \frac{1}{2A} y_{jk} \quad \text{et} \quad \frac{\partial}{\partial y} N_i(x, y) = N_{i,y} = \frac{1}{2A} x_{kj}$$

que nous pouvons exprimer sous la forme pseudo matricielle suivante :

$$[N_{,j}] = \begin{bmatrix} N_{1,x} & N_{1,y} \\ N_{2,x} & N_{2,y} \\ N_{3,x} & N_{3,y} \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} y_{23} & x_{32} \\ y_{31} & x_{13} \\ y_{12} & x_{21} \end{bmatrix}.$$

Nous pouvons également noter que :

$$\begin{cases} \frac{\partial x}{\partial N_1(x, y)} = x_1 ; \frac{\partial x}{\partial N_2(x, y)} = x_2 ; \frac{\partial x}{\partial N_3(x, y)} = x_3 \\ \frac{\partial y}{\partial N_1(x, y)} = y_1 ; \frac{\partial y}{\partial N_2(x, y)} = y_2 ; \frac{\partial y}{\partial N_3(x, y)} = y_3 \end{cases}$$

$$\Rightarrow \frac{\partial x}{\partial N_i(x, y)} = x_i ; \frac{\partial y}{\partial N_i(x, y)} = y_i.$$

Au final, la matrice $\mathbf{B}(M)$ relative à l'élément triangulaire 2D est définie par :

$$\mathbf{B}(x, y) = \frac{1}{2A} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix}.$$

Elle est ainsi identique pour tout point $M = (x, y)$ de l'élément considéré, c'est-à-dire qu'elle est constante sur l'élément. Il en est de même pour le tenseur $\epsilon(M) = \mathbf{B}(M) \mathbf{u}_n$.

Cette propriété a le mérite de faciliter le calcul de la matrice élémentaire des raideurs. Pour un élément triangulaire de surface A et d'épaisseur h , nous avons ainsi :

$$\mathbf{K}_e = \int_{D_e} \left(\mathbf{B}(M)^T \mathbf{C}(M) \mathbf{B}(M) \right) dv = h A \left(\mathbf{B}(M)^T \mathbf{C}(M) \mathbf{B}(M) \right).$$

Passage de l'élément de référence à l'élément réel. Les expressions des matrices élémentaires utilisent des opérateurs différentiels et des intégrales sur le domaine élémentaire D_e . Pour effectuer ces calculs, nous les réalisons tout d'abord sur les éléments de référence (pour lesquels les fonctions d'interpolation et leurs dérivées sont connues et sur lesquels les calculs peuvent être réalisés analytiquement), puis nous employons une transformation géométrique pour passer de l'élément de référence à l'élément réel.

Cette transformation géométrique permet ainsi de définir les coordonnées (x, y, z) de tout point 3D de l'élément réel à partir des coordonnées (s, t, u) du point 3D correspondant de l'élément de référence. Par ailleurs, cette transformation géométrique est une bijection, c'est-à-dire que tout point (x, y, z) de l'élément réel a un seul antécédent de coordonnées (s, t, u) et est ainsi l'image de ce point. Par contre, un élément de référence permet de générer une classe d'éléments réels en fonction de la transformation géométrique employée.

A noter que si les nœuds d'interpolation et les nœuds géométriques sont confondus, les fonctions de transformations géométriques \mathbf{N}_g sont identiques aux fonctions d'interpolation \mathbf{N} utilisées pour définir l'approximation nodale. L'élément de référence est alors dit **iso-paramétrique**. La transformation géométrique est dans ce cas définie par :

$$x = \mathbf{N}_g(s, t, u) \mathbf{x}_n ; y = \mathbf{N}_g(s, t, u) \mathbf{y}_n ; z = \mathbf{N}_g(s, t, u) \mathbf{z}_n.$$

avec $\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n$ les coordonnées des n nœuds de l'élément réel et $\mathbf{N}_g(s, t, u)$ les fonctions de transformation géométrique permettant de passer de l'élément réel à l'élément de référence.

Revenons aux calculs relatifs à la définition des matrices élémentaires. Nous avons vu que ces calculs faisaient apparaître des opérateurs différentiels appliqués aux fonctions d'interpolation \mathbf{N} . Or en pratique nous ne connaissons que les dérivées des fonctions d'interpolation par rapport aux coordonnées $\mathbf{r} = (s, t, u)$ de l'élément de référence, alors qu'il est nécessaire de savoir les exprimer par rapport aux coordonnées réelles $\mathbf{x} = (x, y, z)$.

Pour cela nous définissons la **matrice jacobienne de la transformation** \mathbf{J} en utilisant les règles de dérivation en chaîne. Nous obtenons ainsi :

$$\underbrace{\begin{pmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \\ \frac{\partial}{\partial u} \end{pmatrix}}_{\frac{\partial}{\partial \mathbf{r}}} = \underbrace{\begin{bmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \\ \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} & \frac{\partial z}{\partial u} \end{bmatrix}}_{\frac{\partial \mathbf{x}}{\partial \mathbf{r}} = \mathbf{J}} \underbrace{\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix}}_{\frac{\partial}{\partial \mathbf{x}}}.$$

En considérant un élément de référence iso-paramétrique, nous pouvons alors écrire que la matrice jacobienne est définie par :

$$\begin{aligned} \mathbf{J} = \frac{\partial}{\partial \mathbf{r}} \mathbf{x} &= \begin{pmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \\ \frac{\partial}{\partial u} \end{pmatrix} \begin{pmatrix} x & y & z \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \\ \frac{\partial}{\partial u} \end{pmatrix} \begin{pmatrix} \mathbf{N}_g(s, t, u) \mathbf{x}_n & \mathbf{N}_g(s, t, u) \mathbf{y}_n & \mathbf{N}_g(s, t, u) \mathbf{z}_n \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \\ \frac{\partial}{\partial u} \end{pmatrix} \begin{pmatrix} \sum_{i=1}^n N_{ig}(s, t, u) x_i & \sum_{i=1}^n N_{ig}(s, t, u) y_i & \sum_{i=1}^n N_{ig}(s, t, u) z_i \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \\ \frac{\partial}{\partial u} \end{pmatrix} \begin{bmatrix} N_{1g}(s, t, u) & \dots & N_{ng}(s, t, u) \\ N_{1g}(s, t, u) & \dots & N_{ng}(s, t, u) \\ N_{1g}(s, t, u) & \dots & N_{ng}(s, t, u) \end{bmatrix} \begin{bmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} & \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} & \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix} \end{bmatrix} \\ &= \begin{pmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \\ \frac{\partial}{\partial u} \end{pmatrix} \begin{bmatrix} \mathbf{N}_g(s, t, u) \\ \mathbf{N}_g(s, t, u) \\ \mathbf{N}_g(s, t, u) \end{bmatrix} \begin{bmatrix} \mathbf{x}_n & \mathbf{y}_n & \mathbf{z}_n \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial}{\partial s} \mathbf{N}_g(s, t, u) \\ \frac{\partial}{\partial t} \mathbf{N}_g(s, t, u) \\ \frac{\partial}{\partial u} \mathbf{N}_g(s, t, u) \end{bmatrix} \begin{bmatrix} \mathbf{x}_n & \mathbf{y}_n & \mathbf{z}_n \end{bmatrix}. \end{aligned}$$

Ainsi pour chaque élément de la discrétisation, nous pouvons exprimer la matrice jacobienne \mathbf{J} en fonction des dérivées des fonctions connues de transformation géométrique \mathbf{N}_g et des coordonnées $\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n$ des nœuds géométriques de l'élément réel. \mathbf{J} est ainsi le produit d'une matrice $3 \times n$ par une matrice $n \times 3$, toutes deux connues.

La relation inverse \mathbf{J}^{-1} (qui existe puisque la transformation géométrique doit être une bijection) permet par ailleurs de calculer les dérivées premières par rapport aux coordonnées réelles des fonctions d'interpolation avec :

$$\frac{\partial}{\partial \mathbf{x}} = \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \frac{\partial}{\partial \mathbf{r}} \Rightarrow \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} = \underbrace{\begin{bmatrix} \frac{\partial s}{\partial x} & \frac{\partial t}{\partial x} & \frac{\partial u}{\partial x} \\ \frac{\partial s}{\partial y} & \frac{\partial t}{\partial y} & \frac{\partial u}{\partial y} \\ \frac{\partial s}{\partial z} & \frac{\partial t}{\partial z} & \frac{\partial u}{\partial z} \end{bmatrix}}_{\mathbf{J}^{-1}} \begin{pmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \\ \frac{\partial}{\partial u} \end{pmatrix}.$$

Connaissant les dérivées premières des fonctions d'interpolations N_i par rapport aux coordonnées de l'élément de référence (s, t, u) , ces relations contribuent à la construction de la matrice $\mathbf{B}(s, t, u)$ à partir de la matrice $\mathbf{B}(x, y, z)$ s'exprimant en fonction des dérivées premières des fonctions d'interpolations N_i par rapport aux coordonnées réelles (x, y, z) .

La jacobienne de la transformation permet aussi de passer de l'intégrale d'une fonction $f(x, y, z)$ définie sur l'élément réel D_e à l'intégration sur l'élément de référence D_{ref} avec

$$\int_{D_e} f(x, y, z) dx dy dz = \int_{D_{ref}} f(s, t, u) \det(\mathbf{J}) ds dt du.$$

A noter que le calcul de l'intégration se fait rarement de manière analytique, mais plutôt de manière numérique sous la forme

$$\int_{D_{ref}} f(s, t, u) ds dt du = \sum_{i=1}^{N_{pi}} f(\xi_i) w_i$$

avec N_{pi} le nombre de points d'intégration sur l'élément de référence D_{ref} , ξ_i les coordonnées paramétriques des points d'intégration et w_i les poids d'intégration.

Les matrices élémentaires s'écrivent alors sous la forme suivante :

- matrice élémentaire des masses : $\mathbf{M}_e = \int_{D_{ref}} \left(\mathbf{N}(\xi)^T \rho \mathbf{N}(\xi) \det(\mathbf{J}) \right) ds dt du,$
- matrice élémentaire des raideurs : $\mathbf{K}_e = \int_{D_{ref}} \left(\mathbf{B}(\xi)^T \mathbf{C}(\xi) \mathbf{B}(\xi) \det(\mathbf{J}) \right) ds dt du,$
- vecteur des forces extérieures : $\mathbf{F}_{de} = \int_{D_{ref}} \left(\mathbf{N}(\xi)^T \rho \mathbf{f}_{d_{ext}} \det(\mathbf{J}) \right) ds dt du.$

Triangle T3 ou CST (*Constant Strain Triangle*). Nous avons vu que pour un élément réel triangulaire 2D défini par les nœuds (x_1, y_1) , (x_2, y_2) et (x_3, y_3) , nous pouvons utiliser l'élément de référence T3 défini par les nœuds $(0, 0)$, $(1, 0)$ et $(0, 1)$.

Nous devons alors définir la transformation géométrique qui va permettre de passer de l'élément de référence (défini dans le repère \mathbf{r}) à notre élément réel (défini dans le repère \mathbf{x}). Il faut savoir que l'élément T3 est iso-paramétrique. Les nœuds utilisés pour l'interpolation de l'approximation nodale du champs de déplacement sont ainsi confondus avec les nœuds géométriques utilisés pour la transformation géométrique.

Les fonctions de transformation géométrique $\mathbf{N}_g(s, t)$ sont dans ce cas identiques aux fonctions $\mathbf{N}(s, t)$ de l'approximation nodale. Dans le cas de l'utilisation de la base polynomiale linéaire $(1, s, t)$, nous avons ainsi : $\mathbf{N}_g(s, t) = \mathbf{N}(s, t) = (1 - s - t \ s \ t)$.

La transformation géométrique est alors définie par

$$\begin{aligned} \mathbf{x} &= (1 - s - t \ s \ t) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = x_{21} s + x_{31} t + x_1 \\ \mathbf{y} &= (1 - s - t \ s \ t) \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = y_{21} s + y_{31} t + y_1 \end{aligned}$$

Ensuite, pour le calcul des opérateurs différentiels, nous devons définir la matrice jacobienne de la transformation géométrique. Pour un élément, elle est définie par :

$$\begin{aligned} \mathbf{J} &= \begin{bmatrix} \frac{\partial}{\partial s} \mathbf{N}_g(s, t) \\ \frac{\partial}{\partial t} \mathbf{N}_g(s, t) \end{bmatrix} \begin{bmatrix} \mathbf{x}_n & \mathbf{y}_n \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial}{\partial s} (1 - s - t \ s \ t) \\ \frac{\partial}{\partial t} (1 - s - t \ s \ t) \end{bmatrix} \begin{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} & \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \end{bmatrix} \\ &= \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} & \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \end{bmatrix} \\ &= \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix} = \begin{bmatrix} x_{21} & y_{21} \\ x_{31} & y_{31} \end{bmatrix}. \end{aligned}$$

La matrice inverse est ainsi définie par :

$$\mathbf{J}^{-1} = \frac{1}{2A} \begin{bmatrix} y_{31} & -y_{21} \\ -x_{31} & x_{21} \end{bmatrix}$$

où A est l'aire de l'élément réel et $\det(\mathbf{J}) = 2A = x_{21} y_{31} - x_{31} y_{21}$.

\mathbf{J} étant constante sur l'élément, nous pouvons calculer analytiquement les intégrales des matrices élémentaires. Pour cela, notons h l'épaisseur uniforme de l'élément de référence. Ses frontières sont données par les droites $s = 0, t = 0$ et $s + t = 1$. Dans cet élément de référence, t varie ainsi de 0 à 1. Ensuite pour t fixé, s varie de 0 à $1 - s$.

Le calcul des intégrales élémentaires s'écrit ainsi de la façon suivante :

— pour la matrice élémentaire des masses :

$$\begin{aligned} \mathbf{M}_e &= \int_{D_e} \left(\mathbf{N}(x, y)^T \rho \mathbf{N}(x, y) \right) dx dy dz \\ &= h \int_{D_{ref}} \left(\mathbf{N}(s, t)^T \rho \mathbf{N}(s, t) \det(\mathbf{J}) \right) ds dt \\ &= h \int_{t=0}^1 \left(\int_{s=0}^{1-s} \left(\mathbf{N}(s, t)^T \rho \mathbf{N}(s, t) 2A \right) ds \right) dt \end{aligned}$$

— pour la matrice élémentaire des raideurs :

$$\begin{aligned} \mathbf{K}_e &= \int_{D_e} \left(\mathbf{B}(x, y)^T \mathbf{C}(x, y) \mathbf{B}(x, y) \right) dx dy dz \\ &= h \int_{D_{ref}} \left(\mathbf{B}(s, t)^T \mathbf{C}(s, t) \mathbf{B}(s, t) \det(\mathbf{J}) \right) ds dt \\ &= h \int_{t=0}^1 \left(\int_{s=0}^{1-s} \left(\mathbf{B}(s, t)^T \mathbf{C}(s, t) \mathbf{B}(s, t) 2A \right) ds \right) dt \end{aligned}$$

Il faut ainsi définir la matrice $\mathbf{B}(M)$ nécessitant le calcul des dérivées premières des fonctions d'interpolation \mathbf{N} sachant que $\epsilon(M) = \mathbf{L} \mathbf{u}(M) = \mathbf{L} \mathbf{N}(M) \mathbf{u}_n = \mathbf{B}(M) \mathbf{u}_n$.

Le champ de déplacement $\mathbf{u} = (u \ v)^T$ peut alors être vu comme une fonction de (x, y) ou de (s, t) . Si nous considérons l'approximation nodale dans le repère \mathbf{r} avec $\mathbf{u}^*(s, t) = \mathbf{N}(s, t) \mathbf{u}_n$, les composantes u et v sont définies en fonction de (s, t) par :

$$\begin{pmatrix} u(s, t) \\ v(s, t) \end{pmatrix} = \underbrace{\begin{bmatrix} N_1(s, t) & 0 & N_2(s, t) & 0 & N_3(s, t) & 0 \\ 0 & N_1(s, t) & 0 & N_2(s, t) & 0 & N_3(s, t) \end{bmatrix}}_{\mathbf{N}(s, t)} \underbrace{\begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{pmatrix}}_{\mathbf{u}_n}$$

$$\Rightarrow \begin{cases} u(s, t) = (1 - s - t) u_1 + s u_2 + t u_3 = (u_2 - u_1) s + (u_3 - u_1) t + u_1 \\ v(s, t) = (1 - s - t) v_1 + s v_2 + t v_3 = (v_2 - v_1) s + (v_3 - v_1) t + v_1 \end{cases}$$

Nous pouvons alors facilement calculer les dérivées premières par rapport à \mathbf{r} :

$$\frac{\partial \mathbf{u}(s, t)}{\partial \mathbf{r}} = \begin{pmatrix} \frac{\partial u(s, t)}{\partial s} \\ \frac{\partial u(s, t)}{\partial t} \end{pmatrix} = \begin{pmatrix} u_2 - u_1 \\ u_3 - u_1 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

$$\frac{\partial v(s, t)}{\partial \mathbf{r}} = \begin{pmatrix} \frac{\partial v(s, t)}{\partial s} \\ \frac{\partial v(s, t)}{\partial t} \end{pmatrix} = \begin{pmatrix} v_2 - v_1 \\ v_3 - v_1 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

Puis, nous avons les relations suivantes en utilisant les règles de dérivation en chaînes :

$$\frac{\partial u(s, t)}{\partial \mathbf{x}} = \underbrace{\frac{\partial \mathbf{r}}{\partial \mathbf{x}}}_{\mathbf{J}^{-1}} \frac{\partial u(s, t)}{\partial \mathbf{r}} ; \quad \frac{\partial v(s, t)}{\partial \mathbf{x}} = \underbrace{\frac{\partial \mathbf{r}}{\partial \mathbf{x}}}_{\mathbf{J}^{-1}} \frac{\partial v(s, t)}{\partial \mathbf{r}}$$

$$\begin{aligned} \Rightarrow \frac{\partial u(s, t)}{\partial \mathbf{x}} &= \begin{pmatrix} \frac{\partial u(s, t)}{\partial x} \\ \frac{\partial u(s, t)}{\partial y} \end{pmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial u(s, t)}{\partial s} \\ \frac{\partial u(s, t)}{\partial t} \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} y_{31} & -y_{21} \\ -x_{31} & x_{21} \end{bmatrix} \begin{bmatrix} u_2 - u_1 \\ u_3 - u_1 \end{bmatrix} \\ &= \frac{1}{2A} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} \Rightarrow \frac{\partial v(s, t)}{\partial \mathbf{x}} &= \begin{pmatrix} \frac{\partial v(s, t)}{\partial x} \\ \frac{\partial v(s, t)}{\partial y} \end{pmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial v(s, t)}{\partial s} \\ \frac{\partial v(s, t)}{\partial t} \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} y_{31} & -y_{21} \\ -x_{31} & x_{21} \end{bmatrix} \begin{bmatrix} v_2 - v_1 \\ v_3 - v_1 \end{bmatrix} \\ &= \frac{1}{2A} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \end{aligned}$$

Nous obtenons ainsi :

$$\epsilon(M) = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2 \epsilon_{xy} \end{pmatrix} = \begin{pmatrix} u_{,x} \\ v_{,y} \\ u_{,y} + v_{,x} \end{pmatrix} = \frac{1}{2A} \underbrace{\begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix}}_{\mathbf{B}(M)} \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{pmatrix}.$$

Le champ des déformations $\epsilon(M)$ est ainsi constant à l'intérieur de l'élément de référence ainsi appelé élément T3 ou CST pour *Constant Strain Triangle*.

Nous retombons sur la formulation que nous avons établie précédemment à partir des fonctions d'interpolations $\mathbf{N}_i(x, y)$ exprimées en fonction des coordonnées du repère de l'élément réel. Mais cette fois-ci, la formulation a été établie en partant des expressions des fonctions d'interpolation $\mathbf{N}_i(s, t)$ établies à partir du repère de l'élément de référence.

Assemblage. Après avoir défini une approximation de la solution sur chacun des éléments finis, et construit pour chacun de ces éléments les matrices élémentaires de la forme intégrale du problème, la méthode des éléments finis nécessite l'assemblage de ces matrices élémentaires. Nous obtenons ainsi les équations relatives au domaine D_t entier pour permettre ensuite sa résolution globale. En effet, il est important de noter qu'il n'est pas possible de résoudre indépendamment chacun des systèmes puisque les conditions aux frontières sont définies uniquement sur les frontières du système global et non sur les frontières de chacun des éléments finis.

L'assemblage des matrices élémentaires de masse M_e et de raideur K_e , ainsi que l'assemblage du vecteur des forces s'effectuent en sommant les termes correspondant au travail virtuel calculé pour chaque élément. Si nous considérons le domaine D_t décomposé en n_e éléments, nous avons comme relations définissant les matrices et vecteur globaux :

$$\left\{ \begin{array}{l} \sum_{e=1}^{n_e} D_e \simeq D_t, \\ \sum_{e=1}^{n_e} \delta \mathbf{u}_n^T \mathbf{M}_e \ddot{\mathbf{u}}_n = \delta \mathbf{U}^T \mathbf{M} \ddot{\mathbf{U}}, \\ \sum_{e=1}^{n_e} \delta \mathbf{u}_n^T \mathbf{K}_e \mathbf{u}_n = \delta \mathbf{U}^T \mathbf{K} \mathbf{U}, \\ \sum_{e=1}^{n_e} \delta \mathbf{u}_n^T \mathbf{F}_{de} = \delta \mathbf{U}^T \mathbf{F}_d \end{array} \right.$$

Après assemblage, nous obtenons au final la forme matricielle suivante de la formulation intégrale issue du Principe des Travaux Virtuels sur le domaine D_t :

$$\delta \mathbf{U}^T \mathbf{M} \ddot{\mathbf{U}} + \delta \mathbf{U}^T \mathbf{K} \mathbf{U} = \delta \mathbf{U}^T \mathbf{F}_d$$

$$\Rightarrow \mathbf{M} \ddot{\mathbf{U}} + \mathbf{K} \mathbf{U} = \mathbf{F}_d.$$

Cette équation est ensuite résolue en utilisant une méthode d'intégration numérique.

Récapitulatif des étapes de calculs de la MEF.

Pour résumer, voici les différentes étapes de calculs effectués lors de l'emploi de la méthode des éléments finis.

- **Pour les différents éléments de références utilisés**
 - Création de la table des coordonnées ξ_i et poids w_i pour les N_{pi} points d'intégration
 - Calcul des fonctions d'interpolation \mathbf{N}
 - Calcul des fonctions de transformation géométrique \mathbf{N}_g
 - Calcul des dérivées des fonctions \mathbf{N} et \mathbf{N}_g aux points d'intégration
- **Pour chaque élément faire**
 - **Pour chaque point d'intégration ξ_i faire**
 - Calcul de \mathbf{J} , \mathbf{J}^{-1} et $\det(\mathbf{J})$ au point d'intégration ξ_i
 - Construction des matrices \mathbf{C} , $\mathbf{B}(\xi_i)$ et $\mathbf{N}(\xi_i)$
 - Calcul de $\mathbf{B}(\xi_i)^T \mathbf{C} \mathbf{B}(\xi_i) \det(\mathbf{J}) w_i$
 - Calcul de $\mathbf{N}(\xi_i)^T \rho \mathbf{N}(\xi_i) \det(\mathbf{J}) w_i$
 - Calcul de $\mathbf{N}(\xi_i)^T \mathbf{f}_d(\xi_i) \det(\mathbf{J}) w_i$
 - Accumulation dans les matrices \mathbf{K} , \mathbf{M} et \mathbf{F}_d
 - **Fin** de l'intégration
- **Fin** de la boucle sur les éléments
- Emploi d'une méthode d'intégration numérique pour résoudre l'équation

1.11 Résumé des notions de linéarité ou non-linéarité

Pour finir ce chapitre, je reprends ici certaines explications du livre de [Craveur and Jetteur \(2010\)](#) sur les aspects de linéarité et de non-linéarité. Pour mieux les appréhender, je remets la Fig. 1.8 qui récapitule les liens que nous avons vus entre les différentes grandeurs physiques.

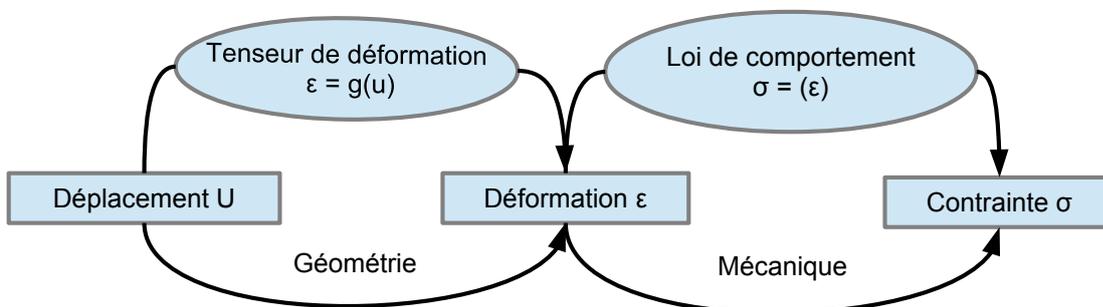


Figure 1.8 – Récapitulatif des relations existantes entre le déplacement $\bar{u}(t)$ de l'objet, sa déformation (tenseur $\bar{\varepsilon}$), et les contraintes (tenseur $\bar{\sigma}$) appliquées sur celui-ci.

Le tenseur de déformation relie ainsi les déplacements aux déformations avec une relation au niveau de la géométrie, et la loi de comportement relie le tenseur de déformation à la contrainte avec une relation au niveau de la mécanique.

Mécanique linéaire

Nous avons vu que le comportement d'un matériau est **élastique** si, quand nous arrêtons d'appliquer une contrainte sur celui-ci, le matériau revient au même point : son comportement est ainsi **réversible**. Si en plus, le même chemin est suivi dans le diagramme contrainte-déformation, nous disons que le matériau est **parfaitement élastique**. Ensuite, la mécanique linéaire repose sur l'hypothèse d'une proportionnalité entre les charges appliquées (ou contraintes) et les déplacements produits. Dit autrement, dans le cadre de l'**élasticité linéaire**, les relations entre les différentes grandeurs physiques sont linéaires, c'est-à-dire que nous avons une **linéarité matérielle** et une **linéarité géométrique**.

Relation géométrique entre déplacement et déformation

Dans le cas de la **linéarité géométrique**, la matrice \mathbf{B} de la cinématique interne est constante. Il y a alors une relation linéaire entre les déformations et les déplacements. C'est le cas quand les déformations axiales et les déformations angulaires sont infinitésimales. Nous sommes alors dans le domaine appelé des **petits déplacements** et nous pouvons employer le tenseur de déformation de Green-Lagrange linéarisé. La structure non-déformée est alors très proche de la structure déformée. Les charges sont implicitement indépendantes des déplacements subis. Nous pouvons alors découpler les comportements membranaires et flexionnels notamment dans le cas des plaques constituées de matériaux isotropes.

Dans le cas de la **non-linéarité géométrique**, il n'y a plus de proportionnalité entre les déformations et les déplacements, c'est-à-dire que les déformations ne sont plus des fonctions linéaires des déplacements. Il y a alors quatre cas possible, si nous considérons les ordres de grandeurs des déformations et des rotations :

- Quand les déformations et les rotations sont suffisamment petites, nous sommes dans le cadre de la linéarité géométrique. C'est la théorie des **petits déplacements**.
- Quand les déformation sont grandes, quelque soit l'amplitude des rotations (grandes ou petites), nous sommes dans la théorie des **grandes déformations** avec un comportement géométriquement non-linéaire.
- Quand les déformation sont petites et les rotations grandes, le comportement est géométriquement non-linéaire et nous sommes en **grands déplacements**.

Relation mécanique/matérielle entre déformation et contrainte

Dans le cas de la **linéarité matérielle**, la loi de comportement est représentée par une matrice \mathbf{C} qui est constante. Il y a ainsi une relation linéaire ou proportionnelle entre les contraintes et les déformations avec $\mathbf{C} = \mathbf{C}_0$ et $\bar{\sigma} = \mathbf{C}_0 \bar{\epsilon}$. De plus les coefficients élastiques (par exemple module de Young et coefficient de Poisson), à partir desquels les termes de la matrice \mathbf{C} sont déterminés, sont tous constants et indépendants de la déformation. Nous avons alors la loi de Hooke définie par $\sigma = \mathbf{C} \epsilon$.

Dans le cas de la **non-linéarité matérielle**, il n'y a pas de proportionnalité entre les contraintes et les déformations. La loi de comportement n'est ainsi plus décrite par des coefficients constants, et nous avons ainsi $\mathbf{C} = \mathbf{C}(\epsilon) \neq \mathbf{C}_0$, impliquant que la matrice de rigidité n'est pas constante avec $\mathbf{K} \neq \mathbf{K}_0$.

Linéarité matérielle et géométrique

Dans le cadre de l'élasticité linéaire, nous avons vu que nous avons une linéarité à la fois matérielle et géométrique. Cela revient à dire que la matrice des raideurs \mathbf{K} est constante. Et comme la matrice de raideur est la composition de la matrice des déformations \mathbf{B} décrivant la cinématique interne de l'élément et de la matrice \mathbf{C} décrivant la loi matérielle, nous retrouvons bien le fait que ces deux matrices sont constantes pour que \mathbf{K} le soit.

1.12 Conclusion

J'ai présenté dans ce chapitre les notions de la Mécanique des Milieux Continus qu'il est nécessaire d'acquérir quand nous souhaitons comprendre et modéliser le comportement d'un objet déformable. Si nous souhaitons être au plus proche de son comportement réel, il est nécessaire d'identifier par des expérimentations mécaniques, sa loi de comportement ainsi que ses paramètres mécaniques associés, permettant la définition des équations de la mécanique décrivant son mouvement. Ces équations sont ensuite résolues à chaque pas de temps de la simulation par la méthode dite des éléments finis. Dans le chapitre 2, nous allons utiliser toutes les notions acquises pour étudier le modèle mécanique d'une plaque mince sous les hypothèses de petits déplacements et de petites déformations.

Bibliographie

- M. Bailet. *Modèles biomécaniques pour la simulation interactive de l'accouchement*. Thèse de doctorat en informatique, Université Joseph Fourier, Dec. 2014. URL <https://hal.archives-ouvertes.fr/tel-01158380>.
- D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 43–54, New York, NY, USA, 1998. ACM. ISBN 0-89791-999-8. doi : 10.1145/280814.280821. URL <http://doi.acm.org/10.1145/280814.280821>.
- T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear finite elements for continua and structures*. J. Wiley & sons, Chichester, New York, Weinheim, 2000. ISBN 0-471-98773-5. URL <http://opac.inria.fr/record=b1128458>. Autres tirages : 2001, 2003, 2004, 2005, 2008.
- J. Coirier and C. Nadot-Martin. *Mécanique des milieux continus*. Dunod, 6 2013.
- J.-C. Craveur and P. Jetteur. *Introduction à la mécanique non linéaire*. Dunod, 2010.
- Y. Debard. Méthode des éléments finis : élasticité plane. Mar. 2011. URL <http://iut.univ-lemans.fr/ydlogi>.
- A. D. Deram. *Environnement générique pour la validation de simulations médicales*. Thèse de doctorat en informatique, Université de Grenoble, Oct. 2012. URL <https://tel.archives-ouvertes.fr/tel-00793236>.
- S. Forest, M. Amestoy, G. Damamme, S. Kruch, V. Maurel, and M. Mazière. Mécanique des milieux continus. 2010. URL http://mms2.ensmp.fr/mmc_paris/mmc_paris.php.
- H. Klöcker. Mécanique des milieux continus. 2003. URL http://mms2.mines-paristech.fr/ressources/ens_polycopies.php.
- V. Majorczyk. *Modélisation des interactions entre solides déformables et films fluides pour la simulation médicale temps-réel*. Thèse de doctorat en informatique, Université des Sciences et Technologies de Lille, Apr. 2015. URL <https://hal.inria.fr/tel-01165320>.
- N. Moës. Mécanique des milieux continus. Sept. 2011. URL <https://cel.archives-ouvertes.fr/cel-00612360>.
- C. Olognon. Elasticité et résistance des matériaux. 11 2010. URL <http://docinsa.insa-lyon.fr/polycop/pont.php?cle=RPNE000047A>.

- H. Oudin. Méthode des éléments finis. Lecture, Sept. 2008. URL <https://cel.archives-ouvertes.fr/cel-00341772>.
- E. Sifakis and J. Barbic. Fem simulation of 3d deformable solids : A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, pages 20 :1–20 :50, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1678-1. doi : 10.1145/2343483.2343501. URL <http://doi.acm.org/10.1145/2343483.2343501>.
- Wikipédia. Tenseur — wikipédia, l'encyclopédie libre, 2016. URL <https://fr.wikipedia.org/wiki/Tenseur>.

Chapitre 2

Modèle mécanique d'une plaque mince

Ce chapitre présente le modèle mécanique d'une plaque mince. Je présente tout d'abord le cadre théorique de son étude, puis je présenterai les différentes notions permettant d'élaborer une simulation d'une plaque mince sous les hypothèses de petits déplacements et de petites déformations. Nous verrons ainsi l'étude de sa cinématique, la définition de sa déformation en utilisant le tenseur de déformation de Green-Lagrange linéarisé, la définition du tenseur de contraintes, la formulation de la loi de Hooke pour une plaque mince, la définition de son énergie de déformation, et enfin la formulation de l'équation de son mouvement.

Sommaire

2.1	Introduction	68
2.2	Cadre théorique de l'étude d'une plaque mince	69
2.3	Cinématique d'une plaque mince	72
2.4	Déformation d'une plaque mince	74
2.5	Tenseur de contraintes et loi de comportement	76
2.6	Énergie de déformation d'une plaque mince	79
2.7	Équation du mouvement d'une plaque mince	81
2.8	Conclusion	82

2.1 Introduction

Dans le chapitre 1, nous avons vu les notions de la mécanique des milieux continus relatives à la modélisation des déformations subies par un objet déformable soumis à une charge. Dans ce chapitre, je vais étendre ces notions à la théorie des plaques et des coques employées dans la modélisation de structures fines.

J'ai souhaité là encore donner l'ensemble du raisonnement mécanique adopté pour la simulation des plaques minces pour permettre une compréhension complète des formulations développées durant la thèse de Mathieu Bailet que nous étudierons dans le chapitre 6 (tous les détails n'ayant pas été mis dans son manuscrit de doctorat).

Ces notions pourront par ailleurs être complétées par d'autres références comme le support d'enseignement de Sylvain [Drapier \(2016\)](#), une explication relative à l'emploi du code Aster écrite par Thomas [De Soza \(2013\)](#), ou encore les pages de Wikipédia¹ donnant de manière concise les premières briques de ces notions (pages relative à la résistance des matériaux, la théorie des poutres, la théorie des plaques, les coques). Je me sers notamment de quelques illustrations issues de Wikipédia. Et bien entendu pour en savoir plus, de nombreux ouvrages de mécanique des structures couvrent ce domaine.

Organisation de ce chapitre : Ce chapitre s'articule de la façon suivante :

- La section 2.2 présente le cadre théorie de l'étude mécanique d'une plaque mince avec notamment l'énoncé des hypothèses de la théorie de Kirchhoff-Love.
- La section 2.3 présente la cinématique d'une plaque mince, c'est-à-dire la formulation de son déplacement.
- La section 2.4 présente la déformation d'une plaque mince, c'est-à-dire la définition de son tenseur de déformation en fonction du déplacement.
- La section 2.5 présente le tenseur de contrainte d'une plaque mince et la formulation de la loi de Hooke reliant le tenseur de contrainte au tenseur de déformation.
- La section 2.6 présente la formulation de l'énergie de déformation d'une plaque mince.
- Enfin, la section 2.7 présente l'équation du mouvement relative à une plaque mince.

1. Pensez à regarder les versions françaises et anglaises des pages (non identiques mais complémentaires).

2.2 Cadre théorique de l'étude d'une plaque mince

La théorie des plaques s'inspirant de la théorie des poutres, je vais commencer par quelques rappels sur cette dernière. Ces théories font partie du domaine de la mécanique des structures qui est la mécanique des solides de dimensions finies où au minimum une des dimensions est faible devant les autres. La Fig. 2.1 de [Drapier \(2016\)](#) illustre les géométries le plus courantes dans ce domaine : en 1D, les poutres et en 2D, les plaques et les coques.

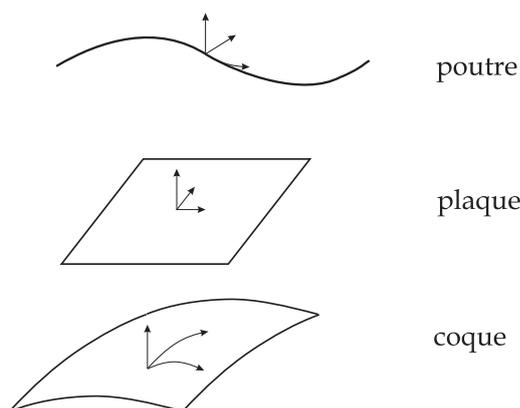


Figure 2.1 – Géométries courantes en mécanique des structures [[Drapier \(2016\)](#)].

Dans nos explications, nous nous restreignons au cadre de la **résistance des matériaux** dont les hypothèses de simplification concernent les conditions de réversibilité et de linéarité. Nous nous plaçons ainsi dans le cadre des **petites déformations et petits déplacements**, c'est-à-dire que les déformations de la structure issues de son chargement sont négligeables en n'affectant pratiquement pas sa géométrie. De plus, nous considérons des matériaux

- élastiques (les déformations sont réversibles, c'est-à-dire que le matériau reprend sa forme initiale après un cycle chargement-déchargement),
- linéaires (les déformations sont proportionnelles aux contraintes),
- homogènes (le matériau est de même nature dans toute sa masse),
- et isotropes (les propriétés du matériau sont identiques dans toutes les directions).

Nous considérons également les deux principes suivants :

- le **principe de Saint-Venant** disant que le comportement en un point quelconque du matériau est indépendant de la façon dont sont appliquées les forces (si ce point est suffisamment loin du point d'application) et peut ainsi être modélisé par un tenseur des forces internes en ce point ;
- le **principe de superposition** disant que l'ordre d'applications des efforts extérieurs sur le solide n'a pas d'importance, c'est-à-dire que quelque soit l'ordre d'applications des efforts nous obtenons le même état final. Ce principe permet de décomposer une sollicitation complexe en une somme de sollicitations élémentaires (traction, compression, cisaillement, torsion, flexion).

Définition d'une poutre. Une poutre est un solide qui a été engendré par le déplacement d'une surface finie S appelée *section droite*. Le mouvement durant ce déplacement du centre de gravité G de la surface S décrit une ligne appelée *courbe (ou fibre) moyenne* qui est une courbe continue. A noter que la section droite est constamment perpendiculaire à cette ligne, et ses deux dimensions sont petites par rapport à la longueur de la poutre. Enfin, si nous considérons une petite portion dS de la section droite suivant la courbe moyenne, celle-ci génère un volume appelé *fibre neutre*. Ces notions sont illustrées par la Fig. 2.2.

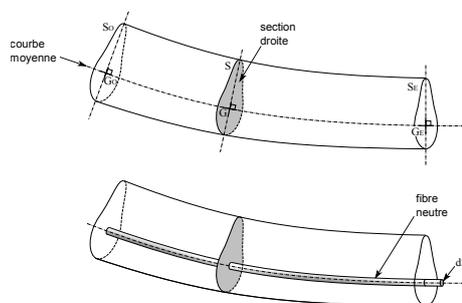


Figure 2.2 – Une poutre est engendrée par le mouvement d'une surface [Wikipédia (2015b)].

Ces hypothèses sur les dimensions de la poutre permettent de simplifier l'étude de son comportement lors d'un chargement. En effet, le fait de considérer que les dimensions des sections sont faibles par rapport à la direction principale de la courbe moyenne, permet de supposer que le déplacement de tout point de la poutre s'exprime simplement en fonction des déplacements et rotations des sections mesurées en leur centre de gravité.

Définition d'une plaque. Une plaque est définie comme étant un solide délimité par deux plans parallèles appelés faces et dont l'épaisseur h est petite par rapport aux deux autres dimensions. Les plaques sont généralement employées pour modéliser des structures minces, car seule une dimension est faible par rapport aux deux autres. Par contre, si la structure présente une courbure géométrique, ce sont les éléments de coque qui sont privilégiés plutôt que les éléments de plaque qui sont plans. Autrement dit, si l'élément n'est pas plan au repos, nous parlons de coque plutôt que de plaque (voir Fig. 2.1).

Pour étudier le comportement de la plaque, nous définissons son *plan moyen* auquel nous associons un repère orthonormé (O, x, y, z) tangent à ce plan. Le plan moyen est ainsi situé dans le plan $(0, x, y)$ à équidistance entre les deux faces. L'origine du repère étant située sur le plan moyen avec z la direction de l'épaisseur, la face inférieure est ainsi positionnée en $z = -h/2$ et la face supérieure en $z = h/2$. Dans la théorie des plaques, le plan moyen représente l'équivalent de la courbe moyenne dans la théorie des poutres.

Autour du plan moyen, le *feuillet neutre (ou feuillet moyen)* constitue un élément de matière d'épaisseur infinitésimale (c'est l'équivalent de la fibre neutre des poutres). Une *fibre normale* est ensuite définie comme l'ensemble des points situés sur une normale au plan moyen (ayant ainsi pour direction z) à une position (x, y) dans le plan $(0, x, y)$.

Pour illustrer ces notions, la Fig. 2.3 présente la déformation d'une plaque mince (pointillés gris) en mettant en évidence le déplacement d'un élément de matière (contour noir), de son feuillet moyen (rouge) et d'une fibre normale (bleue).

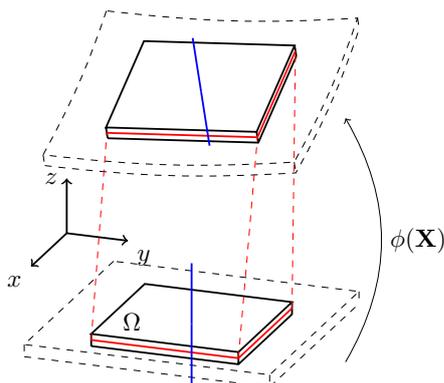


Figure 2.3 – Déformation d'une plaque (pointillés gris) avec mise en évidence du feuillet moyen (rouge) d'un élément de matière (noir) et d'une fibre normale (bleu) [Wikipédia (2015a)].

L'étude du comportement d'une plaque suit ensuite l'approche classique de la MMC décrite dans le chapitre 1. Le tenseur de contrainte σ en un point de la structure est ainsi relié au tenseur de déformation ϵ par la loi de comportement du matériau, et le champ de tenseurs des déformations est relié au champ des déplacements \vec{u} par sa formulation établie en fonction du gradient du déplacement.

- Dans le cas d'un matériau isotrope linéaire, nous considérons la **loi de Hooke généralisée** avec $\bar{\sigma} = E/(1 + \nu)[\bar{\epsilon} + \nu/(1 - 2\nu) \text{Tr}(\bar{\epsilon}) \mathbb{1}]$ avec E le module de Young et ν le coefficient de Poisson de l'objet considéré.
- De plus, dans le cadre de petites perturbations avec des petits déplacements, nous pouvons considérer le **tenseur de déformation de Green-Lagrange linéarisé** défini par $\epsilon^g = \frac{1}{2}(\mathbb{U}^T + \mathbb{U})$ avec \mathbb{U} le gradient du déplacement.

A noter que l'étude du comportement d'une plaque est séparée en deux parties :

- Pour l'étude de la **flexion de la plaque** (c'est-à-dire le fait que la structure soit courbée), nous considérons uniquement les charges perpendiculaires aux faces, c'est-à-dire correspondant à une force appliquée de la forme $\vec{F} = (0, 0, F)$.
- Pour les charges situées dans le plan des faces, nous parlons d'**efforts de membranes**. Dans ce cas, nous considérons que les efforts perpendiculaires sont nuls.

Théorie de Kirchhoff-Love pour les plaques minces. La théorie de Kirchhoff-Love est employée pour l'étude de plaques minces. Elle énonce les hypothèses suivantes :

- le plan moyen est initialement plan, c'est-à-dire qu'il ne présente pas de courbure ;
- le feuillet moyen ne subit pas de déformation dans son plan, c'est-à-dire que nous ne considérons que le déplacement transversal (noté w) des points du feuillet moyen ;
- les sections normales au feuillet moyen restent normales lors de la déformation, c'est-à-dire que nous pouvons négliger le cisaillement ;
- l'épaisseur de la plaque est faible, c'est-à-dire que dans la direction de l'épaisseur, la déformation est nulle impliquant que les contraintes dans cette direction peuvent être négligées ;
- et nous nous plaçons en petites déformations.

2.3 Cinématique d'une plaque mince

Nous considérons pour décrire le mouvement de la plaque, le repère local associé au plan moyen. La position de tout point de la plaque au repos est alors décrite par ses coordonnées cartésiennes exprimées dans ce repère avec $\mathbf{X} = (X_1, X_2, X_3)^T = (x, y, z)^T$.

Suite à la déformation ϕ appliquée à la plaque, le déplacement \vec{u} de ce point est défini par $\vec{u} = (u_1, u_2, u_3)^T = (u, v, w)^T$ correspondant au déplacement du point de la plaque entre l'état de référence et l'état déformé de la plaque. Les composantes du déplacement dépendent ainsi de la position du point de la plaque considéré dans l'état de référence. Autrement dit, les composantes u, v, w sont définies en fonction des composantes x, y, z .

Ensuite dans le même esprit que pour l'étude des poutres, nous allons exprimer ce déplacement en fonction du mouvement du plan moyen de la plaque. Ce mouvement peut être décomposé en une translation et une rotation.

Nous désignons par $\vec{u}_0 = (u_0(x, y, z), v_0(x, y, z), w_0(x, y, z))^T$ le déplacement par translation du plan moyen. Les composantes u_0, v_0 correspondent ainsi au déplacement du plan moyen dans le plan $(0, x, y)$. Le plan moyen étant positionné en $z = 0$, elles ne dépendent ainsi que de x et y avec $u_0(x, y, z) = u_0(x, y)$ et $v_0(x, y, z) = v_0(x, y)$. La composante w_0 correspond quant à elle, au déplacement du plan moyen dans la direction z . Selon les hypothèses de Kirchhoff-Love, les déplacements verticaux sont les mêmes pour tous les points d'une fibre normale positionnée en (x, y) dans le plan $(0, x, y)$ du plan moyen. Nous savons ainsi que la composante w_0 ne dépend que de x et y avec $w_0(x, y, z) = w_0(x, y)$. Selon cette même hypothèse, nous savons également que pour tout point de la plaque $w(x, y, z) = w_0(x, y, z) = w_0(x, y)$.

Étudions désormais la rotation du plan moyen en notant par θ_x, θ_y ses rotations par rapport aux axes x et y . Ainsi une fibre normale au plan moyen positionnée en $(x, y)^T$ dans le plan $(0, x, y)$ tourne de l'angle θ_x autour de l'axe x et de l'angle θ_y autour de l'axe y .

Toutes ces notions et notations sont illustrées par la Fig. 2.4.

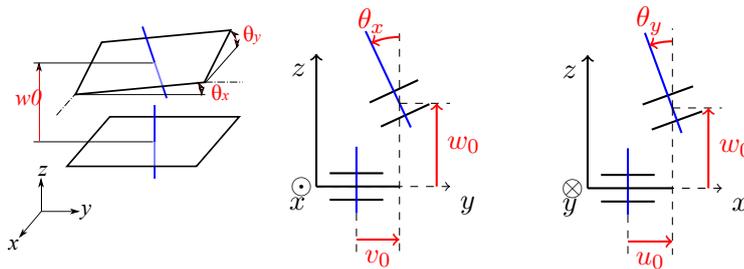


Figure 2.4 – Déplacement du plan moyen avec mise en évidence de la rotation d'une fibre normale autour des axes x et y . Figure extraite de [Wikipédia (2015a)].

Étant en petites déformations, l'arc de cercle décrit lors de la rotation de la fibre

2. Pour les notations, je donne dans un premier temps les notations employées dans le chapitre 1, puis je mets celles usuellement utilisées dans la littérature sur la théorie des plaques et des coques. Notez que le repère local du plan moyen est le repère (O, x, y, z) et que les coordonnées cartésiennes des points de la plaque dans l'état de référence sont également notées x, y, z .

normale en considérant un point situé sur celle-ci, est assimilable à un segment de droite. Ainsi en observant la Fig. 2.5, nous écrivons facilement les relations suivantes :

$$\tan \theta_x = -\frac{dv_0}{dz}, \quad \tan \theta_y = \frac{du_0}{dz}. \quad (2.1)$$

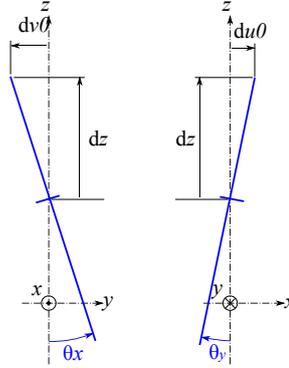


Figure 2.5 – Rotation d'une fibre normale selon les axes x et y mettant en évidence les relations entre u_0, v_0 et les angles θ_x, θ_y correspondant au mouvement du plan moyen [Wikipédia (2015a)].

Toujours sous l'hypothèse de petites déformations, les tangentes des angles θ_x et θ_y peuvent être assimilées à leurs angles. Les relations précédentes deviennent ainsi :

$$\theta_y = \frac{du_0}{dz}, \quad \theta_x = -\frac{dv_0}{dz} \Rightarrow du_0 = \theta_y dz, \quad dv_0 = -\theta_x dz.$$

Le mouvement induit par la rotation du plan moyen est ainsi exprimé par $z \theta_y(x, y)$ selon l'axe x et par $-z \theta_x(x, y)$ selon l'axe y pour le point (x, y, z) dans l'état de référence. La relation précédente permet également d'écrire :

$$\theta_y = \frac{\partial u_0}{\partial z}, \quad \theta_x = -\frac{\partial v_0}{\partial z}.$$

La Fig. 2.4 met également en évidence le fait que les tangentes des angles θ_x et θ_y représentent aussi la pente que prend le feuillet moyen. Ceci induit les relations suivantes :

$$\tan \theta_y = -\frac{\partial w_0}{\partial x}, \quad \tan \theta_x = \frac{\partial w_0}{\partial y} \Rightarrow \theta_y = -\frac{\partial w_0}{\partial x}, \quad \theta_x = \frac{\partial w_0}{\partial y}.$$

Au final, si nous considérons la translation et la rotation du plan moyen, le déplacement \vec{u} de tout point de la plaque de position $(x, y, z)^T$ dans l'état de référence, s'exprime sous la forme suivante :

$$\vec{u} = \begin{pmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{pmatrix} = \begin{pmatrix} u_0(x, y) \\ v_0(x, y) \\ w_0(x, y) \end{pmatrix} + z \begin{pmatrix} \theta_y(x, y) \\ -\theta_x(x, y) \\ 0 \end{pmatrix}.$$

Nous avons ainsi les relations suivantes pour les coordonnées du **déplacement** \vec{u} de la **plaque mince** :

$$\begin{cases} u(x, y, z) = u_0(x, y) + z \theta_y(x, y) \\ v(x, y, z) = v_0(x, y) - z \theta_x(x, y) \\ w(x, y, z) = w_0(x, y) \end{cases} \quad (2.2)$$

2.4 Déformation d'une plaque mince

Soit un point \mathbf{X} dans l'état de référence, correspondant à la plaque au repos, ayant pour position $(X_1, X_2, X_3)^T = (x, y, z)^T$. Suite à la déformation ϕ , nous avons vu que ce point subit un déplacement $\vec{u} = (u_1, u_2, u_3)^T = (u, v, w)^T$. Puis nous avons vu qu'étant dans le cadre de petites déformations avec de petits déplacements, nous pouvons considérer le **tenseur de déformation de Green-Lagrange linéarisé** pour modéliser cette déformation en fonction du déplacement. Dans le chapitre 1, nous avons vu que ce tenseur était symétrique et que ses composantes écrites avec la notation de Voigt sont définies par :

$$\epsilon^g = \begin{pmatrix} \epsilon_{xx}^g \\ \epsilon_{yy}^g \\ \epsilon_{zz}^g \\ \epsilon_{yz}^g \\ \epsilon_{xz}^g \\ \epsilon_{xy}^g \end{pmatrix} = \begin{pmatrix} \frac{\partial u_1}{\partial X_1} \\ \frac{\partial u_2}{\partial X_2} \\ \frac{\partial u_3}{\partial X_3} \\ \frac{1}{2} \left(\frac{\partial u_2}{\partial X_3} + \frac{\partial u_3}{\partial X_2} \right) \\ \frac{1}{2} \left(\frac{\partial u_1}{\partial X_3} + \frac{\partial u_3}{\partial X_1} \right) \\ \frac{1}{2} \left(\frac{\partial u_1}{\partial X_2} + \frac{\partial u_2}{\partial X_1} \right) \end{pmatrix} = \begin{pmatrix} \frac{\partial u(x,y,z)}{\partial x} \\ \frac{\partial v(x,y,z)}{\partial y} \\ \frac{\partial w(x,y,z)}{\partial z} \\ \frac{1}{2} \left(\frac{\partial v(x,y,z)}{\partial z} + \frac{\partial w(x,y,z)}{\partial y} \right) \\ \frac{1}{2} \left(\frac{\partial u(x,y,z)}{\partial z} + \frac{\partial w(x,y,z)}{\partial x} \right) \\ \frac{1}{2} \left(\frac{\partial u(x,y,z)}{\partial y} + \frac{\partial v(x,y,z)}{\partial x} \right) \end{pmatrix}.$$

En employant les relations relatives au déplacement \vec{u} établies dans l'équation (2.2), nous obtenons pour le point (x, y, z) de la plaque au repos :

$$\left\{ \begin{array}{l} \epsilon_{xx}^g = \frac{\partial u_0(x, y)}{\partial x} + z \frac{\partial \theta_y(x, y)}{\partial x} \\ \epsilon_{yy}^g = \frac{\partial v_0(x, y)}{\partial y} - z \frac{\partial \theta_x(x, y)}{\partial y} \\ \epsilon_{zz}^g = \frac{\partial w_0(x, y)}{\partial z} \\ \epsilon_{yz}^g = \frac{1}{2} \left(\frac{\partial v_0(x, y)}{\partial z} - \theta_x(x, y) + \frac{\partial w_0(x, y)}{\partial y} \right) \\ \epsilon_{xz}^g = \frac{1}{2} \left(\frac{\partial u_0(x, y)}{\partial z} + \theta_y(x, y) + \frac{\partial w_0(x, y)}{\partial x} \right) \\ \epsilon_{xy}^g = \frac{1}{2} \left[\frac{\partial u_0(x, y)}{\partial y} + \frac{\partial v_0(x, y)}{\partial x} + z \left(\frac{\partial \theta_y(x, y)}{\partial y} - \frac{\partial \theta_x(x, y)}{\partial x} \right) \right] \end{array} \right. \quad (2.3)$$

Puis en utilisant les relations mises en évidence selon les hypothèses de Kirchhoff-Love, nous pouvons simplifier certaines composantes de la façon suivante :

- le déplacement w_0 ne dépendant pas de z , nous avons $\epsilon_{zz}^g = 0$;
- le déplacement v_0 ne dépendant pas de z , nous avons $\epsilon_{yz}^g = \frac{1}{2} (-\theta_x(x, y) + \partial w_0(x, y)/\partial y)$ et avec $\partial w_0(x, y)/\partial y = \theta_x$, nous avons $\epsilon_{yz}^g = \frac{1}{2} (-\theta_x(x, y) + \theta_x(x, y)) = 0$;
- le déplacement u_0 ne dépendant pas de z , nous avons $\epsilon_{xz}^g = \frac{1}{2} (\theta_y(x, y) + \partial w_0(x, y)/\partial x)$ et avec $\partial w_0(x, y)/\partial x = -\theta_y$, nous avons $\epsilon_{xz}^g = \frac{1}{2} (\theta_y(x, y) - \theta_y(x, y)) = 0$.

Ce qui nous donne comme tenseur des déformation en notation de Voigt :

$$\epsilon^g = \begin{pmatrix} \epsilon_{xx}^g \\ \epsilon_{yy}^g \\ \epsilon_{zz}^g \\ \epsilon_{yz}^g \\ \epsilon_{xz}^g \\ \epsilon_{xy}^g \end{pmatrix} = \begin{pmatrix} \frac{\partial u_0(x, y)}{\partial x} + z \frac{\partial \theta_y(x, y)}{\partial x} \\ \frac{\partial v_0(x, y)}{\partial y} + z \left(-\frac{\partial \theta_x(x, y)}{\partial y} \right) \\ 0 \\ \frac{1}{2} \left(\frac{\partial w_0(x, y)}{\partial y} - \theta_x(x, y) \right) = 0 \\ \frac{1}{2} \left(\frac{\partial w_0(x, y)}{\partial x} + \theta_y(x, y) \right) = 0 \\ \frac{1}{2} \left(\frac{\partial u_0(x, y)}{\partial y} + \frac{\partial v_0(x, y)}{\partial x} \right) + z \frac{1}{2} \left(\frac{\partial \theta_y(x, y)}{\partial y} - \frac{\partial \theta_x(x, y)}{\partial x} \right) \end{pmatrix}. \quad (2.4)$$

Nous pouvons décomposer le tenseur global des déformations ϵ^g pour faire apparaître :

- le tenseur ϵ_m , modélisant les **déformations membranaires** du plan moyen issues des efforts de membrane qui se traduisent par une translation de la plaque dans le plan (x, y) avec

$$\epsilon_m = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{pmatrix} = \begin{pmatrix} \frac{\partial u_0(x, y)}{\partial x} \\ \frac{\partial v_0(x, y)}{\partial y} \\ \frac{1}{2} \left(\frac{\partial u_0(x, y)}{\partial y} + \frac{\partial v_0(x, y)}{\partial x} \right) \end{pmatrix},$$

- le tenseur ϵ_b , modélisant les **déformations de flexion** (ou variation de courbure) issues des efforts de flexion qui se traduisent par une rotation de la plaque autour des axes x, y avec

$$\epsilon_b = \begin{pmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \end{pmatrix} = \begin{pmatrix} \frac{\partial \theta_y(x, y)}{\partial x} \\ -\frac{\partial \theta_x(x, y)}{\partial y} \\ \frac{1}{2} \left(\frac{\partial \theta_y(x, y)}{\partial y} - \frac{\partial \theta_x(x, y)}{\partial x} \right) \end{pmatrix},$$

- le tenseur ϵ_c , modélisant les **déformations de cisaillement** issues des efforts de cisaillement qui sont considérés comme négligeables dans le cadre des plaques minces avec

$$\epsilon_c = \begin{pmatrix} \gamma_x \\ \gamma_y \end{pmatrix} = \begin{pmatrix} \frac{\partial w_0(x, y)}{\partial x} + \theta_y(x, y) \\ \frac{\partial w_0(x, y)}{\partial y} - \theta_x(x, y) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Nous avons ainsi la relation suivante entre ces tenseurs de déformation :

$$\epsilon^g = \begin{pmatrix} \epsilon_{xx}^g \\ \epsilon_{yy}^g \\ \epsilon_{zz}^g \\ \epsilon_{yz}^g \\ \epsilon_{xz}^g \\ \epsilon_{xy}^g \end{pmatrix} = \begin{pmatrix} e_{xx} \\ e_{yy} \\ 0 \\ 0 \\ 0 \\ e_{xy} \end{pmatrix} + z \begin{pmatrix} \kappa_{xx} \\ \kappa_{yy} \\ 0 \\ 0 \\ 0 \\ \kappa_{xy} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1/2 \gamma_y \\ 1/2 \gamma_x \\ 0 \end{pmatrix} = \begin{pmatrix} e_{xx} \\ e_{yy} \\ 0 \\ 0 \\ 0 \\ e_{xy} \end{pmatrix} + z \begin{pmatrix} \kappa_{xx} \\ \kappa_{yy} \\ 0 \\ 0 \\ 0 \\ \kappa_{xy} \end{pmatrix}.$$

Au final, le **tenseur de déformation pour la plaque mince** modélisant à la fois les déformations membranaires (translation dans le plan (x, y)) et de flexion (rotation autour des axes x et y) est défini de la façon suivante :

$$\epsilon^g = \begin{pmatrix} \epsilon_{xx}^g \\ \epsilon_{yy}^g \\ \epsilon_{xy}^g \end{pmatrix} = \begin{pmatrix} e_{xx} \\ e_{yy} \\ e_{xy} \end{pmatrix} + z \begin{pmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \end{pmatrix} = \epsilon_m + z \epsilon_b \quad (2.5)$$

donnant les relations suivantes pour ses composantes

$$\begin{cases} \epsilon_{xx}^g = \frac{\partial u_0(x, y)}{\partial x} + z \frac{\partial \theta_y(x, y)}{\partial x} \\ \epsilon_{yy}^g = \frac{\partial v_0(x, y)}{\partial y} - z \frac{\partial \theta_x(x, y)}{\partial y} \\ \epsilon_{xy}^g = \frac{1}{2} \left(\frac{\partial u_0(x, y)}{\partial y} + \frac{\partial v_0(x, y)}{\partial x} \right) + \frac{1}{2} z \left(\frac{\partial \theta_y(x, y)}{\partial y} - \frac{\partial \theta_x(x, y)}{\partial x} \right) \end{cases}$$

2.5 Tenseur de contraintes et loi de comportement

Étudions maintenant le tenseur de contraintes et son lien avec le tenseur de déformation définissant la loi de comportement de la plaque mince. Rappelons que la plaque de faible épaisseur est sollicitée uniquement dans son plan (x, y) et que la direction z est orthogonale à la plaque. Les composantes $\sigma_{xz}, \sigma_{yz}, \sigma_{zz}$ du tenseur de contraintes sont ainsi nulles sur les deux faces de la plaques et par raison de continuité sont négligeables à l'intérieur de la plaque. Pour la même raison, les composantes $\sigma_{xx}, \sigma_{xy}, \sigma_{yy}$ sont non nulles mais peu dépendantes de z . Nous supposons également qu'elles sont constantes sur toute l'épaisseur de la plaque.

En résumé, le **tenseur de contraintes** s'exprime ainsi pour la plaque mince :

$$\bar{\bar{\sigma}} = \begin{pmatrix} \sigma_{xx}(x, y) & \sigma_{xy}(x, y) & \sigma_{xz} = 0 \\ \sigma_{yx}(x, y) & \sigma_{yy}(x, y) & \sigma_{yz} = 0 \\ \sigma_{zx} = 0 & \sigma_{zy} = 0 & \sigma_{zz} = 0 \end{pmatrix}, \quad \bar{\bar{\sigma}} = \begin{pmatrix} \sigma_{xx}(x, y) \\ \sigma_{yy}(x, y) \\ \sigma_{xy}(x, y) \end{pmatrix}. \quad (2.6)$$

Si nous reprenons la formulation de la **loi de Hooke** en fonction de E et ν , la relation entre le tenseur de contraintes $\bar{\bar{\sigma}}$ et le tenseur de déformation $\bar{\bar{\epsilon}}$ s'écrit sous la forme

matricielle suivante :

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{pmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\nu \end{pmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{yz} \\ \epsilon_{xz} \\ \epsilon_{xy} \end{pmatrix}.$$

Nous obtenons ainsi comme équations (en considérant $\sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0$) :

$$\left\{ \begin{array}{l} \sigma_{xx} = \frac{E}{(1+\nu)(1-2\nu)} [(1-\nu) \epsilon_{xx} + \nu \epsilon_{yy} + \nu \epsilon_{zz}] \\ \sigma_{yy} = \frac{E}{(1+\nu)(1-2\nu)} [\nu \epsilon_{xx} + (1-\nu) \epsilon_{yy} + \nu \epsilon_{zz}] \\ \sigma_{zz} = 0 = \frac{E}{(1+\nu)(1-2\nu)} [\nu \epsilon_{xx} + \nu \epsilon_{yy} + (1-\nu) \epsilon_{zz}] \\ \sigma_{yz} = 0 = \frac{E}{1+\nu} \epsilon_{yz} \Rightarrow \epsilon_{yz} = 0 \text{ (en accord avec résultat précédent)} \\ \sigma_{xz} = 0 = \frac{E}{1+\nu} \epsilon_{xz} \Rightarrow \epsilon_{xz} = 0 \text{ (en accord avec résultat précédent)} \\ \sigma_{xy} = \frac{E}{1+\nu} \epsilon_{xy} = \frac{E}{1-\nu^2} (1-\nu) \epsilon_{xy} \end{array} \right. \quad (2.7)$$

— Regardons la relation issue de la composante σ_{zz} . Nous avons :

$$\begin{aligned} \sigma_{zz} &= 0 = \frac{E}{(1+\nu)(1-2\nu)} [\nu \epsilon_{xx} + \nu \epsilon_{yy} + (1-\nu) \epsilon_{zz}] \\ &\Rightarrow \epsilon_{zz} = -\frac{\nu}{1-\nu} (\epsilon_{xx} + \epsilon_{yy}). \end{aligned}$$

— Regardons la relation issue de la composante σ_{yy} . Nous avons :

$$\begin{aligned} \sigma_{yy} &= \frac{E}{(1+\nu)(1-2\nu)} [\nu \epsilon_{xx} + (1-\nu) \epsilon_{yy} + \nu \epsilon_{zz}] \\ &= \frac{E}{(1+\nu)(1-2\nu)} \left[\nu \epsilon_{xx} + (1-\nu) \epsilon_{yy} - \frac{\nu^2}{1-\nu} (\epsilon_{xx} + \epsilon_{yy}) \right] \\ &= \frac{E}{1-\nu^2} (\nu \epsilon_{xx} + \epsilon_{yy}). \end{aligned}$$

— Regardons la relation issue de la composante σ_{xx} . Nous avons :

$$\begin{aligned} \sigma_{xx} &= \frac{E}{(1+\nu)(1-2\nu)} [(1-\nu) \epsilon_{xx} + \nu \epsilon_{yy} + \nu \epsilon_{zz}] \\ &= \frac{E}{(1+\nu)(1-2\nu)} \left[(1-\nu) \epsilon_{xx} + \nu \epsilon_{yy} - \frac{\nu^2}{1-\nu} (\epsilon_{xx} + \epsilon_{yy}) \right] \\ &= \frac{E}{1-\nu^2} (\epsilon_{xx} + \nu \epsilon_{yy}). \end{aligned}$$

Au final, nous avons la relation suivante entre le tenseur $\bar{\sigma}$ (avec $\sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0$) et $\bar{\epsilon}$ (avec $\epsilon_{xz} = \epsilon_{yz} = 0$) en considérant la loi de Hooke pour la plaque mince :

$$\underbrace{\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix}}_{\bar{\sigma}} = \underbrace{\frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix}}_{\bar{C}} \underbrace{\begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{pmatrix}}_{\bar{\epsilon}} \Rightarrow \bar{\sigma} = \bar{C} : \bar{\epsilon}$$

En considérant le tenseur de déformation de Green-Lagrange linéarisé ϵ^g établi dans l'équation (2.5), nous obtenons comme **loi de comportement pour la plaque mince** :

$$\begin{aligned} \bar{\sigma} = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} &= \bar{C} : \epsilon^g = \bar{C} : (\epsilon_m + z \epsilon_b) \\ &= \left(\frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \right) : \left(\begin{pmatrix} e_{xx} \\ e_{yy} \\ e_{xy} \end{pmatrix} + z \begin{pmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \end{pmatrix} \right), \end{aligned} \quad (2.8)$$

donnant les relations suivantes pour les composantes non nulles du tenseur de contraintes :

$$\begin{cases} \sigma_{xx} = \frac{E}{1-\nu^2} (e_{xx} + z \kappa_{xx} + \nu e_{yy} + \nu z \kappa_{yy}) \\ \sigma_{yy} = \frac{E}{1-\nu^2} (\nu e_{xx} + \nu z \kappa_{xx} + e_{yy} + z \kappa_{yy}) \\ \sigma_{xy} = \frac{E}{1+\nu} (e_{xy} + z \kappa_{xy}) \end{cases} \quad (2.9)$$

Si nous regardons désormais la **relation inversée**, la formulation de la loi de Hooke s'exprime initialement de la manière suivante en fonction de E et ν

$$\begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{yz} \\ \epsilon_{xz} \\ \epsilon_{xy} \end{pmatrix} = \frac{1}{E} \begin{pmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1+\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1+\nu \end{pmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{pmatrix}$$

donnant les équations suivantes (en considérant $\sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0$) :

$$\begin{cases} \epsilon_{xx} = \frac{1}{E} [\sigma_{xx} - \nu(\sigma_{yy} + \sigma_{zz})] = \frac{1}{E} (\sigma_{xx} - \nu \sigma_{yy}) \\ \epsilon_{yy} = \frac{1}{E} [\sigma_{yy} - \nu(\sigma_{xx} + \sigma_{zz})] = \frac{1}{E} (\sigma_{yy} - \nu \sigma_{xx}) \\ \epsilon_{zz} = \frac{1}{E} [\sigma_{zz} - \nu(\sigma_{xx} + \sigma_{yy})] = \frac{-\nu}{E} (\sigma_{xx} + \sigma_{yy}) \\ \epsilon_{yz} = \frac{1+\nu}{E} \sigma_{yz} = 0 \text{ (en accord avec résultat précédent)} \\ \epsilon_{xz} = \frac{1+\nu}{E} \sigma_{xz} = 0 \text{ (en accord avec résultat précédent)} \\ \epsilon_{xy} = \frac{1+\nu}{E} \sigma_{xy} \end{cases}$$

— Regardons la relation issue de la composante ϵ_{zz} . Nous avons :

$$\begin{aligned}
 \epsilon_{zz} &= -\frac{\nu}{E}(\sigma_{xx} + \sigma_{yy}) \\
 &= -\frac{\nu}{E} \frac{E}{1-\nu^2} (\epsilon_{xx} + \nu \epsilon_{yy} + \nu \epsilon_{xx} + \epsilon_{yy}) \\
 &= -\frac{\nu}{1-\nu^2} (1+\nu)(\epsilon_{xx} + \epsilon_{yy}) \\
 &= -\frac{\nu}{1-\nu} (\epsilon_{xx} + \epsilon_{yy}) \text{ (en accord avec résultat précédent)}
 \end{aligned}$$

Au final, nous avons ainsi la relation suivante entre les tenseurs ϵ (avec $\epsilon_{xz} = \epsilon_{yz} = 0$) et σ (avec $\sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0$) en considérant la loi de Hooke pour la plaque mince :

$$\underbrace{\begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{pmatrix}}_{\bar{\epsilon}} = \frac{1}{E} \underbrace{\begin{pmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 1+\nu \end{pmatrix}}_{\bar{D}} \underbrace{\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix}}_{\bar{\sigma}}.$$

2.6 Énergie de déformation d'une plaque mince

Considérons au temps t , une plaque mince d'épaisseur h et de surface A_t pour son plan moyen. Si nous considérons la relation (2.5) définissant les composantes du tenseur de déformation ϵ^g (avec $\epsilon^g = \epsilon_m + z \epsilon_b$), l'énergie de déformation de la plaque mince est donnée par :

$$\begin{aligned}
 W &= \frac{1}{2} \int_{D_t} (\bar{\sigma} : \epsilon^g) dx dy dz \\
 &= \frac{1}{2} \int_{A_t} \int_{-\frac{h}{2}}^{\frac{h}{2}} (\bar{\sigma} : \epsilon^g) dx dy dz \\
 &= \frac{1}{2} \int_{A_t} \int_{-\frac{h}{2}}^{\frac{h}{2}} (\epsilon_{xx}^g \sigma_{xx} + \epsilon_{yy}^g \sigma_{yy} + \epsilon_{xy}^g \sigma_{xy}) dx dy dz \\
 &= \frac{1}{2} \int_{A_t} \int_{-\frac{h}{2}}^{\frac{h}{2}} ((e_{xx} + z \kappa_{xx}) \sigma_{xx} + (e_{yy} + z \kappa_{yy}) \sigma_{yy} + (e_{xy} + z \kappa_{xy}) \sigma_{xy}) dx dy dz \\
 &= \frac{1}{2} \int_{A_t} \left(e_{xx} \left[\int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_{xx} dz \right] + e_{yy} \left[\int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_{yy} dz \right] + e_{xy} \left[\int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_{xy} dz \right] \right. \\
 &\quad \left. + \kappa_{xx} \left[\int_{-\frac{h}{2}}^{\frac{h}{2}} z \sigma_{xx} dz \right] + \kappa_{yy} \left[\int_{-\frac{h}{2}}^{\frac{h}{2}} z \sigma_{yy} dz \right] + \kappa_{xy} \left[\int_{-\frac{h}{2}}^{\frac{h}{2}} z \sigma_{xy} dz \right] \right) dx dy \\
 &= \frac{1}{2} \int_{A_t} (e_{xx} N_{xx} + e_{yy} N_{yy} + e_{xy} N_{xy} + \kappa_{xx} M_{xx} + \kappa_{yy} M_{yy} + \kappa_{xy} M_{xy}) dx dy \\
 &= \frac{1}{2} \int_{A_t} (\epsilon_m^T \vec{N} + \epsilon_b^T \vec{M}) dx dy
 \end{aligned}$$

où $\vec{N} = (N_{xx}, N_{yy}, N_{xy})^T$ représente les efforts résultants de membrane (en N/m) et $\vec{M} = (M_{xx}, M_{yy}, M_{xy})^T$ les efforts résultants de flexion (en N). Ils sont définis par :

$$\begin{aligned}
 \vec{N} &= \int_{-\frac{h}{2}}^{\frac{h}{2}} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} dz \\
 &= \int_{-\frac{h}{2}}^{\frac{h}{2}} \left[\frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{pmatrix} + z \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \begin{pmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \end{pmatrix} \right] dz \\
 &= \frac{E h}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \epsilon_m.
 \end{aligned}$$

$$\begin{aligned}
 \vec{M} &= \int_{-\frac{h}{2}}^{\frac{h}{2}} z \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} dz \\
 &= \int_{-\frac{h}{2}}^{\frac{h}{2}} \left[z \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{pmatrix} + z^2 \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \begin{pmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \end{pmatrix} \right] dz \\
 &= \frac{E h^3}{12(1-\nu^2)} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \epsilon_b.
 \end{aligned}$$

Au final, l'énergie de déformation est définie par la somme de l'énergie de déformation membranaire et de l'énergie de déformation de flexion avec :

$$W = \frac{1}{2} \int_{A_t} \left(\epsilon_m^T \frac{E h}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \epsilon_m + \epsilon_b^T \frac{E h^3}{12(1-\nu^2)} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \epsilon_b \right) dx dy.$$

En utilisant le fait que pour la plaque mince, $\bar{\bar{\sigma}} = \bar{\bar{C}} : \epsilon^g$ avec $\epsilon^g = (\epsilon_m + z \epsilon_b)$, nous pouvons également définir l'énergie de déformation de la façon suivante :

$$\begin{aligned}
 W &= \frac{1}{2} \int_{D_t} (\bar{\bar{\sigma}} : \epsilon^g) dx dy dz = \frac{1}{2} \int_{D_t} ((\bar{\bar{C}} : \epsilon^g) : \epsilon^g) dx dy dz = \frac{1}{2} \int_{A_t} \int_{-\frac{h}{2}}^{\frac{h}{2}} \epsilon^{gT} C \epsilon^g dx dy dz \\
 &= \frac{1}{2} \int_{A_t} \int_{-\frac{h}{2}}^{\frac{h}{2}} (\epsilon_m + z \epsilon_b)^T C (\epsilon_m + z \epsilon_b) dx dy dz \\
 &= \frac{1}{2} \int_{A_t} \int_{-\frac{h}{2}}^{\frac{h}{2}} \left(\epsilon_m^T C \epsilon_m + \epsilon_m^T z C \epsilon_b + z \epsilon_b^T C \epsilon_m + z^2 \epsilon_b^T C \epsilon_b \right) dx dy dz \\
 &= \frac{1}{2} \int_{A_t} \left[\underbrace{\epsilon_m^T \int_{-\frac{h}{2}}^{\frac{h}{2}} (C \epsilon_m + z C \epsilon_b) dz}_{\vec{N}} + \epsilon_b^T \underbrace{\int_{-\frac{h}{2}}^{\frac{h}{2}} (z C \epsilon_m + z^2 C \epsilon_b) dz}_{\vec{M}} \right] dx dy.
 \end{aligned}$$

2.7 Équation du mouvement d'une plaque mince

Dans le chapitre 1, nous avons vu que l'équation du mouvement d'un objet déformable s'écrivait de la façon suivante (en considérant que $\overrightarrow{f_{ext}}(t) = (f_{extx}, f_{exty}, f_{extz})^T$ incorpore la force de la gravité \vec{g}) :

$$\rho \frac{d^2 \vec{u}}{dt^2} = \rho \overrightarrow{f_{ext}} + \overrightarrow{div} \overline{\sigma}. \quad (2.10)$$

Rappelons que pour la plaque mince, $\sigma_{xy} = \sigma_{yx}$ et $\sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0$, donnant :

$$\overrightarrow{div} \overline{\sigma} = \begin{pmatrix} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \\ \frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} \\ \frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} \end{pmatrix} = \begin{pmatrix} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} \\ \frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} \\ 0 \end{pmatrix}.$$

Au final, nous devons ainsi résoudre les équations suivantes pour définir le champ de déplacement $\vec{u} = (u(x, y, z), v(x, y, z), w(x, y, z))^T$ de la plaque mince à chaque instant t :

$$\begin{cases} \rho \frac{d^2}{dt^2} u(x, y, z) = \rho f_{extx} + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} \\ \rho \frac{d^2}{dt^2} v(x, y, z) = \rho f_{exty} + \frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} \\ \rho \frac{d^2}{dt^2} w(x, y, z) = \rho f_{extz} \end{cases}$$

En considérant l'équation (2.9) donnant la relation entre les composantes du tenseur de contraintes et celles du tenseur de déformation de la plaque mince, nous avons également :

$$\left\{ \begin{array}{l} \frac{\partial \sigma_{xx}}{\partial x} = \frac{E}{1-\nu^2} \frac{\partial}{\partial x} (\mathbf{e}_{xx} + z \kappa_{xx} + \nu \mathbf{e}_{yy} + z \nu \kappa_{yy}) \\ = \frac{E}{1-\nu^2} \left(\frac{\partial^2 u_0(x, y)}{\partial x^2} + z \frac{\partial^2 \theta_y(x, y)}{\partial x^2} + \nu \frac{\partial^2 v_0(x, y)}{\partial x \partial y} - z \nu \frac{\partial^2 \theta_x(x, y)}{\partial x \partial y} \right) \\ \frac{\partial \sigma_{xy}}{\partial y} = \frac{E}{1+\nu} \frac{\partial}{\partial y} (\mathbf{e}_{xy} + z \kappa_{xy}) \\ = \frac{E}{2(1+\nu)} \left(\frac{\partial^2 u_0(x, y)}{\partial y^2} + \frac{\partial^2 v_0(x, y)}{\partial x \partial y} + z \frac{\partial^2 \theta_y(x, y)}{\partial y^2} - z \frac{\partial^2 \theta_x(x, y)}{\partial x \partial y} \right) \\ \frac{\partial \sigma_{yx}}{\partial x} = \frac{E}{1+\nu} \frac{\partial}{\partial x} (\mathbf{e}_{xy} + z \kappa_{xy}) \\ = \frac{E}{2(1+\nu)} \left(\frac{\partial^2 u_0(x, y)}{\partial x \partial y} + \frac{\partial^2 v_0(x, y)}{\partial x^2} + z \frac{\partial^2 \theta_y(x, y)}{\partial x \partial y} - z \frac{\partial^2 \theta_x(x, y)}{\partial x^2} \right) \\ \frac{\partial \sigma_{yy}}{\partial y} = \frac{E}{1-\nu^2} \frac{\partial}{\partial y} (\nu \mathbf{e}_{xx} + z \nu \kappa_{xx} + \mathbf{e}_{yy} + z \kappa_{yy}) \\ = \frac{E}{1-\nu^2} \left(\nu \frac{\partial^2 u_0(x, y)}{\partial x \partial y} + z \nu \frac{\partial^2 \theta_y(x, y)}{\partial x \partial y} + \frac{\partial^2 v_0(x, y)}{\partial y^2} - z \frac{\partial^2 \theta_x(x, y)}{\partial y^2} \right) \end{array} \right.$$

Nous pouvons ensuite employer la MEF pour effectuer la résolution de ces équations.

2.8 Conclusion

J'ai rappelé dans ce chapitre comment était modélisée une plaque mince sous les hypothèses de petites déformations et petits déplacements. Nous verrons dans le chapitre 6 comment nous résolvons ensuite les équations décrivant le mouvement d'une plaque mince par la méthode des éléments finis. Nous verrons notamment les éléments finis usuellement employés, appelés CST (*Constant Strain Triangle*) ou DKT (*Discrete Kirchhoff Triangle*), pour ensuite introduire les éléments CST-DKT combinaison des deux précédents.

Bibliographie

T. De Soza. Elements de plaque : modélisations DKT, DST, DKTG et Q4g. oct 2013.

S. Drapier. Mécanique des structures et approximations numériques. Jan. 2016.

Wikipédia. Théorie des plaques — wikipédia, l'encyclopédie libre, 2015a. URL https://fr.wikipedia.org/wiki/Theorie_des_plaques.

Wikipédia. Théorie des poutres — wikipédia, l'encyclopédie libre, 2015b. URL https://fr.wikipedia.org/wiki/Theorie_des_poutres.

Chapitre 3

Modèle physique des masses-ressorts

Ce chapitre présente le modèle discret des masses-ressorts qui permet de simuler le mouvement d'un objet déformable. Ce modèle physique est basé sur la discrétisation des objets en particules qui sont connectées entre elles par des ressorts. La déformation des ressorts permet ainsi de reproduire la déformation de l'objet. Dans ce chapitre, nous ferons tout d'abord une description de ce modèle. Puis nous mettrons en évidence ses limites naturelles. Nous verrons notamment son instabilité en compression et le fait qu'un système masses-ressorts, basé sur une discrétisation cubique de l'objet, ne peut reproduire que le comportement d'un matériau ayant un coefficient de Poisson égal à $1/4$ limitant son utilisation.

Sommaire

3.1	Introduction	86
3.2	Présentation du système discret des masses-ressorts	87
3.3	Méthode d'intégration d'Euler implicite	90
3.4	Limitation naturelle du coefficient de Poisson	97
3.5	Instabilité naturelle en compression	100
3.6	Conclusion	103

3.1 Introduction

Le modèle physique des masses-ressorts a été initialement proposé par [Terzopoulos et al. \(1987\)](#) et continue à être usuellement employé en informatique graphique. Il permet de simuler le comportement des objets déformables avec une complexité en calculs moindre que la méthode des éléments finis classiquement employée pour résoudre les équations issues de la MMC. Ce modèle est basé sur la discrétisation de l'objet en particules (les masses) qui sont reliées ensemble par des ressorts décrivant ainsi la topologie de l'objet.

Ce chapitre présente la formulation initiale du modèle physique des masses-ressorts ainsi que ses limitations naturelles, afin de mieux le comprendre et ainsi mieux appréhender l'extension que nous avons proposée dans le cadre de la thèse de Karolina Golec qui fera l'objet du chapitre 7 de ce manuscrit.

Organisation de ce chapitre : Ce chapitre se poursuit de la façon suivante :

- Dans la section 3.2, je présente la formulation classique d'un système masses-ressorts. Nous verrons notamment la boucle de simulation qui lui est associée, c'est-à-dire l'enchaînement des calculs effectués à chaque pas de temps de la simulation dans le cadre de la dynamique newtonienne.
- Dans la section 3.3, je présente les calculs spécifiques nécessaires à l'emploi du schéma d'intégration numérique d'Euler implicite avec notamment la résolution d'un système linéaire par la méthode du Gradient Conjugué..
- Dans la section 3.4, je mets en évidence les limites naturelles de ce modèle en terme du choix du coefficient de Poisson du matériau simulé. Ceci apparaît dans le cadre d'une discrétisation en éléments cubiques de la représentation géométrique de l'objet.
- Dans la section 3.5, je montre également les limites du modèle en terme de stabilité notamment dans le cadre de la compression de l'objet simulé.

3.2 Présentation du système discret des masses-ressorts

Le modèle physique des masses-ressorts est basé sur une discrétisation de l'objet en un ensemble de particules connectées entre elles par des ressorts. La déformation d'un objet modélisé par un système masses-ressorts est ainsi issue des déformations au cours du temps des ressorts qui le constituent.

Définition mécanique d'un ressort. Tout d'abord, considérons la mécanique d'un ressort. Soit k sa constante de raideur et l sa longueur au repos. Suite à une déformation, la longueur courante du ressort est définie par $l' = l + \delta$ où δ représente son élongation.

Il est ensuite intéressant de savoir calculer l'énergie engendrée par la déformation de ce ressort. Pour une configuration proche de l'état de repos, celle-ci est définie par :

$$W_{ressort}(\delta) = \frac{1}{2} k (l' - l)^2 = \frac{1}{2} k \delta^2. \quad (3.1)$$

La constante de raideur du ressort est ainsi définie par :

$$k = \frac{\partial^2 W_{ressort}(\delta)}{\partial l'^2}. \quad (3.2)$$

La force élastique de ce ressort est définie par :

$$\vec{F}_{ressort}^e(\delta) = -\frac{\partial W_{ressort}(\delta)}{\partial l'} \vec{u} = -k (l' - l) \vec{u} = -k \delta \vec{u}. \quad (3.3)$$

avec \vec{u} le vecteur unitaire donnant la direction du ressort. Le signe négatif signifie que la force s'oppose à la déformation du ressort. Ainsi, si la déformation δ est négative (cas où le ressort est comprimé), la force tend à étirer le ressort ; tandis que si δ est positive (cas où le ressort est étiré), la force tend à compresser le ressort.

Si nous ajoutons un amortisseur pour considérer le cas d'un ressort amorti, celui-ci applique une force inverse à la vitesse qui est appliquée pour compresser le ressort. Cette force est définie par :

$$\vec{F}_{ressort}^v = -d v' \vec{u} \quad (3.4)$$

où v' représente la variation de vitesse des deux extrémités du ressort et d le coefficient d'amortissement.

Première constatation. La définition de la force élastique d'un ressort induit naturellement un comportement mécanique linéaire, puisque la déformation qui est modélisée par l'élongation du ressort, est proportionnelle à l'effort appliqué. Autrement dit, la courbe de la loi de comportement d'un ressort, représentant la déformation en fonction de l'effort appliqué, est une droite. Par ailleurs, nous pouvons observer que cette force élastique dépend uniquement des paramètres mécaniques de raideur et d'amortissement du ressort qui permettent ainsi de moduler l'importance de l'élongation et de sa vitesse de déformation. Ainsi pour pouvoir modéliser un comportement mécanique non-linéaire en utilisant un système masses-ressorts, il est nécessaire de modifier les forces élastiques des ressorts et/ou d'introduire d'autres forces au sein du système pour sortir de cette linéarité.

Modélisation du système masses-ressorts. Pour modéliser un objet par un système masses-ressorts, le plus simple est de partir du modèle géométrique de l'objet : les particules sont positionnées sur les sommets, et les ressorts sur les arêtes du modèle géométrique de l'objet. Nous devons ensuite définir les propriétés des particules et des ressorts.

Au niveau des particules, nous devons modéliser leurs états au cours du temps. Nous notons $x_i(t)$ et $x'_i(t)$ la position et la vitesse de la particule p_i qui sont définies au temps t . Nous notons m_i sa masse qui n'évolue pas au cours du temps si l'objet ne subit pas de modification topologique durant la simulation.

Au niveau des ressorts, nous modélisons leurs paramètres mécaniques. Pour cela, k_{ij} et d_{ij} représentent respectivement la constante de raideur et la constante d'amortissement du ressort connectant les particules p_i et p_j . Nous notons $\delta_{ij}(t)$ sa variation de longueur par rapport à sa longueur au repos. A noter que l'amortissement est généralement défini en fonction des masses des particules placées à ses extrémités avec $d_{ij} = 2\sqrt{k_{ij}(m_i + m_j)}$.

Étapes de calculs pour la particule p_i . Plaçons-nous dans le cadre de la dynamique newtonienne. La boucle de simulation permettant de simuler le comportement de l'objet au cours du temps est alors définie par l'enchaînement des calculs suivants qui sont effectués à chaque instant t pour chacune des particules p_i du système :

1. Calcul de la force $\vec{F}_{ij}(t)$ qui est exercée sur la particule p_i par le ressort reliant cette particule à la particule p_j . Cette force est définie par :

$$\vec{F}_{ij}(t) = \vec{F}_{ij}^e(t) + \vec{F}_{ij}^v(t)$$

avec

$$\begin{cases} \vec{F}_{ij}^e(t) &= -k_{ij} \delta_{ij}(t) \vec{u}_{ij}(t) = -k_{ij} (\|x_i(t) - x_j(t)\| - l_{ij}) \vec{u}_{ij}(t) \\ \vec{F}_{ij}^v(t) &= -d_{ij}(x'_j(t) - x'_i(t)) \end{cases}$$

où $\vec{u}_{ij}(t)$ représente le vecteur unitaire défini entre les particules p_i et p_j donnant ainsi la direction du ressort. Ce vecteur allant de p_i vers p_j est défini par :

$$\vec{u}_{ij}(t) = \frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|}.$$

A noter que la force d'amortissement du ressort peut être projetée sur le ressort donnant la formulation suivante :

$$\vec{F}_{ij}^v(t) = \left(-d_{ij}(x'_j(t) - x'_i(t)) \cdot \vec{u}_{ij}(t) \right) \vec{u}_{ij}(t).$$

2. Calcul de l'ensemble des forces appliquées sur la particule p_i en ajoutant les forces extérieures $\vec{F}_i^{ext}(t)$ (telles que la force de gravité ou des forces d'interactions) aux forces $\vec{F}_{ij}(t)$ des ressorts avec :

$$\vec{F}_i(t) = \sum_j \vec{F}_{ij}(t) + \vec{F}_i^{ext}(t).$$

Pour calculer les forces exercées sur p_i par les différents ressorts du système, nous devons donc considérer tous les voisins p_j de p_i avec $i \neq j$, c'est-à-dire toutes les particules p_j connectées à p_i . Le calcul de ces forces nécessite ainsi une structure

de données décrivant les connexions qu'il existe entre les particules¹, c'est-à-dire donnant pour chacun des ressorts du système, les informations relatives aux particules placées à ses extrémités, ainsi que les propriétés mécaniques de ces ressorts (raideurs, amortissement et longueur au repos). Par ailleurs, notons que ce calcul peut être réalisé soit en bouclant sur les particules, soit en bouclant sur les ressorts du système.

3. Calcul de l'accélération de la particule p_i selon la seconde loi de Newton avec :

$$\frac{\partial^2 x_i(t)}{\partial t^2} = \frac{\partial x'_i(t)}{\partial t} = x''_i(t) = \frac{F_i(t)}{m_i}.$$

4. Calcul des nouvelles vitesse $x'_i(t+h)$ et position $x_i(t+h)$ de la particule p_i en employant un schéma d'intégration numérique, h représentant le pas de temps. A noter qu'il existe de nombreux schémas d'intégration numérique, et son choix est important puisqu'il détermine le pas de temps que nous pourrions employer pour la simulation. Dans un premier temps, nous pouvons citer le schéma d'Euler semi-implicite qui définit les nouvelles vitesses et position de la particule p_i au pas de temps $t+h$ à moindre coût par :

$$\begin{cases} x'_i(t+h) &= x'_i(t) + h x''_i(t) \\ x_i(t+h) &= x_i(t) + h x'_i(t+h) \end{cases}$$

Ce schéma a le mérite d'être très simple d'utilisation. Par contre, pour assurer la stabilité du système, il requière un pas de temps très petit. C'est pourquoi, le schéma d'Euler implicite lui est souvent préféré étant pour sa part inconditionnellement stable ([Hauth and Etmuss \(2001\)](#)). Par contre l'emploi de ce schéma d'intégration nécessite la résolution d'un système défini en considérant l'ensemble des particules ([Baraff and Witkin \(1998\)](#)), alors que les calculs précédents effectués pour la particule p_i étaient indépendants de l'état des autres particules du système (rendant par exemple sa parallélisation simple).

Système différentiel associé au système de particules. Construisons désormais le système en considérant l'ensemble des particules. Pour cela, considérons le cas où l'objet est discrétisé en N particules dans l'espace 3D. Afin de stocker l'état des particules au cours de la simulation, nous devons définir un certain nombre de vecteurs de taille $3N$.

- Soit x le vecteur contenant les positions des particules : $x(t) = \{x_1(t), \dots, x_N(t)\}$.
- Soit x' le vecteur contenant les vitesses des particules : $x'(t) = \{x'_1(t), \dots, x'_N(t)\}$.
- Soit x'' le vecteur contenant les accélérations des particules : $x''(t) = \{x''_1(t), \dots, x''_N(t)\}$.
- Soit F le vecteur contenant les forces appliquées sur chacune des particules :

$$F(t) = \{F_1(t), \dots, F_N(t)\}.$$

Nous avons vu que ces forces correspondent à celles appliquées par les ressorts et aux forces extérieures (gravité, interactions).

1. Dans ce sens, nous avons proposé durant la thèse de Elsa Fléchon (voir chapitre 5), une structure de données fournissant ces informations, mais également facilitant leurs mises à jour afin de permettre une modification de la topologie de l'objet au cours de la simulation. Dans le cadre des masses-ressorts, il s'agit de modifier les connexions entre les particules en supprimant ou en ajoutant des ressorts dans le système.

— Soient m_1, \dots, m_N les masses des N particules contenues dans le système, nous pouvons définir une matrice diagonale M de taille $3N \times 3N$ telle que

$$M = \text{diag}(m'_1, \dots, m'_N) \text{ avec } m'_i = \begin{pmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{pmatrix}.$$

Le système mécanique peut alors être exprimé sous la forme d'un système d'Équations Différentielles Ordinaires (EDO) du second ordre accompagné de valeurs initiales. En considérant la seconde loi de Newton, nous avons :

$$\begin{cases} x''(t) &= M^{-1} f(t, x(t), x'(t)), \\ x'(t_0) &= x'_0, \\ x(t_0) &= x_0. \end{cases} \quad (3.5)$$

Ce système peut être facilement transformé en système du premier ordre, en introduisant une variable caractérisant les vitesses avec :

$$v(t) = x'(t).$$

Nous obtenons alors le système du premier ordre suivant :

$$\begin{pmatrix} x''(t) \\ x'(t) \end{pmatrix} = \frac{d}{dt} \begin{pmatrix} x'(t) \\ x(t) \end{pmatrix} = \begin{pmatrix} M^{-1} f(t, x(t), v(t)) \\ v(t) \end{pmatrix}, \begin{pmatrix} v(t_0) \\ x(t_0) \end{pmatrix} = \begin{pmatrix} v_0 \\ x_0 \end{pmatrix}. \quad (3.6)$$

Supposons que la position $x(t)$ et la vitesse $v(t)$ des particules soient connues au temps t . La résolution de ce système permet l'obtention des vitesses et positions des particules au pas de temps suivant soient respectivement $v(t+h)$ et $x(t+h)$.

3.3 Méthode d'intégration d'Euler implicite

Je présente ici l'utilisation du schéma d'intégration d'Euler implicite dans le cadre de l'emploi d'une modélisation basée sur le système discret des masses-ressorts.

Schéma d'intégration d'Euler implicite. Si nous considérons de manière générale le système d'Équations Différentielles Ordinaires $y'(t) = f(t, y(t)), y(t_0) = y_0$, le schéma d'intégration d'Euler implicite est défini par :

$$y(t+h) = y(t) + hf(t+h, y(t+h)). \quad (3.7)$$

Emploi dans le cadre d'un système de particules. Je reprends ensuite le raisonnement de [Baraff and Witkin \(1998\)](#) permettant de simplifier le système (3.6) lors de l'emploi du schéma d'intégration d'Euler implicite.

Définissons tout d'abord les variables Δv et Δx telles que :

$$\begin{cases} \Delta v &= v(t+h) - v(t), \\ \Delta x &= x(t+h) - x(t). \end{cases} \quad (3.8)$$

Si nous appliquons le schéma d'Euler implicite (3.7) au système (3.6), nous obtenons le système non-linéaire suivant :

$$\begin{aligned} \begin{pmatrix} \Delta v \\ \Delta x \end{pmatrix} &= \begin{pmatrix} h v'(t+h) \\ h v(t+h) \end{pmatrix} = h \begin{pmatrix} M^{-1} f(t, x(t+h), v(t+h)) \\ v(t) + \Delta v \end{pmatrix} \\ &= h \begin{pmatrix} M^{-1} f(x(t) + \Delta x, v(t) + \Delta v) \\ v(t) + \Delta v \end{pmatrix}. \end{aligned} \quad (3.9)$$

Appliquons ensuite les séries de Taylor à la fonction f . Nous obtenons l'approximation du premier degré suivante :

$$f(x(t) + \Delta x, v(t) + \Delta v) = f(t) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial v} \Delta v \quad (3.10)$$

dans laquelle les dérivées $\frac{\partial f}{\partial x}$ et $\frac{\partial f}{\partial v}$ devront être évaluées. En substituant cette approximation dans le système (3.9), nous obtenons le système linéaire suivant :

$$\begin{pmatrix} \Delta v \\ \Delta x \end{pmatrix} = h \begin{pmatrix} M^{-1} \left(f(t) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial v} \Delta v \right) \\ v(t) + \Delta v \end{pmatrix}. \quad (3.11)$$

Continuons à nous focaliser sur la première équation de ce système dans lequel nous introduisons la seconde équation en substituant Δx par $h(v(t) + \Delta v)$. Nous obtenons :

$$\Delta v = h M^{-1} \left(f(t) + \frac{\partial f}{\partial x} h(v(t) + \Delta v) + \frac{\partial f}{\partial v} \Delta v \right). \quad (3.12)$$

Nous en déduisons alors le système suivant avec I la matrice identité :

$$\left(I - h M^{-1} \frac{\partial f}{\partial v} - h^2 M^{-1} \frac{\partial f}{\partial x} \right) \Delta v = h M^{-1} \left(f(t) + h \frac{\partial f}{\partial x} v(t) \right), \quad (3.13)$$

qui peut également s'écrire sous la forme :

$$\underbrace{\left(M - h \frac{\partial f}{\partial v} - h^2 \frac{\partial f}{\partial x} \right)}_A \Delta v = h \underbrace{\left(f(t) + h \frac{\partial f}{\partial x} v(t) \right)}_b. \quad (3.14)$$

Il ne reste plus qu'à résoudre ce système qui est de la forme $A\Delta v = b$, afin d'obtenir Δv et ainsi calculer facilement :

$$\begin{cases} v(t+h) = v(t) + \Delta v, \\ x(t+h) = x(t) + h v(t+h). \end{cases} \quad (3.15)$$

Dans le cadre d'un système de particules, les matrices $\frac{\partial f}{\partial x}$ et $\frac{\partial f}{\partial v}$ représentent les variations des forces d'élasticité et de viscosité par rapport à la position et à la vitesse des particules. Ce sont alors des matrices creuses définies positives. C'est pourquoi, nous pouvons alors appliquer la méthode du gradient conjugué (Baraff and Witkin (1998)) pour cette résolution, permettant de tirer profit du fait que la matrice A est creuse.

Étapes de la méthode d'intégration d'Euler implicite. En résumé, l'utilisation du schéma d'Euler implicite nécessite les étapes suivantes :

1. Évaluation de $f(t)$.
2. Évaluation de $\frac{\partial f}{\partial x}$ et $\frac{\partial f}{\partial v}$.
3. Construction du système d'équations linéaires creux (3.14).
4. Résolution de ce système pour obtenir Δv .
5. Mise à jour de x et v .

Méthode du Gradient Conjugué. Nous pouvons ainsi utiliser la méthode itérative du gradient conjugué pour la résolution du système (6.28). Cette méthode a l'avantage de ne pas nécessiter l'assemblage explicite de la matrice creuse A réduisant ainsi son coût en mémoire. Pour la présenter en détails, je reprends ici les explications de Shewchuk (1994) que j'avais déjà reprises dans mon manuscrit de thèse (Zara (2003)). Cette méthode fait partie de la catégorie des méthodes itératives qui sont connues comme étant des méthodes de minimisation et nous allons voir qu'elle converge asymptotiquement vers une solution.

Problème de minimisation. Rappelons qu'une forme quadratique est une fonction scalaire de la forme

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c, \quad (3.16)$$

où A est une matrice, x et b des vecteurs et c une constante scalaire. Si A est symétrique définie-positive, $f(x)$ est alors minimisée par la solution de $Ax = b$. Jonathan Richard Shewchuk démontre cette idée à l'aide d'un exemple simple. Soient

$$A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -8 \end{bmatrix}, \quad c = 0. \quad (3.17)$$

La Fig. 3.1 illustre ce système $Ax = b$. Dans le cas général, la solution x correspond au point d'intersection des n hyperplans, chacun étant de dimension $n - 1$. Pour ce problème, la solution x du système $Ax = b$ correspond à l'intersection des deux droites définissant le système, soit $x = [2, -2]^T$.

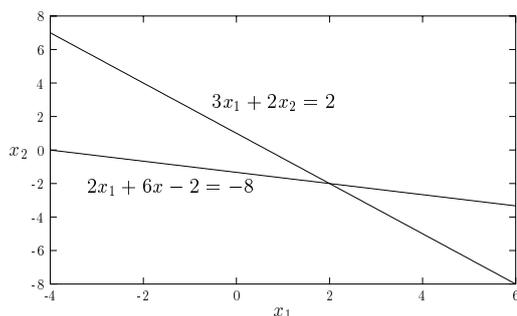


Figure 3.1 – Système linéaire en 2D ayant comme solution le point d'intersection des deux droites.

La Fig. 3.2 représente le graphe et les courbes de niveau de la forme quadratique $f(x)$ pour la matrice A , le vecteur b et le scalaire c donnés. Le point minimum de la surface définie par $f(x)$ est effectivement le point $(2, -2)$ solution du système linéaire.

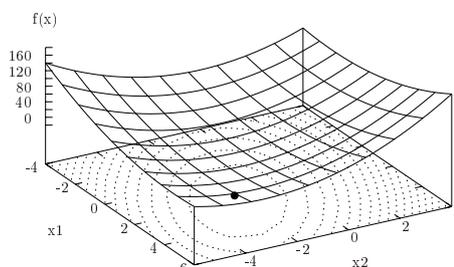


Figure 3.2 – Graphe et courbes de niveau de la forme quadratique $f(x)$: le point minimum de cette surface est la solution de $Ax = b$. Chaque courbe ellipsoïdale représente un $f(x)$ constant et le point correspond au $f(x)$ minimum.

La forme paraboloidale de la surface définie par $f(x)$ provient du fait que A est une matrice définie-positive. Le *gradient* de la forme quadratique est défini par

$$f'(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_0} \\ \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_{n-1}} \end{bmatrix}. \quad (3.18)$$

Le gradient est un champs de vecteurs qui, pour un point x donné, pointe dans la direction de la valeur la plus forte de $f(x)$. Le gradient est nul au bas de la boule paraboloidale signifiant que $f(x)$ est minimale quand $f'(x)$ est nulle. La dérivée de la forme quadratique (3.16) est définie par

$$f'(x) = \frac{1}{2}A^T x + \frac{1}{2}Ax - b. \quad (3.19)$$

Si A est symétrique, cette équation peut s'écrire sous la forme

$$f'(x) = Ax - b. \quad (3.20)$$

En définissant le gradient nul, nous retombons sur le système linéaire $Ax = b$ que nous cherchons à résoudre. La solution de $Ax = b$ est donc un point critique de $f(x)$, c'est-à-dire un x pour lequel $f'(x)$ est nulle. Par conséquent si A est une matrice symétrique définie-positive, cette solution est un minimum de $f(x)$, et ainsi $Ax = b$ peut être résolu en cherchant un x minimisant $f(x)$.

Méthode de la plus grande pente. Dans la méthode de la plus grande pente, nous commençons à un point arbitraire $x_{(0)}$ et ensuite nous descendons vers le bas de la parabolöide. Nous obtenons ainsi une série de pas $x_{(1)}, x_{(2)}, \dots$ qui s'arrête dès que nous considérons être assez proche de la solution x . Quand nous avançons d'un pas, nous choisissons la direction dans laquelle f décroît le plus rapidement, c'est-à-dire la direction opposée de celle désignée par $f'(x_{(i)})$, soit la direction $-f'(x_{(i)}) = b - Ax_{(i)}$ si nous reprenons l'équation (3.20).

Introduisons quelques termes utiles pour la compréhension de la méthode. L'*erreur* $e_{(i)} = x_{(i)} - x$ est un vecteur indiquant si nous sommes encore éloignés ou non de la solution. Le *résidu* $r_{(i)} = b - Ax_{(i)}$ indique quant à lui, la distance nous séparant de la valeur correcte b . La relation $r_{(i)} = -Ae_{(i)}$ apparaît alors entre ces deux entités. De plus le résidu peut être exprimé par $r_{(i)} = -f'(x_{(i)})$, c'est-à-dire que le résidu correspond à la direction de la plus grande pente.

Dans notre exemple, supposons que nous commençons au point $x_{(0)} = [-2, -2]^T$. Le premier pas est effectué le long de la direction de la pente la plus grande, c'est-à-dire en suivant une droite partant du point $x_{(0)}$ et de pente $-f'(x_{(0)})$. En d'autres termes, le point suivant est défini par

$$x_{(1)} = x_{(0)} + \alpha r_{(0)}. \quad (3.21)$$

Reste à choisir la taille du pas que nous effectuons à chaque itération. En fait α est choisi de façon à minimiser f le long de la droite et α minimise f quand la *dérivée directionnelle* $\frac{d}{d\alpha}f(x_{(1)})$ est égale à zéro. Nous obtenons alors :

$$\frac{d}{d\alpha}f(x_{(1)}) = f'(x_{(1)})^T \frac{d}{d\alpha}x_{(1)} = f'(x_{(1)})^T r_{(0)}.$$

Nous en déduisons que α est choisi en faisant en sorte que $r_{(0)}$ et $f'(x_{(1)})$ soient orthogonales. Après quelques calculs intermédiaires (Shewchuk (1994)), nous obtenons :

$$\alpha = \frac{r_{(0)}^T r_{(0)}}{r_{(0)}^T A r_{(0)}}. \quad (3.22)$$

En résumé, à chaque itération de la méthode de la plus grande pente, nous avons les relations suivantes

$$r_{(i)} = b - Ax_{(i)}, \quad (3.23)$$

$$\alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{r_{(i)}^T A r_{(i)}}, \quad (3.24)$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} r_{(i)}. \quad (3.25)$$

L'algorithme écrit de cette façon requière deux multiplications matrice-vecteur par itération, il est donc préférable de l'écrire de la façon suivante

$$r_{(i+1)} = r_{(i)} + \alpha_{(i)} A r_{(i)}, \quad (3.26)$$

permettant ainsi de supprimer une des deux multiplications matrice-vecteur fort coûteuses en terme de calcul. La Fig. 3.3 illustre la méthode avec l'exemple décrit précédemment. Nous obtenons un parcours en zigzag provenant du fait que chaque gradient est orthogonal au gradient précédent.

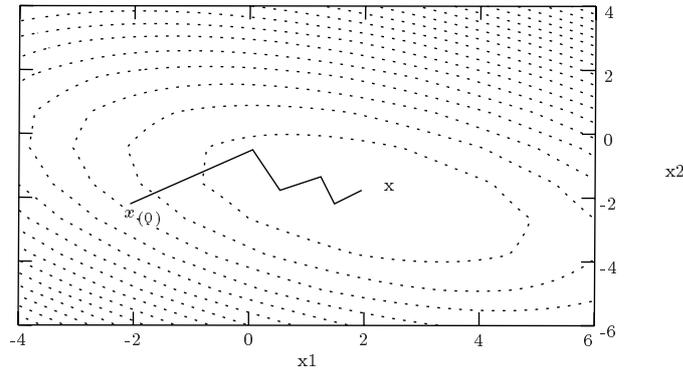


Figure 3.3 – Méthode de la plus grande pente qui débute à $x_{(0)} = [-2, -2]^T$ et converge pour $x = [2, -2]^T$.

Méthode des directions conjuguées. La méthode de la plus grande pente a tendance à ré-effectuer un pas dans une direction déjà prise auparavant, c'est-à-dire qu'elle n'a pas correctement évalué la distance à parcourir dans cette direction la première fois. Introduisons un ensemble de directions orthogonales $d_{(0)}, d_{(1)}, \dots, d_{(n-1)}$. Dans chacune de ces directions nous ne voulons effectuer exactement qu'un seul pas.

La Fig. 3.4 illustre cette idée. Le premier pas horizontal permet l'obtention de la coordonnée x_1 , et le second pas amène à la solution x . Nous pouvons noter que $e_{(1)}$ est orthogonale à la direction $d_{(0)}$. En définitive, à chaque pas nous choisissons le point

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)}. \quad (3.27)$$

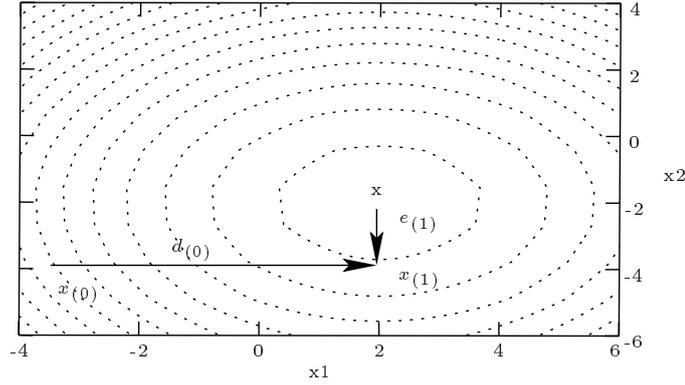


Figure 3.4 – Illustration de la méthode des directions conjuguées.

Pour obtenir la valeur de $\alpha_{(i)}$, nous utilisons le fait que $e_{(i+1)}$ doit être orthogonale à $d_{(i)}$ et que nous ne devons ensuite plus refaire de pas dans cette même direction $d_{(i)}$. Ces conditions peuvent se formuler par

$$d_{(i)}^T e_{(i+1)} = d_{(i)}^T (e_{(i)} + \alpha_{(i)} d_{(i)}) = 0,$$

permettant d'en déduire les valeurs $\alpha_{(i)}$ définies par $\alpha_{(i)} = -\frac{d_{(i)}^T e_{(i)}}{d_{(i)}^T d_{(i)}}$.

Pour le moment si nous souhaitons calculer $\alpha_{(i)}$, nous devons connaître $e_{(i)}$. Or le fait de connaître $e_{(i)}$ signifie que le problème a été résolu. La solution réside dans le fait de rechercher des directions *A-orthogonales* et non orthogonales : deux vecteurs $d_{(i)}$ et $d_{(j)}$ sont *A-orthogonaux* ou *conjugués*, si $d_{(i)}^T A d_{(j)} = 0$. Ainsi, nous souhaitons désormais que $e_{(i+1)}$ soit *A-orthogonal* à $d_{(i)}$. Cela revient à chercher le point minimum le long de la direction $d_{(i)}$ comme pour la méthode de la plus grande pente. Nous avons alors comme relation

$$\alpha_{(i)} = -\frac{d_{(i)}^T A e_{(i)}}{d_{(i)}^T A d_{(i)}} = \frac{d_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}}. \quad (3.28)$$

En remplaçant le vecteur de direction par le vecteur de résidu, nous retombons sur la relation observée pour la méthode de la plus grande pente. Il ne reste plus qu'à trouver une méthode permettant de générer l'ensemble des directions *A-orthogonales* $\{d_{(i)}\}$. Pour cela il existe une méthode nommée *procédé d'orthonormalisation de Gram-Schmidt*.

Soient un ensemble de n vecteurs linéairement indépendants u_0, u_1, \dots, u_{n-1} . Afin de construire $d_{(i)}$, nous prenons u_i et nous lui ôtons toutes les composantes *A-orthogonales* par rapport aux précédents vecteurs d . En d'autres termes, nous avons

$$d_{(0)} = u_0, \quad d_{(i)} = u_i + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)}, \quad i > 0. \quad (3.29)$$

où les β_{ik} sont définies pour $i > k$ par

$$\begin{aligned} d_{(i)}^T A d_{(j)} &= u_i^T A d_{(j)} + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)}^T A d_{(j)}, \\ \Rightarrow 0 &= u_i^T A d_{(j)} + \beta_{ij} d_{(j)}^T A d_{(j)}, \quad i > j \\ \Rightarrow \beta_{ij} &= -\frac{u_i^T A d_{(j)}}{d_{(j)}^T A d_{(j)}}. \end{aligned} \quad (3.30)$$

Le principal inconvénient de cette méthode réside dans le fait que nous devons garder en mémoire tous les vecteurs anciens afin d'en construire un nouveau, et de plus $\mathcal{O}(n^3)$ opérations sont nécessaires pour générer l'ensemble des vecteurs. Cette méthode devient équivalente à la méthode *d'élimination de Gauss*, si les vecteurs sont construits à partir des vecteurs unitaires.

Méthode du Gradient Conjugué. La méthode du Gradient Conjugué est comparable à la méthode des directions conjuguées, dans laquelle les vecteurs de directions sont construits à partir des vecteurs de résidu, c'est-à-dire en remplaçant u_i par $r_{(i)}$. Le fait que les vecteurs de résidu soient orthogonaux aux vecteurs de direction précédemment calculés, garantit le fait de construire une nouvelle direction linéairement indépendante à moins que le vecteur de résidu soit nul, et dans ce cas le problème serait déjà résolu. Les vecteurs de résidu étant orthogonaux aux vecteurs de direction précédemment calculés, ils sont également orthogonaux aux précédents vecteurs de résidu avec :

$$r_{(i)}^T r_{(j)} = 0, i \neq j. \quad (3.31)$$

Les constantes de Gram-Schmidt de l'équation (3.30) sont alors définies par

$$\beta_{ij} = -\frac{u_i^T A d_{(j)}}{d_{(j)}^T A d_{(j)}}.$$

En utilisant les vecteurs de résidu à la place des vecteurs de directions d , la majorité des termes β_{ij} disparaissent (Shewchuk (1994)), et ainsi il ne devient plus nécessaire de stocker tous les vecteurs précédents afin d'assurer la A -orthogonalité des nouveaux vecteurs construits. Par conséquent la complexité en temps et en mémoire de la méthode du Gradient Conjugué passe de $\mathcal{O}(n^2)$ à $\mathcal{O}(m)$ avec m le nombre d'éléments non nuls de la matrice A . La méthode des Gradients Conjugués s'exprime au travers des relations suivantes :

$$d_{(0)} = r_{(0)} = b - Ax_{(0)}, \quad (3.32)$$

$$\alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}}, \quad (3.33)$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)}, \quad (3.34)$$

$$r_{(i+1)} = r_{(i)} + \alpha_{(i)} A d_{(i)}, \quad (3.35)$$

$$\beta_{(i+1)} = \beta_{i+1,i} = \frac{r_{(i+1)}^T r_{(i+1)}}{r_{(i)}^T r_{(i)}}, \quad (3.36)$$

$$d_{(i+1)} = r_{(i+1)} + \beta_{(i+1)} d_{(i)}. \quad (3.37)$$

La performance de la méthode des Gradients Conjugués est illustrée par la Fig. 3.5.

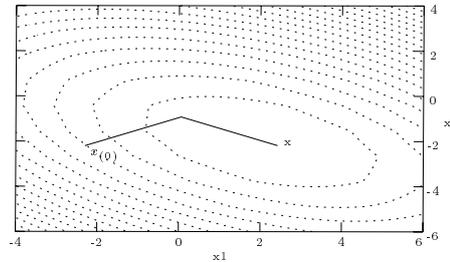


Figure 3.5 – Illustration de la méthode des Gradients Conjugués.

Au final, l'algorithme de la méthode du Gradient Conjugué itère jusqu'à ce que le facteur d'erreur prenne une valeur en dessous de ε qui traduit la précision souhaitée (Kumar et al. (1994); Saad (1996)). L'algorithme 1 présente l'algorithme séquentiel de cette méthode.

Algorithme 1 : Algorithme du Gradient Conjugué pour la résolution de $Ax = b$.

```

1 :  $\beta \leftarrow 0$ ; // Initialisation du facteur d'erreur
2 :  $x \leftarrow 0$ ; // Initialisation de la solution
3 :  $r \leftarrow b - Ax$ ; // Initialisation du vecteur de résidu

4 : Faire
5 :  $\alpha \leftarrow r^T r$ ; // Initialisation du pas
6 : Si ( $\beta \neq 0$ )
7 :  $d \leftarrow r + (\frac{\alpha}{\beta})d$ ; // Calcul de la nouvelle direction
8 : Sinon
9 :  $d \leftarrow r$ ; // Initialisation du vecteur de direction
10 :  $\beta \leftarrow d^T A d$ ; // Calcul du facteur d'erreur
11 :  $r \leftarrow r - (\frac{\alpha}{\beta})A d$ ; // Calcul du vecteur de résidu
12 :  $x \leftarrow x + (\frac{\alpha}{\beta})d$ ; // Calcul de la solution
13 :  $\beta \leftarrow \alpha$ ; // Nouveau facteur d'erreur
14 : Jusqu'à ( $\beta < \varepsilon$ ) // Itération jusqu'à la précision souhaitée
    
```

3.4 Limitation naturelle du coefficient de Poisson

Avant d'employer un système masses-ressorts pour reproduire le comportement d'un matériau déformable, il est important de bien comprendre ses limitations naturelles. Pour cela, considérons un matériel modélisé par un système masses-ressorts composé d'éléments cubiques. Nous notons V_0 et V respectivement son volume initial et volume courant.

Discretisation cubique du matériau. Considérons un matériau dont le modèle géométrique est basé sur une discrétisation cubique, c'est-à-dire que celui-ci est uniquement composé d'éléments cubiques. Pour créer le système masses-ressorts correspondant, nous positionnons les particules sur chacun des sommets de ce treillis. Chaque élément du treillis est ainsi constitué de 8 particules notées p_i avec $i \in [1; 8]$ positionnées aux sommets.

Il s'agit ensuite de positionner les ressorts entre ces particules. Baudet et al. (2009) et Zhao et al. (2011) ont suggéré de lier les particules d'un élément en plaçant tout d'abord des ressorts sur chacune des 12 arêtes de l'élément cubique, ainsi que 4 ressorts sur les diagonales intérieures à l'élément. Cette configuration est illustrée par la Fig. 3.6.

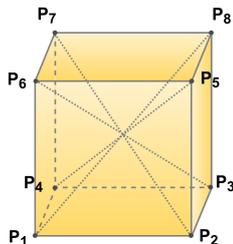


Figure 3.6 – Un élément cubique composé de 8 particules, 12 ressorts sur les arêtes et 4 ressorts sur les diagonales internes à l'élément.

Hypothèse de linéarité. Nous venons de voir que de part l'emploi de ressorts linéaires, le comportement mécanique d'un matériel modélisé par un système masses-ressorts est linéaire, c'est-à-dire qu'il suit la loi de Hooke avec une relation linéaire entre la contrainte σ appliquée sur le matériel et sa déformation ϵ . Nous avons vu dans le chapitre 1 que cette loi s'écrit sous la forme :

$$\sigma = \mathbf{C} : \epsilon ; \sigma_{ij} = C_{ijkl} \epsilon_{kl}, \text{ for } i, j, k, l = x, y, z, \quad (3.38)$$

où \mathbf{C} représente le tenseur élastique. Nous avons également vu dans ce chapitre 1 que dans le cadre de petites déformations, nous pouvons employer le tenseur linéarisé de Green-Lagrange qui est défini en fonction du déplacement \mathbf{u} du matériel avec $u_i = x_i - X_i$ pour le point i de la discrétisation (où x_i et X_i représentent respectivement la position courante et initiale de ce point). Le tenseur de Green-Lagrange est ainsi défini par :

$$\epsilon = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T), \quad \epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} \right).$$

Par ailleurs, le tenseur de contrainte σ représente la quantité associée à la déformation ϵ dans le sens où quand ϵ varie de $d\epsilon$, le changement d'énergie par unité de volume est définie par $d\widetilde{W} = \sigma_{ij} d\epsilon_{ij}$. L'expression du tenseur de contrainte est ainsi aussi définie par :

$$\sigma_{ij} = \frac{\partial \widetilde{W}}{\partial \epsilon_{ij}}.$$

La densité de l'énergie de déformation élastique que nous avons notée \widetilde{W} , est relative à une unité de volume du matériel déformé (ou non), la différence étant d'ordre supérieur en ϵ . L'existence de cette énergie implique les relations suivantes :

$$C_{ijkl} = \frac{\partial \sigma_{ij}}{\partial \epsilon_{kl}} = \frac{\partial \sigma_{kl}}{\partial \epsilon_{ij}} = \frac{\partial^2 \widetilde{W}}{\partial \epsilon_{ij} \partial \epsilon_{kl}}.$$

L'énergie par unité de volume est ainsi définie par la relation quadratique suivante où W représente l'énergie :

$$\widetilde{W} = \frac{W}{V_0} = \frac{1}{2} C_{ijkl} \epsilon_{ij} \epsilon_{kl}. \quad (3.39)$$

Rappelons que nous avons également vu dans le chapitre 1 que la symétrie des tenseurs de contraintes et de déformations ($\sigma_{ij} = \sigma_{ji}$ et $\epsilon_{ij} = \epsilon_{ji}$) permettait de réduire le nombre de coefficients nécessaires à la définition du tenseur \mathbf{C} . Celui-ci est initialement défini en 3D par $3^4 = 81$ coefficients. Ce nombre est ainsi réduit à $6^2 = 36$ coefficients avec $C_{ijkl} = C_{jikl} = C_{ijlk}$ et encore réduit à $6 + 6 \times 5/2 = 21$ coefficients avec $C_{ijkl} = C_{klij}$.

Symétrie d'un élément cubique. Maintenant considérons le fait que nous utilisons une discrétisation cubique de l'objet modélisé. Nous introduisons alors une symétrie d'un point de vue géométrique dans le système permettant de simplifier la formulation de l'énergie de déformation élastique. Le matériel décomposé en éléments cubiques est alors invariant selon une rotation de $\pi/4$ dans les différentes directions x, y, z et par réflexions.

- Par rapport aux symétries induites par rotations, nous pouvons en déduire que les coefficients C_{ijkl} sont les mêmes pour les indices se répétant dans le même ordre tel que $C_{xyxy} = C_{xzxz} = C_{yzyz}$.

- La propriété de réflexion ($x \rightarrow -x$, $y \rightarrow -y$, et $z \rightarrow -z$) permet de déduire des égalités de la forme $\epsilon_{xy} = -\epsilon_{xy}$ ou $\epsilon_{yz} = -\epsilon_{yz}$ impliquant le fait que nous ayons $C_{ijkl} = 0$ pour les composantes $\epsilon_{ij} \epsilon_{kl}$ de l'énergie élastique, où l'un des indices $\{x, y, z\}$ est répété un nombre impair de fois.

Ainsi en considérant cette symétrie cubique, la densité d'énergie initialement définie par l'équation (3.39) peut être reformulée en :

$$\begin{aligned} \widetilde{W}_{cube} = \frac{W_{cube}}{V_0} = & \frac{1}{2} C_{xxxx} (\epsilon_{xx}^2 + \epsilon_{yy}^2 + \epsilon_{zz}^2) + C_{xxyy} (\epsilon_{xx} \epsilon_{yy} + \epsilon_{xx} \epsilon_{zz} + \epsilon_{yy} \epsilon_{zz}) \\ & + 2 C_{xyxy} (\epsilon_{xy}^2 + \epsilon_{xz}^2 + \epsilon_{yz}^2). \end{aligned} \quad (3.40)$$

Hypothèse de l'isotropie. Maintenant considérons le fait que le matériau simulé est isotrope, c'est-à-dire que son comportement mécanique est identique dans les différentes directions de l'espace. Dans le chapitre 1, nous avons vu que le module d'élasticité \mathbf{C} peut alors être réduit à 2 paramètres indépendants. Nous pouvons ainsi considéré soit les deux paramètres de Lamé (notés λ et μ), soit le module de Young et le coefficient de Poisson (notés E et ν) définis par rapport au matériau considéré, sachant que ces paramètres peuvent être exprimés les uns par rapport aux autres avec :

$$E = \frac{\mu (3\lambda + 2\mu)}{\lambda + \mu}, \quad \nu = \frac{\lambda}{2(\lambda + \mu)}. \quad (3.41)$$

Si nous considérons les coefficients de Lamé, la densité d'énergie de l'équation (3.39) peut alors être reformulée de la façon suivante :

$$\begin{aligned} \widetilde{W}_{iso} &= \frac{W_{iso}}{V_0} = \left(\frac{\lambda}{2} + \mu\right) (\epsilon_{xx}^2 + \epsilon_{yy}^2 + \epsilon_{zz}^2) \\ &+ \lambda (\epsilon_{xx} \epsilon_{yy} + \epsilon_{xx} \epsilon_{zz} + \epsilon_{yy} \epsilon_{zz}) \\ &+ 2 \mu (\epsilon_{xy}^2 + \epsilon_{xz}^2 + \epsilon_{yz}^2) = \frac{\lambda}{2} (Tr(\epsilon))^2 + \mu Tr(\epsilon^2). \end{aligned} \quad (3.42)$$

Nous obtenons ainsi la définition suivante pour le tenseur de contrainte dans le cas d'un matériau isotrope

$$\sigma_{ij_{iso}} = \frac{\partial \widetilde{W}_{iso}}{\partial \epsilon_{ij}} = \lambda \delta_{ij} \epsilon_{kk} + 2 \mu \epsilon_{ij}, \quad (3.43)$$

avec comme module d'élasticité isotrope

$$C_{ijkl_{iso}} = \frac{\partial^2 \widetilde{W}_{iso}}{\partial \epsilon_{ij} \partial \epsilon_{kl}} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}). \quad (3.44)$$

Système masses-ressorts cubique pour un matériau isotrope. Au final, si nous souhaitons simuler le comportement d'un matériau élastique isotrope en employant un système masses-ressorts qui est basé sur une discrétisation cubique, il suffit de comparer l'équation (3.40) de la densité d'énergie obtenue en considérant une discrétisation cubique du matériau avec l'équation (3.43) obtenue pour un matériau isotrope. Autrement dit pour ajouter l'hypothèse d'isotropie à un système masses-ressorts basé sur une discrétisation cubique, il suffit de satisfaire les relations suivantes :

$$C_{xxxx} = C_{xxyy} + 2 C_{xyxy}, \quad C_{xxyy} = \lambda, \quad C_{xyxy} = \mu. \quad (3.45)$$

Nous pouvons également prendre en compte les relations de Cauchy statuant que le tenseur élastique est symétrique quand nous inter-changeons les indices $\{i, j, k, l\}$ avec

$$C_{ijkl} = C_{kjli}.$$

Cette propriété permet de réduire notre équation en considérant ainsi la relation

$$C_{xxyy} = C_{xyxy}$$

impliquant l'égalité de Cauchy $\lambda = \mu$.² Ainsi, si nous reprenons les relations établies entre λ, μ et E, ν , ce résultat entraîne un coefficient de Poisson constant fixé à $\nu = 1/4$.

Autrement dit, un système masses-ressorts basé sur une discrétisation cubique de l'objet considéré ne peut simuler correctement que le comportement de matériaux élastiques isotropes ayant un coefficient de Poisson égal à $1/4$ limitant largement son usage.

Par ailleurs, si nous considérons la relation (3.41) entre le module de Young et les coefficients de Lamé, nous obtenons pour $\nu = 1/4$, $\lambda = \mu$ et $E = 5/2 \mu = 5/2 \lambda$.

Dans le domaine de l'informatique graphique, nous considérons souvent le couple (E, ν) pour caractériser un objet élastique, mais il est bien entendu possible de transposer ces résultats en considérant d'autres couples de paramètres mécaniques grâce aux relations connues entre eux. Nous pouvons ainsi considérer le module d'élasticité isostatique K (ou *Bulk modulus*) et le module de rigidité G (ou *shear modulus*) qui sont définis pour un matériau élastique isotrope par rapport au couple (E, ν) par

$$K = \frac{E}{3(1-2\nu)}, \quad G = \frac{E}{2(1+\nu)}$$

avec comme relations inverses

$$E = \frac{9KG}{3K+G}, \quad \nu = \frac{3K-2G}{2(3K+G)}$$

donnant pour un coefficient de Poisson égal à $1/4$, $K = 2/3 E$ et $G = 2/5 E$, soit la relation finale suivante :

$$\lambda = \mu = G. \tag{3.46}$$

3.5 Instabilité naturelle en compression

Nous allons maintenant nous intéresser à la limite de stabilité d'un système masses-ressorts. De la même manière que précédemment, considérons un matériel modélisé par un système masses-ressorts cubiques où les ressorts ont été positionnés sur les arêtes de la représentation géométrique, ainsi que sur les 4 diagonales internes de chaque élément cubique. Dans cette configuration, des instabilités apparaissent sur les bords de l'objet (comme illustré sur la Fig. 3.7) à partir d'un certain taux de compression du matériau. Cette instabilité altère le comportement global du système en le rendant moins rigide.

2. Ce résultat peut être obtenu en insérant la relation de Cauchy $C_{ijkl} = C_{kjli}$ dans l'équation (3.43).

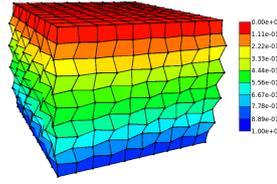


Figure 3.7 – Illustration des effets de l'instabilité sur un système masses-ressorts cubiques.

Explication de cette instabilité. L'origine de cette instabilité peut être expliquée en considérant un modèle composé de 5 particules (notées A, B, C, D, E) connectées par 4 ressorts qui sont définis par une constante de raideur k et une longueur au repos l . Ce système est illustré par la Fig. 3.8. Les particules D et E sont fixées. La particule C est libre de bouger dans la direction horizontale avec une distance notée x depuis sa position initiale. Les particules A et B sont libres de bouger dans la direction verticale depuis leurs positions initiales (illustrées en vert). Après avoir subies une déformation identique, leurs longueurs sont notées ζl . Ainsi la longueur des ressorts AC et BC est désormais définie par $l'(x, \zeta) = \sqrt{(\zeta l)^2 + x^2}$ et l'état d'équilibre de ce système, c'est-à-dire l'état où les ressorts sont de longueurs l , correspond à $x = 0$ et $\zeta = 1$.

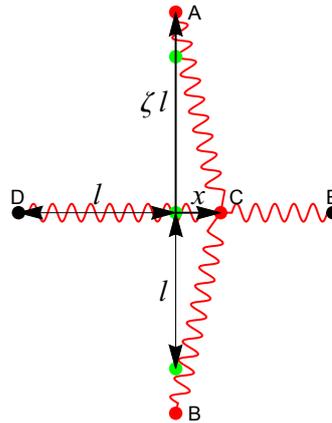


Figure 3.8 – Un système masses-ressorts composé de 5 particules (A, B, C, D, E) connectées par 4 ressorts de raideur k et de longueur l . Les points verts sont les positions d'équilibre des particules A, B et C .

Comme la formulation générale de l'énergie impliquée par la déformation d'un ressort de raideur k et d'élongation x est définie par $(k x^2)/2$, l'énergie $W(x, \zeta)$ induite lors de la déformation des ressorts AC, CB, DC, CE est définie par :

$$W(x, \zeta) = k \left((l'(x, \zeta) - l)^2 + x^2 \right). \quad (3.47)$$

A partir de cette énergie, nous pouvons calculer la force horizontale $F_B(x, \zeta)$ appliquée sur le point B comme le changement d'énergie quand un déplacement x est effectué avec :

$$F_B(x, \zeta) = -\frac{\partial W(x, \zeta)}{\partial x} = 2 k x \left(\frac{l}{l'(x, \zeta)} - 2 \right). \quad (3.48)$$

L'équation à l'équilibre $F_B(x, \zeta) = 0$ dispose ainsi de 3 racines : $x = 0$ et $x = \pm l \sqrt{\frac{1}{4} - \zeta^2}$ correspondant à $l/l'(x, \zeta) - 2 = 0$.

Nous pouvons par ailleurs noter que le changement de signe de ζ revient à inter-changer les positions des points A et B , c'est-à-dire que $l'(x, \zeta)$, $W(x, \zeta)$, $F_B(x, \zeta)$ ne change pas.

Observons plus en détails la force horizontale $F_B(x, \zeta)$:

- Pour $|\zeta| > 1/2$, c'est-à-dire quand $l/l'(x, \zeta) - 2 < 0$, les deux dernières racines sont imaginaires alors que la racine physique $x = 0$ correspond à une position stable. Dans cette configuration, le signe de $F_B(x, \zeta)$ est opposé à celui de x .
- Pour $|\zeta| < 1/2$, c'est-à-dire quand $l/l'(x, \zeta) - 2 > 0$, les deux dernières racines sont réelles et stables, et la dérivée $\partial F_B(x, \zeta)/\partial x$ est positive. Dans le même temps, $x = 0$ est une racine instable et $F_B(x, \zeta)$ a le même signe que x .

Ainsi, en tenant compte de ces différents cas, nous savons que les positions stables du point B correspondent aux cas où x est défini de la façon suivante :

$$x = \begin{cases} 0 & \text{if } |\zeta| > 1/2 \\ \pm l \sqrt{\frac{1}{4} - \zeta^2} & \text{if } |\zeta| < 1/2 \end{cases} \quad (3.49)$$

Le système subit ainsi une bifurcation quand $\zeta = \pm 1/2$. La configuration dans laquelle les particules A , B et C sont alignées, c'est-à-dire quand $x = 0$, devient ainsi instable dès que $|\zeta| < 1/2$. Nous obtenons alors une configuration instable avec $x \neq 0$.

Si nous insérons la solution de l'équation (3.49) dans la définition de $l'(x, \zeta)$, nous trouvons $l'(x, \zeta) = l/2$ correspondant à la zone instable $|\zeta| < 1/2$, tandis que $l'(x, \zeta) = l|\zeta|$ quand $|\zeta| > 1/2$.

Maintenant, supposons que la particule C se trouve dans une position stable définie par l'équation (3.49) et que l'énergie élastique résultante est fonction de ζ avec :

$$\begin{aligned} W_{\text{eq}}(\zeta) &= \begin{cases} W(x = 0, \zeta) & \text{if } |\zeta| > 1/2, \\ W(x = \pm l \sqrt{\frac{1}{4} - \zeta^2}, \zeta) & \text{if } |\zeta| < 1/2 \end{cases} \\ &= \begin{cases} k l^2 (|\zeta| - 1)^2 & \text{if } |\zeta| > 1/2, \\ k l^2 \left(\frac{1}{2} - \zeta^2\right) & \text{if } |\zeta| < 1/2 \end{cases} \end{aligned}$$

La composante verticale de la force que les ressorts exercent sur les particules A et B est donnée par :

$$F_A(\zeta) = -F_B(\zeta) = -\frac{1}{2l} \frac{\partial W_{\text{eq}}(\zeta)}{\partial \zeta} = \begin{cases} k l \left(\frac{|\zeta|}{\zeta} - \zeta\right) & \text{if } |\zeta| > 1/2 \\ k l \zeta & \text{if } |\zeta| < 1/2 \end{cases}$$

Sur la Fig. 3.9, la forme fonctionnelle de $F_A(\zeta)/kl$ présente des pics à $\zeta = \pm 1/2$. Considérons l'effet d'une force de compression d'une grandeur croissante telle qu'illustrée par les flèches en pointillés. Le système va répondre dans un premier temps avec une compression $\zeta = 1 - F_A(\zeta)/kl$ jusqu'à ce que la force de flambage critique $F_A(\zeta) = kl/2$ soit atteinte à $\zeta = 1/2$. Le système se déformera alors et les particules passeront à une configuration inversée pour laquelle la particule A se trouvera en dessous de la particule B avec $\zeta = -3/2$ se déformant selon $\zeta = -1 - F_A(\zeta)/kl$. Sur le chemin de retour de ce comportement de type hystérésis, le flambage débutera pour $\zeta = -1/2$ avec $F_A(\zeta) = -kl/2$ où le système sautera alors vers la configuration $\zeta = 1/2$.

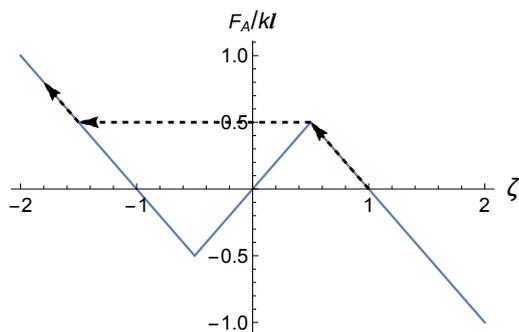


Figure 3.9 – Courbe en bleu : force verticale $F_A(\zeta)$ normalisée. Flèches en pointillés : trajectoire représentant une force de compression de magnitude croissante. Le flambage apparaît pour $-1/2 < \zeta < 1/2$.

La zone de flambage et la pente de la force dépend de la définition du système masses-ressorts utilisé. Pour éviter cela, nous pouvons rajouter des connexions entre les particules (par exemple dans notre cas un ressort entre les particules A et B) ou en faisant en sorte que les ressorts horizontaux diffèrent des ressorts verticaux. Mais malgré cela, il faut savoir qu'un système masses-ressorts en zone de flambage présente non seulement des instabilités visibles mais en plus il reproduit un comportement mécanique moins rigide modifiant ainsi la déformation globale du système. Il est donc important de réussir à faire en sorte que cet effet arrive pour un taux de compression le plus grand possible.

3.6 Conclusion

J'ai présenté dans ce chapitre, le modèle physique des masses-ressorts usuellement employé en animation. Ce modèle est basé sur la discrétisation de l'objets en particules (ou masses) qui sont reliées entre elles par des ressorts (caractérisés par leurs propriétés mécaniques). Ensuite, à chaque pas de temps de la simulation, il s'agit de calculer les forces qui sont appliquées sur les particules par les ressorts. Les calculs s'enchainent ensuite de manière usuelle avec le calcul des accélérations ainsi que leurs intégrations, pour obtenir le nouvel état des particules à chaque instant. Par ailleurs, nous avons vu qu'une discrétisation en éléments cubiques du matériau engendre des limites lors de l'emploi de ce modèle physique. Ainsi, il n'est possible de simuler que le comportement de matériau ayant un coefficient de Poisson égal à $1/4$. Par ailleurs, ce modèle possède une limite d'instabilité lorsque l'objet simulé subit une compression. Nous verrons ainsi dans le chapitre 7 comment nous avons dépassé ces limitations afin de permettre l'emploi de ce modèle dans le cadre de la simulation de tissus dont le comportement mécanique est non-linéaire.

Bibliographie

- D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98*, pages 43–54, New York, NY, USA, 1998. ACM. ISBN 0-89791-999-8. doi : 10.1145/280814.280821. URL <http://doi.acm.org/10.1145/280814.280821>.
- V. Baudet, M. Beuve, F. Jaillet, B. Shariat, and F. Zara. Integrating Tensile Parameters in Hexahedral Mass-Spring System for Simulation. In *International Conference on Computer Graphics, Visualization and Computer Vision'2009 - WSCG'2009*, Feb. 2009. URL <http://liris.cnrs.fr/publis/?id=3673>.
- M. Hauth and O. Eitzmuss. A High Performance Solver for the Animation of Deformable Objects using Advanced Numerical Methods. *Computer Graphics Forum*, 2001. ISSN 1467-8659. doi : 10.1111/1467-8659.00524.
- V. Kumar, A. Grama, A. Gupta, and G. Karypis. *Introduction to parallel computing, design and analysis of algorithms*. Benjamin/Cummings, 1994.
- Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.
- J. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-TR-94-125, Carnegie Mellon University, 1994.
- D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21(4) :205–214, Aug. 1987. ISSN 0097-8930.
- F. Zara. *Algorithmes parallèles de simulation physique pour la synthèse d'images : application à l'animation de textiles*. Informatique : Systèmes et logiciels, Institut National Polytechnique de Grenoble, France, December 2003.
- G.-F. Zhao, J. Fang, and J. Zhao. A 3d distinct lattice spring model for elasticity and dynamic failure. *International Journal for Numerical and Analytical Methods in Geomechanics*, 35(8) :859–885, 2011. ISSN 1096-9853. doi : 10.1002/nag.930.

Partie II

Contributions

Cette seconde partie présente les contributions de mes travaux de recherche qui ont été effectués au sein du LIRIS depuis mon arrivée en septembre 2005 en tant que maître de conférences du département d'informatique de la Faculté des Sciences de l'Université Lyon 1. Elle se décompose en 6 chapitres.

- Le chapitre 4 présente l'apport du calcul formel dans la simulation physique. Il concerne les résultats issus du travail de doctorat de Xavier Faure.
- Le chapitre 5 présente l'apport d'un modèle topologique pour la simulation d'objets déformables. Ce travail a été initié durant le doctorat de Elsa Fléchon et s'est poursuivi durant le doctorat de Karolina Golec (en collaboration avec Stéphane Nicolle du LBMC et Jean-François Palierne du Laboratoire de Physique de l'ENS Lyon) avec une collaboration active avec Guillaume Damiand.
- Le chapitre 6 présente un nouveau modèle physique surfacique couplé à une méthode de préservation de volume permettant une simulation efficace d'objets déformables. Ce travail est issu du doctorat de Mathieu Bailet supervisé avec Emmanuel Promayon (laboratoire TIMC-IMAG, Université de Grenoble).
- Le chapitre 7 présente l'extension du modèle physique discret des masses-ressorts pour la simulation d'objets déformables. Ce travail est la suite du travail de doctorat de Vincent Baudet (effectué au LIRIS et supervisé par Behzad Shariat et Fabrice Jaillet). Il a été effectué durant le doctorat de Karolina Golec.
- Le chapitre 8 présente les travaux autour de la réalisation d'un simulateur pour l'apprentissage des gestes de l'accouchement. Ces travaux ont été initiés durant le doctorat de Romain Buttin effectué en collaboration avec Tanneguy Redarce (laboratoire Ampère, INSA Lyon). Ils ont continué durant le doctorat de Mathieu Bailet et se sont poursuivis au sein du projet ANR SAGA.
- Le chapitre 9 dresse un bilan des contributions apportées dans le domaine de la simulation par modèles physiques et ouvre les perspectives pour la suite.

Rappelons que l'idée sous-jacente à l'ensemble de mes travaux est de pouvoir faire évoluer la définition de l'objet tout au long de l'exécution de la simulation numérique pour jouer sur le compromis qu'il y a entre le temps de calcul et la précision des résultats. Soulignons que pour mettre en pratique cette idée, il faut obtenir des formulations définies au niveau des éléments de l'objet qui permettent ce changement, en tenant compte aussi bien de la topologie que du comportement mécanique de l'objet.

Au fil des années, nous sommes arrivés à ce résultat avec des changements qui peuvent s'effectuer au niveau du modèle physique employé (avec le changement du modèle lui-même ou de la loi de comportement), au niveau de la topologie de l'objet ou encore de sa géométrie (pour augmenter ou diminuer le nombre d'éléments décrivant l'objet ou modifier les connections entre ces éléments).

Chapitre 4

Apport du calcul formel pour la simulation physique

Je présente dans ce chapitre les travaux qui ont été réalisés dans le cadre de la thèse de Xavier Faure (2014b) de 2010 à 2014. Ce travail a conduit à la réalisation de simulations d'objets mixtes basées sur la Mécanique des Milieux Continus. Les objets peuvent ainsi être décomposés en éléments de topologie et de loi de comportement différents. Ces éléments peuvent également évoluer au cours de la simulation pour répondre à des critères spécifiques afin d'adapter la simulation en précision ou en temps d'exécution. Pour cela, nous avons conçu un outil permettant de générer les équations relatives aux calculs des forces et de leurs dérivées en chacun des nœuds d'un élément en fonction de sa topologie et de sa loi de comportement. Nous nous sommes basés sur l'emploi de l'approche masse-tenseur pour réaliser cette simulation permettant une définition des interactions en chacun des nœuds d'un élément. Ces travaux ont été financés par le PRRH (Programme de Recherche de la Région Rhône-Alpes en Hadronthérapie) dans le cadre du projet européen ETOILE (Espace de Traitement Oncologique par Ions Légers dans le cadre Européen) (Alphonse et al. (2013)).

Sommaire

4.1	Contexte de ce travail de recherche	110
4.2	Notions préliminaires sur le modèle masse-tenseurs	112
4.3	Approche adoptée	114
4.4	Génération des équations	115
4.5	Un peu de détails techniques	122
4.6	Adaptation dynamique selon critères	127
4.7	Évaluation de notre approche	129
4.8	Discussion	133

4.1 Contexte de ce travail de recherche

Il est intéressant de pouvoir modifier au cours de la simulation la modélisation des éléments constituant l'objet dont le comportement mécanique est simulé. Ces changements peuvent permettre de répondre à des critères de précision de calcul ou encore de temps d'exécution. Nous pouvons par exemple imaginer débiter une simulation en utilisant une loi linéaire telle que la loi de Hooke (qui n'est valide que dans le cadre de petites déformations), puis si l'objet considéré subit des déformations plus importantes, orienter les calculs sur l'emploi d'une loi de comportement non-linéaire (valide en grands déplacements) qui nécessite plus de calculs. Ces changements peuvent aussi être portés par un raffinement que nous souhaitons opérer suite à une interaction ou un besoin de précision accru à un endroit spécifique de l'objet. Et le fait de pouvoir effectuer ce raffinement en autorisant différentes géométries peut permettre l'emploi d'algorithmes plus performants ou mieux adaptés au raffinement de la zone concernée. Il est donc pertinent de souhaiter modifier dynamiquement les éléments qui constituent un objet, aussi bien au niveau de leur topologie que de leur loi de comportement. Pour cela il faut mettre en place une structure de données permettant de combiner au sein d'un même objet des éléments de types différents.

Durant la thèse de Xavier Faure (2014b) nous nous sommes ainsi concentrés sur la réalisation de simulations physiques d'objets mixant des éléments de type différents en loi de comportement et topologie. De plus nous avons fait en sorte que ces éléments puissent évoluer durant la simulation, afin que celle-ci satisfasse des critères de temps ou de précision de calculs. Pour pouvoir définir les interactions aux nœuds de chacun de ces éléments, nous nous sommes orientés vers l'emploi de l'approche masse-tenseur. Ce modèle a été introduit en Informatique Graphique par Cotin (1997); Cotin et al. (2000b); Picinbono (2003); Picinbono et al. (2000, 2001) et étendu par Schwartz et al. (2005). Il donne une formulation du calcul des forces et de leurs dérivées en chacun des nœuds constituant un élément à partir des équations de la Mécanique des Milieux Continus. Cette formulation dépend de la topologie de l'élément ainsi que de sa loi de comportement. Pour gérer facilement l'emploi de tout type d'élément, nous avons par ailleurs développé un environnement permettant de générer automatiquement ces équations en utilisant une approche basée sur l'emploi du calcul formel. Au final, nous sommes capables de réaliser des simulations avec des objets dits mixtes en utilisant les équations générées pour chaque type d'élément.

Ce travail a été financé par le PRRH (Programme de Recherche de la Région Rhône-Alpes en Hadronthérapie) dans le cadre du projet européen ETOILE (Espace de Traitement Oncologique par Ions Légers dans le cadre Européen). Ce projet (Alphonse et al. (2013)) proposait la construction d'un centre médical dédié au traitement de certaines tumeurs cancéreuses par faisceaux d'ions carbone. Les recherches effectuées au sein du projet ETOILE visaient notamment à améliorer la précision de la balistique dans le cadre du traitement du cancer des poumons, c'est-à-dire à faire en sorte que les ions carbonés atteignent la tumeur du patient présente dans l'appareil respiratoire en limitant la zone de tissus sains touchés par ces ions. Il s'agit ainsi de déterminer la position optimale du patient et celle du faisceau en fonction de la localisation de la tumeur. Dans le cadre de ce projet, l'équipe SAARA du LIRIS dont je faisais partie, s'intéresse depuis plusieurs années à la mise au point d'un modèle bio-mécanique décrivant les mouvements thoraciques (abdomen, côtes, poumons, diaphragme, médiastin) du patient au cours du traitement. L'idée est ainsi de prédire de manière exacte les mouvements de la zone tumorale grâce à ce modèle bio-mécanique qui prend également en entrée des informations relatives au mouvement de la peau du thorax du patient (Faure et al. (2011)). C'est pourquoi, le critère de précision est autant mis en avant que celui du temps, nécessitant ainsi de nouvelles solutions pour gagner en temps de calcul sans perdre en précision des résultats.

Organisation de ce chapitre : Ce chapitre se poursuit de la façon suivante :

- La section 4.2 rappelle le modèle masse-tenseurs que nous avons utilisé pour simuler le comportement d'objets déformables.
- La section 4.3 présente l'approche générale que nous avons adoptée pour effectuer les simulations. L'idée est de permettre la simulation d'objets constitués d'éléments de type différents au niveau de leur topologie ou de leur comportement mécanique.
- La section 4.4 explique les différentes étapes de notre méthode permettant de générer les équations relatives à la formulation des forces et de leurs dérivées pour chacun des nœuds des éléments constituant nos objets dits mixtes. Ces équations sont établies en fonction des caractéristiques topologiques et mécaniques des éléments.
- La section 4.5 donne quelques détails techniques d'implémentation.
- La section 4.6 montre l'intérêt de la mise en place de notre approche pour adapter dynamiquement la simulation afin de mieux gérer le compromis usuel entre temps de calcul et précision numérique de la simulation.
- La section 4.7 présente les premiers résultats que nous obtenons.
- La section 4.8 termine ce chapitre par une discussion sur l'approche que nous proposons au travers de ce chapitre.

4.2 Notions préliminaires sur le modèle masse-tenseurs

Dans le chapitre 1, nous avons rappelé les notions de la Mécanique des Milieux Continus qui permettent de décrire de manière continue la déformation d'un objet soumis à l'action d'une contrainte. Nous avons notamment vu que la déformation ϕ peut être exprimée en fonction du déplacement $\vec{u}(x, y, z)$ au point (x, y, z) de l'objet avec $\phi(x, y, z) = (x, y, z) + \vec{u}(x, y, z)$. Le tenseur de déformation ϵ permet ensuite de quantifier cette déformation. Il est déduit du gradient de la déformation qui est lui-même défini par $\mathbb{F} = \nabla\phi = \mathbb{1} + \mathbb{U} = \mathbb{1} + \nabla\vec{u}$. Le tenseur de déformation ϵ peut ainsi être directement exprimé en fonction du gradient de déplacement $\mathbb{U} = \nabla\vec{u}$. Son choix est important et dépend du comportement mécanique attendu du matériau. Par exemple, dans le cadre d'une loi de comportement non-linéaire comme celle de Saint-Venant Kirchhoff, le tenseur de Green-Lagrange peut être employé (cadre non-linéarité géométrique). Et dans le cadre de petites perturbations avec de petits déplacements où la loi linéaire de Hooke peut être considérée, nous pouvons utiliser plus simplement la formulation linéarisée de ce même tenseur de Green-Lagrange (cadre linéarité géométrique).

Dans le chapitre 1, nous avons également détaillé le principe de la méthode des Éléments Finis qui est la méthode classique pour résoudre les équations de la MMC. Elle permet notamment la définition des matrices globales impliquées dans le système d'équations différentiels, issu de la dynamique Newtonienne, qui gouverne le mouvement de l'objet. Il est défini par :

$$\mathbf{M} \ddot{\mathbf{U}} + \mathbf{D} \dot{\mathbf{U}} + \mathbf{K} \mathbf{U} = \mathbf{F}. \quad (4.1)$$

avec \mathbf{F} le vecteur des forces exercées sur l'objet, \mathbf{U} le déplacement de l'objet, \mathbf{M} sa matrice de masse, \mathbf{D} sa matrice d'amortissement et \mathbf{K} sa matrice de rigidité.

Une méthode d'intégration est alors employée pour calculer l'état du système (déplacement et vitesse) au pas de temps suivant en fonction de l'accélération calculée à l'instant t . Nous avons également vu que l'emploi de la méthode d'Euler implicite qui est inconditionnellement stable (permettant l'emploi de pas de temps assez grands), nécessite en plus, le calcul des dérivées des forces par rapport aux déplacements et aux vitesses.

Nous allons ici nous intéresser à une alternative de la MEF : l'approche masse-tenseurs également basée sur une discrétisation du domaine en un nombre fini d'éléments. Cette approche a été proposée dans le domaine de l'informatique graphique par [Cotin \(1997\)](#) et étendue par la suite par ([Cotin et al. \(2000a\)](#); [Delingette \(2006\)](#); [Picinbono et al. \(2000\)](#)).

L'approche masse-tenseurs est basée sur le fait que la déformation de l'objet induite par des forces externes, génère une énergie de déformation W (naturellement dépendante des caractéristiques du matériau) et que la dérivée de cette énergie fournit ainsi la force opposée correspondante provoquant la déformation. Elle utilise ainsi la dérivée de Fréchet pour le calcul des forces. L'intérêt de cette approche par rapport à la MEF, réside dans le fait que les étapes de calculs sont beaucoup plus systématiques et procédurales, en partant des déplacements pour arriver aux forces internes, impliquant une facilité de formalisation.

Cette approche permet l'obtention d'une formulation explicite des matrices de rigidité qui sont reliées à chacun des éléments du maillage représentant l'objet. Ceci permet notamment d'optimiser les calculs lors de l'assemblage global de la matrice de rigidité \mathbf{K} . A noter qu'une formulation a tout d'abord été fournie dans le cadre de l'élasticité linéaire (emploi

de la loi de Hooke). Puis, une première extension a été fournie dans le cadre de l'élasticité non-linéaire (emploi de la loi de Saint-Venant Kirchhoff dans laquelle il y a une relation quadratique entre le tenseur de déformation ϵ et le tenseur de contraintes σ) et une autre dans le cadre visco-élastique non-linéaire. Par ailleurs, une optimisation des calculs a été proposée dans le cadre linéaire et non-linéaire, en employant une expression analytique de l'énergie de déformation et de la force élastique qui induit cette énergie.

De plus, lors de la présentation de l'approche masse-tenseur, [Cotin \(1997\)](#) montre tout son intérêt pour la gestion de changements topologiques dans la simulation. Dans [Picinbono et al. \(2000\)](#), une structure de donnée particulière est notamment proposée pour stocker les tenseurs élémentaires dans les tétraèdres, faces, arêtes et sommets du maillage de l'objet, facilitant d'autant plus la prise en compte d'éventuels changements topologiques.

Au final, l'approche masse-tenseur implique les étapes de calculs suivantes pour chacun des éléments du modèle discrétisé :

1. Discrétisation du champ de déplacement sur un élément (noté \vec{u}_E) impliquant la définition de fonction d'interpolations Λ_i qui sont choisies et définies en fonction de la géométrie des éléments considérés (hexaèdre, tétraèdre, etc.).
Notons que le déplacement est calculé à partir de la position d'origine, contrairement à la méthode HEML ([Goulette and Chen \(2015\)](#)) où le déplacement est calculé par rapport au pas de temps précédent.
2. Calcul de l'énergie de déformation de l'élément (notée W_E) définie en fonction du déplacement des nœuds de l'élément considéré. [Picinbono \(2001\)](#) a établi une formulation dans le cadre de la loi de Hooke avec le tenseur de Green-Lagrange linéarisé et dans le cadre de la loi de Saint-Venant Kirchhoff avec l'emploi du tenseur de déformation de Green-Lagrange.
3. Calcul de la force d'élasticité de l'élément (notée F_E) en dérivant l'énergie de déformation W_E . [Picinbono](#) a établi une formulation dans le cadre d'un tétraèdre à 4 nœuds. Sa mise en œuvre est facilitée par le stockage des informations de rigidité au niveau des sommets, arêtes, faces et éléments du maillage. [Schwartz \(2003\)](#) propose une formulation dans le cadre non-linéaire visco-élastique.
4. Calcul des nouvelles accélérations en introduisant les forces calculées dans l'équation de la dynamique Newtonienne.
5. Calcul du nouvel état (déplacement, vitesse) par intégration numérique des accélérations. [Picinbono \(2001\)](#) emploie un schéma d'intégration explicite qui ne nécessite pas le calcul des dérivées des forces.

Rappelons que notre souhait réside dans la possibilité d'**adapter dynamiquement la simulation numérique** en fonction des besoins de l'application afin de jouer sur le compromis performance/précision. Nous pouvons notamment faire évoluer : le nombre de nœuds ou le nombre d'éléments de la discrétisation de l'objet ; la géométrie de ces éléments ; la loi de comportement employée pour modéliser le comportement mécanique du matériau ; les fonctions de formes employées pour interpoler sur tout l'élément des informations connues uniquement sur les nœuds, etc. Pour réussir ce challenge, nous avons besoin d'un modèle : (1) qui gère des éléments différents au sein d'un même objet ; (2) qui s'adapte à ces changements notamment d'un point de vue mécanique.

L'approche masse-tenseurs procure le cadre mécanique adéquate pour cela. Nous présentons dans ce chapitre notre solution pour générer facilement les **formulations des forces appliquées sur chacun des nœuds des éléments du maillage, ainsi que celles de leurs dérivées** (nécessaires lors de l'emploi du schéma d'intégration d'Euler implicite). Ces formulations sont établies en fonction des déplacements de chacun des nœuds des éléments, quels que soient la géométrie de l'élément, le tenseur de déformation ou la loi de comportement choisis. Notre approche est basée sur l'emploi du calcul formel. C'est ainsi par ce biais que sont réalisées les étapes ①, ② et ③ de la méthode des masse-tenseurs rendant son usage simple et généralisable à toute loi de comportement.

4.3 Approche adoptée

La formulation des forces et de leurs dérivées dépend de plusieurs paramètres : de la loi de comportement du matériau (Hooke, Saint Venant-Kirchhoff, ...); de la géométrie (triangle, quadrangle, tétraèdre, hexaèdre, prisme, ...) de l'élément considéré au sein de la discrétisation de l'objet; du calcul adopté pour effectuer l'intégration permettant de formuler les forces à l'intérieur de l'élément (intégration analytique ou numérique en utilisant par exemple les points de Gauss); et d'un point de vue exécution du type d'unité de processeur choisi (CPU ou GPU). Ces paramètres, relatif à un élément de la modélisation, sont illustrés par la Fig. 4.1.

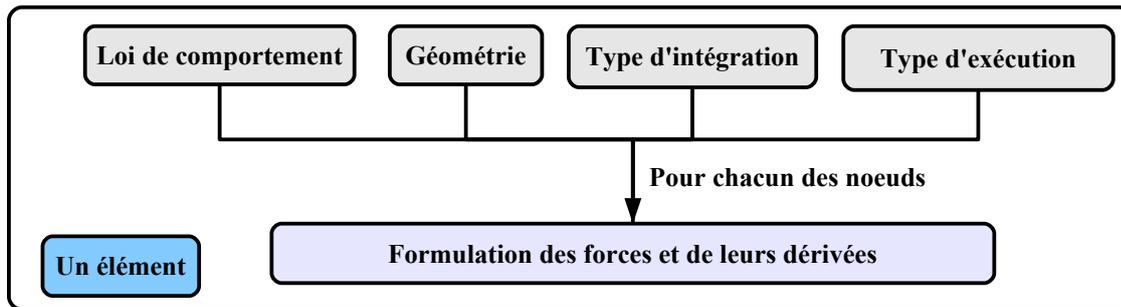


Figure 4.1 – Paramètres à prendre en compte pour formuler les équations des forces et de leurs dérivées en chacun des nœuds d'un élément dans le cadre de l'approche masse-tenseur.

En fonction du choix de ces différents paramètres, les formulations peuvent devenir assez fastidieuses, notamment dès qu'une loi de comportement non-linéaire est choisie. C'est pourquoi, nous avons employé le calcul formel (en utilisant le système Open Source Maxima) afin de générer automatiquement ces équations pour les différents jeux de paramètres possibles. Cette génération s'effectue en plusieurs étapes :

1. Génération de l'**équation relative à l'énergie de déformation** de l'élément considéré du matériau. Cette équation dépend des paramètres précédemment cités.
2. Afin d'optimiser l'exécution, nous **regroupons les données constantes** relatives à ces équations pour les effectuer en pré-traitement de la boucle de simulation.
3. Génération des **équation relatives à la force et leurs dérivées** à partir de l'équation obtenue précédemment de l'énergie de déformation.
4. **Découpe des équations** en plusieurs parties afin de faciliter leur compilation.

Une fois ces formulations obtenues, nous pouvons réaliser la boucle classique de la simulation basée sur la dynamique newtonienne. Pour cela, nous avons intégré ce travail au sein de la librairie Open Source SOFA (Allard et al. (2007)). A noter que l'exécution de la simulation est effectuée soit en séquentiel, soit en parallèle sur GPU.

4.4 Génération des équations

Nous allons voir à partir d'un exemple, les différentes étapes permettant d'établir la formulation des forces et de leurs dérivées en chacun des nœuds d'un élément en fonction de ses paramètres géométriques et mécaniques. Ces étapes sont ainsi exécutées au préalable pour chacun des jeux de paramètres possibles pour obtenir toutes les formulations en fonction du choix de ces paramètres.

Formulation de l'énergie de déformation élastique. En fonction de la géométrie d'un élément et de sa loi de comportement, un premier code `C++` permet de générer le code Maxima permettant de formuler l'énergie de déformation élastique de l'élément considéré. Nous allons voir les différentes étapes de ce premier programme.

Formulation du déplacement d'un point. Soient \vec{X} les coordonnées physiques d'un point à l'intérieur de l'élément (notées (x, y) en 2D ou (x, y, z) en 3D), et \vec{R} les coordonnées locales de ce point définies dans l'élément de référence (notées (ξ, η) en 2D ou (ξ, η, μ) en 3D). Le déplacement d'un point à l'intérieur d'un élément 3D nommé E est défini par

$$\vec{u}_E(x, y, z) \simeq \sum_{i=0}^{n-1} \Lambda_i(x, y, z) u_i,$$

avec n le nombre de nœuds P_i de coordonnées spatiales $X_i = (P_{ix}, P_{iy}, P_{iz})$ définissant cet élément, U_i le déplacement en coordonnées spatiales de chacun des nœuds P_i défini par rapport à leur position initiale, et $\Lambda_i(x, y, z)$ les fonctions de forme choisies en fonction de la géométrie de l'élément (cf. Table 4.1). Si nous considérons les coordonnées définies dans le repère local pour ce point à l'intérieur de l'élément de référence \hat{E} , nous avons :

$$\vec{u}_{\hat{E}}(\xi, \eta, \mu) \simeq \sum_{i=0}^{n-1} \Lambda_i(\xi, \eta, \mu) u_i.$$

Formulation du gradient de déplacement d'un point. Nous avons rappelé dans le chapitre 1 que la transformation géométrique permettant de passer de l'élément physique à l'élément de référence peut être définie par la matrice Jacobienne définie par :

$$J = \frac{\partial(x, y, z)}{\partial(\xi, \eta, \mu)}. \quad (4.2)$$

Ainsi, si nous notons $\nabla_{(x,y,z)}$ le gradient défini dans l'espace physique et $\nabla_{(\xi,\eta,\mu)}$ celui défini dans l'élément de référence, le gradient de la fonction f est exprimée par :

$$\nabla_{(x,y,z)} f = J^{-T} \nabla_{(\xi,\eta,\mu)} f.$$

La dérivée du déplacement \vec{U} exprimé dans les coordonnées locales (ξ, η, μ) est alors définie par :

$$\nabla_{(x,y,z)} \vec{u}_{\hat{E}}(\xi, \eta, \mu) = J^{-T} \nabla_{(\xi,\eta,\mu)} \vec{u}_{\hat{E}}(\xi, \eta, \mu).$$

L'algorithme 2 présente le code `C++` permettant de formuler en calcul formel, le déplacement de chacun des nœuds d'un élément. Puis le code 1 donne la formulation obtenue pour le déplacement ainsi que son gradient, en considérant un élément triangulaire dont le comportement mécanique est défini par la loi de Hooke, et en employant les fonctions de forme suivantes définies dans l'espace de référence :

$$\Lambda_0(\xi, \eta) = 1 - \xi - \eta, \quad \Lambda_1(\xi, \eta) = \xi \quad \text{et} \quad \Lambda_2(\xi, \eta) = \eta.$$

2D		Degré du polynôme = 1	i	$\partial\Lambda_i(\xi, \eta)/\partial\xi$	$\partial\Lambda_i(\xi, \eta)/\partial\eta$	
	Triangle \mathcal{P}^1 (n = 3)	$\Lambda_0(\xi, \eta) = 1 - \xi - \eta$	0	-1	-1	
		$\Lambda_1(\xi, \eta) = \xi$	1	1	0	
		$\Lambda_2(\xi, \eta) = \eta$	2	0	1	
		Degré du polynôme = 2	i	$\partial\Lambda_i(\xi, \eta)/\partial\xi$	$\partial\Lambda_i(\xi, \eta)/\partial\eta$	
	Quadrilatère \mathcal{Q}^1 (n = 4)	$\Lambda_0(\xi, \eta) = \frac{1}{4}(1 - \xi)(1 - \eta)$	0	$-\frac{1}{4}(1 - \eta)$	$-\frac{1}{4}(1 - \xi)$	
		$\Lambda_1(\xi, \eta) = \frac{1}{4}(1 + \xi)(1 - \eta)$	1	$\frac{1}{4}(1 - \eta)$	$-\frac{1}{4}(1 - \xi)$	
		$\Lambda_2(\xi, \eta) = \frac{1}{4}(1 + \xi)(1 + \eta)$	2	$\frac{1}{4}(1 + \eta)$	$\frac{1}{4}(1 + \xi)$	
		$\Lambda_3(\xi, \eta) = \frac{1}{4}(1 - \xi)(1 + \eta)$	3	$-\frac{1}{4}(1 + \eta)$	$\frac{1}{4}(1 - \xi)$	
3D		Degré du polynôme = 1	i	$\partial\Lambda_i(\xi, \eta, \mu)/\partial\xi$	$\partial\Lambda_i(\xi, \eta, \mu)/\partial\eta$	$\partial\Lambda_i(\xi, \eta, \mu)/\partial\mu$
	Tétraèdre \mathcal{P}^1 (n = 4)	$\Lambda_0(\xi, \eta, \mu) = 1 - \xi - \eta - \mu$	0	-1	-1	-1
		$\Lambda_1(\xi, \eta, \mu) = \xi$	1	1	0	0
		$\Lambda_2(\xi, \eta, \mu) = \eta$	2	0	1	0
		$\Lambda_3(\xi, \eta, \mu) = \mu$	3	0	0	1
		Degré du polynôme = 3	i	$\partial\Lambda_i(\xi, \eta, \mu)/\partial\xi$	$\partial\Lambda_i(\xi, \eta, \mu)/\partial\eta$	$\partial\Lambda_i(\xi, \eta, \mu)/\partial\mu$
	Hexaèdre \mathcal{Q}^1 (n = 8)	$\Lambda_0(\xi, \eta, \mu) = \frac{1}{8}(1 - \xi)(1 - \eta)(1 - \mu)$	0	$-\frac{1}{8}(1 - \eta)(1 - \mu)$	$-\frac{1}{8}(1 - \xi)(1 - \mu)$	$-\frac{1}{8}(1 - \xi)(1 - \eta)$
		$\Lambda_1(\xi, \eta, \mu) = \frac{1}{8}(1 + \xi)(1 - \eta)(1 - \mu)$	1	$\frac{1}{8}(1 - \eta)(1 - \mu)$	$-\frac{1}{8}(1 + \xi)(1 - \mu)$	$-\frac{1}{8}(1 + \xi)(1 - \eta)$
		$\Lambda_2(\xi, \eta, \mu) = \frac{1}{8}(1 + \xi)(1 + \eta)(1 - \mu)$	2	$\frac{1}{8}(1 + \eta)(1 - \mu)$	$\frac{1}{8}(1 + \xi)(1 - \mu)$	$-\frac{1}{8}(1 + \xi)(1 - \eta)$
		$\Lambda_3(\xi, \eta, \mu) = \frac{1}{8}(1 - \xi)(1 + \eta)(1 - \mu)$	3	$-\frac{1}{8}(1 - \eta)(1 - \mu)$	$\frac{1}{8}(1 + \xi)(1 - \mu)$	$-\frac{1}{8}(1 - \xi)(1 + \eta)$
		$\Lambda_4(\xi, \eta, \mu) = \frac{1}{8}(1 - \xi)(1 - \eta)(1 + \mu)$	4	$-\frac{1}{8}(1 - \eta)(1 + \mu)$	$-\frac{1}{8}(1 - \xi)(1 + \mu)$	$\frac{1}{8}(1 - \xi)(1 - \eta)$
		$\Lambda_5(\xi, \eta, \mu) = \frac{1}{8}(1 + \xi)(1 - \eta)(1 + \mu)$	5	$\frac{1}{8}(1 - \eta)(1 + \mu)$	$-\frac{1}{8}(1 + \xi)(1 + \mu)$	$\frac{1}{8}(1 + \xi)(1 - \eta)$
		$\Lambda_6(\xi, \eta, \mu) = \frac{1}{8}(1 + \xi)(1 + \eta)(1 + \mu)$	6	$\frac{1}{8}(1 + \eta)(1 + \mu)$	$\frac{1}{8}(1 + \xi)(1 + \mu)$	$\frac{1}{8}(1 + \xi)(1 + \eta)$
		$\Lambda_7(\xi, \eta, \mu) = \frac{1}{8}(1 - \xi)(1 + \eta)(1 + \mu)$	7	$-\frac{1}{8}(1 - \eta)(1 + \mu)$	$\frac{1}{8}(1 - \xi)(1 + \mu)$	$\frac{1}{8}(1 - \xi)(1 + \eta)$

Table 4.1 – Fonctions de forme et dérivées en fonction avec comme coordonnées locales (ξ, η) ou (ξ, η, μ) .**Algorithme 2** : Algorithme du code C++ pour le calcul du déplacement.

- 1: **Pour** chaque nœud i de l'élément **Faire**
- 2: $L_i := \Lambda_i(\vec{R})$; // Fonctions de forme en fonction de la géométrie
- 3: **Pour** chaque nœud i de l'élément **Faire**
- 4: $U_i(R) := L_i * U_i$;
- 5: **Pour** chaque nœud i de l'élément **Faire**
- 6: $U(R) := U(R) + U_i(R)$;

Code 1 Formulation symbolique du déplacement et de son gradient pour un triangle en utilisant l'environnement de calcul formel Maxima.

```

1 // Fonctions de forme avec R = (R0, R1, R2)
2 L0:=1-R0-R1; L1:=R0; L2:=R1;
3
4 // Déplacement
5 U0:=[L0*U0x,L0*U0y]; U1:=[L1*U1x,L1*U1y]; U2:=[L2*U2x,L2*U2y];
6
7 U:=U0+U1+U2;
8
9 // Gradient du déplacement
10 JT := matrix([diff((1-R0-R1)*P0x+(R0)*P1x+(R1)*P2x,R0),
11               diff((1-R0-R1)*P0y+(R0)*P1y+(R1)*P2y,R0)],
12               [diff((1-R0-R1)*P0x+(R0)*P1x+(R1)*P2x,R1),
13               diff((1-R0-R1)*P0y+(R0)*P1y+(R1)*P2y,R1)]);
14
15 GradU:=JT^(-1)matrix([diff(U,R0),diff(U,R1)]);
16
17 GradUT:=matrix(transpose(GradU));

```

Formulation de l'énergie de déformation. Notons ϵ_{loi} et W_{loi} le tenseur de déformation et la densité d'énergie (*i.e.* énergie de déformation définie pour une unité de volume) de la loi de comportement considérée pour modéliser le comportement mécanique de l'élément. L'énergie de déformation W_E de l'élément est ainsi définie par :

$$\begin{aligned} W_E &= \int_E W_{loi}(x, y, z) dx dy dz \\ &= \int_{\hat{E}} W_{loi}(\xi, \eta, \mu) \det(J) d\xi d\eta d\mu \\ &= \sum_{i=1}^{n_g} w_i W_{loi}(\xi_i, \eta_i, \mu_i) \det(J) \end{aligned}$$

où J est la Jacobienne de la déformation définie selon l'équation (4.2) par rapport à la transformation géométrique permettant de passer des coordonnées spatiales (x, y, z) aux coordonnées de l'élément de référence (ξ, η, μ) , et où n_g représente le nombre de points (ξ_i, η_i, μ_i) (appelés points de Gauss) de la discrétisation employée pour faire le calcul de l'intégrale en utilisant les fonctions de pondération w_i .

Pour des petites déformations généralement inférieures à 10% de la taille de l'objet, nous considérons que la loi de comportement est linéaire avec ainsi une relation linéaire entre les tenseurs de contrainte et de déformation. Pour modéliser ce comportement, nous avons vu que nous pouvons alors employer le tenseur linéarisé de Green-Lagrange défini par :

$$\begin{aligned} \epsilon_{Hooke} &= \frac{1}{2}(\mathbb{U}^T + \mathbb{U}) \\ \Rightarrow \epsilon_{Hooke}(\xi, \eta, \mu) &= \frac{1}{2} \left(\nabla_{(x,y,z)} u^T(\xi, \eta, \mu) + \nabla_{(x,y,z)} u(\xi, \eta, \mu) \right). \end{aligned}$$

Pour des déplacements plus importants, nous pouvons considérer le tenseur de Green-Lagrange (ou Green Saint-Venant) dans son intégralité, c'est-à-dire incluant sa partie non-linéaire :

$$\begin{aligned} \epsilon_{svk} &= \frac{1}{2}(\mathbb{U}^T + \mathbb{U}) + \frac{1}{2}(\mathbb{U}^T \cdot \mathbb{U}) \\ \Rightarrow \epsilon_{svk}(\xi, \eta, \mu) &= \frac{1}{2} \left(\nabla_{(x,y,z)} u^T(\xi, \eta, \mu) + \nabla_{(x,y,z)} u(\xi, \eta, \mu) + \nabla_{(x,y,z)} u^T(\xi, \eta, \mu) \nabla_{(x,y,z)} u(\xi, \eta, \mu) \right). \end{aligned}$$

Nous avons également vu que la densité de l'énergie de déformation mesurant l'énergie de déformation par unité de volume d'un domaine infinitésimal (établi autour d'un point matériel de coordonnées locales (ξ, η, μ) dans l'élément de référence) est définie par :

$$W(\xi, \eta, \mu) = \frac{\lambda}{2} (\text{Tr } \epsilon(\xi, \eta, \mu))^2 + \mu \text{Tr } \left(\epsilon(\xi, \eta, \mu)^2 \right),$$

avec λ et μ les coefficients de Lamé caractérisant les propriétés mécaniques du matériau. Le code 2 présente la formulation en calcul formel de l'énergie de déformation d'un élément triangulaire dont le comportement mécanique est modélisé par la loi de Hooke.

Code 2 Formulation symbolique de l'énergie de déformation pour un triangle en Hooke.

```

1 // Code Maxima genere pour un element triangulaire avec loi de Hooke
2 E:=1/2 * (GradU + GradUT);
3
4 // p0 et p1 les coefficients de Lamé
5 W0:=p0/2 * trace(E)^2;
6 W1:=p1 * trace(E^2);
7
8 // Integration analytique
9 WE :=integrate(integrate(W0(R0,R1)*det(JT(R0,R1)),R1,0,1-R0), R0,0,1);
10
11 // Integration en utilisant des points de Gauss
12 // a = 1/6, b = 2/3, omega = 1/6 (par exemple)
13 intW0_0:=omega * W0(a,a) * det(JT(a,a));
14 intW1_0:=omega * W1(a,a) * det(JT(a,a));
15 intW0_1:=omega * W0(b,a) * det(JT(b,a));
16 intW1_1:=omega * W1(b,a) * det(JT(b,a));
17 intW0_2:=omega * W0(a,b) * det(JT(a,b));
18 intW1_2:=omega * W1(a,b) * det(JT(a,b));
19
20 WE:=intW0_0 + intW1_0 + intW0_1 + intW1_1 + intW0_2 + intW1_2;

```

Extraction des données constantes. Pour optimiser l'exécution de la simulation, nous regroupons les parties constantes de la formulation obtenue en calcul formel de l'énergie de déformation afin de ne calculer ces parties constantes qu'une seule fois en tant que pré-traitement de la boucle de simulation. A noter que ce processus est effectué sur la formulation de l'énergie de déformation puisque les autres étapes de la simulation (calcul des forces et de leurs dérivées) n'introduisent pas de nouvelles variables.

Expressions commutative. Afin de faciliter cette extraction, nous convertissons les formulations symboliques en expressions commutatives (*i.e.* des expressions ne contenant ni soustraction, ni division, ni opérateur puissance). Comme illustré sur la Fig. 4.2, les expressions commutatives peuvent être représentées par un graphe¹.

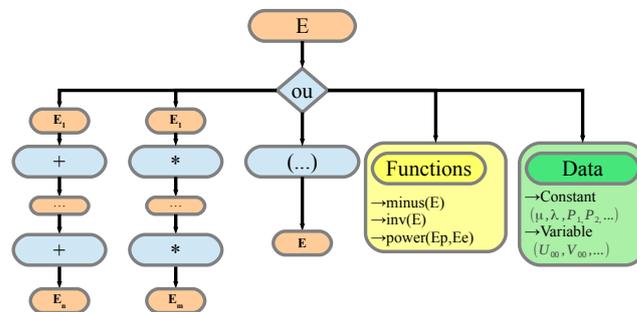


Figure 4.2 – Expressions commutatives représentées par un graphe.

1. Les figures de type graphe de ce chapitre sont issues du manuscrit de la thèse de Xavier Faure (2014b).

Les données de ces expressions sont divisées en deux catégories : les données constantes (coefficients de Lamé, positions initiales des nœuds) et les données variables (déplacement et vitesse des nœuds). Les fonctions $minus()$, $inv()$ et $power()$ permettent de changer les opérateurs mathématiques non-commutatifs $-$, $/$ et \wedge par des fonctions commutatives.

Ainsi l'expression

$$(a - b * (b + a) \wedge 2) / (a + b) \wedge 2$$

est modifiée en l'expression commutative suivante

$$(a + minus(b * power((b + a), 2))) * inv(power((a + b), 2)).$$

Extraction. L'algorithme 3 présente le processus d'extraction des données constantes des expressions commutatives. Les données constantes sont regroupées ensuite pour ne former plus qu'une seule donnée constante. Nous illustrons ce processus par un exemple modifiant l'expression commutative $varExp$ présentée par la Fig. 4.3. Dans cette figure, $P1x, P1y$ correspondent aux coordonnées de la position initiale du nœud 1, $U0x, U0y$ correspond au déplacement du nœud 0, et $p0$ correspond au coefficient de Lamé λ .

Algorithme 3 : Extraction des données constantes de $varExp$.

```

1: varExp = Exp // Initialisation
2: nbCstExp = 0 // Compteur d'expressions constantes
3: Tant Que données constantes dans svarExp Faire
4:   // Trouver la première données constante
5:   FirstCstData = varExp → firstConstantData()
6:   Tant que siblings(FirstCstData) sont toutes constantes Faire
7:     FirstCstData = parent(FirstCstData)
8:   Fin Tant Que
9:   // Parent avec des siblings non constants
10:  ParentVarData = parent(FirstCstData)
11:  // Liste des expr cst parmi les siblings de ParentVarData
12:  CstDataList = constantExpr(siblings(ParentVarData))
13:  Si ParentVarData != fonction Alors
14:    Si ParentVarData == sum Alors
15:      CstDataList = sum(CstDataList)
16:    Sinon
17:      CstDataList = product(CstDataList)
18:    Fin de Si
19:  Fin de Si
20:  // Pour chaque expression de CstDataList
21:  Pour i allant de 0 à nbCstList Faire
22:    // Une expression constante
23:    ExprName =  $D_{nbCstExp}$ 
24:    varExp → searchReplace( $CstDataList_i$ , ExprName)
25:    nbCstExp++
26:  Fin de Pour
27: Fin de Tant Que

```

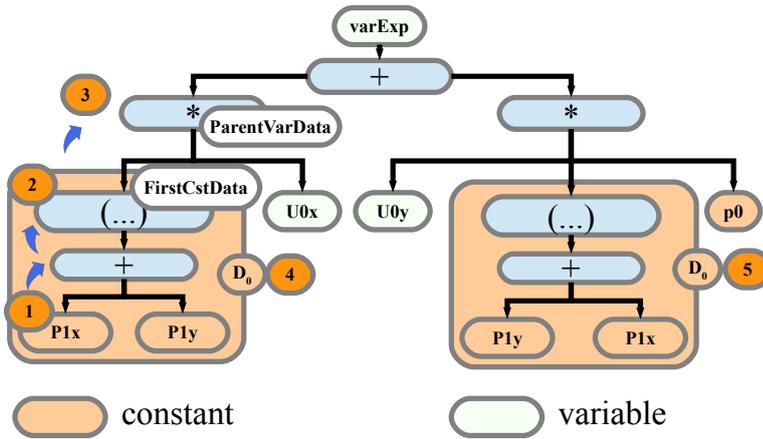


Figure 4.3 – $varExp$ est une expression commutative contenant des données constantes et des variables.

Regardons l'algorithme 3. La première donnée constante trouvée est $P1x$ (étape notée 1). Son parent le plus éloigné avec des frères non-constants est l'expression parenthésée (étape 2). Son parent est noté $ParentVarData$ qui a au moins un enfant variable ($U0x$ dans notre exemple) et comme enfant constant l'expression parenthésée (étape 3). Cette expression constante est notée D_0 (étape 4) également trouvée dans la partie droite de l'expression $varExpr$ (étape 5). Ensuite, la seconde passe de l'algorithme 3 trouve la donnée constante $p0$ ayant une variable frère $U0y$. Son père noté $ParentVarData$ est l'opérateur $*$. Ainsi, une nouvelle donnée constante notée D_1 est créée. Cela correspond au produit de D_0 avec $p0$. La Fig. 4.4 présente l'expression optimisée obtenue après l'extraction de deux données constantes.

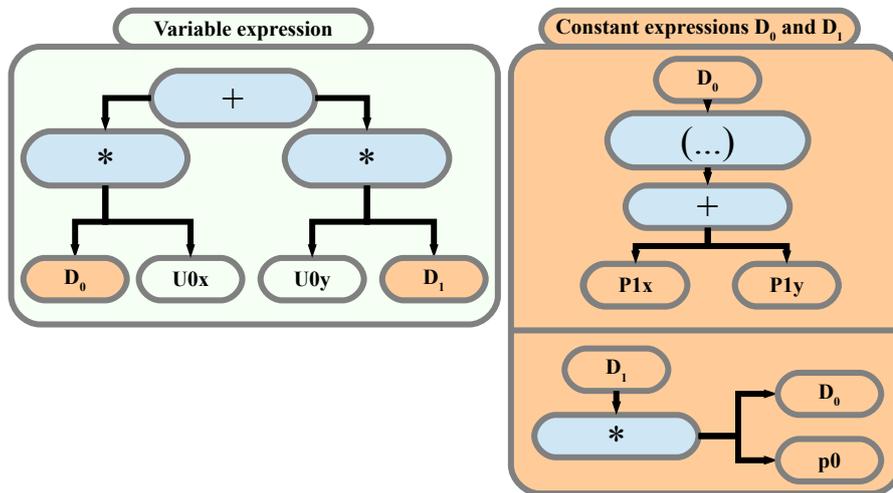


Figure 4.4 – Expression optimisée obtenue après avoir extrait deux données constantes.

Au final, la formulation de l'énergie de déformation est un polynôme dépendant uniquement du déplacement des nœuds. Une fois cette expression obtenue et optimisée, nous pouvons calculer les forces et leurs dérivées pour chacun des nœuds. Ces calculs sont effectués directement à partir des expressions polynomiales.

Formulation Maxima des forces et de leurs dérivées. En considérant un élément E d'énergie de déformation W_E , les forces appliquées en chacun de ces nœuds P_i pour $i \in [0, n - 1]$ sont définies par :

$$\vec{F}_E^\rightarrow(P_i) = -\frac{\partial W_E(P_i)}{\partial u_i}.$$

Ainsi si nous considérons l'objet 3D entier impliquant m nœuds, la force appliquée sur le nœud $P_{\mathcal{I}}$ avec $\mathcal{I} \in [0, m - 1]$ est calculée en sommant toutes les contributions des éléments voisins. Nous avons ainsi :

$$F(P_{\mathcal{I}}) = F_{\mathcal{I}} = \sum_{E \in \mathcal{N}_{\mathcal{I}}} F_E(P_i) = \sum_{E \in \mathcal{N}_{\mathcal{I}}} -\frac{\partial W_E(P_i)}{\partial u_i},$$

où $\mathcal{N}_{\mathcal{I}}$ correspond à l'ensemble des éléments contenant le nœud $P_{\mathcal{I}}$. Soit $i \in [0, n - 1]$ l'indice local du nœud de l'élément E qui correspond à l'indice global \mathcal{I} dans l'objet (en considérant l'indexation globale des sommets de l'objet).

Alors que l'information des forces est suffisante pour un schéma d'intégration explicite ou semi-implicite (permettant de calculer les vitesses et les déplacements à partir des accélérations et vitesses), le calcul de leurs dérivées est nécessaire lorsqu'un schéma d'intégration implicite est utilisé. Compte tenu d'un élément E , les écarts de la force sur le nœud P_i sont donnés par $\delta F_E(P_i) = [\delta F_E^0(P_i) \dots \delta F_E^{n-1}(P_i)]$, où $\delta F_E^j(P_i) = [\partial F_E(P_i) / \partial u_j]_{3,3}$ pour $j \in [0, n - 1]$ en considérant un objet à 3 dimensions.

Un second programme C++ est alors employé pour générer le code formel qui sera exécuté pour obtenir la formulation des forces et de leurs dérivées appliquées en chacun des nœuds i . L'algorithme 4 présente les étapes principales de ce programme. Les forces sont ainsi calculées en utilisant la formulation de l'énergie obtenue précédemment, c'est-à-dire à partir d'une expression polynomiale \mathcal{P} qui est fonction de u_i . Les coefficients des termes dépendent des données constantes. Le code 3 présente ainsi la formulation Maxima des forces et de leurs dérivées obtenue pour l'élément triangulaire de notre exemple.

Algorithme 4 : Génération de la formulation des forces et de leurs dérivées pour chacun des nœuds d'un élément.

```

1: // Reformulation de l'énergie totale de déformation
2: IW = P(U_i ∈ [0..n - 1], D_j ∈ [0..m - 1])

3: // Calcul des forces
4: Pour i allant de 0 à n - 1 Faire
5:   F_i = -∂IW/∂u_i

6: // Calcul des dérivées des forces
7: Pour i allant de 0 à n - 1 Faire
8:   Pour k allant de 0 à dim - 1 Faire
9:     Pour j allant de 0 à n - 1 Faire
10:      Pour l allant de 0 à dim - 1 Faire
11:        dF_{ik,jl} = ∂F_i[k]/∂u_{jl}
    
```

Code 3 Formulation des forces et de leurs dérivées en utilisant Maxima.

```

1 // Code Maxima pour les noeuds du triangle
2 F0x: -diff(WE,U0x); F0y: -diff(WE,U0y); F0:[F0x,F0y];
3 F1x: -diff(WE,U1x); F1y: -diff(WE,U1y); F1:[F1x,F1y];
4 F2x: -diff(WE,U2x); F2y: -diff(WE,U2y); F2:[F2x,F2y];
5
6 dF00 := diff(F0,U0); dF01 := diff(F0,U1); dF02 := diff(F0,U2);
7 dF10 := diff(F1,U0); dF11 := diff(F1,U1); dF12 := diff(F1,U2);
8 dF20 := diff(F2,U0); dF21 := diff(F2,U1); dF22 := diff(F2,U2);

```

Découpe des formulations symboliques. L'étape suivante consiste à découper les expressions commutatives obtenues dans Maxima en une somme d'expressions plus petites. En effet, les formulations initialement obtenues sont trop longues pour le compilateur (par exemple, la mémoire du processeur est insuffisante pour un hexaèdre dont le comportement mécanique est régi par la loi de Saint Venant Kirchhoff). Cette découpe peut être effectuée à trois endroits :

- Une loi de comportement est généralement définie comme une somme de plusieurs termes induisant une découpe aisée de l'expression.
Par exemple, $W = \frac{\lambda}{2} (\text{tr } \epsilon)^2 + \mu \text{ tr } \epsilon^2$ peut être facilement découpée en 2. Nous notons n_l le nombre de découpes dues à la définition de la loi de comportement.
- Le calcul de l'énergie totale de déformation induit une intégrale impliquant n_g points de Gauss ce qui induit facilement une découpe en n_g termes.
- Le calcul des dérivées des forces est défini comme une somme de produits de matrices divisible selon le nombre de dimensions (noté n_d) de l'objet simulé considéré.

4.5 Un peu de détails techniques

Selon les choix effectués (loi de comportement, géométrie de l'élément, découpe des expressions générées par le calcul formel), nous pouvons obtenir des centaines d'équations pour une simple combinaison. Nous devons alors gérer de manière efficace l'ensemble des fichiers générés sachant qu'une équation est stockée par fichier.

Parties constantes des expressions. Les fichiers relatifs aux parties constantes des équations des forces et de leurs dérivées sont stockées dans le répertoire `FunctionConstantData`, chacune des combinaisons correspondant ensuite à un sous-répertoire. Ensuite, les noms des fichiers correspondent aux informations relatives aux paramètres choisis. Ainsi, le fichier `dFHookeHexa8N3DI8DerivativeSplit4.data` correspond au 4^e groupe d'équations de la partie constante de la dérivée des forces (dF) d'un hexaèdre ayant 8 nœuds (`Hexa8N3D`), modélisé par la loi de Hooke où les découpes des équations ont été effectués uniquement sur le produit matrice vecteur (`DerivativeSplit`).

Nous avons ensuite à définir le programme C++ correspondant. Ainsi illustré par l'Algorithme 5, une classe de base abstraite C++ (appelée FCD pour `FunctionConstantData`) est définie. Cette classe est composée de deux fonctions virtuelles : `init()` effectuant l'allocation de la mémoire nécessaire au stockage des équations issus de chacun des groupes ; `compute()` calculant le résultat de l'équation en séquentiel. Rappelons que ce calcul n'est effectué qu'une seule fois lors de l'initialisation. Le paramètre `DataElement` contient les constantes physiques et les informations relatives à l'élément considéré. Ensuite, pour chacun des groupes de combinaisons, nous définissons la classe dérivée correspondante contenant les expressions des équations.

Algorithme 5 : Définitions de la classe abstraite `FCD` et d'une classe dérivée servant pour la fonction `dForce`.

```

1: class FCD
2: {
3:     public :
4:     virtual FCD() {}
5:     virtual void operator()(DataElement &, VectorDouble &) = 0;
6:     virtual void init(DataElement &) = 0;
7: };
8:
9: class FCDdFHookeHexa8N3DI8DerivativeSplit4 : public FCD
10: {
11:     public :
12:     void compute(DataElement &, VectorDouble &) {
13:         #include "FCDdFSaintVenantHexa8N3DDerivativeSplit4.data" }
14:     void init(DataElement &);
15: };
    
```

Parties variables des expressions. Les fichiers relatifs aux parties variables des expressions des forces et de leurs dérivées sont stockés dans le répertoire `FunctionConstitutiveLaw`, chacune des combinaisons de paramètres correspondant à un sous-répertoire. Les noms des fichiers correspondent quant à eux à la combinaison ainsi qu'au numéro du nœud considéré et de sa dimension (*i.e.* en x , y ou z).

Ensuite, le code `C++` correspondant peut être écrit. L'algorithme 6 illustre cela. Nous avons à définir deux classes abstraites (nommées `FCLForce` et `FCLDForce` - `FCL` pour `FunctionConstitutiveLaw`) composées d'une fonction virtuelle permettant d'effectuer le calcul des forces et de leurs dérivées. Ensuite, pour chaque groupe d'équations d'une combinaison, nous devons définir la classe dérivée correspondante. Les calculs peuvent ensuite être exécutés en séquentiel ou en parallèle (Faure et al. (2012b)).

Algorithme 6 : Définition de la classe `FunctionConstitutiveLaw`.

```

1: class FCLForce
2: {
3:     public :
4:     FCLForce() {}
5:     virtual FCLForce() {}
6:     virtual void compute(VecDeriv &, const Coord *, DataElement &) = 0;
7: };
8:
9: class FCLDForce
10: {
11:     public :
12:     FCLDForce() {}
13:     virtual FCLDForce() {}
14:     virtual void compute(VecDeriv&, const Coord*, const Coord*, DataElement &, double)=0;
15: };
16:
17: class FCLdFHookeHexa8N3DI8DerivativeSplit4 : public FCLForce
18: {
19:     public :
20:     FCLDForce() {}
21:     FCLDForce() {}
22:     void compute(VecDeriv & dF, const Coord *, const Coord *, DataElement & d, double k)
23:     {
24:         Pour  $i$  allant de 0 à  $n - 1$  Faire
25:         Pour  $j$  allant de 0 à  $dim - 1$  Faire
26:             dF[d.ind[i]][j] -= (#include "FCLdFHookeHexa8N3DI8DerivativeSplit4.ij.h")*k;
27:     }
28: };
    
```

Implémentation dans SOFA. Nous avons utilisé la librairie Open Source SOFA ([Allard et al. \(2007\)](#)) pour réaliser nos simulations. Dans cette librairie, les objets simulés sont définis par leur topologie, géométrie, modèle visuel (généralement en chargeant par exemple un fichier `.obj`), leur état mécanique (position, vitesse, accélération, forces stockées dans des vecteurs), leur comportement mécanique (loi de comportement), ainsi que par un modèle de collision. La simulation est décrite par un graphe de scène permettant de lier les différents composants (modules C++ de la librairie). Pour intégrer notre modèle dans la librairie SOFA ([Faure \(2014a\)](#)), nous avons ensuite à définir deux fonctions appelées (`addForce()` et `addDforce()`) effectuant le calcul des forces et des dérivées des forces en chacun des nœuds de l'objet simulé.

L'algorithme 7 présente la récupération des résultats des calculs des forces de chacun des nœuds d'un élément. Ainsi pour chaque élément, nous utilisons la formulation Maxima correspondant aux caractéristiques de l'élément (loi de comportement et géométrie). Rappelons que la partie constante des forces est calculée uniquement lors de l'initialisation de la simulation, tandis que la partie variable est calculée au sein de la fonction `addForce()` et la dérivée des forces au sein de la fonction `addDforce()`.

Algorithme 7 : Récupération des expressions Maxima des forces des nœuds.

```

1: // loi : loi de comportement de l'élément
2: // geom : géométrie de l'élément
3: // split : une partie de la formulation issue de sa découpe
4: // force : 0 pour la force ; 1 pour la dérivée de la force

5: // Au sein de l'initialisation de la simulation
6: // Calcul des parties constantes
7: Pour chaque élément i Faire
8:   Pour chaque split j Faire
9:     listFCD[force][law][geom][split][j].compute(data[i].data[force][j]) ;

10: // Au sein des fonctions addForce() ou addDforce()
11: // Calcul des parties variables
12: Pour chaque élément i Faire
13:   Pour chaque split j Faire
14:     listFCL[force][law][geom][split][j].compute(f,data[i]) ;

```

L'implémentation GPU ([Faure et al. \(2012a,b\)](#)) est basée sur une boucle sur chacune des combinaisons (loi, topologie et découpe). Ainsi pour chacun des fragments, nous exécutons n kernels, présentés par le code 4 où n_a correspond au nombre d'éléments de la combinaison.

En exemple de simulation d'objets hybrides, la Fig. 4.5 présente le maillage d'un lapin réalisé avec différents types d'éléments au niveau topologique (tétraèdres, hexaèdres, prismes, pyramides – réalisé avec l'aide de Claudio Lobos), avec différentes lois de comportement mécanique (Hooke, Saint Venant-Kirchhoff). Les transitions entre les régions fines et plus grossières sont effectuées en utilisant des *patterns* ou schéma de raffinement. Cette technique combine deux avantages : les hexaèdres permettent l'obtention d'une simulation robuste et précise et les tétraèdres, pyramides et prismes permettent un ajustement plus juste au niveau de la surface du maillage ([Lobos et al. \(2009, 2012\)](#)).

Code 4 Définition du kernel `addForce()`.

```

1  const sampler_t samplerA = CLK_NORMALIZED_COORDS_FALSE |
2  CLK_ADDRESS_NONE | CLK_FILTER_NEAREST;
3  __kernel void compute(
4  __global __read_only float4 * const nodes,
5  __global __read_only float4 * const nodesT,
6  __global __read_only int4 * const elts,
7  __global __read_only int4 * const eltsToF,
8  __global __read_only float4 * const data,
9  __global __write_only float4 * F) {
10
11  // Recuperation du nombre de kernels
12  unsigned int e = get_global_id(0);
13
14  // Cas ou l element courant est valide
15  if (elts[e].s0!=-1)
16  {
17  // Variables
18  int4 n0; int4 nToF0; float4 tmp;
19  float u[3][2]; float D[12]; float4 vecd;
20
21  // nodes et nodesToForce
22  n0 = elts[0+1*e]; nToF0 = eltsToF[0+1*e];
23
24  // Deplacement
25  tmp = nodesT[n0.s0] - nodes[n0.s0];
26  tmp = nodesT[n0.s1] - nodes[n0.s1];
27  tmp = nodesT[n0.s2] - nodes[n0.s2];
28
29  u[0][0] = tmp.s0; u[0][1] = tmp.s1;
30  u[1][0] = tmp.s0; u[1][1] = tmp.s1;
31  u[2][0] = tmp.s0; u[2][1] = tmp.s1;
32
33  // Passage de la memoire globale a la memoire locale
34  vecd = data[e*3 + 0];
35  D[0] = vecd.s0; D[1] = vecd.s1; D[2] = vecd.s2; D[3] = vecd.s3;
36  vecd = data[e*3 + 1];
37  D[4] = vecd.s0; D[5] = vecd.s1;
38  ...
39
40  // F0 dans chaque dimension
41  F[n0.s0+nbNodesMax*nToF0.s0] = (float4)(
42  (#include "FHookeTriangle3N2DNoSplitForce0C0_00.h")*-1,
43  (#include "FHookeTriangle3N2DNoSplitForce0C0_01.h")*-1, 0,0);
44  ...
45  }
46  }

```

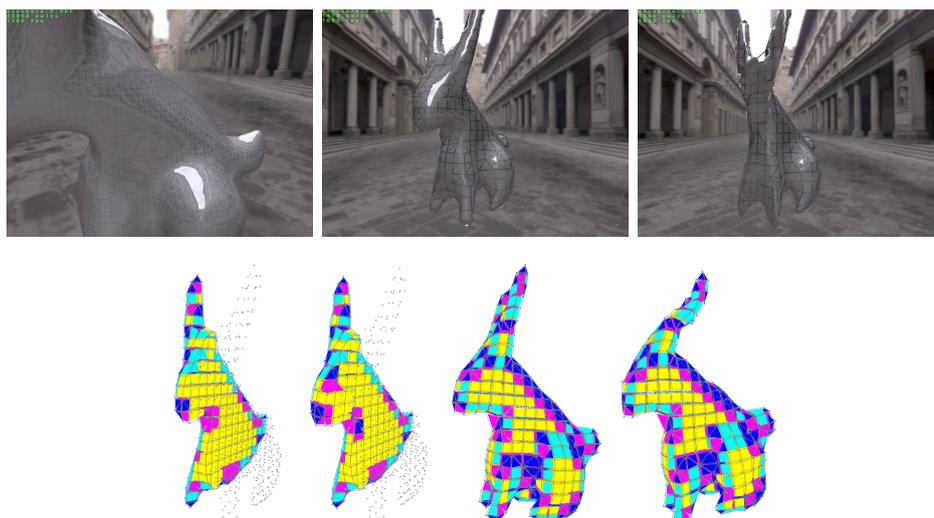


Figure 4.5 – Modélisation d'un objet comprenant différents types d'éléments.

La Fig. 4.6 montre la déformation d'une barre composée d'éléments de topologie différente. La dimension de la poutre est de $8 \times 1 \times 1 \text{ m}^3$ avec une masse volumique $\rho = 1\,000 \text{ Kg.m}^{-3}$ et un module de Young de 100 MPa. La Fig. 4.7 présente une comparaison de cette simulation avec la solution analytique. Nous obtenons un écart de 1 pour 1000.

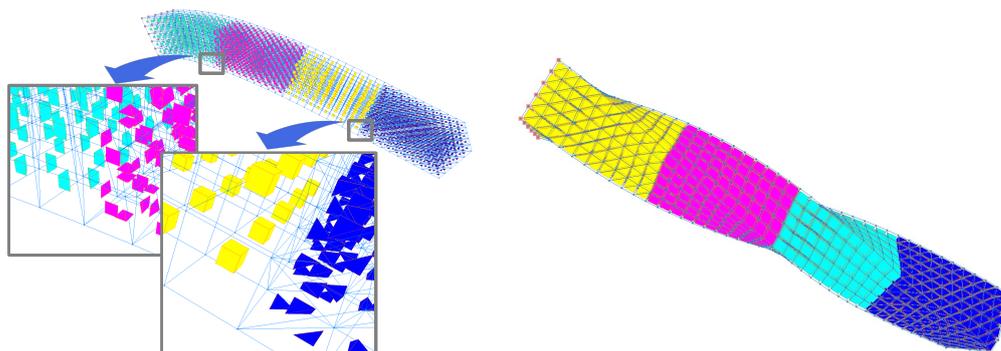


Figure 4.6 – Simulation en flexion et torsion d'une barre intégrant des éléments de topologie différente : prismes, pyramides, hexaèdres et tétraèdres avec la loi de comportement de Saint-Venant Kirchhoff.

Nous avons réalisé un autre test en considérant plusieurs lois de comportement : les prismes sont associés à une loi de Hooke, et les autres éléments à la loi de Saint-Venant Kirchhoff. La Fig. 4.8 présente une comparaison de notre résultat avec la solution analytique. Nous obtenons un écart de 1 pour 1000. Par ailleurs, la flèche du maillage déformé a la même précision que lorsque toute la barre est avec une loi de comportement de Saint-Venant Kirchhoff, les prismes étant situés dans la zone où il y a le moins de déformation.

Nous sommes ainsi capables de simuler des modèles mixtes au niveau de la loi de comportement et du type d'élément. Par contre, rappelons que la création de modèles mixtes soulève le problème de la gestion des jonctions entre les différents éléments aussi bien au niveau géométrique que mécanique pour résoudre correctement les équations sur les nœuds communs à des éléments de différents types.

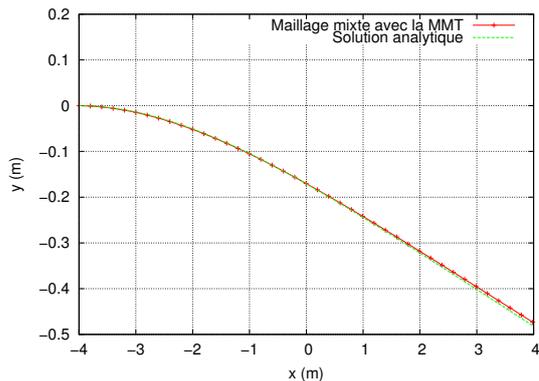


Figure 4.7 – Test de flexion en employant la loi de Saint-Venant Kirchhoff pour une poutre comportant des éléments de topologie différente (prismes, pyramides, hexaèdres et tétraèdres).

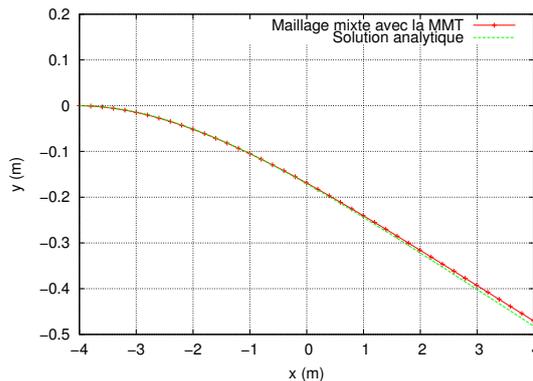


Figure 4.8 – Test de la flexion d’une poutre en masse-tenseur avec un modèle mixte au niveau de la loi de comportement et du type d’éléments.

4.6 Adaptation dynamique selon critères

Ensuite, l’idée est de modifier les objets au cours de leur simulation en fonction de critères mis en place pour jouer sur le compromis entre temps de calcul et précision des résultats. Nous pouvons ainsi changer la loi de comportement des éléments (loi de Hooke, Saint Venant-Kirchhoff, etc.), leur géométrie (quadrangle, triangle, etc.), leur raffinement (par l’ajout ou la suppression de nœuds à l’élément) ou encore les fonctions de forme employées pour chacun des éléments. La Fig. 4.9 et l’algorithme 8 résument l’approche que nous avons mis en place (Caillaud et al. (2013)).

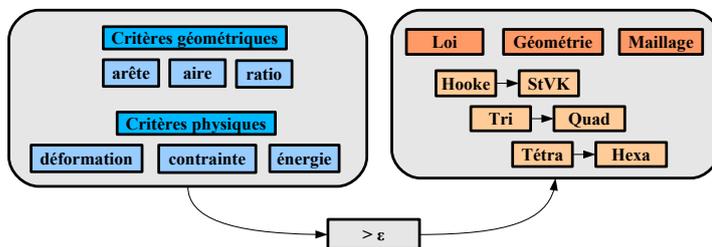


Figure 4.9 – Illustration de notre approche pour dynamiquement adapter la simulation en fonction de critères géométriques ou mécaniques.

La Fig. 4.10 illustre une stratégie consistant à utiliser le modèle linéaire de Hooke dans les limites de sa validité, puis d’utiliser une loi non-linéaire quand nous passons en grands déplacements (au delà de 10%) ou en rotation (au delà de 5°).

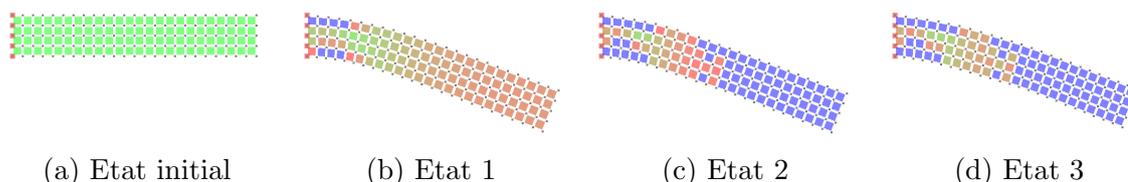


Figure 4.10 – Illustration 2D. Simulation d’une poutre en flexion. Les quadrangles dans la gamme vert-rouge correspondent au max $(\mathcal{R}_L, \mathcal{R}_A) \in [0, 1]$ et les bleus correspondent au passage des éléments initialement modélisés par la loi de Hooke (en vert) à la loi de Saint Venant-Kirchhoff.

Algorithme 8 : Adaptation de la simulation en fonction de l'évaluation de critères.

```

1: // lc ← liste des critères
2: Pour chaque élément e faire
3:   p ← position des noeuds des éléments
4:   u ← déplacement des noeuds des éléments
5:   f ← force des noeuds des éléments

6:   Pour chaque critère c Faire
7:     c.eval(e,p,u,f)
8:   FinPour

9:   Pour chaque stratégie s Faire
10:    Si s.actionNeeded(e,lc) Alors
11:      s.action(e)
12:    FinSi
13:   FinPour
14: FinPour

```

Critères géométriques de validité. Deux critères géométriques peuvent être utilisés pour détecter ces limites de validité :

— R_L permet de considérer l'allongement des arêtes de chacun des éléments avec

$$R_L = \max_{i \in [0, \dots, n_e]} \frac{|l_0^i - l_t^i|}{l_0^i},$$

où n_e est le nombre d'arêtes de l'élément considéré et l_0^i, l_t^i la longueur initiale et courante de l'arête i .

— R_A permet de considérer l'aire de chacun des éléments avec

$$R_A = \frac{|S_0 - S_t|}{S_0},$$

où S_0, S_t sont l'aire initiale et courante de la surface de l'élément considéré.

Critères géométriques de qualité. D'autres critères géométriques peuvent être employés pour mesurer la qualité des éléments et ainsi agir en raffinant les éléments :

— R_M permet de considérer la moyenne des longueurs \bar{l} des arêtes de l'élément avec

$$R_M = \frac{S_t}{\bar{l}^2}.$$

— R_{HM} est la moyenne harmonique des longueurs des arêtes de l'élément définie par

$$R_{HM} = 6\sqrt{2} S_t \frac{H}{\bar{l}^3} \text{ avec } H = \frac{1}{n_e} \sum_{i=1}^{n_e} l_i^{-1}.$$

Critères physiques. Nous pouvons également considérer des critères physiques comme le critère R_S relié à la contrainte appliquée à l'instant t sur chacun des élément de surface $A(t)$. Ce critère est ainsi défini par

$$R_S = \frac{1}{A(t)} \int_E \mathbf{F}_E(x, y) dx dy.$$

avec $\mathbf{F}_E(x, y)$ la force appliquée sur l'élément 2D E considéré. Ce critère doit alors être discrétisé avec

$$R_S = \frac{1}{A_t} \sum_{i=0}^{n_g-1} w_i \mathbf{F}_E(g_i)$$

où w_i sont les poids associés aux points de Gauss g_i . Mais comme nous ne connaissons pas les forces appliquées en ces points, nous les interpolons en fonction des forces appliquées aux sommets avec

$$R_S = \frac{1}{A(t)} \sum_{i=0}^{n_g-1} w_i \sum_{j=0}^{n-1} \|\Lambda_j(g_i) \mathbf{F}_E(P_j)\|$$

avec $F_E(P_j)$ la force appliquée au sommet P_j et Λ_j les fonctions d'interpolations utilisées. Nous pouvons noter que la majorité des termes employés sont déjà calculés au cours de la simulation. La Fig. 4.11 illustre les résultats pour une poutre fixée soumise à la gravité.

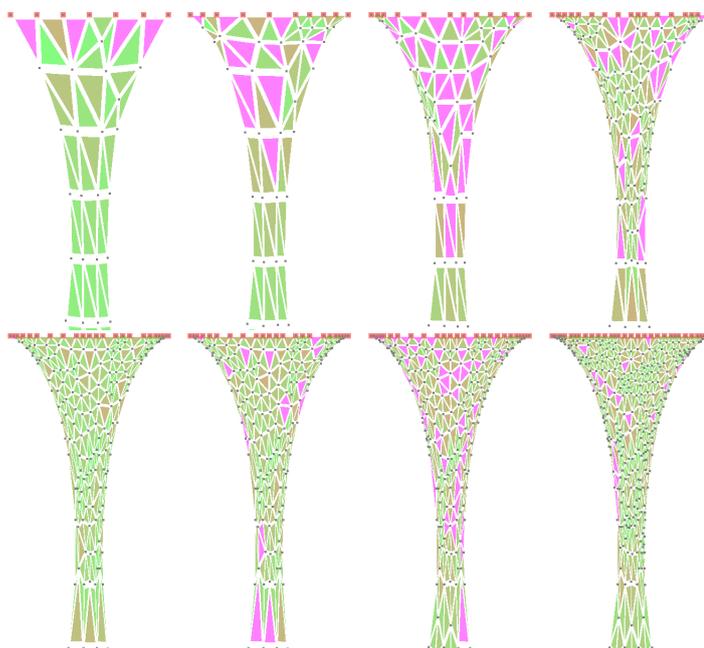


Figure 4.11 – Illustration 2D. Poutre fixée soumise à la gravité. Raffinements successifs selon le critère R_S . A chaque pas de temps, les triangles dépassant 90 % de $\max(R_S)$ sont subdivisés.

4.7 Évaluation de notre approche

Évaluation en performance. La Fig. 4.12 présente le coût du calcul formel par rapport à l'approche employée par [Picinbono \(2003\)](#). Comme notre méthode débute par une étape qui extrait les données constantes pour les calculs avant la simulation, notre approche est d'autant plus efficace que les équations employées sont compliquées. Par ailleurs, le compilateur est plus efficace pour de grandes expressions que pour des boucles imbriquées. De ce fait, notre approche n'est pas efficace lors de l'emploi de la loi de comportement linéaire de Hooke sur CPU, puisque les équations sont trop simples dans ce cas. C'est pourquoi, nous obtenons des temps de simulation moins efficaces d'un facteur 3 par rapport à une approche usuelle. Par contre, lors de l'emploi de la loi de Saint Venant-Kirchhoff, nous pouvons obtenir un facteur 15 d'efficacité par rapport à l'approche classique.

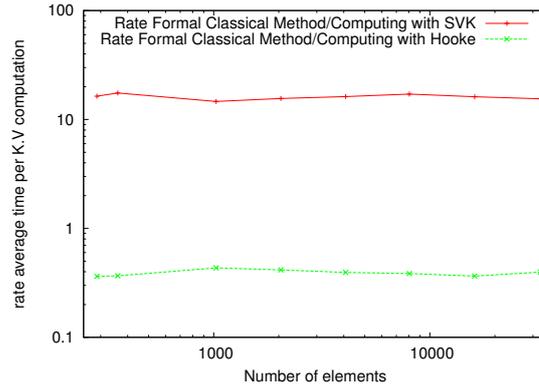


Figure 4.12 – Coût du calcul symbolique par rapport à l'approche employée par Picinbono (2003).

La Fig. 4.13 présente le temps moyen, pour un pas de temps, pour une poutre triangulaire de taille 1.2×0.2 m et une poutre tétraédrique de taille $1.2 \times 0.2 \times 0.2$ m en considérant la loi de Hooke ou de Saint Venant-Kirchhoff. La densité est $\rho = 800 \text{ Kg.m}^{-2}$ ou Kg.m^{-3} et le module de Young est $E = 100 \text{ MPa}$. Naturellement, le temps moyen pour un maillage 3D est plus important que celui pour un maillage 2D. Par ailleurs, l'emploi de la loi de Saint Venant-Kirchhoff requière plus de temps de calcul que pour l'emploi de la loi de Hooke. Par contre, nous pouvons voir que l'emploi du GPU est efficace dans le cas de grandes expressions résultant à une différence faible entre les deux simulations.

La Fig. 4.14 donne les temps de calcul moyen obtenu pour différents types d'éléments. Ces temps de calcul sont très similaires pour les pyramides, prismes et hexaèdres. Les tétraèdres restent beaucoup plus rapides. A noter que les prismes requièrent naturellement plus de temps avec l'emploi de fonctions de forme de degrés 2.

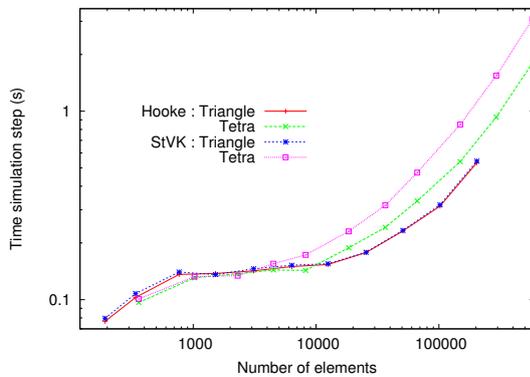


Figure 4.13 – Temps moyen pour un pas de temps en considérant une poutre triangulaire ou tétraédrique avec la loi de Hooke ou de Saint Venant-Kirchhoff (StVK).

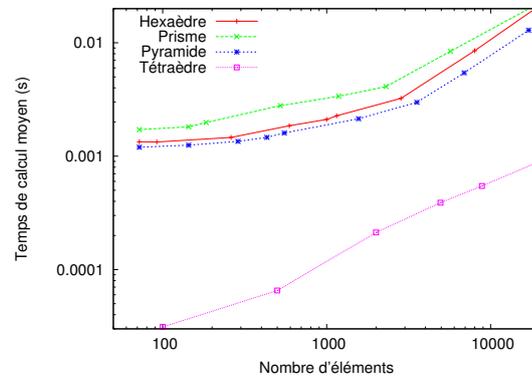


Figure 4.14 – Temps de calcul moyen pour le calcul des dérivées partielles des forces internes en fonction du type d'élément (hexaèdre, prisme, pyramide, tétraèdre).

Évaluation en précision. La Fig. 4.15 présente les résultats obtenus pour la simulation d'une poutre de taille $1200 \times 200 \times 200$ mm constituée d'hexaèdres avec une échelle logarithmique. Nous faisons varier la résolution, c'est-à-dire le nombre d'éléments de 1000 à 100000. Nos résultats sont comparés à ceux obtenus en utilisant le logiciel d'éléments finis Abaqus en considérant 10000 tétraèdres et la loi de comportement de Mooney-Rivlin. Nous obtenons une erreur d'environ 3 mm pour chacun des nœuds avec 100000 éléments.

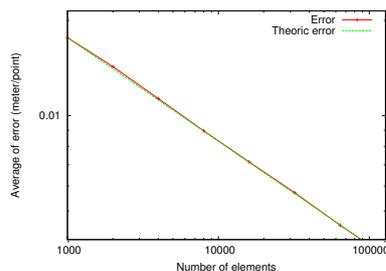


Figure 4.15 – Erreur moyenne en fonction de la résolution.

La Fig. 4.16 présente les résultats pour la simulation statique d’une poutre encastree en utilisant un maillage triangulaire avec la loi de Saint Venant-Kirchhof en considérant des triangles à 3 ou 6 noeuds. La courbe grise, que nous appellerons ”gold standard” correspond à l’équation analytique. Nous avons fait varier le nombre d’éléments de 100 à 3200 triangles (pour les triangles à 3 noeuds) ; et de 100 à 200 pour les triangles à 6 noeuds. Alors que nous multiplions la résolution par 16000, nous pouvons observer qu’un maillage utilisant des triangles à 3 noeuds reste moins précis qu’un maillage avec des triangles à 6 noeuds. Par ailleurs, au niveau du temps de calcul, la simulation avec 200 éléments de type triangle à 6 noeuds s’exécute en 10 frames par seconde, alors que la simulation avec 3200 éléments de type triangles à 3 noeuds s’exécute en 4 frames par seconde.

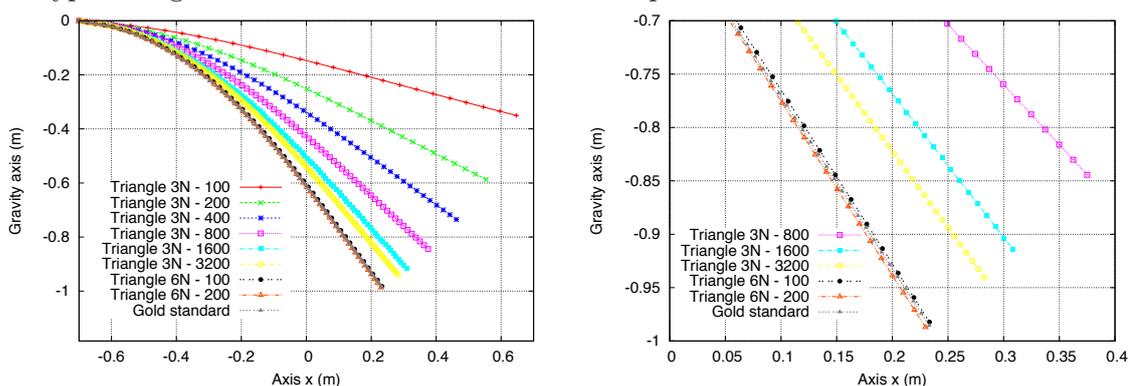


Figure 4.16 – Courbe relative à la déformation d’une poutre : (gauche) résultats obtenus avec différentes résolutions et types d’éléments (triangle à 3 ou 6 noeuds) ; (droite) zoom sur l’extrémité.

La Fig. 4.17 présente les résultats obtenus en utilisant d’autres types d’éléments (hexaèdre, tétraèdre, prisme, pyramide) avec la loi de Saint Venant-Kirchhof. Une comparaison est faite avec les résultats obtenus en utilisant le progiciel Abaqus et l’équation analytique correspondante. Les courbes de droite présentent les résultats sur une zone plus restreinte en se focalisant sur l’extrémité de la poutre. Le Tableau 4.2 récapitule l’ensemble des écarts entre la solution analytique et les différentes flèches simulées. L’hexaèdre est l’élément le plus précis. Ensuite, le prisme reste très précis, suivi par la pyramide. Le tétraèdre est le moins précis du fait de sa fonction d’interpolation de degré 1.

	Nombre d’éléments	Hexaèdre	Prisme	Pyramide	Tétraèdre
Notre approche	3072	0.0007906	0.0012905	0.0053035	0.0446796
	6000	0.0006380	0.0012286	0.0043081	0.0323418
	10800	0.0005798	0.0011668	0.0033639	0.0202163
	16644	0.0004615	0.0010429	0.0032380	0.0105572
Abaqus	16644	0.0005081	0.0010618	-	0.0159292

Table 4.2 – Écarts moyens en mètre entre les flèches obtenues et la solution analytique.

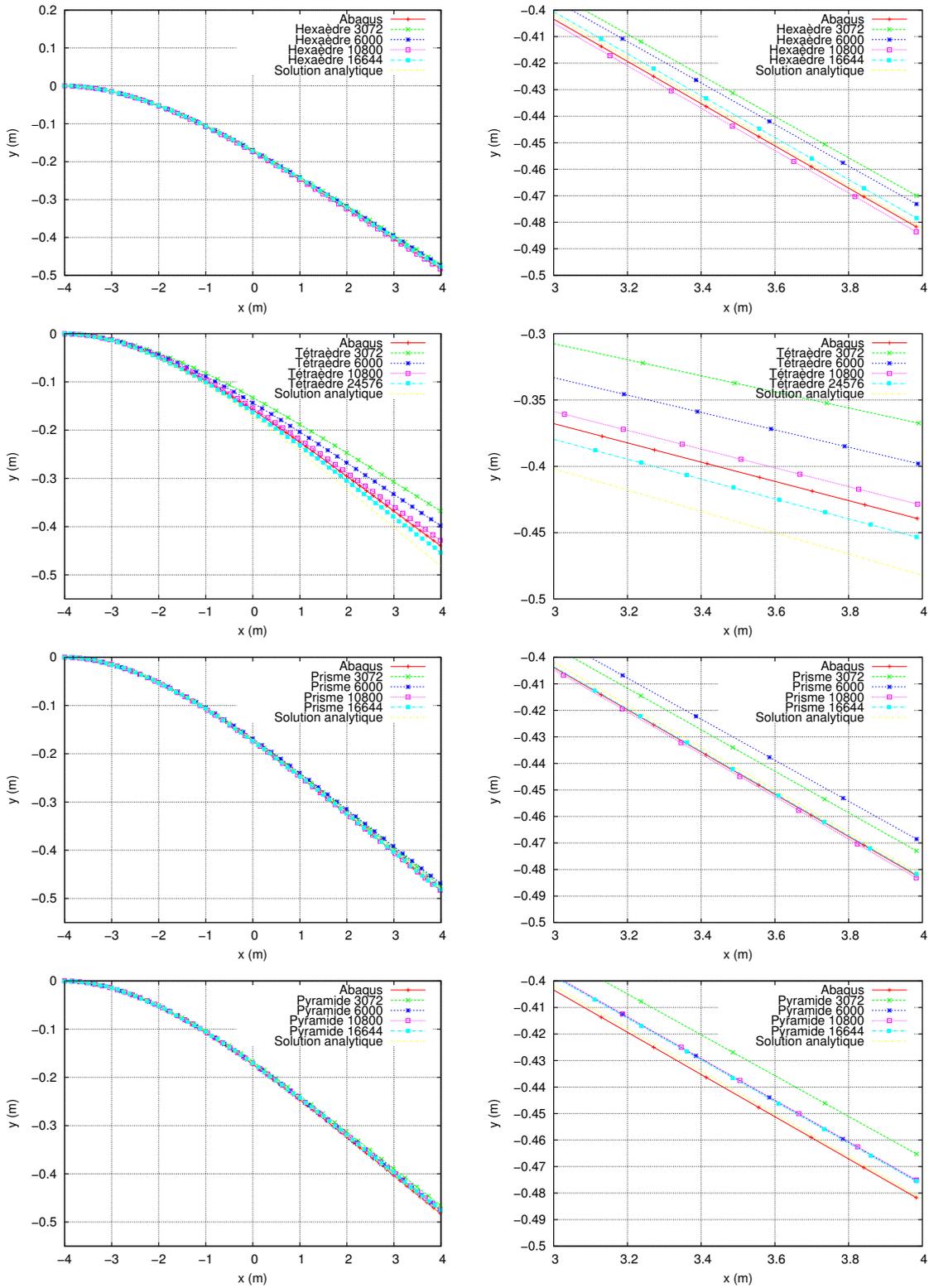


Figure 4.17 – Courbe relative à la déformation d'une poutre constituée d'éléments de différents types : (gauche) résultats en comparaison avec la solution obtenue en utilisant le progiciel Abaqus à la plus grande résolution et la solution analytique; (droite) zoom sur l'extrémité de la poutre considérée.

4.8 Discussion

Au final, nous avons mis en place une chaîne de traitement qui permet d'effectuer des simulations d'objets mixtes au niveau de leurs éléments, c'est-à-dire des objets qui combinent des éléments de lois de comportement et de géométries différentes. De plus, nous avons fait en sorte de pouvoir faire évoluer ces éléments au cours de la simulation pour répondre à des critères en changeant leur géométrie ou loi de comportement.

- Nous avons adopté une approche qui emploie directement durant la simulation la formulation explicite du calcul des forces et de leurs dérivées, c'est-à-dire que les équations sont totalement déroulées, et à chaque pas de temps celles-ci sont simplement évaluées en fonction de l'état courant des nœuds. Il reste à faire une phase de validation plus approfondie de cette approche pour évaluer son coût en temps de calcul par rapport à une approche plus traditionnelle où les équations sont résolues à chaque pas de temps de la simulation. Pour cela, d'autres lois de comportement doivent également être prises en compte puisque l'objectif réside notamment dans le fait de faciliter l'emploi de lois de comportement dont la formulation peut s'avérer compliquée de prime abord. Il s'agira de voir si la génération des équations posent problème notamment en terme de gestion de fichiers et de compilation.
- L'enjeu suivant réside dans la définition de critères pertinents pour effectuer ces changements à bon escient. En effet, la précision des calculs n'est pas obligatoirement améliorée par le raffinement des mailles de l'objet si ces mailles ne sont pas de bonne qualité pour la simulation. Ou encore, il nous avons vu qu'il peut être opportun de débiter une simulation en utilisant la loi de Hooke, et de changer de loi de comportement dès que nous passons en grande déformation. C'est pourquoi des critères spécifiques doivent être élaborés dans le cadre d'une simulation physique afin de proposer un modèle géométrique et mécanique adapté à la simulation que nous souhaitons effectuer et qui évolue au cours du temps. L'idée est toujours de trouver le compromis adéquat entre la précision des résultats et le temps de calculs nécessaires pour cela en fonction de l'application visée. Nous avons vu que ces critères pouvaient être géométriques (qualité des mailles), mécaniques (état de la déformation des mailles), ou encore relatifs à la convergence des méthodes numériques employées.
 - En terme de géométrie, il s'agit de créer le maillage optimal (en fonction du compromis retenu entre le temps de calcul et la précision souhaitée) par rapport à la simulation physique effectuée. Il s'agit ainsi de définir des métriques donnant la qualité des mailles de l'objet et de proposer les opérations associées afin de les améliorer en fonction du besoin. Il s'agit aussi de déterminer les zones dans lesquelles un raffinement doit être effectué : dans une zone de contact avec un autre objet, dans une zone complexe nécessitant plus de précision, etc.
 - En terme de mécanique, il s'agit de trouver la modélisation qui permet de s'approcher le plus possible de la solution optimale connue tout en maîtrisant les erreurs effectuées pour diminuer par exemple le coût en calculs (choix du modèle physique, de la loi de comportement, des méthodes d'intégration numérique, du pas de temps des méthodes numériques, etc.).
- Il faut également mettre en place la gestion des changements de topologie au cours de la simulation, en intégrant notamment des schémas de raffinement ou dé-raffinement. Ces travaux ont déjà été abordés par des membres de mon équipe de recherche (Fabrice Jaillet, Vincent Vidal). Il s'agira de les intégrer au sein de notre librairie.
- Il nous reste encore à évaluer en terme de temps, la version sur GPU des algorithmes d'adaptation dynamique des éléments en fonction des critères mis en place.

Contributions

Publications

- X. Faure. *Approche formelle pour la simulation interactive de modèles mixtes*. Thèse de doctorat en informatique, Université Lyon 1, Sept. 2014.
- X. Faure. *Implémentation de la Méthode des Masse-Tenseurs dans la librairie de simulation SOFA*. Technical Report, LIRIS UMR 5205, Mar. 2014.
- Alphonse, G. and Ardail, D. and Armandy, E. and Bajard, M. and Balosso, J. and Baron, M.-H. and Battiston-Montagne, P. and Bertrand, G. and Beuve, M. and Biston, Marie-Claude and Blanchard, G. and Bouhadiba, O. and Boutin, B. and Cajavec-Bernard, B. and Chandez, F. and Chanrion, M.-A. and Chollier, L. and Constanzo, J. and Cunha, M. and Dabli, Djamel and Dahoumane, M. and Dauvergne, D. and De Rydt, M. and Dedes, G. and Deng, S. and Ducimetière, F. and Fallavier, M. and Fayet, Y. and Ferrandon, S. and Force, P. and Freud, N. and Gagnon, C. and Ganne, C. and Gautier, M. and Gilormini, M. and Glatard, T. and Hu, Y. and Insa, C. and Jaillet, F. and Joly, Baptiste and Krimmer, J. and Ladjal, Hamid and Lambert, D. and Lestand, Loïc and Letang, J.M. and Ley, J.-L. and Magné, N. and Malesys, C. and Martin, F. and Mathez, H. and Maxim, Voichita and Moncharmont, C. and Montarou, Gerard and Moreau, J.-M. and Pequegnot, L. and Patin, S. and Pauna, N. and Pinto, M. and Pommier, P. and Poncet, D. and Pop-Camarasu, S. and Prost, R. and Ray, C. and Ray-Coquard, Isabelle and Reithinger, V. and Remillieux, J. and Ribba, S. and Richard, M.-H. and Rit, S. and Rodriguez-Lafrasse, C. and Roellinghoff, F. and Royer, L. and Sarrut, D. and Shariat, B. and Testa, E. and Vert, P.-E. and Vial, L. and Vogin, G. and Zara, F. and Zoccarato, Y. *2012 Activity Report of the Regional Research Programme on Hadrontherapy for the ETOILE Center*, Research Report, Centre Etoile ; Université Lyon 1 - Claude Bernard, Jun. 2013.
- F. Caillaud, X. Faure, F. Zara, F. Jaillet, and J.-M. Moreau. *Multi-criteria adaptation of physical simulations*, Eurographics Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS 2013), « Work in Progress » session, Nov. 2013.
- X. Faure, F. Zara, F. Jaillet, and J.-M. Moreau. *An Implicit Tensor-Mass solver on the GPU for soft bodies simulation*. Eurographics Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS), pages 1–10, Dec. 2012.
- X. Faure, F. Zara, F. Jaillet, and J.-M. Moreau. *Implicit Tensor-Mass solver on the GPU*, Symposium on Computer Animation (SCA'2012), July 2012.

Logiciels

- Outil générant les équations relatives aux calculs des forces et de leurs dérivées.
- Outil permettant de réaliser des *benchmarks* de simulation.
- **Plugin dans la librairie SOFA** permettant de réaliser une simulation physique avec l'approche masse-tenseur. Possibilité de modifier les éléments constituant l'objet simulé au cours de la simulation. Parallélisation en utilisant OpenCL.

Bibliographie

- J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni. SOFA - an open source framework for medical simulation. In *Medicine Meets Virtual Reality (MMVR)*, volume 15, pages 1–6, Long Beach, California, Etats-Unis, February 2007.
- G. Alphonse, D. Ardail, E. Armandy, M. Bajard, J. Balosso, M.-H. Baron, P. Battiston-Montagne, G. Bertrand, M. Beuve, M.-C. Biston, G. Blanchard, O. Bouhadiba, B. Boutin, B. Cajavec-Bernard, F. Chandez, M.-A. Chanrion, L. Chollier, J. Constanzo, M. Cunha, D. Dabli, M. Dahoumane, D. Dauvergne, M. De Rydt, G. Dedes, S. Deng, F. Ducimetière, M. Fallavier, Y. Fayet, S. Ferrandon, P. Force, N. Freud, C. Gagnon, C. Ganne, M. Gautier, M. Gilormini, T. Glatard, Y. Hu, C. Insa, F. Jaillet, B. Joly, J. Krimmer, H. Ladjal, D. Lambert, L. Lestand, J. Letang, J.-L. Ley, N. Magné, C. Malesys, F. Martin, H. Mathez, V. Maxim, C. Moncharmont, G. Montarou, J.-M. Moreau, L. Pequegnot, S. Patin, N. Pauna, M. Pinto, P. Pommier, D. Poncet, S. Pop-Camarasu, R. Prost, C. Ray, I. Ray-Coquard, V. Reithinger, J. Remillieux, S. Ribba, M.-H. Richard, S. Rit, C. Rodriguez-Lafresse, F. Roellinghoff, L. Royer, D. Sarrut, B. Shariat, E. Testa, P.-E. Vert, L. Vial, G. Vogin, F. Zara, and Y. Zoccarato. 2012 Activity Report of the Regional Research Programme on Hadrontherapy for the ETOILE Center. Research report, Centre Etoile; Université Lyon 1 - Claude Bernard, June 2013. URL <https://hal.archives-ouvertes.fr/hal-00840163>.
- F. Caillaud, X. Faure, F. Zara, F. Jaillet, and J.-M. Moreau. Multi-criteria adaptation of physical simulations, Nov. 2013. URL <http://liris.cnrs.fr/publis/?id=6347>. Eurographics Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS 2013), "Work in Progress" session.
- S. Cotin. *Real-time anatomical deformable models : Application to laparoscopic surgery with force feedback*. Theses, Université Nice Sophia Antipolis, Nov. 1997. URL <https://tel.archives-ouvertes.fr/tel-00630956>.
- S. Cotin, H. Delingette, and N. Ayache. A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation. *The Visual Computer*, 16(8) :437–452, 2000a. ISSN 0178-2789.
- S. Cotin, H. Delingette, and N. Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. *The Visual Computer*, 16(8) :437–452, 2000b. URL <http://www.inria.fr/sophia/asclepios/Publications/Herve.Delingette/TensorMass-VC2000.pdf>.

- H. Delingette. *Modélisation de structures déformables*. Habilitation à diriger des recherches, Université Nice Sophia-Antipolis, March 2006. URL <http://www.inria.fr/sophia/asclepios/Publications/Herve.Delingette/hdr.pdf>.
- X. Faure. Implémentation de la Méthode des Masse-Tenseurs dans la librairie de simulation SOFA. Technical Report RR-LIRIS-2014-003, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon, Mar. 2014a. URL <https://liris.cnrs.fr/Documents/Liris-6529.pdf>.
- X. Faure. *Approche formelle pour la simulation interactive de modèles mixtes*. Thèse de doctorat en informatique, Université Lyon 1, Sept. 2014b. URL <http://liris.cnrs.fr/publis/?id=7026>.
- X. Faure, F. Jaillet, F. Zara, and J.-M. Moreau. Acquisition multi-modale en temps réel pour le suivi du mouvement respiratoire. In *ORASIS 2011*, June 2011. URL <http://liris.cnrs.fr/publis/?id=5013>.
- X. Faure, F. Zara, F. Jaillet, and J.-M. Moreau. Implicit Tensor-Mass solver on the GPU, July 2012a. URL <http://liris.cnrs.fr/publis/?id=5652>. ACM/Eurographics Symposium on Computer Animation (SCA'2012).
- X. Faure, F. Zara, F. Jaillet, and J.-M. Moreau. An Implicit Tensor-Mass solver on the GPU for soft bodies simulation. In *Eurographics Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS)*, pages 1–10, Dec. 2012b. URL <http://liris.cnrs.fr/publis/?id=5783>.
- F. Goulette and Z.-W. Chen. Fast computation of soft tissue deformations in real-time simulation with hyper-elastic mass links. *Computer Methods in Applied Mechanics and Engineering*, 295 :18 – 38, 2015. ISSN 0045-7825. doi : <http://dx.doi.org/10.1016/j.cma.2015.06.015>. URL <http://www.sciencedirect.com/science/article/pii/S0045782515002078>.
- C. Lobos, Y. Payan, and N. Hitschfeld. Techniques for the generation of 3d finite element meshes of human organs. *arXiv preprint arXiv :0911.3884*, 2009.
- C. Lobos et al. A set of mixed-elements patterns for domain boundary approximation in hexahedral meshes. *Studies in health technology and informatics*, 184 :268–272, 2012.
- G. Picinbono. *Geometrical and physical models for surgery simulation*. Theses, Université Nice Sophia Antipolis, Feb. 2001. URL <https://tel.archives-ouvertes.fr/tel-00633965>.
- G. Picinbono. Non-linear anisotropic elasticity for real-time surgery simulation. *Graphical Models*, 65(5) :305–321, Sept. 2003. ISSN 15240703. doi : 10.1016/S1524-0703(03)00045-6.
- G. Picinbono, H. Delingette, and N. Ayache. Real-Time Large Displacement Elasticity for Surgery Simulation : Non-linear Tensor-Mass Model. In *Proceedings of MICCAI'00*, pages 643–652, London, UK, 2000. Springer-Verlag. ISBN 3-540-41189-5.
- G. Picinbono, H. Delingette, and N. Ayache. Nonlinear and anisotropic elastic soft tissue models for medical simulation. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1370–1375. IEEE, 2001.

- J. Schwartz, M. Denninger, D. Rancourt, C. Moisan, and D. Laurendeau. Modelling liver tissue properties using a non-linear visco-elastic model for surgery simulation. *Medical Image Analysis*, 9(2) :103–112, 2005.
- J.-M. Schwartz. *Calcul rapide de forces et de déformations mécaniques non-linéaires et visco-élastiques pour la simulation de chirurgie*. PhD thesis, Université Laval, Québec, Oct. 2003. URL <http://hdl.handle.net/20.500.11794/17855>.

Apport d'un modèle topologique pour la simulation physique

Je présente dans ce chapitre les travaux qui ont démarré dans le cadre de la thèse de Elsa [Flechon \(2014\)](#) financée par une bourse ministérielle de 2011 à 2014. L'objectif était de pouvoir définir facilement des objets ayant des éléments de type différent aussi bien du point de vue mécanique que topologique. Dans la continuité des travaux réalisés durant la thèse de Xavier Faure, nous souhaitions également pouvoir modifier ces éléments au cours de la simulation, mais l'idée première durant la thèse de Elsa Fléchon était de permettre d'effectuer facilement des modifications au niveau de la topologie de l'objet pour permettre des opérations de découpe ou de raffinement/dé-raffinement du maillage volumique décrivant l'objet. Dans ce contexte, notre contribution concerne la conception d'une structure de données topologique adaptée à la simulation physique permettant une définition générique des objets en éléments de types différents ainsi que leur adaptation au cours de la simulation. Ces travaux ont été financés dans le cadre du projet transversal TopoSim (<http://liris.cnrs.fr/toposim>) du LIRIS élaboré entre l'équipe SAARA et l'équipe M2Disco. Ce projet m'a permis d'amorcer une collaboration active avec Guillaume Damiand qui a pris part à la supervision de Elsa Fléchon dès sa seconde année de thèse en apportant son expertise du monde de la topologie.

Sommaire

5.1	Contexte de ce travail de recherche	140
5.2	Notions préliminaires en topologie	141
5.3	Changements topologiques durant la simulation	144
5.4	Modèle unifié combinant topologie et physique	146
5.5	Discussion	156

5.1 Contexte de ce travail de recherche

Il est intéressant de pouvoir modifier dynamiquement la topologie d'un objet. Ces changements peuvent par exemple survenir dans le cadre d'une simulation médicale lors de l'interaction de l'organe simulé avec un outil de type scalpel où le changement topologique désiré concerne la découpe de l'objet. Ces changements topologiques peuvent également être bénéfiques quand nous souhaitons raffiner ou dé-raffiner des parties spécifiques du modèle géométrique de l'objet en fonction du besoin. En effet, il peut être pertinent de raffiner les mailles de l'objet (par exemple présentes dans des zones de contacts avec d'autres objets) pour obtenir une plus grande précision des calculs, alors que d'autres zones peuvent conserver des mailles plus grossières pour une même précision de calculs. A noter que malgré l'intérêt de pouvoir faire ces changements topologiques au cours de la simulation, peu de méthodes s'attaquent à cette problématique dans le cadre de maillages volumiques. C'est pourquoi nous nous sommes attelés à cet objectif avec toujours la même idée sous-jacente en ligne de mire : "Comment gagner en temps de calcul pour proposer des simulations interactives, tout en conservant une précision adéquate à la conception de simulateurs d'apprentissage de gestes médicaux?"

Notre travail de recherche sur cette problématique a débuté durant la thèse de Elsa Fléchon avec la mise en place d'une collaboration interne au LIRIS avec Guillaume Damiand spécialiste en topologie. Cette collaboration a été officialisée au sein du projet transversal TopoSim unissant l'équipe SAARA et l'équipe M2Disco, pour étudier les apports de l'approche de la topologie dans la simulation par modèles physiques. Dans ce contexte, nous avons proposé un modèle unifié permettant de gérer à la fois la topologie de l'objet et sa modélisation physique. L'emploi de ce modèle permet notamment des changements topologiques des objets simulés au cours de leur simulation.

Nous avons ainsi dès le départ fait le choix d'employer un modèle topologique afin d'assurer la qualité du résultat obtenu et sa validité topologique. Nous avons débuté notre étude en employant comme modèle physique, le modèle discret des masses-ressorts (avec le formalisme de [Baudet et al. \(2009\)](#)), et comme structure topologique, les complexes linéaires cellulaires basés sur les cartes combinatoires généralisables en n dimension. L'idée était de débiter avec un modèle physique simple pour vérifier la faisabilité de notre approche. Nous avons ensuite étendu notre modèle en intégrant le modèle physique des masse-tenseurs. En effet, notre souhait est de proposer une solution générique, permettant de s'étendre à toute dimension, à toute topologie et à tout modèle physique.

Ce travail de recherche a permis de réunir les résultats obtenus durant la thèse de Vincent Baudet (dont j'avais pris part à son encadrement dès mon arrivée au LIRIS sur l'intégration des paramètres mécaniques de l'objet simulé dans le calcul des paramètres physiques du système masses-ressorts), de Xavier Faure (sur la génération automatique des équations relatives au calcul des forces dans le modèle masse-tenseur) et de Elsa Fléchon (sur l'apport de la topologie dans le monde de la simulation physique).

Organisation de ce chapitre : Ce chapitre se poursuit dans la section 5.2 par la présentation du vocabulaire et des notions de base en topologie permettant d'aborder sereinement la suite du chapitre. La section 5.3 dresse un état de l'art des solutions proposées dans la littérature pour effectuer des changements topologiques sur les objets animés. La section 5.4 présente ensuite notre modèle unifié avec l'opération topologique de la découpe et celle du raffinement du maillage de l'objet simulé, ainsi que le calcul générique des forces appliquées sur les objets simulés. Enfin la section 5.5 tire un bilan de notre approche.

5.2 Notions préliminaires en topologie

Je présente dans cette section les notions et terminologies employées dans le domaine de la topologie. J'expliciterai également le modèle topologique des cartes combinatoires sur lequel nos travaux se sont basés ainsi que les opérations topologiques qui lui sont associées. L'idée est de comprendre le monde de la topologie quand on vient du monde de l'animation par modèle physique afin d'en tirer tous les avantages.

Une structure de données topologique décompose un objet en **cellules** et définit les **relations d'adjacence et d'incidence** entre ces cellules.

- Une cellule de dimension i est appelée **i -cellule**. Ainsi une 0-cellule correspond à un sommet, une 1-cellule correspond à une arête, une 2-cellule correspond à une face et une 3-cellule correspond à un volume en 3D.
- Une cellule de dimension i (avec $i \geq 1$) est **bornée** par des cellules de dimension $i - 1$. Et ces cellules constituent sa **frontière**. Par exemple, une face (2-cellule) est bornée par des arêtes (1-cellules).
- Deux cellules sont dites **incidentes** si une cellule appartient à la frontière de l'autre. Par exemple, les arêtes d'une face (1-cellules) sont incidentes à la face considérée (2-cellule).
- Si deux cellules sont de même dimension et sont incidentes à une même cellule, nous diront que ces cellules sont **adjacentes**. Par exemple, deux faces ayant une arête en commun sont adjacentes.

En terme de structure de données topologique, une **arête ailée** est une arête orientée ayant huit pointeurs sur les cellules l'entourant. Ainsi chaque arête ailée pointe sur ses deux sommets incidents, sur ses deux faces incidentes et sur les quatre arêtes précédentes et suivantes appartenant à la bordure de ses deux faces incidentes.

Une arête ailée orientée peut être scindée en deux **demi-arêtes** orientées et opposées. Ainsi chaque demi-arête est associée uniquement à un sommet incident et une face incidente et possède seulement quatre pointeurs. Ainsi une demi-arête d pointe sur son sommet incident, sur la demi-arête précédente et suivante appartenant à la même face que d et sur la demi-arête appartenant à la même arête que d mais pas à la même face.

De manière formelle, une **structure de données des demi-arêtes** \mathcal{S} est un ensemble de demi-arêtes \mathcal{A} , d'une involution α et d'une permutation σ . Elle est ainsi notée $\mathcal{S} = (\mathcal{A}, \alpha, \sigma)$. L'involution α permet à partir d'une demi-arête de récupérer son opposé appartenant à la même arête et au même volume mais pas à la même face. En appliquant deux fois α sur une demi-arête a , nous retombons sur a . La permutation σ permet de donner le successeur d'une demi-arête autour d'un sommet. A partir d'une demi-arête a , σ doit être appliquée plusieurs fois pour retomber sur a .

La Fig. 5.1¹ présente un exemple de l'emploi de ces deux structures de données pour un objet 2D constitué de 4 faces notées F_0, F_1, F_2 et F_3 et de 7 sommets notés de A à G . La Fig. 5.1 (a) présente le cas des arêtes ailées où l'objet a 9 arêtes ailées notées de a à i . L'arête ailée b (en rouge) est représentée avec l'ensemble de ses pointeurs. Les flèches

1. Pour illustrer les propos de ce chapitre, j'utilise notamment les schémas réalisés par Elsa Fléchon pour son manuscrit de thèse.

vertes représentent ses pointeurs sur ses 2 sommets incidents B et C , les flèches violettes correspondent à ses pointeurs sur ses 2 faces incidentes F_0 et F_2 et les flèches fuchsias sont ses pointeurs sur les 4 arêtes ailées précédentes et suivantes de F_0 et F_2 .

Dans le cas des demi-arêtes de la Fig. 5.1 (b), l'objet dispose de 18 demi-arêtes notées de a à r . La demi-arête b (en rouge) est représentée avec l'ensemble de ses pointeurs. La flèche verte représente son pointeur sur son sommet incident C , la flèche violette correspond à son pointeur sur sa face incidente F_0 et les flèches fuchsias sont ses pointeurs sur les 2 demi-arêtes précédentes et suivantes incidentes à F_0 . Les liens bleus relient 2 demi-arêtes opposées.

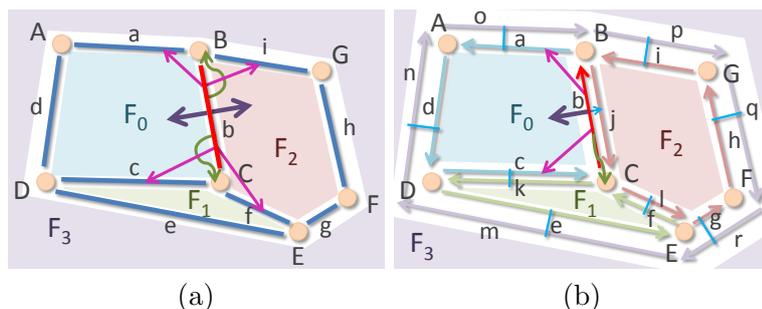


Figure 5.1 – Emploi de la structure de données de l'arête ailée (a) ou des demi-arêtes (b).

Par ailleurs, le terme de **plongement** est employé quand une géométrie est associée à une représentation topologique.

Les **cartes combinatoires** permettent une représentation générique des objets de n dimensions qui peuvent être subdivisés en cellules de topologies et de dimensions quelconques. Ce modèle correspond à une généralisation de la structure de données des demi-arêtes en toute dimension. En effet, la définition formelle faite pour la structure de données des demi-arêtes correspond à celle des cartes combinatoires 2D.

Une carte combinatoire est définie à partir d'un élément basique appelé **brin** et d'un ensemble de **pointeurs** entre ces brins. Un brin correspond à une partie d'une arête orientée (1-cellule) faisant partie des cellules incidentes de dimension i avec $i \in \{0, 2, 3, \dots, n\}$. Les pointeurs définis entre les brins permettent de définir toutes les relations d'incidence et d'adjacence entre les cellules de différentes dimensions. Ainsi la relation d'adjacence entre les arêtes est représentée par une permutation permettant d'obtenir le brin suivant d'un brin donné appartenant à la même 2-cellule. Cette permutation est notée β_1 mettant en relation des arêtes de dimension 1. La relation d'adjacence entre les faces est représentée par une involution notée β_2 mettant en relation des cellules de dimension 2. Pour une dimension n quelconque de la carte combinatoire, la définition est étendue en définissant les β_i (avec $i \in \{0, \dots, n\}$).

Définition. Une n -carte combinatoire (carte combinatoire de dimension n avec $n \geq 0$) est une algèbre $\mathcal{C} = (B, \beta_1, \dots, \beta_n)$ où (Damiand (2012); Lienhardt (1991)) :

1. B est un ensemble fini de brins ;
2. β_1 est une permutation sur B ;
3. $\forall i : 2 \leq i \leq n : \beta_i$ est une involution sur B ;
4. $\forall i, j : 1 \leq i \leq i + 2 \leq j \leq n : \beta_i \circ \beta_j$ est une involution.

Une carte combinatoire nD^2 définit ainsi 1 permutation et $n - 2$ involutions.

La Fig. 5.2 illustre une carte combinatoire composée de deux 3-cellules adjacentes notées Vol_1 et Vol_2 . Les brins de la 2-cellule commune entre ces deux volumes sont représentés en pointillés. Ils sont numérotés de a à d pour Vol_1 et de 1 à 4 pour Vol_2 . Les deux volumes sont reliés ensemble par l'involution β_3 (traits violets). La liaison β_1 (trait bleu) permet de passer d'une 1-cellule à une autre. La liaison β_2 (traits rouges) permet de passer d'une 2-cellule à une autre.

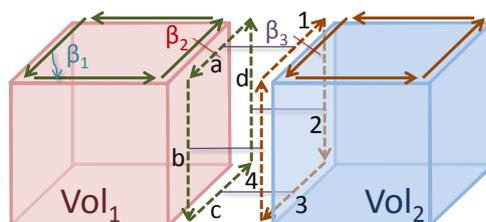


Figure 5.2 – Représentation de deux 3-cellules adjacentes notées Vol_1 et Vol_2 .

Enfin, pour rajouter dans ce modèle topologique des informations supplémentaires (comme la géométrie, la couleur des sommets ou encore l'aire des faces), la notion de *i-attribut* est employée permettant de stocker les informations relatives à une i -cellule. Cet i -attribut est associé à tous les brins appartenant à une même i -cellule.

Comme opérations topologiques sur les cartes combinatoires :

- L'opération de *i-couture* permet de relier deux brins par β_i . Ainsi si deux brins b_1 et b_2 sont tels que $\beta_i(b_1) = b_2$ alors b_1 est dit *i-cousu* à b_2 . De plus, si β_i est une involution à b_1 alors b_2 est également *i-cousu* à b_1 .
- L'opération de *i-décousure* (avec $i \in \{2, 3, \dots, n\}$) permet de découdre deux i -cellules cousues selon une $(i - 1)$ -cellule commune. Les liens β_i sont supprimés entre tous les brins appartenant à la $(i - 1)$ -cellule commune. Suite à l'application de cette opération, la $(i - 1)$ -cellule initiale est divisée en deux $(i - 1)$ -cellules. Ainsi la $(i - 1)$ -cellule initiale n'existe plus et ne relie donc plus les deux i -cellules.

Les **complexes linéaires cellulaires** (LCC pour *Linear Cell Complexes* en anglais) représentent une extension des cartes combinatoires permettant le plongement de ce modèle topologique dans un espace géométrique. La géométrie d'un sommet correspond alors à une coordonnée nD , une arête correspond à un segment, une face correspond à une portion de plan, et ainsi de suite pour les dimensions supérieures. Nous avons donc une distinction dans les LCC entre la géométrie et la topologie, permettant une plus grande efficacité des traitements qui ne nécessitent qu'une des deux informations. Le modèle unifié que nous avons proposé se base sur ce modèle topologique.

2. Les lecteurs désireux d'obtenir des détails sur ce modèle topologique peuvent se référer au livre écrit par Damiant and Lienhardt (2014) et à l'Habilitation à Diriger des Recherches de G. Damiant (2010).

Notre objectif concerne notamment la possibilité d'effectuer des changements topologiques sur des objets déformables au cours de leur simulation physique. Dès le départ, nous avons fait le choix d'employer un modèle topologique comme structure données pour décrire les objets à manipuler. En effet, ce type de structure de données permet non seulement de décrire la subdivision de l'objet en tout type de cellules, mais assure également la description de toutes les relations d'incidence et d'adjacence entre les cellules (informations nécessaires pour les modifications topologiques). Il est également possible de rajouter au sein de ce type de structure de données, d'autres informations qui sont nécessaires pour nos algorithmes de simulation (dans notre cas les informations relatives à la géométrie et aux propriétés mécaniques de l'objet). Par ailleurs, les cartes combinatoires sont définies en toute dimension avec un formalisme mathématique robuste, permettant ainsi de décrire de manière inhérente les opérations de modification topologique de l'objet en toute dimension.

5.3 Changements topologiques durant la simulation

En informatique graphique, l'opération de la découpe est vue comme une modification du modèle géométrique de l'objet afin de séparer les éléments le long d'une ligne de découpe. Le phénomène physique de la découpe n'est ainsi pas encore abordé. C'est-à-dire qu'il ne s'agit pas de simuler la déformation physique issue par exemple de l'enfoncement du scalpel dans le tissu mou avant sa découpe effective, mais l'idée est de "simplement" modifier le maillage volumique de l'objet en veillant à ne pas introduire d'éléments de mauvaises qualités qui créeraient des instabilités numériques. Dans ce contexte, nous pouvons nous référer à l'état de l'art proposé par [Wu et al. \(2014\)](#) sur les méthodes employées en simulation basée physique pour la découpe des objets déformables.

Pour effectuer une découpe dans un objet, plusieurs approches sont adoptées. Nous pouvons tout simplement supprimer les éléments qui sont touchés par l'objet créant la découpe. [Cotin et al. \(2000\)](#) utilisent cette approche dans le cadre d'une simulation temps réel. Par contre, cette méthode crée des zones non lissées dans le maillage de l'objet au niveau de la découpe, réduisant la précision de la simulation numérique tout en dégradant l'aspect visuel de l'objet. Une autre approche consiste alors à déplacer les sommets sur la surface de découpe avant d'effectuer une découpe le long des faces. Cette méthode permet d'obtenir une surface de l'objet lissée mais nécessite des traitements supplémentaires pour assurer la qualité des éléments dans la zone de découpe afin de ne pas créer d'éléments dégénérés. Pour cela de nombreux papiers s'attaquent au problème du re-maillage. Nous pouvons par exemple citer le papier de [Paulus et al. \(2015\)](#) parmi les plus récents.

En ce qui concerne les **travaux combinant simulation physique et découpe, trois extensions de la méthode des éléments finis ont été proposées**. La XFEM (*Extended Finite Element Method*) a été employée pour effectuer des coupes ou fractures dans des objets déformables ([Jeřábková and Kuhlen \(2009\)](#)). L'idée de cette extension est de gérer les discontinuités introduites par la découpe, en adaptant les fonctions de forme employées pour interpoler les déplacements à l'intérieur des éléments en fonction du déplacement de leurs nœuds. Un terme est ainsi rajouté aux fonctions de forme pour gérer les discontinuités ajoutant ainsi des degrés de liberté aux nœuds initiaux. Au final, deux représentations de l'objet sont ainsi employées : une pour la discrétisation du domaine de simulation et une pour la discrétisation numérique.

La CFEM (*Composite Finite Element Method*) est une autre extension employée pour la découpe d'objets déformables. Elle peut notamment être associée à des techniques de

détection de collisions (Wu et al. (2013)). L'idée est de combiner un ensemble d'éléments finis, en un élément fini de plus grande taille (dit élément composite) afin de réduire le nombre de degrés de liberté du système. Les fonctions de forme des éléments grossiers sont issues de la combinaison des fonctions de formes des éléments finis initiaux. La découpe se fait au niveau des éléments finis de grande résolution. Au final, un objet a trois représentations : un maillage hexaédrique (sur lequel la découpe s'effectue) avec un raffinement adaptatif possible au niveau de la découpe, un ensemble d'éléments composites (sur lesquels la simulation se fait), et un maillage surfacique (pour le rendu visuel). Durant la simulation, les déplacements calculés pour les sommets des éléments composites sont propagés (en utilisant une interpolation tri-linéaire) aux deux autres modèles.

Une dernière extension nommée PFEM (*Polyhedral Finite Element Method*) a été mise en place pour pouvoir gérer tout type de polyèdres (Wicke et al. (2007)), permettant ainsi d'éviter un remaillage fastidieux lors de la découpe issu du souhait de conserver des éléments finis standards (tétraèdres, hexaèdres).

Par rapport à ces travaux, notre idée était de proposer un modèle unique permettant de gérer à la fois la topologie de l'objet (pour effectuer au besoin des changements topologiques sur cet objet) et la physique de l'objet (pour simuler un comportement réaliste de l'objet) avec une mise à jour efficace au cours de la simulation. En effet, notre objectif était de n'avoir à manipuler qu'une seule représentation de l'objet au cours de la simulation. Nous souhaitions par ailleurs proposer une solution générique du point de vue topologique (maillage composé de tout type de cellules) et du point de vue physique (objet modélisé selon différents modèles physiques et/ou lois de comportement) pour ne pas se limiter, mais également pour pouvoir tester les différentes approches mises au point au fil des années. C'est pourquoi, nous avons préféré nous tourner vers une solution permettant de gérer des cellules de différents types géométriques avec la possibilité de mixer des éléments de différentes tailles, plutôt que de retenir une approche multi-résolution nécessitant une gestion entre les différents niveaux de résolution.

En ce qui concerne les **travaux combinant modèle topologique et simulation physique**, Meseure et al. (2010a,b) emploient la structure de données topologique des cartes généralisées pour effectuer une simulation physique basée sur un système masses-ressorts. Cette structure permet la représentation de quasi-variétés orientées ou non-orientées³. L'implémentation de leur modèle est basée sur la librairie MOKA⁴. Une correspondance 1 : 1 est proposée entre les arêtes du maillage et les ressorts du modèle physique. L'opération topologique de la découpe est basée sur celle de la décousure des cartes généralisées. Par contre, les temps de calculs annoncés sont 4 fois plus importants que pour l'implémentation usuelle d'un système masses-ressorts. Et la structure de données des cartes généralisées requiert un espace mémoire deux fois plus important que l'emploi des cartes combinatoires, sans compter qu'une structure de données additionnelle est proposée pour améliorer leurs performances, rajoutant une complexité au niveau de la mise à jour des informations nécessaires lors d'un changement topologique.

Darles et al. (2011) proposent de coupler les cartes généralisées au modèle physique des masses-interactions en utilisant le framework CORDIS-ANIMA⁵. Ce modèle physique est un modèle physique de type *meshless* (sans maillage) interconnectant des composants modé-

3. Une surface est une variété 2D (*manifold* en anglais), si chacun de ses points est homéomorphe à un disque ou un demi-disque au niveau de sa frontière. En 3D, les points doivent être homéomorphes à une boule. Une quasi-variété en dimension n est un objet n D obtenu uniquement par assemblage de n -cellules le long de $(n-1)$ -cellules, et tel que chaque $(n-1)$ -cellule est incidente à au plus deux n -cellules.

4. <http://moka-modeller.sourceforge.net>.

5. <http://acroe.imag.fr/produits/logiciel/cordis/cordis.html>.

lisant l'inertie du modèle physique. Des modifications topologiques (fracture, déchirement, fusion) sont effectuées en fonction de l'état de l'objet simulé.

[Jung et al. \(2012\)](#) utilisent également le framework CORDIS-ANIMA pour associer un système de particules (permettant des mouvements basés points) à une carte combinatoire en utilisant la librairie CGoGN ([Kraemer et al. \(2014\)](#)). L'idée est d'associer une forme volumique à un mouvement basé points et d'effectuer des changements topologiques. Les points du modèle particulaire (fonctions d'évolution 3D) sont ainsi associés aux cellules de la carte combinatoire. Au final, chaque sommet de la carte combinatoire est influencé par un ensemble de points. Puis au cours de la simulation, des changements topologiques peuvent être effectués selon des critères géométriques entraînant par exemple la séparation entre des cellules de la carte combinatoire. Il s'agit ensuite de mettre à jour le couplage entre la carte combinatoire et le modèle particulaire. Par contre, ce changement topologique est présenté comme une opération complexe suivant le positionnement des zones d'influence des points par rapport aux cellules topologiques.

En terme de structure topologique, nous nous sommes tournés vers les cartes combinatoires plutôt que les cartes généralisées nécessitant deux fois moins d'espace mémoire.

5.4 Modèle unifié combinant topologie et physique

Je présente dans cette section le modèle unifié que nous avons élaboré. Ce modèle a été initié durant la thèse de Elsa Fléchon. Lors de la rédaction de son manuscrit de thèse, nous nous sommes rendu compte de la complexité de sa mise à jour lors de changements topologiques. Cette mise à jour était devenue délicate car une même information était stockée dans plusieurs éléments de la structure de données. Cela partait généralement de l'idée de rendre directement accessibles les informations nécessaires pour un calcul donné (par exemple de faire en sorte qu'un ressort connaisse ses deux particules présentes à ses extrémités) mais cela a le désagrément de devoir répercuter des modifications à différents endroits de la structure de données lors d'un changement dans le modèle. Or notre objectif est de faire ce type de modifications de manière efficace.

Après ce constat, Guillaume Damiand et moi-même avons repris entièrement l'implantation du modèle (librairie TopoSIM). L'idée était de structurer l'ensemble de façon à ne pas avoir de redondance d'information, c'est-à-dire que chacun des paramètres des éléments ne soit stocké qu'à un seul endroit, limitant et simplifiant ainsi considérablement les traitements lors des modifications topologiques, et améliorant du même coup l'efficacité de notre modèle unifié. Nous avons ainsi bien veillé à séparer les informations topologiques (stockées dans la carte combinatoire) des informations liées au modèle physique, rendant le modèle beaucoup plus générique, facilitant ainsi l'intégration de nouveaux modèles physiques. Je présente dans cette section directement cette dernière version du modèle unifié.

Pour se rendre compte de l'évolution de ce modèle, l'article [Golec et al. \(2015\)](#) présente le travail effectué avec Guillaume Damiand, en comparant les performances obtenues par rapport au modèle initialement proposé par Elsa Fléchon. Ce modèle initial avait par ailleurs fait l'objet de deux articles et du manuscrit de thèse de Elsa Fléchon :

- [Flechon et al. \(2013\)](#) présente le modèle initial ainsi que les opérations topologiques de découpe et de perçage qui lui sont associées (illustré par la Fig. 5.3).

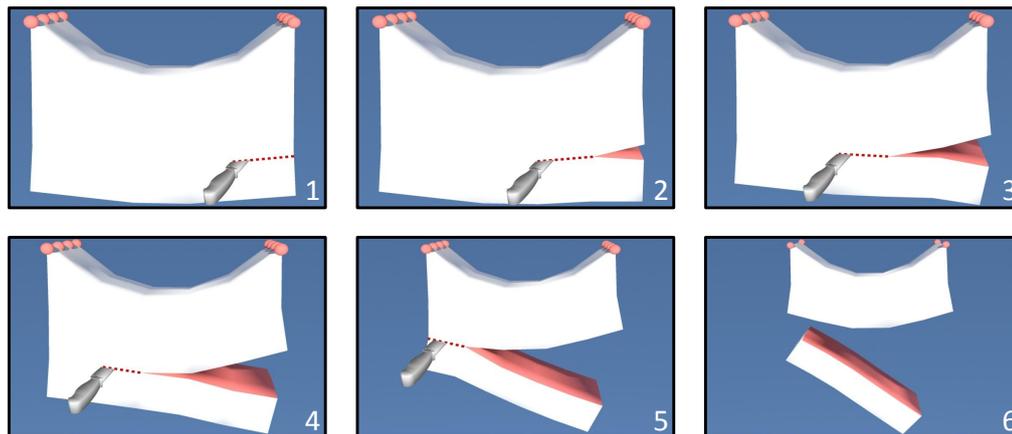


Figure 5.3 – Poutre composée de $5 \times 3 \times 3$ éléments progressivement découpée.

- [Flechon et al. \(2014\)](#) présente l'évolution de ce premier modèle pour permettre un raffinement adaptatif du modèle géométrique de l'objet simulé (illustré par la Fig. 5.4 en 2D et la Fig. 5.5 en 3D).

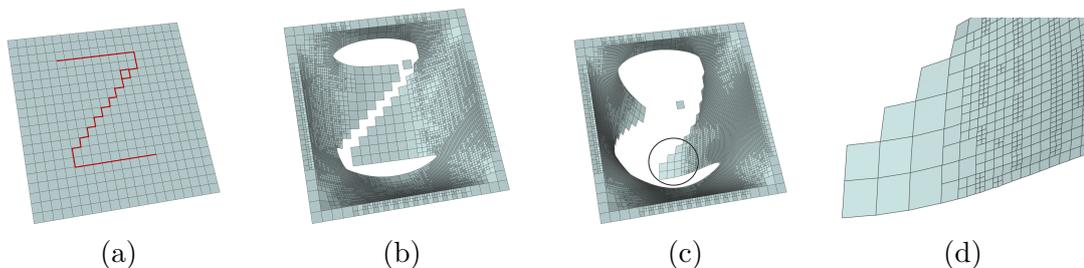


Figure 5.4 – En 2D. Les éléments du bord du maillage sont fixés et les autres sont soumis à la gravité. (a) Maillage initial composé de 20×20 éléments. Découpe ensuite suivant le chemin en rouge. (b) Les éléments sont automatiquement subdivisés. (c) 4 niveaux de subdivision différents visibles. (d) Zoom sur la zone entourée.

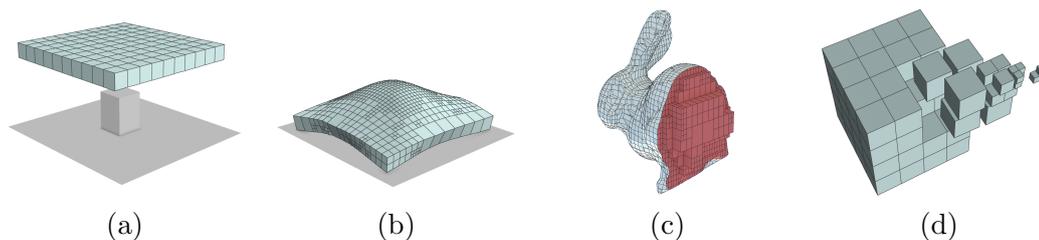


Figure 5.5 – En 3D. (a) Maillage initial composé de 10×10 éléments. (b) 3 niveaux de subdivision. (c) Un lapin composé de plusieurs niveaux de subdivision. (d) Subdivision d'un cube.

- Enfin, le manuscrit de thèse de Elsa [Flechon \(2014\)](#) revient en détails sur ces opérations topologiques et les traitements à effectuer pour la mise à jour des propriétés physiques de l'objet afin de garantir un comportement cohérent.

Construction du modèle. La Fig. 5.6 présente le processus de construction de notre modèle unifié pour un objet 3D constitué de 2 hexaèdres adjacents. Notre modèle unifié est basé sur l'emploi de la structure topologique des LCC avec l'utilisation de la librairie

Open Source CGAL (Damiand (2012)) donnant accès aux algorithmes manipulant cette structure de données topologique.

- La construction du modèle débute par le chargement d'un maillage. Dans notre exemple, ce maillage est composé de 2 hexaèdres.
- La carte combinatoire 3D correspondante est ensuite construite à partir des données géométriques et topologiques fournies en entrée. Le maillage initial est ainsi décomposé en deux 3-cellules (volumes), onze 2-cellules (faces), vingt 1-cellules (arêtes) et douze 0-cellules (sommets).
- La carte combinatoire 3D est ensuite plongée dans \mathbb{R}^3 pour obtenir la LCC correspondante. Ainsi les 0-cellules du modèle topologique sont attribuées de coordonnées géométriques 3D, associant ainsi la géométrie à la carte combinatoire.
- Enfin les attributs stockant les informations nécessaires à la simulation physique sont associés aux différentes cellules de la LCC créant ainsi le modèle unifié. Nous détaillons par la suite cette dernière étape en fonction du modèle physique employé (systèmes masses-ressorts ou modèle masse-tenseur). L'idée est d'associer de manière adéquate ces informations afin d'assurer l'aspect générique du modèle unifié et de factoriser les calculs qui sont communs à tous les modèles physiques (par exemple le calcul des accélérations et leur intégration pour obtenir le nouvel état de l'objet est commun à tout objet basé sur la dynamique Newtonienne). Pour le moment nous avons effectué une association 1 : 1 entre les cellules du modèle topologique et les éléments du modèle physique.

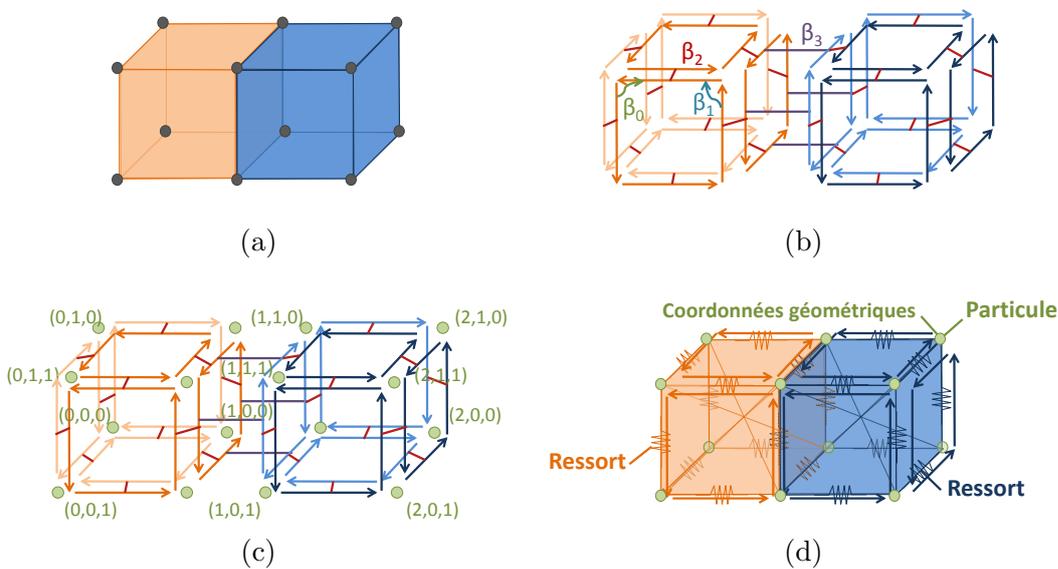


Figure 5.6 – Processus de construction de notre modèle unifié pour un maillage ayant deux hexaèdres basés sur l'emploi d'un système masses-ressorts pour la simulation de ses déformations.

Ensuite, afin d'assurer un accès direct à toutes les relations d'incidence entre les cellules et ainsi un accès en $O(1)$ aux informations requises par le modèle physique, nous stockons également dans chacun des éléments un tableau noté **dart** contenant ses brins. Par exemple pour un maillage hexaédrique, le tableau `dart[6]4` est associé à chacun des éléments du maillage regroupant ses brins par face. L'emploi de ce tableau permet ainsi un accès direct à chacun des points des éléments.

Intégration d'un système masses-ressorts. En Informatique Graphique, le modèle physique appelé "système masses-ressorts" (noté MSS pour *Mass-Spring System* en anglais) est communément employé pour animer un objet déformable (voir chapitre 3). Introduit par [Terzopoulos et al. \(1987\)](#), ce modèle est basé sur une discrétisation de l'objet en un ensemble de particules (ou masses) connectées entre-elles par des ressorts. Gouverné par la dynamique newtonienne, le mouvement de l'objet est alors induit par les forces appliquées sur les particules qui sont dues aux ressorts et aux forces externes (gravité et interaction).

Les principales difficultés dans l'emploi de ce modèle résident dans le choix du positionnement des ressorts entre les particules (c'est-à-dire dans la connectivité des particules entre elles), et dans la formulation des paramètres physiques des ressorts. En effet, un ressort est caractérisé par deux grandeurs physiques : un coefficient de raideur et un coefficient d'amortissement. Il s'agit alors de savoir relier ces paramètres physiques aux paramètres mécaniques de l'objet simulé (module de Young et coefficient de Poisson), afin de pouvoir reproduire le mouvement d'un objet déformable avec précision. Durant ces dernières années, différentes formulations ont été proposées dans ce sens ([Baudet et al. \(2009\)](#); [Kot and Nagahashi \(2015\)](#); [Kot et al. \(2014\)](#); [Lloyd et al. \(2007\)](#); [San-Vicente et al. \(2012\)](#)) permettant l'emploi de ce modèle dans le domaine médical.

Pour utiliser ce modèle physique, les informations suivantes ont alors été associées aux différentes cellules de notre modèle unifié :

- A chaque 0-cellule est associée une particule du MSS par son association avec l'attribut **Particule**. La structure de données **Particule** stocke la masse, l'accélération et la vitesse de la particule ainsi que la somme de l'ensemble des forces qui lui sont appliquées. Elle stocke également des informations relatives aux contraintes pouvant s'exercer sur la particule, en indiquant si elle est fixée selon un des axes du repère.
- A chaque 1-cellule est associée un ressort du MSS grâce à son association avec l'attribut **Ressort**. Cette structure de données **Ressort** stocke le coefficient de raideur et le coefficient d'amortissement du ressort en question.
- En 2D, à chaque 2-cellule sont associés les 2 ressorts diagonaux interne à l'élément quadratique. En 3D, à chaque 3-cellule sont associés les 4 ressorts diagonaux internes à l'élément hexaédrique.

Lors de la simulation, la force $\vec{f}_i(t)$ appliquée sur la particule i correspond à la somme des forces exercées par les ressorts connectés à cette particule avec $\vec{f}_i(t) = \sum_j \vec{f}_{i,j}(t)$, où j correspond aux différentes particules connectées à i par un ressort. Considérons le ressort connectant les particules i et j . Si nous notons l_{ij} sa longueur au repos, k_{ij} sa raideur et ν_{ij} son coefficient d'amortissement, la force qu'il exerce sur la particule i est définie par :

$$\vec{f}_{i,j}(t) = \vec{f}_{i,j}^e(t) + \vec{f}_{i,j}^v(t)$$

avec :

$$\begin{cases} \vec{f}_{i,j}^e(t) &= k_{ij} (\|x_i(t) - x_j(t)\| - l_{ij}) \frac{x_i(t) - x_j(t)}{\|x_i(t) - x_j(t)\|} \\ \vec{f}_{i,j}^v(t) &= \left(\nu_{ij} (\dot{x}_i(t) - \dot{x}_j(t)) \cdot \frac{x_i(t) - x_j(t)}{\|x_i(t) - x_j(t)\|} \right) \frac{x_i(t) - x_j(t)}{\|x_i(t) - x_j(t)\|} \end{cases}$$

où $x_i(t), x_j(t)$ correspondent aux positions des particules i et j et $\dot{x}_i(t), \dot{x}_j(t)$ à leurs vitesses. Le calcul des forces induites par les ressorts sur les particules requiert ainsi la connaissance des particules présentes aux extrémités de chacun des ressorts. Cette information est directement accessible en utilisant la carte combinatoire et le tableau `dart` sans à avoir à stocker directement le lien entre les ressorts et les particules.

Intégration du modèle masse-tenseur. Dans le cadre de l'emploi du modèle masse-tenseur (*cf.* chapitre 4), les informations suivantes sont associées aux différentes cellules :

- A chaque 0-cellule est associée un nœud grâce à son association avec l'attribut `Particule` permettant de stocker les informations relatives au nœud.
- A chaque 3-cellule est associée la formulation des forces relatives à chacun de ses nœuds issue du travail de thèse de Xavier Faure. Cette formulation dépend de la loi de comportement employée (Hooke, Saint-Venant Kirchhoff, etc.).

Lors de la simulation, pour effectuer le calcul des forces appliquées à chacun des éléments, chaque élément commence par calculer le déplacement de ses nœuds en utilisant leur position initiale et courante. L'emploi du tableau `dart` permet de retrouver chacun des nœuds des 3-cellules en temps constant. Ensuite la force appliquée à chacun des nœuds est calculée en utilisant sa formulation explicite générée par le calcul formel.

Calcul générique des forces. Ainsi pour intégrer n'importe quel modèle physique, il suffit de définir les propriétés mécaniques associées aux différentes i -cellules, ainsi que la fonction `addForce()` qui effectuera le calcul des forces internes à un élément de l'objet. Ensuite, le calcul des forces appliquées sur l'objet est effectué en réalisant un parcours sur les i -cellules concernées, c'est-à-dire en parcourant les i -cellules affectées par le calcul des forces (les 1-cellules et les 2-cellules pour le MSS en 2D ; les 1-cellules et les 3-cellules pour le MSS en 3D ; les 3-cellules pour le modèle masse-tenseur), et en appelant la fonction `addForce()` correspondante effectuant le calcul des forces selon le modèle physique.

Informations physiques stockées dans le modèle unifié. En fonction du modèle physique utilisé, les informations physiques des éléments constituant l'objet doivent être parfois calculées avant de les stocker dans le modèle unifié.

Le calcul de la **masse de chacun des sommets du maillage** de l'objet se fait de la façon suivante quelque soit le modèle physique employé.

- En 2D, la masse d'un sommet s est calculée par accumulation des contributions de chacun de ses éléments incidents. Ainsi, la formulation de la masse du sommet s , incident à n_f faces, est donnée par :

$$m_s = \frac{1}{n_f} \sum_{e \in F_s} \rho_e \mathcal{A}_e$$

où m_s est la masse de s , F_s est l'ensemble des faces incidentes à s , ρ_e est la densité surfacique de l'élément e dont l'aire est \mathcal{A}_e .

- En 3D, la formulation de la masse de s , incident à n_{vol} volumes, est donnée par :

$$m_s = \frac{1}{n_{vol}} \sum_{e \in Vol_s} \rho_e \mathcal{V}_e$$

où m_s est la masse de s , Vol_s est l'ensemble des volumes incidents à s , \mathcal{V}_e et ρ_e sont respectivement le volume et la densité volumique de l'élément e .

Le calcul de la masse des sommets de l'objet est effectué lors de la phase de construction du modèle unifié, et peut être refait durant la simulation lors d'un changement topologique

de l'objet. Par exemple, une coupure au sein de l'objet peut amener un changement dans le nombre d'éléments incidents à un sommet donné induisant la mise à jour de sa masse en fonction de ce nouveau nombre d'éléments incidents.

Dans le cadre du système masses-ressorts, nous avons employé le formalisme de [Baudet et al. \(2009\)](#) pour le calcul des **raideurs des ressorts**. Dans ce formalisme, le calcul des raideurs est dépendant du nombre d'éléments incidents aux ressorts. Il est donc effectué lors de la construction du modèle unifié et lors d'une modification au cours de la simulation du nombre d'éléments incidents à un ressort donné ([Flechon et al. \(2013\)](#)).

Résultats de simulation. La Fig. 5.7 (a) présente les résultats obtenus pour le test de traction entre une simulation effectuée en utilisant le modèle masses-ressorts et une autre avec le modèle masse-tenseur mises en comparaison à la solution analytique. Pour cela, nous considérons une poutre décomposée en $5 \times 5 \times 20$ et $10 \times 10 \times 40$ éléments et la solution analytique donnée par

$$y = \frac{\rho g l h}{24 E I} (4Lx^3 - 6L^2x^2 - x^4)$$

pour une poutre de longueur $L = 1$ m, de hauteur $h = 0.25$ m, de largeur $l = 0.25$ m et avec un moment d'inertie $I = lh^3/12$. Un visuel à la position d'équilibre est donné par la Fig. 5.7 (b) pour la poutre décomposée en $10 \times 10 \times 40$ modélisée en masses-ressorts (MSS) ou masse-tenseur (MT).

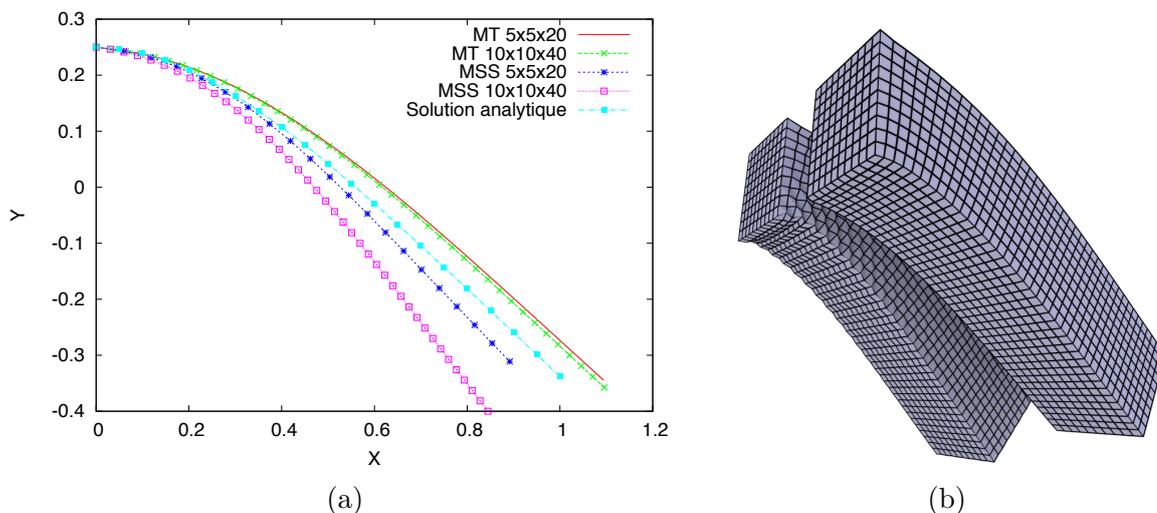


Figure 5.7 – (a) Comparaison entre le modèle masses-ressorts, l'approche masse-tenseur et la solution analytique pour le test de traction. (b) Comparaison between the MSS (left) and the MT (right) for a traction test performed on beams discretized in $10 \times 10 \times 40$ elements.

Opération topologique de la découpe. L'opération de la découpe du modèle unifié repose sur l'opérateur de découpe des cartes combinatoires. La Fig. 5.8 illustre une opération de 3-découpe effectuée sur un objet 3D (présenté sur la Fig. 5.8(a)) composé de quatre hexaèdres notés Vol_0 , Vol_1 , Vol_2 et Vol_3 qui sont cousus ensemble par β_3 . La face commune aux volumes Vol_0 et Vol_1 est décrite par les sommets (s_1, s_2, s_3, s_4) . L'opérateur de 3-découpe est employé pour découper ces deux volumes, supprimant ainsi les liaisons

β_3 entre les brins incidents à Vol_0 et Vol_1 . Suite à cette opération, la 2-cellule qui leur est commune va être scindée en deux. Sur la Fig. 5.8(b), nous voyons que les sommets s_1 et s_2 ont ainsi été scindés en s'_1 , s'_2 , et les arêtes initiales (s_4, s_1) , (s_1, s_2) et (s_2, s_3) en (s_4, s'_1) , (s'_1, s'_2) et (s'_2, s_3) . La scission d'un sommet ou d'une arête entraîne la copie de l'attribut stocké permettant de donner des valeurs aux nouveaux éléments créés.

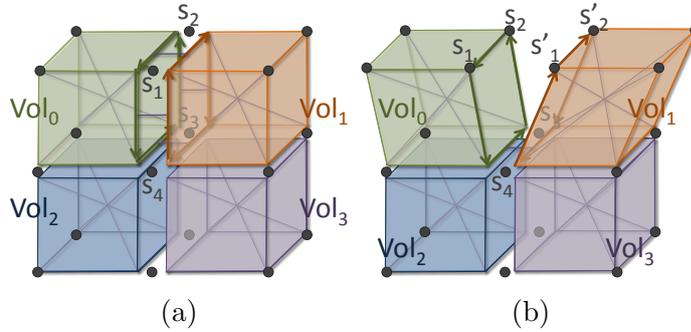


Figure 5.8 – Un exemple de 3-découure.

Notons que le tableau `dart` stocké dans chacun des éléments n'a pas à être modifié puisque les brins sont toujours valides quelque soit l'opération de modification topologique effectuée. Ainsi, aucune autre opération n'est à faire lors de la découpe grâce à l'emploi d'une carte combinatoire, mise à part la mise à jour éventuelle des masses ou raideur des ressorts impliqués dans la découpe si leur nombre d'éléments incidents est modifié.

La Fig. 5.9 (a) présente dix coupes successives appliquées sur un maillage représentant une girafe sans simulation contenant 85405 éléments. Les différentes parties ont ensuite été séparées pour un meilleur rendu visuel. La Fig. 5.9 (b) présente une découpe effectuée durant la simulation d'une poutre soumise à la gravité fixée en ses deux extrémités, en employant le modèle masse-tenseur avec la loi de comportement de Saint-Venant Kirchhoff.

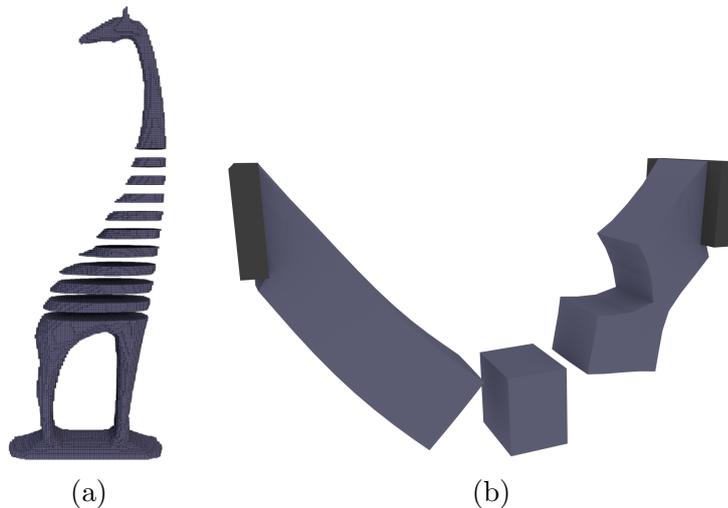


Figure 5.9 – Découpe (a) d'un maillage représentant une girafe et (b) d'une poutre soumise à la gravité fixée en ses deux extrémités (modèle masse-tenseur avec la loi de Saint-Venant Kirchhoff).

Informations supplémentaires pour le raffinement. Un hexaèdre non subdivisé est composé de 6 faces et de 12 arêtes. Ensuite, si il est possible de le raffiner localement, des informations supplémentaires sont nécessaires pour suivre les arêtes (ou faces) se trouvant sur le même côté de l'élément, celui-ci présentant un nombre variable de faces et d'arêtes. Dans ce sens, nous avons rajouté les informations suivantes à notre modèle unifié :

- Le **niveau de subdivision (noté l_b) des brins** de l'objet. Ce niveau de subdivision est relatif à l'hexaèdre auquel appartient le brin considéré. Il est initialisé lors de la création de la structure de données et mis à jour durant le processus de raffinement. Si nous considérons un hexaèdre non subdivisé, à l'initialisation chacun des brins de ses arêtes a un niveau de subdivision égal à 1. Puis, si un brin est subdivisé en 2 (mais pas l'hexaèdre lui-même), son niveau de subdivision est multiplié par 2. Par contre, si l'hexaèdre est lui-même subdivisé en 8, chaque brin a un niveau de subdivision égal à 1 car relatif aux sous-hexaèdres auxquels ils appartiennent désormais.
- Une **marque de coin sur les brins** se trouvant aux coins de chaque face. Pour un hexaèdre, ce sont ainsi 4 brins qui sont marqués pour chaque face.
- La **direction des arêtes** avec 3 directions possibles en 3D notées dl_1, dl_2, dl_3 .
- Nous introduisons aussi la notion de **face (ou arête) hiérarchique** correspondant à l'ensemble des faces (ou arêtes) générées par la subdivision de la même face (ou arête) initiale. Si la face (ou arête) initiale n'a pas été subdivisée, la face (ou arête) hiérarchique correspond à elle-même. Il faut noter que les arêtes d'une même arête hiérarchique ont la même direction. Cette propriété rend facile le fait de suivre le bord d'un élément. Pour cela, nous passons d'une 1-cellule à une autre en utilisant le pointeur β_1 : si les 1-cellules ont la même direction, elles font partie du même bord (autrement dit de la même arête hiérarchique), sinon nous avons changé de bord.

Pour clarifier ces notions, considérons l'exemple illustré par la Fig. 5.10 qui concerne un objet initialement composé de 6 hexaèdres dont certains ont ensuite été subdivisés.

- Sur la partie (a) de la figure nous observons ainsi que l'hexaèdre central (en vert) n'a pas été subdivisé alors que certains de ses voisins ont été raffinés. Par conséquence, les faces ou arêtes hiérarchiques de l'hexaèdre central sont composées d'un nombre variable de faces ou d'arêtes. Ceci est visible sur la partie (b) qui représente uniquement l'hexaèdre central. Ainsi, la face F_0 est sa propre face hiérarchique, tandis que la face hiérarchique F_1 est composée de 7 faces.
- De même, si nous observons les arêtes de la face F_0 , l'arête hiérarchique du fond est composée d'une seule arête de direction dl_1 et le niveau de subdivision du brin e_0 de cette face est $l_b(e_0) = 1$. Tandis que l'arête hiérarchique vers l'avant est composée de 3 arêtes de même direction dl_1 avec $l_b(e_1) = l_b(e_2) = 4$ et $l_b(e_3) = 2$ pour les brins de cette face. Enfin les arêtes hiérarchiques de gauche et de droite sont chacune composées de 4 arêtes de même direction dl_2 avec par $l_b(e_4) = 4$ pour le brin e_4 .
- Nous pouvons noter que les brins mis en relation par le pointeur β_2 ont le même niveau de subdivision puisqu'il met en relation les cellules de dimension 2 adjacentes. Nous avons ainsi $l_b(e_7) = l_b(e_3)$ et $l_b(e_5) = l_b(e_1)$ avec $\beta_2(e_7) = e_3$ et $\beta_2(e_5) = e_1$.
- En orange sont notés les brins qui marquent les 4 coins des faces hiérarchiques. Nous pouvons les observer aux 4 coins de la face F_1 qui a été subdivisée.
- La figure (c) revient en détails sur les niveaux de subdivisions des brins des arêtes du bord de l'hexaèdre central et montre les directions des arêtes hiérarchiques.

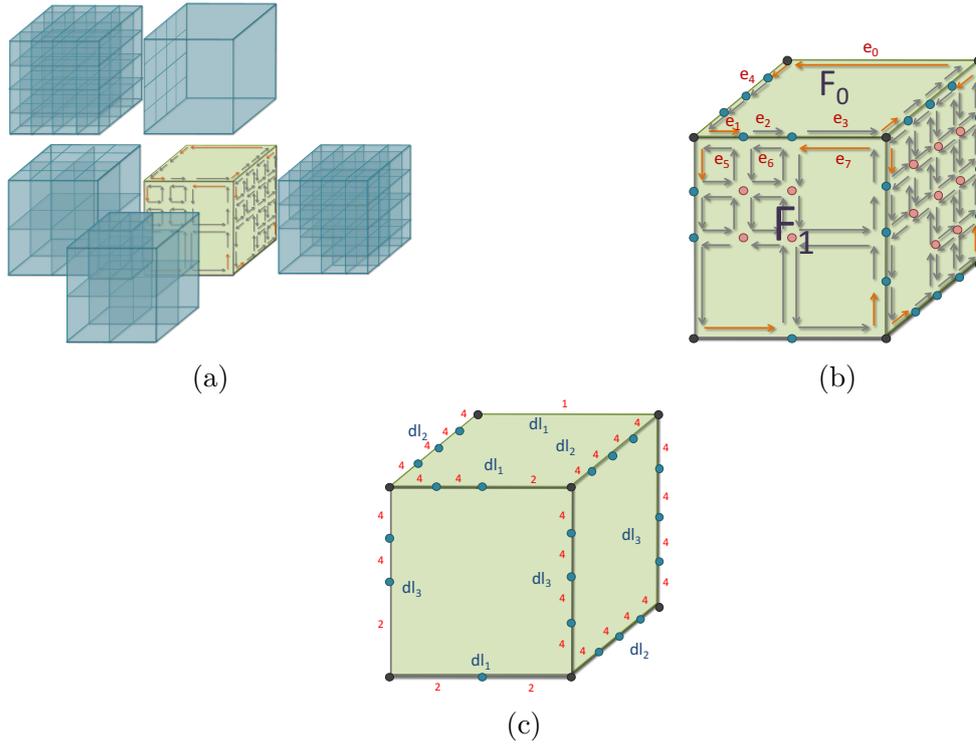


Figure 5.10 – (a) Un modèle 3D composé de 6 hexaèdres dont l’hexaèdre central (en vert) n’a pas subi de subdivision. (b) Zoom sur l’hexaèdre central qui est adjacent à plusieurs autres hexaèdres de différents niveaux de subdivision. Les brins de coin sont représentés en orange. (c) Mise en évidence des niveaux de subdivision (en rouge) des brins du bord de l’hexaèdre central et des directions (dl_1, dl_2, dl_3) possibles pour les arêtes (notées uniquement pour les hiérarchiques).

La Fig. 5.11 insiste sur le fait que le niveau de subdivision des brins est relatif à l’élément auquel ils appartiennent. Ainsi sur la partie (a), nous pouvons voir que tous les brins de la face de gauche de l’hexaèdre ont un niveau de subdivision de 4 (car adjacent à un autre hexaèdre subdivisé). Puis si l’hexaèdre est lui même subdivisé en 64 sous-hexaèdres comme sur la partie (b), le niveau de subdivision des brins repasse à 1 car relatif au sous-hexaèdre auquel ils appartiennent désormais. Ces 64 sous-hexaèdres ont été obtenus en effectuant une première opération de raffinement 1:8 sur l’hexaèdre initial (opération qui permet de subdiviser un élément en 8 éléments), puis un second raffinement 1:8 sur chacun des 8 hexaèdres obtenus précédemment (avec $8 \times 8 = 64$).

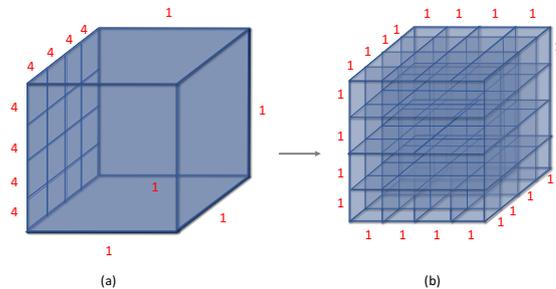


Figure 5.11 – (a) Les brins de la face de gauche ont tous un niveau de subdivision de 4 (noté sur les arêtes) relatif à l’hexaèdre qui n’est pas subdivisé. (b) Les mêmes brins ont désormais un niveau de subdivision de 1 une fois l’hexaèdre initial subdivisé en 64 sous-hexaèdres.

Opération topologique du raffinement 1:4. Nous avons précédemment employé une opération de raffinement qui permet de subdiviser un élément 3D en 8 éléments 3D. Cette opération topologique (appelée raffinement 1:8) s'appuie sur l'opération de raffinement 1:4 qui permet de raffiner une face en 4 sous-faces⁶. Considérons la Fig. 5.12 pour illustrer les différentes étapes de cette opération de raffinement d'un élément 2D.

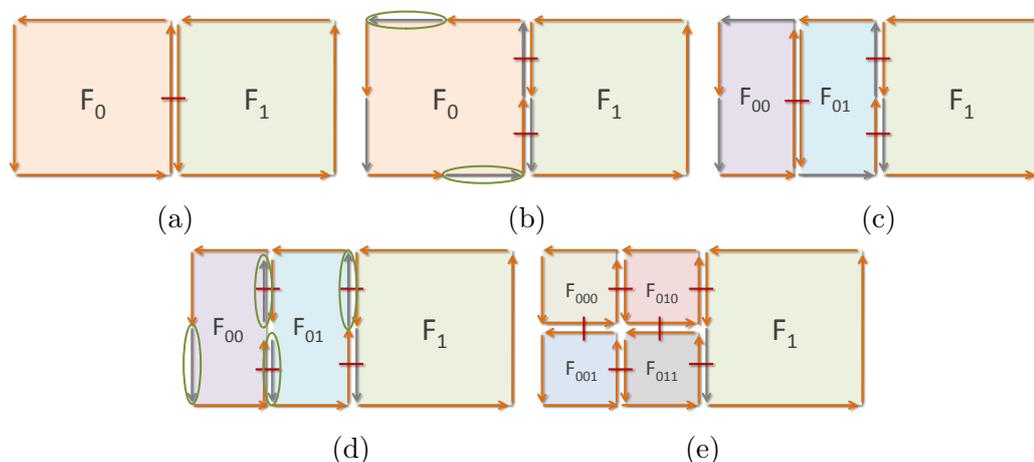


Figure 5.12 – Les différentes étapes relatives au raffinement 1:4 d'un quadrilatère.

- (a) Nous considérons un objet 2D composé de deux faces notées F_0 et F_1 de type quadrilatère adjacentes et donc composées de quatre brins chacune.
- (b) Un nouveau sommet est inséré au milieu de chacun des brins de F_0 . L'insertion d'un sommet (0-cellule) divise ainsi chacune des brins en deux dont le niveau de subdivision est ainsi multiplié par 2. Cette étape crée 5 nouveaux brins (représentés en gris) en observant que la scission de la 1-cellule commune à F_0 et F_1 génère deux nouveaux brins. Ensuite, un attribut **Particule** est associé à chacune des 4 nouvelles 0-cellules et un attribut **Ressort** est associé à chacune des 4 nouvelles 1-cellules nécessitant la mise à jour des différentes propriétés physiques.
- (c) Une nouvelle 1-cellule est insérée au milieu de F_0 (entre les brins entourés de vert ayant pour origine les sommets nouvellement créés). L'insertion de la nouvelle 1-cellule crée deux nouveaux brins. F_0 est ainsi scindée en deux sous-2-cellules F_{00} et F_{01} . Un nouveau **Ressort** est associé à la nouvelle 1-cellule.
- (d) Un nouveau sommet est inséré au milieu de la nouvelle 1-cellule. Puis, une nouvelle **Particule** est associée à ce nouveau sommet et un nouveau **Ressort** est associé à la nouvelle arête.
- (e) F_{00} et F_{01} sont ensuite scindées en deux de la même manière. Deux 1-cellules vont être insérées dans ces deux dernières faces en utilisant deux brins opposés appartenant à la même 2-cellule (brins entourés de vert). F_0 a ainsi été subdivisée en quatre 2-cellules notées F_{000} , F_{010} , F_{001} et F_{011} .

Opération topologique du raffinement 1:8. La Fig. 5.13 illustre les étapes relatives à l'opération topologique du raffinement 1:8 effectuée pour subdiviser un objet 3D en 8.

- (a) L'élément E est initialement composé de 8 sommets/particules et 12 arêtes/ressorts.

6. Ce processus de subdivision s'appuie sur les opérations d'insertion des cartes combinatoires.

- (b) L'opération de raffinement 1:4 est appliquée sur chacune des 6 faces de l'élément créant ainsi de nouveaux sommets/particules (en rouge) et de nouvelles arêtes/ressorts.
- (c) Création de la forme illustrée composée de 19 sommets et 12 arêtes.
- (d) Insertion de la forme précédente en cousant ensemble les bords permettant d'obtenir au final 8 hexaèdres à partir de l'hexaèdre E initial. Chacun des brins des arêtes a un niveau de subdivision égale à 1.

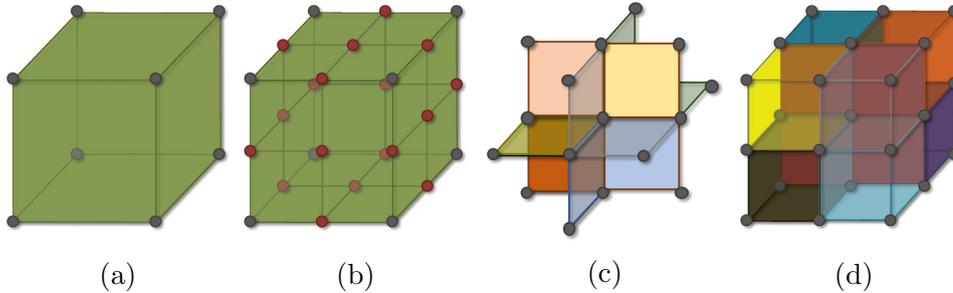


Figure 5.13 – Les différentes étapes relatives au raffinement 1:8 d'un hexaèdre.

Ces opérations topologiques de raffinement ont été grandement simplifiées par l'amélioration de notre modèle par rapport à celui proposé durant la thèse de Elsa Fléchon. Nous veillons par ailleurs à garantir l'aspect générique des algorithmes et opérations topologiques mises en place. Ces améliorations vont permettre d'implanter n'importe quel schéma de raffinement et ainsi sortir des schémas classiques réalisables en utilisant un octree.

5.5 Discussion

Nous souhaitons pouvoir modifier dynamiquement la topologie d'objets simulés par l'emploi d'un modèle physique plus ou moins complexe. De plus, nous souhaitons proposer une solution générique permettant de gérer des objets décomposés en cellules de tout type. Pour cela nous avons mis en place un modèle permettant de gérer de manière efficace à la fois la topologie et les paramètres physiques des objets simulés.

- Le fait d'avoir employé la structure de données des cartes combinatoires a apporté l'aspect générique au niveau de la topologie des cellules. Les cellules peuvent ainsi être de topologie et de dimension quelconque.
- La façon dont nous avons stocké les paramètres physiques de l'objet au sein des éléments du modèle a assuré l'aspect générique de la solution au niveau du modèle physique, c'est-à-dire qu'il est possible d'un point de vue technique de mixer des éléments ayant un modèle physique différent et de les modifier durant la simulation.
- D'autre part, notre modèle garantit une gestion aisée et efficace de toute modification topologique de l'objet, grâce à la façon dont nous y avons stocké ses paramètres physiques et l'ajout d'un tableau donnant un accès direct aux brins de chacun des éléments. Ainsi, nous évitons des mises à jours fastidieuses en cas de modifications topologiques. Par exemple, une découpe au sein de l'objet est effectuée en utilisant directement l'opérateur correspondant de la carte combinatoire. Ceci a été rendu possible car nous avons veillé à stocker les paramètres physiques qu'à un seul endroit dans la structure de données et à les associer aux cellules concernées par ces paramètres physiques. La mise à jour de la carte combinatoire est donc faite de manière naturelle en dupliquant les attributs concernés (traitement effectué en natif par l'opération de

découpe de la carte combinatoire) et les seules mises à jour éventuelles concernent celles des masses et des raideur des ressorts dues aux changements élaborés sur l'objet.

- Par contre, il faut savoir que la communauté d'informatique graphique a souvent peur des modèles topologiques. Dès que nous commençons à parler de brins, de pointeurs, les auditeurs frissonnent. Et bien essayer c'est l'adopter ! Dites-moi comment vous gérez efficacement les changements de topologie dans une implantation usuelle d'un système masses-ressorts ? Comment vous retrouvez facilement les voisins d'un élément donné, si vous ne voulez pas rajouter une autre structure que le graphe pour gérer cela ? Puis une autre qui va gérer un autre changement topologique, puis encore une autre pour... Bref, testez les cartes combinatoires, et vous comprendrez... De manière plus formelle, l'emploi des cartes combinatoires permet de bénéficier d'un modèle topologique qui assure la validité topologique du modèle après avoir effectué une opération topologique dessus. Le modèle topologique décrit également toutes les relations d'incidences et d'adjacence entre les cellules du modèle, ces cellules pouvant être de dimension et de topologie quelconques. Par contre, il peut parfois être fastidieux d'accéder à ces relations si un parcours devient complexe au travers des brins selon la relation que nous recherchons. Mais pour pallier ce problème, nous avons vu que nous avons optimisé les accès qui nous intéressaient en rajoutant à chaque élément (2-cellules en 2D, 3-cellules en 3D) un accès direct à ses brins.

Reste ensuite à mettre en place et valider des modèles physiques basés sur des éléments de topologies et géométries quelconques (fonctions de forme à définir selon la géométrie des éléments, etc.) et gérer correctement le passage d'un modèle physique (ou loi de comportement) à un autre, et enfin d'identifier si la modification de leur définition engendre des instabilités numériques ou physiques en modifiant notamment l'énergie du système.

Contributions

Publications

- K. Golec, M. Coquet, F. Zara, G. Damiand. *Improvement of a Topological-Physical Model to manage different physical simulations*. In *WSCG 2015 - International Conference in Central Europe on Computer Graphics and Visualization*, Plzen (République Tchèque), June 2015.
- E. Fléchon. *Définition d'un modèle unifié pour la simulation physique adaptative avec changements topologiques*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1, Dec. 2014.
- E. Fléchon, F. Zara, G. Damiand, F. Jaillet. *A unified topological-physical model for adaptive refinement*. In *11th Workshop on Virtual Reality Interaction and Physical Simulation*, Eurographics Digital Library, pages 39–48. Eurographics Association, Bremen (Allemagne), Sept. 2014.
- E. Fléchon, F. Zara, G. Damiand, F. Jaillet. *A generic topological framework for physical simulation*. In *WSCG 2013 - International Conference in Central Europe on Computer Graphics and Visualization*, pages 104–113, Plzen (République Tchèque), June 2013.

Logiciel

- **TopoSIM** - Framework topologique générique pour la simulation physique. 4 co-développeurs. Environ 20 000 lignes de code.

Bibliographie

- V. Baudet, M. Beuve, F. Jaillet, B. Shariat, and F. Zara. Integrating Tensile Parameters in Hexahedral Mass-Spring System for Simulation. In *International Conference on Computer Graphics, Visualization and Computer Vision'2009 - WSCG'2009*, Feb. 2009. URL <http://liris.cnrs.fr/publis/?id=3673>.
- S. Cotin, H. Delingette, and N. Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. *The Visual Computer*, 16(8) :437–452, 2000. URL <http://www.inria.fr/sophia/asclepios/Publications/Herve.Delingette/TensorMass-VC2000.pdf>.
- G. Damiand. *Contributions aux Cartes Combinatoires et Cartes Généralisées : Simplification, Modèles, Invariants Topologiques et Applications*. Habilitation à diriger des recherches, INSA de Lyon/Université Lyon1, Sept. 2010. URL <http://liris.cnrs.fr/publis/?id=4921>.
- G. Damiand. Combinatorial maps. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.0 edition, 2012.
- G. Damiand and P. Lienhardt. *Combinatorial Maps : Efficient Data Structures for Computer Graphics and Image Processing*. A K Peters/CRC Press, September 2014.
- E. Darles, S. Kalantari, X. Skapin, B. Crespín, and A. Luciani. Hybrid Physical – Topological Modeling of Physical Shapes Transformations. In *Proc. of DMDCM 2011*, pages 154–157, Washington, DC, USA, 2011.
- E. Flechon. *Définition d'un modèle unifié pour la simulation physique adaptative avec changements topologiques*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1, Dec. 2014.
- E. Flechon, F. Zara, G. Damiand, and F. Jaillet. A generic topological framework for physical simulation. In *WSCG 2013 - International Conference in Central Europe on Computer Graphics and Visualization*, pages 104–113, June 2013. ISBN 978-80-86943-74-9. URL <http://liris.cnrs.fr/publis/?id=6139>.
- E. Flechon, F. Zara, G. Damiand, and F. Jaillet. A unified topological-physical model for adaptive refinement . In *11th Workshop on Virtual Reality Interaction and Physical Simulation*, Eurographics Digital Library, pages 39–48. Eurographics Association, Sept. 2014. doi : 10.2312/vriphys.20141222.

- K. Golec, M. Coquet, F. Zara, and G. Damiand. Improvement of a Topological-Physical Model to manage different physical simulations. In *WSCG 2015 - International Conference in Central Europe on Computer Graphics and Visualization*, June 2015.
- L. Jeřábková and T. Kuhlen. Stable cutting of deformable objects in virtual environments using xfem. *IEEE Comput. Graph. Appl.*, 29(2) :61–71, Mar. 2009. ISSN 0272-1716. doi : 10.1109/MCG.2009.32. URL <http://dx.doi.org/10.1109/MCG.2009.32>.
- T. Jung, A. Allaoui, E. Darles, X. Skapin, P. Meseure, and A. Luciani. Motion/Geometry Mapping for Volumetric Representation. In *Proc. of VRIPHYS 2012*, Dec. 2012.
- M. Kot and H. Nagahashi. Second degree of freedom of elastic objects - adjustable poisson's ratio for mass spring models. In *GRAPP 2015 - Proceedings, Berlin, Germany, 11-14 March, 2015*, pages 138–142. SciTePress, 2015. ISBN 978-989-758-087-1. doi : <http://dx.doi.org/10.5220/0005303601380142>.
- M. Kot, H. Nagahashi, and P. Szymczak. Elastic moduli of simple mass spring models. *The Visual Computer*, Sept. 2014. ISSN 0178-2789. doi : 10.1007/s00371-014-1015-5.
- P. Kraemer, L. Untereiner, T. Jund, S. Thery, and D. Cazier. Cgogn : n-dimensional meshes with combinatorial maps. In J. Sarrate and M. Staten, editors, *Proceedings of the 22nd International Meshing Roundtable*, pages 485–503. Springer International Publishing, 2014. ISBN 978-3-319-02334-2. doi : 10.1007/978-3-319-02335-9_27.
- P. Lienhardt. Topological models for boundary representation : a comparison with n-dimensional generalized maps. *Comput. Aided Des.*, 23(1) :59–82, 1991. ISSN 0010-4485.
- B. A. Lloyd, G. Székely, and M. Harders. Identification of spring parameters for deformable object simulation. *IEEE Transactions on Visualization and Computer Graphics*, 13(1) : 1081–1093, 2007. ISSN 10772626. doi : 10.1109/TVCG.2007.1055.
- P. Meseure, E. Darles, and X. Skapin. A Topology-Based Mass/Spring System. In *Proc. of CASA'2010 (short papers)*, St Malo (F), June 2010a.
- P. Meseure, E. Darles, and X. Skapin. Topology-based Physical Simulation. In *Proc. of VRIPHYS 2010*, pages 1–10, Copenhagen, Denmark, Nov. 2010b.
- C. J. Paulus, L. Untereiner, H. Courtecuisse, S. Cotin, and D. Cazier. Virtual cutting of deformable objects based on efficient topological operations. *The Visual Computer*, pages 1–11, 2015.
- G. San-Vicente, I. Aguinaga, and J. Tomás Celigüeta. Cubical Mass-Spring Model design based on a tensile deformation test and nonlinear material model. *IEEE transactions on visualization and computer graphics*, 18(2) :228–41, Feb. 2012. ISSN 1941-0506. doi : 10.1109/TVCG.2011.32.
- D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21(4) :205–214, Aug. 1987. ISSN 0097-8930.
- M. Wicke, M. Botsch, and M. Gross. A finite element method on convex polyhedra. *Comput. Graph. Forum*, 26(3) :355–364, 2007. URL <http://dblp.uni-trier.de/db/journals/cgf/cgf26.html#WickeBG07>.

- J. Wu, C. Dick, and R. Westermann. Efficient collision detection for composite finite element simulation of cuts in deformable bodies. *The Visual Computer*, 29(6-8) :739–749, 2013.
- J. Wu, R. Westermann, and C. Dick. Physically-based simulation of cuts in deformable bodies : A survey. In *Eurographics 2014 State-of-the-Art Report*, pages 1–19, Strasbourg, France, 2014. Eurographics Association.

Apport d'un modèle surfacique pour la simulation physique

Je présente dans ce chapitre les travaux qui ont été réalisés dans le cadre de la thèse de Mathieu Bailet (2014) financée par l'ARC 6 de la région Rhône-Alpes de 2011 à 2014. Ce travail de recherche rentre dans le cadre plus global de la conception d'un simulateur médical pour l'entraînement aux gestes médicaux de l'accouchement et plus particulièrement à la pose de forceps. L'idée du simulateur est de coupler une simulation numérique de l'accouchement avec une interface haptique. En ce qui concerne le travail de recherche de Mathieu Bailet, l'objectif concernait la proposition d'un modèle physique adéquat pour simuler les déformations de la tête fœtale au cours de l'accouchement. Ces déformations sont produites par les forces exercées sur elle à la suite des pressions intra-utérines et de l'application des forceps. De manière plus générale, l'objectif de cette thèse résidait dans la conception d'un modèle physique à la fois précis et efficace, afin de simuler le comportement mécanique d'un objet déformable incompressible soumis à des pressions. A noter que cet encadrement de thèse m'a également permis de débiter une collaboration active avec Emmanuel Promayon de l'équipe GMCAO du laboratoire TIMC-IMAG de Grenoble, en effectuant ce co-encadrement de thèse avec lui. Par ailleurs, j'ai fait le choix de détailler la mise en place du modèle CST-DKT, ainsi que la formulation co-rotationnelle associée pour la simulation, car de mon point de vue, un certain nombre d'explications manque dans le manuscrit de thèse de Mathieu Bailet et il était donc nécessaire de le mettre par écrit pour la poursuite de ce travail de recherche.

Sommaire

6.1	Contexte de ce travail de recherche	164
6.2	Approches existantes sur les structures fines	165
6.3	Modèle d'une plaque mince par éléments finis CST-DKT	167
6.4	Formulation co-rotationnelle pour les éléments CST-DKT	183
6.5	Contrainte de préservation de volume	187
6.6	Simulation parallèle par coloriage de graphes	191
6.7	Évaluation du modèle CST-DKT et de sa parallélisation	194
6.8	Discussion	198

6.1 Contexte de ce travail de recherche

Dans le cadre du développement de simulateurs basés sur les technologies de la Réalité Virtuelle et dédiés à l'apprentissage de gestes médicaux-chirurgicaux, il est primordial de réussir à simuler le comportement des organes de manière réaliste, tout en assurant un temps de calcul compatible avec l'emploi d'une interface haptique. Il s'agit ainsi de relever deux challenges au niveau de la simulation numérique : la précision et le temps de calculs. Je présente ici le travail de doctorat de Mathieu Bailet (2014), qui rentrait dans ce type de problématique avec la conception d'un simulateur pour l'apprentissage des gestes médicaux liés à l'accouchement (détaillé dans le chapitre 8). Ce travail de recherche a été co-encadré par Emmanuel Promayon de l'équipe GMCAO du laboratoire TIMC-IMAG de Grenoble et moi-même, et a été financé par l'ARC 6 de la région Rhône-Alpes

De manière plus précise, l'objectif de la thèse de Mathieu Bailet consistait à proposer un modèle mécanique permettant de simuler en temps interactif, les déformations subies par le crâne du fœtus durant l'accouchement. Ces déformations sont issues des pressions intra-utérines exercées sur le crâne et de l'usage potentiel de forceps pour l'extraction du fœtus. D'un point de vue mécanique, le crâne peut être vu comme un objet déformable incompressible avec une structure fine. C'est pourquoi, l'approche adoptée par Mathieu s'est orientée vers l'utilisation d'un modèle surfacique. Ce type de modèle permet une modélisation des objets avec un nombre de degrés de libertés moindre que celui issu d'un modèle volumique. L'enjeu réside à faire en sorte que ce type de modèle puisse reproduire finement le comportement complexe d'un objet volumique en garantissant la conservation de son volume malgré les déformations subies durant la simulation. Dans ce contexte, Mathieu Bailet a proposé une modélisation basée sur les éléments finis coques, qui sont des éléments finis surfaciques issus de la théorie des plaques minces (dont les bases sont rappelées dans le chapitre 2). Cette modélisation est associée à une méthode de préservation de volume assurant la conservation de celui-ci au cours de la simulation, et à une approche originale de parallélisation réduisant les temps de calculs de la simulation numérique.

Organisation de ce chapitre : Ce chapitre se poursuit de la façon suivante :

- La section 6.2 aborde les approches existantes pour la simulation de structures fines.
- La section 6.3 débute en rappelant brièvement la modélisation des plaques minces et le principe de la méthode des éléments finis (MEF). Puis, cette section présente la modélisation qui a été adoptée pour la résolution de l'équation du mouvement d'une plaque mince. Elle s'appuie sur une discrétisation de la plaque en éléments finis appelés CST-DKT, combinant les éléments finis CST (*Constant Strain Triangle*) aux éléments finis DKT (*Discrete Kirchhoff Triangle*).
- La section 6.4 présente la formulation co-rotationnelle que nous avons employée afin d'utiliser notre modèle en grands déplacements et en grandes rotations malgré l'emploi de la loi de comportement linéaire de Hooke.
- La section 6.5 propose la méthode de préservation de volume que nous avons associée à la modélisation par éléments finis, afin de pouvoir simuler le comportement d'objets incompressibles. A noter que cette méthode peut être associée à d'autres contraintes, comme celles relatives aux traitements des collisions entre les objets de la simulation.
- La section 6.6 présente l'approche par coloriage de graphes employée pour la parallélisation des algorithmes de la simulation réduisant ses temps de calculs.
- La section 6.7 présente une évaluation du modèle CST-DKT en précision et en performance, avant de dresser un bilan de ce travail de recherche dans la section 6.8.

6.2 Approches existantes sur les structures fines

Dans le cadre du développement de simulateurs pour l'apprentissage de gestes médicaux-chirurgicaux, l'objectif est de coupler la simulation médicale à un dispositif haptique, afin de permettre à l'utilisateur d'interagir avec celle-ci. Dans ce contexte, nous devons simuler de manière précise le comportement des organes en interaction avec d'autres organes ou avec des instruments médicaux, en un temps interactif. Afin de répondre à cette contrainte, les modèles physiques et les algorithmes associés qui ont été proposés au fil des années dans le domaine de l'Informatique Graphique, offrent un compromis entre le temps de calculs et la précision numérique de la simulation dynamique. Ce compromis est ajustable en fonction des besoins de l'application visée (Bro-Nielsen (1998)). Au niveau des modèles physiques employés, deux approches pré-dominent : une *approche continue* visant la résolution des équations issues de la MMC, et une *approche discrète* basée sur une représentation non continue du matériau sous la forme d'un ensemble de particules en interaction.

En ce qui concerne les travaux sur **les modèles discrets**, le système masses-ressorts présenté dans le chapitre 3, a été largement employé en Informatique Graphique de part sa simplicité d'implémentation et sa rapidité de calculs, notamment dans le cadre de la simulation de textiles (Baraff and Witkin (1998)). Mais le comportement reproduit par le système est dépendant de sa topologie, c'est-à-dire qu'il dépend grandement de la configuration des connections établies entre les masses par les ressorts. De plus, pour pouvoir utiliser ce modèle dans le cadre d'applications médicales, une difficulté réside dans l'établissement de la formulation des raideurs des ressorts afin qu'ils intègrent les propriétés mécaniques du matériau (notamment le module de Young et le coefficient de Poisson du matériau). Mais des travaux¹ tendent à améliorer cet aspect-là (Baudet et al. (2009); Delingette (2008a,b); Kot and Nagahashi (2017); Natsupakpong and Çavuşoğlu (2010)).

Dans le cadre plus particulier des travaux relatifs aux **modèles surfaciques**, Grinspun et al. (2003) proposent une modélisation discrète des coques avec une implémentation basée sur le modèle masses-ressorts de Baraff and Witkin (1998). Ils mettent en avant la proposition d'un modèle simple à utiliser par rapport au modèle continu usuel des coques. L'idée est de modifier le modèle discret de Baraff avec une autre formulation de l'énergie. Pour cela, ils considèrent séparément l'énergie non-linéaire de membrane et l'énergie de flexion. Celle-ci est calculée, en considérant dans la configuration initiale et celle déformée, les angles diédraux des arêtes du maillage triangulaire de l'objet. L'énergie de membrane est calculée, quant à elle, à partir des aires des triangles de la discrétisation. Cette formulation a le mérite d'être invariante par rotation et donc utilisable dans le cas de grands déplacements. Mais comme pour un système masses-ressorts classique, le comportement de ce modèle dépend de la discrétisation géométrique, c'est-à-dire du maillage.

Choi et al. (2007) utilisent la même approche que Grinspun et al. (2003) avec une modélisation discrète des coques ayant la même formulation pour l'énergie de flexion. Ils introduisent ensuite une technique d'analyse modale, appelée *modal warping*, permettant d'accélérer le calcul de la simulation numérique. Cette analyse modale est effectuée au niveau des déplacements et des rotations pour prendre en compte tous les composants de la déformation de la coque. Ils obtiennent ainsi des simulations en temps réel.

Hammer et al. (2011) proposent l'emploi d'un système masses-ressorts afin de simuler la déformation de la valve mitrale qui est la valve cardiaque qui sépare l'atrium gauche du ventricule gauche. L'idée est de réaliser une simulation à moindre coût de calcul que le

1. Nous verrons dans le chapitre 7 les résultats que nous avons obtenus sur cette thématique de recherche durant le doctorat de Karolina Golec (2018).

modèle éléments finis. Pour cela, ils emploient la formulation des raideurs de Van Gelder (1998) intégrant le coefficient de Poisson du matériau et l'aires des triangles du maillage de l'objet. Pour le calcul des forces engendrées par les ressorts, deux formulations sont employées en fonction de l'étirement avec des raideurs considérés en régime forcé ou non. Ils obtiennent une simulation dix fois plus rapide que la simulation éléments finis d'une membrane. Par contre, le modèle masses-ressorts a une précision moindre en flexion et une erreur moyenne de 80% en cisaillement.

En ce qui concerne les **modèles continus**, rappelons que la résolution des équations de la Mécanique des Milieux Continus par la méthode des éléments finis est basée sur une discrétisation spatiale de l'objet en éléments finis (EF). Différents types d'EF ont été proposés pour cette discrétisation.

- Les **éléments finis volumiques** permettent des déformations en traction et en cisaillement sans aucune hypothèse restrictive. Par contre, pour des déformations en flexion, la modélisation nécessite beaucoup d'éléments, rendant la simulation coûteuse en temps de calculs. Ce type d'élément fini n'est ainsi pas adapté pour la modélisation de structures fines qui nous intéresse.
- Les **éléments de membranes** sont des éléments plans avec une épaisseur très petite par rapport aux deux autres dimensions. De plus, ce sont des éléments bidimensionnelles en état plan de contrainte (*i.e.* la contrainte s'exerce nécessairement dans un plan, avec une contrainte normale nulle). Au final, les déformations s'effectuent uniquement dans le plan, sans déformation en flexion. Ces éléments sont adaptés pour modéliser des structures fines travaillant dans le plan.
- Les **plaques** ont aussi une épaisseur très petite par rapport aux deux autres dimensions, permettant la modélisation de structures fines. Par rapport aux éléments précédents, les plaques ont des déformations en flexion avec une théorie restreinte aux petites déformations. Mais, les éléments de plaques sont des éléments qui sont plan au repos et ne permettent pas de déformation de membrane.
- Si nous souhaitons modéliser des éléments avec une courbure au repos, nous parlons alors de **coques**. Ces éléments sont bien adaptés pour modéliser des déformations complexes avec des déformations de membrane et de flexion qui sont découplés.

Par rapport au contexte applicatif dans lequel s'inscrivait la thèse de Mathieu Bailet, l'objectif était de proposer un modèle permettant de simuler les déformations subies par le crâne du fœtus durant l'accouchement. Celui-ci peut être considéré comme une **structure fine composée de matériaux de rigidités différentes** (os, tissus mous), avec une matière intra-cranienne vue comme des tissus mous quasi-incompressibles.

Par rapport aux deux approches employées en simulation, nous nous sommes orientés vers un **modèle continu**, puisqu'il assure une modélisation physique du matériau prenant en compte ses propriétés mécaniques. Pour la discrétisation employée dans le cadre de la méthode des éléments finis, nous avons choisi d'utiliser les **éléments finis coques basés sur la théorie de Kirchhoff-Love pour les plaques minces** permettant de reproduire la flexion de la structure fine. Nous verrons que nous avons également ajouté une méthode permettant de **contrôler le volume de l'objet** simulé afin de garantir son incompressibilité, critère important dans le cadre de la simulation de tissus mous. A cela s'ajoute une **formulation co-rotationnelle** du modèle afin de pouvoir prendre en compte des grands déplacements (non-linéarités géométriques).

6.3 Modèle d'une plaque mince par éléments finis CST-DKT

Dans le chapitre 2, nous avons présenté en détails la théorie des plaques minces soumises aux hypothèses de Kirchhoff-Love, et dans la section 1.10 du chapitre 1, nous avons vu la formulation discrète du problème issu de la dynamique newtonienne décrivant le mouvement d'un objet, ainsi que sa résolution par la méthode des éléments finis (MEF). Nous faisons ici un bref rappel des formulations qui vont nous être utiles dans ce chapitre en ce qui concerne la théorie des coques, ainsi qu'un résumé du principe général de la MEF et des quantités élémentaires à définir pour la résolution de l'équation du mouvement.

Rappel - Théorie de Kirchhoff-Love des plaques minces

Cinématique. Soit $\vec{u}_0 = (u_0(x, y, z), v_0(x, y, z), w_0(x, y, z))^T$ le déplacement par translation du plan moyen de la plaque mince. Nous considérons que ce plan moyen est positionné en $z = 0$, et que u_0, v_0 correspondent à son déplacement dans le plan $(0, x, y)$ et w_0 à celui dans la direction z . Nous notons θ_x, θ_y ses rotations par rapport aux axes x et y .

Le déplacement \vec{u} de la plaque mince est alors défini par :

$$\vec{u} = \begin{pmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{pmatrix} = \begin{pmatrix} u_0(x, y) \\ v_0(x, y) \\ w_0(x, y) \end{pmatrix} + z \begin{pmatrix} \theta_y(x, y) \\ -\theta_x(x, y) \\ 0 \end{pmatrix}. \quad (6.1)$$

Tenseur de déformation sous les hypothèses de Kirchhoff-Love. Étant dans le cadre de petites déformations avec de petits déplacements, le tenseur de déformation de Green-Lagrange linéarisé est utilisé pour modéliser la déformation de la plaque mince en fonction de son déplacement. L'application des hypothèses de Kirchhoff-Love simplifie ensuite un certain nombre des composantes du tenseur de déformation.

Au final, le tenseur ϵ_c des déformations de cisaillement est défini par :

$$\epsilon_c = \begin{pmatrix} \frac{\partial w_0(x, y)}{\partial x} + \theta_y(x, y) \\ \frac{\partial w_0(x, y)}{\partial y} - \theta_x(x, y) \end{pmatrix} = \begin{pmatrix} w_{0,x} + \theta_y \\ w_{0,y} - \theta_x \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (6.2)$$

et le tenseur global de déformation d'une plaque mince est défini par :

$$\epsilon^g = \begin{pmatrix} \frac{\partial u_0(x, y)}{\partial x} \\ \frac{\partial v_0(x, y)}{\partial y} \\ \frac{1}{2} \left(\frac{\partial u_0(x, y)}{\partial y} + \frac{\partial v_0(x, y)}{\partial x} \right) \end{pmatrix} + z \begin{pmatrix} \frac{\partial \theta_y(x, y)}{\partial x} \\ -\frac{\partial \theta_x(x, y)}{\partial y} \\ \frac{1}{2} \left(\frac{\partial \theta_y(x, y)}{\partial y} - \frac{\partial \theta_x(x, y)}{\partial x} \right) \end{pmatrix} \quad (6.3)$$

$$= \underbrace{\begin{pmatrix} u_{0,x} \\ v_{0,y} \\ \frac{1}{2} (u_{0,y} + v_{0,x}) \end{pmatrix}}_{\epsilon_m} + z \underbrace{\begin{pmatrix} \theta_{y,x} \\ -\theta_{x,y} \\ \frac{1}{2} (\theta_{y,y} - \theta_{x,x}) \end{pmatrix}}_{\epsilon_b} \quad (6.4)$$

où ϵ_m correspond au tenseur de déformation membranaires (translation dans le plan (x, y)) et ϵ_b au tenseur de déformation de flexion (rotation autour des axes x et y).

Tenseur de contraintes et loi de comportement. Soient E, ν le module de Young et le coefficient de Poisson de la plaque mince. La loi de Hooke, reliant le tenseur des contraintes au tenseur de déformation, s'écrit alors de la façon suivante pour une plaque mince :

$$\underbrace{\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix}}_{\underline{\underline{\sigma}}} = \frac{E}{1-\nu^2} \underbrace{\begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix}}_{\underline{\underline{C}}} : \underbrace{(\epsilon_m + z \epsilon_b)}_{\underline{\underline{\epsilon^g}}}. \quad (6.5)$$

Et sa relation inversée se formule ainsi :

$$\underbrace{\begin{pmatrix} \epsilon_{xx}^g \\ \epsilon_{yy}^g \\ \epsilon_{xy}^g \end{pmatrix}}_{\underline{\underline{\epsilon^g}}} = \frac{1}{E} \underbrace{\begin{pmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 1+\nu \end{pmatrix}}_{\underline{\underline{D}}} \underbrace{\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix}}_{\underline{\underline{\sigma}}}. \quad (6.6)$$

Énergie de déformation. Au final, l'énergie de déformation d'une plaque mince d'épaisseur h et de surface A_t pour son plan moyen, est définie par la somme de l'énergie de déformation membranaire et de l'énergie de déformation de flexion avec :

$$W = \frac{1}{2} \int_{A_t} \left(\epsilon_m^T \frac{E h}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \epsilon_m + \epsilon_b^T \frac{E h^3}{12(1-\nu^2)} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \epsilon_b \right) dx dy. \quad (6.7)$$

Pour modéliser son comportement mécanique, nous devons ainsi réussir à modéliser à la fois son comportement membranaire et celui en flexion.

Rappel - Résolution par la Méthode des Éléments Finis

Principe de la MEF. La méthode des EF repose sur la discrétisation de l'objet continu en un ensemble d'éléments finis. Pour une surface définie en 2D, nous pouvons ainsi définir cette surface comme un ensemble de triangles 2D. La MEF se base ensuite sur la construction pour chacun de ces éléments d'une approximation \mathbf{u}^* du champ des variables \mathbf{u} que nous recherchons. Cette approximation est établie à partir des valeurs obtenues aux nœuds de chacun des éléments. Ensuite, les solutions élémentaires calculées sur chacun des éléments sont assemblées pour obtenir la solution complète du problème.

Soit \mathbf{N} la matrice ligne des fonctions d'interpolation de l'élément fini D_e considéré. Pour tout point M de cet élément, la valeur de la fonction approchée est ainsi définie en fonction des variables nodales \mathbf{u}_n relatives aux nœuds d'interpolation de l'élément par :

$$\forall M \in D_e, \mathbf{u}^*(M) = \mathbf{N}(M) \mathbf{u}_n.$$

Cette approximation nodale est généralement construite sur des éléments dits de références ayant une géométrie simple, sur lesquels les calculs sont faciles à effectuer. Une transformation géométrique est ensuite employée pour passer d'un point de l'élément de référence défini en 2D dans le repère $\mathbf{r} = (s, t)$ à un point de l'élément réel (ayant la même topologie que l'élément de référence) défini en 2D dans le repère $\mathbf{x} = (x, y)$.

Dans le cadre de la dynamique newtonienne, l'enjeu réside dans le calcul du déplacement de tout point de l'objet considéré. Après avoir discrétisé l'objet en un ensemble d'éléments finis, nous construisons ainsi pour chacun des éléments finis, l'approximation nodale \mathbf{u}^* pour exprimer le champ de déplacement \vec{u} en chacun des points de ces éléments, en fonction des valeurs approchées obtenues aux nœuds des éléments. Cette approximation nodale sert ensuite à construire des quantités élémentaires relatives au problème de la dynamique newtonienne. Elles seront ensuite assemblées pour obtenir l'équation matricielle complète relative au mouvement de l'objet, qui sera résolue par une méthode d'intégration numérique, pour obtenir au final le champ de déplacement \vec{u} en tout point de l'objet.

Quantités élémentaires pour chaque élément. En utilisant le Principe des Travaux Virtuels (PTV), nous avons vu que la formulation intégrale du problème de la dynamique newtonienne s'exprime sous la forme matricielle suivante pour chacun des éléments finis :

$$\mathbf{M}_e \ddot{\mathbf{u}}_n + \mathbf{K}_e \mathbf{u}_n = \mathbf{F}_{de}.$$

Cette formulation nécessite la définition des formes matricielles et vectorielles élémentaires suivantes pour chaque élément D_e de masse volumique ρ issu de la discrétisation de l'objet :

- la **matrice élémentaire des masses** définie par : $\mathbf{M}_e = \int_{D_e} (\mathbf{N}(M)^T \rho \mathbf{N}(M)) dv$
- la **matrice élémentaire des raideurs** définie par : $\mathbf{K}_e = \int_{D_e} (\mathbf{B}(M)^T \mathbf{C}(M) \mathbf{B}(M)) dv$
- sachant que le **tenseur des déformations** est défini par :

$$\epsilon(M) = \mathbf{L} \vec{u}(M) = \mathbf{L} \mathbf{N}(M) \mathbf{u}_n = \mathbf{B}(M) \mathbf{u}_n$$

où \mathbf{L} est la matrice d'opérateurs différentiels correspondant à l'expression du gradient symétrique du champ de déplacements, et \mathbf{B} est la matrice d'opérateurs différentiels appliquées aux fonctions d'interpolations \mathbf{N} .

- et que la **loi de comportement** est définie par :

$$\sigma(M) = \mathbf{C}(M) \epsilon(M) = \mathbf{C}(M) \mathbf{B}(M) \mathbf{u}_n$$

- le **vecteur des forces appliquées sur l'élément** défini par :

$$\mathbf{F}_{de} = \int_{D_e} (\mathbf{N}(M)^T \rho \mathbf{f}_{dext}) dv + \int_{\partial D_e} (\mathbf{N}(M)^T \mathbf{T}_d) ds$$

où \mathbf{f}_{dext} et \mathbf{T}_d sont les vecteurs de forces extérieures et de contraintes pour l'élément.

L'assemblage de ces quantités élémentaires permet ensuite la construction de l'équation matricielle globale de la dynamique newtonienne s'exprimant sous la forme :

$$\mathbf{M} \ddot{\mathbf{U}} + \mathbf{K} \mathbf{U} = \mathbf{F}_d$$

Modélisation avec des éléments finis CST-DKT

Nous venons de rappeler que pour résoudre par la méthode des éléments finis, l'équation du mouvement d'un objet, nous devons le discrétiser en un ensemble d'éléments. Afin de réduire les coûts en calcul, nous avons tout d'abord choisi de modéliser les objets

uniquement par leur surface. Pour la discrétisation de celle-ci, nous avons alors choisi d'utiliser deux types d'EFs de référence de forme triangulaire : les triangles T3 ou CST (*Constant Strain Triangle*) modélisant un comportement membranaire (se traduisant par la translation de la surface dans le plan $(0, x, y)$), et les triangles DKT (*Discrete Kirchhoff Triangle*) modélisant une flexion (se traduisant par une rotation de la surface autour des axes). Au final, chaque élément triangulaire de la discrétisation de la surface correspond à un élément de référence appelé CST-DKT (Bailet et al. (2013, 2014, 2016)). La combinaison de ces deux types d'éléments finis nous permet de modéliser à la fois un comportement membranaire et un comportement en flexion, c'est-à-dire qu'ils permettent de modéliser le comportement mécanique d'une plaque mince soumise aux hypothèses de Kirchhoff-Love.

Les éléments finis CST-DKT ont déjà été employés en sciences appliquées et des articles donnent une comparaison de leurs résultats avec d'autres types d'EF. Dans Belytschko et al. (1999), une comparaison a été effectuée dans le cadre d'une modélisation en éléments finis coques avec trois tests permettant d'observer la qualité des résultats de simulation en comparaison avec la solution analytique correspondante. Plusieurs EF triangulaires basés sur la formulation discrète des éléments DKT (*Discrete Kirchhoff Triangle*) de Batoz et al. (1980) ont été employés. Ces éléments finis différaient par la formulation employée pour le champs de déformation membranaire. Les éléments finis CST-DKT se sont révélés avoir un bon comportement sauf pour le problème de Scordelis-Lo déterminant la capacité d'un élément à résoudre avec précision les états complexes de déformation membranaire.

Dans Gil Rama and Zehn, une comparaison est faite entre les formulations des éléments finis CS-DSG3, CST-DKT et les S3 proposés dans le progiciel Abaqus. Ils ont effectué un certain nombre de tests en statique aussi bien dans le cas linéaire que non-linéaire. Par contre, la comparaison avec les éléments CST-DKT n'a été réalisée que dans le cas linéaire puisqu'ils ont proposé une formulation co-rotationnelle que dans le cadre des EF CS-DSG3. Dans le cadre linéaire, les éléments finis CST-DKT présentent de bons résultats.

Bartels (2013) présentent une étude faite pour simuler la déformation d'une plaque carrée subissant un chargement vertical, ainsi que la compression d'une bande. Pour cela, la MEF a été employée avec une discrétisation basée sur l'emploi des triangles DKT de Batoz et al. (1980). Nous pouvons ainsi citer plusieurs papiers employant des éléments DKT ou CST ou encore une combinaison des deux dans le domaine des sciences appliquées.

A notre connaissance, cette discrétisation en éléments CST-DKT n'avait pas encore été employée dans le domaine de l'informatique graphique. Nous avons donc testé son usage dans le cadre d'une application médicale avec la simulation des déformations subies par la tête fœtale lors d'un accouchement, c'est-à-dire dans le cadre de la simulation d'un objet déformable incompressible subissant des déformations membranaires et en flexion.

Élément CST - Constant Strain Triangle²

Considérons comme élément fini de référence, le triangle CST défini dans le repère $\mathbf{r} = (s, t)$ par les nœuds de coordonnées $(0, 0)$, $(1, 0)$, $(0, 1)$. Comme illustré sur la Fig. 6.1, cet élément est un triangle rectangle. Une transformation géométrique $\mathbf{x}(\mathbf{r})$ est ensuite utilisée pour passer de cet élément de référence à un élément triangulaire réel défini par 3 nœuds de coordonnées (x_1, y_1) , (x_2, y_2) , (x_3, y_3) dans un repère local $\mathbf{x} = (x, y)$ positionné en son barycentre G . Par ailleurs, chacun des nœuds i (pour $i = 1, 2, 3$) de l'élément réel dispose de deux degrés de libertés (u_i, v_i) - soit 6 degrés de liberté au total pour l'élément - permettant la modélisation du déplacement des nœuds dans le plan $(0, x, y)$ suite aux

2. Vous pouvez retrouver plus de détails sur ce type d'EF dans la section 1.10 du chapitre 1.

efforts exercés dans ce plan, c'est-à-dire la modélisation du comportement membranaire de la plaque. Nous faisons ensuite correspondre les valeurs des nœuds de l'élément de référence avec celles des nœuds de l'élément réel. Au final, la transformation géométrique $\mathbf{x}(\mathbf{r})$ permet de faire en sorte que chaque point (s, t) de l'élément fini de référence ait un correspondant (x, y) sur l'élément réel.

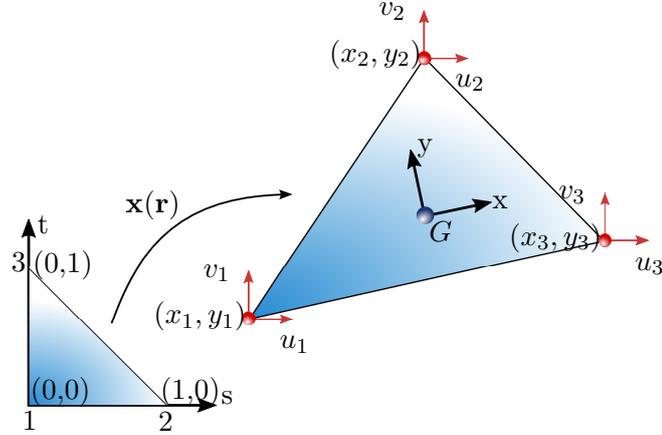


Figure 6.1 – (A gauche) L'élément fini de référence appelé T3 ou CST dans le repère $\mathbf{r} = (s, t)$ en correspondance (à droite) avec un élément fini triangulaire défini dans le repère réel $\mathbf{x} = (x, y)$.

L'approximation du champ de déplacement $\mathbf{u}^*(s, t)$ en tout point de l'élément de référence est ensuite obtenue en utilisant une interpolation linéaire des valeurs de ses nœuds. Notons $\mathbf{u}_i = (u_i, v_i)$ la valeur calculée au nœud i (pour $i = 1, 2, 3$) de l'élément réel. En faisant correspondre les valeurs des nœuds de l'élément de référence avec celles des nœuds de l'élément réel, nous obtenons pour tout point M de l'élément de référence :

$$\begin{aligned} \mathbf{u}^*(s, t) &= \begin{pmatrix} u_0(s, t) \\ v_0(s, t) \end{pmatrix} = \mathbf{u}^*(0, 0) N_1(s, t) + \mathbf{u}^*(1, 0) N_2(s, t) + \mathbf{u}^*(0, 1) N_3(s, t) \\ &= \mathbf{u}_1 N_1(s, t) + \mathbf{u}_2 N_2(s, t) + \mathbf{u}_3 N_3(s, t) \\ &= \underbrace{\begin{bmatrix} N_1(s, t) & 0 & N_2(s, t) & 0 & N_3(s, t) & 0 \\ 0 & N_1(s, t) & 0 & N_2(s, t) & 0 & N_3(s, t) \end{bmatrix}}_{\mathbf{N}(s, t)} \underbrace{\begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{pmatrix}}_{\mathbf{u}_m} \end{aligned}$$

où \mathbf{u}_m correspond au vecteur des variables nodales des nœuds d'interpolation de l'élément réel et $\mathbf{N}(s, t)$ correspond à la matrice ligne des fonctions d'interpolation $N_i(s, t)$ attachées aux nœuds i de l'élément de référence. Les fonctions $N_i(s, t)$ (pour $i = 1, 2, 3$) sont ainsi appelées fonctions de forme (ou fonctions d'interpolation linéaire). En prenant la base linéaire $(1, s, t)$ comme base polynomiale dans le repère $\mathbf{r} = (s, t)$, nous avons vu que ces fonctions d'interpolation étaient définies par :

$$N_1(s, t) = 1 - s - t, \quad N_2(s, t) = s, \quad N_3(s, t) = t.$$

Quantités élémentaires pour l'élément CST. Lors de l'emploi de la MEF, nous devons définir la matrice élémentaire de raideur relative aux éléments finis. Considérons un

élément réel D_e de surface A_e et d'épaisseur h . La matrice élémentaire de raideur relative au comportement membranaire modélisée par un élément CST est définie par :

$$\begin{aligned} \mathbf{K}_{me} &= \int_{D_e} \mathbf{B}_m(M)^T \mathbf{C}(M) \mathbf{B}_m(M) dx dy dz \\ &= h \int_{A_e} \mathbf{B}_m(M)^T \mathbf{C}(M) \mathbf{B}_m(M) dx dy \\ &= h \int_{t=0}^1 \int_{s=0}^{1-s} \mathbf{B}_m(M)^T \mathbf{C}(M) \mathbf{B}_m(M) \det(\mathbf{J}) ds dt \end{aligned}$$

où :

— pour la loi de Hooke, $\mathbf{C}(M) = \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix}$.

- \mathbf{J} est la matrice jacobienne de la transformation géométrique permettant le passage de l'élément de référence à l'élément réel. Dans le chapitre 1, nous avons vu que l'élément CST était iso-paramétrique et que la matrice jacobienne de la déformation \mathbf{J} et son inverse \mathbf{J}^{-1} étaient définies pour cet élément fini par :

$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \mathbf{r}} = \begin{bmatrix} x_{21} & y_{21} \\ x_{31} & y_{31} \end{bmatrix} \quad \text{et} \quad \mathbf{J}^{-1} = \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \frac{1}{2A} \begin{bmatrix} y_{31} & -y_{21} \\ -x_{31} & x_{21} \end{bmatrix},$$

où $x_{ij} = x_i - x_j$ et $y_{ij} = y_i - y_j$, A est l'aire de l'élément réel avec $\det(\mathbf{J}) = 2A = x_{21} y_{31} - x_{31} y_{21}$.

- \mathbf{B}_m est la matrice d'opérateurs différentiels appliquées aux fonctions d'interpolation de l'élément fini avec $\mathbf{B}_m = \mathbf{L} \mathbf{N}(s, t)$ où \mathbf{L} est la matrice d'opérateurs différentiels correspondant à l'expression du gradient du champ de déplacement. Dans le chapitre 1, nous avons vu que cette matrice était définie par :

$$\mathbf{B}_m = \frac{1}{2A} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix}.$$

- Par ailleurs, rappelons que le tenseur des déformations membranaires est défini par $\epsilon_m = \mathbf{B}_m \mathbf{u}_m$ donnant comme relation :

$$\epsilon_m = \begin{pmatrix} e_{xx} \\ e_{yy} \\ 2 e_{xy} \end{pmatrix} = \underbrace{\begin{pmatrix} u_{0,x} \\ v_{0,y} \\ u_{0,y} + v_{0,x} \end{pmatrix}}_{\text{cf. équation (6.4)}} = \frac{1}{2A} \underbrace{\begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix}}_{\mathbf{B}_m} \underbrace{\begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{pmatrix}}_{\mathbf{u}_m}.$$

La Fig. 6.2 illustre la taille de la matrice \mathbf{K}_{me} par la présentation des matrices à multiplier entre elles lors de son calcul. Cette matrice est de taille globale 6×6 . Elle est composée des éléments K_{me}^{ij} qui sont de taille 2×2 (pour $i, j = 1, 2, 3$).

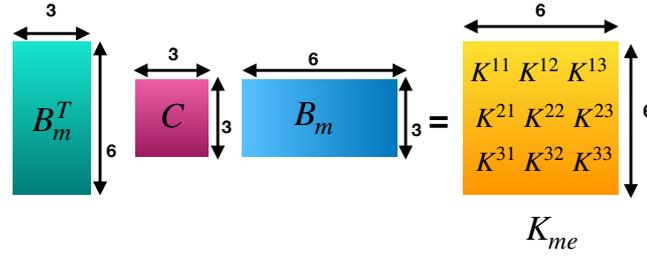


Figure 6.2 – Illustration de la matrice élémentaire K_{me} d'un élément fini CST.

Notons que l'intégrale de \mathbf{K}_{me} est calculée directement puisque l'interpolation linéaire employée pour les déplacements implique des déformations constantes sur D_e . Nous avons :

$$\mathbf{K}_{me} = \frac{1}{2A} \underbrace{\begin{bmatrix} y_{23} & 0 & x_{32} \\ 0 & x_{32} & y_{23} \\ y_{31} & 0 & x_{13} \\ 0 & x_{13} & y_{31} \\ y_{12} & 0 & x_{21} \\ 0 & x_{21} & y_{12} \end{bmatrix}}_{\mathbf{B}_m^T} \underbrace{\frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix}}_{\mathbf{C}} \frac{1}{2A} \underbrace{\begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix}}_{\mathbf{B}_m} h 2A (1-s)$$

$$= \frac{E h (1-s)}{2A (1-\nu^2)} \begin{bmatrix} y_{23} & 0 & x_{32} \\ 0 & x_{32} & y_{23} \\ y_{31} & 0 & x_{13} \\ 0 & x_{13} & y_{31} \\ y_{12} & 0 & x_{21} \\ 0 & x_{21} & y_{12} \end{bmatrix} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix}.$$

Élément DKT - Discrete Kirchhoff Triangle

Désormais, considérons comme EF de référence, le triangle DKT proposé par [Batoz et al. \(1980\)](#). Comme illustré sur la Fig. 6.3, cet élément est défini dans le repère $\mathbf{r} = (s, t)$ par 6 nœuds formant un triangle rectangle : 3 nœuds de coordonnées $(0, 0)$, $(1, 0)$, $(0, 1)$ situés aux sommets du triangle et 3 nœuds de coordonnées $(1/2, 1/2)$, $(0, 1/2)$, $(1/2, 0)$ situés au milieu des arêtes. Une transformation géométrique $\mathbf{x}(\mathbf{r})$ est ensuite utilisée pour passer de cet élément de référence à un élément réel. Ce dernier est défini par 6 nœuds i de coordonnées (x_i, y_i) : 3 nœuds situés aux sommets du triangle (pour $i = 1, 2, 3$) et 3 nœuds situés au milieu des arêtes (pour $i = 4, 5, 6$). Les coordonnées de l'élément réel sont exprimées dans un repère local $\mathbf{x} = (x, y)$ positionné au niveau du barycentre G de l'élément réel. De plus, chacun des nœuds situés aux sommets de l'élément réel disposent de 3 degrés de liberté notés $(w_i, \theta_{x_i}, \theta_{y_i})$ avec $i = 1, 2, 3$. Soit 9 degrés de liberté au total pour cet élément. Ces degrés de liberté permettent la modélisation d'une rotation de la plaque autour des axes x et y , ainsi qu'un déplacement dans la direction z (direction perpendiculaire au plan de l'élément). Autrement dit, ils permettent de modéliser le comportement de flexion de la plaque issue des efforts exercés perpendiculairement au plan de l'élément. Nous faisons ensuite correspondre les valeurs des nœuds de l'élément de référence situés aux sommets avec celles des nœuds situés aux sommets de l'élément réel. Au final, la transformation géométrique $\mathbf{x}(\mathbf{r})$ permet de faire en sorte que chaque point (s, t) de l'élément fini de référence ait un correspondant (x, y) sur l'élément réel.

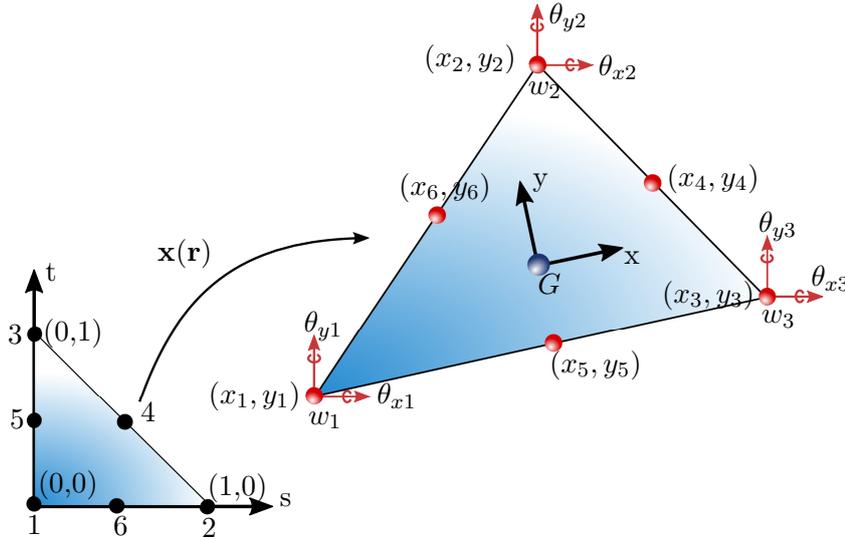


Figure 6.3 – (A gauche) L'élément fini de référence appelé DKT dans le repère $\mathbf{r} = (s, t)$ en correspondance (à droite) avec un élément fini triangulaire défini dans le repère réel $\mathbf{x} = (x, y)$.

Plusieurs hypothèses sont ensuite faites pour définir l'approximation du champ de déplacement $\mathbf{u}^*(s, t)$ en tout point de l'élément de référence en fonction des valeurs nodales notées $u_i = (w_i, \theta_{x_i}, \theta_{y_i})$ des nœuds i de l'élément réel (pour $i = 1, 2, 3$) situés aux sommets de celui-ci. Ces hypothèses permettent notamment de relier le déplacement w aux rotations θ_x, θ_y . En effet, si nous observons les équations que nous avons rappelées dans la section 6.3, issues de la théorie des plaques minces soumises aux hypothèses de Kirchhoff-Love, nous pouvons voir que le déplacement w n'intervient pas dans la formulation du tenseur de déformation de flexion. La formulation de ϵ_b (et ainsi donc de la matrice des raideurs) ne dépend en effet que des dérivées premières des rotations θ_x, θ_y . C'est pourquoi, il est important d'établir un lien entre le déplacement w et les rotations θ_x, θ_y pour prendre en compte l'ensemble des degrés de liberté de l'élément DKT si nous souhaitons nous en servir pour modéliser la déformation en flexion d'une plaque mince.

Pour présenter cela, nous reprenons les explications proposées dans l'article de [Batoz et al. \(1980\)](#). A noter que Batoz introduit la notation β_x, β_y (avec $\beta_x = -w_{,x} = \theta_y$ et $\beta_y = -w_{,y} = -\theta_x$) pour définir les rotations. Afin de simplifier la compréhension, nous utiliserons pour notre part uniquement la notation en θ .

H1 - Les rotations θ_x, θ_y varient de manière quadratique sur l'élément.

Une interpolation quadratique est effectuée pour définir les valeurs des rotations en tout point $M = (s, t)$ de l'élément de référence en fonction des valeurs aux 6 nœuds de l'élément. Nous avons ainsi :

$$\theta_x(s, t) = \sum_{i=1}^6 N_i(s, t) \theta_{xi} \quad \text{et} \quad \theta_y(s, t) = \sum_{i=1}^6 N_i(s, t) \theta_{yi} \quad (6.8)$$

où $N_i(s, t)$ représentent les fonctions de forme quadratiques attachées aux nœuds i et θ_{xi}, θ_{yi} les valeurs des rotations aux nœuds i pour $i \in \{1, \dots, 6\}$. En considérant comme base polynomiale la base quadratique $(1, s, t, s^2, st, t^2)$ dans le repère $\mathbf{r} = (s, t)$, les fonctions d'interpolation $N_i(s, t)$ de cet élément de référence sont données par :

$$\begin{aligned} N_i(s, t) &= L_i (2 L_i - 1) && \text{pour } i = 1, 2, 3 \\ N_{i+3}(s, t) &= 4 L_j L_k && \text{pour } i = 1, 2, 3 \text{ avec } j \neq i, k \neq j \end{aligned}$$

où $L_1 = 1 - s - t$, $L_2 = s$ et $L_3 = t$. Autrement dit, elles sont définies par :

$$\begin{cases} N_1(s, t) = 2(1 - s - t)(\frac{1}{2} - s - t) \\ N_2(s, t) = s(2s - 1) \\ N_3(s, t) = t(2t - 1) \\ N_4(s, t) = 4st \\ N_5(s, t) = 4t(1 - s - t) \\ N_6(s, t) = 4s(1 - s - t) \end{cases} \quad (6.9)$$

La Fig. 6.4 illustre les fonctions d'interpolation $N_1(s, t)$ et $N_5(s, t)$.

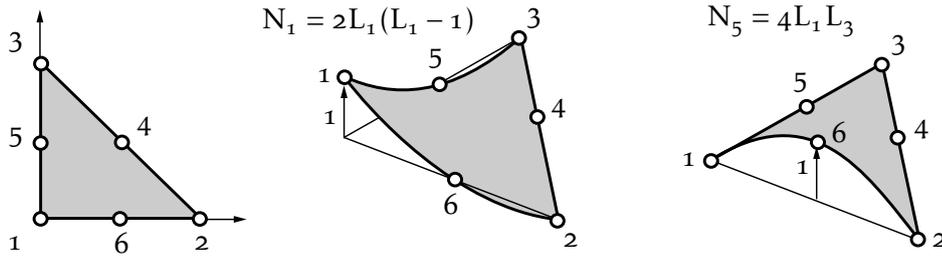


Figure 6.4 – Fonctions d'interpolation quadratique $N_1(s, t)$ et $N_5(s, t)$ de l'élément de référence T6 (triangle à 6 noeuds). Image issue de [Oudin \(2008\)](#).

Comme l'interpolation de l'équation (6.8) fait intervenir les valeurs des rotations θ_{xi}, θ_{yi} aux 6 noeuds de l'élément, alors qu'elles ne sont initialement définies que pour les noeuds i situés aux sommets du triangle (*i.e.* pour $i = 1, 2, 3$), nous devons établir un certain nombre de relations et d'hypothèses pour les définir aux noeuds i situés au milieu des arêtes (*i.e.* pour $i = 4, 5, 6$).

A cette fin, nous positionnons au milieu des arêtes, un repère 2D orthogonal de coordonnées curvilignes (s, n) . La Fig. 6.5(a) illustre ce repère. Nous notons :

- $k \in \{4, 5, 6\}$ le numéro du noeud situé au milieu de l'arête $ij \in \{23, 31, 12\}$,
- (x_k, y_k) les coordonnées du noeud k avec $x_k = \frac{1}{2}(x_i + x_j)$ et $y_k = \frac{1}{2}(y_i + y_j)$,
- l_{ij} la longueur de l'arête ij avec $l_{ij} = \sqrt{(x_{ij}^2 + y_{ij}^2)}$, $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$,
- s l'abscisse curviligne autour de l'arête ij considérée,
- \vec{n}_{ij} le vecteur du plan de l'élément, perpendiculaire à l'arête ij considérée,
- θ_{sk} la rotation autour de l'axe de l'abscisse curviligne s de la normale au feuillet moyen considérée au noeud k ,
- θ_{nk} la rotation autour de \vec{n}_{ij} de la normale au feuillet moyen considérée au noeud k .

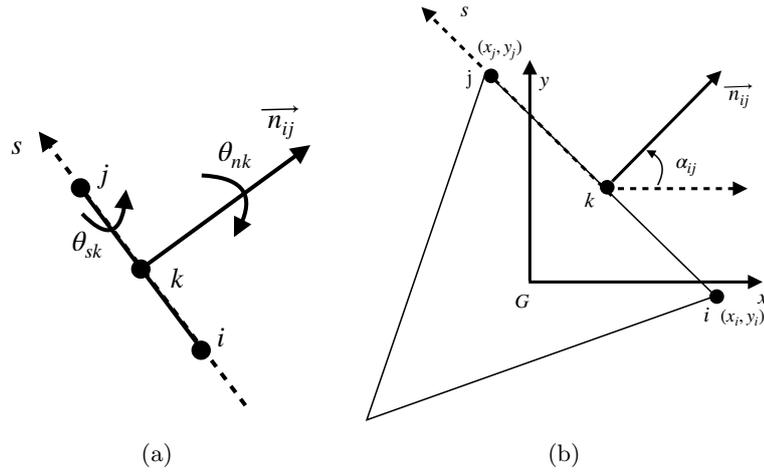


Figure 6.5 – (a) Repère curviligne positionné au point k milieu de l'arête ij . (b) Angle α_{ij} entre l'axe des abscisses du repère local $\mathbf{x} = (x, y)$ de l'élément DKT et la normale \vec{n}_{ij} à l'arête ij .

H2 - Tous les noeuds de l'élément répondent aux hypothèses de Kirchhoff.

Nous avons vu que le tenseur des déformations de cisaillement de la plaque mince était défini par :

$$\epsilon_c = \begin{pmatrix} w_{0,x} + \theta_y \\ w_{0,y} - \theta_x \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Pour les nœuds i situés aux sommets du triangle (*i.e.* pour $i = 1, 2, 3$) soumis aux hypothèses de Kirchhoff, nous avons ainsi comme relations :

$$\begin{cases} w_{0i,x} + \theta_{yi} = 0 \\ w_{0i,y} - \theta_{xi} = 0 \end{cases} \Rightarrow \begin{cases} \theta_{yi} = -w_{0i,x} \\ \theta_{xi} = w_{0i,y} \end{cases} \quad (6.10)$$

Pour les nœuds k situés au milieu des arêtes du triangle (*i.e.* pour $k = 4, 5, 6$) également soumis aux hypothèses de Kirchhoff, nous avons aussi comme relation :

$$-w_{k,s} + \theta_{nk} = 0 \Rightarrow \theta_{nk} = w_{k,s} \quad (6.11)$$

Ces relations permettent de relier les degrés de liberté correspondant au déplacement w aux degrés de liberté correspondants aux rotations θ .

H3 - Les variations des déplacements w sont cubiques le long des arêtes.

La variation du déplacement w le long d'une arête est approximée par un polynôme de degré 3 en s avec :

$$w(s) = a + b s + c s^2 + d s^3$$

où s est l'abscisse curviligne autour de l'arête considérée. Nous avons ainsi une variation cubique des w le long des arêtes, impliquant une variation quadratique de $\partial w / \partial s = w_{,s}$ le long des arêtes (et donc de la même façon, nous retrouvons bien une variation quadratique pour θ_n). Aux nœuds i et j nous avons $w(i) = w_i$ et $w(j) = w_j$. Nous en déduisons que pour le nœud k situé au milieu de l'arête ij de longueur l_{ij} , nous avons :

$$w_{k,s} = -\frac{3}{2} \frac{w_i}{l_{ij}} - \frac{1}{4} w_{i,s} + \frac{3}{2} \frac{w_j}{l_{ij}} - \frac{1}{4} w_{j,s}, \quad (6.12)$$

avec $w_{i,s} = \partial w(i)/\partial s$ et $w_{j,s} = \partial w(j)/\partial s$. Ainsi, nous exprimons les valeurs de $w_{,s}$ en k milieu de l'arête ij en fonction des valeurs de $w_{,s}$ en i et j . Par contre, notons que nous n'avons pas besoin de définir une fonction d'interpolation pour calculer w sur tout l'élément, la relation le long de l'arête (et ainsi sur le bord de l'élément) est suffisante.

H4 - La variation de θ_s est linéaire le long des arêtes.

La valeur de θ_s pour le nœud $k \in \{4, 5, 6\}$ situé au milieu de l'arête $ij \in \{23, 31, 12\}$ est calculée en fonction des valeurs en i et j par simple interpolation linéaire avec :

$$\theta_{sk} = \frac{1}{2}(\theta_{si} + \theta_{sj}). \quad (6.13)$$

Relations géométriques entre θ_x, θ_y et θ_n, θ_s

Nous avons ensuite besoin de relier les rotations θ_s, θ_n exprimées en fonction du repère orthogonal curviligne (s, n) , aux rotations θ_x, θ_y exprimées dans le repère local (x, y) de l'élément. La Fig. 6.5(b) illustre l'angle α_{ij} qui existe entre l'axe des abscisses du repère local $\mathbf{x} = (x, y)$ de l'élément DKT et la normale \vec{n} à l'arête ij . La Fig. 6.6 illustre le calcul de $\cos(\alpha_{ij}) = -y_{ij}/l_{ij}$ et $\sin(\alpha_{ij}) = x_{ij}/l_{ij}$ avec l_{ij} longueur de l'arête ij . Au final, nous avons les relations suivantes issues de la rotation d'un angle α_{ij} autour de z :

$$\begin{pmatrix} \theta_y \\ \theta_x \end{pmatrix} = \begin{pmatrix} \cos \alpha_{ij} & -\sin \alpha_{ij} \\ \sin \alpha_{ij} & \cos \alpha_{ij} \end{pmatrix} \begin{pmatrix} \theta_s \\ \theta_n \end{pmatrix} \quad (6.14)$$

$$\begin{pmatrix} w_{,n} \\ w_{,s} \end{pmatrix} = \begin{pmatrix} \cos \alpha_{ij} & \sin \alpha_{ij} \\ \sin \alpha_{ij} & -\cos \alpha_{ij} \end{pmatrix} \begin{pmatrix} \theta_x \\ \theta_y \end{pmatrix}. \quad (6.15)$$

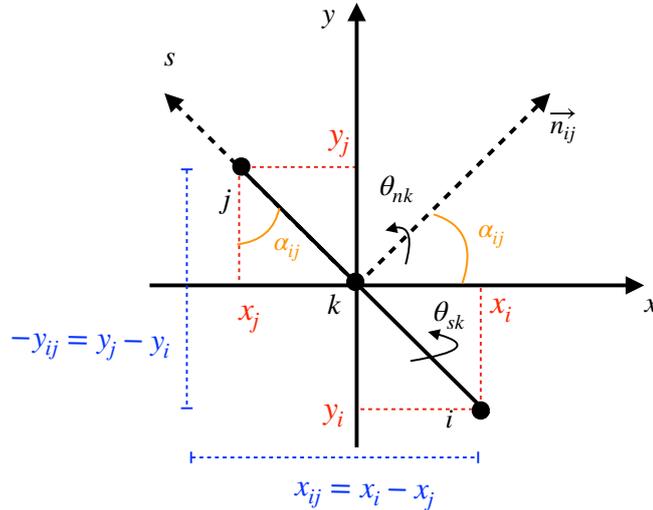


Figure 6.6 – Mise en évidence des relations autour des angles liés au repère curviligne : $\cos(\alpha_{ij}) = -y_{ij}/l_{ij}$ et $\sin(\alpha_{ij}) = x_{ij}/l_{ij}$ avec l_{ij} la longueur de l'arête ij .

Nous pouvons ainsi exprimer les rotations θ_{xk}, θ_{yk} pour $k \in \{4, 5, 6\}$ en fonction des variables nodales $w_i, \theta_{xi}, \theta_{yi}$ exprimées pour les nœuds i situés aux sommets de l'élément DKT (*i.e.* pour $i \in 1, 2, 3$) et utiliser l'interpolation de l'équation (6.8).

Interpolation des rotations θ_x, θ_y en fonction des valeurs nodales

En introduisant deux nouvelles fonctions de forme $\mathbf{H}_x(s, t), \mathbf{H}_y(s, t)$ issues des relations établies dans les équations (6.8) à (6.16), nous pouvons alors exprimer les rotations θ_x, θ_y en tout point $M = (s, t)$ de l'élément de référence en fonction des valeurs des nœuds qui sont situés au sommet du triangle. Soit \mathbf{u}_b le vecteur des variables nodales des nœuds d'interpolation de l'élément réel défini par

$$\mathbf{u}_b = (w_1 \theta_{x1} \theta_{y1} w_2 \theta_{x2} \theta_{y2} w_3 \theta_{x3} \theta_{y3})^T. \quad (6.16)$$

Nous avons alors :

$$\theta_y = \mathbf{H}_x(s, t) \mathbf{u}_b \text{ et } \theta_x = -\mathbf{H}_y(s, t) \mathbf{u}_b \text{ où} \quad (6.17)$$

$$\mathbf{H}_x(s, t) = \begin{bmatrix} H_{x1}(s, t) \\ H_{x2}(s, t) \\ H_{x3}(s, t) \\ H_{x4}(s, t) \\ H_{x5}(s, t) \\ H_{x6}(s, t) \\ H_{x7}(s, t) \\ H_{x8}(s, t) \\ H_{x9}(s, t) \end{bmatrix} = \begin{bmatrix} \frac{3}{2} (a_6 N_6 - a_5 N_5) \\ b_5 N_5 + b_6 N_6 \\ N_1 - c_5 N_5 - c_6 N_6 \\ \frac{3}{2} (a_4 N_4 - a_6 N_6) \\ b_6 N_6 + b_4 N_4 \\ N_2 - c_6 N_6 - c_4 N_4 \\ \frac{3}{2} (a_5 N_5 - a_4 N_4) \\ b_4 N_4 + b_5 N_5 \\ N_3 - c_4 N_4 - c_5 N_5 \end{bmatrix},$$

$$\mathbf{H}_y(s, t) = \begin{bmatrix} H_{y1}(s, t) \\ H_{y2}(s, t) \\ H_{y3}(s, t) \\ H_{y4}(s, t) \\ H_{y5}(s, t) \\ H_{y6}(s, t) \\ H_{y7}(s, t) \\ H_{y8}(s, t) \\ H_{y9}(s, t) \end{bmatrix} = \begin{bmatrix} \frac{3}{2} (d_6 N_6 - d_5 N_5) \\ -N_1 + e_5 N_5 + e_6 N_6 \\ -b_5 N_5 - b_6 N_6 \\ \frac{3}{2} (d_4 N_4 - d_6 N_6) \\ -N_2 + e_6 N_6 + e_4 N_4 \\ -b_6 N_6 - b_4 N_4 \\ \frac{3}{2} (d_5 N_5 - d_4 N_4) \\ -N_3 + e_4 N_4 + e_5 N_5 \\ -b_4 N_4 - b_5 N_5 \end{bmatrix} \text{ avec } \begin{cases} a_k = -x_{ij}/l_{ij}^2 \\ b_k = \frac{3}{4} x_{ij} y_{ij}/l_{ij}^2 \\ c_k = (\frac{1}{4} x_{ij}^2 - \frac{1}{2} y_{ij}^2)/l_{ij}^2 \\ d_k = -y_{ij}/l_{ij}^2 \\ e_k = (\frac{1}{4} y_{ij}^2 - \frac{1}{2} x_{ij}^2)/l_{ij}^2 \end{cases}$$

Approximation du champ de déplacement

L'approximation du champ de déplacement $\mathbf{u}^*(s, t) = (w_0(s, t), \theta_x(s, t), \theta_y(s, t))^T$ en tout point de l'élément de référence est ensuite obtenue en utilisant une interpolation des valeurs de ses nœuds. Notons $\mathbf{u}_i = (w_i, \theta_{xi}, \theta_{yi})$ la valeur calculée au nœud i (pour $i = 1, 2, 3$) de l'élément réel correspondant au mouvement dans la direction z de la plaque mince et à ses rotations autour des axes x et y . Ensuite, en faisant correspondre les valeurs des nœuds de l'élément de référence avec celles des nœuds de l'élément réel, nous obtenons pour tout point M défini dans le repère \mathbf{r} :

$$\begin{aligned} \mathbf{u}^*(s, t) &= \begin{pmatrix} w_0(s, t) \\ \theta_x(s, t) \\ \theta_y(s, t) \end{pmatrix} = \mathbf{u}^*(0, 0) H_1(s, t) + \mathbf{u}^*(1, 0) H_2(s, t) + \mathbf{u}^*(0, 1) H_3(s, t) \\ &= \mathbf{u}_1 H_1(s, t) + \mathbf{u}_2 H_2(s, t) + \mathbf{u}_3 H_3(s, t) \\ &= \underbrace{\begin{bmatrix} * & 0 & 0 & * & 0 & 0 & * & 0 & 0 \\ -H_{y1}(s, t) & -H_{y2}(s, t) & -H_{y3}(s, t) & -H_{y4}(s, t) & -H_{y5}(s, t) & -H_{y6}(s, t) & -H_{y7}(s, t) & -H_{y8}(s, t) & -H_{y9}(s, t) \\ H_{x1}(s, t) & H_{x2}(s, t) & H_{x3}(s, t) & H_{x4}(s, t) & H_{x5}(s, t) & H_{x6}(s, t) & H_{x7}(s, t) & H_{x8}(s, t) & H_{x9}(s, t) \end{bmatrix}}_{\mathbf{H}(s, t)} \underbrace{\begin{pmatrix} w_1 \\ \theta_{x1} \\ \theta_{y1} \\ w_2 \\ \theta_{x2} \\ \theta_{y2} \\ w_3 \\ \theta_{x3} \\ \theta_{y3} \end{pmatrix}}_{\mathbf{u}_b}. \end{aligned}$$

où $\mathbf{H}(s, t)$ ³ correspond à la matrice ligne des fonctions d'interpolation $H_i(s, t)$ attachées aux nœuds i de l'élément de référence.

Quantités élémentaires pour l'élément DKT. Comme pour les éléments finis CST, nous devons définir la matrice élémentaire de raideur relative aux éléments finis DKT pour permettre l'emploi de la MEF. Considérons à nouveau pour cela un élément réel D_e de surface A_e et d'épaisseur h . La matrice élémentaire de raideur relative au comportement de flexion modélisée par cet élément DKT est définie par :

$$\begin{aligned} \mathbf{K}_{be} &= \int_{D_e} \mathbf{B}_b(M)^T \mathbf{C}(M) \mathbf{B}_b(M) dx dy dz \\ &= \frac{h^3}{12} \int_{A_e} \mathbf{B}_b(M)^T \mathbf{C}(M) \mathbf{B}_b(M) dx dy \\ &= \frac{h^3}{12} \int_{t=0}^1 \int_{s=0}^{1-s} \mathbf{B}_b(M)^T \mathbf{C}(M) \mathbf{B}_b(M) \det(\mathbf{J}) ds dt \end{aligned}$$

où :

— pour la loi de Hooke, $\mathbf{C}(M) = \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{pmatrix}$.

— \mathbf{J} est la matrice jacobienne de la transformation géométrique permettant le passage de l'élément de référence à l'élément réel avec $\det(\mathbf{J}) = 2A = x_{21} y_{31} - x_{31} y_{21}$.

— \mathbf{B}_b est la matrice d'opérateurs différentiels appliquées aux fonctions d'interpolation de l'élément fini avec $\mathbf{B}_b = \mathbf{L} \mathbf{H}(s, t)$ où \mathbf{L} est la matrice d'opérateurs différentiels correspondant à l'expression du gradient du champ de déplacement.

— Rappelons que nous pouvons calculer les dérivées premières du champ de déplacement par rapport au repère \mathbf{x} en utilisant les règles de dérivation en chaînes. Nous avons ainsi :

$$\frac{\partial w(s, t)}{\partial \mathbf{x}} = \mathbf{J}^{-1} \frac{\partial w(s, t)}{\partial \mathbf{r}}, \quad \frac{\partial \theta_x(s, t)}{\partial \mathbf{x}} = \mathbf{J}^{-1} \frac{\partial \theta_x(s, t)}{\partial \mathbf{r}}, \quad \frac{\partial \theta_y(s, t)}{\partial \mathbf{x}} = \mathbf{J}^{-1} \frac{\partial \theta_y(s, t)}{\partial \mathbf{r}}.$$

— Par ailleurs, rappelons également que le tenseur des déformations de flexion est défini par $\epsilon_b = \mathbf{B}_b \mathbf{u}_b$ donnant comme relation :

$$\epsilon_b = \begin{pmatrix} \kappa_{xx} \\ \kappa_{yy} \\ 2 \kappa_{xy} \end{pmatrix} = \underbrace{\begin{pmatrix} \theta_{y,x} \\ -\theta_{x,y} \\ \theta_{y,y} - \theta_{x,x} \end{pmatrix}}_{\text{cf. équation (6.4)}} = \frac{1}{2A} \underbrace{\begin{bmatrix} y_{31} H_{x,s}^T + y_{12} H_{x,t}^T \\ -x_{31} H_{y,s}^T - x_{12} H_{y,t}^T \\ -x_{31} H_{x,s}^T - x_{12} H_{x,t}^T + y_{31} H_{y,s}^T + y_{12} H_{y,t}^T \end{bmatrix}}_{\mathbf{B}_b} \underbrace{\begin{pmatrix} w_1 \\ \theta_{x1} \\ \theta_{y1} \\ w_2 \\ \theta_{x2} \\ \theta_{y2} \\ w_3 \\ \theta_{x3} \\ \theta_{y3} \end{pmatrix}}_{\mathbf{u}_b}$$

3. Rappelons qu'il n'y a pas de fonction d'interpolation définie pour calculer w sur tout l'élément.

où les dérivées partielles de $H_x(s, t)$ et $H_y(s, t)$ par rapport à s et t sont définies par :

$$\mathbf{H}_{x,s}(s, t) = \begin{bmatrix} P_6(1-2s) + (P_5 - P_6)t \\ q_6(1-2s) + (q_5 + q_6)t \\ -4 + 6(s+t) + r_6(1-2s) - t(r_5 + r_6) \\ -P_6(1-2s) + t(P_4 + P_6)t \\ q_6(1-2s) + t(q_6 - q_4) \\ -2 + 6s + r_6(1-2s) + t(r_4 - r_6) \\ -t(P_5 + P_4) \\ t(q_4 - q_5) \\ -t(r_5 - r_4) \end{bmatrix}, \quad \text{où} \quad \begin{cases} P_k = -6 x_{ij}/l_{ij}^2 = 6 a_k \\ q_k = 3 x_{ij} y_{ij}/l_{ij}^2 = 4 b_k \\ t_k = -6 y_{ij}/l_{ij}^2 = 6 d_k \\ r_k = 3 y_{ij}^2/l_{ij}^2 \end{cases}$$

$$\mathbf{H}_{x,t}(s, t) = \begin{bmatrix} -P_5(1-2t) - s(P_6 - P_5) \\ q_5(1-2t) - s(q_5 + q_6) \\ -4 + 6(s+t) + r_5(1-2t) - s(r_5 + r_6) \\ s(P_4 + P_6) \\ s(q_4 - q_6) \\ -s(r_6 - r_4) \\ P_5(1-2t) - s(P_4 + P_5) \\ q_5(1-2t) + s(q_4 - q_5) \\ -2 + 6t + r_5(1-2t) + s(r_4 - r_5) \end{bmatrix},$$

$$\mathbf{H}_{y,s}(s, t) = \begin{bmatrix} t_6(1-2s) + t(t_5 - t_6) \\ 1 + r_6(1-2s) - t(r_5 + r_6) \\ -q_6(1-2s) + t(q_5 + q_6) \\ -t_6(1-2s) + t(t_4 + t_6) \\ -1 + r_6(1-2s) + t(r_4 - r_6) \\ -q_6(1-2s) - t(q_4 - q_6) \\ -t(t_4 + t_5) \\ t(r_4 - r_5) \\ -t(q_4 - q_5) \end{bmatrix}, \quad \mathbf{H}_{y,t}(s, t) = \begin{bmatrix} -t_5(1-2t) - s(t_6 - t_5) \\ 1 + r_5(1-2t) - s(r_5 + r_6) \\ -q_5(1-2t) + s(q_5 + q_6) \\ s(t_4 + t_6) \\ s(r_4 - r_6) \\ -s(q_4 - q_6) \\ t_5(1-2t) - s(t_4 + t_5) \\ -1 + r_5(1-2t) + s(r_4 - r_5) \\ -q_5(1-2t) - s(q_4 - q_5) \end{bmatrix}.$$

L'intégrale de \mathbf{K}_{be} peut être calculée en employant une quadrature de Gauss à 3 points. La Fig. 6.7 illustre la taille de cette matrice en montrant les multiplications de matrices qui composent son calcul. Cette matrice est de taille globale 9×9 . Elle est composée des éléments K_{be}^{ij} qui sont de taille 3×3 (pour $i, j = 1, 2, 3$).

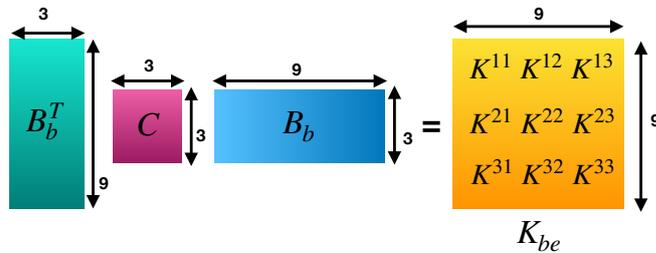


Figure 6.7 – Illustration de la matrice élémentaire K_{be} d'un élément fini DKT.

Définition de l'élément CST-DKT

Pour simuler à la fois le comportement membranaire et de flexion d'une plaque mince, nous combinons les deux types d'éléments finis décrits précédemment créant un élément fini appelé CST-DKT. Comme illustré sur la Fig. 6.8, cet élément est un triangle rectangle défini en 2D dans le repère de référence $\mathbf{r} = (s, t)$. Il est constitué de 6 nœuds : 3 nœuds de coordonnées $(0, 0)$, $(1, 0)$, $(0, 1)$ situés aux sommets du triangle et 3 nœuds de coordonnées $(1/2, 0)$, $(0, 1/2)$, $(1/2, 1/2)$ situés au milieu des arêtes du triangle. Une transformation géométrique $\mathbf{x}(\mathbf{r})$ est ensuite utilisée pour passer de cet élément de référence à un élément triangulaire réel. Ce dernier dispose de 6 nœuds i de coordonnées (x_i, y_i, z_i) définies dans le repère local $\mathbf{x} = (x, y, z)$ positionné en son barycentre G : 3 nœuds situés aux sommets de l'élément (pour $i = 1, 2, 3$) et 3 nœuds situés au milieu des arêtes (pour $i = 4, 5, 6$). Par ailleurs, les nœuds i situés aux sommets de l'élément disposent chacun de 6 degrés de liberté : 2 degrés de liberté (u_i, v_i) issus de l'élément fini CST, 3 degrés de liberté $(w_i, \theta_{xi}, \theta_{yi})$ issus de l'élément fini DKT, et 1 degré de liberté (θ_{zi}) permettant une rotation de la plaque autour de l'axe z . Nous avons ainsi 18 degrés de liberté pour cet élément. Nous faisons ensuite correspondre les valeurs des nœuds de l'élément de référence avec celles des nœuds de l'élément réel. Au final, la transformation géométrique $\mathbf{x}(\mathbf{r})$ permet de faire en sorte que chaque point (s, t) de l'élément fini de référence ait un correspondant (x, y, z) sur l'élément réel.

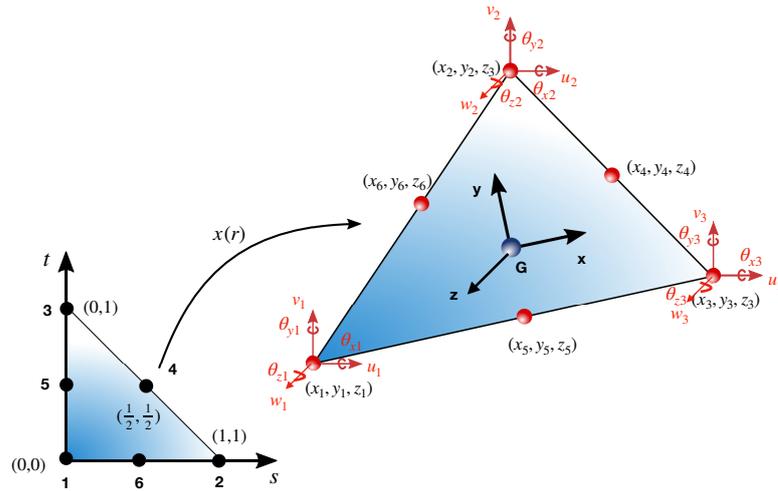


Figure 6.8 – (A gauche) L'élément fini de référence appelé CST-DKT dans le repère $\mathbf{r} = (s, t)$ en correspondance (à droite) avec un élément fini triangulaire défini dans le repère réel $\mathbf{x} = (x, y, z)$.

L'approximation du champ de déplacement $\mathbf{u}^*(s, t)$ en tout point de l'élément de référence est ensuite obtenue en utilisant une interpolation linéaire des valeurs de ses nœuds. Notons $\mathbf{u}_i = (u_i, v_i, w_i, \theta_{xi}, \theta_{yi}, \theta_{zi})$ la valeur calculée au nœud i (pour $i = 1, 2, 3$) de l'élément réel. En faisant correspondre les valeurs des nœuds de l'élément de référence avec celles des nœuds de l'élément réel, nous obtenons pour tout point M de l'élément de référence :

$$\mathbf{u}^*(s, t) = \begin{pmatrix} u_0(s, t) \\ v_0(s, t) \\ w_0(s, t) \\ \theta_{x0} \\ \theta_{y0} \\ \theta_{z0} \end{pmatrix} = \mathbf{u}^*(0, 0) NH_1(s, t) + \mathbf{u}^*(1, 0) NH_2(s, t) + \mathbf{u}^*(0, 1) NH_3(s, t)$$

$$\begin{aligned} \Rightarrow \mathbf{u}^*(s, t) &= \mathbf{u}_1 NH_1(s, t) + \mathbf{u}_2 NH_2(s, t) + \mathbf{u}_3 NH_3(s, t) \\ &= \mathbf{NH}(s, t) \mathbf{u}_m \end{aligned}$$

où \mathbf{u}_m correspond au vecteur des variables nodales des nœuds d'interpolation de l'élément réel défini par $\mathbf{u}_m = (u_1, v_1, w_1, \theta_{x1}, \theta_{y1}, \theta_{z1}, u_2, v_2, w_2, \theta_{x2}, \theta_{y2}, \theta_{z2}, u_3, v_3, w_3, \theta_{x3}, \theta_{y3}, \theta_{z3})^T$, et $\mathbf{NH}(s, t)$ correspond à la matrice ligne des fonctions d'interpolation $NH_i(s, t)$ attachées aux nœuds i (pour $i = 1, 2, 3$) de l'élément de référence. Cette matrice est définie par :

$$\mathbf{NH}(s, t) = \left[\begin{array}{ccc} NH_1(s, t) & NH_2(s, t) & NH_3(s, t) \end{array} \right] \text{ avec}$$

$$NH_1(s, t) = \begin{bmatrix} N_1(s, t) & 0 & 0 & 0 & 0 & 0 \\ 0 & N_1(s, t) & 0 & 0 & 0 & 0 \\ 0 & 0 & \star & 0 & 0 & 0 \\ 0 & 0 & -H_{y1}(s, t) & -H_{y2}(s, t) & -H_{y3}(s, t) & 0 \\ 0 & 0 & H_{x1}(s, t) & H_{x2}(s, t) & H_{x3}(s, t) & 0 \\ 0 & 0 & 0 & 0 & 0 & \diamond \end{bmatrix},$$

$$NH_2(s, t) = \begin{bmatrix} N_2(s, t) & 0 & 0 & 0 & 0 & 0 \\ 0 & N_2(s, t) & 0 & 0 & 0 & 0 \\ 0 & 0 & \star & 0 & 0 & 0 \\ 0 & 0 & -H_{y4}(s, t) & -H_{y5}(s, t) & -H_{y6}(s, t) & 0 \\ 0 & 0 & H_{x4}(s, t) & H_{x5}(s, t) & H_{x6}(s, t) & 0 \\ 0 & 0 & 0 & 0 & 0 & \diamond \end{bmatrix},$$

$$NH_3(s, t) = \begin{bmatrix} N_3(s, t) & 0 & 0 & 0 & 0 & 0 \\ 0 & N_3(s, t) & 0 & 0 & 0 & 0 \\ 0 & 0 & \star & 0 & 0 & 0 \\ 0 & 0 & -H_{y7}(s, t) & -H_{y8}(s, t) & -H_{y9}(s, t) & 0 \\ 0 & 0 & H_{x7}(s, t) & H_{x8}(s, t) & H_{x9}(s, t) & 0 \\ 0 & 0 & 0 & 0 & 0 & \diamond \end{bmatrix},$$

où $N_i(s, t)$ (pour $i = 1, 2, 3$) sont les fonctions d'interpolation définies pour l'élément CST et $H_{xj}(s, t), h_{yj}(s, t)$ (pour $j = 1, \dots, 9$) sont les fonctions d'interpolation définies pour l'élément DKT. Par contre, il n'y a pas de fonction d'interpolation définie pour calculer w et θ_z sur tout l'élément.

Quantités élémentaires pour l'élément CST-DKT. La matrice élémentaire de raideur relative à l'élément fini CST-DKT doit être définie pour employer la MEF. Considérons à nouveau pour cela un élément réel D_e de surface A_e et d'épaisseur h . La Fig. 6.9 illustre la composition de cette matrice élémentaire \mathbf{K}_{mbe} :

- Elle est de taille globale 18×18 .
- Elle est composée des éléments K_{mbe}^{ij} de taille 6×6 (pour $i, j = 1, 2, 3$).
- Chacun de ces éléments K_{mbe}^{ij} est construit en combinant :
 - l'élément K_{me}^{ij} (de taille 2×2) de la matrice élémentaire de raideur \mathbf{K}_{me} relatif au comportement membranaire de l'élément fini CST,
 - l'élément K_{be}^{ij} (de taille 3×3) de la matrice élémentaire de raideur \mathbf{K}_{be} relatif au comportement de flexion de l'élément fini DKT,
 - un terme de rigidité fictif K_e (scalaire) qui correspond à la rotation θ_z autour de l'axe perpendiculaire à l'élément. Notons que ce terme permet notamment d'éviter que la matrice K_{mbe}^{ij} soit singulière. En pratique, la valeur minimale de la diagonale de $K_e/1000$.

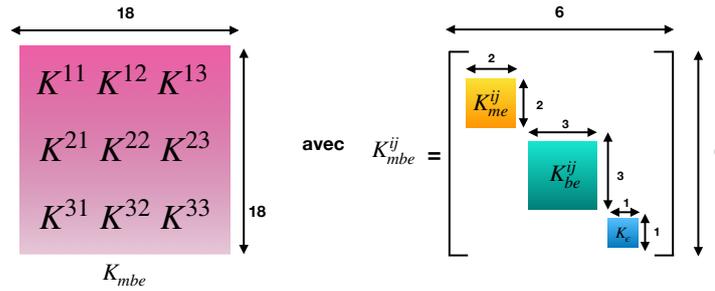


Figure 6.9 – Illustration de la matrice élémentaire K_{mbe} d'un élément fini CST-DKT.

Après avoir construit les quantités élémentaires dont nous avons besoin pour employer la MEF, regardons maintenant comment va se dérouler la simulation dynamique.

6.4 Formulation co-rotationnelle pour les éléments CST-DKT

Rappelons que le modèle que nous présentons ici est un modèle linéaire, puisqu'au niveau mécanique, nous avons employé la loi de Hooke (qui est valide dans le cadre de petites déformations) et qu'au niveau géométrique, nous avons employé le tenseur de déformations de Green-Lagrange linéarisé (qui est valide dans le cadre de petits déplacements). Pour pouvoir conserver l'emploi de ce tenseur de déformations dans le cadre de grands déplacements, il est ainsi nécessaire d'extraire la rotation dite rigide avant de calculer les déformations subies par l'objet, puisque ce tenseur de déformations n'est pas invariant par rotation. En effet, sans traitement particulier, des artefacts géométriques apparaîtraient notamment lors des rotations. Pour remédier à cette limitation, nous employons une formulation dite co-rotationnelle qui permet de séparer la partie déplacement de l'objet de la partie déformation. Pour cela, les déformations sont dans un premier temps exprimées dans un repère qui est local à l'élément considéré, avant dans un second temps de les exprimer dans le repère global (ou monde). Ainsi, les forces qui engendrent un déplacement modifient la position de l'objet de manière rigide, tandis que les déformations agissent sur l'objet par le modèle local. Nous pouvons ainsi effectuer de grands déplacements. Par contre, les déformations au niveau de l'objet doivent rester petites. Cette approche permet l'obtention d'un résultat similaire à celui de l'emploi d'un tenseur de déformation qui serait invariant par rotation, comme le tenseur de Green non-linéaire, mais avec un coût de calculs moindre.

Cette approche a été initialement proposée pour les objets 3D par Müller et al. (2002) pour un système masses-ressorts basé sur une modélisation en tétraèdres. Elle suppose que les déformations dans la configuration co-rotationnelle sont petites. Elle a ensuite été largement employée en Informatique Graphique (Allard et al. (2007, 2011b); Nesme et al. (2005b)). Par contre, dans le cadre de l'emploi de notre modèle physique basé sur les éléments CST-DKT, il est nécessaire de prendre en compte les degrés de liberté relatifs aux rotations qui ne se manipulent pas de la même façon que le déplacement. C'est pourquoi, nous nous sommes aussi basés sur l'approche proposée par Felippa and Haugen (2005).

Formulation employée. Rappelons que dans le cadre de la simulation d'un objet rigide, l'état de l'objet rigide est défini par une translation (vecteur 3D noté $\mathbf{x}(t)$) et une rotation (matrice $\mathbf{R}(t)$ de taille 3×3) qui est considérée par rapport au barycentre de l'objet rigide (voir Fig. 6.10). Ensuite, pour définir un repère local à l'objet rigide, nous prenons son barycentre comme origine du repère (qui devient le point $x(t)$ dans le repère du monde), et les axes du repère local à l'objet rigide sont obtenus dans le repère du monde en les multipliant par la matrice de rotation $\mathbf{R}(t)$. Ainsi, les vecteurs colonnes de la matrice de rotation $\mathbf{R}(t)$ donne la direction dans le repère monde des axes du repère local. Par ailleurs, rappelons que pour une matrice de rotation, nous avons comme propriété : $\mathbf{R}^{-1} = \mathbf{R}^T$.

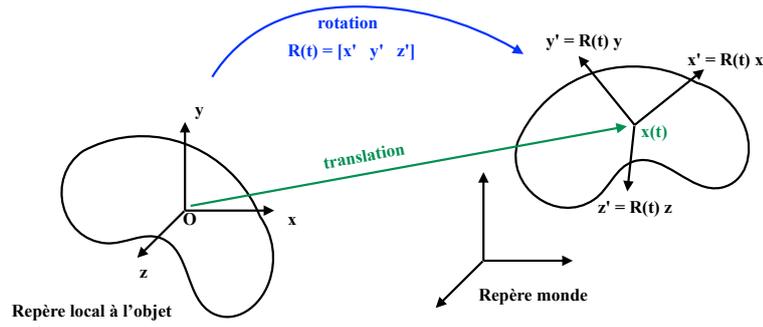


Figure 6.10 – Rappel sur la simulation d'un objet rigide dont le mouvement peut être décomposé en une translation et une rotation via la définition d'un repère local dont l'origine est le barycentre de l'objet.

Dans le cadre de l'emploi d'une formulation co-rotationnelle au sein de notre simulation basée sur des éléments finis CST-DKT, nous allons utiliser ce concept pour extraire le déplacement rigide des nœuds en définissant des repères locaux aux éléments triangulaires constituant l'objet simulé et en définissant la rotation subie par ces éléments. Pour chacun des éléments CST-DKT, nous considérerons ainsi deux configurations (illustrées par la Fig. 6.11) : la configuration non-déformée (état initial) et celle déformée (état courant). L'idée est ensuite d'aligner sur le même repère l'élément dans son état initial et l'élément dans son état courant afin d'extraire les rotations et les déformations pures.

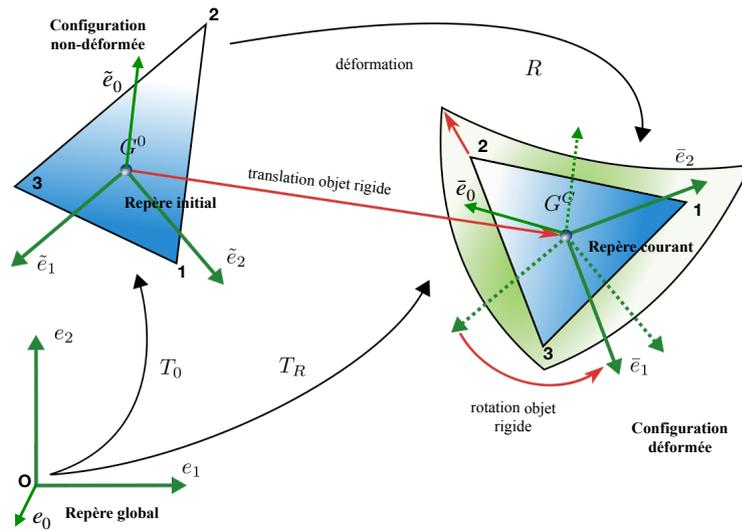


Figure 6.11 – Cinématique pour la formulation co-rotationnelle : la matrice de rotation T_0 (resp. T_R) permet le passage du repère global au repère local de la configuration non-déformée ou initiale (resp. déformée ou courante). Ce repère dit initial (resp. courant) est défini par les vecteurs unitaires \tilde{e}_i (resp. \tilde{e}_i) (pour $i = 0, 1, 2$) avec comme origine G^0 (resp. G^C) barycentre de l'élément dans sa configuration non-déformée (resp. déformée).

Nous pouvons observer sur la Fig. 6.11 que la matrice de rotation R (de taille 3×3) permet le passage de la configuration non-déformée à la configuration déformée. Cette matrice doit ainsi être déterminée pour extraire les rotations. Mais sa détermination n'est pas triviale, même si différentes méthodes ont été proposées telles que les décompositions QR ou polaire. Par contre l'emploi de ces méthodes, qui nécessitent un algorithme de minimisation, ne sont pas forcément assez rapides dans le cadre d'une simulation interactive. C'est pourquoi, nous avons opté pour une méthode empirique similaire à celle utilisée par Nesme et al. (2005a), mais appliquée à un élément triangulaire au lieu d'un tétraèdre.

Avant de présenter en détails notre formulation co-rotationnelle pour les éléments CST-DKT, la Fig.6.12 issue de Müller et al. (2008) illustre l'idée de cette formulation pour un tétraèdre en présentant le calcul de la force élastique $F = K_e \Delta x$ appliquée sur celui-ci. K_e est la matrice de raideur (de taille 12×12) de l'élément et Δx le déplacement (vecteur de taille 12) des 4 nœuds 3D de l'élément. L'approche suppose ensuite que nous connaissons la matrice globale de rotation R (de taille 3×3) de la transformation rigide de l'élément permettant la construction de la matrice R_e (de taille 12×12) contenant 4 copies de la matrice R sur sa diagonale et des 0 ailleurs. Cette matrice permet d'appliquer la même rotation R aux quantités présentes en chacun des 4 nœuds de l'élément.

Ensuite, le calcul de la force s'effectue de la façon suivante. La position déformée p (vecteur de taille 12) des nœuds est tout d'abord replacée dans le repère initial (qui ne présente pas la rotation) par $R_e^{-1} p$; puis le déplacement $R_e^{-1} p - x$ (avec x la position initiale) est multiplié par la matrice de raideur de l'élément considéré générant le calcul de la force avec $K_e (R_e^{-1} p - x)$ dans ce repère qui ne présente pas la rotation; avant d'être replacée dans le repère du tétraèdre déformé en multipliant le tout par R_e . La force est alors définie par : $F = R_e K_e (R_e^{-1} p - x)$.

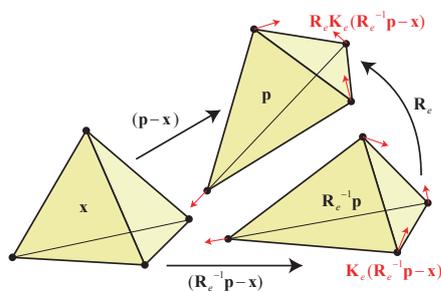


Figure 6.12 – Image issue de Müller et al. (2008) présentant le calcul des forces élastiques exercées sur les sommets d'un tétraèdre en utilisant une approche co-rotationnelle. Notation : x, p position des sommets dans le repère initial ou courant, R_e matrice de rotation de l'élément, K_e matrice de raideur de l'élément.

Nous allons maintenant voir comment cette approche se formule dans le cadre d'une modélisation en éléments finis CST-DKT pour laquelle nous devons gérer à la fois les degrés de libertés des nœuds en déplacement et ceux en rotation. Avant tout, nous devons définir un repère local pour chacun des éléments CST-DKT pour les deux configurations, initiale et courante, considérées. Comme illustré par la Fig. 6.11, nous avons ainsi :

- Un **repère global** (ou monde). Ce repère est défini par les vecteurs unitaires e_i (pour $i = 0, 1, 2$) et son origine est en $(0,0,0)$.
- Un **repère local à l'élément dans la configuration non-déformée/initiale**. Ce repère dit initial est défini par les vecteurs unitaires \tilde{e}_i (pour $i = 0, 1, 2$) et son origine est G^0 le barycentre de l'élément triangulaire dans cette configuration.
- Un **repère local à l'élément dans la configuration déformée/courante**. Ce repère dit courant est défini par les vecteurs unitaires \bar{e}_i (pour $i = 0, 1, 2$) et son origine est G^C le barycentre de l'élément triangulaire dans cette configuration.

Une fois ces repères définis, les vecteurs \tilde{e}_i (*resp.* \bar{e}_i) définissent les vecteurs colonnes de la matrice de rotation T_0 (*resp.* T_R) permettant le passage du repère global au repère initial (*resp.* *courant*). Puis, à partir de ces deux matrices de passage, nous pouvons définir la matrice de rotation permettant de passer de la configuration non-déformée à celle déformée.

Nous utilisons pour cela la formulation de Müller et al. (2002, 2008) avec :

$$R = T_R T_0^T.$$

Nous proposons ensuite la notation suivante pour les différents vecteurs :

- \mathbf{x} : les vecteurs exprimés dans le **repère global** avec \mathbf{x}_i la valeur du nœud i (pour $i = 1, 2, 3$) exprimée dans ce repère global ;
- \mathbf{x}^0 : les **valeurs initiales** des vecteurs exprimés dans le repère global avec \mathbf{x}_i^0 la valeur initiale du nœud i (pour $i = 1, 2, 3$) exprimée dans ce repère global ;
- $\tilde{\mathbf{x}}$: les vecteurs exprimés dans le **repère initial** (configuration non-déformée) avec $\tilde{\mathbf{x}}_i$ la valeur du nœud i (pour $i = 1, 2, 3$) exprimée dans ce repère initial ;
- $\bar{\mathbf{x}}$: les vecteurs exprimés dans le **repère courant** (configuration déformée) avec $\bar{\mathbf{x}}_i$ la valeur du nœud i (pour $i = 1, 2, 3$) exprimée dans ce repère courant.

Nous pouvons alors définir par rapport au repère global, les origines des deux repères locaux. Nous avons ainsi :

$$\mathbf{G}^0 = \frac{1}{3} \sum_{i=1}^3 \mathbf{x}_i^0 \quad \text{et} \quad \mathbf{G}^C = \frac{1}{3} \sum_{i=1}^3 \mathbf{x}_i.$$

Les vecteurs unitaires des repères locaux sont ensuite définis à partir des arêtes de l'élément triangulaire CST-DKT en le considérant dans sa configuration non-déformée ou dans sa configuration déformée :

- Soit \tilde{e}_0 (*resp.* \bar{e}_0) un vecteur colinéaire à l'arête définie par les nœuds 1 – 2.
- Soit \tilde{e}_2 (*resp.* \bar{e}_2) le vecteur obtenu en effectuant le produit vectoriel de \tilde{e}_0 (*resp.* \bar{e}_0) et du vecteur défini par les nœuds 1 – 3.
- Soit \tilde{e}_1 (*resp.* \bar{e}_1) le vecteur défini par le produit vectoriel de \tilde{e}_0 et \tilde{e}_2 (*resp.* \bar{e}_0 et \bar{e}_2).

Une fois les deux repères locaux créés pour la configuration initiale et déformée, nous obtenons les matrices de rotation T_0 et T_R (de taille 3×3) permettant d'exprimer dans ces deux repères, les coordonnées des nœuds i (pour $i = 1, 2, 3$) de l'élément avec :

$$\tilde{\mathbf{x}}_i^0 = T_0 \left(\mathbf{x}_i^0 - \mathbf{G}^0 \right), \quad \bar{\mathbf{x}}_i = T_R \left(\mathbf{x}_i - \mathbf{G}^C \right).$$

Nous pouvons alors exprimer le déplacement des degrés de liberté en translation par $\bar{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^0$ et le déplacement des degrés de liberté en rotation par $\bar{\theta}_i - \tilde{\theta}_i^0$ pour les nœuds i (pour $i = 1, 2, 3$) induisant le déplacement suivant pour l'élément CST-DKT :

$$\mathbf{u} = \left(\bar{\mathbf{x}}_1 - \tilde{\mathbf{x}}_1^0, \bar{\theta}_1 - \tilde{\theta}_1^0, \bar{\mathbf{x}}_2 - \tilde{\mathbf{x}}_2^0, \bar{\theta}_2 - \tilde{\theta}_2^0, \bar{\mathbf{x}}_3 - \tilde{\mathbf{x}}_3^0, \bar{\theta}_3 - \tilde{\theta}_3^0 \right)^T.$$

Puis, en utilisant la même approche que Müller et al. (2008), nous construisons la matrice T_{Re} (de taille 18×18) constituée de 6 copies de la matrice T_R sur sa diagonale. Cette matrice permettra la rotation, par la même matrice de rotation T_R , des quantités des 3 nœuds d'un élément CST-DKT (chacun étant défini par une composante 3D de translation et une composante 3D de rotation). Elle est ainsi définie par :

$$T_{Re} = \begin{pmatrix} T_R & & & & & 0 \\ & T_R & & & & \\ & & T_R & & & \\ & & & T_R & & \\ & & & & T_R & \\ 0 & & & & & T_R \end{pmatrix}.$$

Nous pouvons alors considérer le calcul de la force élastique $F_e = \bar{\mathbf{K}}_{mbe} \mathbf{u}$ avec $\bar{\mathbf{K}}_{mbe}$ la matrice de rigidité de l'élément CST-DKT calculée dans le repère courant. De la même manière que précédemment, nous remplaçons le déplacement des nœuds de l'élément dans le repère initial (qui ne présente pas la rotation) par $T_{Re}^{-1} \mathbf{u}$; puis nous le multiplions par la matrice de raideur de l'élément pour générer le calcul de la force dans le repère initial avec $\bar{\mathbf{K}}_{mbe} T_{Re}^{-1} \mathbf{u}$; avant de la replacer dans le repère de l'élément CST-DKT déformé en multipliant le tout par T_{Re} . La force est alors définie par :

$$F_e = T_{Re} \bar{\mathbf{K}}_{mbe} T_{Re}^{-1} \mathbf{u}.$$

6.5 Contrainte de préservation de volume

Ayant choisi de modéliser un objet 3D uniquement par sa surface en utilisant des éléments finis surfaciques, sa déformation induite par des forces extérieures risque naturellement de modifier son volume au cours de la simulation. Ainsi, si nous souhaitons préserver son volume malgré les déformations qu'il subit, il est nécessaire de rajouter une contrainte dans la simulation de son comportement. En ce qui concerne les travaux antérieurs sur **les méthodes de préservation du volume**, [Promayon et al. \(1996\)](#) et [Chang et al. \(2013\)](#) utilisent une solution basée sur une méthode de projection de gradient en utilisant le théorème de flux-divergence pour calculer et maintenir un volume constant. [Hong et al. \(2005, 2006\)](#) proposent une solution basée sur des contraintes afin de préserver le volume global d'un système masses-ressorts défini à partir d'une modélisation surfacique de l'objet simulé. Ils partent également du théorème de flux-divergence pour définir les contraintes implicites à appliquer au système masses-ressorts. Ces contraintes sont modélisées en utilisant les multiplicateurs de Lagrange engendrant la résolution d'un système non-linéaire. Cette approche augmente significativement le temps de calcul de la simulation. [Diziol et al. \(2011\)](#) proposent une méthode de préservation de volume qu'ils ont associée au modèle surfacique de [Müller et al. \(2005\)](#) (appelé *shape matching*) qui permet la simulation d'objets déformables à un coût moindre que lors de l'emploi d'un modèle volumique. Cette méthode est également basée sur une projection de gradient mais simplifiée qui ne garantit pas une préservation exacte du volume. [Irving et al. \(2007\)](#) proposent une méthode de simulation pour des solides déformables incompressibles et non-linéaires. Elle est basée sur l'emploi d'éléments finis volumiques. Le volume du maillage est conservé localement près de chaque nœud en corrigeant leur position et vitesse. Par contre, leur méthode nécessite l'introduction d'un terme de pression (similaire à celle d'un fluide).

Nous nous sommes inspirés de la méthode de [Promayon et al. \(1996\)](#) pour préserver le volume des objets modélisés avec les éléments finis CST-DKT. Notre souhait était d'obtenir une méthode qui soit utilisable lors de l'emploi d'autres modèles physiques, qui ne soit pas coûteuse en temps de calcul, et garantisse une préservation complète du volume dans le cas de simulations d'objets incompressibles tels que des tissus mous.

Approche générale. L'idée de notre approche consiste à déplacer durant la simulation les sommets de l'objet modélisant sa surface afin de préserver son volume initial. L'ajout de cette contrainte permet notamment de pouvoir simuler des objets dits incompressibles, c'est-à-dire des objets dont le volume reste constant quelque soit les contraintes qui lui sont appliquées. Dans cette section, nous notons $\mathbf{x}_i = (x_i, y_i, z_i)$ pour $i \in \{0, \dots, n-1\}$ la position respective des n nœuds du maillage composant la surface de l'objet, $\mathbf{X}^t = (\mathbf{x}_0, \dots, \mathbf{x}_i, \dots, \mathbf{x}_{n-1})^T$ le vecteur global contenant l'ensemble des positions des n nœuds au temps t , \mathcal{V} la fonction de volume, et dt le pas de temps de la simulation.

Calcul du volume de l'objet. Le théorème de flux-divergence (ou théorème de Green-Ostrogradski) permet de transformer l'intégrale sur un volume d'un champ vectoriel noté $f(\mathbf{x})$ en l'intégrale sur une surface avec

$$\int_V \operatorname{div} \mathbf{f}(x) \, d\mathbf{x} = \int_{\partial V} f(\mathbf{x})^T \mathbf{n}(\mathbf{x}) \, d\mathbf{x} \quad (6.18)$$

où ∂V représente la surface extérieure du volume V et \mathbf{n} la normale extérieure à la surface. Comme pour $f(\mathbf{x}) = \mathbf{x}$, $\operatorname{div} \mathbf{f}(x) = 3$, nous avons :

$$3 \mathcal{V} = \int_{\partial V} \mathbf{x}^T \mathbf{n}(\mathbf{x}) \, d\mathbf{x}. \quad (6.19)$$

Si nous considérons un maillage composé de m faces triangulaires (dénombrées par j), le volume de l'objet est alors défini par

$$\mathcal{V} = \frac{1}{3} \sum_{j=0}^{m-1} \mathbf{G}_j \cdot A_j \quad (6.20)$$

où \mathbf{G}_j est l'isobarycentre et A_j représente l'aire du triangle j calculée par

$$A_j = \frac{1}{2} \sum_{k=1}^3 \mathbf{x}_k^j \wedge \mathbf{x}_{k \oplus 1}^j \quad (6.21)$$

où \mathbf{x}_k^j représente la position du sommet k du triangle j , $\oplus 1$ est un opérateur circulaire retournant l'indice du nœud suivant dans le triangle.

Soit s_i la liste des triangles partageant le sommet i . La fonction de volume peut être directement exprimée en fonction de la position des nœuds du maillage avec

$$\mathcal{V} = \frac{1}{3} \sum_{i=0}^{n-1} \mathbf{x}_i \cdot \left(\sum_{j \in s_i} \frac{1}{3} A_j \right) = \frac{1}{3} \sum_{i=0}^{n-1} \mathbf{x}_i \cdot N_i \quad (6.22)$$

où n est le nombre de nœuds du maillage et N_i représente ici la normale discrète au point i , pouvant être assimilée à la discrétisation du gradient du volume.

Résolution de la contrainte de volume. Notre méthode de préservation de volume s'effectue en plusieurs étapes. La première consiste à calculer le volume initial \mathcal{V}_0 de l'objet au début de la simulation. Puis à chaque pas de temps de la simulation, trois étapes (illustrées par la Fig. 6.13) sont nécessaires :

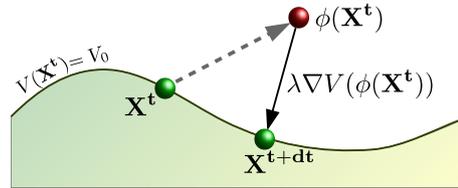


Figure 6.13 – Illustration de la projection liée à la contrainte de volume.

1. Calcul de la déformation $\phi(X^t)$ subie par notre modèle.
2. Mise à jour du vecteur \mathbf{X}^t afin de déplacer les nœuds vers la position $\phi(X^t)$.

3. Application de la contrainte de volume en projetant le vecteur $\phi(\mathbf{X}^t)$ dans l'espace de la contrainte de volume. Nous obtenons alors le nouveau vecteur de positions \mathbf{X}^{t+dt} satisfaisant cette contrainte. Ce processus d'effectue en résolvant le système suivant :

$$\begin{cases} \mathcal{V}(\mathbf{X}^{t+dt}) = \mathcal{V}(\mathbf{X}^0) = \mathcal{V}_0 \\ \mathbf{X}^{t+dt} = \phi(\mathbf{X}^t) + \lambda \nabla \end{cases} \Rightarrow \mathcal{V}(\phi(\mathbf{X}^t) + \lambda \nabla) = \mathcal{V}_0. \quad (6.23)$$

La solution du système donne ainsi le vecteur des déplacements globaux $\lambda \nabla$ (où ∇ représente la direction de projection) induit sur la position des nœuds pour respecter la contrainte, c'est-à-dire pour conserver le volume \mathcal{V}_0 .

En utilisant l'équation (6.22) et en considérant la position $\phi(\mathbf{x}_i^t) + \lambda \nabla_i$ pour chaque nœud i du maillage, le système (6.23) devient :

$$\frac{1}{3} \left[\sum_{i=0}^{n-1} (\phi(\mathbf{x}_i^t) + \lambda \nabla_i) \cdot \left(\sum_{j \in s_i} \frac{1}{3} A_j^{t+dt} \right) \right] = \mathcal{V}_0. \quad (6.24)$$

Notons ∇_k^j la projection dans la direction du k^e nœud de la j^e face triangulaire du maillage déformé. En considérant l'équation (6.21), nous obtenons :

$$\begin{aligned} A_j^{t+dt} &= \frac{1}{2} \sum_{k \in f_j} (\mathbf{x}_k^j + \lambda \nabla_k^j) \wedge (\mathbf{x}_{k \oplus 1}^j + \lambda \nabla_{k \oplus 1}^j) \\ &= \lambda^2 \frac{1}{2} \sum_{k \in f_j} \nabla_k^j \wedge \nabla_{k \oplus 1}^j \\ &\quad + \lambda \frac{1}{2} \sum_{k \in f_j} (\mathbf{x}_k^j \wedge \nabla_{k \oplus 1}^j - \mathbf{x}_{k \oplus 1}^j \wedge \nabla_k^j) + \frac{1}{2} \sum_{k \in f_j} \mathbf{x}_k^j \wedge \mathbf{x}_{k \oplus 1}^j. \end{aligned} \quad (6.25)$$

Soit N_i défini pour chacun des nœuds i et α_j, β_j définis pour chaque face triangulaire i comme présentés dans l'algorithme 9 (lignes 8, 12, 13). Nous obtenons :

$$N_i = \sum_{j \in s_i} \frac{1}{3} A_j^{t+dt} = \sum_{j \in s_i} \frac{1}{3} (\lambda^2 \alpha_j + \lambda \beta_j + A_j^t). \quad (6.26)$$

En utilisant cette équation (6.26), l'équation (6.24) se transforme en l'équation du troisième degré suivante :

$$\lambda^3 \sum_{i=0}^{n-1} (\nabla_i \cdot \alpha_i) + \lambda^2 \sum_{i=0}^{n-1} (\phi(\mathbf{x}_i^t) \cdot \alpha_i + \nabla_i \cdot \beta_i) + \lambda \sum_{i=0}^{n-1} (\phi(\mathbf{x}_i^t) \cdot \beta_i + \nabla_i \cdot N_i) + \sum_{i=0}^{n-1} (\phi(\mathbf{x}_i^t) \cdot N_i) = 3 \mathcal{V}_0. \quad (6.27)$$

où $\alpha_i = \frac{1}{3} \sum_{j \in s_i} \alpha_j$ et $\beta_i = \frac{1}{3} \sum_{j \in s_i} \beta_j$. L'algorithme 9 résume les différentes étapes de ce calcul pour un pas de temps de la simulation. Parmi les différentes solutions possibles pour λ , la valeur minimale absolue est choisie afin de minimiser la correction à effectuer.

Algorithme 9 : Calcul de la contrainte de volume et mise à jour des positions.

```

1:  $\mathbf{x}'_i = \phi(\mathbf{x}_i^t)$ 
2: // Calcul de l'aire de chaque face triangulaire  $j$ 
3: for  $j = 0 \rightarrow m - 1$  do
4:    $A_j = \frac{1}{2} \sum_{k=1}^3 \mathbf{x}_k^j \wedge \mathbf{x}_{k\oplus 1}^j$ 
5: end for
6: // Calcul de la normale discrète pour chaque sommet  $i$ 
7: for  $i = 0 \rightarrow n - 1$  do
8:    $N_i = \sum_{j \in s_i} \frac{1}{3} A_j$ 
9: end for
10: // Pour chaque face triangulaire  $j$ 
11: for  $j = 0 \rightarrow m - 1$  do
12:    $\alpha_j = \frac{1}{2} \sum_{k=1}^3 \nabla_k^j \wedge \nabla_{k\oplus 1}^j$ 
13:    $\beta_j = \frac{1}{2} \sum_{k=1}^3 (\mathbf{x}_k^j \wedge \nabla_{k\oplus 1}^j - \mathbf{x}_{k\oplus 1}^j \wedge \nabla_k^j)$ 
14: end for
15:  $a, b, c, d \leftarrow 0$ 
16: for  $i = 0 \rightarrow n - 1$  do
17:    $\alpha_i = \frac{1}{3} \sum_{j \in s_i} \alpha_j$ 
18:    $\beta_i = \frac{1}{3} \sum_{j \in s_i} \beta_j$ 
19:    $a \leftarrow a + \nabla_i \cdot \alpha_i$ 
20:    $b \leftarrow b + \mathbf{x}'_i \cdot \alpha_i + \nabla_i \cdot \beta_i$ 
21:    $c \leftarrow c + \mathbf{x}'_i \cdot \beta_i + \nabla_i \cdot N_i$ 
22:    $d \leftarrow d + \mathbf{x}'_i \cdot N_i$ 
23: end for
24:  $d \leftarrow d - 3\mathcal{V}_0$ 
25:  $\lambda = \min(\text{roots}(a,b,c,d))$ 
26: // Correction de la position de chacun des noeuds  $i$ 
27: for  $i = 0 \rightarrow n - 1$  do
28:    $\mathbf{x}_i^{t+dt} \leftarrow \phi(\mathbf{x}_i^t) + \lambda \nabla_i$ 
29: end for

```

Contrôle local de la contrainte de volume. Notre méthode permet un contrôle local de la contrainte de volume au niveau des nœuds du maillage en permettant la définition du vecteur ∇_i pour chacun des nœuds i du maillage :

- Dans le cas non-contraint, ce vecteur correspond au gradient du volume du maillage avec $\nabla_i = N_i$ pour chaque nœud i , c'est-à-dire que les positions des nœuds sont directement projetées dans cette direction (*cf.* Fig. 6.14(a)).
- Si un déplacement constant ou nul est imposé pour un nœud durant la simulation, le vecteur ∇_i peut être nul (*cf.* Fig. 6.14(b)), c'est-à-dire que le nœud i n'est ainsi pas affecté par la contrainte de volume. Dans ce cas, la contribution à la préservation du volume de ce nœud est répartie sur les autres nœuds.
- Il est également possible d'utiliser tout autre type de vecteur comme illustré par la Fig. 6.14(c) permettant de combiner facilement notre méthode de préservation de volume à tout autre contrainte projective comme des contraintes de collision.

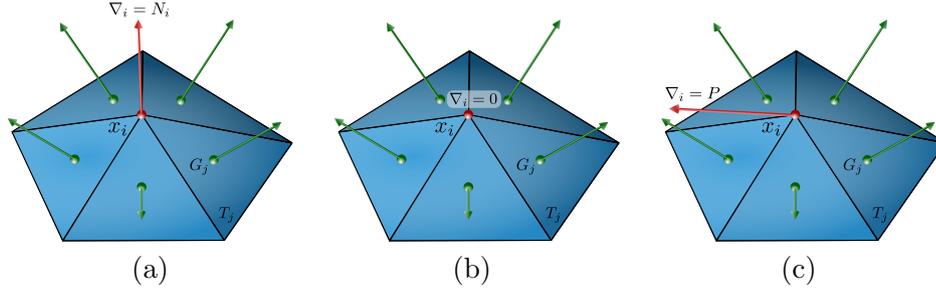


Figure 6.14 – Illustration du contrôle local de la contrainte de volume par la définition du vecteur ∇_i au nœud i : (a) projection sur le gradient du volume; (b) projection nulle; (c) projection arbitraire.

6.6 Simulation parallèle par coloriage de graphes

Si nous récapitulons le début de ce chapitre, nous avons tout d'abord présenté les éléments finis surfaciques CST-DKT qui ont été utilisés pour modéliser un objet volumique. Nous avons notamment défini leurs matrices élémentaires des raideurs. Puis, nous avons présenté la formulation co-rotationnelle employée pour garantir un comportement adéquate en grands déplacements malgré l'utilisation de la loi de Hooke et du tenseur de déformation de Green-Lagrange linéarisé. Cette formulation induit une nouvelle définition de la matrice de raideur des éléments. Ensuite, nous avons présenté la contrainte employée pour préserver le volume de l'objet simulé afin d'obtenir un comportement similaire à celui qu'on aurait obtenu en utilisant des éléments finis volumiques. Cette contrainte modifie directement la position des sommets des éléments du maillage surfacique. Elle est appliquée après avoir calculé la nouvelle position des nœuds issue de la déformation de l'objet simulé. Il nous reste ainsi à présenter comment cette simulation est effectuée à chaque pas de temps.

Simulation. Pour simuler le comportement d'un objet modélisé par les éléments CST-DKT, nous devons résoudre à chaque pas de temps l'équation matricielle globale de la dynamique newtonienne qui s'exprime sous la forme :

$$\mathbf{M} \ddot{\mathbf{U}} + \mathbf{K} \mathbf{U} = \mathbf{F}_d,$$

avec $\mathbf{U} = (u_0, v_0, w_0, \theta_{x0}, \theta_{y0}, \theta_{z0}, \dots, u_{n-1}, v_{n-1}, w_{n-1}, \theta_{xn-1}, \theta_{yn-1}, \theta_{zn-1})^T$ le vecteur généralisé des inconnus de notre modèle où n est le nombre de nœuds du maillage, \mathbf{F} les forces appliquées sur l'objet (somme des forces internes au système produites par le modèle CST-DKT et des forces extérieures telles que la gravité ou des forces d'interaction), \mathbf{M} la matrice diagonale des masses, et \mathbf{K} la matrice de rigidité du système. Par ailleurs, en considérant une loi de comportement linéaire, les forces élastiques sont de la forme $\mathbf{F}_{\text{elast}} = \mathbf{K} \mathbf{U}$.

Une méthode d'intégration numérique est ensuite employée pour calculer les nouvelles vitesses $\mathbf{V} = \dot{\mathbf{U}}$ et déplacement \mathbf{U} en fonction des accélérations $\ddot{\mathbf{U}}$. Dans le chapitre 3, nous avons vu que l'emploi du schéma d'Euler implicite, qui a l'avantage d'être inconditionnellement stable, implique la résolution du système linéaire suivant de type $Ax = B$ (Baraff and Witkin (1998)) dans lequel h représente le pas de temps de la simulation :

$$\underbrace{\left(M - h \frac{\partial F(t)}{\partial V} - h^2 \frac{\partial F(t)}{\partial U} \right)}_A \underbrace{\Delta V}_x = h \underbrace{\left(F(t) + h \frac{\partial F(t)}{\partial U} V(t) \right)}_B.$$

Sa résolution par la méthode du Gradient Conjugué permet l'obtention de ΔV rendant possible le calcul des nouvelles valeurs de \mathbf{V} et \mathbf{U} au pas de temps $t + h$ avec :

$$\begin{cases} V(t+h) &= V(t) + \Delta V \\ U(t+h) &= U(t) + h V(t+h) \end{cases}$$

En prenant l'hypothèse que les forces extérieures ne sont pas dépendantes de \mathbf{U} et \mathbf{V} , la matrice $\partial F(t)/\partial U$ est égale à la matrice \mathbf{K} qui est semi-définie positive⁴ et la matrice $\partial F(t)/\partial V$ est la matrice d'amortissement \mathbf{D} définie comme une combinaison linéaire des matrices \mathbf{M} et \mathbf{K} avec $\mathbf{D} = (\alpha \mathbf{M} + \beta \mathbf{K})$ où α et β sont les coefficients de Rayleigh (Sifakis and Barbic (2012)) avec $\alpha, \beta > 0$. La matrice \mathbf{D} est donc également semi-définie positive. Au final, le système précédent s'écrit alors sous la forme suivante, où A est une matrice de taille $6n \times 6n$ qui est semi-définie positive :

$$\underbrace{(\mathbf{M} - h(\alpha \mathbf{M} + \beta \mathbf{K}) - h^2 \mathbf{K})}_A \underbrace{\Delta V}_x = \underbrace{h(F(t) + h \mathbf{K} V(t))}_B. \quad (6.28)$$

Travaux existants en simulation parallèle sur GPU. Pour réduire le temps de calcul de la simulation, une implémentation sur le GPU a été réalisée. Notons qu'il existe un certain nombre de travaux concernant ce type de parallélisation, puisque depuis plusieurs années il est aisé d'employer les cartes graphiques pour des calculs complexes. Bartezzaghi et al. (2015) proposent ainsi une implémentation sur GPU des coques minces en employant un schéma d'intégration explicite (qui ne convient pas pour la simulation de matériaux ayant une certaine raideur). Allard et al. (2011a) présentent une parallélisation sur GPU d'une simulation basée sur des EF volumiques en employant le schéma d'intégration d'Euler implicite. La méthode du Gradient Conjugué est employée pour résoudre le système linéaire. Ils montrent explicitement le graphe de tâches associé et expliquent différentes méthodes pour sa parallélisation, avant de présenter plus en détails leur approche basée sur le traitement parallèle des nœuds avec accumulation des résultats en considérant les tétraèdres connectés aux nœuds. Ils proposent également une stratégie permettant d'optimiser les accès mémoire. Bosman et al. (2013) présentent une implémentation GPU similaire en utilisant CUDA. Pour notre part, nous avons choisi une stratégie basée sur le traitement en parallèle des éléments avec un partitionnement par coloriage de graphes. Nous avons également cherché à optimiser les accès mémoire.

Choix d'une approche parallèle par coloriage de graphes. Dans le Chap. 3, nous avons vu que la méthode de résolution du Gradient Conjugué induit plusieurs itérations pour calculer le produit matrice-vecteur $A \Delta V$. Ainsi, à chaque itération l , le produit local $A_{e_i} \Delta V_{e_i}^l$ est effectué pour chaque élément e_i du maillage et mis dans le vecteur résultat ΔV . Une stratégie de parallélisation sur GPU consiste à effectuer en parallèle les calculs sur les différents éléments. Par contre, pour éviter les accès concurrents aux données, il est nécessaire de partitionner les éléments afin qu'ils n'accèdent pas aux mêmes données au même moment, puisque des nœuds sont partagés par plusieurs éléments. Pour cela, le partitionnement des éléments est calculé de façon à ce que les éléments d'une même partition n'ait pas de nœuds en commun. Cette méthode nécessite le calcul du graphe d'adjacence $G = (S, E)$ où S est l'ensemble des éléments et $e = (e_1, e_2) \in E$ si et seulement si les éléments e_1 et e_2 partagent au moins un nœud. Un algorithme de coloriage de graphes (Kubale (2004)) est ensuite utilisé pour effectuer ce partitionnement.

4. Nous rappelons qu'une matrice A est *définie-positif* si, pour tout vecteur non nul x , $x^T A x > 0$.

La Fig. 6.15 présente un exemple obtenu en utilisant l'heuristique DSATUR (Brélaz (1979)) dont la complexité est linéaire en fonction du nombre de sommets.

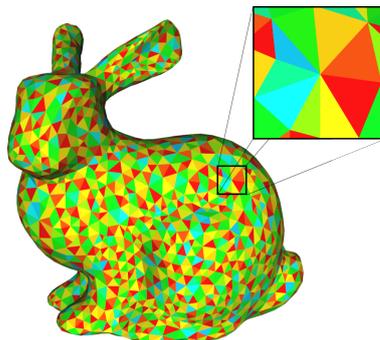


Figure 6.15 – Partitionnement d'un maillage représentant un lapin composé de 5000 éléments : 10 partitions ont été obtenues, chacune étant représentée par une couleur.

Au final, comme les éléments d'une partition ne partagent aucun nœud, le calcul du produit $A_{e_i} \Delta V_{e_i}^l$ peut être effectué en parallèle pour les éléments e_i (pour $i = 0, \dots, n - 1$) d'une même partition p . Par ailleurs, rappelons que dans le cas d'un modèle linéaire, la matrice de rigidité \bar{K}_e relative à chacun des éléments e est constante dans le repère local de l'élément et peut donc être pré-calculée et stockée. Au final, la Fig. 6.17 illustre la boucle de simulation faite sur le GPU. L'efficacité de cette parallélisation dépend du nombre de partitions dépendant de la qualité du maillage. Notons qu'un maillage de bonne qualité a généralement une valence égale à 5 (la valence d'un sommet correspondant au nombre d'arêtes incidentes) engendrant un nombre de partitions d'environ 10 pour un maillage triangulaire. Par ailleurs, dans le cas d'une simulation n'engendrant pas de modifications topologiques des objets simulés, ce partitionnement est établi en pré-calcul de la simulation, puisqu'il ne sera pas modifié.

Optimisation des accès mémoire. Une autre optimisation concerne celle relative aux accès mémoire afin que ces derniers se fassent de manière contiguë sur le GPU par les processus légers. Pour cela, il est préférable de décomposer les structures en plusieurs tableaux de données. Par exemple, il est plus judicieux d'utiliser un tableau de scalaires pour chacune des directions x, y, z plutôt que d'employer un seul tableau dont chacun des éléments correspond à un élément défini dans l'espace 3D. De plus, comme chacun des nœuds d'une partition n'appartient qu'à un seul élément de la partition (par définition des partitions), les données relatives à un nœud peuvent être restructurées pour assurer un accès contiguë à la mémoire par les différents threads. En effet, l'efficacité est maximale si un groupe de processus légers consécutifs accèdent à des données qui sont stockées de manière contiguë dans la mémoire globale et alignées sur 4, 8 ou 16 octets. Ainsi dans cette configuration, un ensemble de 32 threads CUDA (appelé *warp*) n'effectuera qu'une seule transaction mémoire pour l'accès aux données. Cette stratégie est illustrée par la Fig. 6.16 qui présente à gauche l'implémentation naïve sans cette optimisation mémoire, et à droite celle où les données sont contiguës par nœud dans le cas de notre simulation.

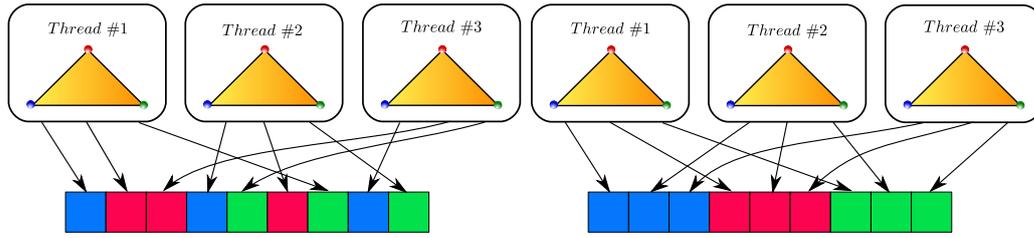


Figure 6.16 – Illustration des accès mémoire non fusionnés (gauche) ou fusionnés (droite).

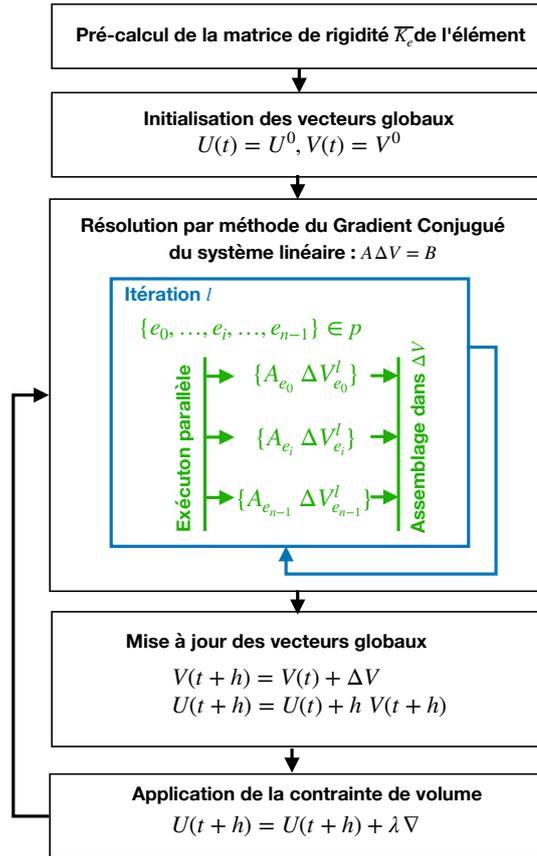


Figure 6.17 – Illustration de la boucle de simulation sur GPU : calcul effectué en parallèle pour les éléments d'une même partition p avec $|p| = n$.

6.7 Évaluation du modèle CST-DKT et de sa parallélisation

Évaluation en précision. Pour évaluer la précision du modèle physique CST-DKT, nous avons comparé ses résultats avec ceux que nous obtenons en utilisant un modèle éléments finis solides et un modèle masses-ressorts. La Fig. 6.18 donne un comparatif visuel entre les résultats obtenus par ces 3 modèles en considérant une sphère compressée par deux plans (le plan du dessus n'est pas affiché pour plus de visibilité). Les simulations ont été effectuées en imposant le déplacement des nœuds en contact avec les deux plans. Les caractéristiques de la sphère sont les suivantes : module de Young de 3 kPa, coefficient de Poisson de 0.49, masse volumique de 100 kg/m^3 . Puis, concernant les différents modèles :

- pour le modèle EF volumiques : le volume de la sphère a été discrétisé en utilisant 6665 tétraèdres avec une loi de comportement linéaire (Hooke) ;
- pour le modèle masses-ressorts : le volume de la sphère a été discrétisé en utilisant des tétraèdres engendrant 5000 ressorts de raideur ;
- pour le modèle CST-DKT : la surface de la sphère a été discrétisée en utilisant 1280 triangles avec une épaisseur de 10 cm.

Nous avons effectué deux comparatifs :

- La Fig. 6.18(a) présente le déplacement des nœuds à la surface en comparaison avec les résultats obtenus en utilisant le modèle EF volumiques. Nous observons que les résultats obtenus par le modèle CST-DKT sont plus proches du modèle EF volumique, que ceux obtenus avec le modèle masses-ressorts.
- La Fig. 6.18(b) présente la courbure de Gauss en comparaison avec les résultats obtenus en utilisant le modèle EF volumiques. Nous observons que la courbure résultante de l'utilisation du modèle CST-DKT est plus proche de celle obtenue par le modèle EF volumique, que celle obtenue avec le modèle masses-ressorts (pour lequel des arêtes vives apparaissent aux endroits de contact avec le plan supérieur).

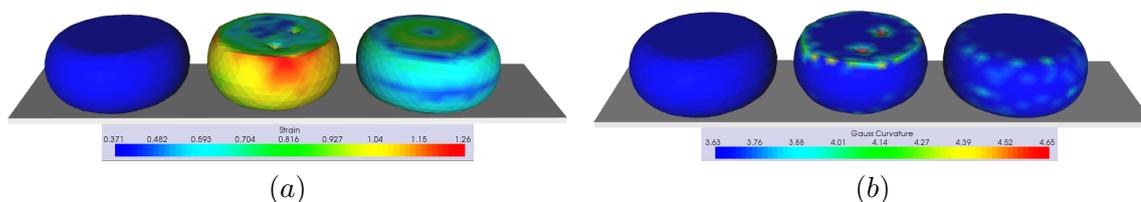


Figure 6.18 – Comparaisons par rapport au modèle EF volumique en élasticité linéaire (à gauche) avec un modèle masses-ressorts (au centre) et notre modèle CST-DKT incluant la contrainte de volume (à droite). (a) déplacements relatifs des noeuds à la surface. (b) courbure de Gauss.

Par ailleurs, durant ces simulations, le volume perdu par le modèle EF volumiques est de 1,9 % ; celui pour le modèle masses-ressorts est de 12,5 % ; et celui pour le modèle CST-DKT est de 0 % permettant ainsi de simuler le comportement d'un objet incompressible (coefficient de Poisson à 0,49) grâce à la contrainte de conservation de volume.

La Fig. 6.19 présente ensuite une comparaison entre le modèle EF volumique basé sur une discrétisation en tétraèdres du matériau quasi-incompressible (avec une loi de comportement linéaire de Hooke), et le modèle CST-DKT avec ou sans l'utilisation de la contrainte de volume. Les résultats concernent la simulation du comportement d'un cylindre. Celui-ci est placé entre deux plaques et subit un étirement vers le haut à 10 % (gauche), 20 % (milieu) ou 30 % (droite) par rapport à sa hauteur initiale. Les caractéristiques du cylindre sont les suivantes : module de Young de 3 kPa, coefficient de Poisson de 0.45, masse volumique de $100 \text{ kg}/m^3$. Le modèle EF volumique est basé sur des tétraèdres et le modèle CST-DKT a une épaisseur de 10 cm et est discrétisé en triangles. Les courbes montrent la forme du cylindre en donnant les valeurs du rayon du cylindre le long de l'axe longitudinal. L'emploi de la contrainte de volume additionnée à l'utilisation du modèle CST-DKT permet de garantir des résultats similaires aux modèles éléments finis volumiques.

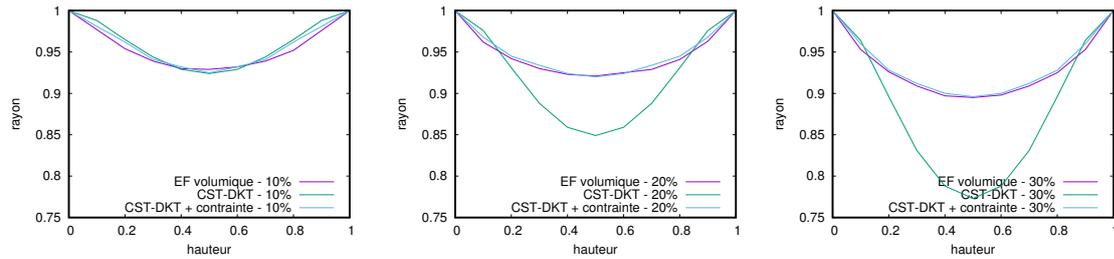


Figure 6.19 – Comparaison entre le modèle CST-DKT (avec ou sans contrainte de volume) et le modèle EF volumiques dans le cadre de la simulation d'un cylindre étiré vers le haut à 10 % (gauche), 20 % (milieu) ou 30 % (droite) par rapport à sa hauteur initiale.

Ensuite, la Fig. 6.20 présente le résultat visuel de différentes simulations induisant la déformation de quatre objets différents. Pour ces objets, le module de Young est de 3 kPa, le coefficient de Poisson est de 0.45, l'épaisseur est de 10 cm et la masse volumique est de 100 kg/m^3 . A noter que nous obtenons systématiquement une conservation complète du volume (perte de 0 % du volume) lors de l'emploi de la contrainte de volume dans le cadre du modèle surfacique CST-DKT.

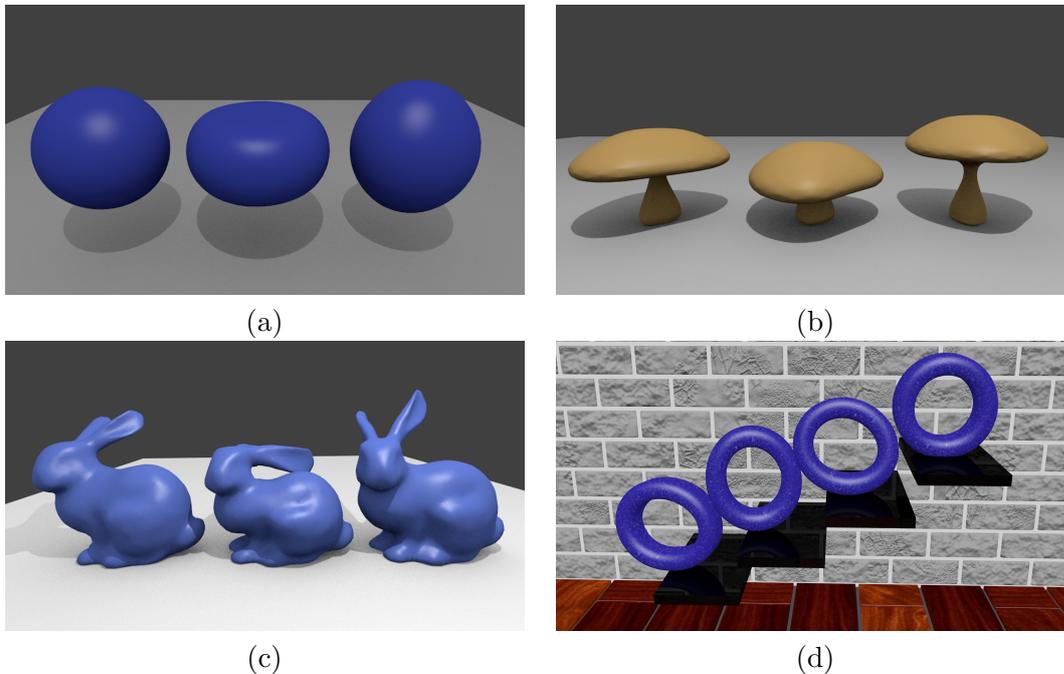


Figure 6.20 – (a-b) Balle et champignon : non-déformé (à gauche), compressé (au milieu) et étiré (à droite). (c) Lapin : non-déformé (à gauche), déformé sous l'effet de la gravité (au milieu) et après lui avoir tiré l'oreille gauche (à droite). (d) Utilisation de contraintes multiples (collisions et préservation de volume).

- Les Fig. 6.20(a-b) présente une balle et un champignon subissant une pression ou un étirement appliqué sur les parties haute et basse des objets. En ce qui concerne la sphère, une pression surfacique est appliquée de manière périodique alternant entre 5 et -5 kPa avec une période de 5 secondes. En ce qui concerne le champignon, les nœuds de son pied sont fixés au sol. Puis il subit une gravité périodique alternant entre 2 et -2 kPa avec une période de 5 s. Sans l'emploi de la contrainte de volume, nous obtenons une perte maximale de volume de 7,22 %. Le surcout en calcul de l'ajout de la contrainte de volume est de l'ordre de 1 à 2 % du temps global de calculs.

- La Fig. 6.20(c) présente le "Stanford Bunny" dont les nœuds du bas sont fixés au sol et qui est soumis à une gravité de 10 Pa relâchée au bout de 10 secondes. Une force ponctuelle est également appliquée sur l'oreille gauche. Nous voyons qu'il est possible de simuler des changements de courbures importantes avec le modèle CST-DKT combiné à la contrainte de volume. Par ailleurs, la formulation co-rotationnelle permet au modèle de subir des déformations issues d'une rotation, c'est-à-dire de prendre en compte une non-linéarité géométrique. Notons que nous obtenons une perte maximale de volume de 19,5 % si la contrainte de volume n'est pas employée.
- La Fig. 6.20(d) présente un tore soumis à la gravité qui descend un escalier. Cette simulation montre qu'il est possible d'utiliser à la fois la contrainte de volume et une contrainte issue du traitement des collisions que l'objet peut subir durant la simulation. La méthode des pénalités a été employée pour ce traitement. Nous observons que le tore subit une déformation lisse en flexion durant la collision.

La Table 6.1 récapitule les caractéristiques des maillages employés ainsi que les résultats obtenus en donnant également les temps de calcul des simulations pour une frame en séquentiel et en utilisant le GPU (cette parallélisation GPU sera explicitée juste après).

Maillage	# sommets	# triangles	Perte de volume	Temps (CPU)	Temps (GPU)	Partitions
Balle sans contrainte	5002	10000	7,22 %	1,23 s	51 ms	10
Balle avec contrainte	5002	10000	0 %	1,41s	56 ms	10
Champignon avec c.	2002	4000	19,5 %	0,95 s	31 ms	12
Champignon sans c.	2002	4000	0%	1,07 s	35 ms	12
Lapin avec contrainte	5002	10000	0 %	1,35 s	65 ms	11
Tore avec contrainte	2400	4800	0 %	1,02 s	52 ms	11

Table 6.1 – Perte de volume et temps de calcul. Le schéma d'intégration d'Euler implicite a été employé avec une résolution par la méthode du gradient Conjugué (20 itérations). Le temps représente le temps utilisé pour une frame.

La Fig. 6.21 présente la simulation d'une balle soumise à la gravité et passant entre deux plans rigides. Cette sphère dispose de 2000 triangles et de 1002 nœuds avec un module de Young de 3 kPa, un coefficient de Poisson de 0.45, une épaisseur de 10 cm et une masse volumique de 100 kg/m³. La contrainte de volume est prise en compte après le traitement des collisions avec les deux plans rigides puisqu'elle agit directement sur les déplacements des nœuds. Il y a deux cas à considérer durant la collision :

- Les vecteurs ∇_i des nœuds i impliqués dans la collision sont nuls.
- Les vecteurs ∇_i des nœuds i non impliqués dans la collision correspondent à la direction des plans de collision, c'est-à-dire qu'ils engendrent la projection de ces nœuds sur les plans. La contrainte de volume agit ainsi sur les nœuds mais sans violer la contrainte d'interpénétration et provoque un effet d'étalement de la balle. Au final, pour garantir un volume constant de la sphère tout au long de la simulation, les effets de la contrainte de volume se répartissent sur les nœuds i non impliqués dans la collision.

Cette simulation montre qu'il est possible de considérer la contrainte de volume avec d'autres contraintes comme celles engendrées par le traitement des collisions.

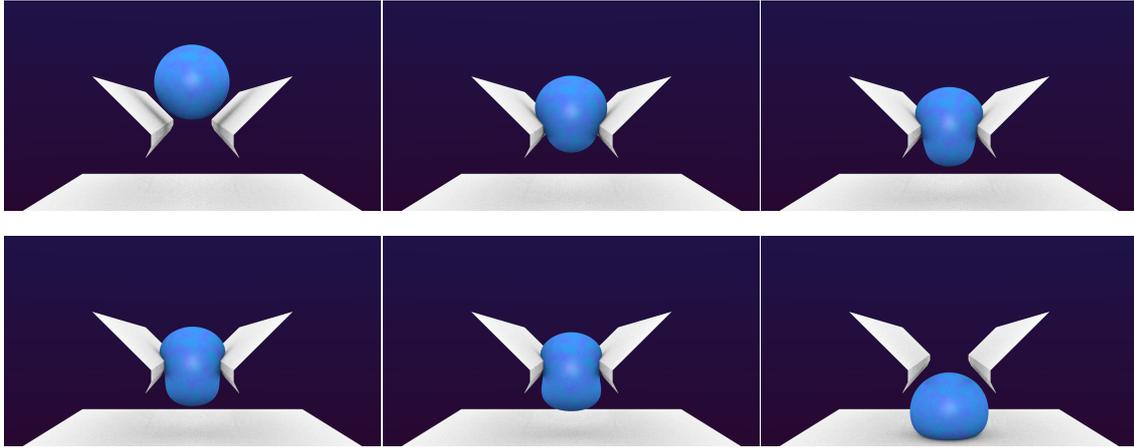


Figure 6.21 – Déformation d'une balle soumise à la gravité en collision avec deux plans rigides.

Évaluation en performance. Pour évaluer les performances de notre implémentation parallèle sur GPU, nous comparons le temps CPU et GPU de la simulation d'un cylindre en traction en faisant varier son nombre de triangles. Cette simulation a été effectuée en employant comme carte graphique une Tesla K20 avec 512 cœurs et comme CPU un Intel Xeon 3.4 Ghz. La Fig. 6.22 présente les temps obtenus en considérant 50 itérations pour la méthode du Gradient Conjugué. Un faible nombre de triangles implique un faible nombre de triangles par partition. Le *speedup* est ainsi faible dans ce cas entre la version CPU et GPU. Par contre, pour un grand nombre de triangles, nous obtenons un *speedup* de 12 entre la version CPU et GPU.

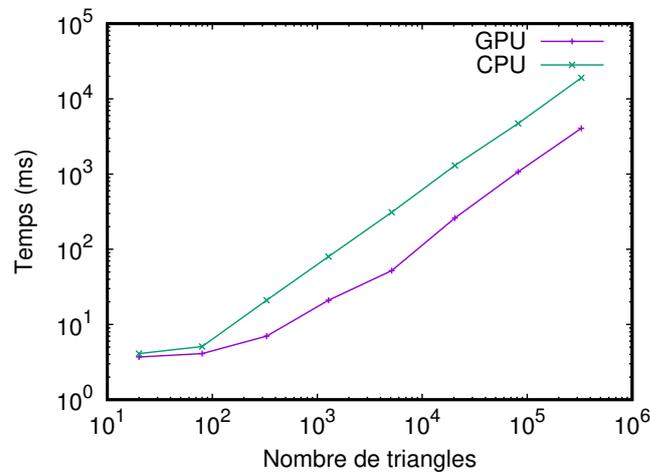


Figure 6.22 – Comparaison du temps entre la version CPU et GPU de la simulation d'un cylindre en traction en fonction du nombre de triangles du maillage surfacique (échelle log-log).

6.8 Discussion

Nous avons présenté dans ce chapitre, le modèle physique CST-DKT qui combine les éléments finis surfaciques CST aux éléments finis surfaciques DKT. Ce modèle physique permet ainsi de modéliser à la fois un comportement en membrane et un comportement en flexion avec un coût moindre en calcul que lors de l'emploi d'un modèle physique basé sur

une discrétisation volumique de l'objet 3D. Par ailleurs, ce modèle physique gère à la fois des déformations locales ou des déformations importantes, alors qu'il est parfois difficile de le faire avec un modèle éléments finis 3D dans lequel les éléments peuvent s'inverser. Les déformations importantes sont permises par l'utilisation d'une formulation co-rotationnelle permettant de séparer la partie déplacement de l'objet de la partie déformation, donnant un résultat similaire à l'emploi d'un tenseur de déformation invariant en rotation, mais avec un coût de calculs moindre. Cette approche est désormais classique en simulation.

Concernant le modèle éléments finis CST-SDKT, une limitation peut provenir dans le choix des paramètres physiques puisqu'il nécessite la valeur d'une épaisseur dont la valeur ne peut pas être en corrélation directe avec les valeurs des paramètres d'un modèle éléments finis basés sur des éléments volumiques de référence. Ce paramètre se calibre ainsi de manière empirique en testant et comparant avec des résultats issus par exemple d'un modèle utilisant des éléments finis volumiques.

Dans ce chapitre, nous avons également présenté une méthode de préservation de volume qui assure le caractère incompressible des objets 3D simulés malgré l'emploi d'un modèle physique surfacique. Cette méthode a l'avantage de pouvoir être appliquée avec n'importe quel modèle physique comme par exemple celui des masses-ressorts. Par contre, nous pouvons noter qu'elle peut présenter quelques limitations :

- La première limitation provient de la manière dont la contrainte de volume est appliquée. A l'heure actuelle, les nœuds sont déplacés une fois la déformation physique calculée. Cela signifie que si le pas de temps de la méthode d'intégration utilisée dans la boucle de simulation est trop important, le déplacement induit pour satisfaire la contrainte de volume peut être important et engendrer des instabilités.
- La deuxième limitation provient de la courbure du maillage, c'est-à-dire si celui-ci présente des arêtes vives ou non. En effet, notre méthode de préservation de volume est bien adaptée pour des maillages d'aspects lisses. Dans le cas où il y aurait des arêtes vives, la direction du gradient du volume peut ne pas être définie correctement et par conséquent engendrée un déplacement de nœuds non réaliste pour satisfaire la contrainte de volume. Par contre, la possibilité d'un contrôle local de cette contrainte (avec l'emploi de ∇_i spécifique pour le nœud i) permet la gestion de ces cas particuliers. Une autre solution peut consister à rajouter des éléments au niveau de la courbure importante du maillage afin de gérer ce problème plus globalement et en amont de la simulation. Ces modifications n'engendrent pas de surcoût important en calculs.

Pour finir, rappelons que les travaux présentés dans ce chapitre ont été réalisés dans le cadre de la conception d'un simulateur pour l'apprentissage des gestes médicaux de l'accouchement, et plus particulièrement pour l'apprentissage de l'usage des forceps dans le cadre d'une extraction par instruments. Le modèle physique par éléments finis CST-DKT, combiné à l'emploi de la contrainte de volume, a ainsi été employé pour modéliser la tête fœtale et reproduire les déformations induites sur celle-ci durant l'accouchement par les pressions utérines et les forceps. Les résultats obtenus dans ce cadre seront présentés au sein du chapitre 8 dédié à la présentation de ce simulateur médical.

Contributions

Publications

- M. Bailet, F. Zara, and E. Promayon. *A FEM based volume preserving method for closed thin-shells*. Soumis le 11/01/2019 à *Graphical Models*. Nous devons répondre aux remarques des reviewers afin de soumettre une nouvelle version de l'article. Nous tenons à le faire, car cette publication rassemble l'ensemble des résultats obtenus durant la thèse de Mathieu Bailet avec la présentation du modèle CST-DKT, la méthode de préservation de volume et la parallélisation de la simulation.
- M. Bailet, F. Zara, and E. Promayon. Biomechanical model of the fetal head for interactive childbirth simulation. In *SURGETICA 2014*, Dec. 2014.
- M. Bailet. Modèles biomécaniques pour la simulation interactive de l'accouchement. *Thèse de doctorat en informatique*, Université de Grenoble, dec 2014.
- M. Bailet, F. Zara, and E. Promayon. Shell finite element model for interactive fetal head deformation during childbirth. *CMBBE - Computer Methods in Biomechanics and Biomedical Engineering*, 16 :312–314, Sept. 2013.

Logiciel

- **NExT - New Experimental Tool** - Librairie de simulation physique intégrant plusieurs modèles physiques (masses-ressorts, masses-tenseurs, coque, éléments finis volumiques). Le développement de cette librairie a été initié par Mathieu Bailet durant sa thèse, et a été poursuivi par lui-même et par différents ingénieurs (Matthieu Coquet, Florian Canezin) durant le projet SAGA. Son développement se poursuit actuellement durant la thèse de Charles Barnouin et lors d'encadrement de différents projets d'étudiants de Master.

Bibliographie

- J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni. SOFA - an open source framework for medical simulation. In *Medicine Meets Virtual Reality (MMVR)*, volume 15, pages 1–6, Long Beach, California, Etats-Unis, February 2007.
- J. Allard, H. Courtecuisse, and F. Faure. Implicit FEM Solver on GPU for Interactive Deformation Simulation. In W. mei W. Hwu, editor, *GPU Computing Gems Jade Edition, Applications of GPU Computing Series*, pages 281–294. Elsevier, Nov. 2011a. doi : 10.1016/B978-0-12-385963-1.00021-6. URL <https://hal.inria.fr/inria-00589200>.
- J. Allard, H. Courtecuisse, and F. Faure. Implicit FEM Solver on GPU for Interactive Deformation Simulation. In W. mei W. Hwu, editor, *GPU Computing Gems Jade Edition, Applications of GPU Computing Series*, pages 281–294. Elsevier, Nov. 2011b. doi : 10.1016/B978-0-12-385963-1.00021-6. URL <https://hal.inria.fr/inria-00589200>.
- M. Bailet. *Modèles biomécaniques pour la simulation interactive de l'accouchement*. Thèse de doctorat en informatique, Université Joseph Fourier, Dec. 2014. URL <https://hal.archives-ouvertes.fr/tel-01158380>.
- M. Bailet, F. Zara, and E. Promayon. Shell finite element model for interactive fetal head deformation during childbirth. *CMBBE - Computer Methods in Biomechanics and Biomedical Engineering*, 16 :312–314, Sept. 2013.
- M. Bailet, F. Zara, and E. Promayon. Biomechanical model of the fetal head for interactive childbirth simulation. In *SURGETICA 2014*, Dec. 2014.
- M. Bailet, F. Zara, and E. Promayon. A FEM based volume preserving method for closed thin-shells. *soumis - Computer Graphics Forum*, 2016.
- D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98*, pages 43–54, New York, NY, USA, 1998. ACM. ISBN 0-89791-999-8. doi : 10.1145/280814.280821. URL <http://doi.acm.org/10.1145/280814.280821>.
- S. Bartels. Approximation of large bending isometries with discrete kirchhoff triangles. *SIAM Journal on Numerical Analysis*, 51(1) :516–525, 2013. doi : 10.1137/110855405. URL <https://doi.org/10.1137/110855405>.
- A. Bartzzaghi, M. Cremonesi, N. Parolini, and U. Perego. An explicit dynamics GPU structural solver for thin shell finite elements. *Computers & Structures*, 154(0) :29 – 40,

2015. ISSN 0045-7949. doi : <http://dx.doi.org/10.1016/j.compstruc.2015.03.005>. URL <http://www.sciencedirect.com/science/article/pii/S0045794915000875>.
- J.-L. Batoz, K.-J. Bathe, and L.-W. Ho. A study of three-node triangular plate bending elements. *International Journal for Numerical Methods in Engineering*, 15(12) :1771–1812, 1980. ISSN 1097-0207. doi : [10.1002/nme.1620151205](https://doi.org/10.1002/nme.1620151205). URL <http://dx.doi.org/10.1002/nme.1620151205>.
- V. Baudet, M. Beuve, F. Jaillet, B. Shariat, and F. Zara. Integrating Tensile Parameters in Mass-Spring System for Deformable Object Simulation. Technical Report RR-LIRIS-2009-034, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon, Sept. 2009. URL <http://liris.cnrs.fr/publis/?id=4449>.
- T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 1999. ISBN 471-98773-5.
- J. Bosman, C. Duriez, and S. Cotin. Connective Tissues Simulation on GPU. In J. Bender, J. Dequidt, C. Duriez, and G. Zachmann, editors, *Workshop on Virtual Reality Interaction and Physical Simulation*. The Eurographics Association, 2013. ISBN 978-3-905674-57-6. doi : [10.2312/PE.vriphys.vriphys13.041-050](https://doi.org/10.2312/PE.vriphys.vriphys13.041-050).
- D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4) :251–256, 1979.
- M. Bro-Nielsen. Finite element modeling in surgery simulation. *Proceedings of the IEEE*, 86(3) :490–503, Mar 1998. ISSN 0018-9219. doi : [10.1109/5.662874](https://doi.org/10.1109/5.662874).
- C. Chang, A. Borgart, A. Chen, and M. Hendriks. Direct gradient projection method with transformation of variables technique for structural topology optimization. *Structural and Multidisciplinary Optimization*, pages 1–13, 2013. ISSN 1615-147X. doi : [10.1007/s00158-013-0964-z](https://doi.org/10.1007/s00158-013-0964-z). URL <http://dx.doi.org/10.1007/s00158-013-0964-z>.
- M. G. Choi, S. Yong Woo, and H.-S. Ko. Real-time simulation of thin shells. *Computer Graphics Forum*, 26(3) :349–354, 2007. ISSN 1467-8659. doi : [10.1111/j.1467-8659.2007.01057.x](https://doi.org/10.1111/j.1467-8659.2007.01057.x). URL <http://dx.doi.org/10.1111/j.1467-8659.2007.01057.x>.
- H. Delingette. Biquadratic and quadratic springs for modeling st venant kirchhoff materials. In F. Bello and P. Edwards, editors, *Biomedical Simulation*, volume 5104 of *Lecture Notes in Computer Science*, pages 40–48. Springer Berlin Heidelberg, 2008a. ISBN 978-3-540-70520-8. doi : [10.1007/978-3-540-70521-5_5](https://doi.org/10.1007/978-3-540-70521-5_5). URL http://dx.doi.org/10.1007/978-3-540-70521-5_5.
- H. Delingette. Triangular springs for modeling nonlinear membranes. *Visualization and Computer Graphics, IEEE Transactions on*, 14(2) :329–341, 2008b. ISSN 1077-2626. doi : [10.1109/TVCG.2007.70431](https://doi.org/10.1109/TVCG.2007.70431).
- R. Diziol, J. Bender, and D. Bayer. Robust real-time deformation of incompressible surface meshes. In *Proceedings of the 2011 ACM Siggraph/Eurographics Symposium on Computer Animation*, SCA'11, pages 237–246, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0923-3. doi : [10.1145/2019406.2019438](https://doi.org/10.1145/2019406.2019438). URL <http://doi.acm.org/10.1145/2019406.2019438>.

- C. Felippa and B. Haugen. A unified formulation of small-strain corotational finite elements : I. theory. *Computer Methods in Applied Mechanics and Engineering*, 194(21) :2285–2335, 2005.
- A. V. Gelder. Approximate simulation of elastic membranes by triangulated spring meshes. *Journal of Graphics Tools*, 3(2) :21–41, 1998. doi : 10.1080/10867651.1998.10487490.
- D. M. Gil Rama and M. Zehn. Efficient co-rotational 3-node shell element. *American Journal of Engineering and Applied Sciences*.
- K. Golec. *Hybrid 3D Mass Spring System for Soft Tissue Simulation*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1, feb 2018.
- E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 62–67. Eurographics Association, 2003.
- P. E. Hammer, M. S. Sacks, P. J. del Nido, and R. D. Howe. Mass-spring model for simulation of heart valve tissue mechanical behavior. *Annals of Biomedical Engineering*, 39(6) :1668–1679, Jun 2011. ISSN 1573-9686. doi : 10.1007/s10439-011-0278-5. URL <https://doi.org/10.1007/s10439-011-0278-5>.
- M. Hong, M.-H. Choi, S. Jung, S. Welch, and J. Trapp. Effective constrained dynamic simulation using implicit constraint enforcement. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4520–4525, 2005. doi : 10.1109/ROBOT.2005.1570816.
- M. Hong, S. Jung, M.-H. Choi, and S. Welch. Fast volume preservation for a mass-spring system. *Computer Graphics and Applications, IEEE*, 26(5) :83–91, 2006. ISSN 0272-1716. doi : 10.1109/MCG.2006.104.
- G. Irving, C. Schroeder, and R. Fedkiw. Volume conserving finite element simulations of deformable models. In *ACM Transactions on Graphics (TOG)*, volume 26, page 13. ACM, 2007.
- M. Kot and H. Nagahashi. Mass spring models with adjustable poisson’s ratio. *Vis. Comput.*, 33(3) :283–291, Mar. 2017. ISSN 0178-2789. doi : 10.1007/s00371-015-1194-8. URL <https://doi.org/10.1007/s00371-015-1194-8>.
- M. Kubale. *Graph Colorings*. Contemporary mathematics (American Mathematical Society) v. 352. American Mathematical Society, 2004. ISBN 9780821856871. URL <http://books.google.fr/books?id=nEmcID52qtsC>.
- M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’02, pages 49–54, New York, NY, USA, 2002. ACM. ISBN 1-58113-573-4. doi : 10.1145/545261.545269. URL <http://doi.acm.org/10.1145/545261.545269>.
- M. Müller, B. Heidelberger, M. Teschner, and M. Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24(3) :471–478, July 2005. ISSN 0730-0301. doi : 10.1145/1073204.1073216. URL <http://doi.acm.org/10.1145/1073204.1073216>.

-
- M. Müller, J. Stam, D. James, and N. Thiéry. Real time physics : Class notes. In *ACM SIGGRAPH 2008 Classes*, SIGGRAPH '08, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781450378451. doi : 10.1145/1401132.1401245. URL <https://doi.org/10.1145/1401132.1401245>.
- S. Natsupakpong and M. C. Çavuşoğlu. Determination of elasticity parameters in lumped element (mass-spring) models of deformable objects. *Graphical Models*, 72(6) :61–73, 2010. ISSN 1524-0703. doi : <http://dx.doi.org/10.1016/j.gmod.2010.10.001>.
- M. Nesme, M. Marchal, E. Promayon, M. Chabanas, Y. Payan, and F. Faure. Physically realistic interactive simulation for biological soft tissues. *Recent Research Developments in Biomechanics*, 2 :1–22, 2005a. URL <https://hal.archives-ouvertes.fr/hal-00080378>.
- M. Nesme, Y. Payan, and F. Faure. Efficient, physically plausible finite elements. In J. Dingliana and F. Ganovelli, editors, *Eurographics 2005, Short papers, August, 2005*, Trinity College, Dublin, Irlande, 2005b.
- H. Oudin. Méthode des éléments finis. Lecture, Sept. 2008. URL <https://cel.archives-ouvertes.fr/cel-00341772>.
- E. Promayon, P. Baconnier, and C. Puech. Physically-Based Deformations Constrained in Displacements and Volume. *Computer Graphics Forum*, 1996. ISSN 1467-8659. doi : 10.1111/1467-8659.1530155.
- E. Sifakis and J. Barbic. Fem simulation of 3d deformable solids : A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, pages 20 :1–20 :50, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1678-1. doi : 10.1145/2343483.2343501. URL <http://doi.acm.org/10.1145/2343483.2343501>.

Amélioration du modèle physique des masses-ressorts

Je présente dans ce chapitre les travaux effectués dans le cadre de la thèse de Karolina Golec financée par le LabEx PRIMES (Laboratoire d'Excellence "Physique, Radiobiologie, Imagerie Médicale et Simulation"). Notre objectif consistait à étendre le modèle discret des masses-ressorts pour pouvoir l'utiliser dans le cadre de la simulation dynamique de tissus mous. En effet, notre souhait était de reproduire numériquement les résultats de lois mécaniques non-linéaires mises en évidence lors d'expérimentations effectuées sur le parenchyme d'organes de l'abdomen et du cerveau par le LBMC (Laboratoire de Biomécanique et Mécanique des Chocs). Cet encadrement de thèse a ainsi permis une collaboration active avec Stéphane Nicolle (LBMC) et Jean-François Palierne (laboratoire de Physique de l'ENS de Lyon) auteurs de ces expérimentations et de la formulation des nouvelles lois de comportement mécanique mises en évidence lors de ces tests. Ils ont ainsi supervisé le travail de doctorat de Karolina Golec avec Guillaume Damiand et moi-même (dans la continuité de la thèse de Elsa Fléchon).

Sommaire

7.1	Contexte de ce travail de recherche	206
7.2	Travaux existants sur les modèles masses-ressorts	208
7.3	Proposition d'un système hybride	215
7.4	Formulation des constantes de raideur des ressorts	217
7.5	Formulation des forces de correction	219
7.6	Validation de notre système hybride	222
7.7	Proposition de forces non-linéaires pour les ressorts	229
7.8	Discussion	232

7.1 Contexte de ce travail de recherche

Dans le chapitre 3, nous avons vu que le système masses-ressorts est un modèle physique qui a été initialement proposé dans le domaine de l'Informatique Graphique par [Terzopoulos et al. \(1987\)](#) pour simuler le comportement d'objets dits déformables à moindre coût de calculs que la MEF. Dans l'approche masses-ressorts, les objets sont discrétisés en un ensemble de particules ou masses qui sont reliées entre elles par des ressorts définissant ainsi la topologie de l'objet. Ces ressorts sont caractérisés par trois paramètres : une longueur dite au repos, une constante de raideur et un coefficient d'amortissement (dans le cadre de l'emploi de ressorts dits amortis). La déformation des ressorts se traduisant par leur élongation permet ainsi de déformer la représentation géométrique de l'objet au cours de la simulation. Ce modèle physique, ainsi basé sur un concept mécanique assez simple, a souvent été employé en animation où un résultat visuellement satisfaisant est suffisant.

La question qui se pose lors de l'emploi de ce modèle physique concerne en premier lieu sa configuration. C'est-à-dire qu'il s'agit de positionner les ressorts entre les particules afin de reproduire correctement les déformations du matériau. Mais la déformation d'un ressort ne se produit que dans la direction de celui-ci, alors que les déformations d'un matériau se produisent dans toutes les directions de l'espace. Autrement dit, le premier enjeu réside dans la mise en place de la connexion entre les particules. Ensuite, dans un contexte biomécanique ou médical, un second enjeu consiste à générer des résultats suffisamment précis pour qu'ils puissent être validés et acceptés dans un domaine exigeant en terme de précision. Il s'agit alors de réussir à relier les coefficients mécaniques connus du matériau (tels que le module de Young ou le coefficient de Poisson) avec ceux des ressorts (raideur et amortissement). D'autant plus qu'un ressort étant par nature linéaire, un système masses-ressorts ne peut reproduire qu'un comportement linéaire du point de vue mécanique, alors que les tissus mous suivent généralement une loi de comportement non-linéaire.

Nous nous sommes ainsi intéressés à ces problématiques dans le cadre de la thèse de Karolina [Golec \(2018\)](#) dont le co-encadrement a permis de débiter une collaboration avec Stéphane Nicolle (équipe Biomécanique des chocs du LBMC) et Jean-François Palierne (laboratoire de physique de l'ENS de Lyon). La problématique de la thèse de Karolina Golec concernait ainsi l'utilisation du modèle physique des masses-ressorts dans le but de reproduire le comportement d'un matériau dont la déformation est régie par une loi visco-élastique non linéaire. Cette loi de comportement a été mise en évidence à l'issue d'expérimentations effectuées sur tissus réels (parenchyme d'organes de l'abdomen et du cerveau) par Stéphane Nicolle et Jean-François Palierne ([Nicolle, S. ; Noguer, L. ; Palierne \(2013\)](#)). Il s'agissait alors durant ce travail de doctorat, de modifier la formulation initiale de l'approche masses-ressorts pour obtenir ce comportement mécanique non-linéaire et ainsi être capable de reproduire le comportement en cisaillement de chaque tissu parenchymateux en petites et grandes déformations à différentes vitesses de déformation.

Pour étendre la formulation du modèle physique des masses-ressorts, nous sommes repartis de la formulation de Vincent Baudet (dont j'avais supervisé la fin de thèse dès mon arrivée au LIRIS en septembre 2005). Rappelons que cette formulation ([Baudet et al. \(2009\)](#)) n'est valide que pour des matériaux ayant un coefficient de Poisson égale à 1/4 décomposés en éléments hexaédriques et pour des simulations en petites déformations. C'est pourquoi, nous avons commencé par rajouter de la non-linéarité dans la formulation des

forces exercées par les ressorts (Golec et al. (2016)). Ainsi grâce à l’ajustement de différents paramètres, nous avons reproduit les résultats issus des expérimentations du LBMC. Puis, nous avons proposé une nouvelle configuration du système masses-ressorts permettant à la fois de repousser la limite d’instabilité du système en rajoutant des connexions entre les particules, et de contrôler le volume du matériau déformé en rajoutant des forces de corrections différentes de celles initialement proposées par Vincent Baudet (Golec et al. (2017, 2019)). Nous reproduisons ainsi le comportement d’un matériau en petites et grandes déformations défini par un module de Young et un coefficient de Poisson qui n’est pas forcément égal à $1/4$ en utilisant un système masses-ressorts basé sur un maillage cubique.

Organisation de ce chapitre : Ce chapitre se poursuit de la manière suivante :

- Dans la section 7.2, je présente l’état de l’art sur les formulations analytiques proposées dans la littérature pour le calcul des raideurs d’un système masses-ressorts.
- Dans la section 7.3, je présente l’approche générale adoptée durant la thèse de Karolina Golec pour palier aux deux limitations du modèle des masses-ressorts.
- Dans la section 7.4, je présente la nouvelle formulation des raideurs des ressorts que nous avons élaborée afin de prendre en compte les propriétés mécaniques du matériau, avec notamment l’ajout de ressorts placés sur les diagonales des faces des éléments cubiques de la discrétisation.
- Dans la section 7.5, je présente les forces de corrections non-linéaires que nous avons ajoutées dans le système afin de sortir de la loi de Hooke initiale.
- Dans la section 7.6, je présente les résultats obtenus avec notre modèle masses-ressorts hybride.
- Puis dans la section 7.7, je présente une nouvelle formulation non-linéaire des forces exercées par les ressorts, permettant de reproduire les lois de comportement mises en évidence par le LBMC lors d’expérimentations sur des tissus réels.
- Enfin pour finir ce chapitre, la section 7.8 dresse un bilan de cette extension du modèle physique des masses-ressorts.

7.2 Travaux existants sur les modèles masses-ressorts

En Informatique Graphique, le modèle physique des masses-ressorts est souvent employé pour effectuer des animations interactives (Chen et al. (1998)) puisqu'il est simple à mettre en œuvre et qu'il permet aussi bien de simuler des objets 2D, comme des textiles (Bridson et al. (2002)), ou des objets 3D dits déformables (Teschner et al. (2004)). Mais son emploi devient un peu plus compliqué quand nous souhaitons l'utiliser dans le cadre d'applications médicales où la précision reste un critère important. En effet, la mise en place de la modélisation physique d'un objet par un système masses-ressorts nécessite la définition des propriétés mécaniques du système, c'est-à-dire la définition des masses et des raideurs des ressorts, alors que le lien entre ces raideurs et les propriétés mécaniques d'un matériau (module de Young et coefficient de Poisson par exemple) n'est pas évident à mettre en place. C'est pourquoi au fil des ans, des travaux se sont focalisés sur cet aspect, ainsi que sur le contrôle du volume de l'objet simulé, afin d'assurer le caractère incompressible des tissus mous dans le cadre d'une simulation médicale.

En ce qui concerne les travaux sur la **formulation des raideurs**, deux approches existent : une approche tente d'identifier les paramètres du système en fonction de données réelles issues d'expérimentations mécaniques ou de vidéos ; et une autre vise à obtenir une formulation analytique des raideurs en fonction des paramètres mécaniques du matériau.

Notre objectif étant l'obtention d'une solution qui soit applicable pour différents tissus mous définis par leurs propriétés mécaniques, nous nous sommes concentrés sur la recherche d'une **formulation analytique des raideurs intégrant les propriétés mécaniques (module de Young, coefficient de Poisson) du matériau**.

Le Tableau 7.1 fournit un récapitulatif des formulations analytiques des raideurs qui ont été établies dans la littérature. Je donne ensuite quelques détails sur ces formulations, afin de positionner nos propres travaux sur cette thématique de recherche.

Table 7.1 – Récapitulatif des formulations des raideurs proposées dans la littérature dans le cadre de la simulation d'objets déformables. Tableau modifié initialement issu de la thèse de Karolina Golec (2018).

N°	Auteurs	Formulation des raideurs	Propriétés géométriques et mécaniques du matériel	Bases physiques
1	Wilhelms and Van Gelder (1997)	$k_{tri} = \frac{A_1 + A_2}{l_0^2}$ $k_{tri}^a = c \sum_i \frac{A_i}{3}$	maillage triangulaire, membrane élastique	approximation géométrique
2	Van Gelder (1998)	$k_{tri}^c = \frac{E_2 \sum_e A_e}{ l_0 ^2}$ $k_{tetra}^c = \frac{E \sum_e V_e}{ l_0 ^2}$	maillage triangulaire ou maillage tétraédrique, isotropique, élastique linéaire	approximation géométrique

La suite sur la page suivante

TABLE 7.1 – Suite du tableau récapitulatif des formulations des raideurs.

N°	Auteurs	Formulation des raideurs	Propriétés géométriques et mécaniques du matériel	Bases physiques
3	Maciel et al. (2003)	$k_{hexa} = \frac{k_0 \cos_0 + \sum_i k_i \cos_i}{l_0 + \sum_i l_i \cos_i}$	objets élastiques linéaires, maillage hexaédrique avec ressorts diagonaux	ajustement angulaire de la formulation de Van Gelder (1998)
4	Lloyd et al. (2007)	$k_{tri} = \sum_e E h \frac{\sqrt{3}}{4}$ $k_{rect}^e = \sum_e \frac{5}{16} h E$ $k_{rect}^d = \frac{7}{16} h E$ $k_{tetra} = \sum_e \frac{2\sqrt{2}}{25} l E$ $k_{tetra}^V = \sum_e \frac{2\sqrt{2}}{21} l E \frac{4}{5}$ $k_V = \frac{\sqrt{2}}{84} l^3 E \frac{2}{5}$	élastique linéaire, raideurs pour la préservation du volume pour maillage tétraédrique	basé sur modèle FEM de référence
5	Zerbato et al. (2007)	$k_{tetra} = \frac{E_1 + E_2}{2} \frac{\sum_i V_i}{l_0^2}$	échantillons de foie	approximation géométrique
6	Delingette (2008a,b)	$k_{tri} = \frac{2 \cot^2 \alpha_i (\lambda + \mu) + \mu}{16 A_p} = \frac{E (2 \cot^2 \alpha_i + 1 - \nu)}{16(1 - \nu^2) A_p}$ $C_{tri} = \frac{2 \cot \alpha_i \cot \alpha_j (\lambda + \mu) - \mu}{16 A_p} = \frac{E (2 \cot \alpha_i \cot \alpha_j + \nu - 1)}{16(1 - \nu^2) A_p}$ $k_{tetra} = \frac{(\lambda + \mu) V}{2} (D_i \cdot D_j)^2 + \frac{\mu V}{2} \ D_i\ ^2 \ D_j\ ^2$ $C_{tetra} = \frac{(\lambda + \mu) V}{2} (D_i \cdot D_j)(D_i \cdot D_k) + \frac{\mu V}{2} \ D_i\ ^2 (D_j \cdot D_k)$	élastique linéaire, membrane non-linéaire (Saint Venant Kirchhoff), maillage triangulaire ou maillage tétraédrique	basé sur l'énergie de déformation d'un triangle ou d'un tétraèdre
7	Arnab and Raja (2008)	$k_{tri}^E = E \sum_{i \in j} \frac{V_j}{l_i^2}$ $k_{tri}^G = E \sum_{i \in j} \frac{V_j}{2 l_i^2 (1 - \nu)}$ $k_{tri}^B = \frac{E}{3(1 - 2\nu)} \sum_{i \in j} \frac{V_j}{l_i^2}$	objets 3D élastiques	modèle géométrique avec propriétés élastiques

La suite sur la page suivante

TABLE 7.1 – Suite du tableau récapitulatif des formulations des raideurs.

N°	Auteurs	Formulation des raideurs	Propriétés géométriques et mécaniques du matériel	Bases physiques
8	Baudet et al. (2009)	$k_{rect}^e = \frac{E(j^2(3\nu+2)-i^2)}{4 l_0 h_0(1+\nu)}$ $k_{rect}^d = \frac{E(l_0^2+h_0^2)}{4 l_0 h_0(1+\nu)}$ $k_{hexa}^e = \frac{E l_0(4\nu+1)}{8(1+\nu)}$ $k_{hexa}^d = \frac{E i \sum_{j \in \{x_0, y_0, z_0\}} j^2}{8(1+\nu) \prod_{\{l \in \{x_0, y_0, z_0\}, l \neq i\}} l}$	matériel élastique linéaire avec maillage rectangulaire et hexaédrique	paramétrisation basée sur déformations en cisaillement et étirement
9	Natsupakpong and Çavuşoğlu (2010)	$k_{rect}^e = \frac{\lambda}{3} + \frac{\mu}{2}$ $k_{rect}^d = \left(\frac{5\lambda}{12} + \frac{3\mu}{4} \right)$	élastique linéaire, maillage rectangulaire	approximation d'un modèle FEM

1. Une des premières formulations a été proposée par [Wilhelms and Van Gelder \(1997\)](#) dans le cadre d'animation de personnages ou d'animaux représentés par différents composants : os, muscles, tissus mous et peau. Pour simuler la déformation de la peau, qui se produit lors de la contraction du muscle, celle-ci peut être vue comme une membrane élastique et être modélisée par un système masses-ressorts basé sur un maillage triangulaire. Pour un triangle donné, les raideurs k_{tri} des ressorts sont définies en fonction des surfaces A_1 et A_2 des deux triangles adjacents avec l_0 la longueur au repos du ressort considéré. La formulation des raideurs est établie pour un coefficient de Poisson nul. Une autre raideur k_{tri}^a est proposée pour les ressorts (de longueur nulle) assurant un lien entre la peau et les autres composants (muscle, os, tissus mous) également établie en fonction des aires A_i des triangles adjacents.
2. L'année suivante, [Van Gelder \(1998\)](#) propose une nouvelle formulation des raideurs pour un système masses-ressorts également utilisé pour modéliser le comportement d'une membrane élastique. Ce papier met en avant le fait que pour améliorer la précision de la simulation, il ne faut pas utiliser une unique formulation des raideurs, même dans le cas d'une membrane uniforme. Pour un triangle donné composé des ressorts a, b, c , la formulation de la raideur k_{tri}^c du ressort c est à nouveau établie en fonction des aires A_e des triangles qui lui sont adjacents, mais en intégrant cette fois-ci le module de Young 1D et 2D notés respectivement E_2 et E (avec $E_2 = E h$ où h est l'épaisseur du matériau), ainsi que le coefficient de Poisson ν du matériau simulé. La formulation complète est donnée par :

$$k_{tri}^c = \left(\frac{E_2}{1+\nu} \right) \frac{A_e}{|l_c|^2} + \left(\frac{E_2 \nu}{1-\nu^2} \right) \frac{|l_a|^2 + |l_b|^2 - |l_c|^2}{8A_e} \quad (7.1)$$

où l_a, l_b, l_c représentent les longueurs des ressorts a, b, c . Par contre, au niveau des résultats, [Van Gelder \(1998\)](#) utilise les formulations simplifiées mises dans le tableau 7.1 correspondant au cas où $\nu = 0$. Par ailleurs, les résultats fournis sont uniquement visuels, c'est-à-dire qu'il ne donne pas l'erreur obtenue par le système masses-ressorts par rapport, par exemple, à un modèle élément fini d'une membrane élastique qui pourrait être modélisée par le modèle CST (*Constant Strain Model* - cf.

- chapitre 2*). A noter que cette formulation, qui a également été étendue à un maillage tétraédrique, a ensuite été employée par différents auteurs ([Arnab and Raja \(2008\)](#); [Mollemans et al. \(2003\)](#); [Zerbato et al. \(2007\)](#)).
3. [Maciel et al. \(2003\)](#) proposent un système masses-ressorts basé sur un maillage hexaédrique avec des ressorts diagonaux internes aux éléments. Les raideurs sont calculées en fonction des angles qu'il y a entre les différents ressorts. Ils mettent en avant les inconvénients de cette approche, liés au fait que les angles évoluant durant la déformation du modèle doivent être re-calculés, engendrant un coût de calcul important.
 4. [Lloyd et al. \(2007\)](#) proposent différentes formulations en fonction de la topologie des éléments du système masses-ressorts considéré. Ces formulations ont été établies en se basant sur la formulation élément fini correspondante (par exemple en considérant un triangle CST pour le cas d'un maillage triangulaire). Ils fournissent ainsi les raideurs k_{tri} pour un maillage basé sur des triangles équilatéraux, (k_{rect}^e, k_{rect}^d) - e pour edge et d pour diagonal - pour un maillage rectangulaire ayant des ressorts internes diagonaux dans chacun des éléments. Ces formulations tiennent compte de l'épaisseur h du matériau et de son module de Young E . Par contre, ces formulations sont valides uniquement pour un coefficient de Poisson fixé à $\nu = 1/3$. Dans la cas d'un maillage tétraédrique, les raideurs (k_{tetra}, k_{tetra}^V) ont été établies en fonction de la longueur l des ressorts pour $\nu = 1/4$. La seconde offre une préservation du volume avec k_V le coefficient de la force de préservation de volume. Les résultats montrent une comparaison visuelle avec le modèle éléments finis correspondant en fonction des différents types d'éléments (triangles, quadrangles, tétraèdres).
 5. [Zerbato et al. \(2007\)](#) modifient la formulation de Van [Gelder \(1998\)](#) dans le cadre d'un système masses-ressorts basé sur des éléments tétraédriques et utilisent la contrainte de conservation de volume de [Mollemans et al. \(2003\)](#). La raideur d'un ressort est établie en fonction des modules de Young E_1 et E_2 des deux sommets connectés à celui-ci, et des volumes V_i des i tétraèdres incidents à ce ressort. Au final, le module de Young E de la formulation de Van Gelder est remplacé par la moyenne des termes E_1 et E_2 . Ensuite, les masses du système sont obtenues en employant un algorithme génétique minimisant l'erreur entre le résultat donné par le système masses-ressorts et des données réelles issues de l'imagerie médicale; puis ce procédé est de nouveau employé pour calibrer le système complet et affiner les paramètres des ressorts. La formulation de Van Gelder ne sert ainsi qu'à réduire l'espace des possibilités. Ils obtiennent une erreur de 9,49% entre le module de Young théorique et celui simulé.
 6. [Delingette \(2008a\)](#) propose une connexion formelle entre les ressorts et la mécanique des milieux continus dans le cadre de l'élasticité à une et deux dimensions. Il introduit ainsi la notion de ressorts quadratiques et bi-quadratiques permettant la modélisation de membranes non-linéaires (suivant la loi de Saint Venant-Kirchhoff). Ces ressorts sont caractérisés par une raideur en étirement k , et une raideur angulaire c . Ces raideurs sont exprimées en fonction de l'aire A des triangles et de leurs angles α . Cette notion est ensuite étendue à 3 dimensions dans [Delingette \(2008b\)](#) en considérant des maillages tétraédriques. Les formulations des raideurs k et c sont alors exprimées en fonction du volume V des tétraèdres et des vecteurs de forme D .
 7. [Arnab and Raja \(2008\)](#) proposent un système masses-ressorts basé sur un maillage triangulaire avec une contrainte de préservation de volume. Leur modèle intègre des ressorts de surfaces et des ressorts de volume. La raideur des ressorts positionnés sur la surface est établie en utilisant les méthodes proposées par [Lloyd et al. \(2007\)](#);

- [Maciel et al. \(2003\)](#). Pour les ressorts de volume, une raideur k_{tri}^E a été établie en fonction du module de Young E , une autre k_{tri}^G en fonction du module de cisaillement G (dépendant de E et ν), et une troisième k_{tri}^B en fonction du module de Bulk (ou de compressibilité). Pour l'évaluation de leur méthode, une comparaison est réalisée entre le module de Young employé dans la formulation des raideurs, et celui obtenu par la simulation en considérant la relation d'élasticité linéaire ($\sigma = E \epsilon$). Une comparaison visuelle avec un modèle par éléments finis est réalisé pour un cube.
8. [Baudet et al. \(2009\)](#) proposent une formulation pour des systèmes masses-ressorts basés sur des maillages rectangulaires et hexaédriques ayant des diagonales internes à chacun des éléments. Ces formulations ont été établies en considérant des tests de cisaillement et d'étirement. Elles intègrent le module de Young E et coefficient de Poisson ν du matériau, ainsi que la longueur des arêtes de l'élément considéré. Pour un élément rectangulaire de dimensions l_0 par h_0 , les arêtes i et j sont considérées avec $(i, j) \in \{l_0, h_0\}, i \neq j$. Pour un élément hexaédrique de dimensions $x_0 \times y_0 \times z_0$, les arêtes i sont considérées avec $i \in \{x_0, y_0, z_0\}$. Les formulations des raideurs peuvent être simplifiées dans le cas d'un élément carré ou cubique. Ces formulations ne sont valides en 2D que pour $\nu = 1/3$ et en 3D que pour $\nu = 1/4$. Des forces de corrections doivent ensuite être insérées dans le système pour permettre de simuler des matériaux ayant un coefficient de Poisson différents de ces valeurs. Les résultats donnés présentent une comparaison avec un modèle éléments finis.
 9. [Natsupakpong and Çavuşoğlu \(2010\)](#) proposent une formulation des raideurs pour un système masses-ressorts (appelé dans l'article *Lumped Element Method*) basé sur un maillage rectangulaire avec des diagonales internes. Les raideurs (k_{rect}^e, k_{rect}^d) des ressorts positionnés sur les arêtes et les diagonales sont établies en fonction des coefficients de lamé (λ, μ) du matériau. Elles ont été obtenues en effectuant une optimisation pour minimiser l'écart (selon deux normes) par rapport au modèle élément fini correspondant, sous les hypothèses d'isotropie et d'élasticité linéaire. Des résultats sont présentés pour différents types de maillages (triangulaires et rectangulaires en 2D, tétraédriques et hexaédriques en 3D) avec un coefficient de Poisson fixé à 0,3 et un module de Young à 10 *kPa*. Par contre, les formulations obtenues pour les éléments autres que rectangulaires ne sont pas données. Ils présentent ainsi l'erreur entre les paramètres (E, ν) initiaux et ceux calculés après la déformation, ainsi que l'erreur par rapport au modèle élément fini correspondant. Une comparaison est également faite par rapport aux formulations de [Baudet et al. \(2009\)](#); [Gelder \(1998\)](#); [Lloyd et al. \(2007\)](#) avec des résultats assez proches, sauf pour Van Gelder dont les erreurs sont beaucoup plus importantes. Enfin, ils tentent de contrôler le module de Young et le coefficient de Poisson de manière indépendante et concluent par le fait qu'il est difficile de contrôler le coefficient de Poisson (ce qui a également été observé par [Baudet et al. \(2009\)](#); [Lloyd et al. \(2007\)](#) et mis en évidence dans le chapitre 3).

Par rapport à ces travaux, notre objectif était de proposer une **formulation analytique des raideurs** qui intègre les propriétés mécaniques du matériau à simuler, et qui soit **valide quelque soit le module de Young E et le coefficient de Poisson ν** de l'objet simulé. L'idée est de pouvoir employer ce modèle pour la modélisation physique de différents tissus mous dans le cadre de nos applications médicales (modélisation de la loi de comportement de [Nicolle, S. ; Noguier, L. ; Paliarne \(2013\)](#), simulation de la descente du fœtus durant un accouchement, simulation de l'appareil respiratoire, etc.).

Le Tableau 7.2 récapitule les formulations données dans la littérature afin d'assurer la **conservation du volume** d'un objet modélisé en utilisant un système masses-ressorts. Il s'agit essentiellement de forces de correction appliquées sur les sommets du système. Pour référence, j'ai rappelé dans la première ligne du tableau, la formulation des forces initiales qui sont appliquées par un ressort sur un sommet j connecté à ses voisins i , avec k_s et k_d la raideur et le coefficient d'amortissement du ressort. Je donne ensuite quelques détails sur ces formulations afin de positionner nos travaux de recherche.

Table 7.2 – Récapitulatif des forces de corrections proposées dans la littérature pour la conservation du volume dans le cadre de l'emploi d'un système masses-ressorts.

N°	Auteurs	Force ajoutée dans le système masses-ressorts
0	<i>Formulation initiale</i>	$F_{ressort}^j = - \left(k_s (\ x_j - x_i\ - l_0) + k_d \frac{(v_j - v_i) \cdot (x_j - x_i)}{\ x_j - x_i\ } \right) \frac{x_j - x_i}{\ x_j - x_i\ }$
1	Bourguignon and Cani (2000)	$F_{tetra}^j = -k_s \left(\sum_{i=0}^3 \ x_i - x_B\ - \sum_{i=0}^3 \ x_i - x_B\ _{t=0} \right) \frac{x_j - x_B}{\ x_j - x_B\ }$ $F_{hexa}^j = - \left(k_s (\ x_j - x_B\ - \ x_j - x_B\ _{t=0}) + k_d \frac{(v_j - v_B) \cdot (x_j - x_B)}{\ x_j - x_B\ } \right) \frac{x_j - x_B}{\ x_j - x_B\ }$
2	Mollemans et al. (2003)	$F_{tetra}^{volj} = \sum_t (V_t - V_0) \frac{x_j - x_{b_t}}{\ x_j - x_{b_t}\ }$, $x_{b_t} = \frac{1}{4} \sum_{k=0}^4 x_k$, t indice des tétraèdres
3	Baudet et al. (2009)	$F_{rect}^{\perp e} = \frac{i F_i (1 - 3\nu)}{8j}$, $(i, j) \in \{l_0, h_0\}, i \neq j$ $F_{hexa}^{\perp e} = -\frac{F_i (4\nu - 1)}{16}$, $i \in \{x_0, y_0, z_0\}$
4	Jarrousse et al. (2010)	$W_v = W + \frac{1}{2} k (\det(\mathbb{F}) - 1)^2 = W + \frac{1}{2} k \left(\frac{V - V_0}{V_0} \right)^2$, $k = 2 \times 10^6$
5	Kot and Nagahashi (2015, 2017); Kot et al. (2014)	$F^j = F_\mu^j + F_*^j$, $F_\mu^j = -\kappa^\mu (\ x_j - x_i\ - l_0)$, $F_*^j = -q \kappa^\mu (\ x_j - x_i\ - l_0)$
6	Arnab and Raja (2008)	$F^j = k_{tri}^B (V - V_0) \alpha N$

1. Afin d'employer le modèle des masses-ressorts dans le cadre de simulations médicales, Bourguignon and Cani (2000) proposent une approche permettant de conserver le volume des éléments du maillage durant la déformation de l'objet. Une formulation est donnée pour les éléments d'un maillage tétraédrique et une autre pour les éléments d'un maillage hexaédrique. Il s'agit d'une force qui est appliquée sur les sommets des éléments. Cette force relie le barycentre de position x_B aux autres sommets de l'élément. Elle remplace la formulation usuelle des forces appliquées par des ressorts sur les sommets. A noter que pour les tétraèdres, ils prennent en compte uniquement la raideur k_s des ressorts, alors que pour les hexaèdres, ils considèrent également le coefficient d'amortissement k_d des ressorts. En terme de résultats, ils annoncent une variation de volume inférieure à 1,5%, mais en signalant que ces résultats sont très dépendants du type de matériau et des tests effectués.

2. [Mollema et al. \(2003\)](#) proposent une formulation pour les maillages tétraédriques. La force appliquée à un sommet j tient compte de la variation de volume des tétraèdres t auxquels il appartient et de la position x_{b_t} de leur barycentre respectif. Cette force est modulée par un coefficient α et est ajoutée à celle engendrée par le ressort avec $F^j = F_{ressort}^j + \alpha F_{tetra}^{volj}$.
3. Nous avons vu que la formulation des raideurs de [Baudet et al. \(2009\)](#) se combine à l'emploi de forces de correction ajoutées dans le système masses-ressorts, afin de simuler le comportement de matériaux ayant des coefficients de Poisson différents de $1/3$ ou $1/4$ dans le cas de maillages rectangulaires ou hexaédriques. Ces forces sont orthogonales à l'arête considérée. Pour un élément rectangulaire de dimensions $l_0 \times h_0$, les arêtes i et j sont considérées avec $(i, j) \in \{l_0, h_0\}, i \neq j$. Pour un élément hexaédrique de dimensions $x_0 \times y_0 \times z_0$, les arêtes i sont considérées avec $i \in \{x_0, y_0, z_0\}$. A noter que ces forces ne sont valides que pour des petites déformations.
4. [Jarrousse et al. \(2010\)](#) introduisent un terme hydrostatique dans la formulation de l'énergie du système, dépendant du tenseur de déformation \mathbb{F} . Ils disent que ce terme est employé pour le calcul de la force de préservation de volume (mais sans plus de précision). Ils annoncent une variation de volume de 0,1 % sans aucun détail.
5. [Kot and Nagahashi \(2015, 2017\)](#); [Kot et al. \(2014\)](#) considèrent un système masses-ressorts basés sur un maillage cubique. Les forces appliquées aux sommets sont définies comme la somme des forces d'interactions (force usuelle engendrée par un ressort) et des forces de dispersion avec $F^j = F_{\mu}^j + F_{*}^j$. Ils assurent pouvoir simuler le comportement d'un matériau avec n'importe quel coefficient de Poisson.
6. Afin de contrôler le volume de l'objet simulé durant sa déformation, [Arnab and Raja \(2008\)](#) ajoutent dans leur système masses-ressorts, des forces qui sont appliquées aux sommets du maillage triangulaire. Ces forces sont perpendiculaires à la surface du triangle auquel appartient le sommet j considéré. Elles sont définies en fonction de la raideur k_{tri}^B des ressorts (établie par rapport au module de Bulk B du matériau) et de la variation de volume de l'objet. Ces forces sont modulées par un coefficient α .

Nos travaux de recherche vont dans la continuité de ceux de [Baudet et al. \(2009\)](#) en ajoutant dans le système masses-ressorts des **forces de correction** qui permettent de contrôler la variation de volume de l'objet simulé durant sa déformation. L'ajout de ces forces, combiné à l'emploi de notre formulation analytique des raideurs, va permettre la simulation de tout matériau quelque soit leur module de Young et coefficient de Poisson. Par rapport aux méthodes présentées, notre souhait était d'obtenir une formulation qui soit valide pour tout coefficient de Poisson, et simple à mettre en place dans la simulation.

7.3 Proposition d'un système hybride

Rappelons que notre objectif est de simuler avec précision la déformation de matériaux en employant le modèle physique des masses-ressorts. Par contre, nous avons vu dans le chapitre 3 que (i) ce modèle physique ne peut naturellement simuler que des matériaux élastiques isotropes ayant un coefficient de Poisson égal à $1/4$ de part les ressorts linéaires qui le constituent dans le cadre d'une discrétisation cubique de l'objet ; (ii) que ce modèle devient instable en compression quand nous dépassons une certaine déformation. Ainsi pour simuler le comportement de tissus mous non-linéaires avec ce modèle physique, nous devons dépasser ces deux limites, tout en conservant un modèle de complexité moindre que l'emploi de la méthode des éléments finis résolvant les équations issues de la MMC.

Repousser la limite d'instabilité. Dans un premier temps, nous devons trouver une solution permettant de repousser la limite d'instabilité du système masses-ressorts apparaissant lors de la compression du système. Une solution consiste à ajouter des ressorts au système. Nous avons ainsi choisi de rajouter deux ressorts positionnés sur les diagonales des faces de chacun des éléments pour un système basé sur une discrétisation cubique.

Comme illustré par la Fig. 7.1, un élément cubique de notre système masses-ressorts est ainsi composé de 12 ressorts positionnés sur les arêtes (de raideur k_e et de longueur l_e), de 4 ressorts positionnés sur les diagonales internes de l'élément (de raideur k_c et de longueur l_c) et de 12 ressorts positionnés sur les diagonales de chacune des faces de l'élément (de raideur k_f et de longueur l_f). D'un point de vue géométrique, nous obtenons les relations suivantes entre les longueurs des différents types de ressorts :

$$l_e = \frac{l_f}{\sqrt{2}} = \frac{l_c}{\sqrt{3}}. \quad (7.2)$$

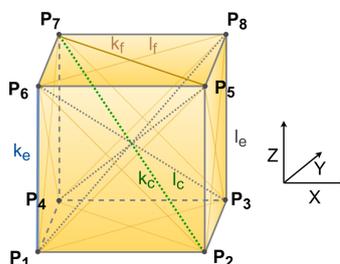


Figure 7.1 – Un élément cubique de notre système masses-ressorts composé de 8 particules, 12 ressorts sur les arêtes, 12 ressorts sur les diagonales des faces et 4 ressorts sur les diagonales internes de l'élément.

Au final, le matériau est représenté par un treillis régulier composé d'éléments cubiques identiques à l'intérieur desquels les ressorts établissent des connexions avec les masses comme présentés ci-dessus. Toute masse est ainsi centre de symétrie du système. A noter que sans force extérieure, le système se met dans une configuration stable dite de repos.

Simuler des matériaux de ν quelconque. Nous devons également trouver une solution pour échapper à la limitation de Cauchy induisant qu'un système masses-ressorts cubique ne puisse simuler que le comportement d'un matériau élastique isotrope avec $\nu = 1/4$. Pour cela, nous introduisons de nouvelles forces appelées forces de corrections, qui seront appliquées sur les particules du système à chaque pas de temps de la simulation en complément des forces linéaires des ressorts.

- Ainsi le comportement mécanique du système n'est plus uniquement induit par les forces linéaires des ressorts, mais prend également en compte ces nouvelles forces.
- De plus, pour assurer un comportement correct d'un point de vue volumique, tout en préservant la forme de la déformation, nous verrons que ces forces de corrections ont été établies de façon à contrôler la variation de volume du matériau lors de sa déformation. Elles prennent ainsi en considération les propriétés de compression du matériau définies par son coefficient de Poisson.
- Au final, nous obtenons des forces quadratiques permettant d'augmenter le degré des forces du système et ainsi simuler le comportement de tout matériau défini par un coefficient de Poisson $\nu \in [0; 0.5[$ et un module de Young E .

Établissement du système. Pour définir les paramètres de notre système masses-ressorts hybride (HybMSS), nous considérons son énergie globale. Elle correspond à la somme entre l'énergie induite par la déformation des ressorts et l'énergie résultante de la variation de volume des éléments avec :

$$W_{HybMSS} = W_{springs} + W_{vol}.$$

La dérivée de ces énergies va ainsi permettre de définir :

- les forces linéaires appliquées par les ressorts sur chacune des particules avec une formulation des raideurs des ressorts en fonction des paramètres du matériau ;
- les forces de correction non-linéaires appliquées sur les particules en fonction de la variation de volume du matériau.

Pour établir ces formulations, nous allons considérer un seul élément cubique de notre système en supposant que cet élément se trouve au milieu d'un treillis composé de mêmes éléments cubiques. L'assemblage de ces éléments permet ainsi l'établissement du système hybride complet relatif à l'objet que nous souhaitons simulé qui est ainsi basé sur une discrétisation cubique de sa représentation géométrique.

Pour créer cet assemblage, nous devons :

- Répartir correctement la densité volumique de l'objet sur les différentes masses issues de la discrétisation de l'objet. Si nous considérons un objet ayant une masse volumique $\rho = m/V_0$ avec m sa masse globale et V_0 son volume initial, la masse m_i de chacune des particules P_i du système sera alors définie par :

$$m_i = \sum_{j | P_i \in E_j} \frac{\rho}{8} V_{E_j}^c,$$

où E_j représente l'ensemble des éléments cubiques de volume $V_{E_j}^c$ contenant P_i .

- Définir les raideurs de chacun des ressorts du système (*cf.* Fig 7.1). Rappelons que les paramètres de notre élément cubique de référence seront établis en supposant qu'il se trouve entouré d'éléments adjacents. C'est-à-dire que nous supposons que chacune des arêtes sur lesquelles sont positionnés les ressorts de paramètres (l_e, k_e) est commune à 4 éléments cubiques adjacents, et que chacune des diagonales des faces sur lesquelles sont positionnés les ressorts de paramètres (l_f, k_f) est commune à 2 éléments adjacents. Ainsi de manière générale, les ressorts supportés par les arêtes

communes à n éléments adjacents sont de raideurs $k_e * n/4$, et si nous considérons une face d'un élément cubique qui n'est adjacente à aucune autre face, les raideurs des ressorts positionnés sur ses diagonales sont égales à $k_f/2$ (selon les lois de Kirchhoff).

- Puis au cours de la simulation, les forces appliquées sur chacune des particules du système hybride complet seront calculées en additionnant les forces des ressorts ainsi que les forces de correction calculées en considérant chacun des éléments du système.

7.4 Formulation des constantes de raideur des ressorts

Regardons comment nous avons établi les formulations des raideurs k_e, k_c, k_f des trois types de ressorts présents dans notre système à partir de l'énergie W_{spring} .

Principe général. Considérons un élément cubique d'un système masses-ressorts ne contenant qu'un seul type de ressort défini par une raideur k et une longueur au repos l . Introduisons les vecteurs \mathbf{r} et \mathbf{r}' représentant les ressorts dans leurs états au repos et déformé. Ainsi, un ressort de longueur l placé horizontalement a comme représentations sous forme de vecteurs : $\mathbf{r} = (l, 0, 0)^T$ et $\mathbf{r}' = (l', 0, 0)^T$ avec $l = \sqrt{\mathbf{r}^2}$ et $l' = \sqrt{\mathbf{r}'^2}$.

Appliquons ensuite une déformation homogène au treillis sur le lequel est basé le système masses-ressorts. Le point \mathbf{X} du système est alors déplacé en $\mathbf{x} = \mathbf{X} + \mathbf{u}$. Son déplacement est ainsi défini par $u_i = x_i - X_i = \partial u_i / \partial X_j X_j$ avec $i, j = x, y, z$ en considérant que le gradient du déplacement $\partial u_i / \partial X_j = \nabla u_{ij}$ est petit.

Puis sous l'hypothèse que toute masse du système est centre de symétrie des ressorts qui lui sont connectés, tout ressort est relié à des ressorts identiques ayant la même représentation \mathbf{r} alignée sur l'ensemble du treillis. Ce vecteur \mathbf{r} subit également un déplacement $\nabla \mathbf{u} \cdot \mathbf{r}$ et sa longueur varie de $\delta = l' - l = \sqrt{(\mathbf{r} + \nabla \mathbf{u} \cdot \mathbf{r})^2} - l$.

En considérant une approximation du second ordre en $\nabla \mathbf{u}$, nous obtenons :

$$\delta = \frac{1}{l} \nabla u_{ij} r_i r_j + \frac{1}{2l} \nabla u_{ki} \nabla u_{kj} r_i r_j - \frac{1}{2l^3} \nabla u_{ij} \nabla u_{kl} r_i r_j r_k r_l. \quad (7.3)$$

En considérant cette approximation, l'énergie élastique d'un ressort définie dans l'équation (3.1) s'écrit sous la forme :

$$W_{spring}(\delta) = \frac{1}{2} \nabla u_{ij} \nabla u_{kl} \frac{k}{l^2} r_i r_j r_k r_l. \quad (7.4)$$

En utilisant le tenseur de déformation linéarisé de Green-Lagrange, il faut garder à l'esprit que la densité d'énergie élastique d'un élément du système définie à l'équation (3.39), est obtenue en sommant tous les ressorts α de l'élément avec :

$$C_{ijkl} = \frac{1}{V} \sum_{\alpha \in \text{élément}} \frac{k}{l^2} r_{\alpha i} r_{\alpha j} r_{\alpha k} r_{\alpha l}, \quad (7.5)$$

où $r_{\alpha i}$ représente la $i^{\text{ème}}$ composante du vecteur \mathbf{r}_α . Notons que nous effectuons la sommation sur les indices latins répétés de i à l , alors que la somme sur les indices grecs α identifiant les ressorts est explicitement indiquée.

Cas d'un élément cubique de notre système. Considérons un élément cubique de notre système masses-ressorts se trouvant au milieu du treillis. Il est composé de 12 ressorts (k_e, l_e) sur les arêtes, 12 ressorts (k_f, l_f) sur les diagonales des faces et 4 ressorts (k_f, l_f) sur les diagonales internes de l'élément (voir la Fig. 7.1). Établissons la représentation de ces différents ressorts.

- Pour les ressorts positionnés sur les arêtes. Quatre d'entre eux sont à chaque fois parallèles à la même direction et chaque ressort est commun à 4 éléments adjacents, résultant en un seul ressort (k_e, l_e) à comptabiliser dans chacune des directions. Nous pouvons ainsi les représenter par 3 vecteurs de longueur identique égale à l_e :

$$\mathbf{r}_1 = (l_e, 0, 0)^T, \mathbf{r}_2 = (0, l_e, 0)^T, \mathbf{r}_3 = (0, 0, l_e)^T.$$

- Les quatre ressorts positionnés sur les diagonales internes à l'élément sont représentés par les vecteurs suivants de longueur égale à $l_c = \sqrt{3} l_e$:

$$\mathbf{r}_4 = (l_e, l_e, l_e)^T, \mathbf{r}_5 = (l_e, l_e, -l_e)^T, \mathbf{r}_6 = (l_e, -l_e, l_e)^T, \mathbf{r}_7 = (-l_e, l_e, l_e)^T.$$

- Pour les ressorts des diagonales des faces. Il y a deux ressorts identiques sur chacune des deux faces opposées d'un élément, mais comme chaque face est commune à deux éléments adjacents, nous comptons une seule fois chacun des vecteurs suivants de longueur identique $l_f = \sqrt{2} l_e$:

$$\begin{aligned} \mathbf{r}_8 &= (l_e, l_e, 0)^T, \mathbf{r}_9 = (l_e, -l_e, 0)^T, \mathbf{r}_{10} = (0, l_e, l_e)^T, \\ \mathbf{r}_{11} &= (0, l_e, -l_e)^T, \mathbf{r}_{12} = (l_e, 0, l_e)^T, \mathbf{r}_{13} = (l_e, 0, -l_e)^T. \end{aligned}$$

Sous ces hypothèses, l'équation (7.5) prend la forme suivante en considérant un seul élément de notre système :

$$\begin{aligned} C_{ijkl} &= l_e^{-3} \left\{ \sum_{\alpha=1}^3 l_e^{-2} k_e r_{\alpha i} r_{\alpha j} r_{\alpha k} r_{\alpha l} + \sum_{\alpha=4}^7 l_c^{-2} k_c r_{\alpha i} r_{\alpha j} r_{\alpha k} r_{\alpha l} + \sum_{\alpha=8}^{13} l_f^{-2} k_f r_{\alpha i} r_{\alpha j} r_{\alpha k} r_{\alpha l} \right\} \\ &= l_e^{-5} \left\{ k_e \sum_{\alpha=1}^3 r_{\alpha i} r_{\alpha j} r_{\alpha k} r_{\alpha l} + \frac{1}{3} k_c \sum_{\alpha=4}^7 r_{\alpha i} r_{\alpha j} r_{\alpha k} r_{\alpha l} + \frac{1}{2} k_f \sum_{\alpha=8}^{13} r_{\alpha i} r_{\alpha j} r_{\alpha k} r_{\alpha l} \right\}. \end{aligned}$$

De cette équation, nous pouvons en déduire directement :

$$\begin{aligned} C_{xxxx} &= l_e^{-5} \left\{ k_e \sum_{\alpha=1}^3 (r_{\alpha x})^4 + \frac{1}{3} k_c \sum_{\alpha=4}^7 (r_{\alpha x})^4 + \frac{1}{2} k_f \sum_{\alpha=8}^{13} (r_{\alpha x})^4 \right\} = l_e^{-1} \left(k_e + \frac{4}{3} k_c + 2 k_f \right), \\ C_{xxyy} &= l_e^{-5} \left\{ k_e \sum_{\alpha=1}^3 (r_{\alpha x} r_{\alpha y})^2 + \frac{1}{3} k_c \sum_{\alpha=4}^7 (r_{\alpha x} r_{\alpha y})^2 + \frac{1}{2} k_f \sum_{\alpha=8}^{13} (r_{\alpha x} r_{\alpha y})^2 \right\} = l_e^{-1} \left(\frac{4}{3} k_c + k_f \right). \end{aligned}$$

Nous avons ensuite à satisfaire l'équation (3.45) ainsi que la relation de Cauchy établissant le lien entre les coefficients élastiques pour un matériau isotrope modélisé par un système masses-ressorts cubique. Nous avons ainsi établi les relations suivantes

$$C_{xxxx} = C_{xxyy} + 2 C_{xyxy}, \quad C_{xxyy} = C_{xyxy} = \lambda = \mu$$

amenant dans le cadre de notre modèle aux relations

$$k_e = \frac{8}{3} k_c + k_f; \quad l_e^{-1} \left(\frac{4}{3} k_c + k_f \right) = \lambda = \mu. \quad (7.6)$$

Il nous manque ainsi une relation permettant le calcul de chacune des constantes de raideurs k_e, k_f, k_c . Nous introduisons alors un degré de liberté supplémentaire avec $\mathbf{A} = k_f/k_c$. Nous obtenons alors les formulations suivantes pour les raideurs définies en considérant $\nu = 1/4$ de part la limitation de Cauchy (impliquant que $\lambda = \mu = G = 2/5 E$) :

$$\begin{cases} k_c = \frac{3 E l_c}{\sqrt{3} (10 + 7.5 \mathbf{A})} = \frac{3 E l_e}{10 + 7.5 \mathbf{A}} = \frac{6}{5} \frac{E l_e}{4 + 3\mathbf{A}} = \frac{3 G l_e}{4 + 3\mathbf{A}} \\ k_f = \mathbf{A} k_c \\ k_e = k_f + \frac{8}{3} k_c = G l_e \frac{8 + 3\mathbf{A}}{4 + 3\mathbf{A}} = \frac{2 E l_e}{5} \frac{8 + 3\mathbf{A}}{4 + 3\mathbf{A}} \end{cases} \quad (7.7)$$

Observons ce degré de liberté \mathbf{A} établi par rapport à la géométrie du système :

- Supposer que $\mathbf{A} = 0$ signifie que le terme $k_f = 0$, c'est-à-dire que nous ne considérons pas les ressorts placés sur les diagonales des faces.
- Supposer que $\mathbf{A} = \infty$ signifie que k_c tend vers 0, impliquant :

$$\frac{k_f}{k_c} = \infty ; k_c = 0 ; k_f = k_e = \frac{3 E l_e}{7.5} = \frac{3 E l_f}{7.5 \sqrt{2}}. \quad (7.8)$$

Ce degré de liberté géométrique supplémentaire permet ainsi de stabiliser le mouvement critique des particules du système et repousser sa limite d'instabilité.

7.5 Formulation des forces de correction

Regardons comment nous définissons les forces de correction à partir de l'énergie W_{vol} .

Énergie issue de la variation de volume. Pour formuler ces forces de correction, nous considérons l'énergie issue de la variation de volume du matériau durant sa déformation. Pour cela, considérons un élément cubique c de notre système. Nous notons V^c son volume courant et V_0^c son volume initial. Considérons κ le coefficient du matériel lié à cette variation de volume. L'énergie de notre élément c issue de cette variation est ainsi définie par

$$W_{vol}^c = \frac{1}{2} \kappa \frac{(V^c - V_0^c)^2}{V_0^c}, \quad (7.9)$$

avec la densité d'énergie correspondante

$$\widetilde{W}_{vol}^c = \frac{W_{vol}^c}{V_0^c} = \frac{1}{2} \kappa \left(\frac{V^c - V_0^c}{V_0^c} \right)^2 \quad (7.10)$$

ainsi que sa variation

$$\delta W_{vol}^c = \kappa \frac{V^c - V_0^c}{V_0^c} \delta V^c. \quad (7.11)$$

En considérant la densité d'énergie, nous pouvons en déduire la force de correction correspondante que nous devons appliquer sur chacune des particules de notre élément pour contrôler sa variation de volume. Si nous considérons une particule P_i positionnée en \mathbf{P}_i , la force de correction que nous devons lui appliquer est alors définie par :

$$\overrightarrow{F}_{P_i} = \frac{\partial W_{vol}^c}{\partial \mathbf{P}_i} = \kappa \frac{V^c - V_0^c}{V_0^c} \frac{\partial V^c}{\partial \mathbf{P}_i}. \quad (7.12)$$

Pour calculer cette force de correction, nous avons ainsi besoin durant la simulation :

- (i) d'estimer le volume courant V^c de l'élément ainsi que son gradient $\partial V^c / \partial \mathbf{P}_i$ calculé en fonction de chaque particule P_i du système ;
- (ii) d'exprimer le coefficient κ en fonction des coefficients E et ν du matériau.

Approximation du volume courant. Reprenons l'illustration de l'élément cubique de la Fig. 7.1 composés de 8 particules notées P_i pour $i \in [1, 8]$. Nous pouvons calculer trois vecteurs $\vec{X}, \vec{Y}, \vec{Z}$ représentant la valeur moyenne des arêtes du cube dans les 3 directions cartésiennes (en considérant le centre du cube comme l'origine), soit :

$$\begin{cases} \vec{X} = \frac{1}{4}(\overrightarrow{P_1P_2} + \overrightarrow{P_4P_3} + \overrightarrow{P_6P_5} + \overrightarrow{P_7P_8}) \\ \vec{Y} = \frac{1}{4}(\overrightarrow{P_2P_3} + \overrightarrow{P_1P_4} + \overrightarrow{P_5P_8} + \overrightarrow{P_6P_7}) \\ \vec{Z} = \frac{1}{4}(\overrightarrow{P_1P_6} + \overrightarrow{P_3P_8} + \overrightarrow{P_2P_5} + \overrightarrow{P_4P_7}). \end{cases}$$

En considérant une déformation linéaire du volume, une approximation du volume courant V^c peut alors être définie en utilisant ces 3 vecteurs avec :

$$V^c = \vec{X} \cdot (\vec{Y} \times \vec{Z}) = \vec{Y} \cdot (\vec{Z} \times \vec{X}) = \vec{Z} \cdot (\vec{X} \times \vec{Y}). \quad (7.13)$$

Formulation des forces de correction. Les forces de correction appliquée en chacune des particules P_i de l'élément pour $i \in [1, 8]$ peuvent alors être exprimées. En considérant l'équation (7.12) et en calculant le gradient du volume $\partial V^c / \partial \mathbf{P}_i$ en utilisant l'équation (7.13), nous obtenons ainsi les formulations suivantes pour les forces de correction :

$$\begin{aligned} \overrightarrow{F_{P_1}} &= \frac{\kappa (V^c - V_0^c)}{4 V_0} \left(-(\vec{Y} \times \vec{Z}) - (\vec{Z} \times \vec{X}) - (\vec{X} \times \vec{Y}) \right), \\ \overrightarrow{F_{P_2}} &= \frac{\kappa (V^c - V_0^c)}{4 V_0} \left((\vec{Y} \times \vec{Z}) - (\vec{Z} \times \vec{X}) - (\vec{X} \times \vec{Y}) \right), \\ \overrightarrow{F_{P_3}} &= \frac{\kappa (V^c - V_0^c)}{4 V_0} \left((\vec{Y} \times \vec{Z}) + (\vec{Z} \times \vec{X}) - (\vec{X} \times \vec{Y}) \right), \\ \overrightarrow{F_{P_4}} &= \frac{\kappa (V^c - V_0^c)}{4 V_0} \left(-(\vec{Y} \times \vec{Z}) + (\vec{Z} \times \vec{X}) - (\vec{X} \times \vec{Y}) \right), \\ \overrightarrow{F_{P_5}} &= \frac{\kappa (V^c - V_0^c)}{4 V_0} \left((\vec{Y} \times \vec{Z}) - (\vec{Z} \times \vec{X}) + (\vec{X} \times \vec{Y}) \right), \\ \overrightarrow{F_{P_6}} &= \frac{\kappa (V^c - V_0^c)}{4 V_0} \left(-(\vec{Y} \times \vec{Z}) - (\vec{Z} \times \vec{X}) + (\vec{X} \times \vec{Y}) \right), \\ \overrightarrow{F_{P_7}} &= \frac{\kappa (V^c - V_0^c)}{4 V_0} \left(-(\vec{Y} \times \vec{Z}) + (\vec{Z} \times \vec{X}) + (\vec{X} \times \vec{Y}) \right), \\ \overrightarrow{F_{P_8}} &= \frac{\kappa (V^c - V_0^c)}{4 V_0} \left((\vec{Y} \times \vec{Z}) + (\vec{Z} \times \vec{X}) + (\vec{X} \times \vec{Y}) \right), \end{aligned}$$

induisant les relations attendues suivantes :

$$\overrightarrow{F_{P_1}} = -\overrightarrow{F_{P_8}}, \quad \overrightarrow{F_{P_2}} = -\overrightarrow{F_{P_7}}, \quad \overrightarrow{F_{P_3}} = -\overrightarrow{F_{P_6}}, \quad \overrightarrow{F_{P_4}} = -\overrightarrow{F_{P_5}}. \quad (7.14)$$

Définition du coefficient κ . Il nous reste désormais à exprimer le coefficient κ en fonction du coefficient de Poisson ν et du module de Young E du matériau. Rappelons que nous notons V_0 et V le volume initial et courant du matériau.

Tout d'abord, si nous considérons le tenseur de déformation linéarisé de Green-Lagrange (noté ϵ), nous avons vu dans le chapitre 1 que sa trace correspond à la variation de volume du matériau durant sa simulation, soit $Tr(\epsilon) = (V - V_0)/V_0$.

Ensuite en reprenant l'expression de l'énergie de notre système masses-ressorts incluant les forces de correction, nous savons que sa densité d'énergie est définie par :

$$\widetilde{W}_{HybMSS} = \frac{W_{HybMSS}}{V_0} = \frac{W_{springs}}{V_0} + \frac{W_{vol}}{V_0} = \widetilde{W}_{springs} + \widetilde{W}_{vol}.$$

- Considérons la densité d'énergie engendrée par la déformation des ressorts en observant uniquement le système masses-ressorts cubiques sans les forces de correction. Puisque nous simulons un matériau élastique isotrope, nous savons en considérant les équations (3.43) et (3.46) que sa densité d'énergie élastique est définie par :

$$\widetilde{W}_{springs} = \frac{G}{2} (Tr(\epsilon))^2 + G Tr(\epsilon^2).$$

- Considérons la densité d'énergie engendrée par la variation de volume. En considérant l'équation (7.10) et l'expression de la trace de tenseur de déformation, nous avons :

$$\widetilde{W}_{vol} = \frac{1}{2} \kappa (Tr(\epsilon))^2. \quad (7.15)$$

- Au final, la densité d'énergie de notre système hybride est donnée par :

$$\widetilde{W}_{HybMSS} = \frac{1}{2} (G + \kappa) (Tr(\epsilon))^2 + G Tr(\epsilon^2)$$

A partir de cette densité d'énergie, nous pouvons exprimer les composantes du tenseur de contrainte :

$$\sigma_{ij} = \frac{\partial \widetilde{W}_{HybMSS}}{\partial \epsilon_{ij}} = (G + \kappa) \delta_{ij} \epsilon_{kk} + 2 G \epsilon_{ij}.$$

Maintenant considérons un test de déformation uni-axial effectué sur notre matériau élastique isotrope. Nous avons vu dans le chapitre 1 que ce test est caractérisé par $\sigma_{11} \neq 0$ et $\sigma_{22} = \sigma_{33} = 0$ si nous appliquons une contrainte uniquement dans la direction x . En reprenant la formulation précédente, nous obtenons les expressions suivantes :

$$\begin{cases} \sigma_{11} &= (3G + \kappa) \epsilon_{11} + (G + \kappa)(\epsilon_{22} + \epsilon_{33}) \\ \sigma_{22} &= (3G + \kappa) \epsilon_{22} + (G + \kappa)(\epsilon_{11} + \epsilon_{33}) = 0 \\ \sigma_{33} &= (3G + \kappa) \epsilon_{33} + (G + \kappa)(\epsilon_{11} + \epsilon_{22}) = 0 \\ \sigma_{ij} &= \sigma_{ji} = 2G \epsilon_{ij}, \quad i \neq j. \end{cases} \quad (7.16)$$

Nous pouvons ainsi en déduire que $\epsilon_{22} = \epsilon_{33}$ donnant :

$$\begin{cases} \sigma_{11} &= (3G + \kappa) \epsilon_{11} + 2 (G + \kappa) \epsilon_{22} \\ \sigma_{22} &= \sigma_{33} = (G + \kappa) \epsilon_{11} + 2 (2G + \kappa) \epsilon_{22} = 0 \\ \sigma_{ij} &= \sigma_{ji} = 2G \epsilon_{ij}, \quad i \neq j. \end{cases} \quad (7.17)$$

Dans cette équation, l'expression de la composante σ_{22} permet d'obtenir la relation

$$\epsilon_{22} = -\frac{G + \kappa}{2 (2G + \kappa)} \epsilon_{11} = \epsilon_{33},$$

donnant l'expression suivante pour la composante σ_{11} :

$$\sigma_{11} = \frac{G (5 G + 3 \kappa)}{2 G + \kappa} \epsilon_{11},$$

Ensuite sachant que le coefficient de Poisson et le module de Young sont définis dans ce test uni-axial par $\nu = -\epsilon_{22}/\epsilon_{11}$ et $E = \sigma_{11}/\epsilon_{11}$, nous avons :

$$\nu = \frac{G + \kappa}{2 (2G + \kappa)}, \quad E = \frac{G (5 G + 3 \kappa)}{2 G + \kappa}$$

avec comme relations inverses

$$G = \frac{\kappa (2 \nu - 1)}{1 - 4 \nu}, \quad \kappa = \frac{G (5 G - 2E)}{(E - 3G)}$$

permettant au final d'obtenir les relations suivantes :

$$G = \frac{E}{2 (1 + \nu)}, \quad \kappa = \frac{1}{2} \frac{E (4 \nu - 1)}{(1 + \nu)(1 - 2\nu)} = K - \frac{5}{3} G. \quad (7.18)$$

Discussion. Si nous observons la formulation du coefficient κ , nous pouvons noter que :

- pour $\nu = 1/4$, $\kappa = 0$ impliquant que $\widetilde{W}_{vol} = 0$. En effet, nous n'avons dans ce cas pas besoin de rajouter au système des forces de correction puisqu'un système masses-ressorts cubique permet naturellement de reproduire le comportement d'un matériau ayant un coefficient de Poisson égal à $1/4$;
- pour $\nu = 1/2$, κ tend vers l'infini comme des forces infinies doivent être appliquées pour préserver le volume d'un matériau incompressible ;
- pour $\nu = 0$, $\kappa = -\frac{1}{2} E$. En effet, dans le cadre d'un test de compression effectué sur ce type de matériau, aucune dilatation n'est observable dans la direction transversale et la variation de volume est égale au déplacement de la face supérieure du matériau totalement compressible.

7.6 Validation de notre système hybride

Pour la validation, nous comparons les résultats de simulation obtenus en utilisant :

- le système masses-ressorts (dénomé MSS-Baudet) qui utilise les formulations des raideurs de [Baudet et al. \(2009\)](#). Ce modèle ne comporte ni force de correction, ni ressort sur les diagonales des faces.
- le système masses-ressorts (dénomé HybMSS-FD) utilisant la formulation des raideurs présentée dans la section 7.4 et ayant les forces de correction présentées dans la section 7.5, mais avec le paramètre A mis à 0 afin de simuler le fait de ne pas avoir de ressorts diagonaux sur les faces ;
- notre système masses-ressorts hybride (dénomé HybMSS) ayant des ressorts sur les diagonales des faces, les forces de correction présentées dans la section 7.5, et en utilisant la formulation des raideurs présentée dans la section 7.4 ;
- la méthode des éléments finis dénoté FEM en utilisant le logiciel LS-Dyna version 971dR4 (LSCT, Livermore, CA, USA, [Hallquist \(2006\)](#)) employant la loi élastique linéaire de Hooke et une solution implicite.

Présentation des expérimentations. La Fig. 7.2 présente les tests usuellement effectués pour vérifier la validité d'un modèle mécanique : le cisaillement, la compression, l'étirement, la torsion et la flexion.

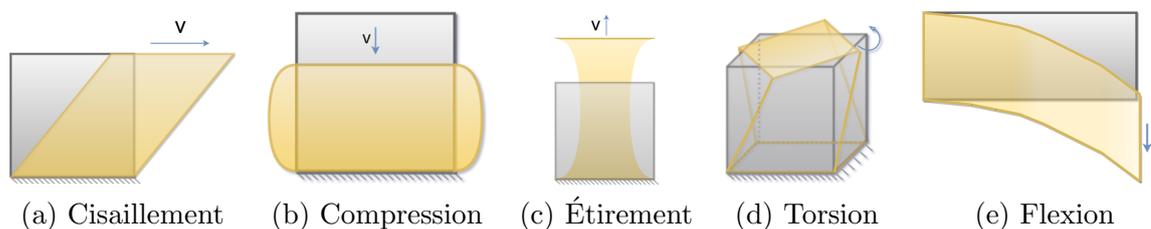


Figure 7.2 – Illustrations des tests de simulation effectués.

- Pour chacun de ces tests (excepté le test de flexion qui est étudié à l'état d'équilibre en appliquant la gravité), le pas de temps a été fixé à 0,001 s avec un nombre totale d'itérations de 10000 (*i.e.* 10 s de simulation), sans application de la force de gravité. Nous avons limité les déformations à 5% et 20%.
- Pour les forces de correction exprimées dans l'équation (7.12), nous avons employé le module de compression exprimé dans l'équation (7.18) avec $\nu \in [0; 0,5[$. Rappelons que pour $\nu = 0,25$, ces forces de correction sont nulles puisque le modèle MSS classique, *i.e.* sans force de correction, traite naturellement ce cas.
- Le test d'étirement a été effectué sur une poutre composée de $5 \times 5 \times 20$ cubes. Chacun des cubes ayant une taille de $0,05 \times 0,05 \times 0,05$ m créant ainsi un maillage de la poutre de taille $0,25 \times 0,25 \times 0,5$ m.
- Le test de flexion a été effectué sur une poutre de taille $0,4 \times 0,4 \times 2,4$ m avec deux résolutions de maillage composées de $4 \times 4 \times 24$ et $8 \times 8 \times 48$ éléments. La poutre est soumise à une force de gravité de constante $g = -10 \text{ m/s}^2$.
- Les autres tests ont été effectués sur des maillages composés de $10 \times 10 \times 10$ cubes de côté unitaire de 0,5 m engendrant une poutre de volume $V_0 = 125 \text{ m}^3$.
- Nous avons employé les mêmes tailles de maillages pour les expérimentations effectuées sous LS-Dyna.

Résultats en compression. La Table 7.3 présente les résultats en compression après 10 secondes de simulation. La colonne ϵ présente la variation relative de volume obtenue avec $\epsilon = |V_0 - V^c|/V_0$ permettant d'étudier le comportement volumique de l'objet déformé. La Fig. 7.3 permet quant à elle d'observer la forme de la poutre en compression avec des comportements différents entre les modèles.

Nous percevons l'instabilité du modèle MSS-Baudet et HybMSS-FD, alors que les modèles HybMSS et FEM présentent un gonflement régulier du volume de l'objet. Par contre avec le modèle FEM, nous pouvons tout de même observer que la surface de l'objet n'est pas lisse, alors que c'est le cas avec notre modèle HybMSS.

Si nous observons les valeurs, pour $\nu = 0,499$, c'est-à-dire pour un objet incompressible, nous obtenons une erreur de 0,01% sur le volume. A noter que la variation de volume entre les résultats obtenus avec le modèle HybMSS et FEM sont très proches.

Par ailleurs, nous observons que pour $\nu = 0$, c'est-à-dire pour un objet totalement compressible, nous obtenons bien des parois qui restent perpendiculaires aussi bien en FEM qu'avec notre modèle HybMSS.

Modèle	E	ν	Déformation	ϵ
MSS-Baudet	10000	0	5%	2,6%
HybMSS-FD				5,37%
HybMSS				5,61%
FEM				5,00%
MSS-Baudet	10000	0,25	5%	3,17%
HybMSS-FD				3,25%
HybMSS				3,17%
FEM				3,10%
MSS-Baudet	10000	0,499	5%	3,49%
HybMSS-FD				0,00%
HybMSS				0,01%
FEM				0,02%
MSS-Baudet	10000	0	20%	12,99%
HybMSS-FD				23,62%
HybMSS				19,88%
FEM				20,00%
MSS-Baudet	10000	0,25	20%	16,68%
HybMSS-FD				15,99%
HybMSS				13,52%
FEM				12,70%
MSS-Baudet	10000	0,499	20%	18,73%
HybMSS-FD				0,01%
HybMSS				0,00%
FEM				0,10%

Table 7.3 – Résultats obtenus en compression.

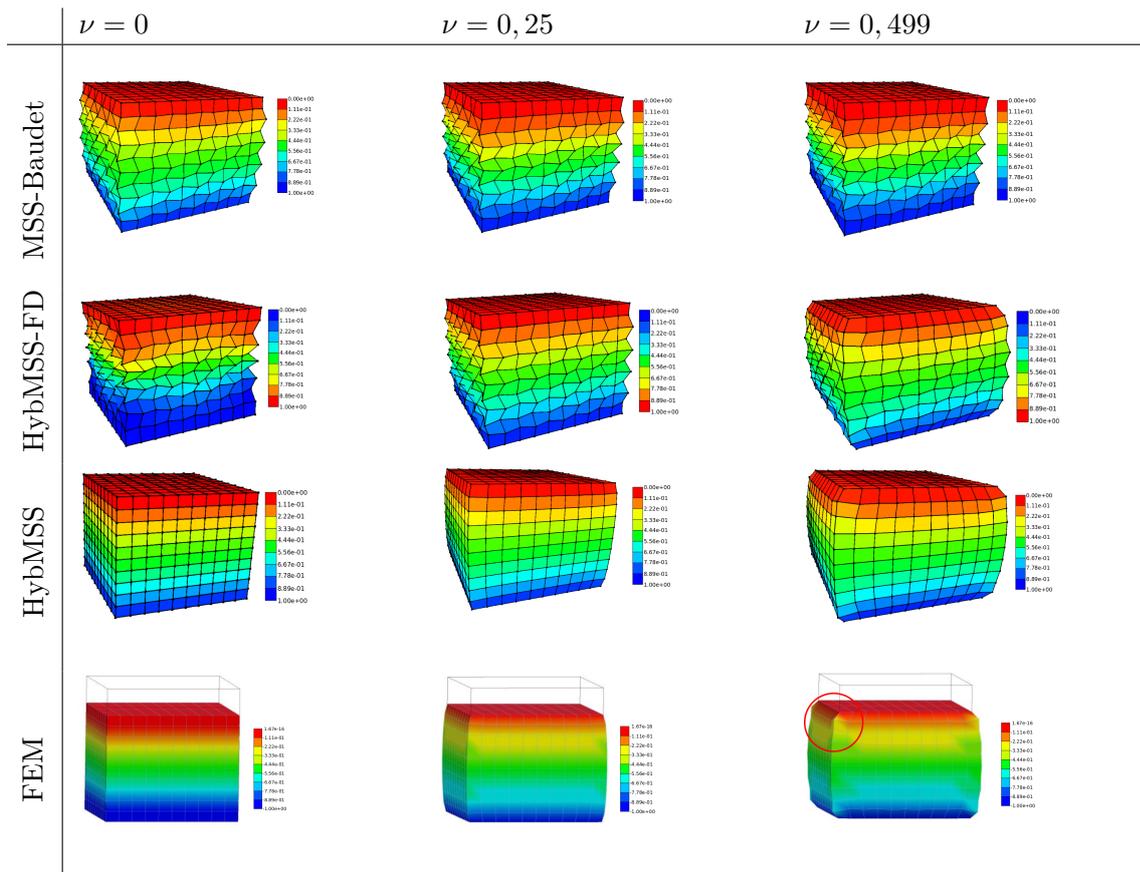


Figure 7.3 – Test en compression : MSS-Baudet, HybMSS-FD, HybMSS et FEM en grande déformation (20 %) avec différents coefficients de Poisson. L'échelle de couleur correspond au déplacement depuis la position initiale.

Résultats en étirement. La Table 7.4 présente les résultats en étirement après 10 secondes de simulation. La Fig. 7.4 permet d’observer la forme de la poutre soumise à l’étirement pour les différents modèles. Dans le cas de la poutre compressible avec $\nu = 0$, nous pouvons observer un comportement similaire entre le cas FEM et notre modèle HybMSS avec des parois qui restent horizontales à l’axe des X sans aucune courbure. Nous pouvons également observer une meilleure répartition du volume avec notre modèle HybMSS par rapport au modèle MSS-Baudet.

Modèle	E	ν	Déformation	ϵ
MSS-Baudet	10000	0	5%	0,63%
HybMSS-FD				7,46%
HybMSS				4,40%
FEM				5,00%
MSS-Baudet	10000	0,25	5%	1,14%
HybMSS-FD				0,85%
HybMSS				2,54%
FEM				2,61%
MSS-Baudet	10000	0,499	5%	1,54%
HybMSS-FD				0,00%
HybMSS				0,00%
FEM				0,00%
MSS-Baudet	10000	0	20%	0,25%
HybMSS-FD				24,92%
HybMSS				20,00%
FEM				19,99%
MSS-Baudet	10000	0,25	20%	0,73%
HybMSS-FD				3,15%
HybMSS				8,90%
FEM				10,11%
MSS-Baudet	10000	0,499	20%	1,13%
HybMSS-FD				0,00%
HybMSS				0,00%
FEM				0,00%

Table 7.4 – Résultats obtenus en étirement.

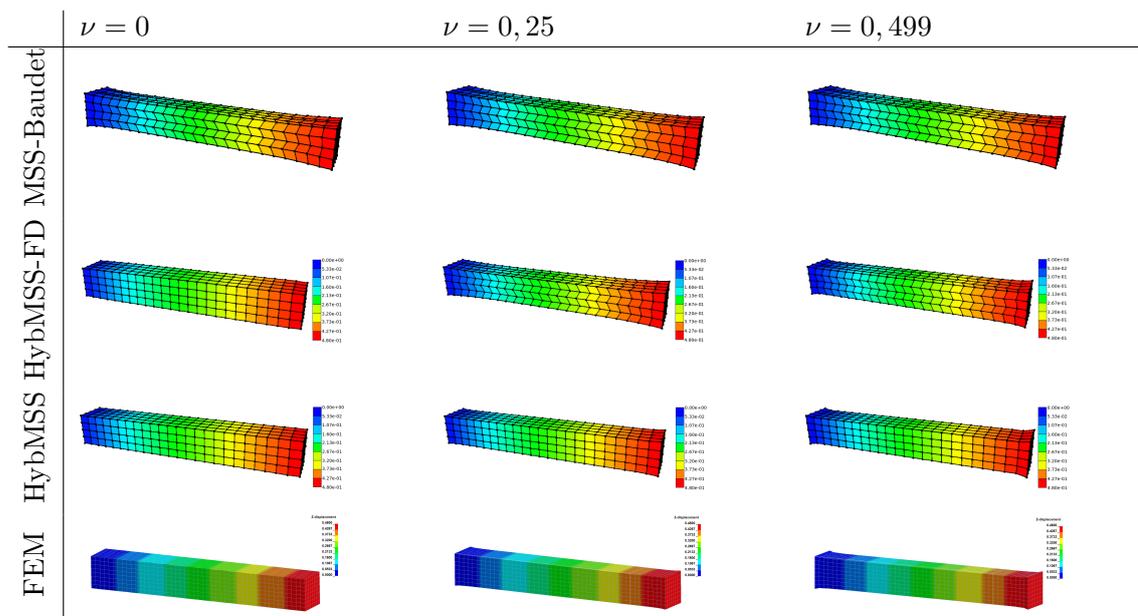


Figure 7.4 – Test en étirement en grande déformation (20 %).

Résultats en flexion. La Fig. 7.5 présente les résultats relatifs au test de la poutre en flexion (c'est-à-dire qui possède un côté fixé et qui est soumise à la gravité) pour les modèles HybMSS-FD et HybMSS (avec $E = 10$ MPa à gauche et $E = 2$ MPa à droite) en comparaison avec la solution analytique donnée par l'équation d'Euler-Bernoulli suivante :

$$y(x) = \frac{\rho g l h}{24 E I} (4Lx^3 - 6L^2x^2 - x^4), \quad (7.19)$$

où l , h , L correspondent à la largeur, hauteur et longueur de la poutre, ρ est la densité volumique de l'objet, g la constante de gravité et $I = lh^3/12$ le moment d'inertie.

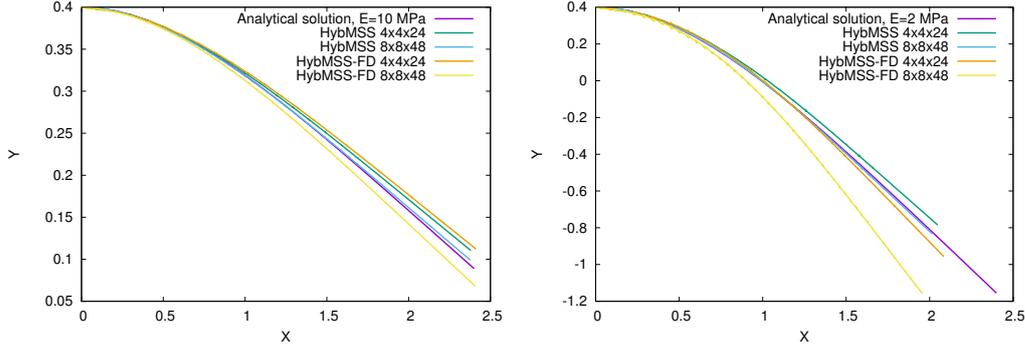


Figure 7.5 – Poutre soumise à la gravité (test de flexion) : comparaison entre les modèles HybMSS-FD, HybMSS et la solution analytique avec $E = 10$ MPa (à gauche), $E = 2$ MPa (à droite), $\nu = 0,25$, $g = -10$, $\mathbf{A} = 1$.

Nous pouvons observer que notre modèle HybMSS est très proche de la solution analytique, alors que le modèle HybMSS-FD semble moins rigide d'autant plus si nous augmentons le nombre d'éléments du maillage. Par ailleurs la Fig. 7.6 montre que le modèle HybMSS-FD (à gauche) présente des instabilités dans la zone la plus comprimée alors qu'elles n'apparaissent pas avec le modèle HybMSS (à droite) de part l'ajout des ressorts sur les diagonales des faces.

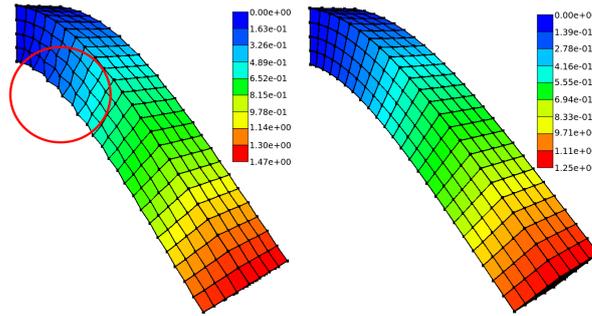


Figure 7.6 – Test de flexion : HybMSS-FD (à gauche), HybMSS (à droite). $E = 2$ MPa, $\nu = 0,25$ et $g = -10$.

Résultats en cisaillement. Les Fig. 7.7 et 7.8 présentent les résultats obtenus en cisaillement. La Fig 7.8 donne la forme obtenue en montrant une coupe transversale prise dans le plan $X - Y$ pour $Z = 5$. Nous pouvons observer une légère vague caractéristique de la déformation d'un objet visco-élastique aussi bien pour le modèle HybMSS que pour FEM. La différence de comportement entre ces deux modèles est par ailleurs mineure.

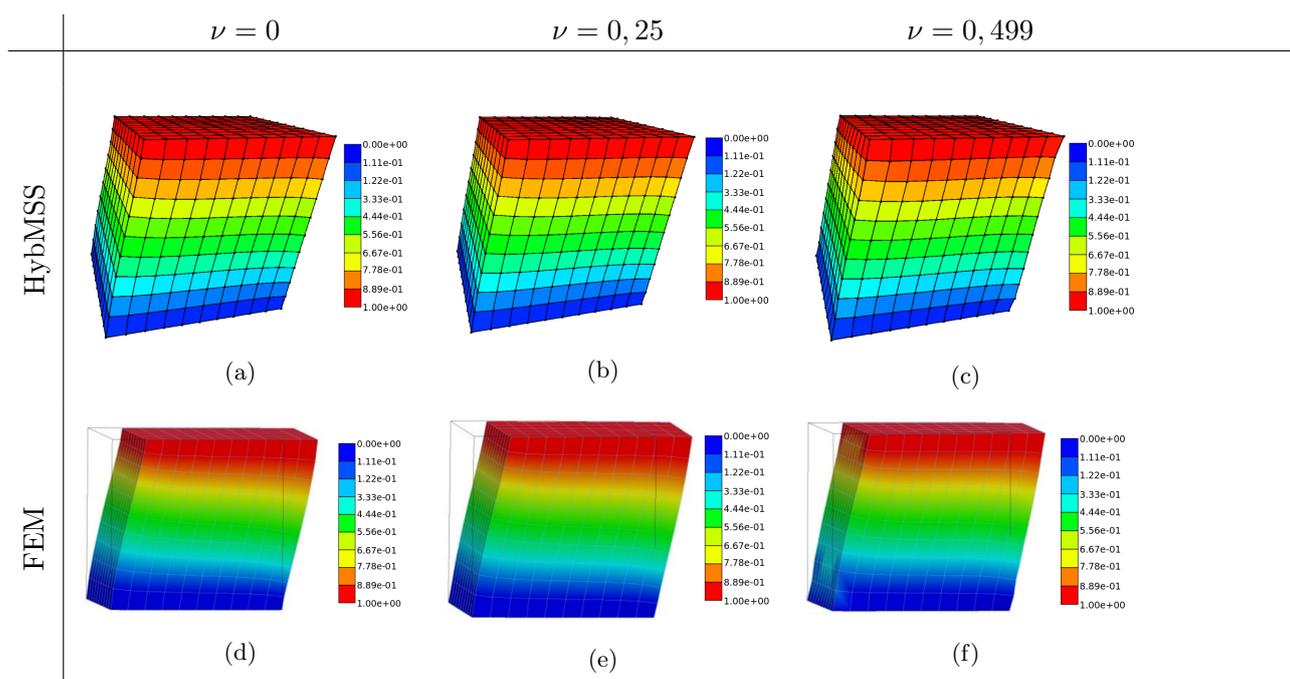


Figure 7.7 – Test de cisaillement : HybMSS et FEM en grande déformation (20%). $E = 10000$. L'échelle de couleur représente le déplacement par rapport à la position d'origine.

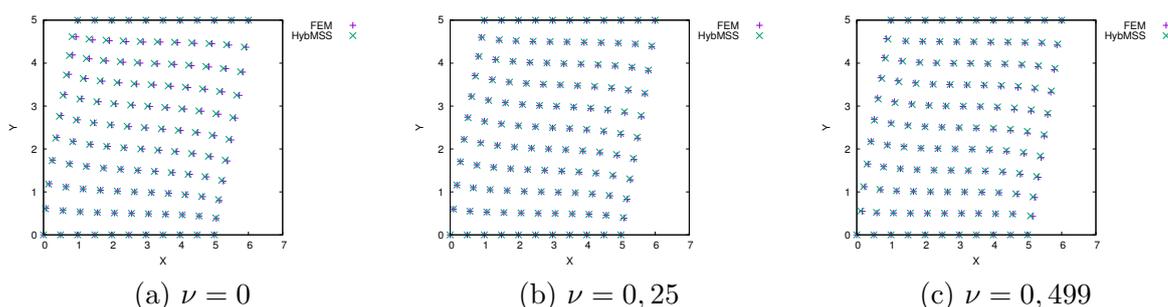


Figure 7.8 – Superposition de HybMSS et FEM en cisaillement en grande déformation (20 %). $E = 10000$.

La Table 7.5 présente les résultats obtenus en observant la partie supérieure de la poutre soumise au cisaillement dans le cadre d'un coefficient de Poisson $\nu = 0,25$. Nous comparons ainsi la force de cisaillement (en Newton) obtenue dans la simulation avec la formulation analytique de Timoshenko définie par :

$$\mathbf{F}_{shear}(x) = \frac{12 E I}{l^3} \frac{x}{(1 + \beta)}, \quad \beta = 2(1 + \nu). \quad (7.20)$$

Déformation	MSS-Baudet	HybMSS-FD	HybMSS	FEM	Analytique
5%	10985	3390	3396	3317	3571
20%	46460	14662	14600	13234	14285

Table 7.5 – Force de cisaillement de la partie supérieure en cisaillement avec $E = 10000$, $\nu = 0,25$.

Nous pouvons voir que la force engendrée par le modèle MSS-Baudet ne correspond pas à la solution théorique, alors que c'est le cas pour les modèles HybMSS et FEM.

Résultats en torsion. Les Fig. 7.9 et 7.10 présentent les résultats obtenus en torsion. La Fig. 7.10 donne la forme obtenue en considérant une coupe transversale dans le plan $X - Y$ pour $Z = 5$. Nous pouvons observer que les résultats obtenus avec notre modèle HybMSS sont très proches des résultats obtenus en FEM.

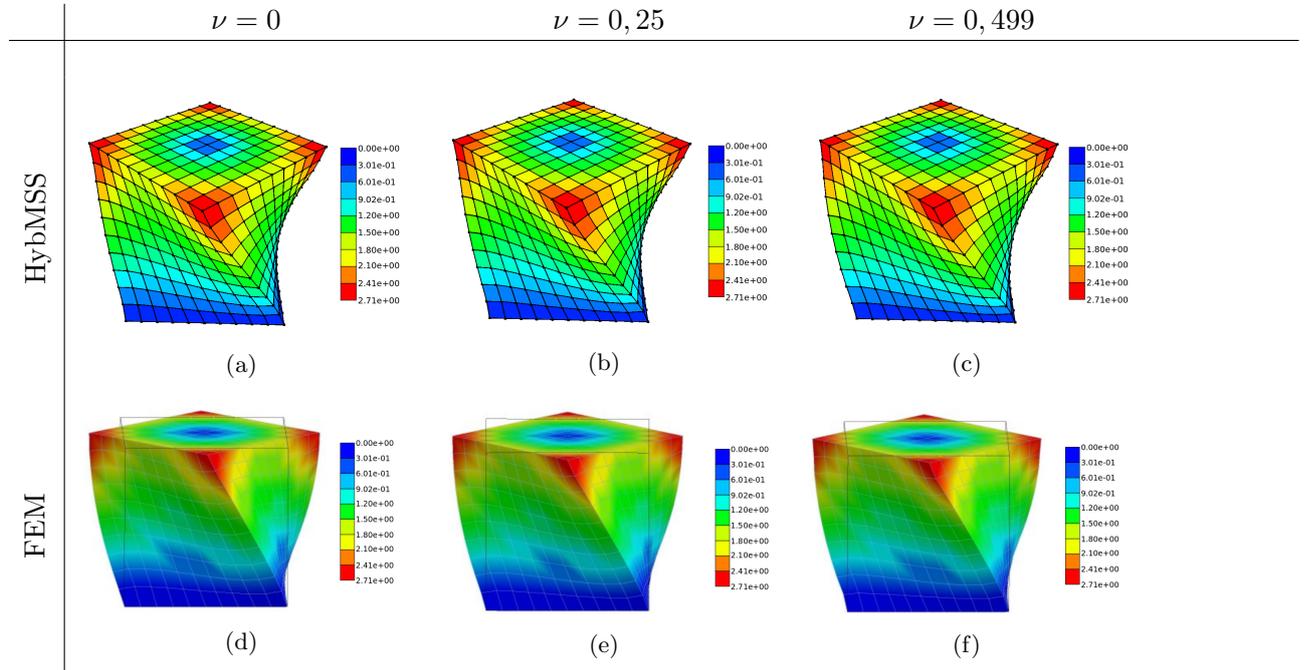


Figure 7.9 – Test de torsion : HybMSS et FEM en grande déformation (20%). $E = 10,000$. L'échelle de couleur représente le déplacement par rapport à la position d'origine.

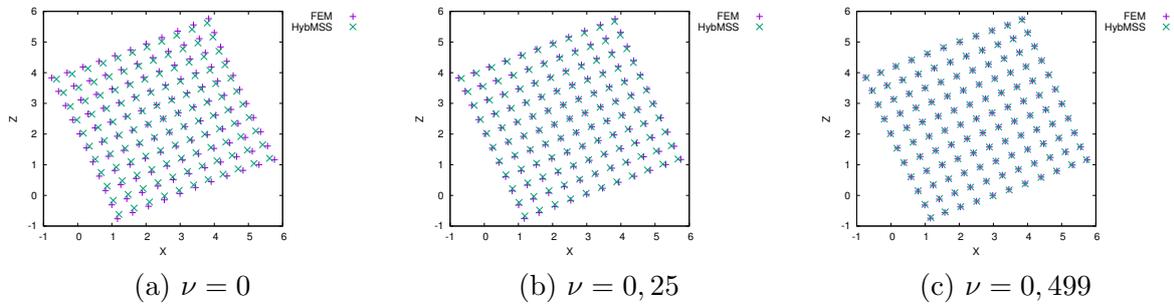


Figure 7.10 – Superposition de HybMSS et FEM en torsion en grande déformation (20 %). $E = 10000$.

La Table 7.6 présente les résultats obtenus en observant la partie supérieure de la poutre soumise à la torsion avec un coefficient de Poisson $\nu = 0,25$. Nous comparons ainsi le couple (en N·m) obtenu dans la simulation avec la formulation analytique définie par :

$$\tau(\theta) = 1.125 \frac{1}{2} \left(\frac{l}{2} \right)^3 \frac{E}{(1 + \nu)} \theta, \quad (7.21)$$

où θ représente l'angle de rotation de la partie supérieure. Les résultats des modèles HybMSS et FEM sont très proches de la solution analytique, alors que ce n'est pas le cas pour le modèle MSS-Baudet.

Déformation	MSS-Baudet	HybMSS-FD	HybMSS	FEM	Analytique
5%	3463	1277	1306	1279	1227
20%	160892	60005	60565	57981	55223

Table 7.6 – Couple de la partie supérieure obtenu en torsion pour $E = 10000$, $\nu = 0,25$.

Bilan des résultats. En résumé, nous avons vu que l'utilisation des forces de correction et l'ajout des ressorts diagonaux sur les faces jouent un rôle important dans le comportement de l'objet. Au final, les résultats obtenus avec notre modèle HybMSS sont assez proches de ceux obtenus par la méthode des éléments finis. Enfin, les forces de correction assurent un comportement adéquate du volume de l'objet déformé permettant ainsi la simulation d'objets élastiques de coefficient de Poisson quelconque.

7.7 Proposition de forces non-linéaires pour les ressorts

Nous venons de présenter l'établissement de notre système hybride, basé sur un système masses-ressorts cubiques, dans lequel des forces de correction ont été rajoutées (afin de pallier la limitation de Cauchy) et dans lequel des ressorts ont été positionnés sur les faces des éléments (afin de repousser l'instabilité constatée en compression). Par contre, la formulation des forces élastiques des ressorts n'a pour le moment pas été modifiée, alors qu'elle ne convient pas si nous souhaitons simuler le comportement non-linéaire de tissus. C'est pourquoi, nous proposons une formulation non-linéaire des forces des ressorts [Golec et al. \(2016\)](#) afin de reproduire les résultats expérimentaux obtenus par le LBMC sur des tissus réels.

Expérimentations. Les expérimentations effectuées par le LBMC ont été réalisées sur un disque de foie ou rein de porc de 2 – 3,7 mm d'épaisseur et de 15 mm de diamètre. La Fig. 7.11 illustre cette expérimentation dont tous les détails sont fournis dans l'article de [Nicolle et al. \(2012\)](#). Pour réaliser les tests caractérisant leurs propriétés mécaniques, les disques de tissus porcins ont été tout d'abord collés entre les plaques d'un rhéomètre rotatif. Puis, les tests consistent à faire subir à la plaque supérieure une rotation d'une fréquence variant de 0.1 Hz à 2 Hz alors que la plaque inférieure est fixée. Cette expérience équivaut à une déformation locale des disques porcins.

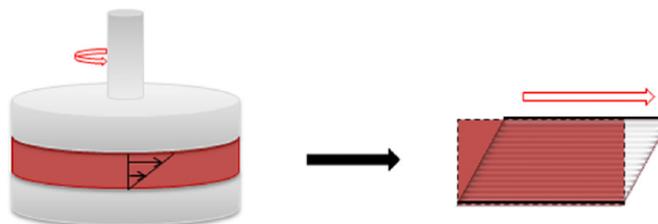


Figure 7.11 – Illustration de l'expérimentation effectuée sur des tissus de porcs.

A partir des résultats obtenus lors de ces tests, une loi de comportement non-linéaire a été mise en évidence. Cette loi, reliant la contrainte σ à la déformation $\epsilon(t)$ obtenue au temps t , est définie pour sa partie linéaire par la relation suivante ([Kelly and McGough](#)

(2009); Nicolle et al. (2010)) :

$$\sigma[\epsilon(t)] = \int_0^t G(t-t') \dot{\epsilon}(t') dt'. \quad (7.22)$$

avec $G(t)$ le coefficient de relaxation défini au temps t par

$$G(t) = \frac{K}{\Gamma(1-n)} t^{-n}$$

où dans le cadre des expérimentations effectuées sur le foie présentées dans Nicolle et al. (2012), nous avons :

- $\dot{\epsilon} \in \{0.0151 \text{ s}^{-1}, 0.133 \text{ s}^{-1}, 0.67 \text{ s}^{-1}\}$ – le taux de déformation,
- $K = 861 \text{ Pa}\cdot\text{s}^n$ – la cohérence,
- $n = 0.1138$ – l'indice constitutif linéaire,
- la fonction Γ est définie par $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$.

La table 7.7 présente les valeurs de ces paramètres pour le foie, le rein et la rate.

Organe	K	n	γ_{lim}	p
Rate	758	0.1043	0.73	1.6
Foie	861	0.1138	0.40	4.17
Rein	1818	0.0892	0.11	2.64

Table 7.7 – Paramètres relatifs aux organes internes du porc obtenus en faisant correspondre l'équation (7.22) aux résultats expérimentaux.

Formulation des forces. Afin de reproduire la loi de comportement mise en évidence par Nicolle et al. (2012), nous proposons la formulation suivante pour les forces des ressorts de notre système qui sont définis par une constante de raideur k , un coefficient d'amortissement d , une longueur initiale l_0 et une longueur courante l :

$$F = \left(d \cdot \frac{dl}{dt} + \left(k \cdot |l - l_0|^{(1-n)} \cdot \frac{l - l_0}{|l - l_0|} \right) \right) \left(1 + \alpha |l - l_0|^{p-(1-n)} \right) \quad (7.23)$$

où n, p sont des constantes, et où le paramètre α est défini par

$$\alpha = \left(\frac{1}{\gamma_{lim} \cdot l_0} \right)^{p-(1-n)}$$

avec γ_{lim} la limite de déformation établie en fonction du matériau. A noter que le comportement visqueux est assuré par le terme mis dans la première parenthèse de l'équation (7.23).

La Fig. 7.12 présente la courbe non-linéaire contrainte-déformation obtenue pour le rein pour $d\gamma/dt = 0.67 \text{ s}^{-1}$. Notre formulation permet de reproduire ses différentes parties :

- Partie I. La première pente de la courbe (partie linéaire de la déformation) est dépendante du paramètre n et plus particulièrement de l'exposant $(1-n)$ qui se trouve dans la première parenthèse de l'équation (7.23).

- Partie II. La partie non-linéaire de la courbe est dépendante du paramètre p et plus particulièrement de l'exposant $p - (1 - n)$ qui se trouve dans la seconde parenthèse de l'équation (7.23) ainsi que dans la définition du terme α .
- Limite de déformation. Le placement de la limite de déformation, à partir de laquelle la seconde partie de la courbe débute, est déterminée par la valeur de γ_{lim} .

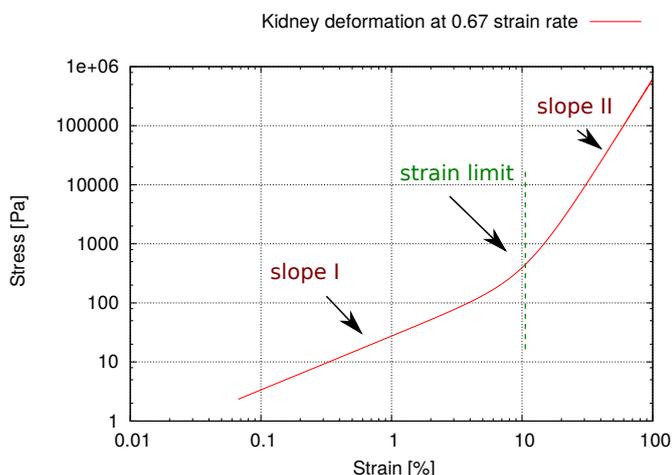


Figure 7.12 – Loi analytique établie pour le rein pour $d\gamma/dt = 0.67 \text{ s}^{-1}$. Spécification des régions de déformation linéaires et non linéaires, ainsi que la limite de déformation.

Cette approche incorpore dans un système masses-ressorts un comportement non-linéaire d'un corps élastique en modifiant uniquement la formulation des forces des ressorts, c'est-à-dire sans ajout d'éléments supplémentaires dans la structure. Il suffit ensuite de changer la valeur des paramètres p, n, γ_{lim} pour modifier le comportement de l'objet simulé. Une fois les valeurs ajustées par rapport aux données réelles, nous pouvons alors appliquer cette formulation sur n'importe quelle représentation géométrique de l'objet.

Résultats. Nous avons considéré le cas du foie qui est incompressible avec comme propriétés mécaniques $E = 1000 \text{ Pa}$, $\nu = 0.499$ et $\rho = 1000 \text{ Kg/m}^3$. Par ailleurs, la force de gravité ayant un effet négligeable dans l'expérimentation réelle, nous l'avons mise à zéro.

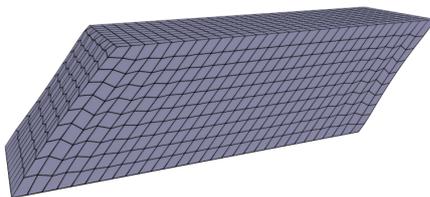


Figure 7.13 – Représentation géométrique du tissu en utilisant un maillage composé de $10 \times 10 \times 30$ hexaèdres. Au cours de la simulation, les nœuds de la partie supérieure sont déplacés, tandis que ceux de la partie inférieure sont fixés afin d'effectuer le test de cisaillement.

Au niveau géométrique, nous supposons que localement le tissu peut être représenté par un bloc rectangulaire de plaques infiniment longues. Ce bloc rectangulaire, composé de 3751 nœuds a été décomposé en $10 \times 10 \times 30$ éléments cubiques de 1 mm de côté. Puis

pour réaliser la simulation, les nœuds de la partie basse du tissu ont été fixés, tandis que ceux de la partie haute ont subi un déplacement dans la direction des X (cf. Fig. 7.13).

Cette simulation permet ainsi de reproduire le test de cisaillement en petites déformations. A noter que la déformation étant négligeable dans ce cas, nous avons observé l'évolution de la contrainte au cours du temps pour analyser nos résultats de simulation. Pour cela nous avons effectué la simulation en employant le modèle physique des masse-tenseurs et celui des masses-ressorts intégrant la nouvelle formulation des forces des ressorts. La Fig. 7.14 présente l'évolution de la contrainte σ au cours du temps lors de l'emploi de ces deux modèles en comparaison avec la loi analytique du LBMC. A noter que la contrainte a été calculée en faisant le rapport entre la force appliquée et la surface de la plaque supérieure du tissu (dans notre cas 300 mm^2). Nous observons pour les deux modèles physiques un comportement linéaire semblable en petites déformations.

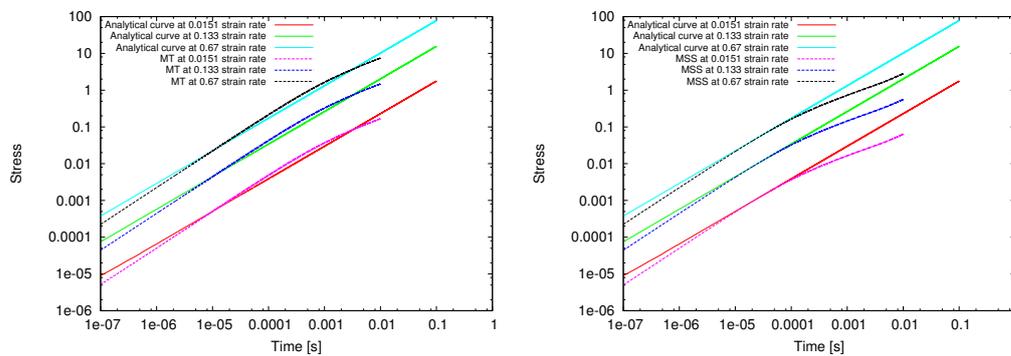


Figure 7.14 – Évolution de la contrainte en fonction du temps en considérant le modèle masse-tenseur (à gauche) et le modèle masses-ressorts (à droite) pour les taux $\dot{\epsilon} \in \{0.0151 \text{ s}^{-1}, 0.133 \text{ s}^{-1}, 0.67 \text{ s}^{-1}\}$.

Nous arrivons ainsi à reproduire le comportement des tissus issu de la loi de comportement mise en évidence par le LBMC.

7.8 Discussion

Nous avons vu dans le chapitre 3 les limitations naturelles d'un système masses-ressorts. Ainsi, de part l'emploi de ressorts qui sont linéaires, un système masses-ressorts ne peut simuler qu'un comportement linéaire. De plus, si nous prenons un système masses-ressorts basé sur une discrétisation cubique de l'objet, nous ne pouvons reproduire que le comportement d'objets ayant un coefficient de Poisson égal à $1/4$. Ces deux limitations font qu'il est difficile d'employer un système masses-ressorts pour simuler le comportement mécanique de tissus mous, puisqu'ils ont généralement un comportement mécanique non-linéaire avec des coefficients de Poisson souvent différents de $1/4$. De plus, un système masses-ressorts basé sur des ressorts positionnés uniquement sur les arêtes et sur les diagonales internes des éléments, devient vite instable dans le cas de la compression.

Pour pallier ces limitations, nous avons proposé un système masses-ressorts hybride :

- L'ajout de ressorts sur les diagonales des faces permet de repousser la limite d'instabilité du modèle en compression.
- L'ajout de forces de correction définies à partir du coefficient de Poisson de l'objet assure un comportement volumique correct du système. Il est ainsi possible de simuler

le comportement d'objets ayant un coefficient de Poisson $\nu \in [0; 0,5]$.

- Notre formulation des raideurs des ressorts intègre le module de Young du matériau simulé. Cette formulation comprend également un nouveau degré de liberté qui doit être ajusté en fonction de la géométrie du système.
- La proposition d'une formulation non-linéaire des forces des ressorts permet la simulation de comportement non-linéaire. Les paramètres de cette formulation peuvent également être ajustés en fonction du matériau simulé.

Par ailleurs, l'idée initiale était d'étendre le système masses-ressorts pour pouvoir simuler le comportement de tissus mous et ainsi proposer une alternative à la méthode des éléments finis qui est coûteuse en temps de calcul et dont l'utilisation n'est pas évidente quand nous souhaitons modifier la topologie de l'objet au cours de la simulation.

Nous avons ainsi effectué la première brique nécessaire en proposant un système masses-ressorts qui réponde à nos attentes. Rappelons que celui-ci est basé sur une discrétisation cubique de l'objet. Nous devons désormais continuer ce travail pour d'autres types de discrétisation (en hexaèdres, prismes, pyramides, etc.) afin de pouvoir employer notre système masses-ressorts quelque soit le schéma de raffinement ou dé-raffinement choisi, ou lors d'une découpe quelconque de l'objet, c'est-à-dire une découpe qui ne se situe pas forcément le long des arêtes de la représentation géométrique.

Une fois ce travail terminé (qui est actuellement en cours de développement au sein de notre plateforme TopoSIM), nous aurons tous les éléments à notre disposition pour proposer des simulations efficaces autorisant des changements topologiques en cours de simulation. En effet, rappelons que durant le travail d'Elsa Fléchon nous avons proposé une structure permettant la gestion de tout type d'élément. Il suffit ainsi de rajouter les formulations adéquates des forces des ressorts, des raideurs, et des forces de corrections établies notamment en fonction de la géométrie de l'élément.

Nous pourrons alors nous concentrer sur la problématique liée aux maillages mixtes d'un point de vue topologique mais également mécanique, afin de savoir si nous ajoutons de l'instabilité lors des changements de topologie et voir comment gérer les transitions entre différentes représentations géométrique et topologique de l'objet en cours de simulation.

Contributions

Publications

- K. Golec, J.-F. Palierne, F. Zara, S. Nicolle, G. Damiand. *Hybrid 3D Mass-Spring System for simulation of isotropic materials with any Poisson's ratio*. In *Visual Computer*, mai 2019.
- K. Golec. *Hybrid 3D Mass Spring System for Soft Tissue Simulation*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1, janvier 2018.
- K. Golec, J.-F. Palierne, F. Zara, S. Nicolle, G. Damiand. *Hybrid 3D Mass-Spring System to simulate isotropic materials*. In Work In Progress Session, VRIPHYS 2017 : 13th Workshop on Virtual Reality Interaction and Physical Simulation, 23-24 April 2017, Lyon, France.
- K. Golec, F. Zara, S. Nicolle, J.-F. Palierne, G. Damiand. *New Mass Spring System formulation to model the behavior of soft tissues*. In *22nd European Society of Biomechanics Congress (ESB 2016)*, Lyon (France), juillet 2016.

Logiciel

- **TopoSIM** - Ajout du système masses-ressorts hybride dans le framework TopoSIM incluant le calcul des raideurs des ressorts en fonction du module de Young et du coefficient de Poisson du matériau, ainsi que la configuration présentant des ressorts positionnés sur les diagonales des faces des éléments d'une discrétisation cubique. Ajout de la formulation non-linéaire des forces des ressorts.

Bibliographie

- S. Arnab and V. Raja. Chapter 4 : Simulating a deformable object using a surface mass spring system. In *2008 3rd International Conference on Geometric Modeling and Imaging*, pages 21–26, 07 2008. doi : 10.1109/GMAI.2008.24.
- V. Baudet, M. Beuve, F. Jaillet, B. Shariat, and F. Zara. Integrating Tensile Parameters in Hexahedral Mass-Spring System for Simulation. In *International Conference on Computer Graphics, Visualization and Computer Vision'2009 - WSCG'2009*, Feb. 2009. URL <http://liris.cnrs.fr/publis/?id=3673>.
- D. Bourguignon and M.-P. Cani. *Controlling Anisotropy in Mass-Spring Systems*, pages 113–123. Springer Vienna, Vienna, 2000. ISBN 978-3-7091-6344-3.
- R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.*, 21(3) :594–603, 07 2002. ISSN 0730-0301.
- Y. Chen, Q. hong Zhu, and A. Kaufman. Physically-based animation of volumetric objects. In *In Proceedings of IEEE Computer Animation '98*, pages 154–160, 1998.
- H. Delingette. Biquadratic and quadratic springs for modeling st venant kirchhoff materials. In F. Bello and P. Edwards, editors, *Biomedical Simulation*, volume 5104 of *Lecture Notes in Computer Science*, pages 40–48. Springer Berlin Heidelberg, 2008a. ISBN 978-3-540-70520-8. doi : 10.1007/978-3-540-70521-5_5. URL http://dx.doi.org/10.1007/978-3-540-70521-5_5.
- H. Delingette. Triangular springs for modeling nonlinear membranes. *Visualization and Computer Graphics, IEEE Transactions on*, 14(2) :329–341, 2008b. ISSN 1077-2626. doi : 10.1109/TVCG.2007.70431.
- A. V. Gelder. Approximate simulation of elastic membranes by triangulated spring meshes. *Journal of Graphics Tools*, 3(2) :21–41, 1998. doi : 10.1080/10867651.1998.10487490.
- K. Golec. *Hybrid 3D Mass Spring System for Soft Tissue Simulation*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1, feb 2018.
- K. Golec, F. Zara, S. Nicolle, J.-F. Palierne, and G. Damiand. New Mass Spring System formulation to model the behavior of soft tissues. In *22nd European Society of Biomechanics Congress (ESB 2016)*, Lyon, France, July 2016. URL <https://hal.archives-ouvertes.fr/hal-01320714>.

- K. Golec, J.-F. Paliarne, F. Zara, S. Nicolle, and G. Damiand. Hybrid 3D Mass-Spring System to simulate isotropic materials. Apr. 2017. URL <https://hal.archives-ouvertes.fr/hal-01504413>. Work In Progress Session, VRIPHYS 2017 : 13th Workshop on Virtual Reality Interaction and Physical Simulation, 23-24 April 2017, Lyon, France.
- K. Golec, J.-F. Paliarne, F. Zara, S. Nicolle, and G. Damiand. Hybrid 3d mass-spring system for simulation of isotropic materials with poisson's ratio. *Visual Computer*, 2019.
- J. O. Hallquist. LS-DYNA Theory Manual, 2006.
- O. Jarrousse, T. Fritz, and O. Dössel. *Implicit Time Integration in a Volumetric Mass-Spring System for Modeling Myocardial Elastomechanics*, pages 876–879. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-03882-2.
- J. F. Kelly and R. J. McGough. Fractal ladder models and power law wave equations. *The Journal of the Acoustical Society of America*, 126(4) :2072–81, Oct. 2009. ISSN 1520-8524. doi : 10.1121/1.3204304. URL <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2771060&tool=pmcentrez&rendertype=abstract>.
- M. Kot and H. Nagahashi. Second degree of freedom of elastic objects - adjustable poisson's ratio for mass spring models. In *GRAPP 2015 - Proceedings, Berlin, Germany, 11-14 March, 2015*, pages 138–142. SciTePress, 2015. ISBN 978-989-758-087-1. doi : <http://dx.doi.org/10.5220/0005303601380142>.
- M. Kot and H. Nagahashi. Mass spring models with adjustable poisson's ratio. *Vis. Comput.*, 33(3) :283–291, Mar. 2017. ISSN 0178-2789. doi : 10.1007/s00371-015-1194-8. URL <https://doi.org/10.1007/s00371-015-1194-8>.
- M. Kot, H. Nagahashi, and P. Szymczak. Elastic moduli of simple mass spring models. *The Visual Computer*, Sept. 2014. ISSN 0178-2789. doi : 10.1007/s00371-014-1015-5.
- B. A. Lloyd, G. Székely, and M. Harders. Identification of spring parameters for deformable object simulation. *IEEE Transactions on Visualization and Computer Graphics*, 13(1) : 1081–1093, 2007. ISSN 10772626. doi : 10.1109/TVCG.2007.1055.
- A. Maciel, R. Boulic, and D. Thalmann. *Deformable Tissue Parameterized by Properties of Real Biological Tissue*, pages 74–87. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-45015-3. doi : 10.1007/3-540-45015-7_8.
- W. Mollemans, F. Schutyser, J. Van Cleynenbreugel, and P. Suetens. *Tetrahedral Mass Spring Model for Fast Soft Tissue Deformation*, pages 145–154. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-45015-3.
- S. Natsupakpong and M. C. Çavuşoğlu. Determination of elasticity parameters in lumped element (mass-spring) models of deformable objects. *Graphical Models*, 72(6) :61–73, 2010. ISSN 1524-0703. doi : <http://dx.doi.org/10.1016/j.gmod.2010.10.001>.
- S. Nicolle, P. Vezin, and J.-F. Paliarne. A strain-hardening bi-power law for the non-linear behaviour of biological soft tissues. *Journal of biomechanics*, 43(5) :927–32, Mar. 2010. ISSN 1873-2380. doi : 10.1016/j.jbiomech.2009.11.002. URL <http://www.sciencedirect.com/science/article/pii/S0021929009006356>.

- S. Nicolle, L. Noguer, and J.-F. Paliarne. Shear mechanical properties of the spleen : experiment and analytical modelling. *Journal of the mechanical behavior of biomedical materials*, 9 :130–6, May 2012. ISSN 1878-0180. doi : 10.1016/j.jmbbm.2012.02.005. URL <http://www.sciencedirect.com/science/article/pii/S1751616112000549>.
- J.-F. Nicolle, S. ; Noguer, L. ; Paliarne. Shear mechanical properties of the porcine pancreas : Experiment and analytical modelling. *Journal of the mechanical behavior of biomedical materials*, 26 :90–97, 2013. URL http://ac.els-cdn.com/S1751616113002063/1-s2.0-S1751616113002063-main.pdf?_tid=165ead5c-b6c6-11e4-8cc5-00000aacb35d&acdnat=1424192442_b1fbddb342e9d7148db0820e1536054.
- D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21(4) :205–214, Aug. 1987. ISSN 0097-8930.
- M. Teschner, B. Heidelberger, M. Muller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Proceedings Computer Graphics International*, pages 312–319, 06 2004. doi : 10.1109/CGI.2004.1309227.
- J. Wilhelms and A. Van Gelder. Anatomically based modeling. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '97*, pages 173–180, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0-89791-896-7. doi : 10.1145/258734.258833.
- D. Zerbato, S. Galvan, and P. Fiorini. Calibration of mass spring models for organ simulations. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 370–375, 10 2007. doi : 10.1109/IROS.2007.4399289.

Vers la réalisation d'un simulateur d'accouchement

Je présente dans ce chapitre, les travaux qui ont été réalisés autour de la conception d'un simulateur pour l'apprentissage des gestes médicaux de l'accouchement. Ils ont été réalisés dans le cadre du travail de thèse de Romain [Buttin \(2010\)](#) (thèse en co-encadrement de 2007 à 2010 avec Behzad Shariat et Tanneguy Redarce du laboratoire Ampère et financée par le projet GMCAO du cluster ISLE de la région Rhône-Alpes) et dans le cadre du travail de thèse de Mathieu [Bailet \(2014\)](#) (thèse en co-encadrement de 2011 à 2014 avec Emmanuel Promayon du laboratoire TIMC-IMAG et financée par l'ARC 6 de la région Rhône-Alpes). Ces travaux ont ensuite continué dans le cadre du projet SAGA (Simulateurs pour l'Apprentissage des Gestes de l'Accouchement) financé par l'appel Modèles Numériques 2012 de l'ANR (Agence Nationale pour la Recherche) du 01/01/2013 au 31/07/2016.

Sommaire

8.1	Contexte de ce travail de recherche	240
8.2	Pourquoi un simulateur d'accouchement ?	242
8.3	Avant tout, il faut créer les maillages	247
8.4	Comment simuler numériquement un accouchement ?	248
8.5	Couplage avec un dispositif physique	261
8.6	Discussion	264

8.1 Contexte de ce travail de recherche

J'ai été recrutée en 2005 au sein de l'équipe SAARA du LIRIS pour apporter mes connaissances en parallélisme dans les travaux menés au sein du projet ETOILE. Ce projet vise le développement de nouvelles techniques de soins basées sur l'hadronthérapie pour traiter les tumeurs présentes dans l'appareil respiratoire. Les personnes de mon équipe impliquées dans ce projet travaillent sur la modélisation et la simulation du système respiratoire, afin de suivre au cours du traitement le mouvement de la tumeur au sein du poumon. Mais à mon arrivée, les travaux menés n'étaient pas encore assez aboutis pour envisager la phase de parallélisation des algorithmes de simulation.

Je me suis donc intéressée à un autre projet de recherche qui était en suspens faute de financement. Ce projet visait la réalisation d'un simulateur pour l'apprentissage des gestes de l'accouchement. Le laboratoire Ampère qui est l'initiateur de ce projet avec le Professeur Olivier Dupuis (actuellement obstétricien à la maternité Lyon Sud) avaient développé un premier prototype physique permettant l'entraînement aux gestes de la pose des forceps. Après la conception de ce premier prototype, ils sont venus voir les membres de mon équipe et ceux de l'équipe GMCAO du laboratoire TIMC-IMAG de Grenoble, avec l'idée de coupler ce dispositif physique à une simulation numérique, afin de dépasser les limitations du dispositif physique. L'idée était de pouvoir tester différentes morphologies (taille du bassin maternel, taille de la tête du fœtus, etc.) et de ne pas rester limité à celles imposées par les mannequins en plastique employés dans le dispositif physique.

J'ai ainsi repris ce projet à ma charge au sein de l'équipe SAARA, en m'impliquant entièrement dans la recherche de financement et dans le montage du consortium. J'ai ainsi effectué des demandes de financement auprès de la région Rhône-Alpes et de l'Agence Nationale de la Recherche, pour enfin aboutir à l'obtention avec le laboratoire Ampère d'une première bourse de thèse en 2007 (financement du doctorat de Romain Buttin) grâce au projet GMCAO du cluster ISLE de la région Rhône-Alpes, puis d'une seconde bourse de doctorat en 2011 avec le laboratoire TIMC-IMAG (financement du doctorat de Mathieu Bailet) financée par l'ARC 6 de la région Rhône-Alpes, et enfin en 2012 au financement du projet complet SAGA (Simulateurs pour l'Apprentissage des Gestes de l'Accouchement) auprès de l'appel Modèles Numériques de l'ANR. Le consortium du projet inclut des laboratoires spécialistes en modélisation biomécanique (laboratoire TIMC-IMAG, LIRIS et CAOR-ARMINES), en mécatronique (CAOR-ARMINES, laboratoire Ampère) et en didactique (Laboratoire des Sciences de l'Education de Grenoble), avec l'implication de la maternité Lyon Sud et de l'école de sages-femmes de Grenoble et de deux industriels (HRV, All4Tech) pour concevoir le prototype final.

Ce projet de recherche me tient particulièrement à cœur car au travers d'une application médicale concrète répondant à un réel besoin médical, il concentre toutes mes problématiques de recherche en modélisation et simulation physique et en parallélisation d'algorithmes. Comme toute application de simulation médicale, le pipeline complet des images médicales à la simulation numérique est présent, incluant la compréhension des mécanismes physiologiques, la modélisation pertinente du point de vue géométrique et mécanique des organes en présence, et cela avec une contrainte forte sur le temps de calcul puisque l'objectif est de coupler cette simulation à un dispositif physique. Que demander de plus pour un scientifique de notre domaine ? Et autre point intéressant dans la conception de ce simulateur, concerne la présence dès le début du projet de personnes spécialistes en apprentissage afin de concevoir un simulateur pertinent de ce point de vue. En effet, il ne s'agit pas forcément de souhaiter reproduire la réalité telle quelle est, mais de reproduire ce qui est pertinent pour l'apprentissage. L'idée est de faire comprendre le mécanisme de

ce geste pour que l'apprentis ne se contente pas de le reproduire, mais qu'il le comprenne pour en faire son propre geste et ainsi se l'approprier. Enfin, un dernier point qui me tient également à cœur, concerne le fait de travailler avec des médecins et des sages-femmes, afin de produire un outil utile d'un point de vue de la société et ainsi valoriser notre travail de recherche avec une retombée concrète dans le domaine médical.

Je vais donc au sein de ce chapitre présenter les avancées que nous avons faites sur la modélisation et la simulation de l'accouchement. Nous verrons que les travaux proposés dans l'état de l'art autour de cette thématique ont majoritairement été effectués dans le domaine de la mécanique. Ils portent généralement sur l'analyse d'un organe précis pour une compréhension fine de l'anatomie et de la physiologie du système pelvien féminin pour notamment analyser les dommages causés sur les tissus de la parturiente (terme médical pour désigner la femme enceinte) durant l'accouchement. Notre approche est tout autre, car pour notre part il ne s'agit pas de réaliser une simulation qui serait précise au niveau de chacun des organes mis en jeu (dur d'obtenir du temps réel dans ces conditions), mais de réaliser une simulation dont le comportement global est suffisamment réaliste pour que l'obstétricien ou la sage-femme perçoive des sensations tactiles proches de la réalité.

D'un point de vue chronologique, la première thèse soutenue sur cette thématique (doctorat de Romain Buttin) a permis la lancée du projet en permettant la compréhension du mécanisme complexe de l'accouchement, la récupération des données médicales nécessaires à cette réalisation, la modélisation des organes, la détermination du comportement mécanique de ces différents organes, et surtout l'objectif de ce doctorat était de vérifier qu'une telle simulation était réalisable. Cette thèse était au final plus tournée vers le domaine de la mécanique que celui de l'informatique, avec la réalisation de la première simulation de l'accouchement sous le progiciel Abaqus développé par Dassault Systèmes en employant la méthode des éléments finis. La seconde thèse soutenue sur cette thématique (doctorat de Mathieu Bailet) s'est concentrée sur le développement d'un nouveau modèle physique permettant de modéliser la tête du fœtus de manière efficace afin de simuler au mieux les déformations subies lors de l'accouchement. En effet, lors d'un accouchement un obstétricien vous dira que "si la tête passe, tout passe" c'est-à-dire que dans notre cas la modélisation de la tête du fœtus est très importante, et d'autant plus importante si nous souhaitons simuler la pose de forceps autour d'elle. Imaginez-vous en train de tirer la tête d'un futur enfant pour la faire sortir du ventre de sa mère avec deux cuillères à salade et cela sous le regard bienveillant du papa... Mieux vaut s'être entraîné avant pour savoir les manipuler et ne pas endommager ni le futur bébé, ni la maman. Tout ce travail s'est ensuite poursuivi au sein du projet SAGA.

Organisation de ce chapitre. Ce chapitre se poursuit dans la section 8.2 par une présentation succincte du déroulement d'un accouchement, suivi d'un état de l'art des simulateurs existants dans le domaine de l'obstétrique. Je terminerai cette section en positionnant les objectifs du simulateur que nous souhaitons développer au sein du projet SAGA. Puis la section 8.3 expliquera le processus effectué durant le doctorat de Romain Buttin pour créer les maillages des organes concernés par la simulation, à savoir l'abdomen maternel, le bassin osseux, l'utérus et le fœtus. Je présenterai ensuite dans la section 8.4 les résultats que nous avons obtenus durant la thèse de Romain Buttin et de Mathieu Bailet, en donnant au préalable les connaissances de base en anatomie pour comprendre ce que nous souhaitons modéliser, ainsi que les travaux qui ont déjà été réalisés sur la modélisation de ces organes. Ce chapitre rassemble ainsi les connaissances que nous avons acquises sur l'accouchement au fil des années.

8.2 Pourquoi un simulateur d'accouchement ?

L'accouchement se produit en général entre la 38^e et la 41^e semaine de grossesse. La première phase de l'accouchement, qui est appelée dilatation, démarre avec l'apparition des premières contractions utérines régulières. Ces contractions sont involontaires et permettent de rétracter les parois utérines diminuant le volume de l'utérus et générant ainsi une force de poussée permettant la descente du fœtus dans le canal pelvi-génital (aussi appelée filière pelvienne). Durant cette phase, le col de l'utérus va s'effacer et va progressivement se dilater jusqu'à environ 10 cm pour être assez ouvert pour permettre le passage de la tête du fœtus (cf. Fig. 8.1). Nous considérerons pour la simulation que cette phase est terminée, c'est-à-dire que le col est ouvert d'environ 10 cm.

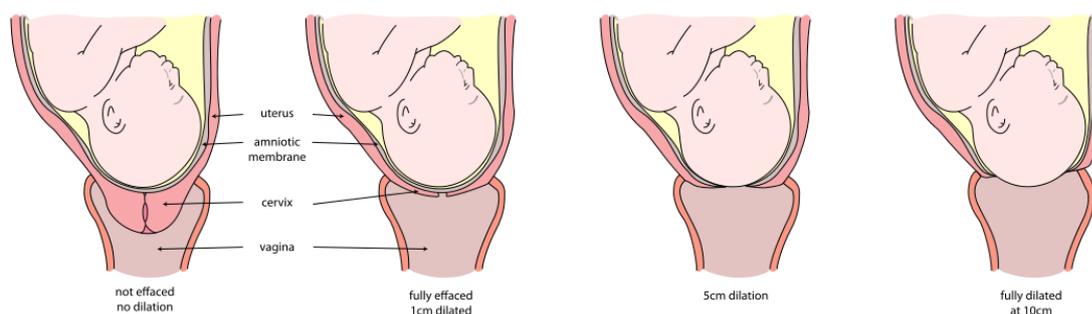


Figure 8.1 – Dilatation du col de l'utérus (Wikipedia).

La délivrance qui est la troisième phase de l'accouchement correspond à l'expulsion du placenta. Cette troisième phase débute après l'expulsion du nouveau-né et se termine en quelques minutes. Le placenta se détache de la paroi utérine et est expulsé généralement en une seule contraction utérine. Nous ne nous intéressons pas à simuler cette phase de l'accouchement pour le moment.

La phase qui nous intéresse est la seconde phase de l'accouchement qui correspond à l'expulsion du nouveau-né. Elle débute lorsque le fœtus s'engage dans le grand bassin et se termine lorsque celui-ci sera expulsé après avoir traversé le bassin mou. Elle dure entre trente minutes et deux heures. Les contractions utérines (CU), plus élevées durant cette phase vont permettre la descente du fœtus. Ces contractions sont involontaires. Elles ont une fréquence d'environ 3 ou 4 toutes les dix minutes et la durée moyenne d'une contraction au cours d'une période est de 90 secondes. L'amplitude de la contraction va varier entre le "tonus de base" (pression régnante dans l'utérus causée par sa forte déformation) et "l'intensité de la CU". L'intensité vraie est la différence entre ces deux amplitudes. Elle correspond aux efforts effectifs de poussée des CU au cours de l'accouchement.

Mais cette poussée est insuffisante pour permettre l'effacement des muscles pelviens et la sortie du fœtus. C'est pourquoi les CU doivent obligatoirement être additionnées aux poussées maternelles volontaires, afin de générer une force suffisante permettant la progression du fœtus au travers du plancher pelvien. Le rôle de la sage-femme durant cette phase est ainsi de guider la parturiente à effectuer une série de poussées abdominales importantes au même moment que les CU. En effet même si ces forces (appelée forces d'expulsion) sont environ 4 fois plus importantes, elles doivent absolument se cumuler aux CU pour permettre de franchir le seuil de résistance du plancher pelvien. Ceci est illustré sur la Fig. 8.2.

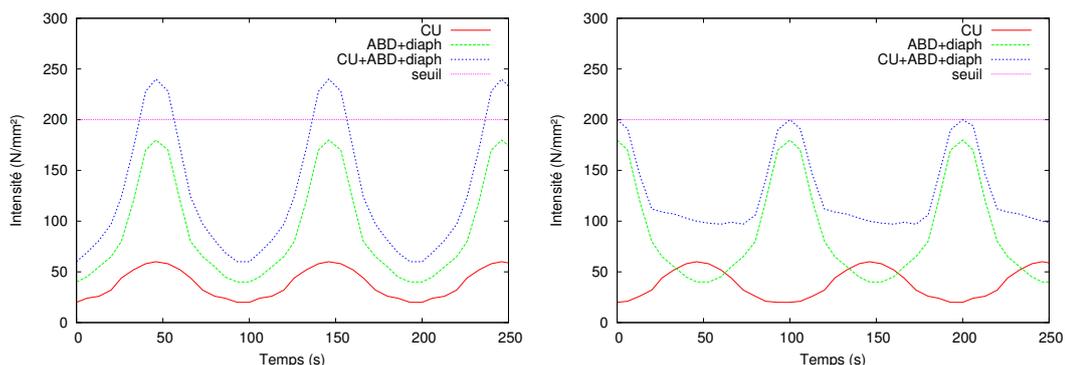


Figure 8.2 – Évolution des différentes forces utérines : forces synchronisées (gauche) et forces non synchronisées (droite). La ligne correspond au seuil permettant l'expulsion du fœtus.

Ces forces d'expulsion sont provoquées par la contraction des abdominaux et du diaphragme. Les abdominaux sont des muscles situés sur la partie antérieure de l'abdomen, mais ils se retrouvent rehaussés à cause de la présence du fœtus. Les muscles ainsi étirés viennent englober la surface utérine et exercent une pression uniforme sur la partie supérieure de l'utérus. Le diaphragme pousse également le fœtus dans le canal vaginal. Au final, la descente du fœtus est provoquée par la combinaison des forces de CU et des forces d'expulsion (forces abdominales et du diaphragme) appliquées sur l'utérus, rétrécissant les parois utérines et causant ainsi une force expulsant le fœtus dans le canal vaginal.

Cette seconde phase durant laquelle le fœtus passe le plancher pelvien pour sortir de l'abdomen maternel, ne doit pas dépasser 30 à 40 minutes afin que le fœtus ne soit pas en souffrance. Ainsi, il est parfois nécessaire d'aider l'extraction du nouveau-né à l'aide d'instruments de type forceps ou ventouses comme cela est illustré dans la Fig. 8.3.

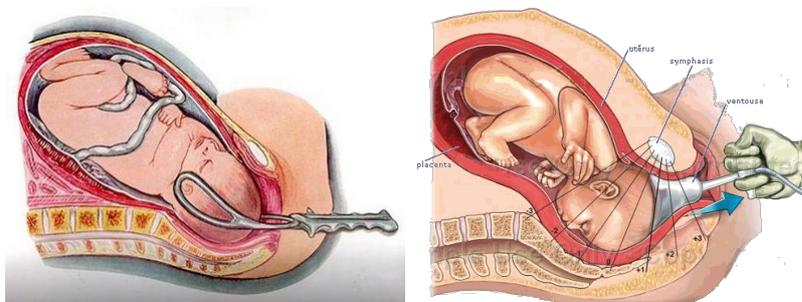


Figure 8.3 – Utilisation de forceps (gauche) ou de ventouses (droite) pour l'extraction du fœtus.

Il est alors primordial d'acquiescer le geste adéquat pour utiliser ces instruments, afin d'une part de ne pas endommager ni les tissus de la parturiente ni ceux du fœtus, et d'autre part oser faire ce geste et ne pas recourir à une césarienne quand cela peut être évité. Par ailleurs, l'apprentissage de ce geste directement auprès de la parturiente est rendu d'autant plus délicat que celle-ci est consciente et que le futur-père est généralement présent dans la salle d'accouchement.

Dans ce contexte, un certain nombre de simulateurs ont vu le jour et sont utilisés pour l'entraînement de ces gestes médicaux de l'accouchement.

Simulateurs existants. En 2007, Gardner (2007) a proposé un état de l'art sur l'utilisation de simulateurs dans le domaine de la gynécologie obstétrique. Il peut être complété par celui de la thèse de Moreau (2007) axé uniquement sur les simulateurs d'accouchement. Les simulateurs les plus employés car les moins onéreux sont les *simulateurs anatomiques* qui reproduisent les repères anatomiques utilisés en gynécologie obstétrique. Par exemple le simulateur proposé par Graves and Savannah (1951) est composé du mannequin d'une femme à l'échelle un, avec trois niveaux de gestation, d'un placenta et d'un mannequin de fœtus pouvant se positionner de différentes façons dans l'abdomen. Ensuite les *simulateurs anatomiques instrumentés* permettent d'offrir un retour haptique en plus des caractéristiques des simulateurs anatomiques. Nous pouvons citer comme simulateurs de ce type : Allen et al. (1995); Eggert et al. (2003); Knapp and Eads (1974); Riener and Burgkart (2003); Sielhorst et al. (2004). Parmi ces simulateurs, le simulateur Noelle de la société Gaumard Scientific représente un produit très complet dans la philosophie "mannequin d'apprentissage" avec la réalisation de simulations permettant l'apprentissage des réflexes à avoir dans différentes situations types. Enfin, la dernière classe de simulateurs concernent les *simulateurs basés sur la Réalité Virtuelle* avec une visualisation de la descente du fœtus dans le canal pelvien comme par exemple Boissonnat and Geiger (1993); Forster et al. (2001); Geiger (1993); Kheddar et al. (2004). Dans ce sens, Letterie (2002) a mis en évidence en 2002 l'apport bénéfique de la réalité virtuelle pour l'entraînement en gynécologie obstétrique.

Les simulateurs anatomiques (instrumentés ou non) étant basés sur des mannequins en plastique, ne permettent pas la simulation des interactions des tissus et donc n'offrent pas un rendu sensoriel réaliste. Or la précision du geste est fonction de la réaction des tissus et des structures anatomiques. C'est pourquoi, il est nécessaire de concevoir des simulateurs capables de restituer des sensations réalistes, tout en prenant en compte différents paramètres tels que les contractions utérines, les efforts de poussée abdominale, ou encore la morphologie de la parturiente ainsi que celle du fœtus.

Par rapport à ces objectifs, quatre simulateurs retiennent notre attention. Ils sont illustrés par la Fig. 8.4.

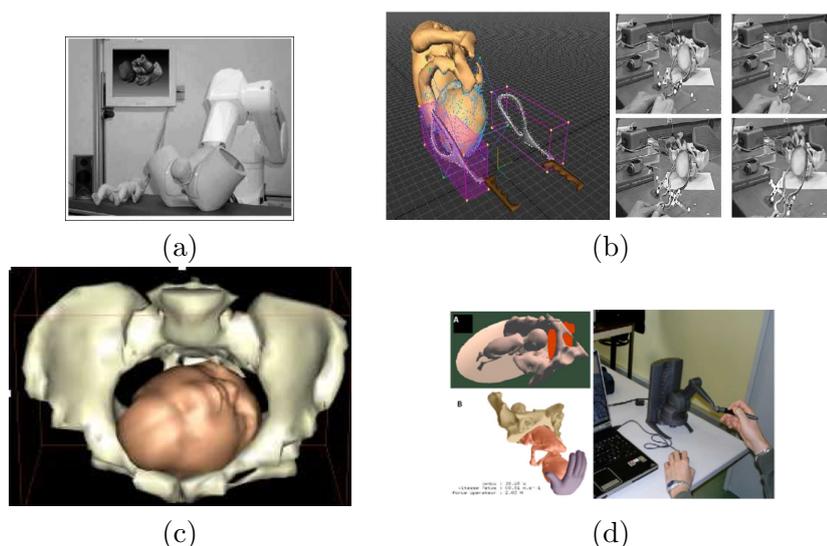


Figure 8.4 – Les simulateurs de Riener (a), Lapeer (b), Boissonnat-Geiger (c) et Kheddar (d).

- Le Geburtensimulator breveté par [Obst et al. \(2004\)](#); [Riener and Burgkart \(2003\)](#); [Sielhorst et al. \(2004\)](#) propose un dispositif de Réalité Augmentée, couplant une interface haptique à une partie logicielle. La partie logicielle simule des fonctions physiologiques (pression sanguine, rythme cardiaque, douleur, manque d'oxygène) et biomécaniques (position de la tête du fœtus calculée à partir des forces de contraction mais avec une direction imposée) de la parturiente et du fœtus. La partie physique est composée d'un mannequin du bassin, d'un modèle de tête fœtale, d'actionneurs et d'une unité de contrôle programmable. Au final, ce simulateur rend des sensations analogues à celles ressenties par l'accoucheur lors de l'utilisation de forceps. Mais l'utilisation d'un robot 6 axes pour le positionnement de la tête fœtale le rend volumineux, lourd, non transportable et d'un coût très élevé. De plus, la trajectoire suivie par la tête du fœtus dans le canal vaginal est imposée c'est-à-dire qu'elle ne tient compte ni de la morphologie du fœtus et de la parturiente, ni de l'interaction de l'utilisateur avec le système.
- Le simulateur développé par [Lapeer \(2005\)](#); [Lapeer and Prager \(2001\)](#); [Lapeer et al. \(2004\)](#) propose un dispositif de Réalité Augmentée pour l'entraînement à la pose de forceps. L'utilisation de forceps munis de capteurs permet leur visualisation en 3D lors de leur positionnement autour de la tête virtuelle du fœtus visualisée également en 3D. L'objectif est de mettre en évidence les contacts qui ont lieu entre la tête et les forceps, mais la déformation du crâne du fœtus aux points de contact n'est pas encore modélisée. C'est pourtant un aspect fondamental de la formation que d'apprendre à doser la pression exercée par les instruments sur le crâne.
- Le simulateur conçu par [Boissonnat and Geiger \(1993\)](#); [Boissonnat and Yvinec \(1995\)](#) est uniquement basé sur une partie logicielle. Il permet la réalisation d'un pronostic précis de l'accouchement en analysant différents paramètres (taille, forme du bassin maternel, forme de la tête fœtale, contractions utérines et position de la tête fœtale). Au final, ce simulateur simule le mouvement de rotation du bébé à l'intérieur du bassin maternel, mais là encore à partir de trajectoires théoriques imposées. Ce simulateur porte ainsi plus sur une phase amont de l'accouchement.
- Le simulateur développé par [Kheddar et al. \(2004\)](#) est composé de deux parties : une partie logicielle permettant la visualisation en 3D du fœtus et du bassin osseux et mou de la parturiente, et une partie haptique composée d'un dispositif à retour d'efforts trois axes et d'une main virtuelle représentant la main du praticien. Le fœtus avance dans le canal pelvien sous l'effet d'une force représentant les contractions utérines, l'effort abdominal et le frottement entre le fœtus et le bassin mou. Ce simulateur permet la réalisation du geste obstétrique de manière concrète, mais l'emploi d'un bras à retour d'effort est loin du geste initial de l'accouchement, et l'emploi de trajectoires imposées du fœtus ne permet pas de prendre en compte les interactions et la morphologie des organes, ni le geste du praticien.

Parmi ces simulateurs, aucun d'entre eux ne donne la possibilité de simuler différents cas cliniques, ni de proposer différents scénarios. Or cet aspect est primordial pour appréhender différents types de situations pour la formation.

Notre approche dans le projet SAGA. L'objectif que nous nous sommes fixés dans le cadre du projet SAGA (Simulateurs pour l'Apprentissage des Gestes de l'Accouchement) de l'appel ANR Modèles Numériques 2012, réside dans la conception d'un simulateur complet conçu à la fois pour l'usage des sages-femmes pour les aider à la prise de décision

(permettant une diminution des appels non nécessaires aux obstétriciens), et à l'usage des obstétriciens pour acquérir les bons gestes de l'accouchement (afin qu'ils osent faire ces gestes plutôt que de s'orienter vers une césarienne, acte chirurgical en constante hausse malgré les risques liés à la chirurgie et les coûts engendrés). Ce simulateur va coupler :

- un dispositif physique (mannequin anthropomorphe du bassin de la parturiente et de la tête du fœtus),
- une simulation numérique de la descente du fœtus prenant en compte la morphologie de la parturiente et du fœtus,
- et un dispositif didactique permettant la gestion de différents scénarios pertinents pour l'apprentissage.

Ce couplage va permettre la prise en compte des actions de l'utilisateur dans la partie numérique et l'intégration des efforts calculés par le modèle numérique dans le dispositif physique.

La partie physique s'appuie sur le simulateur BirthSIM illustré sur la Fig. 8.5. Il a été développé par le laboratoire Ampère, en partie durant le doctorat de [Moreau \(2007\)](#). Ce simulateur est constitué de mannequins anthropomorphes usuellement employés par les obstétriciens et sages-femmes pour leur apprentissage. L'emploi d'un actionneur pneumatique permet la descente de la tête du fœtus au travers du bassin pelvien. Des capteurs de position à six degrés de liberté situés sur les forceps et la tête fœtale permettent de visualiser les mouvements effectués. Ce prototype est en cours d'amélioration pour augmenter son nombre de degrés de liberté avec l'ajout des rotations de la tête du fœtus lors de sa descente, permettant le suivi de la trajectoire calculée par la simulation numérique.

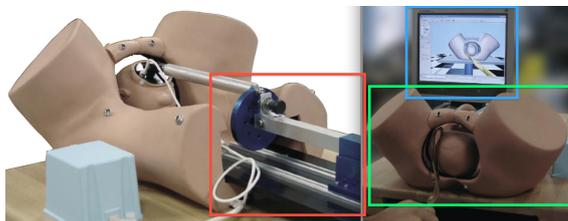


Figure 8.5 – Le simulateur physique BirthSIM développé par le laboratoire Ampère.

Mes travaux de recherche dans ce projet concernent la réalisation de la simulation biomécanique de l'accouchement. La difficulté est alors de réussir à effectuer une simulation suffisamment réaliste - permettant d'apporter les informations adéquates au dispositif physique (position de la tête fœtale, forces appliquées sur différents points stratégiques) - à une vitesse suffisamment rapide pour assurer l'interactivité du dispositif. Il s'agit ainsi de simplifier la modélisation en conservant un résultat réaliste et de coupler une simulation numérique et une interface haptique (afin de répercuter en temps interactif les actions de l'utilisateur sur la simulation). Par rapport aux travaux existants, nous souhaitons effectuer la simulation sans imposer une trajectoire au fœtus, c'est-à-dire que la descente se fait de manière naturelle par l'application des forces de contractions et des poussées maternelles.

Notons que la réalisation de cette simulation seule (sans être couplée à un dispositif haptique) a déjà un impact important dans l'apprentissage du geste, par la compréhension du rôle de chacun des organes en interaction. L'apprenti peut ainsi visualiser les mécanismes

qui ne sont normalement pas visibles. Ceci permet de mieux comprendre et analyser le rôle de chacun, ainsi que les répercussions de son geste.

8.3 Avant tout, il faut créer les maillages

Le travail de doctorat de Romain Buttin sur la simulation de l'accouchement a été effectué en collaboration avec le Pr Adamsbaum du Groupe Hospitalier Cochin-Saint Vincent de Paul de Paris (AP-HP) et le Pr Gilles Grangé de la maternité Port Royal (AP-HP). Cette collaboration a permis la récupération de données d'imageries médicales (scanners et IRM) de femmes en fin de grossesse. La précision de ces données médicales est de l'ordre de 5 mm (espace entre deux coupes), avec une précision de 2 mm pour le bassin osseux, constituant ainsi la limite de précision de la simulation numérique. Cette limite est largement suffisante dans le contexte de la réalisation d'un simulateur interactif.

Génération des maillages surfaciques. Pour les tissus mous, la segmentation et la génération des maillages surfaciques ont été réalisées par Jérémie Anquez (TELECOM ParisTech, CNRS, UMR-5141, Laboratoire Traitement et Communication de l'Information). De notre côté, nous avons réalisé le maillage surfacique du bassin osseux à partir des images scanners en employant le logiciel MRICro.

Simplification des maillages surfaciques. Les maillages générés à partir des données médicales comportent un grand nombre d'éléments que nous avons diminué, afin de limiter le temps d'exécution de la simulation numérique. Nous avons employé le logiciel ReMESH développé par [Attene and Falcidieno \(2006\)](#). Les zones ayant une petite variation angulaire entre les éléments du maillage ont ainsi été simplifiées. La Fig. 8.6 présente le maillage du fœtus (a) et du bassin osseux (b), ainsi que le nombre de nœuds présents dans les différents maillages surfaciques des organes (c) avant et après leur simplification.

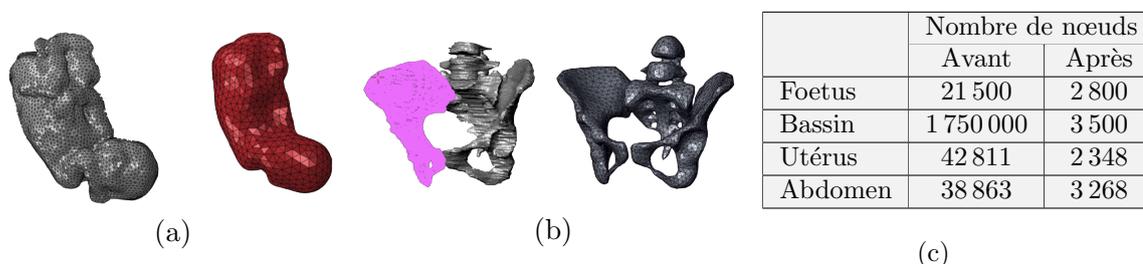


Figure 8.6 – Maillages surfaciques du fœtus (a) et du bassin osseux (b) et nombre de nœuds présents dans les maillages (c) avant et après leur simplification.

Pour cette simplification, nous avons veillé à conserver les caractéristiques principales des organes en mettant en place un critère de précision. Nous avons alors employé le logiciel MESH proposé par [Aspert et al. \(2002\)](#) permettant de calculer la distance moyenne d_{moy} entre le maillage initial et le maillage simplifié en faisant varier le taux de suppression des éléments. Pour chacun des organes, nous avons ainsi considéré une grandeur caractéristique d_{organe} pour déterminer le critère de précision α défini par :

$$\alpha = 100 \times \frac{d_{organe} - d_{moy}}{d_{organe}}.$$

Comme grandeur caractéristique, nous avons choisi l'épaisseur de l'aile iliaque pour le bassin osseux, l'épaisseur de la membrane utérine pour l'utérus, et le diamètre de la

tête fœtale pour le fœtus. La Fig. 8.7 présente l'évolution de la précision des maillages en fonction du pourcentage de simplification. Ainsi en simplifiant à 90 % les maillages, nous arrivons à conserver une précision supérieure à 85 %.

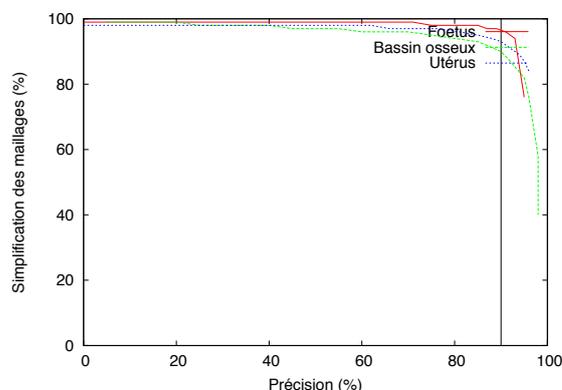


Figure 8.7 – Comparaison de l'évolution de la précision des maillages en fonction du pourcentage de simplification du maillage pour le bassin (vert), le fœtus (rouge) et l'utérus (bleu).

Cette simplification a ainsi permis de diminuer considérablement le nombre de nœuds des maillages surfaciques des organes.

Création des maillages volumiques. Les maillages volumiques ont ensuite été réalisés au sein du progiciel de simulation Abaqus à partir des maillages surfaciques. La méthode employée de création des tétraèdres est basée sur l'algorithme de Delaunay. Nous avons ainsi créé les maillages du bassin osseux, de l'abdomen maternel, du fœtus et de l'utérus.

8.4 Comment simuler numériquement un accouchement ?

Après l'obtention du modèle géométrique des organes, nous pouvons nous attaquer à la simulation de leur comportement mécanique. Je présente ainsi dans cette section les connaissances que nous avons acquises sur les principaux organes rentrant en jeu dans le déroulement d'un accouchement. Les structures anatomiques qui vont nous intéresser se situent dans la cavité pelvienne de la parturiente. Cette zone contient une partie de l'appareil digestif, l'appareil urinaire, l'appareil reproducteur (avec le vagin, l'utérus gravide et les trompes utérines et les ovaires), et bien entendu le fœtus. Comme approche de présentation de la suite, je vais présenter pour chacun des organes, les notions d'anatomie requises pour leur modélisation, les modèles déjà proposés dans la littérature ainsi que l'approche que nous avons adoptée et les résultats obtenus. Je tirerai un bilan à chaque fois pour analyser ce qui peut être encore amélioré dans l'objectif de l'obtention d'une descente du fœtus en temps réel. Les lecteurs désireux d'en connaître un peu plus sur l'anatomie et la physiologie d'un accouchement (tout en restant dans une approche pragmatique pour la réalisation de la simulation) peuvent se référer aux manuscrits de doctorat de Romain [Buttin \(2010\)](#) et de Mathieu [Bailet \(2014\)](#). Pour des informations obstétricales plus poussées, le livre de [Schaal et al. \(2007\)](#) constitue une très bonne référence en la matière.

Bassin osseux et plancher pelvien

Si nous souhaitons faire une simulation simple de l'accouchement, un premier test consiste à essayer de faire passer la tête du fœtus dans le bassin. Celui-ci est constitué de

deux parties : le bassin osseux et le bassin mou (aussi appelé plancher pelvien). Le bassin osseux est constitué de plusieurs os comme nous pouvons le voir sur la Fig. 8.8 (a) qui sont reliés entre eux par un certain nombre de ligaments visibles sur la Fig. 8.8 (c) :

- le sacrum (en rouge) qui se situe dans la partie inférieure de la colonne vertébrale,
- le coccyx (en vert) qui se trouve dans le prolongement du sacrum,
- les deux os coxaux-iliaques (en bleu) qui forment les parties latérales et antérieures du bassin. Ces deux os sont constitués de trois parties détaillées sur la Fig. 8.8 (b) : l'ischium (en bleu), l'ilium (en rouge) et le pubis (en blanc). Les ailes iliaques de ces os sont constituées d'une zone lisse et creuse appelée fosse iliaque. Cette zone va accueillir la tête du fœtus au moment de sa descente.

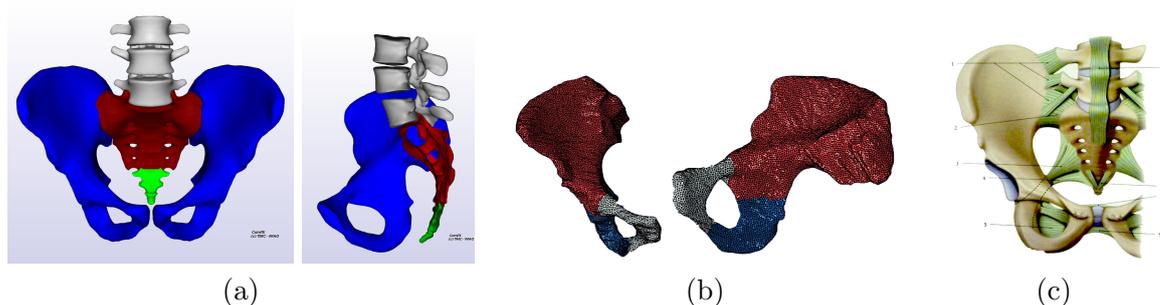


Figure 8.8 – (a) Bassin osseux constitué des os coxaux-iliaques (bleu), du sacrum (rouge) et du coccyx (vert) (Images issues de [Bailet \(2014\)](#)). (b) Les deux os coxaux-iliaques sont constitués de trois parties (Images issues de [Buttin \(2010\)](#)). (c) Articulation ligamentaire sacro-iliaque (Image issue de [Kamina \(2006\)](#)).

Le bassin mou (ou plancher pelvien) est constitué de deux muscles principaux visibles sur la Fig. 8.9 (a) : le muscle releveur de l'anus et le muscle coccygien. Ces muscles s'insèrent dans les parois latérales du bassin formant un diaphragme musculaire sur lequel reposent les organes pelviens, empêchant la descente du fœtus au cours de la grossesse. De part sa densité, il est difficile d'identifier les différents muscles de ce réseau musculaire.

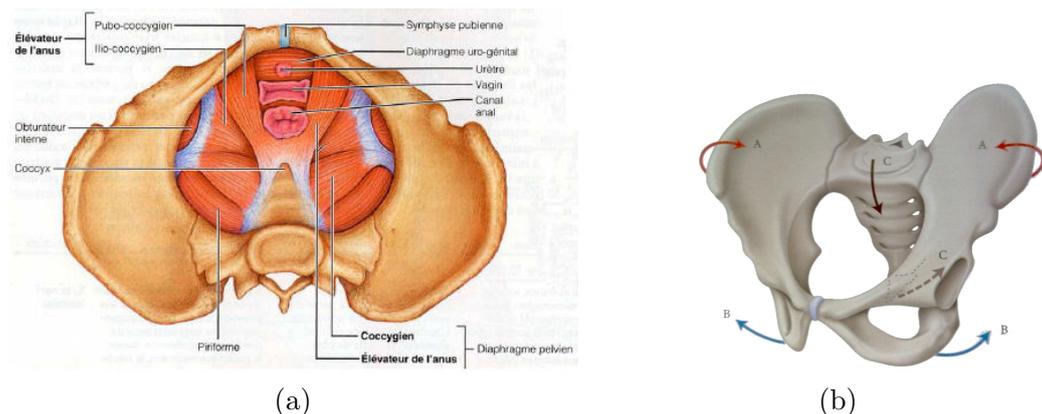


Figure 8.9 – (a) Muscles du plancher pelvien (Image issue de [Kamina \(2009\)](#)). (b) Mouvement du bassin suite à la poussée de la tête foetale sur celui-ci.

Durant la seconde phase de l'accouchement, le fœtus rentre complètement dans l'orifice d'entrée du bassin (appelé détroit supérieur), puis le traverse pour en sortir par l'orifice de sortie (appelé détroit inférieur). Pour franchir ces différents obstacles, le fœtus va effectuer une série de mouvements pour s'adapter à la forme de la cavité pelvienne. Ainsi en même

temps que la tête s’engage dans le détroit supérieur, elle est fléchie sur la poitrine, puis elle va se tourner légèrement vers le côté droit ou gauche pour entrer dans le bassin (pour présenter sa plus petite taille). Ce premier détroit franchi, la tête descend ensuite dans le bassin pour effectuer une seconde rotation. Pour accompagner ces mouvements, le bassin effectue également un mouvement particulier appelé “nutation”, conséquence de la poussée de la tête fœtale sur celui-ci. Ces mouvements sont illustrés par la Fig. 8.9 (b).

Ainsi au cours de la descente du fœtus, la partie supérieure des ailes iliaques effectue un mouvement d’adduction (flèches rouges qui se rapprochent du plan médian) accompagné d’une abduction de la partie basse de l’ischium (flèches bleues qui s’éloignent du plan médian). Durant ces rotations, la tête va appuyer sur le coccyx qui effectue alors un mouvement de flexion. Le coccyx effectuera ensuite un mouvement de contre-nutation lors du dégagement de la tête du bassin. Les mouvements du coccyx sont illustrés par les flèches bordeaux. Il est ainsi important de modéliser de manière pertinente à la fois les mouvements du bassin osseux et ceux de la tête du fœtus afin d’obtenir une trajectoire réaliste de la descente fœtale.

Notre approche pour le bassin osseux. Afin de limiter le temps d’exécution de la simulation, nous avons considérablement simplifié le maillage surfacique du bassin osseux en supprimant les arêtes vives, tout en conservant ses caractéristiques essentielles telles que les épines ischiatiques, la pointe du coccyx ainsi que la zone du pubis. Pour cela, nous avons tout d’abord réalisé un maillage très grossier qui s’appuie sur les boîtes englobantes des différentes parties connexes du bassin et au final nous obtenons un maillage comportant 1 750 nœuds comme illustré sur la Fig 8.10 (a). Pour les conditions limites visibles sur la Fig. 8.10 (b), nous avons considéré les ailes iliaques comme des organes fixes et indéformables, le rachis comme fixé et nous n’avons ainsi laissé mobile que la bascule au niveau inférieur.

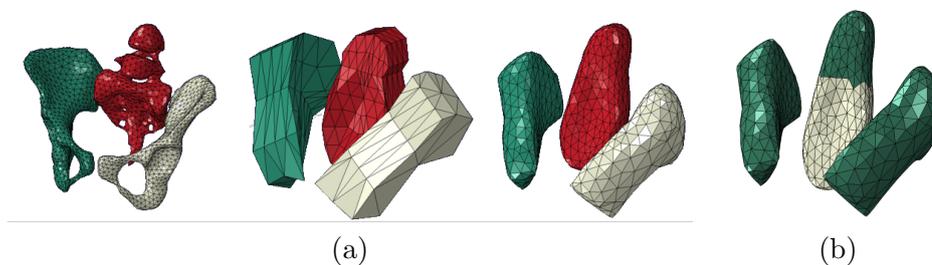


Figure 8.10 – (a) Simplification du maillage du bassin osseux (de gauche à droite) : maillage lissé (3 500 nœuds), boîtes englobantes, maillage final (1 750 nœuds). (b) Conditions limites du bassin osseux avec en vert les parties fixées..

Pour le comportement mécanique, [Fung \(1993\)](#) rappelle les expériences réalisées par [Yamada \(1970\)](#) suggérant que la loi de Hooke est applicable aux os dans le cadre de contraintes limitées. Nous avons ainsi utilisé la loi de Hooke pour la modélisation du bassin osseux permettant de petites déformations. Pour le module de Young de l’os, [Dufour and Pillu \(2007\)](#) présentent une valeur moyenne de 15 à 19 MPa comprenant les parties trabéculaire et cortical de l’os. L’os cortical étant beaucoup plus dense que l’os trabéculaire (la partie spongieuse de l’os), nous avons choisi un module de Young $E = 23$ MPa afin de se concentrer sur l’os cortical. Le coefficient de Poisson pour l’os étant compris entre 0,2 et 0,3 selon [Dalstra et al. \(1993\)](#), nous avons choisi une valeur $\nu = 0,3$ pour le bassin osseux. En outre, nous avons estimé la valeur de la densité à 1000 kg/m³, proche de celle de l’eau. A noter que la partie musculaire du bassin a été directement incorporée dans

la modélisation de l'abdomen de la parturiente et n'a donc pas été modélisée comme un organe indépendant, afin de simplifier notre modélisation de l'accouchement.

Comportement du bassin osseux lors de la simulation. Pour analyser le comportement du bassin lors de la simulation, nous avons observé la bascule du sacrum. La Fig. 8.11 présente ainsi l'évolution angulaire de la pointe du sacrum dans le plan sagittal. Nous constatons deux pics sur cette courbe. Le premier correspond au premier contact de la tête avec le sacrum, qui est poussé en arrière par les os du crâne du fœtus. Puis, lorsque la tête commence à entrer dans le canal vaginal, le second pic est causé par le passage du reste du corps du fœtus. En outre, nous pouvons noter qu'à la fin du travail (32 minutes plus tard), le bassin ne revient pas à sa position initiale. Cette simulation donne ainsi des premiers résultats en concordance avec la réalité.

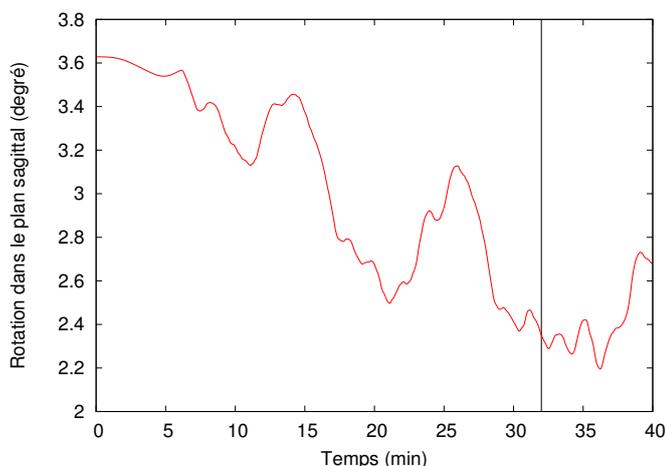


Figure 8.11 – Evolution angulaire de la pointe du sacrum dans le plan sagittal.

Amélioration du bassin osseux. Même si nous avons veillé à conserver la pointe du sacrum constituant une des zones les plus importantes du bassin, la simplification que nous avons faite sur le maillage du bassin osseux, a réduit la ligne de promontoire longeant l'intérieure ilio-pubienne au niveau du détroit supérieur, comme nous pouvons le voir sur la Fig. 8.12. Or durant l'accouchement, la tête du fœtus se place dans cette zone conditionnant le positionnement initial et l'arrivée dans la filière génitale du fœtus. La trajectoire de la descente fœtale est donc forcément perturbée par cette simplification. De plus nous pouvons voir sur cette même figure, que la simplification du bassin osseux a agrandi de manière très significative la symphyse pubienne. Or son rôle durant l'accouchement est important, provoquant une rotation et un re-positionnement autour de l'axe sagitto/frontal. Une amélioration du modèle géométrique du bassin osseux permettrait ainsi la correction du phénomène de déflexion de la tête à la sortie de la bascule du sacrum, phénomène que nous ne pouvons donc pas retrouver actuellement dans la simulation.

Notre approche pour le plancher pelvien. Durant la grossesse le volume de l'abdomen augmente significativement et les organes qui y sont présents (vessie, rectum, colonne vertébrale, côtes, foie, etc.) changent de position afin de laisser de la place pour le fœtus. Au cours de l'accouchement, le volume de l'abdomen va diminuer et les organes internes vont reprendre leur position initiale grâce à la présence d'une pression interne assurant la cohésion des organes. Pour des raisons évidentes de temps de calcul, nous ne pouvons

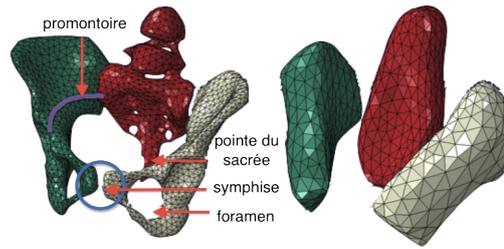


Figure 8.12 – Ligne de promontoire trop peu creusée (ligne violette), symphyse pubienne beaucoup trop importante (cercle bleu). Images issues de *Buttin (2010)*.

pas modéliser individuellement tous les organes qui y sont présents. C'est pourquoi, nous avons considéré l'abdomen comme un organe unique dont le contour a été défini à partir des images IRM et l'intérieur a été maillé à l'aide de tétraèdres en ôtant les autres organes déjà modélisés (bassin, utérus et fœtus) comme nous pouvons le voir sur la Fig. 8.13(a).

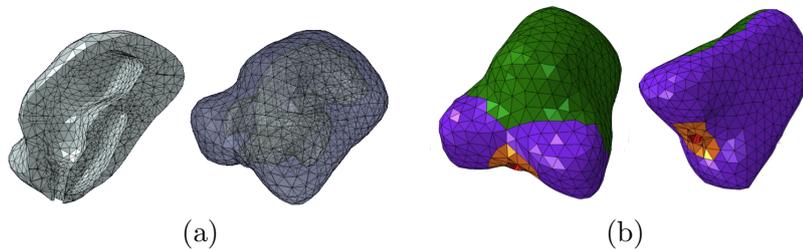


Figure 8.13 – (a) Maillage de l'abdomen de la parturiente (à droite) avec visualisation de sa coupe sagittale (à gauche). (b) Conditions limites de l'abdomen : (violet) parties fixées, (orange) étirement possible, (rouge) déplacements verticaux interdits, (vert) pas de contrainte.

Nous avons mentionné précédemment que le plancher pelvien avait été intégré dans le modèle de l'abdomen. Ainsi, nous avons donné un comportement mécanique assez proche des propriétés des tissus musculaires du bassin mou à notre abdomen, c'est-à-dire élastique et compressible. Ces propriétés permettent le re-positionnement des éléments de l'abdomen autour de l'utérus au cours de la descente du fœtus. Au final, l'abdomen a été modélisé comme un matériel hyper-élastique en utilisant la loi de Neo-Hooke avec une densité de 2500 kg/m^3 et $C10 = 5 \text{ kPa}$. Pour les conditions limites, nous avons fixé l'arrière de l'abdomen afin de prendre en compte la position de la parturiente (assise avec le dos fixe) durant l'accouchement. Mais nous ne pouvons pas imposer un déplacement nul à l'ensemble du contour de l'abdomen maternel. En effet, en fixant la partie basse de l'abdomen maternel, la sortie du fœtus ne serait pas autorisée. Cette dernière condition est réglée en autorisant uniquement les déplacements latéraux sur la partie basse de l'abdomen maternel autour de la zone vaginale. Ces conditions limites sont illustrées par la Fig. 8.13 (b). De plus afin d'assurer une cohésion entre les organes modélisés à l'intérieur de l'abdomen (utérus, bassin et fœtus), nous avons ajouté une légère pression représentant le tonus musculaire résiduel. Pour la gestion des contacts, le bassin osseux étant inclus dans l'abdomen avec plusieurs nœuds en commun, nous n'avons pas à traiter de collisions.

Amélioration pour le plancher pelvien. Une réelle modélisation du plancher pelvien permettrait la simulation de ses déchirures causées durant l'accouchement et une meilleure compréhension des efforts qui lui sont appliqués. Par contre, cette simulation est coûteuse en temps de calcul et devra être optimisée mais elle serait indéniablement un plus dans l'apprentissage des gestes de l'accouchement afin de comprendre et préserver au mieux

les tissus du plancher pelvien. Par ailleurs, un autre aspect que nous n'avons pas pris en compte concerne la déformation extérieure du périnée. En effet, à la fin de la sortie du fœtus, la vulve se gonfle et provoque une sortie temporaire des organes de manière horizontale. Or dans notre première modélisation, les mouvements autorisés pour la région externe de la vulve ne se situent que dans le plan transverse afin de permettre la dilatation du vagin et la descente du fœtus puisque l'absence de modélisation de la peau associée à la forte simplification de l'abdomen nous a contraint à bloquer les déplacements de la vulve le long de l'axe crâno-caudal. Pour remédier à ces lacunes, nous pouvons nous référer aux travaux proposés dans la littérature sur la modélisation du plancher pelvien.

Modèles existants pour le plancher pelvien. Plusieurs travaux de recherche se sont intéressés à la modélisation des muscles élévateurs de l'anus (MRA) du plancher pelvien. L'idée est d'étudier leur étirement lors du passage de la tête du fœtus afin de mieux comprendre leurs dommages durant l'accouchement et ainsi prévenir le phénomène du prolapsus. Dans ce sens, [d'Aulignac et al. \(2005\)](#); [Martins et al. \(2007\)](#) ont proposé un modèle éléments finis basé sur la loi de comportement de Neo-Hooke pour les MRA. Ces muscles ont été considérés comme quasi-incompressibles et isotropes avec un sens unique pour la fibre musculaire et les propriétés des matériaux des muscles du plancher pelvien ont été approchées en utilisant des données obtenues sur des tissus cardiaques. Un modèle réaliste du fœtus représenté par des éléments tétraédriques, a également été utilisé en le considérant comme un matériau quasi indéformable avec une très grande rigidité. Au final, le ratio maximum de l'étirement musculaire a été calculé en faisant varier la géométrie du bassin et de la tête du fœtus, ainsi que des paramètres tels que la présentation de la tête.

Plus récemment, [Li et al. \(2010a,b\)](#) ont étudié l'effet de l'anisotropie mécanique sur la réponse biomécanique des MRA pendant l'accouchement. Ils ont ainsi simulé le passage d'un crâne foetal au travers de deux modèles de plancher pelvien, incorporant les muscles releveurs de l'anus à différents ratios d'anisotropie. Les interactions entre les MRA et la tête du fœtus ont été modélisées en considérant une déformation élastique et un contact mécanique sans frottement. Les résultats ont montré une diminution substantielle de la force nécessaire pour l'accouchement à mesure que l'anisotropie des fibres augmentait.

En ce qui concerne la mesure des caractéristiques des organes pelviens, [Janda et al. \(2003\)](#) a publié une base de données contenant des paramètres géométriques du plancher pelvien et mécaniques des fibres musculaires établie à partir d'un cadavre d'une femme de 72 ans. [Brieu et al. \(2010\)](#); [Rubod et al. \(2007, 2008\)](#) ont mis au point un protocole d'expérimentation pour l'étude des propriétés mécaniques des tissus vaginaux mettant en avant le caractère élastique, non linéaire et grandement déformable des tissus organiques impliqués dans la statique pelvienne. Ils ont ensuite continué leur travail en proposant un modèle du système pelvien basé sur un modèle élément fini élastique linéaire en grands déplacements, afin de mieux comprendre le rôle des ligaments utérins dans le phénomène du prolapsus ([Rubod et al. \(2013\)](#)). Leur dernière publication s'intéresse au cas particulier de la femme enceinte ([Lepage et al. \(2014\)](#)) pour mieux comprendre l'impact du passage du fœtus sur les ligaments. Le modèle proposé est très complet mais demande plusieurs heures de calculs pour réaliser une simulation basée sur une trajectoire imposée pour la tête du fœtus.

Nous sommes actuellement en train de prospecter pour monter un projet commun entre le laboratoire TIMC-IMAG (avec Emmanuel Promayon) et le Laboratoire de Mécanique de Lille (avec Mathias Brieu). L'objectif de notre projet concerne la conception d'un nouveau modèle numérique de l'accouchement basé sur les connaissances du modèle numérique quasi

complet du système pelvien réalisé à Lille permettant des simulations interactives patient spécifique. L'idée est d'analyser ce modèle afin de le simplifier de manière pertinente pour diminuer son temps de simulation et arriver à du temps réel tout en conservant un réalisme adéquate pour un simulateur de gestes de l'accouchement.

Utérus

L'utérus est une poche musculaire creuse située dans la cavité pelvienne. Durant la grossesse, l'utérus est ramené vers l'avant de l'abdomen de la parturiente et sa taille va passer d'environ 8 cm à 35 cm de long en moyenne en fin de grossesse. Ce changement important modifie ses caractéristiques mécaniques qui sont ainsi difficiles à identifier et qui sont de plus très variables d'une femme à l'autre et selon le nombre de grossesses antérieures. Le col utérin est séparé du corps utérin par un étranglement à la base de l'utérus appelé isthme. Durant la grossesse, le col utérin reste fermé avec une longueur de 30 à 40 mm. Puis il va se ramollir et se raccourcir en fin de grossesse pour atteindre environ 13 mm. La Fig. 8.14 illustre un utérus gravide.

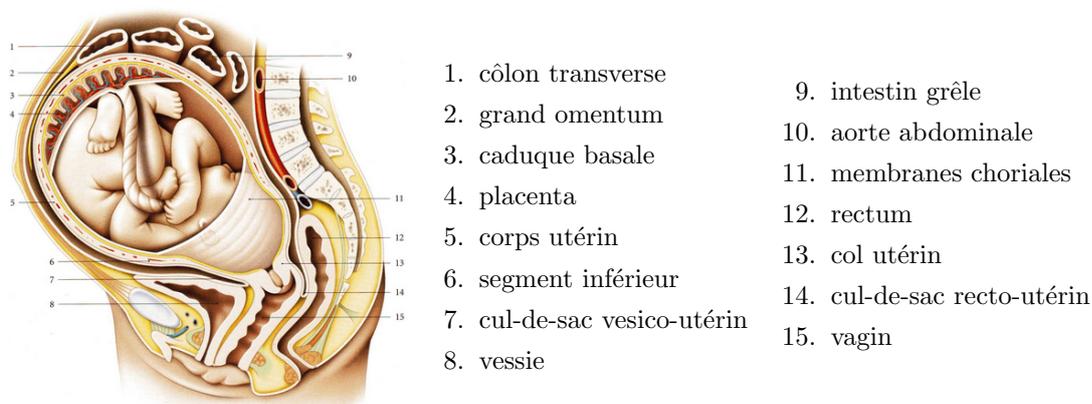


Figure 8.14 – Utérus gravide. Image issue de [Kamina \(2009\)](#).

Modèles existants pour l'utérus. [Mizrahi and Karni \(1975\)](#) ont présenté un modèle mécanique de l'utérus en utilisant une approche cinématique. Comme conditions limites, ils ont considéré que les déplacements du col de l'utérus sont nuls durant une contraction et restent constants pendant la seconde phase du travail c'est-à-dire à partir du moment où le col est complètement dilaté. En outre, ils ont supposé que lors de la déformation, le volume délimité par l'organe restait constant, en raison de l'incompressibilité du liquide intra-utérin. En ce qui concerne les propriétés mécaniques, [Mazza et al. \(2006\)](#) présentent une étude réalisée avec un dispositif d'aspiration permettant de caractériser les propriétés mécaniques du col de l'utérus *in vivo*. Les valeurs moyennes obtenues pour la raideur varient de 0,095 à 0,24 bar/mm, pour des tests effectués sur huit patientes, âgées de 47 à 69 ans ayant eu 1 à 4 enfants. Toutefois, il est difficile d'utiliser ces valeurs obtenues chez des femmes non-enceintes à cause des changements importants qui apparaissent dans les propriétés mécaniques durant la grossesse. Dans ce sens, [Bauer et al. \(2009\)](#) présentent une autre étude réalisée avec le même dispositif, mais sur des femmes enceintes (entre 21 et 36 semaines de gestation) mettant ainsi en avant les différences physiologiques entre les tissus d'une femme non gravide et enceinte. Les valeurs de rigidité obtenues varient alors entre 0,013 et 0,068 bar/mm confirmant le fait que les tissus de femmes non-enceintes sont significativement plus rigides que ceux de femmes enceintes en traction et en compression.

D'autres travaux ont ensuite suivi pour améliorer la modélisation de l'utérus. Un état de l'art plus complet sur ces différents modèles est présenté dans [Zara and Dupuis \(2017\)](#).

Notre approche pour l'utérus. Pour le modèle géométrique de l'utérus, nous avons modélisé comme un seul objet, l'utérus, le col utérin et le canal vaginal. La Fig. 8.15 présente ce modèle géométrique : à gauche le maillage initial issu des données médicales comportant 42 811 nœuds et à droite le maillage obtenu après lissage et reconstruction manuelle du col de l'utérus à partir de l'anatomie ([Kamina \(2006\)](#); [Schaal et al. \(2007\)](#)) en considérant une dilatation complète.

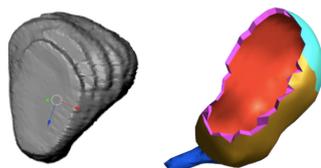


Figure 8.15 – Modèle géométrique de l'utérus : (gauche) maillage obtenu à partir des données médicales, (droite) maillage après lissage et reconstruction du col de l'utérus.

Pour la modélisation biomécanique de l'utérus, les travaux de [Mizrahi et al. \(1980\)](#) ont montré que le comportement des muscles utérins changeaient lors de l'accouchement, avec un comportement isotrope dans les premières phases de l'accouchement et un comportement anisotrope durant la progression du travail. Pour simplifier notre modèle, nous avons considéré un comportement anisotrope pour la membrane utérine. Ainsi, l'utérus a été modélisé comme un matériau hyper-élastique en utilisant la loi de comportement de Neo-Hooke, avec une densité de 950 kg/m^3 et $C10 = 30 \text{ kPa}$.

Comme conditions limites, les déplacements du canal vaginal sont limités dans le plan transversal pour permettre l'ouverture et la fermeture du canal vaginal, en évitant la descente des organes. Pour le traitement des collisions, les contacts entre l'utérus et le fœtus sont considérés sans frottement. Cette hypothèse est due au fait que lorsque la phase de travail commence, la poche utérine se vide du liquide amniotique mais les parois internes n'en restent pas moins très humides engendrant un comportement extrêmement lubrifié. Les contacts entre l'utérus et l'abdomen ont également été considérés sans frottement pour prendre en compte le phénomène visqueux existant entre tous les organes de l'abdomen.

Par contre, au lieu de modéliser le comportement du muscle que représente l'utérus, nous avons choisi de modéliser ses conséquences à savoir les contractions utérines (CU). Conformément à la réalité [Schaal et al. \(2007\)](#), les CU et les forces d'expulsion ont été modélisées comme deux champs de pression périodiques, uniformes sur la surface interne et externe de l'utérus, avec 12 périodes pour une durée de travail de 30 à 40 minutes. La Fig. 8.16 présente ces deux champs de forces.

Comportement de l'utérus lors de la simulation. Comme paramètre de validation, nous savons que les contractions utérines exercées sur l'utérus font diminuer son volume durant la descente du fœtus. Pour vérifier ce comportement, nous considérons deux points de l'utérus choisis dans le plan transversal. La Fig. 8.17 montre le mouvement de ces deux points lors de la simulation dans l'axe frontal/transversal. Nous pouvons voir que les déplacements de ces deux points suivent des directions opposées indiquant que le comportement de l'utérus correspond bien au comportement des CU. En outre, la taille de l'utérus est diminuée de $2/3$ environ à la fin d'un accouchement. Ce phénomène est vérifié en suivant l'évolution de la trajectoire sagito-frontal d'un point au sommet de l'utérus et

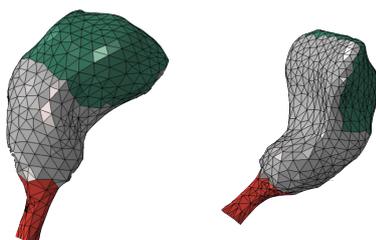


Figure 8.16 – Champs de forces appliqués sur l’utérus : en gris la partie sur laquelle sont appliquées les CU; en vert partie sur laquelle sont appliquées les CU, les forces de poussées abdominales et du diaphragme; en rouge la zone fixée sur le bassin.

de la comparer à un point de la partie basse de l’utérus. Sur la Fig. 8.18, nous pouvons observer que durant la simulation cette différence est de 230 mm au début de la phase de travail et de 80 mm à la sortie du fœtus ce qui nous donne approximativement une diminution de sa taille de l’ordre de 2/3, en concordance avec la réalité.

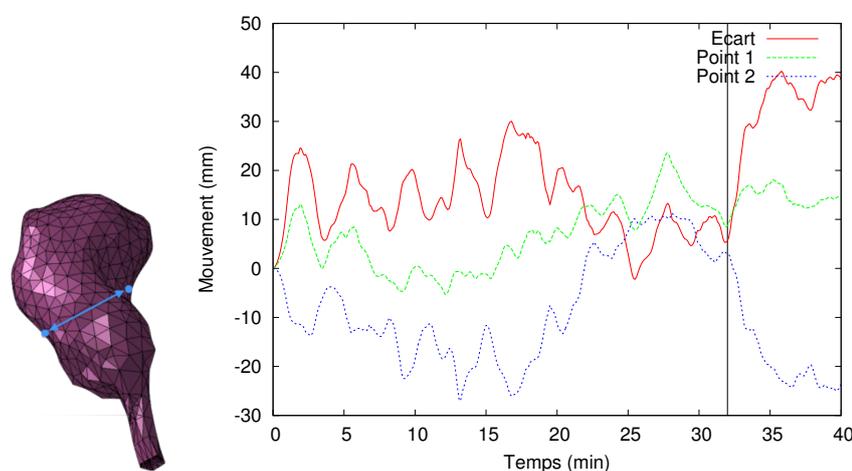


Figure 8.17 – Points de l’utérus suivis durant la simulation (à gauche). Déplacement de 2 points de l’utérus durant la simulation (à droite).

Amélioration de l’utérus. Les efforts de poussées et des CU ont été modélisés sous la forme de fonctions périodiques à partir de patrons de forces de contraction (valeur maximale, périodicité, cycles, issus de la littérature). Il serait intéressant d’analyser la sensibilité de la trajectoire du fœtus en fonction des efforts musculaires. La modélisation de ces forces peut également être modifiée pour tenir compte du fait que ces forces proviennent du fond de l’utérus, pour ensuite se propager sur l’ensemble de l’utérus.

Fœtus

Pour la simulation de l’accouchement, nous considérons uniquement le cas où l’accouchement se produit à terme c’est-à-dire à partir de la 37^e semaine de grossesse. Le fœtus pèse alors environ 3,5 Kg avec une taille d’environ 50 cm. La Fig. 8.19 (a) issue de la thèse de Mathieu Bailet (2014) représente la position de flexion du fœtus (position dite foetale) rendue possible par une grande souplesse de ses articulations. Notons que la proportion des organes d’un fœtus n’est pas la même que celle des adultes ou d’un enfant, avec un crâne de taille plus importante en proportion.

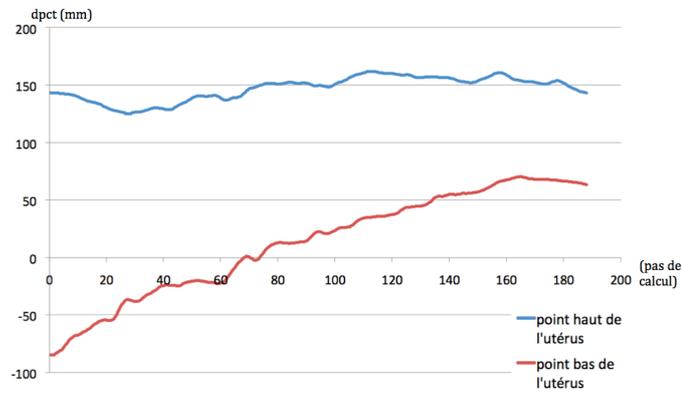


Figure 8.18 – Évolution de la trajectoire sagito-frontal d'un point de l'utérus.

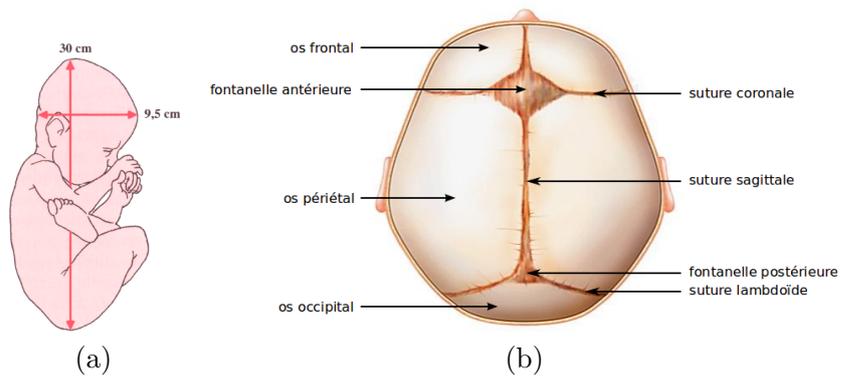


Figure 8.19 – (a) Position de flexion du fœtus dans l'utérus. (b) Voûte crânienne.

La tête fœtale est ainsi la partie la plus importante du fœtus, et lors de l'accouchement nous considérons que si la tête passe, le reste du corps passe également. Par contre les rotations de la tête sont primordiales lors de l'accouchement, afin de traverser les cavités du bassin. Mais il faut également que le crâne puisse se déformer. Ceci est rendu possible par la mobilité des os de la voûte crânienne les uns par rapport aux autres. Sur la Fig. 8.19 (b) sont représentées les sutures de la voûte crânienne. Ce sont des bandes membraneuses qui relient les différents os du crâne et qui se rejoignent au niveau de la fontanelle permettant le "modelage de la tête fœtale" au cours de sa traversée. Bien entendu, le volume intracrânien est quasi-constant grâce au déplacement du liquide cérébro-spinal dans le canal vertébral, assurant la préservation du cerveau pendant l'accouchement.

Modèles existants pour la tête fœtale. Lapeer and Prager (2001) ont proposé une extension des travaux de McPherson and Kriewall (1980a,b) sur la modélisation de la déformation du crâne fœtal issue des pressions exercées sur lui par le col de l'utérus durant la première phase de l'accouchement. Un modèle non linéaire statique par éléments finis a été employé pour modéliser la déformation du crâne du fœtus avec l'emploi d'un modèle théorique pour modéliser les pressions intra-utérines et les pressions du col de l'utérus. Les simulations sont statiques et le nombre important d'éléments ne permettent pas une simulation en temps réel. Ces travaux se sont ensuite poursuivis en proposant une plateforme de réalité augmentée basée sur une simulation d'accouchement instrumenté (avec forceps) (Lapeer et al. (2004)). Le but était de placer un forceps réel, traqué par des marqueurs optiques passifs, autour d'une tête fœtale virtuelle. Le contact entre le crâne virtuel et le forceps est alors détecté et visible sur le simulateur. Enfin dans Lapeer (2005), la faisabilité d'un modèle de contact mécanique temps réel a été testée pour décrire l'interaction entre l'instrument et la tête du fœtus. La conclusion était qu'il était sans doute possible d'atteindre des performances en temps réel, en employant un modèle dynamique explicite pour calculer la déformation des principaux os du crâne du fœtus, ou en employant un modèle quasi-statique pour calculer la déformation de l'intégralité de la tête fœtale. Mais cette modélisation n'était pas présentée.

Notre première approche pour le fœtus. D'un point de vue complexité, il n'est pas possible de modéliser les différents organes du fœtus. C'est pourquoi durant la thèse de Romain Buttin, nous avons créé trois zones au sein du fœtus comme illustré sur la Fig. 8.20 : le crâne, le corps et le tissu cutané. A noter que le crâne et le corps sont inclus dans le tissu cutané (avec plusieurs nœuds en commun) afin d'éviter d'avoir à traiter des collisions entre ces trois parties.

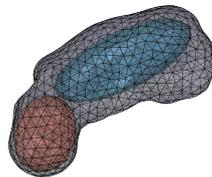


Figure 8.20 – Modèle du fœtus en 3 parties : crâne (rouge), corps (bleu) et tissu cutané (blanc).

- D'un point de vue mécanique, le crâne est considéré comme un objet déformable puisque celui-ci subit d'importantes déformations au cours de l'accouchement.
- Le corps est considéré comme un objet légèrement déformable afin de permettre à l'arrière du fœtus de se déplacer librement et de simuler les différentes articulations.
- Le tissu cutané est considéré comme plus élastique que le corps et le crâne, c'est-à-dire avec un module d'élasticité moins élevé. Par ailleurs, notons que le tissu cutané est le

seul organe compressible de notre modèle, caractéristique permettant de réduire les forces de répulsion induites par le contact entre le fœtus et l'utérus.

Les trois parties ont été modélisés avec la loi de comportement de Neo-Hooke avec $C10 = 130$ kPa pour le tissu cutané, $C10 = 75$ kPa pour le crâne et $C10 = 70$ kPa pour le corps. Par ailleurs, en considérant qu'un fœtus possède une densité musculaire moins importante qu'un adulte, nous avons choisi de prendre une densité moyenne fœtale légèrement plus faible que 1000 kg/m³ avec une densité de 400 kg/m³ pour le tissu cutané, et une densité similaire de 950 kg/m³ pour le crâne et le corps.

Comportement du fœtus lors de la simulation. Cette modélisation nous permet de simplifier considérablement le fœtus, tout en gardant une articulation au niveau de la tête induite par la déformation du tissu cutané comme nous pouvons le voir sur la Fig. 8.21(a). Cette figure montre également les déformations de la tête fœtale durant la simulation causée par la compression des muscles pelviens. Ainsi, même si nous n'avons pas modélisé un comportement plastique pour la tête du fœtus, le modèle hyper-élastique employé permet l'obtention de ce type de déformation durant la descente du fœtus.

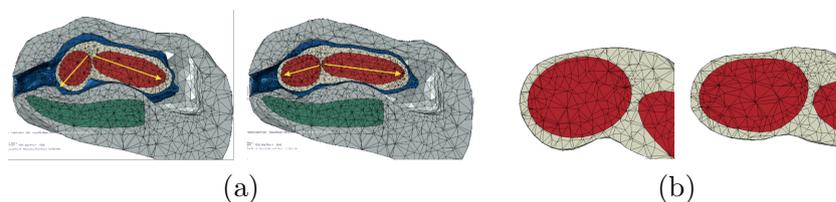


Figure 8.21 – (a) Articulation de la tête fœtale induite par la déformation du tissu cutané. (b) Écrasement de la tête fœtale au cours de la simulation.

Notre seconde approche pour la tête fœtale. Une modélisation plus fine de la tête fœtale a ensuite été réalisée durant le doctorat de Mathieu Bailet afin de prendre en compte ses différentes plaques permettant sa déformation lors de la descente dans le canal pelvien. Ce travail a été effectué en collaboration avec le Pr Olivier Dupuis de la maternité Lyon Sud et a été intégré au sein de la plateforme CamiTK (<http://camitk.imag.fr/>).

Pour le modèle géométrique, nous avons employé le modèle numérisé de la tête du mannequin utilisé dans le simulateur BirthSIM. Le maillage comporte $4\,466$ nœuds et $8\,928$ triangles. Afin de pouvoir modéliser les différentes parties du crânes ayant des propriétés mécaniques différentes, le maillage a été séparé en deux zones. Ce maillage est présenté par la Fig. 8.22 (a). Nous pouvons voir en bleu les os du crâne, et en vert les fontanelles et sutures. La Fig. 8.22 (b) présente les mesures employées pour évaluer les déformations subies par la tête fœtale. Ces mesures ont été mises en avant dans les travaux de [Sorbe and Dahlgren \(1983\)](#) et correspondent aux diamètres reflétant le plus les déformations issues de la pression intra-utérine appliquée sur la tête durant l'accouchement : le SOB (sous-occipito-bregmatique), le OrVD et le MaVD.

La modélisation mécanique de la tête est basée sur le modèle de coque CST-DKT présenté dans le chapitre 2, intégrant une contrainte pour la préservation du volume. Les propriétés mécaniques employées sont issues des travaux de [McPherson and Kriewall \(1980a\)](#) et de [McElhaney et al. \(1970\)](#) résumées dans le tableau 8.1.

En terme de conditions limites, les nœuds à la base du cou sont fixés pour éviter des déplacements rigides.

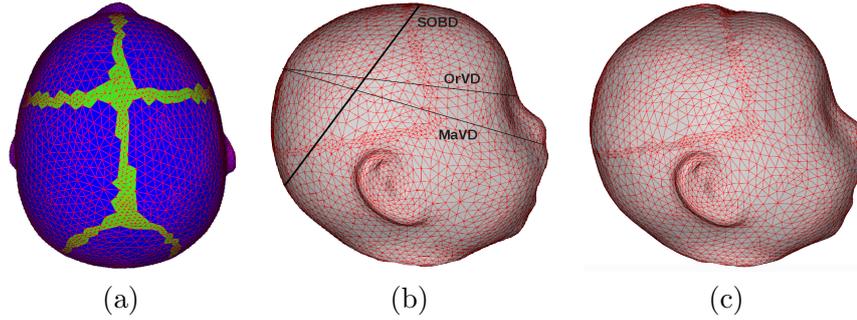
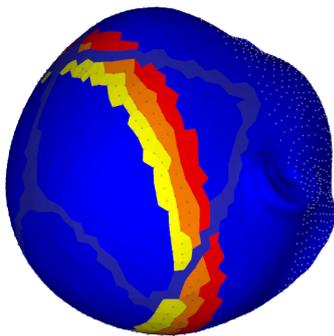


Figure 8.22 – (a) Maillage de la tête fœtale intégrant les os du crâne (bleu) ainsi que les fontanelles et sutures (vert). (b) Mesures du diamètre de la tête. (c) Déformation de la tête fœtale subissant les pressions intra-utérines et le contact avec le col de l’utérus.

	Module de Young	Coefficient de Poisson	Masse volumique	Épaisseur
Os du crâne	$E = 2 \text{ GPa}$	$\nu = 0,22$	$\rho = 1,8$	0,8 mm
Fontanelles et sutures	$E = 31,5 \text{ MPa}$	$\nu = 0,45$	$\rho = 1,0$	0,6 mm

Table 8.1 – Propriétés mécaniques des parties de la tête fœtale.

Comportement de la tête fœtale lors de la simulation. La Fig. 8.23 présente les pressions que nous avons appliquées sur la tête fœtale pour simuler celles subies durant un accouchement correspondant aux pressions intra-utérines et aux pressions exercées par le col de l’utérus. Nous nous sommes basés sur les mesures de [Rempen and Kraus \(1991\)](#) pour établir ces champs de pression. La Fig. 8.22(c) présente les déformations de la tête fœtale obtenues par la simulation, ainsi que les mesures de ces déformations mises en comparaison avec celles effectuées par [Sorbe and Dahlgren \(1983\)](#) et celles de [Lapeer and Prager \(2001\)](#) obtenues par simulation avec un modèle anisotropique pour les os du crâne et hyper-élastique pour les fontanelles et les sutures. Nous obtenons ainsi des résultats similaires à ceux de Lapeer, avec une variation plus importante pour le SOB. Nous observons également un chevauchement des os pariétaux (os sur les côtés du crâne) constituant un phénomène classique lors d’un accouchement [Schaal et al. \(2007\)](#).



- Les fontanelles et sutures ne subissent aucune pression.
- La zone autour du plan SOB correspond à la zone en contact avec le col de l’utérus. Trois différents champs de pression sont appliqués : 45 KPa pour la partie la plus haute du crâne (jaune), 40 KPa pour la partie du milieu (marron) et 30 KPa pour le champ le plus bas (rouge).
- Le reste du crâne est soumis à une pression de 7 KPa correspondant à la pression amniotique.

Figure 8.23 – Pressions intra-utérines et du col de l’utérus appliquées sur le crâne.

Amélioration pour la tête fœtale. Le modèle de coque CST-DKT proposé par Mathieu Bailet permet de simuler finement les déformations subies par la tête fœtale au cours de l’accouchement ([Bailet et al. \(2013, 2014\)](#)). Le travail suivant consiste à intégrer cette modélisation de la tête dans la simulation globale de l’accouchement, afin d’obtenir les

Mesures	Sorbe 1983		Lapeer 2001		Bailet et al. (2013)	
	ϕ	variation	ϕ	variation	ϕ	variation
MaVD	140.5	+1.90	129.3	+1.43	124.6	+1.29
OrVD	126.9	+2.20	119.3	+1.24	110.9	+1.13
SOBD	117.1	-1.70	88.7	-2.52	107.4	-3.57

Table 8.2 – Mesures du diamètres de la tête foetale (ϕ) et ses variations.

déformations issues de la simulation et observer si elles sont en concordance avec celles que nous avons obtenues en appliquant directement les pressions théoriques dessus. L'objectif final est bien entendu d'observer ensuite le comportement de la tête en interaction avec les forceps à l'issue de la réalisation du couplage de la simulation avec le simulateur physique.

Et les autres organes ?

Dans les travaux que nous avons réalisés sur la modélisation de l'accouchement, nous n'avons pas modélisé pour le moment les autres organes présents dans la cavité pelvienne comme le placenta, la vessie ou encore le rectum :

- En effet, le placenta est un organe relativement fin qui est situé à l'intérieur de l'utérus. Au niveau mécanique, cet organe n'apporte qu'une augmentation partielle de l'épaisseur de la paroi utérine au prix d'une forte augmentation du traitement des contacts. Au cours de l'accouchement nous verrons que le placenta est libéré quelques minutes après le fœtus durant la phase appelée "délivrance du placenta". Comme nous ne souhaitons pas simuler cette phase, nous avons fait le choix de ne pas intégrer cet organe dans notre modèle biomécanique.
- La vessie est un organe assez imposant, puisque elle peut contenir environ 350 mL de liquide. Or avant le début de l'accouchement, celle-ci est vidée réduisant significativement sa taille et limitant ainsi son implication sur la simulation des mouvements des organes lors de l'accouchement. C'est pourquoi, cet organe n'a pas été intégré à notre modèle.
- De la même manière, nous n'avons pas considéré le rectum qui n'a pas un grand rôle durant l'accouchement.

8.5 Couplage avec un dispositif physique

La Fig. 8.24 présente des images issues de la simulation complète de l'accouchement. Nous avons ainsi réussi à simuler la descente du fœtus provoquée par les contractions utérines et les forces de poussées maternelles.

Pour continuer sur une analyse cinématique de la simulation de la descente foetale, la Fig. 8.25 (à gauche) montre l'évolution durant la deuxième phase de l'accouchement de la tête foetale selon les 3 axes anatomiques.

- Cette phase dure 32 minutes dans notre simulation avec une vitesse moyenne de descente du fœtus de 0,09 mm/s, ce qui est conforme à la réalité avec une durée moyenne de 30 à 45 minutes.
- Nous pouvons voir que le déplacement principal s'effectue selon l'axe sagittal/frontal. L'amplitude de déplacement de la tête est en effet de 51,5 mm le long de l'axe

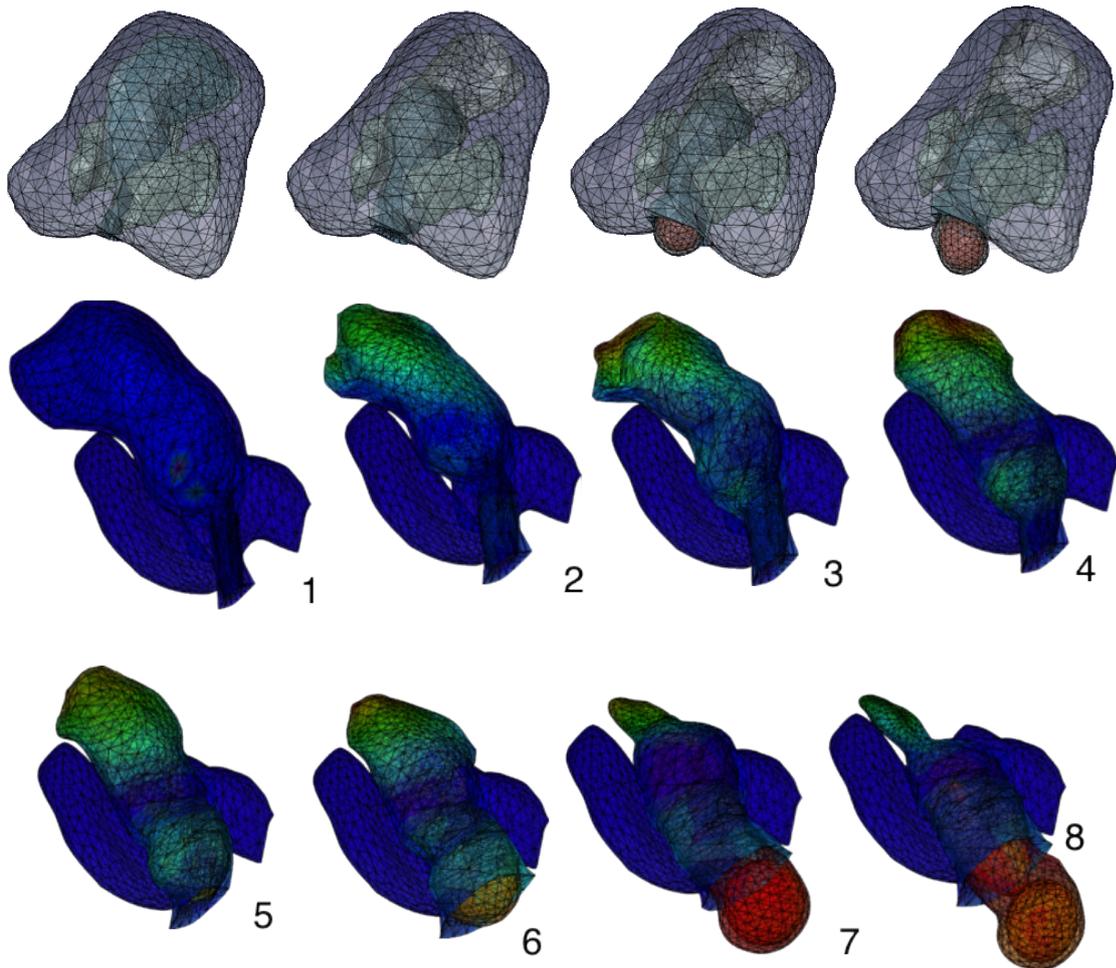


Figure 8.24 – Différentes phases de la simulation de l'accouchement.

frontal/transversal ; 63,5 mm le long de l'axe sagittal/transversal ; 184 mm pour l'axe sagittal/frontal.

- Si nous nous concentrons sur le comportement selon l'axe sagittal/frontal, nous pouvons observer que la vitesse de la tête fœtale n'est pas linéaire. En effet, la parturiente ne pousse pas en continu, impliquant ainsi des augmentations ou diminutions de vitesse de la tête fœtale au cours de l'accouchement. Ce comportement est plus visible dans la Fig. 8.25 (à droite) qui présente l'évolution de la vitesse de la tête fœtale dans l'axe sagittal/frontal. Les parties négatives de cette courbe correspondent ainsi au mouvement inverse de la tête.

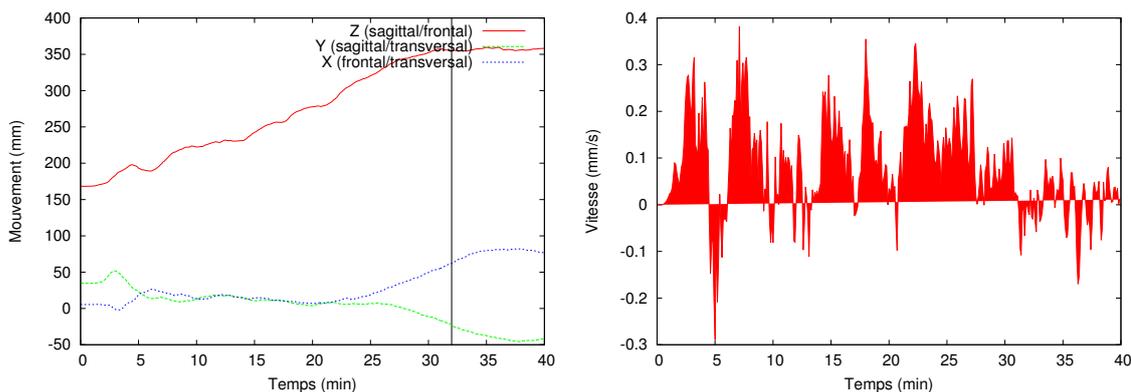


Figure 8.25 – Déplacement de la tête fœtale le long des axes anatomiques (à gauche). Vitesse de la tête fœtale le long de l'axe sagittal/frontal (à droite).

Premier couplage avec BirthSIM. Les résultats de la progression du fœtus dans le canal vaginal ont ensuite été intégrés dans le simulateur BirthSIM. Il s'agissait pour le moment de simplement charger les positions de la tête fœtale calculées au cours de la simulation au sein du simulateur physique pour faire avancer la tête en conséquence. La Fig. 8.26 présente une comparaison entre notre trajectoire calculée par la simulation et celle habituellement employée dans BirthSIM. La trajectoire initialement employée dans BirthSIM (en rouge) ne tient pas compte de la morphologie de la parturiente et du fœtus. Son déplacement est ainsi supposé linéaire et cyclique, alors que sur la courbe verte issue de la simulation, nous percevons à 15 min un changement correspondant au passage de la tête dans le plancher pelvien.

Ce premier travail autour de la simulation permet de fournir des trajectoires de la descente du fœtus plus réalistes que celles précédemment employées pour alimenter le simulateur physique BirthSIM. Par contre, dans le projet SAGA nous sommes allés plus loin en réalisant un réel couplage entre la simulation numérique et le simulateur physique BirthSIM, afin que l'obstétricien puisse interagir avec des instruments de type forceps et s'entraîner à leurs poses. Il s'agit donc de réaliser une réelle interaction en gérant les flux de données entre les deux composants du simulateur, et en assurant une certaine synchronisation entre les deux. Les données qui vont transiter concernent la position de la tête fœtale, les forces appliquées sur celle-ci, et le mouvement des forceps. Pour cela des capteurs de positions et d'efforts sont positionnés sur la tête du mannequin de BirthSIM et sur les forceps. Son nombre de degrés de liberté va également être augmenté, afin de permettre les rotations de la tête et ainsi suivre au mieux la trajectoire calculée par la simulation.

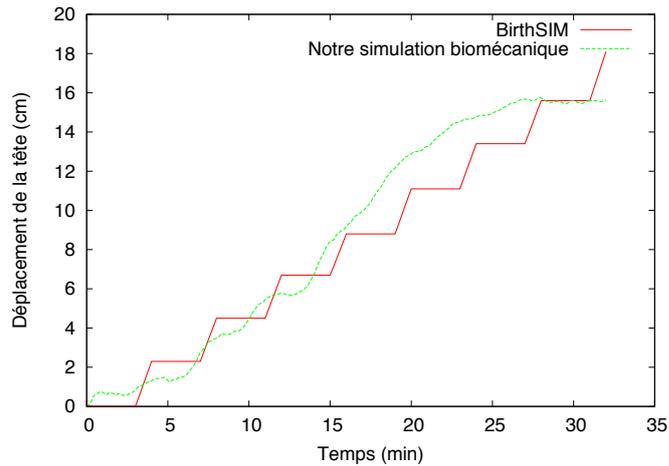


Figure 8.26 – Comparaison entre la trajectoire de la tête fœtale employée dans BirthSIM avec celle obtenue par notre simulation biomécanique.

8.6 Discussion

Les travaux antérieurs effectués sur la simulation de l'accouchement font une approximation des efforts exercés sur l'utérus et l'abdomen en imposant des conditions limites cinématiques, c'est-à-dire en imposant la trajectoire de la tête fœtale. Dans le travail réalisé dans [Buttin et al. \(2009, 2011, 2013\)](#), le comportement cinématique du crâne du fœtus est calculé en fonction des efforts exercés.

La Fig. 8.27 illustre l'intégralité de ce modèle avec les conditions limites des différents organes. Le tableau 8.4 présente les propriétés mécaniques et lois de comportement des organes. Enfin, le tableau 8.3 (issu de la thèse de Romain Buttin) récapitule l'intégralité des paramètres que nous avons modélisés ainsi que leurs perspectives d'amélioration.

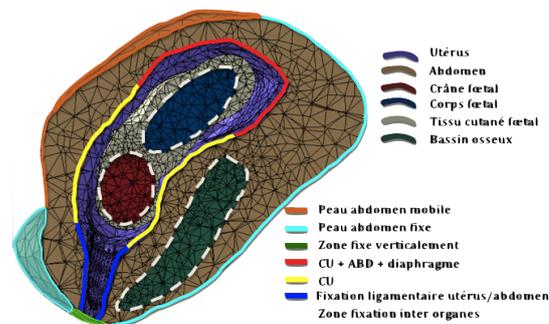


Figure 8.27 – Schéma global du modèle de l'accouchement (ABD : poussées abdominales).

Durée de l'accouchement	V	Durée moyenne entre 30 et 40 min. Simulation dure 32 min.
Trajectoire du fœtus	V	Trajectoire réaliste avec des phases de légères remontées du fœtus qui mériteraient d'être amplifiées (remontée moyenne de l'ordre du centimètre; notre modélisation est à 3-4 mm).
Vitesse d'expulsion du fœtus	V	La vitesse moyenne des accouchements est de l'ordre de 0,1 mm/s. Notre modélisation est à 0,09 mm/s.
Choix des organes modélisés	V	Les organes non pris en compte (placenta, vessie, rectum) avaient suffisamment peu d'impact pour être occultés.
Contacts sans frottement	V	Lubrification justifiée par la présence du liquide amniotique et l'ajout de savon en début d'accouchement par les infirmières.
Périodicité et cycle des contractions	V	Conforme à la moyenne.
Déformation du périnée	X	Prendre en compte l'aspect étirable de la peau.
Bascule du sacrum	V	Comportement réaliste du sacrum avec des déplacements angulaires de quelques degrés.
Comportement du sacrum en fin d'accouchement	V	Ecart de 1,5 degrés avec la position initiale. Conforme à la réalité.
Comportement de l'utérus	V	Volume du fond utérin diminué de 66 % en fin d'accouchement. Conforme à la réalité.
Comportement de la tête fœtale	X	Modélisation actuelle élastique. Ajout de loi plastique et modélisation des plaques osseuses et des fontanelles.
Simplification géométrique	~	Bon pour tous les organes à l'exception du bassin (revoir symphyse pelvienne et promontoire).
Intégration de la simulation dans un système haptique	V	Première évaluation réalisée avec BirthSIM.

Table 8.3 – Récapitulatif des paramètres modélisés et des points à améliorer.

		Loi de comportement	Propriétés mécaniques
Bassin		Elastique (Hooke)	$E = 23 \text{ MPa}$ $\nu = 0.3$ $\rho = 1,000 \text{ kg/m}^3$
Foetus	peau	Hyper-élastique (Néo-Hooke)	$C10 = 130 \text{ kPa}$ $\rho = 400 \text{ kg/m}^3$
	crâne		$C10 = 75 \text{ kPa}$ $\rho = 950 \text{ kg/m}^3$
	corps		$C10 = 70 \text{ kPa}$ $\rho = 950 \text{ kg/m}^3$
Abdomen		Hyper-élastique (Néo-Hooke)	$C10 = 5 \text{ kPa}$ $\rho = 2,500 \text{ kg/m}^3$
Utérus		Hyper-élastique (Néo-Hooke)	$C10 = 30 \text{ kPa}$ $\rho = 950 \text{ kg/m}^3$

Table 8.4 – Propriétés mécaniques et lois de comportement des organes modélisés.

Apport de la simulation numérique pour un simulateur médical. L'emploi d'une simulation numérique dans la conception d'un simulateur pour l'apprentissage de gestes médicaux permet de multiplier les situations possibles par rapport à l'emploi d'un simulateur uniquement basé sur un dispositif physique qui comporte un seul type de mannequin anthropomorphique. Nous pouvons ainsi plus facilement gérer les variables des scénarios mis en place pour l'apprentissage. Dans le cadre du simulateur pour l'apprentissage des geste de l'accouchement, cela se traduit par l'emploi au sein de la simulation numérique de maillages permettant de prendre en compte des morphologies de bassins pelviens et de tête de fœtus différentes. Nous pouvons ainsi analyser les forces de poussées maternelles et l'amplitude des contractions utérines nécessaires au dégagement du fœtus, et ainsi mesurer pour quelle intensité l'emploi des forceps devient nécessaire. Cela peut être dès lors que la parturiente est fatiguée (l'apprentis doit alors saisir que les poussées maternelles seront insuffisantes et qu'une aide extérieure est nécessaire) ou pour une certaine morphologie de bassin et de tête fœtale pour lesquels une force extérieure est de toute façon nécessaire. La simulation numérique permet ainsi de tester et simuler ces différentes situations. Nous pouvons également effectuer une simulation à différents niveaux d'engagement du fœtus dans le canal pelvien ou encore effectuer une bascule du bassin ou une rotation de la tête du fœtus au cours de la simulation permettant de simuler les manœuvres pouvant être faites par la sage-femme durant l'accouchement.

Naturellement la simulation permet également de visualiser la descente du fœtus durant l'accouchement et de comprendre les répercussions des gestes effectués sur la tête du fœtus lors de l'emploi de forceps. Cette visualisation aide à l'acquisition des gestes par l'apprentis en visualisant ce qui n'est normalement pas visible, avec une compréhension direct des implications de chacun des gestes effectués (par exemple la visualisation des conséquences liées à un mauvais positionnement des forceps ou encore à l'application d'une force d'extraction trop importante).

Enfin, l'usage premier de la simulation numérique couplée à un dispositif haptique est de renvoyer les informations au dispositif haptique relatives aux mouvements effectués par l'apprentis. Ceci permet de ne pas employer des trajectoires de descente fœtale prédéfinies au sein du dispositif physique, mais de calculer la descente relative aux paramètres du scénarios et aux gestes effectués par l'apprentis. Par contre, il s'agira dans la phase de validation du

simulateur complet, d'analyser si il y a des répercussions négatives sur l'apprentissage dans le fait de coupler un dispositif physique ayant toujours la même géométrie à une simulation dont les maillages des objets ne correspondent pas à ceux du mannequin physique et qui peuvent varier d'un exercice à un autre.

Encore un peu de travail pour réaliser le simulateur complet. Pour réaliser la simulation de l'accouchement qui sera intégrée dans le simulateur proposé dans le projet SAGA, il reste des verrous techniques et scientifiques à lever. En effet, le doctorat de Romain Buttin a constitué une première étape pour vérifier la faisabilité de la simulation de la descente du fœtus guidée par les contractions utérines et les poussées maternelles. Pour ce travail, nous avons employé le progiciel Abaqus puisque l'idée était de nous concentrer sur la détermination du modèle mécanique des différents organes impliqués permettant cette descente. Or la réalisation du simulateur complet nécessite un couplage entre la simulation numérique et le dispositif physique. Il s'agit alors de pouvoir maîtriser entièrement la simulation pour pouvoir diminuer les temps de calculs et gérer le couplage avec le dispositif physique. La suite des travaux sur la réalisation du simulateur SAGA a ainsi concerné la mise en place d'une nouvelle simulation de l'accouchement à partir des apports que nous avons réalisés autour de cette thématique (présentés dans les chapitres précédents).

L'idée est ainsi d'employer les modèles que nous avons mis en place au fil des années qui permettent de gérer des simulations d'objets mixtes dont les éléments peuvent évoluer au cours de la simulation. Le temps réel est bien entendu une contrainte forte du projet avec en plus un traitement efficace des collisions à effectuer entre les organes (en contact permanent) et les instruments rigides présents dans la simulation complète. La gestion du temps est ainsi très importante, avec des latences à prévoir entre les sorties de la simulation et les entrées nécessaires pour le dispositif physique. Nous nous inspirons pour cela des travaux réalisés par Christian [Duriez \(2013\)](#) qui permettent de gérer ce couplage à différentes fréquences de restitutions des résultats.

La réalisation du projet SAGA a ainsi permis de poursuivre le développement de la plateforme NExT en collaboration avec le laboratoire TIMC-IMAG de Grenoble.

Contributions

Publications

- F. Zara. Simuler en temps réel la descente du fœtus. *Quadrature*, numéro 107, EDP Sciences, 2018.
- F. Zara, O. Dupuis. Chap. 15 : Uterus - Biomechanical modeling of uterus. Application to a childbirth simulation. *Biomechanics of Living Organs : Hyperelastic Constitutive Laws for Finite Element Modeling*, 2017.
- M. Bailet. *Modèles biomécaniques pour la simulation interactive de l'accouchement*. Thèse de doctorat en informatique, Université de Grenoble, dec 2014.
- M. Bailet, F. Zara, and E. Promayon. Biomechanical model of the fetal head for interactive childbirth simulation. In *SURGETICA 2014*, Dec. 2014.
- M. Bailet, F. Zara, and E. Promayon. Shell finite element model for interactive fetal head deformation during childbirth. *CMBBE - Computer Methods in Biomechanics and Biomedical Engineering*, 16 :312–314, Sept. 2013.
- R. Buttin, F. Zara, B. Shariat, T. Redarce, and G. Grangé. Biomechanical simulation of the fetal descent without imposed theoretical trajectory. *Computer Methods and Programs in Biomedicine*, 111(2) :389–401, Aug. 2013. ISSN 0169-2607.
- R. Buttin, F. Zara, B. Shariat, T. Redarce, and G. Grangé. Simulation biomécanique de la descente fœtale sans trajectoire théorique imposée. *Revue Electronique Francophone d'Informatique Graphique*, 5(2) :1–13, Dec. 2011.
- R. Buttin. *Modélisation biomécanique du système reproductif féminin et du fœtus pour la réalisation d'un simulateur virtuel d'accouchement*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1, oct 2010.
- R. Buttin, F. Zara, B. Shariat, and T. Redarce. A Biomechanical Model of the Female Reproductive System and the Fetus for the realization of a Childbirth Virtual Simulator. In *IEEE Engineering in Medicine and Biology Society (EMBC'09)*, Sept. 2009.

Conférences invité

- F. Zara. Présentation lors de la "French American Innovation Day" (FAID), organisée par le Service pour la "Science et la Technologie" de l'Ambassade de France aux Etats-Unis. Houston, Etats-Unis, le 7 mars 2019.
- F. Zara. Présentation lors du colloque "IA & médecine : une chance?" organisé par le conseil de l'ordre des médecins de la région Rhône-Alpes. le 9 novembre 2018 à Lyon.
- F. Zara. Présentation lors du congrès annuel de la SIFUD PP (Société Interdisciplinaire Francophone d'UroDynamique et de Pelvi Périnéologie). La Baule, France, septembre 2015.

Logiciel

- **NExT - New Experimental Tool** - Librairie de simulation physique intégrant plusieurs modèles physiques (masses-ressorts, masses-tenseurs, coque, éléments finis volumiques). Gestion des communications entre la simulation numérique et différentes interfaces haptiques permettant un couplage entre les deux et ainsi de l'interaction.

Bibliographie

- R. H. Allen, B. R. Bankoski, and D. A. Nagey. Simulating birth to investigate clinician-applied loads on newborns. *Medical engineering and Physics*, 17(5) :380–384, 1995. ISSN 1350-4533.
- N. Aspert, D. Santa-Cruz, and T. Ebrahimi. MESH : Measuring errors between surfaces using the hausdorff distance. In *IEEE International Conference on Multimedia and Expo*, volume I, pages 705 – 708, 2002.
- M. Attene and B. Falcidieno. ReMESH : An interactive environment to edit and repair triangle meshes. In *Shape Modeling and Applications (SMI)*, pages 271–276, 2006.
- M. Bailet. *Modèles biomécaniques pour la simulation interactive de l'accouchement*. Thèse de doctorat en informatique, Université Joseph Fourier, Dec. 2014. URL <https://hal.archives-ouvertes.fr/tel-01158380>.
- M. Bailet, F. Zara, and E. Promayon. Shell finite element model for interactive fetal head deformation during childbirth. *CMBBE - Computer Methods in Biomechanics and Biomedical Engineering*, 16 :312–314, Sept. 2013.
- M. Bailet, F. Zara, and E. Promayon. Biomechanical model of the fetal head for interactive childbirth simulation. In *SURGETICA 2014*, Dec. 2014.
- M. Bauer, E. Mazza, M. Jabareen, L. Sultan, M. Bajka, U. Lang, R. Zimmermann, and G. Holzapfel. Assessment of the in vivo biomechanical properties of the human uterine cervix in pregnancy using the aspiration test : A feasibility study. *European Journal of Obstetrics & Gynecology and Reproductive Biology*, 144 :S77–S81, 5 2009.
- J.-D. Boissonnat and B. Geiger. 3D simulation of delivery. In G. M. Nielson and D. Bergeron, editors, *Visualization 93*, pages 416–419, San Jose CA, 1993. IEEE Computer Society Press.
- J.-D. Boissonnat and M. Yvinec. *Géométrie algorithmique*. Ediscience edition, 1995. ISBN 2840741121.
- M. Brieu, N. Bhatnagar, M. Boukerrou, M. Cosson, and C. Rubod. Experiments and finite element modelling for the study of prolapse in the pelvic floor system. *Computer Methods in Biomechanics and Biomedical Engineering*, 13(3) :349–357, 2010.
- R. Buttin. *Modélisation biomécanique du système reproductif féminin et du fœtus pour la réalisation d'un simulateur virtuel d'accouchement*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1, oct 2010.

- R. Buttin, F. Zara, B. Shariat, and T. Redarce. A Biomechanical Model of the Female Reproductive System and the Fetus for the realization of a Childbirth Virtual Simulator. In *IEEE Engineering in Medicine and Biology Society (EMBC'09)*, Sept. 2009.
- R. Buttin, F. Zara, B. Shariat, T. Redarce, and G. Grangé. Simulation biomécanique de la descente foetale sans trajectoire théorique imposée. *Revue Electronique Francophone d'Informatique Graphique*, 5(2) :1–13, Dec. 2011.
- R. Buttin, F. Zara, B. Shariat, T. Redarce, and G. Grangé. Biomechanical simulation of the fetal descent without imposed theoretical trajectory. *Computer Methods and Programs in Biomedicine*, 111(2) :389–401, Aug. 2013. ISSN 0169-2607.
- M. Dalstra, R. Huiskes, A. Odgaard, and L. Van Erning. Mechanical and textural properties of pelvic trabecular bone. *Journal of biomechanics*, 26(4-5) :523–535, 1993. ISSN 0021-9290.
- D. d'Aulignac, J. A. C. Martins, T. Mascarenhas, R. M. N. Jorge, and E. B. Pires. A shell finite element model of the pelvic floor muscles. 2005.
- M. Dufour and M. Pillu. *Biomécanique fonctionnelle : Membres-Tête-Tronc*. Masson, 2007.
- C. Duriez. *Real-time haptic simulation of medical procedures involving deformations and device-tissue interactions*. Habilitation à diriger des recherches, Université des Sciences et Technologie de Lille - Lille I, Feb. 2013.
- J. S. Eggert, M. S. Eggert, and P. Vallejo. Interactive education system for teaching patient care. patent no. us2003/0081968a1, May 2003.
- C. Forster, I. Bechtold, and F. Melchert. Clinical use of the birth simulation program "anapelvis 2.0" for prediction of feto-maternal disproportion]. *Zentralblatt fur gynecakologie*, 123(8) :465–468, 2001.
- Y. Fung. *Biomechanics. Mechanical properties of living tissues*. Springer, second edition, 1993.
- R. Gardner. Simulation and simulator technology in obstetrics : past, present and future. *Expert Review in Obstetrics & Gynecology*, 2(6) :775–90, nov 2007.
- Gaumard Scientific. S575 noelle with newborn hal. <http://www.gaumard.com> (site Web consulté en novembre 2010).
- B. Geiger. *Three-dimensional modeling of human organs and its application to diagnosis and surgical planning*. PhD thesis, Ecole des Mines de Paris, 1993.
- J. Graves and G. Savannah. Educational apparatus for teaching obstetrics and midwifery, 1951.
- S. Janda, F. van der Helm, and S. de Blok. Measuring morphological parameters of the pelvic floor for finite element modelling purposes. *Journal of Biomechanics*, 36(6) : 749–757, 2003.
- P. Kamina. *Anatomie clinique : anatomie générale, membre*. Maloine, third edition, 2006.
- P. Kamina. *Anatomie clinique : Tome 1, Anatomie générale, membres*. Anatomie clinique. Maloine, 4e édition edition, 2009.

- A. Kheddar, C. Devine, M. Brunel, C. Duriez, and O. Sidony. Preliminary design of a childbirth simulator haptic feedback. In *IEEE/RSJ, International Conference on Intelligent Robots and Systems*, volume 4, pages 3270–3275, 2004.
- C.-F. Knapp and G.-S. Eads. dynamic childbirth simulator for teaching maternity patient care, 1974.
- R. J. Lapeer. A mechanical contact model for the simulation of obstetric forceps delivery in a virtual/augmented environment. *Studies in Health Technology and Informatics*, 111 : 284–289, feb 2005.
- R. J. Lapeer and R. W. Prager. Fetal head moulding : finite element analysis of a fetal skull subjected to uterine pressures during the first stage of labour. *Journal of Biomechanics*, 34 :1125–1133, 2001.
- R. J. Lapeer, M. Chen, and J. Villagrana. An Augmented Reality based Simulation of Obstetric Forceps Delivery. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004)*, pages 274–275, nov 2004.
- J. Lepage, , C. Jayyosi, P. Lecomte-Grosbras, M. Brieu, C. Duriez, M. Cosson, and C. Rubod. Biomechanical pregnant pelvic system model and numerical simulation of childbirth : impact of delivery on the uterosacral ligaments, preliminary results. *International urogynecology journal*, 26(4) :497–504, 2014.
- G. Letterie. How virtual reality may enhance training in obstetrics and gynecology. *American journal of obstetrics and gynecology*, 187 :S37–S40, sep 2002.
- X. Li, J. Kruger, M. Nash, and P. Nielsen. Anisotropic effects of the levator ani muscle during childbirth. *Biomechanics and modeling in mechanobiology*, aug 2010a.
- X. Li, J. A. Kruger, M. P. Nash, and P. M. Nielsen. Effects of fetal head motion on pelvic floor mechanics. pages 129–137, 2010b.
- J. A. C. Martins, M. P. M. Pato, E. B. Pires, R. M. N. Jorge, M. Parente, and T. Mascarenhas. Finite element studies of the deformation of the pelvic floor. *Annals of the New York Academy of Sciences*, pages 316–334, apr 2007.
- E. Mazza, A. Nava, M. Bauer, R. Winter, M. Bajka, and G. Holzapfel. Mechanical properties of the human uterine cervix : an in vivo study. *Medical Image Analysis*, 10(2) :125–136, sep 2006.
- J. H. McElhaney, J. L. Fogle, J. W. Melvin, R. R. Haynes, V. L. Roberts, and N. M. Alem. Mechanical properties of cranial bone. *Journal of Biomechanics*, 3(5) :495–511, 1970.
- G. McPherson and T. J. Kriewall. The elastic modulus of fetal cranial bone : A first step towards an understanding of the biomechanics of fetal head molding. *Journal of Biomechanics*, 13(1) :9–16, 1980a.
- G. McPherson and T. J. Kriewall. Fetal head molding : An investigation utilizing a finite element model of the fetal parietal bone. *Journal of Biomechanics*, 13(1) :17–26, 1980b.
- J. Mizrahi and Z. Karni. A mechanical model for uterine muscle activity during labor and delivery. *Israel Journal of Technology*, 13 :185–191, 1975.

- J. Mizrahi, Z. Karni, and W. Polishuk. Isotropy and anisotropy of uterine muscle during labor contraction. *Journal of Biomechanics*, 13(3) :211–218, 1980.
- R. Moreau. *Le simulateur BirthSIM : un outil complet pour la formation sans risque en obstétrique*. PhD thesis, Institut National des Sciences Appliquées, Lyon, France, 2007.
- T. Obst, R. Burgkart, E. Ruckhäberle, and R. Riener. The delivery simulator : A new application of medical VR. In *Annual Medicine Meets Virtual Reality Conference (MMVR'04)*, pages 281–287, Jan 2004.
- A. Rempen and M. Kraus. Pressures on the fetal head during normal labor. *Journal of Perinatal Medicine-Official Journal of the WAPM*, 19(3) :199–206, 1991.
- R. Riener and R. Burgkart. Birth simulator (geburtensimulator). Patent No : WO 03/001482 A1, 03 jan 2003.
- C. Rubod, M. Boukerrou, M. Brieu, P. Dubois, and M. Cosson. Biomechanical properties of vaginal tissue. part 1 : New experimental protocol. *The Journal of urology*, 178(1) : 320–325, 2007.
- C. Rubod, M. Boukerrou, M. Brieu, J.-C. Clay, P. Dubois, and M. Cosson. Biomechanical properties of vaginal tissue : preliminary results. *international Urogynecology Journal*, 2008.
- C. Rubod, P. Lecomte-Grosbras, M. Brieu, G. Giraudet, N. Betrouni, and M. Cosson. 3d simulation of pelvic system numerical simulation for a better understanding of the contribution of the uterine ligaments. *International urogynecology journal*, 24(12) : 2093–2098, 2013.
- J.-P. Schaal, D. Riethmuller, R. Maillet, and M. Uzan. *Mécanique et Technique Obstétricales*. sauramps medical, third edition, fev 2007. ISBN 9782-84023-471-5.
- T. Sielhorst, T. Obst, R. Burgkart, R. Riener, and N. Navab. An augmented reality delivery simulator for medical training. In *In International Workshop on Augmented Environments for Medical Imaging - MICCAI Satellite Workshop*. 141, 2004.
- B. Sorbe and S. Dahlgren. Some important factors in the molding of the fetal head during vaginal delivery - a photographic study. *International Journal of Gynecology & Obstetrics*, 21(3) :205 – 212, 1983. ISSN 0020-7292. doi : [http://dx.doi.org/10.1016/0020-7292\(83\)90081-4](http://dx.doi.org/10.1016/0020-7292(83)90081-4).
- H. Yamada. *Strength of biological materials*. Williams & Wilkins in Baltimore, 1970.
- F. Zara and O. Dupuis. Chapter 15 :uterus. In Elsevier, editor, *Biomechanics of Living Organs : Hyperelastic Constitutive Laws for Finite Element Modeling*, pages 325–346. Joe Hayton, 2017. ISBN 978-0-12-804009-6.

Chapitre 9

Conclusion et perspectives

Ce dernier chapitre fait le bilan des contributions obtenues durant mes années de recherche en tant que maître de conférences à l'Université Lyon 1 au sein du LIRIS. Il se termine par la proposition d'un certain nombre de projets de recherche qui vont naturellement dans la continuité des travaux que j'ai présentés dans ce manuscrit.

Sommaire

9.1	Bilan des contributions	274
9.2	Poursuite des travaux de recherche	279
9.3	Vers la réalisation d'un simulateur de ponction	284
9.4	Vers l'usage de l'Intelligence Artificielle ?	286
9.5	Bilan de la rédaction de mon HDR	287

9.1 Bilan des contributions

Contexte médical. La formation médicale directement réalisée sur le patient pose des problèmes éthiques voire médicaux-légaux en confrontant le patient à des risques directement liés à la formation. Dans ce sens, la Haute Autorité de Santé (HAS) préconise de ne plus réaliser de gestes médicaux-chirurgicaux la première fois sur le patient. Il s'agit alors de faire évoluer la formation médicale en mettant à la disposition des médecins de nouveaux outils d'apprentissage permettant un entraînement sans risque pour le patient, mais également de les limiter lorsque les médecins non experts effectuent leurs actes la première fois sur le patient. Une des solutions consiste à introduire l'emploi de simulateurs bénéficiant des avancées en nouvelles technologies dans l'apprentissage des gestes des médecins. Ces simulateurs doivent assurer un apprentissage du geste médical et de sa compréhension, tout en garantissant un passage naturel vers le patient, c'est-à-dire qu'il ne s'agit pas d'apprendre à utiliser le simulateur mais bien apprendre à effectuer un geste médical et le comprendre pour se l'approprier et savoir le restituer en situation réelle.

Cette nouvelle génération de simulateurs médicaux peut se baser sur la puissance des machines actuelles en concevant des simulateurs couplant un dispositif physique à une simulation numérique. Le rôle du dispositif physique est de restituer les sensations perçues lors du geste à l'apprentis, tandis que celui de la simulation numérique est de reproduire le comportement des organes en interaction.

L'objectif est alors double pour la simulation. Il s'agit d'une part de fournir au dispositif physique les informations dont il a besoin pour restituer les sensations perçues lors du contact de l'apprentis avec les organes, et d'intégrer le mouvement de l'apprentis dans le calcul de la simulation numérique. Et d'autre part, il s'agit de permettre une visualisation du comportement des organes en interaction pour une meilleure compréhension de la part de l'apprentis. En effet, un des avantages de la simulation numérique réside dans le fait de pouvoir visualiser et analyser des mécanismes généralement peu visibles lors d'une intervention réelle, et de cibler les situations pertinentes pour l'apprentissage.

Problématique. Pour réaliser cette simulation du comportement des organes, nous avons vu que le modèle géométrique des organes est issu des données d'imagerie médicales afin de construire un modèle fidèle à la réalité. Reste ensuite à reproduire à l'aide de modèles physiques, leurs mouvements et déformations issus de leurs interactions entre eux et de leurs contacts avec des outils chirurgicaux. La complexité de ces simulations provient ainsi de la modélisation des organes en un plus ou moins grand nombre d'éléments, des lois de comportements associées à ces différents organes, et des méthodes numériques employées lors de la simulation. L'enjeu réside alors dans la gestion du compromis entre la précision des résultats obtenus et le temps d'exécution nécessaire aux calculs de cette simulation afin d'assurer l'interactivité requise pour son couplage à un dispositif physique.

Deux approches prédominent dans le cadre de la modélisation physique des objets déformables. Une approche continue basée sur la mécanique des milieux continus permet une simulation précise avec la résolution des équations différentielles décrivant le mouvement des objets simulés. Cette approche continue intègre ainsi directement les caractéristiques physiques des objets considérés tels que le module de Young ou le coefficient de Poisson. Par contre, cette approche nécessite des temps de calculs importants surtout lorsque de la non-linéarité est requise dans la formulation du comportement de l'objet. C'est pourquoi dans le domaine de l'animation, une alternative a été proposée basée sur une approche discrète. Cette approche appelée système masses-ressorts, permet ainsi l'allégement des

calculs effectués durant la simulation, grâce à l'utilisation de modèles simplifiés par rapport aux équations directement issues de la mécanique des milieux continus. Par contre, cette diminution du temps de calculs se fait au détriment de la précision de la simulation avec également la difficulté d'établir un lien direct entre les paramètres mécaniques de l'objet et ceux du modèle physique employé pour sa simulation.

Méthodologie et contributions. J'ai présenté dans ce manuscrit les travaux de recherche que j'ai réalisés dans le domaine de la modélisation basée physique d'objets déformables lors notamment de co-encadrements d'étudiants en doctorat.

Mon apport réside essentiellement dans la proposition de solutions permettant de gérer le compromis entre le temps et la précision des calculs de la simulation numérique. Ceci s'est traduit par la mise en place de modèles permettant de gérer une modélisation complexe des organes en interaction avec l'idée de pouvoir ajuster dynamiquement leurs définitions en fonction des besoins de la simulation. Cet ajustement s'effectue en fonction de critères relatifs au comportement des objets au cours de leur simulation.

Pour permettre l'obtention de simulations en temps interactif quelque soit le type d'approche adoptée (continue ou discrète), plusieurs pistes ont été suivies au fil des années.

La définition ou l'amélioration de différents modèles physiques.

- Durant le doctorat de Xavier Faure, nous avons étendu l'utilisation du modèle masses-tenseurs en proposant un environnement permettant la génération automatique des équations définissant les forces et leurs dérivées appliquées en chacun des nœuds des éléments constituant un objet. Cette génération automatique par l'emploi du calcul formel effectuée en fonction de la loi de comportement et du type des éléments, permet l'utilisation de lois de comportement complexes.
- Durant le doctorat de Elsa Fléchon, le modèle masses-ressorts a été intégré au sein d'une structure de données topologique avec l'emploi de la formulation de Vincent Baudet pour relier les raideurs des ressorts aux paramètres mécaniques des objets (module de Young et coefficient de Poisson).
- Durant le doctorat de Karolina Golec, nous avons proposé une nouvelle formulation du modèle discret des masses-ressorts en extension au travail de doctorat de Vincent Baudet. De la non-linéarité a ainsi été ajoutée dans la formulation des forces exercées par les ressorts, ainsi que des forces de corrections et une nouvelle configuration des ressorts. Ces améliorations permettent au final la simulation tissus mous isotropes non-linéaires définis par leur modèle de Young et coefficient de Poisson.
- Durant le doctorat de Mathieu Bailet, une formulation des éléments finis coques a été proposée en intégrant de la flexion dans la définition des membranes. Elle permet de modéliser des objets incompressibles en couplant cette formulation à une méthode de préservation de volume. L'idée est d'employer un modèle surfacique qui est moins coûteux en terme de calculs qu'un modèle volumique.

L'obtention d'une modélisation permettant de gérer des objets mixtes.

- Durant le doctorat de Xavier Faure, l'approche masse-tenseurs permettant une formulation élément par élément des forces appliquées aux nœuds et de leurs dérivées a été employée pour mixer au sein d'un objet des éléments de type différent en géométrie et loi de comportement.

- Durant le doctorat de Elsa Fléchon, un modèle unifié a été mis en place combinant topologie et physique. Ce modèle permet de gérer des objets mixant des éléments de types différents aussi bien en topologie qu'en propriétés mécaniques. Il permet ainsi d'intégrer au sein d'un même environnement, les différents modèles employés au fil des années (système masses-ressorts, modèle masse-tenseur avec différentes lois de comportement, et à moyen terme éléments finis volumiques et surfaciques avec différentes lois de comportement) définis sur différentes topologies.

L'adaptation dynamique de la modélisation des objets.

- Il s'agit de changer au cours de la simulation les propriétés mécaniques ou topologiques des éléments constituant les objets. L'idée est de pouvoir modifier ces paramètres pour les ajuster en fonction du compromis global entre le temps de calcul de la simulation et la précision des résultats obtenus. Ces changements s'effectuent à la demande ou en fonction de critères. Au niveau de la topologie, l'idée est de pouvoir simuler une découpe des objets ou encore de pouvoir raffiner le modèle géométrique afin de concentrer les calculs aux endroits pertinents de l'objet. Au niveau mécanique, l'idée est de pouvoir choisir le modèle physique ou la loi de comportement adéquate.
- Durant le doctorat de Xavier Faure, des techniques d'estimation de l'erreur ont été employées *a posteriori* pour définir les zones du maillage subissant les plus fortes contraintes et nécessitant un changement du type des éléments.
- Au final, le modèle unifié mis en place à la fin du doctorat de Elsa Fléchon permet d'utiliser les résultats obtenus par Xavier Faure en modifiant de manière aisée les éléments au niveau de leur topologie et de leurs propriétés mécaniques.

La parallélisation des algorithmes de simulation.

- L'approche généralement adoptée est basée sur un découpage des données avec la création de tâches de calculs traitant ces fragments de données. La taille de ces blocs de données définit alors la granularité de l'algorithme parallèle.
- Durant le doctorat de Xavier Faure, une parallélisation GPU basée sur l'utilisation du langage de programmation OpenCL a été intégrée au sein de la librairie SOFA permettant un code plus générique que la parallélisation CUDA déjà proposée.
- Durant le doctorat de Mathieu Bailet, une parallélisation GPU a été proposée avec un partitionnement des données basé sur une méthode de coloriage de graphe permettant d'optimiser la gestion de la mémoire en évitant les écritures concurrentes.

Développement logiciel. Nos travaux actuels s'attellent à fusionner ces contributions. Nous sommes ainsi en train de développer un environnement qui permet d'effectuer la simulation d'objets dits mixtes, c'est-à-dire qui comportent des éléments de types différents en terme de topologie et propriétés physiques. L'aspect important de cet environnement concerne la définition générique des éléments constituant les objets simulés. D'un point de vue technique, il s'agit notamment de fusionner les deux librairies NExT et TopoSim qui ont été développées durant les travaux de recherche que j'ai mentionnés dans ce manuscrit.

- A terme, tout modèle physique peut être intégré au sein de cet environnement. Nous avons déjà intégré le modèle physique des masses-ressorts, celui des masse-tenseurs avec les lois de comportement de Hooke et de Saint Venant Kirchhoff, et les éléments

finis pleins et surfaciques sont en cours d'intégration.

- Toute topologie de cellules peut être employée pour décrire les éléments des objets. Il faut bien entendu pouvoir y associer les informations nécessaires aux calculs de la physique, comme par exemple les fonctions de forme employées dans le cadre de l'approche masse-tenseurs ou des éléments finis classiques.
- La boucle de simulation s'écrit ensuite de manière générique faisant appel aux fonctions respectives calculant les forces et leurs dérivées en chacun des nœuds constituant les objets. Les collisions entre les objets sont également prises en compte avec l'emploi de l'approche classique par pénalités.
- L'intérêt de l'emploi de ce modèle réside ensuite dans le fait de pouvoir ajuster au cours de la simulation la définition des objets en modifiant facilement la topologie et les propriétés mécaniques de leurs éléments. Il est alors possible de changer la loi de comportement de l'objet ou d'une partie de l'objet, ou encore le modèle physique employé, voire le schéma d'intégration numérique utilisé dans la boucle de simulation. Bien entendu cet ajustement n'est réalisable que si une formulation élément par élément est possible dans le modèle physique employé.
- Cet ajustement s'effectue en fonction de critères basés sur la géométrie ou le comportement mécanique de l'objet simulé. Nous pouvons ainsi débiter une simulation avec une loi de comportement linéaire qui est valide en petites déformations, et passer à une loi de comportement plus complexe dès que la limite de validité de la loi précédente est atteinte. Nous pouvons également modifier le modèle géométrique de l'objet en changeant le nombre de points de subdivision des éléments ou encore leur topologie en raffinant les zones nécessitant plus de détails. Il est également possible de supprimer des connexions entre les éléments afin de pouvoir simuler des découpes au sein des objets durant la simulation.
- Rappelons que l'environnement développé par Xavier Faure permet de générer la formulation des forces et de leurs dérivées en tout nœud d'un élément de loi de comportement et de topologie quelconque. Il suffit pour cela de définir la fonction de forme associée à cet élément. Cette approche par le calcul formel a été rendue possible par l'emploi du modèle masse-tenseurs qui permet une formulation élément par élément des forces appliquées aux nœuds, contrairement à l'approche classique des éléments finis nécessitant un assemblage des matrices pour l'ensemble des éléments. Nous pouvons ainsi tester au sein de notre environnement un grand nombre de lois de comportement dont la formulation a été ainsi générée au préalable par cet outil.

Application pour le domaine médical. Nous sommes en train de tester et valider ces approches en les mettant en place au sein de simulateurs pour l'apprentissage de gestes médicaux. Les thèses de Romain Buttin et Mathieu Bailet ont permis de débiter la conception d'un simulateur pour l'usage des forceps lors d'un accouchement. Ce développement s'est poursuivi grâce au travail de Matthieu Coquet dans le projet SAGA financé par l'appel de l'ANR MN 2012. Et depuis octobre 2016, la thèse de Charles Barnouin s'intéresse au développement d'un simulateur de ponction sous échographie (détaillé par la suite).

- Dans le cadre du simulateur des gestes de l'accouchement, le doctorat de Romain Buttin a mis en évidence les organes et les mécanismes liés à la descente du fœtus au cours de l'accouchement. Elle a également permis de vérifier la faisabilité d'une simulation en temps réel de l'accouchement. La difficulté résidait dans l'obtention d'une simulation temps réel dont le réalisme global était conservé, c'est-à-dire que malgré la simplification de la modélisation des organes et de leurs lois de comportement, le

calcul de la trajectoire devait être réaliste et suffisamment précise du point de vue des besoins d'un simulateur d'apprentissage de gestes.

- Durant le doctorat de Mathieu Bailet, une simulation réaliste des déformations de la tête du fœtus a été effectuée. Celle-ci est basée sur une modélisation de la tête fœtale mettant en évidence les différentes zones du crâne telles que les fontanelles et les sutures qui ont des comportements mécaniques différents. Le modèle surfacique associé à la contrainte de volume a été utilisé pour simuler le comportement de la tête fœtale soumise à la pression utérine.
- Durant le travail d'ingénieur de Matthieu Coquet, la simulation complète de l'accouchement a été reproduite. Cette simulation est basée sur l'emploi du modèle surfacique associé à la contrainte de volume pour le comportement de la tête fœtale, et sur l'emploi d'un modèle d'objets rigides articulés pour le mouvement du bassin osseux de la parturiente ainsi que pour la flexion de la tête fœtale reliée au reste du corps du fœtus. Des forces périodiques reproduisant les contractions utérines sont ensuite appliquées afin de permettre la descente du fœtus.
- Le couplage a ensuite été réalisé entre la simulation numérique et le dispositif physique (appelé BirthSIM) développé par le laboratoire Ampère. La librairie VRPN (Virtual Reality Peripheral Network) a été employée pour effectuer cette connexion, puisqu'elle permet de relier facilement tout dispositif de réalité virtuelle à une application. La connexion entre la simulation et BirthSIM est effectuée par le bus de communication RS-232 plus communément appelé "port série".
- A noter que dans le cadre du simulateur de ponction, le doctorat de Charles Barnouin a bénéficié des avancées faites lors du projet SAGA et du développement de la librairie NExT du point de vue des modèles physiques et de son interaction avec des interfaces haptiques. Ainsi, ce travail de recherche s'est concentré sur le calcul de l'image échographique en temps réel de la scène restituant les déformations des tissus lors du geste médical avec l'emploi de deux bras haptiques reproduisant une sonde échographique et une aiguille.

9.2 Poursuite des travaux de recherche

Vers un modèle unifié adaptatif. Je commence par présenter des projets de recherche concernant l'amélioration des modèles physiques employés pour la simulation interactive d'objets déformables. Ces pistes de recherche font directement suite aux travaux que j'ai détaillés tout au long de ce manuscrit dont l'objectif concerne la gestion du compromis entre le temps de calcul de la simulation et la précision des résultats obtenus.

Structure de données topologique sur architecture parallèle.

Nous avons mis en place une structure de données décrivant les objets simulés en un ensemble de cellules définies par leurs propriétés topologiques et mécaniques. Dans notre quête de toujours aller plus vite tout en conservant une précision adéquate, l'approche du parallélisme reste à étudier au niveau de cette structure de données, puisqu'il s'agit d'effectuer les mêmes traitements en mobilisant plusieurs processeurs/cœurs ainsi que la carte graphique pour aller plus vite.

Au moment où les travaux de recherche que j'ai mentionnés ont été effectués, le modèle topologique des cartes combinatoires que nous utilisons n'avait pas encore été implanté en parallèle. C'est pourquoi dans l'environnement actuel, les fonctions de calcul sont simplement exécutées en parallèle en employant la librairie C++ Intel®TBB (Threading Building Blocks) de parallélisme de tâches. Pour effectuer une implantation parallèle complète de l'ensemble de la simulation, il s'agit alors d'utiliser une version GPU et ou multi-CPU du modèle topologique LCC et des opérations topologiques qui lui sont associées. Ce travail a été initié par Guillaume [Damiand et al. \(2018\)](#) avec la proposition d'une version distribuée de ce modèle topologique.

Pour aller plus loin, nous débutons un travail entre Fabrice Jaillet, Guillaume Damiand, Vincent Vidal (LIRIS-ORIGAMI) et moi-même sur la parallélisation des algorithmes de raffinement de maillage. L'idée est de pouvoir bénéficier de toute la puissance de calculs fournie par les machines actuelles pour nos simulations numériques.

Gestion de plusieurs représentations d'un même objet.

En informatique graphique, il est usuel d'employer plusieurs représentations pour un même objet de la scène. Ainsi pour effectuer le rendu réaliste de l'objet et garantir une représentation lisse de sa surface, un modèle géométrique possédant un grand nombre de nœuds est employé, alors qu'une seconde représentation ayant moins de nœuds est utilisée pour effectuer les calculs de la simulation, et qu'une troisième représentation de l'objet (par exemple une hiérarchie de boîtes englobantes) permet de gérer les interactions de l'objet avec les autres objets.

Il s'agit maintenant de garantir le lien entre ces différentes représentations afin de répercuter correctement au sein de celles-ci, les changements topologiques survenus au cours de la simulation (notamment dans le cadre de la découpe).

Plus précisément, il s'agira d'intégrer au sein de notre modèle unifié le fait d'employer ces différentes représentations sans forcément avoir à les créer, afin de limiter les coûts en terme de mémoire et rester dans la continuité de notre souhait de ne gérer qu'une seule structure de données pour un même objet de la scène.

Adaptation dynamique de l'objet simulé en fonction de critères.

La structure de données que nous avons mise en place permet la simulation d'objets de cellules de types différents en terme de topologie et de propriétés mécaniques.

- L'étape suivante concerne la gestion efficace des frontières entre les cellules de type différent, ainsi que la gestion de leurs modifications au cours de la simulation. Il s'agit notamment d'assurer une continuité entre ces différents types de cellules.
- Lors des modifications effectuées au cours de la simulation, l'enjeu réside dans la préservation de la stabilité numérique des méthodes numériques employées, en effectuant un changement progressif de ces cellules aussi bien au niveau de leur topologie que de la physique associée. Il faudra également assurer la validité physique d'un objet mixant des cellules de type différent permettant de modéliser des organes dont le comportement n'est pas homogène.
- Au niveau topologique, la suite réside dans la proposition de schémas de raffinement permettant d'exploiter les avantages de l'aspect générique de notre modèle par l'emploi de schémas de subdivision basés sur des cellules quelconques. Nous pourrions ainsi sortir des schémas usuellement employés qui divisent de manière régulière les cellules en créant des cellules de même topologie. Nous pourrions notamment nous baser sur les motifs de subdivision mis en place par Claudio Lobos (2013); Lobos and González (2015) en employant au sein d'un même objet des hexaèdres, des tétraèdres, des prismes ou encore des pyramides. Il s'agira par contre de savoir gérer le passage d'un type topologique d'élément à un autre en limitant les perturbations engendrées dans la simulation numérique. A noter que la mise en place de ces schémas de subdivision permettra d'effectuer une découpe des objets de la scène en suivant n'importe quel chemin, et non pas effectuer des coupes uniquement le long des arêtes. Comme mentionné précédemment, il s'agira également de paralléliser l'ensemble des algorithmes en employant une structure de données adaptée afin de gagner en temps de calculs et permettre ces changements topologiques au cours de la simulation.
- Au niveau mécanique, il s'agit de proposer une modélisation adéquate en fonction de la topologie des cellules. Si nous employons la méthode des éléments finis comme modèle mécanique, il suffit d'explicitier certains paramètres (nombre de nœuds sur chaque élément, points d'intégration, loi de comportement, etc.) mais les schémas sont connus. Par contre, si nous souhaitons utiliser le modèle des masses-ressorts, il s'agira de spécifier le calcul des raideurs que nous avons pour le moment établi uniquement avec les hypothèses d'isotropie et de cellules cubiques.
- Un autre verrou scientifique concerne la définition de critères pour effectuer ces changements à bon escient. Pour le moment, nous avons employés des critères géométriques sur la qualité des éléments de l'objet et des critères mécaniques en calculant les contraintes appliquées sur les éléments. Il s'agira de mettre en place de nouveaux critères pertinents dans le cadre de la simulation dynamique d'objets déformables.
- Une fois ces différents verrous scientifiques et techniques levés, la modélisation des objets pourra ainsi s'adapter au fil de l'exécution en fonction des interactions entre les objets et des besoins requis pour un temps d'exécution interactif. L'idée est en effet d'adapter le maillage volumique représentant l'objet selon les besoins. Il s'agira ainsi de raffiner les parties dont plus de précision est requise (comme les zones de contact) tout en laissant des zones plus grossières si la précision n'est pas requise à ces endroits et si ces zones plus grossières n'engendrent pas d'erreur pour le reste de la simulation.

- Pour arriver à un maillage optimal en fonction des besoins, un premier maillage adapté pourra être généré selon les techniques de calcul de l'erreur *a posteriori*, c'est-à-dire que la simulation pourra être lancée avec différents niveaux de résolution pour le maillage volumique et l'idée est ensuite de mixer les zones de différents niveaux de résolution pour créer un maillage optimal pour la simulation visée. En effet, pour un même objet à simuler, le maillage optimal peut être complètement différent si nous considérons l'objet isolé ou le même objet en collision avec d'autres objets de la scène, d'autant plus si nous considérons un cas comme pour la simulation de l'accouchement où les objets sont en contact continu les uns par rapport aux autres. Fabrice Jaillot et Vincent Vidal travaillent déjà sur cette problématique, et donc l'idée à court terme est d'intégrer leurs travaux au sein de la plateforme que nous sommes en train de mettre au point.

Validation des modèles proposés.

La question de la validation est toujours un moment difficile mais pourtant nécessaire pour valider l'ensemble de notre approche. Il s'agira d'effectuer un certain nombre d'expérimentations pour évaluer notre approche, aussi bien au niveau des modèles physiques proposés (modèle masse-ressort étendu, modèle masse-tenseur avec différentes lois de comportement, modèle surfacique basée sur les coques, etc.), qu'au niveau du choix de la topologie des éléments définissant le maillage volumique de l'objet simulé. Des expérimentations simples pourront débiter en effectuant la simulation sur différents maillages volumiques uniformes à différentes résolutions en utilisant le même modèle physique. Ensuite, la difficulté réside dans le choix optimal lors de l'adaptation des éléments aussi bien au niveau de leur topologie que de leur modèle physique et loi de comportement. Nous pouvons d'ores et déjà penser que nous ne pourrons pas comparer tous les cas possibles, mais qu'il s'agira plutôt de mettre en place des heuristiques pour savoir *a posteriori* ou au cours de la simulation, s'il vaut mieux changer dans une zone du maillage la loi de comportement et le modèle physique ou plutôt la topologie des éléments. A noter que des travaux sur l'élaboration de *benchmarks* pertinents sur cette problématique ont été initiés à Strasbourg par Stéphane Cotin (ICube, INRIA). L'idée est alors de comparer nos résultats en intégrant ces tests au sein de la plateforme CamiTK (Computer Assisted Medical Intervention Tool Kit) développée par le laboratoire TIMC-IMAG.

Vers la réalisation d'un simulateur d'accouchement. Au niveau des travaux sur la réalisation du simulateur de l'apprentissage des gestes de l'accouchement, le projet se poursuit et des points de recherche applicative sont encore à aborder.

Pertinence des choix de modélisation et de simulation.

- Afin de s'assurer du comportement global de la simulation et gérer au mieux le compromis entre la précision des résultats et les temps de calculs, une analyse doit être effectuée afin de mesurer l'impact qu'engendrent les différentes simplifications effectuées sur les modèles géométriques et biomécaniques des différents organes.
- Pour la modélisation géométrique, il s'agit de comparer les résultats de la simulation obtenus avec les maillages de grande résolution (directement issus de la segmentation des données médicales) avec ceux obtenus en employant des maillages à différents niveaux de simplification.

- Pour la modélisation biomécanique, l'idée serait de partir d'un modèle biomécanique idéal (basé par exemple sur celui du Laboratoire de Mécanique de Lille et du CHRU de Lille issu des travaux de Mathias Brieu et du Pr Michel Cosson - [Lepage et al. \(2014\)](#); [Rubod et al. \(2013\)](#)) et de le simplifier progressivement jusqu'à l'obtention d'un modèle adapté pour le temps interactif. En effet, dans la modélisation proposée par Romain Buttin, seules les lois de Hooke et de Néo-Hooke ont été employées avec des valeurs des paramètres mécaniques qui ne sont pas toujours les mieux adaptées. Il faut donc réussir à déterminer pour chacun des organes impliqués dans la simulation, la loi de comportement adéquate en accord avec la précision et le temps d'exécution souhaitée, et évaluer l'impact de ces différents choix sur la précision de la trajectoire fœtale au cours de l'accouchement. Dans ce sens dans le cadre du projet SAGA, une étude est en train d'être effectuée sur la comparaison de différents modèles physiques employés pour modéliser les organes. Cette étude concerne les éléments finis volumiques (classiquement employés en mécanique), les éléments finis surfaciques associés à l'emploi d'une contrainte de volume (modèle mis au point durant le doctorat de Mathieu Bailet), la méthode HEML (Hyper-Elastic Mass Links) (proposée par notre partenaire François [Goulette and Chen \(2015\)](#)) et le système discret des masses-ressorts. Il s'agit de comparer le pourcentage de déformation obtenus sur les éléments des maillages par rapport aux mesures disponibles dans la littérature et de les valider par rapport aux connaissances médicales.
- Une étude doit également être effectuée sur le choix de la méthode d'intégration et son pas de temps, sur la convergence des différentes méthodes numériques employées (notamment lors de la résolution des systèmes linéaires ou non-linéaires) ainsi que sur le traitement des collisions. Actuellement, le choix s'était porté sur le schéma d'intégration explicite d'Euler et sur la méthode des pénalités pour le traitement des collisions dont les contacts sont considérés sans frottement.
- Pour aller plus loin sur le réalisme de la simulation, l'idée est de compléter la modélisation de la déformation de la tête en modélisant les bosses séro-sanguines qui peuvent apparaître si une force trop importante est appliquée sur la tête par l'emploi des forceps. Ceci permet de mettre en évidence les risques liés à une mauvaise maîtrise du geste médical.

Influence du mouvement de la parturiente.

Dans le travail de Romain Buttin et dans le projet SAGA, nous n'avons pas pris en compte les mouvements potentiels de la parturiente puisque nous avons imposé des conditions limites basées sur une position de la parturiente allongée sur le dos. Or la parturiente peut être amenée à bouger afin de faire basculer son bassin, et ainsi changer la trajectoire du fœtus. Il serait ainsi intéressant de rajouter cette possibilité, permettant dans le cadre d'un simulateur d'apprentissage des gestes de l'accouchement, de comprendre les répercussions de telle ou telle position pour l'accouchement. Pour cela, le squelette articulaire de la parturiente doit être modélisé.

Validation de la simulation numérique.

La validation de la simulation se fera en partie sur les scénarios mis en avant par l'observation du terrain : selon les paramètres en entrée, il faudra tester la simulation pour voir si elle arrive à reproduire le déroulement statistiquement attendu de l'accouchement.

Par exemple, un blocage de la tête si le bassin est trop étroit ou encore une intervention à effectuer de la part de l'obstétricien pour extraire le fœtus avec des forceps si les forces de contractions utérines et de poussées maternelles ne sont pas assez importantes pour dégager le fœtus.

Vers du diagnostic et des simulateurs “patient spécifique” ?

Dans le travail de doctorat de Romain Buttin autour de l'accouchement, nous passons bien entendu sous silence le temps conséquent qui a été passé à analyser les données médicales pour permettre l'obtention de maillages surfaciques et volumiques adéquates pour la simulation numérique ; le recalage qui a été fait entre un fœtus récupéré sur une IRM et le bassin récupéré sur le scanner d'une autre parturiente ; les maillages terminés “à la main” sous Blender pour les zones non capturées sur les images médicales ou les parties difficiles à voir comme le col de l'utérus. Nous ne parlerons bien entendu pas du choix des caractéristiques mécaniques issues de la littérature ou dirons-nous des moyennes issues de la littérature, ces valeurs changeant considérablement d'une parturiente à l'autre voire de la mesure effectuée.

Nous avons vu que dans le cadre de la réalisation du simulateur pour l'apprentissage des gestes, notre critère de précision concerne le résultat global de la simulation. Ainsi quelques jeux de données géométriques et lois de comportements avec des valeurs des paramètres physiques adéquates suffisent. Mais il est déjà important de pouvoir traiter plusieurs jeux de données médicales pour pouvoir analyser l'impact sur la trajectoire suivie par le fœtus, de la taille du bassin de parturiente, ou encore de celle de la tête fœtale.

Par contre, si à plus ou moins long terme nous voulons réaliser des simulations de diagnostics pour savoir si un accouchement par voie basse est réalisable ou non, une certaine précision sera alors nécessaire. La question qui se posera alors est : “Comment allons-nous pouvoir faire du patient (parturiente) spécifique dans ce cas ?” car il s'agit alors de réussir à automatiser toute la chaîne de traitement des données médicales jusqu'à la génération des maillages volumiques adaptés pour la simulation numérique. Sans compter que pour être réellement patient (parturiente) spécifique, il faut également pouvoir mesurer les caractéristiques mécaniques des organes modélisés. Les verrous scientifiques et techniques permettant cette phase d'automatisation devront alors être levés.

Dans ce sens, nous souhaiterions rédiger un projet avec Fabrice Jaillet, Mathias Brieu (Laboratoire de Mécanique de Lille) et Emmanuel Promayon (laboratoire TIMC-IMAG) pour unir nos travaux sur la modélisation et la simulation du système pelvien et lever ces verrous scientifiques. Nous avons vu qu'à Lille, un modèle très complet du système pelvien a été conçu intégrant les ligaments et les propriétés mécaniques mesurées sur différentes patientes. Ce modèle numérique requière des heures de calculs mais est sans doute le modèle le plus complet existant dans le domaine. Il se base sur un modèle géométrique issu d'un modèle basé NURBS qui a été conçu à Lille pour s'adapter facilement à de nouvelles données médicales. L'idée est ensuite de réussir à modifier à la fois les modèles géométriques de grande résolution et le modèle mécanique complet, pour diminuer les temps de calcul tout en conservant un comportement réaliste du système pelvien.

9.3 Vers la réalisation d'un simulateur de ponction

Le geste de la ponction de grosses articulations a évolué ces dix dernières années avec l'utilisation d'une sonde échographique permettant de mieux guider l'insertion de l'aiguille durant l'opération. La principale difficulté dans la réalisation et dans l'apprentissage de ce geste réside ainsi dans la manipulation simultanée de deux instruments. Il faut en effet savoir manipuler la sonde échographique (pour visualiser le geste effectué sur un écran déporté) tout en insérant l'aiguille dans l'articulation. Ces deux instruments peuvent également être inter-changés durant la ponction, rendant l'opération d'autant plus délicate.

Dans ce contexte, nous avons débuté en octobre 2016 un projet nommé SPARTE (Simulator of Puncture for ARTiculations under Echography), financé par l'IDEFI-SAMSEI (Stratégies d'Apprentissage des Métiers de Santé en Environnement Immersif), en collaboration avec le laboratoire Ampère, le LIBM (Laboratoire Interuniversitaire de Biologie de la Motricité), et le service de rhumatologie de l'Hôpital Lyon Sud. L'objectif concerne la réalisation d'un **simulateur d'apprentissage du geste de la ponction de grosses articulations sous échographie** pour un apprentissage du geste sans risque pour le patient.

Le cahier des charges du simulateur a été établi par le LIBM par le biais d'une analyse en amont de la transposition de l'acte réel dans le simulateur. Une étude de métier a été réalisée pour analyser et comprendre l'apprentissage actuel du geste de la ponction afin de lister les éléments pertinents du simulateur. Différents scénarios adéquates pour l'apprentissage doivent également être proposés. Le simulateur pourra également permettre une évaluation du geste effectué dont la réussite est actuellement mal évaluée sur le terrain.

Dans la continuité du simulateur d'accouchement (réalisé en collaboration avec le laboratoire Ampère), ce simulateur de ponction est constitué de deux parties :

- Un dispositif haptique composé de deux bras à retour d'effort permettant à la fois de manipuler une sonde échographique et de restituer les sensations de toucher perçues par l'apprenant lors de l'insertion de l'aiguille dans l'articulation¹ ;
- Une simulation numérique permettant de suivre l'insertion de l'aiguille dans l'articulation avec un rendu visuel de type échographie. La simulation prendra naturellement en compte les mouvements de l'apprenti (c'est-à-dire la position et l'orientation de la sonde échographique et de l'aiguille qu'il manipule).

La réalisation de ce simulateur (basé à la fois sur une simulation numérique et sur un dispositif haptique) permettra ainsi de traiter indifféremment plusieurs types d'articulation et différentes pathologies. À terme, il pourra aussi permettre la recherche d'une solution optimale mécanique de conduite de l'aiguille de ponction. Il s'affranchira ainsi de l'utilisation de mannequins anthropomorphiques.

La validation du simulateur sera réalisée avec l'aide de notre partenaire médical (Dr Fabienne Coury, service de rhumatologie de l'Hôpital Lyon Sud). Il s'agira d'étudier les gestes réalisés par l'apprenti et d'étudier l'apport pédagogique de l'utilisation du simulateur par rapport à un apprentissage classique de ce geste.

1. La partie haptique est réalisée par le laboratoire Ampère par le biais d'un financement de thèse ministérielle débutant en octobre 2016 avec une soutenance au 7 mai 2020.

Développement de la partie simulation numérique. Pour mener à bien ce travail, le projet IDEFI-SAMSEI finance le doctorat de Charles Barnouin, encadré par Fabrice Jaillet et moi-même, depuis le mois d’octobre 2016 avec une soutenance au 7 mai 2020. Ce travail de doctorat concerne principalement la partie simulation numérique du simulateur qui doit permettre la restitution du comportement des tissus lors de l’insertion de l’aiguille. L’idée étant de permettre une visualisation de l’acte médical sous la forme d’un rendu de type échographique. Sa mise en œuvre nécessite la réalisation des éléments suivants :

- *Modélisation géométrique 3D de l’articulation* intégrant les différents tissus et comportements bio-mécaniques. Une première modélisation a été réalisée par le LBMC à partir d’images médicales (IRM, scanner) pour le cas de l’épaule. Ce modèle doit encore être complété par l’ajout d’éléments non aisément visibles comme les ligaments. A terme, il faudra également le faire pour d’autres articulations (genou, hanche, etc.) et pour différentes pathologies.
- Réalisation d’un *couplage entre corps rigides (os) et tissus mous* afin de simuler le comportement de l’articulation en interaction avec les tissus mous du voisinage.
- *Modélisation 3D de l’aiguille* en considérant différents types d’aiguille (*i.e.* taille et modèle variables). Pour le moment, nous avons considéré uniquement une aiguille rigide usuellement employée pour l’épaule.
- *Modélisation de l’insertion de l’aiguille* dans le modèle 3D de l’articulation couplée avec les tissus mous afin de simuler la déformation des tissus au cours de l’acte.
 - Nous sommes en train d’effectuer la simulation de l’insertion de l’aiguille dans les tissus de l’articulation de l’épaule. Cette simulation est basée sur les expérimentations réalisées par le LBMC mettant en avant les forces de rupture et de friction mises en jeu lors de l’insertion.
 - A terme, il faudra prendre en compte les caractéristiques mécaniques de l’aiguille (rigide, plus ou moins souple) selon la profondeur de la ponction, notamment dans le cas de la hanche qui nécessite une aiguille de taille plus grande.
- *Calcul de l’image ultra-sonore (US)* effectuée durant la simulation en fonction de la position et orientation de la sonde. Un premier travail a été réalisé par Charles Barnouin et al. (2018, 2020) permettant l’obtention d’un rendu échographique temps réel calculé à partir de la position de la sonde (correspondant à un bras haptique).
- *Couplage des bras robotisés et de la simulation numérique.* Un premier travail a été réalisé par Charles permettant de déterminer la position de l’aiguille et de la sonde échographique durant leur usage afin d’envoyer ces informations à la simulation.

Ce travail est réalisé à l’aide de la librairie de simulation physique NExT que nous avons développée durant le projet ANR SAGA en collaboration avec le laboratoire TIMC-IMAG. Rappelons que cette librairie propose un certain nombre de modèles physiques pour la simulation physique (éléments finis surfaciques, éléments finis volumiques, système masses-ressorts, systèmes articulés), ainsi que la possibilité de coupler la simulation à un dispositif haptique. Par l’emploi de cette librairie et les connaissances acquises durant ces dernières années, nous proposons une première version du simulateur numérique/haptique de ponction en fin de doctorat de Charles Barnouin.

Il s’agira ensuite d’aller plus loin dans la simulation en abordant les points suivants :

- Optimisation du calcul de la déformation avec la mise en place de nouveaux algorithmes génériques de simulation et collision sur architecture parallèle (type GPU).

- Optimisation du modèle : adaptation de la géométrie ou de l'espace du maillage le long de l'aiguille sans connaissance de la trajectoire a priori. Éventuellement, utilisation du modèle sans maillage pour éviter le remaillage
- Étude de l'apport de la réalité augmentée et des environnements immersifs dans les simulateurs destinés à l'apprentissage des gestes médicaux.
- Validation des résultats sur fantôme. Fantôme simple constitué de différents matériaux dont les caractéristiques sont connues couplées à des mesures des déformations subies lors de l'insertion de l'aiguille, et mesures des forces exercées en retour sur l'aiguille.
- Pré-interventionnel pour la ponction des articulations, avec proposition du meilleur point d'entrée et calcul de la meilleure trajectoire. En effet, il faut noter que le geste devient difficile dans le cas d'une aiguille souple.
- Extension "patient-spécifique" du simulateur nécessitant la génération automatique du modèle de l'articulation à partir de l'imagerie médicale.
- Extension à d'autres applications médicales similaires.

9.4 Vers l'usage de l'Intelligence Artificielle ?

Enfin, difficile de ne pas mentionner l'Intelligence Artificielle ou IA à l'heure où ce terme est employé tous les jours dans de nombreux médias ou lors de colloques. Son usage dans le domaine médical est déjà d'actualité notamment pour l'aide au diagnostic par exemple en ophtalmologie ou en dermatologie pour les cancers de la rétine ou de la peau.

Positionnons-nous dans la thématique de recherche concernant le développement de simulateurs d'apprentissage de gestes médicaux. L'Intelligence Artificielle peut être employée à différents niveaux :

- En amont de la simulation numérique, afin d'aider à concevoir les modèles géométriques des organes par l'amélioration des techniques de segmentation d'images.
- Au niveau du développement du composant didactique, l'IA peut être employée pour analyser l'environnement réel et ainsi aider à l'analyse des situations, du métier pour élaborer des scénarios pertinents pour l'apprentissage ou encore déterminer des trajectoires optimales du geste à réaliser avec une meilleure évaluation de celui-ci.
- Au niveau du composant haptique, l'IA peut être employée de la même façon pour analyser le métier et ainsi déterminer les mouvements que l'interface haptique doit être capable de reproduire.

Ensuite au niveau de la simulation numérique elle-même, l'usage de l'IA changerait la façon de concevoir les simulations puisqu'elles sont actuellement basées sur la résolution dynamique des équations de la mécanique. Notons que [Mendizabal et al. \(2020\)](#) proposent d'employer les techniques de *deep learning* pour l'apprentissage de la relation qu'il existe entre une force et la déformation qu'elle induit. L'apprentissage est réalisé pour différentes géométries de l'objet dans le cadre de la loi de comportement non-linéaire de Saint-Venant Kirchhoff. Cet apprentissage est réalisé à partir de simulations numériques basées sur la Méthode des Éléments finis en appliquant des forces aléatoires sur les nœuds de l'objet. Les données stockées correspondent ainsi à un couple force/déformation. Leur souhait est de réaliser des prédictions avec une grande précision en un temps record.

Au final, la limitation principale de cette approche réside dans le fait qu'il n'est pas possible de restituer avec précision la déformation induite par une force qui n'est pas

présente dans les données initiales de l'apprentissage surtout dans le cas où plusieurs objets sont en interaction. De la même manière, nous sommes limités par la géométrie employée lors de l'apprentissage. Ainsi, si nous souhaitons employer ce type d'approche dans le cadre d'un simulateur de gestes, où l'objectif est de pouvoir interagir avec la simulation numérique, il est nécessaire de réussir à prendre en compte tous les mouvements possibles de l'utilisateur. Reste à évaluer si le coût et les contraintes liés à un tel apprentissage permettent ensuite une simulation précise et compatible avec les contraintes d'un simulateur d'apprentissage. Par contre, bien entendu, l'usage de l'IA est intéressant pour trouver en amont de la simulation interactive, les différentes valeurs adéquates des paramètres présents dans une simulation (paramètres mécaniques, mathématiques, géométriques, etc.).

9.5 Bilan de la rédaction de mon HDR

Pour finir, un dernier bilan plus personnel suite à la rédaction de ce manuscrit. Nous nous apercevons au quotidien de l'étendue des connaissances que nous devons acquérir pour avancer dans le domaine de la simulation d'objets déformables. En tant que Docteur en Informatique, j'ai appris par moi-même les notions de Mécanique des Milieux Continus et nous nous apercevons chaque jour que nous n'en connaissons qu'une infime partie suite à la lecture des articles de recherche. Et le temps nécessaire pour acquérir ces connaissances et surtout savoir lesquelles acquérir est immense et lié à une personne. C'est-à-dire qu'il est à refaire pour toute personne qui démarre dans cette thématique de recherche. Et sans ces connaissances, c'est difficile d'innover et d'avoir la réflexion nécessaire et suffisante à cela. Les astuces ou analyses qui paraissent logiques et instinctives pour une personne du domaine de la mécanique, ne le sont pas pour une personne du domaine des sciences de l'informatique, car elle n'en a tout simplement pas connaissance.

Comment alors progresser rapidement dans ce domaine si nous ne réunissons pas toutes ces connaissances au sein de projets ou de laboratoires communs ? Que faut-il mettre en place pour innover plus rapidement dans nos laboratoires ? Le temps pour l'acquisition des connaissances est-il ainsi incompressible ? Et si une personne travaillant sur cette thématique de recherche depuis de nombreuses années part, tout son savoir, ses réflexions et les idées associées disparaissent alors avec elle ? C'est pourquoi, la transmission du savoir auprès de nos étudiants est une mission importante. En effet, le savoir de base peut se trouver dans les livres mais il y restera si nous ne le sortons pas, et la réflexion suite à ce savoir avec une transposition dans une thématique de recherche particulière ne se trouve pas forcément dans les livres. Encore faut-il réussir à faire comprendre cela aux étudiants, qu'il faut apprendre les notions de base pour ensuite aller bien plus loin, même si des logiciels font déjà tous ces calculs pour nous... Et quel avenir pour ce savoir, si nous n'avons plus d'étudiants qui suivent nos cursus en recherche ?

On nous demande des résultats rapidement et même de plus en plus rapidement. Je suis arrivée au LIRIS en 2005. Les premières années au niveau recherche ont été floues, entre la fin des recherches précédentes et la recherche de sa nouvelle voie au sein de la structure. Au bout de 15 ans (ce qui est long à mon sens), les travaux que je mène avec Fabrice Jaillet et Guillaume Damiand commencent à porter leurs fruits. Mais nos moyens et le temps que nous pouvons y consacrer sont dérisoires et sans doute ridicules face aux moyens que nous trouvons dans les universités d'autres pays ou face à des entreprises telles que Dassault Système dont les thématiques (que nous voyons au travers des sujets de stages de nos étudiants de Master) sont souvent proches des nôtres. Et ce manque de

temps est très frustrant quand nous ne pouvons pas aider concrètement les étudiants de doctorat ou les ingénieurs sur nos projets en promettant de développer avec eux pour aller plus vite et que nous n'y arrivons pas, alors que c'est notre seul souhait de pouvoir le faire. Nous perdons ainsi du temps et de manière personnelle j'ai l'impression de ne pas remplir totalement mon rôle et ma mission. Je pense ainsi à Elsa Fléchon avec qui j'aurais aimé aller plus loin et à Matthieu Coquet qui a travaillé sur le projet SAGA mais de manière trop autonome à mon sens car je n'ai pu lui consacrer plus de temps.

Durant mon année de délégation, j'ai pu pendant un trop bref temps, reprendre avec Guillaume Damiaud le développement effectué par Elsa Fléchon. Quelle efficacité quand deux chercheurs avec de l'expérience travaillent ensemble (sans parler des grandes qualités de Guillaume dans ce domaine). Pourquoi ceci n'est-il pas possible au quotidien sans avoir à renoncer à sa vie personnelle ? Cela crée une schizophrénie en nous, entre devoir renoncer à avancer vite en recherche ou devoir consacrer nos soirées, week-end et vacances à cela au détriment de notre vie personnelle. Pourquoi nous ne nous sentons pas bien quand nous ne travaillons pas le week-end ? Voire même, nous avons honte de le dire, alors que c'est sensé être un fait normal. Et pourquoi nous n'arrivons pas à avancer en recherche dans notre temps normal d'enseignant-chercheur ? Par ailleurs, la recherche en France avancerait beaucoup plus rapidement si les chercheurs expérimentés pouvaient avancer par eux-mêmes et aider plus concrètement les jeunes chercheurs au quotidien. Sans parler du temps consacré à rédiger des projets de recherche qui ne passent pas toujours...

Par ailleurs, une année sans enseignement, c'est merveilleux pour prendre du recul mais pas assez long pour avancer surtout si nous nous mettons en tête de rédiger une Habilitation à Diriger les Recherches. D'ailleurs, la prise de recul est bien plus longue que la rédaction, mais ô combien bénéfique. Et cet arrêt dans les enseignements est au final bénéfique pour les étudiants que nous retrouvons avec plaisir et fraîcheur l'année suivante pour leur transmettre notre savoir. Mais pourquoi n'est-il possible de faire cela que tous les 6 ans ?

Au final, cela fait plusieurs années que j'ai débuté ce manuscrit. J'ai ainsi passé beaucoup de temps pour le rédiger et surtout comprendre de manière approfondie chacun des éléments des doctorats que j'ai encadrés. Certains diront peut être que j'ai rédigé un manuscrit de doctorat. Mais pour moi ce travail était nécessaire pour aller de l'avant et avancer dans cette thématique de recherche bien complexe, mais ô combien enrichissante.

Bibliographie

- C. Barnouin, F. Zara, and F. Jaillet. Real-time Ultrasound Rendering for Ultrasound-Guided Puncture Training. In *Journées Françaises d'Informatique Graphique (JFIG 2018)*, Poitiers, France, Nov. 2018. URL <https://hal.archives-ouvertes.fr/hal-01929043>.
- C. Barnouin, F. Zara, and F. Jaillet. A real-time ultrasound rendering with model-based tissue deformation for needle insertion. In *15th International Conference on Computer Graphics Theory and Applications, GRAPP 2020*, Valletta, Malta, Feb. 2020. URL <https://hal.archives-ouvertes.fr/hal-02415740>.
- G. Damiand, A. Gonzalez-Lorenzo, F. Zara, and F. Dupont. Distributed combinatorial maps for parallel mesh processing. *Algorithms*, 11(7), August 2018. URL <https://doi.org/10.3390/a11070105>.
- F. Goulette and Z.-W. Chen. Fast computation of soft tissue deformations in real-time simulation with hyper-elastic mass links. *Computer Methods in Applied Mechanics and Engineering*, 295 :18 – 38, 2015. ISSN 0045-7825. doi : <http://dx.doi.org/10.1016/j.cma.2015.06.015>. URL <http://www.sciencedirect.com/science/article/pii/S0045782515002078>.
- J. Lepage, , C. Jayyosi, P. Lecomte-Grosbras, M. Brieu, C. Duriez, M. Cosson, and C. Rubod. Biomechanical pregnant pelvic system model and numerical simulation of childbirth : impact of delivery on the uterosacral ligaments, preliminary results. *International urogynecology journal*, 26(4) :497–504, 2014.
- C. Lobos. A set of mixed-elements patterns for domain boundary approximation in hexahedral meshes. *Studies in health technology and informatics*, 184 :268–272, 2013. doi : 10.3233/978-1-61499-209-7-268. Proceedings of MMVR20.
- C. Lobos and E. González. Mixed-element octree : a meshing technique towards fast and real-time simulations in biomedical applications. *International Journal for Numerical Methods in Biomedical Engineering*, 2015. doi : 10.1002/cnm.2725.
- A. Mendizabal, P. Márquez-Neila, and S. Cotin. Simulation of hyperelastic materials in real-time using deep learning. *Medical Image Analysis*, 59 :101569, 2020. ISSN 1361-8415. doi : <https://doi.org/10.1016/j.media.2019.101569>. URL <http://www.sciencedirect.com/science/article/pii/S1361841519301094>.
- C. Rubod, P. Lecomte-Grosbras, M. Brieu, G. Giraudet, N. Betrouni, and M. Cosson. 3d simulation of pelvic system numerical simulation for a better understanding of the

contribution of the uterine ligaments. *International urogynecology journal*, 24(12) : 2093–2098, 2013.

Partie III

Annexe

Cette annexe se décompose en deux chapitres.

- L'annexe **A** présente mon rapport d'activités en tant que maître de conférences depuis septembre 2005 au sein de l'Université Lyon 1.
- L'annexe **B** présente la liste de mes publications scientifiques obtenues durant mes années de recherche.

Annexe **A**

Rapport d'activités

Après une brève présentation de mon curriculum vitæ, je présente dans ce chapitre mes activités effectuées depuis le mois de septembre 2005 en tant que maître de conférences à l'université Lyon 1 avec une activité de recherche effectuée au sein du LIRIS. Je détaillerai mes responsabilités à l'Université Lyon 1 et au sein du LIRIS, les projets de recherche dans lesquels se sont effectuées mes recherches, les collaborations extérieures que j'ai tissées au fil des années, mon activité en animation scientifique, les charges d'enseignements effectués à Lyon 1 et enfin les encadrements que j'ai effectués liés à la recherche.

Etat civil et coordonnées

Identité	Florence Zara
Née le	22 octobre 1977 à Grenoble (38)
Situation	3 enfants (nés le 3 mars 2009, le 19 octobre 2011 et le 21 février 2018)
Adresses	Domaine scientifique de la Doua - Bâtiment Nautibus 23-25 Av. Pierre de Coubertin - 69100 Villeurbanne Cedex
Section	Informatique - CNU 27 - CNRS 7
Enseignement	Université Lyon 1, Faculté des Sciences et Technologies (FST), Département d'Informatique
Recherche	Laboratoire d'InfoRmatique en Images et Systèmes d'information (LIRIS), équipe ORIGAMI, UMR5205
Email	florence.zara@liris.cnrs.fr
URL	http://liris.cnrs.fr/florence.zara

Formation universitaire

1999-2000	DEA de l'Université Joseph Fourier de Grenoble, spécialité "Informatique : Systèmes et Communications". Laboratoire de Modélisation et de Calcul de Grenoble, équipe APACHE. Sujet : Serveur interactif de cartes géographiques parallélisé sur grappe de processeurs.
2000-2003	Doctorat de l'Institut National Polytechnique de Grenoble (INPG), spécialité "Informatique : Systèmes et Logiciels", soutenu le 01/12/2003. Laboratoire ID-IMAG (Informatique et Distribution), projet INRIA-APACHE, en collaboration avec le laboratoire GRAVIR-IMAG (GRAphisme, VIsion et Robotique), équipe iMAGIS. Sujet : Algorithmes parallèles de simulation physique pour la synthèse d'images : application à l'animation de textiles.

Fonctions assurées

2000-2003	Allocataire de recherche MENRT et moniteur . Université Pierre Mendès France (UPMF) de Grenoble, UFR Sciences de l'Homme et de la Société (SHS), département Informatique et Mathématiques en Sciences Sociales (IMSS), Laboratoire ID-IMAG, projet INRIA-APACHE.
2003-2004	ATER (Attaché Temporaire Enseignement Recherche) à temps complet. Université Louis Pasteur de Strasbourg (ULP), département d'Informatique, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection (LSIIT), équipe IGG (Informatique Géométrique et Graphique).
2004-2005	ATER à mi-temps. ULP de Strasbourg, département d'Informatique, LSIIT, équipe IGG.
Sept. 2005 -	Maître de conférences . Université Lyon 1, département d'Informatique de la FST. LIRIS, équipe SAARA.
2014-2015	Chercheur invité au sein de l'équipe GMCAO du laboratoire TIMC-IMAG.
2015	Obtention de la Prime d'Encadrement Doctoral et de Recherche (PEDR)

Congés au niveau des enseignements

2008-2009	Congé de maternité du 09/02/2009 au 31/05/2009 - Service d'enseignement de 96 heures.
2011-2012	Congé de maternité du 11/09/2011 au 31/12/2011 - Service d'enseignement de 96 heures.
2014-2015	Délégation CNRS du 01/09/2014 au 28/02/2015 et CRCT Lyon 1 du 01/03/2015 au 31/08/2015 - Service d'enseignement de 0 heure.
2017-2018	Congé de maternité du 15/01/2018 au 15/07/2018 - Service d'enseignement de 38.4 heures.

Thématiques de recherche et mobilité géographique

Doctorat (2000-2003 - Grenoble) : Algorithmes parallèles pour la synthèse d'images. Mon doctorat a donné lieu à une collaboration entre le laboratoire ID-IMAG (projet INRIA-APACHE, parallélisme) et le laboratoire GRAVIR-IMAG (projet INRIA-iMAGIS, informatique graphique). Mon travail de recherche a ouvert un nouveau domaine de recherche dans le laboratoire ID-IMAG concernant la gestion efficace du flux entre applications parallèles, et dans l'équipe INRIA-iMAGIS avec l'apport du parallélisme en informatique graphique.

Post-doctorat (2003-2005 - Strasbourg) : Visualisation scientifique. Mon travail post-doctoral, réalisé au sein de l'équipe IGG-LSIIT et du projet INRIA-CALVI, a abordé le problème de la visualisation scientifique d'une grande masse de données 4-D issues de simulations de plasmas ou de faisceaux de particules. L'enjeu était de fournir des outils interactifs d'analyse de ces données, c'est-à-dire à allier une représentation judicieuse des données à des méthodes de compressions de données.

Maître de conférences (depuis sept. 2005 - Lyon) : Simulation 3D pour le médical. Mon travail de recherche réalisé au sein de l'équipe SAARA, puis de l'équipe ORIGAMI du LIRIS, concerne de manière générale l'élaboration de modèles physiques pour la modélisation d'objets déformables et plus particulièrement la réalisation de simulateurs pour l'apprentissage de gestes médicaux. Je me suis intéressée aux gestes de l'accouchement avec la modélisation biomécanique de la descente du fœtus au cours de l'accouchement, et aux gestes de la ponction avec l'insertion d'aiguilles dans des articulations. J'ai également suivi les travaux faits autour du projet ETOILE et du LabEx PRIMES sur la simulation biomécanique de la respiration dans le cadre de traitements contre le cancer.

Ces différentes activités de recherche, effectuées à **Grenoble, Strasbourg et Lyon**, m'ont permis de collaborer avec des chercheurs de différentes disciplines (médecins, physiciens, numériciens et mécaniciens, informaticiens en parallélisme, en informatique graphique et en automatique), me permettant d'acquérir de solides expériences dans plusieurs thématiques de recherche en informatique :

- en **algorithmique parallèle** avec la parallélisation multi-CPU et/ou GPU d'applications et l'élaboration de couplage entre simulations parallèles et visualisation multi-écrans,
- en **visualisation scientifique** avec l'élaboration d'outils de visualisation multi-dimensionnelle,
- en **rendu (texture)** avec des travaux sur l'élaboration d'un outil de génération de textures et l'élaboration d'un rendu type échographique en temps réel,
- en **synthèse d'images (animation) et Réalité Virtuelle** avec l'élaboration d'animations d'objets 3D,
- et en **modélisation et simulation numérique** par la création de nouveaux modèles physiques pour la simulation d'objets déformables dans le cadre notamment d'applications médicales.

Responsabilités administratives

2020	Membre du jury du CAPES Numérique et Sciences Informatiques (NSI)
2019	Membre du comité consultatif de l'Université Lyon 2 (section CNU 27)
2019	Membre de l'atelier sur la refonte de l'interface utilisateur de HAL
2016	Membre du comité consultatif de l'Université de Bourgogne (section CNU 27)
2015	Membre du comité consultatif de l'Université d'Auvergne (section CNU 27)
2013, 2014, 2019, 2020	Membre du comité consultatif de l'Université Lyon 1 (section CNU 27)
Depuis 09/2012	Membre de la commission des suivis de thèse du LIRIS
Depuis 06/2011	Membre élue au conseil du laboratoire du LIRIS
2010-2012	Réfèrent Pédagogique en Informatique, portail Maths-Info de la Licence Sciences, Technologie, Santé (STS) de l'Université Lyon 1.
2007-2010	Directrice des Etudes (DE) en Informatique, portail Maths-Info de la Licence STS. Membre de la commission pédagogique des dossiers de Demande d'Admission Préalable (DAP) - Portail Maths-Info de la Licence STS. Membre du Groupe de Travail LYCEE du CEVU (Conseil des Etudes et de la Vie Universitaire) de l'Université Lyon 1.
2006 et 2016	Représentante de l'Université Lyon 1 au jury du baccalauréat .

Animation scientifique

- **Membre du comité d'organisation de MEDINFO 2019** (*the 17th World Congress of Medical and Health Informatics*). Cette conférence internationale réunit les chercheurs autour des problématiques de l'informatique pour la santé du 26 au 30 août 2019 à Lyon.
- **Membre du comité de programme de VRIPHYS** (Eurographics Workshop on Virtual Reality Interaction and Physical Simulation) depuis 2013. **Organisation du workshop en novembre 2015 et avril 2017** au LIRIS. Ce workshop est la référence dans la thématique de recherche concernant la modélisation physique d'objets déformables.
- **Organisatrice d'une école thématique** du 30 juin au 4 juillet 2014 sur la simulation biomécanique dans le cadre de la conception de simulateurs médicaux. Les orateurs invités étaient les acteurs nationaux majeurs du domaine. Toutes les informations sont disponibles sur le site web que j'avais conçu : <http://ecole-simu2014.sciencesconf.org>.
- **Création et animation du thème F du GdR STIC-Santé** avec T. Redarce de janvier 2007 à décembre 2014. Membre du **Comité de Pilotage du GdR STIC-Santé** sur la même période. Durant cette période, j'ai organisé 10 journées thématiques et gérer le site Web du thème F qui permettait de diffuser les présentations des journées. Nous avons arrêté l'animation scientifique de ce thème suite à l'arrêt du financement du GdR STIC-Santé par les instituts du CNRS.
- **Membre du Comité Scientifique de la Journée Scientifique 2014 de la FST** (Faculté des Sciences et Technologies) de l'Université Lyon 1. Thématique de la journée : "les grandes masses de données scientifiques : gestion et analyse".
- **Co-organisatrice de la Journée Thématique du LabEx PRIMES 2014** sur la simulation biomécanique.
- **Membre du groupe de travail du Conseil général de l'économie, de l'industrie, de l'énergie et des technologies (CGEiet)** mandaté pour effectuer un éclairage prospectif sur le thème de la "maîtrise des technologies et connaissance en santé". Rapport "Technologies et connaissances en santé", numéro 2012/11/CGEiet/SG, décembre 2013.
- **Participation à l'organisation de la conférence RFIA 2012** (janvier 2012 à Lyon).
- **Organisatrice du GTAS 2007** (Groupe de Travail Animation et Simulation) de l'AFIG (Association Française d'Informatique Graphique).
- **Responsable du Groupe de Travail Visualisation** du projet INRIA-CALVI (2004-2005).

Participation à des comités de lecture

- Relectrice pour les revues internationales IEEE EMBS, IEEE TEVC.
 - Relectrice régulière pour la conférence nationale AFIG et la revue nationale REFIG.
 - Relectrice régulière pour les conférences internationales VRIPHYS, EuroGraphics.
-

Projets de recherche (par ordre chronologique)

- Membre du projet SPARTE (Simulator of Puncture for ARTiculations under Echography) financé par l'IDEFI-SAMSEI (Stratégies d'Apprentissage des Métiers de Santé en Environnement Immersif).
 - Membre du projet "Adaptive mixed-element mesh generation for a training simulator of childbirth gestures", ECOS Sud-CONICYT 2015.
 - Membre du projet GenSim, projet transverse LIRIS 2013.
 - Membre du LabEx PRIMES (Physics, Radiobiology, Medical Imaging and Simulation) (WP5), LabEx 2012.
 - Membre du **projet SAGA, ANR MN 2012** (responsable scientifique de la tâche 3, création site web). **Coordinatrice du projet lors des soumissions précédentes.**
 - Membre du projet TOPOSim, projet transverse LIRIS 2012.
 - Membre du projet ETOILE et du projet ENVISION (European NoVel Imaging Systems for ION therapy) (WP4), projet européen.
 - Membre du projet SIMED, cluster ISLE de la région Rhône-Alpes.
 - Membre du projet GMCAO, cluster ISLE de la région Rhône-Alpes.
 - Membre du projet INRIA-CALVI (durant poste d'ATER à Strasbourg).
 - Membre du projet INRIA-APACHE (durant doctorat à Grenoble).
-

Séminaires invités

- Présentation lors de la "French American Innovation Day" (FAID), organisée par le Service pour la "Science et la Technologie" de l'Ambassade de France aux Etats-Unis, le 7 mars 2019 à Houston, Etats-Unis.
- Présentation lors du colloque "IA & médecine : une chance ?" organisé par le conseil de l'ordre des médecins de la région Rhône-Alpes, le 9 novembre 2018 à Lyon.
- Présentation lors des 19^e journées du GTAS (Groupe de Travail Animation et Simulation) de l'AFIG (Association Française d'Informatique Graphique) en juillet 2017 à Strasbourg.
- Présentation lors du congrès annuel de la SIFUD PP (Société Interdisciplinaire Francophone d'UroDynamique et de Pelvi Périnéologie) à La Baule en septembre 2015.
- Séminaire invité lors des journées au vert de l'équipe ALCoV du laboratoire ISIT (Image Science for Interventional Techniques, UMR6284) en juillet 2015.
- Séminaire de l'équipe GMCAO du Laboratoire TIMC-IMAG de Grenoble en juin 2015.
- Présentation au ministère des finances à Bercy en septembre 2013 dans le cadre d'un groupe de travail sur le thème de la "maîtrise des technologies et connaissance en santé".
- Présentations lors de l'AG du GdR STIC Santé en 2010, 2012 et 2014.
- Présentations lors de réunion inter-thèmes du GdR STIC Santé en 2007 et 2009.
- Séminaire de l'équipe SAARA du LIRIS à plusieurs reprises.
- Séminaire de l'équipe ORIGAMI du LIRIS.

Collaborations extérieures

En plus des collègues du LIRIS comme Behzad Shariat (professeur, équipe SAARA), Jean-Michel Moreau (professeur désormais à la retraite, équipe SAARA), Fabrice Jaillet (Maître de conférences, HDR, équipe SAARA, puis ORIGAMI), Guillaume Damianiand (CR CNRS, HDR, équipe M2Disco, puis ORIGAMI) et Jean-Claude Iehl (maître de conférences, équipe R3AM, puis ORIGAMI) avec lesquels je travaille quotidiennement et co-encadre des thèses, voici la liste actuelle de mes collaborations extérieures :

- **Service de rhumatologie de l’Hôpital Lyon Sud.** J’ai débuté en 2016 une collaboration avec le Dr Fabienne Coury autour de la réalisation d’un simulateur pour l’apprentissage du geste de la ponction sous échographie.
- **Laboratoire Interuniversitaire de Biologie de la Motricité (LIBM).** J’ai également débuté en 2016 une collaboration avec Nady El Hoyek autour de la réalisation de ce simulateur de ponction sous échographie.
- **Laboratoire de Mécanique de Lille.** Discussion avec Mathias Brieu et le Pr Michel Cosson (CHRU de Lille) autour de la compréhension des organes du bassin pelvien afin de proposer une modélisation pertinente du mécanisme de l’accouchement.
- **Laboratoire de Biomécanique et Mécanique des Chocs de l’IFSTTAR de Lyon.** Collaboration avec Stéphane Nicolle et Jean-Francois Paliarne (Laboratoire de Physique de l’ENS de Lyon) avec le co-encadrement d’une thèse financée par le LabEx PRIMES autour de l’amélioration du modèle masses-ressorts permettant la simulation de nouvelles lois de comportement obtenues par expérimentations.
- **Laboratoire Ampère, INSA de Lyon.** Je travaille avec Richard Moreau et TanneGuy Redarce autour de la problématique de la conception de simulateurs médicaux. Nous étions réunis au sein du projet ANR SAGA (Simulateurs pour l’Apprentissage des Gestes de l’Accouchement) qui visait la réalisation d’un simulateur pour l’apprentissage des gestes médicaux de l’accouchement. J’ai animé également le thème F du GdR STIC Santé avec TanneGuy Redarce de 2007 à 2014, et nous avons co-encadré de 2007 à 2010 la thèse de Romain Buttin sur la thématique du simulateur d’accouchement. Nous travaillons actuellement sur le simulateur de ponction sous échographie.
- **Laboratoire TIMC-IMAG, équipe GMCAO.** Je travaille avec Emmanuel Promayon sur la modélisation de tissus. Ce travail s’est traduit par le co-encadrement de 2011 à 2014 de la thèse de Mathieu Baillet et par une implication dans le projet SAGA. Cette collaboration a été renforcée en 2015 avec une mobilité au sein de l’équipe GMCAO.
- **Ecole des Mines de Paris, CAOR.** J’ai travaillé avec François Goulette et Frédéric Xerri sur la conception de simulateurs médicaux dans le cadre du projet SAGA.
- **Laboratoire des Sciences de l’Education de Grenoble.** J’ai travaillé avec Lucile Vadcuard au sein du projet ANR-SAGA et nous avons animé quelques années ensemble le thème F du GdR STIC Santé.
- **Maternité Lyon Sud.** J’ai travaillé avec le Pr Olivier Dupuis au travers du projet SAGA. Il a supervisé l’aspect médical du simulateur d’accouchement.
- **Hôpital Saint Vincent de Paul et maternité Port Royal** de l’Assistance Publique-Hôpitaux de Paris. J’ai travaillé avec le docteur Gilles Grangé sur le simulateur d’accouchement. Cette collaboration a notamment permis l’obtention de données médicales (IRM, scanners), et la validation du travail de thèse effectué par Romain Buttin.
- Je suis toujours en contact avec **François Faure (INRIA-IMAGINE, Grenoble)** qui était co-encadrant de ma thèse. Nous échangeons régulièrement autour des problématiques de la simulation pour le domaine médical.

Encadrements liés à la recherche

Co-encadrement de doctorats (Université Lyon 1)

1. **Charles Barnouin**, *Un outil pédagogique de la ponction des grosses articulations sous échographie* - Début : 01/10/2016 ; soutenance : 07/05/2020 (fin de contrat le 31/03/2020) - **Directeurs de thèse** : Fabrice Jaillet et **moi-même (50 %)** (co-encadrement de 3 ans et 6 mois). **Jury de soutenance** : Stéphane Cotin (rapporteur), Maud Marchal (rapporteur), Fabienne Coury-Lucas (examinatrice), Philippe Delachartre (examinateur), Jean-Michel Dischler (examinateur), Jocelyne Troccaz (examinatrice). Bourse du projet idefi SAMSEI.
2. **Karolina Golec**, *New formulation of mass-spring system for soft tissue modeling* - Début : 01/10/2014 ; soutenance : 19/01/2018 - **Directeurs de thèse** : Guillaume Damiand, Stéphane Nicolle (LBMC, Lyon) et **moi-même (33 %)** (co-encadrement de 3 ans et 3 mois). **Jury de soutenance** : Maud Marchal (rapporteur), Yohan Payan (rapporteur), Sébastien Laporte (examinateur), Laurence Cheze (examinatrice), Philippe Meseure (Président). Bourse du LabEx PRIMES.
3. **Grégoire Lemasson**, *Optimisation multi-échelle de modèle 3D de CAO* - Début : 01/02/2012 ; soutenance prévue en février 2015. Abandon durant la rédaction - **Directeurs de thèse** : Behzad Shariat, Jean-Claude Iehl et **moi-même (15 %)**. Bourse CIFRE avec Core Technologie.
4. **Mathieu Bailet**, *Simulation de la dynamique pelvienne de la femme enceinte et interaction du fœtus avec les instruments* - Début : 01/10/2011 ; soutenance : 15/12/2014 - **Directeurs de thèse** : Emmanuel Promayon (laboratoire TIMC-IMAG, Grenoble) et **moi-même (50 %)** (co-encadrement de 3 ans et 3 mois). **Jury de soutenance** : Mathias Brieu (président), François Faure (examinateur), Hervé Delingette (rapporteur), Loïc Barthe (rapporteur). Bourse Cluster ISLE.
5. **Elsa Fléchon**, *Simulation d'objets déformables* - Début : 01/10/2011 ; soutenance : 09/12/2014 - **Directeurs de thèse** : Fabrice Jaillet et **moi-même (50 %)** (co-encadrement de 3 ans et 3 mois). **Jury de soutenance** : Stéphane Cotin (président), Benjamin Gilles (examinateur), Guillaume Damiand (examinateur), Emmanuel Promayon (examinateur), David Cazier (rapporteur), Philippe Meseure (rapporteur). Bourse ministérielle.
6. **Xavier Faure**, *Vers un modèle humain multi-échelle parallélisé* - Début : 01/10/2010 ; soutenance : 29/09/2014 - **Directeurs de thèse** : Fabrice Jaillet, Jean-Michel Moreau et **moi-même (33 %)** (co-encadrement de 4 ans). **Jury de soutenance** : Hervé Delingette (président), Yohan Payan (examinateur), François Faure (rapporteur), Christian Duriez (rapporteur). Bourse du projet ETOILE.
7. **Romain Buttin**, *Réalisation d'un environnement virtuel pour la conception d'un simulateur d'accouchement* - Début : 01/10/2007 ; soutenance : 08/10/2010 - **Directeurs de thèse** : Behzad Shariat, Tanneguy Redarce (laboratoire Ampère) et **moi-même (50 %)** (co-encadrement de 3 ans). **Jury de soutenance** : François Goulette (président), Gilles Grangé (examinateur), Yohan Payan (rapporteur), André Crosnier (rapporteur). Bourse Cluster ISLE.

Participation à l'encadrement de doctorats (ULP de Strasbourg et Lyon 1)

1. **Vincent Baudet**, *Modélisation et simulation paramétrable d'objets déformables. Applications aux traitements des cancers pulmonaires* - Début : janvier 2002 ; soutenance : 30/06/2006 - **Directeurs de thèse** : Behzad Shariat et Fabrice Jaillet - **Participation à l'encadrement depuis mon arrivée en septembre 2005 : 33%**. **Jury de soutenance** : Dominique Faudot (présidente), Christian Carrie (invité), Christophe Chaillou (rapporteur), Yohan Payan (rapporteur), Emmanuel Promayon (invité).
2. **Matthieu Haefele**, *Simulation adaptative et visualisation haute-performance de plasmas et de faisceaux de particules* - Début janvier 2003 ; soutenance : 05/04/2007 - **Directeurs de thèse** : Jean-Michel Dischler (LSIIT-IGG) et Eric Sonnendrücker (IRMA) - **Participation à**

l'encadrement depuis mon arrivée à Strasbourg en décembre 2003 : 30% (membre du jury de thèse). Jury de soutenance : Guillaume Latu (examinateur), Olivier Coulaud (rapporteur), Wilfried Lefer (rapporteur), Catherine Mongenet (rapporteur), Florence Zara (invitée).

Encadrement d'ingénieurs de recherche

1. **Matthieu Coquet** - Ingénieur de recherche financé par le projet ANR SAGA - *Simulation de la descente du fœtus au cours de l'accouchement* - Mars 2014-Septembre 2015 (18 mois) - Encadrement : F. Jaillet et **moi-même (50 %)**.

Encadrement de Masters 2 Recherche / Professionnel, PFE ou DEA

1. **Matthieu Coquet** - 5e année et PFE de l'INSA de Lyon - *Modèle topologique générique pour la simulation physique* - Février-Août 2013 - Encadrement : G. Damiand et **moi-même (50 %)**.
2. **Elsa Fléchon** - M2 Professionnel de Lyon 1 - *Simulation physique d'objets déformables* - Mars-Août 2011 - Encadrement : F. Jaillet et **moi-même (50 %)**.
3. **Grégoire Lemasson** - M2 Professionnel de Lyon 1 - *Détection de Zone de Crique par l'utilisation du GPU* - Mars-Août 2011 - Encadrement : J.C. Iehl, B. Shariat et **moi-même (33 %)**.
4. **Zhifan Jiang** - M2 Recherche Lyon 1 et PFE de l'INSA de Lyon - *De l'imagerie médicale aux maillages volumiques optimaux pour la simulation biomécanique* - Février-Août 2011 - Encadrement : F. Jaillet et **moi-même (50 %)**.
5. **Jean-Francois El Hajjar** - DEA de l'ULP - *Implantation d'un raymarching adapté pour la visualisation de données volumiques issues de simulations gyrocinétiques* - Février-juillet 2005 - Encadrement : Jean-Michel Dischler et **moi-même (50 %)**.
6. **Christophe Mion** - DEA de l'ULP - *Exploration interactive de données 4-D+t issues de simulations de plasma* - Février-juillet 2004 - Encadrement : Jean-Michel Dischler et **moi-même (50 %)**.

Encadrement de Masters 1 (ULP et Lyon 1) - 21 étudiants

- **2 étudiants (B. A. Nguyen et A. Zahidi)** - Master 1 Lyon 1 - *Méthode d'intégration* - Novembre 2017-mai 2018 - **Encadrement à 50% avec Guillaume Damiand.**
- **1 étudiant (C. Lemeunier)** - Master 1 Lyon 1 - *Traitement de collisions* - Novembre 2017-mai 2018 - **Encadrement à 50% avec Guillaume Damiand.**
- **2 étudiants (P. Lafoix-Tranchant et A. Oleksiak)** - Master 1 Lyon 1 - *Calcul parallèle pour la manipulation de maillages* - Novembre 2017-mai 2018 - **Encadrement à 50% avec Fabrice Jaillet.**
- **2 étudiants (A. Loup et LDi Sanza)** - Master 1 Lyon 1 - *Gestion de scènes de simulation NExT* - Novembre 2016-mai 2017 - **Encadrement à 50% avec Fabrice Jaillet.**
- **2 étudiants (E. Fléchon et G. Lemasson)** - Master 1 Lyon 1 - *Simulation d'un système de particules* - Novembre 2009-février 2010 - **Encadrement à 100%.**
- **2 étudiants** - Master 1 Lyon 1 - *Simulation physique d'objets déformables* - Novembre 2009-février 2010 - **Encadrement à 100%.**
- **2 étudiants** - Master 1 Lyon 1 - *Développement d'une interface de visualisation dans un environnement virtuel* - Novembre 2008-février 2009 - **Encadrement à 100%.**
- **2 étudiants** - Master 2 Lyon 1 - *Traitement des collisions dans une simulation d'objets déformables : applications à une simulation de textiles* - Novembre 2007-février 2008 - **Encadrement à 100%.**

- **4 étudiants** - Master 1 Lyon 1 - *Traitement des collisions dans une simulation d'objets déformables : applications à une simulation de textiles* - Novembre 2006-février 2007 - **Encadrement à 100%**.
 - **1 étudiant** - Master 1 Lyon 1 - *Modélisation d'organes* - Novembre 2006-février 2007 - **Encadrement à 100%**.
 - **1 étudiant** - Maîtrise ULP - *Étude et développement d'un module de contrôle pour une plate-forme de simulation numérique* - Février-juin 2004. **Encadrement à 50%**.
-

Participation à des jurys de doctorat

Membre du jury en tant que **co-encadrante** pour les soutenances de doctorat de :

- **Charles Barnouin** - Date de soutenance : 07/05/2020.
- **Karolina Golec** - Date de soutenance : 19/01/2018.
- **Mathieu Bailet** - Date de soutenance : 15/12/2014.
- **Elsa Fléchon** - Date de soutenance : 09/12/2014.
- **Xavier Faure** - Date de soutenance : 20/09/2014.
- **Romain Buttin** - Date de soutenance : 08/10/2010.

Membre du jury en tant qu'**examinatrice** pour les soutenances de doctorat de :

- **Noëlie Debs**
 - **Titre** : Prédiction de l'issue lésionnelle et clinique de l'accident vasculaire cérébral par approche d'intelligence artificielle
 - **Date de soutenance** : 13/10/2020.
 - **Laboratoire** : Laboratoire Creatis
 - **Directeurs** : Tae-Hee Cho (HCL), Carole Frindel (Creatis), David Rousseau (LARIS).
 - **Jury de soutenance** : Stéphanie Allasonnière (rapporteur), Caroline Petitjean (rapporteur), Emmanuel Barbier (examineur), Yves Berthezène (examineur), **Florence Zara (examinatrice)**.
- **Ma de Los Angeles Alamilla Daniel**
 - **Titre** : Development of a haptic simulator for practicing the intraarticular needle injection under echography
 - **Date de soutenance** : 12/03/2020.
 - **Laboratoire** : Laboratoire Ampère, INSA de Lyon.
 - **Directeurs** : Tanneguy Redarce, Richard Moreau.
 - **Jury de soutenance** : Gérard Poisson (rapporteur), Christine Barthod (rapporteur), Pierre Vieryres (examineur), Emmanuel Promayon (examineur), **Florence Zara (examinatrice)**.
- **Gaëlle Fiard**
 - **Titre** : Apprentissage des biopsies prostatiques par la simulation : vers la validation du simulateur Biopsym
 - **Date de soutenance** : 03/12/2018.
 - **Laboratoire** : Laboratoire TIMC-IMAG, Université de Grenoble.
 - **Directeurs** : Jocelyne Troccaz, Jean-Luc Descotes.

-
- **Jury de soutenance** : Jacques Hubert (rapporteur), Maud Marchal (rapporteur), Arnauld Villers (examinateur), Emmanuel Promayon (examinateur), Olivier Palombi (président), **Florence Zara (examinatrice)**.

 - **Pierre-Luc Manteaux**
 - **Titre** : Simulation et contrôle de phénomènes physiques
 - **Date de soutenance** : 03/10/2016.
 - **Laboratoire** : Laboratoire Jean Kuntzmann (LJK), Université de Grenoble.
 - **Directeurs** : Marie-Paule Cani (LJK-IMAGINE) et François Faure (LJK-IMAGINE).
 - **Jury de soutenance** : Christian Duriez (rapporteur), Maud Marchal (rapporteur), Paul Kry (examinateur), Joelle Thollot (présidente), **Florence Zara (examinatrice)**.

 - **Nicolas Herzig**
 - **Titre** : De la Conception à la Commande d'une Nouvelle Interface Haptique 4 Axes Hybride Pneumatique Electrique pour la Simulation d'Accouchement : le BirthSIM.
 - **Date de soutenance** : 24/06/2016.
 - **Laboratoire** : Laboratoire Ampère, INSA de Lyon.
 - **Directeurs** : Tanneguy Redarce (Ampère), Richard Moreau (Ampère).
 - **Jury de soutenance** : Guillaume Morel (rapporteur), Pierre Renaud (rapporteur), Christine Barthod (examinatrice), Olivier Dupuis (examinateur), Jacques Lottin (président), **Florence Zara (examinatrice)**.

 - **Matthieu Haefele**
 - **Titre** : Simulation adaptative et visualisation haute-performance de plasmas et de faisceaux de particules.
 - **Date de soutenance** : 05/04/2007.
 - **Laboratoire** : LSIIT-IGG, Université Louis Pasteur de Strasbourg.
 - **Directeurs** : Jean-Michel Dischler (LSIIT-IGG) et Eric Sonnendrücker (IRMA).
 - **Jury de soutenance** : Guillaume Latu (examinateur), Olivier Coulaud (rapporteur), Wilfried Lefer (rapporteur), Catherine Mongenet (rapporteur), **Florence Zara (invitée)**.

Charges d'enseignement depuis 2005

J'ai effectué un total de 2735 **heures équivalents TD** en tant que maître de conférences au département d'Informatique de l'Université Lyon 1 depuis le 1er septembre 2005. Pour analyser ces heures, la table A.1 récapitule tout d'abord depuis 2005 mes charges totales d'enseignements déclarées en heures en fonction du niveau des enseignements et de leur type (CM, TD, TP, autre). La moitié de mon service a ainsi été effectué en Licence première année et en TPs.

Niveau	CM	TD	TP	Autre	Total	%
L1 Informatique	171	383,5	838,5	120,5	1513,5	55,21 %
L2 Informatique	75	78	294	42,75	489,75	17,86 %
L3 Informatique	0	12	48	15	75	2,73 %
M1 Informatique	14,5	33	60	15	122,5	4,46 %
M1 Biomédicale	24	24	0	0	48	1,75 %
M2 Informatique	69,5	117	90	3	279,5	10,19 %
Autre	0	10,5	0	202,4	212,9	7,76 %
Total	354	658	1330,5	398,65	2741,15	100 %
%	12,91%	24 %	48,53 %	14,54 %	100 %	

Table A.1 – Récapitulatif par niveau des charges d'enseignements effectuées depuis 2005.

A noter que :

- L'intitulé "Autre" pour le niveau englobe la décharge d'enseignements de 42 heures accordée aux nouveaux nommés de l'Université Lyon 1, ainsi que les primes de responsabilités pédagogiques (PRP) obtenues en tant que Directrice des Etudes, Référente Pédagogique ou en charge de l'enseignement PCI (explicité par la suite).
- L'intitulé "Autre" pour le type d'enseignement correspond aux heures de Plan Licence accordées pour effectuer des heures supplémentaires dans les enseignements de Licence (soutien, TDs supplémentaires), aux heures attribuées pour la responsabilité d'Unités d'Enseignements de Licence ou encore à l'encadrement de stages en troisième année de Licence, ou en Master première ou deuxième année.
- En L3 Informatique, 3 h de TD correspondent en pratique à des encadrements de stages.
- En M1 Informatique, il y a 18 h de TD qui correspondent à des encadrements de stages.
- En M1 Biomédicale, toutes les heures correspondent à des heures de cours, mais déclarées pour certaines années directement en équivalent heures TD.
- Dans les 117 h de TD effectuées en M2 Informatique, 108 correspondent en pratique à 54 h de cours et 54 h de TP (mais toutes les heures sont déclarées en heures TD dans un des M2 Informatique), et les 6 autres heures correspondent à l'encadrement de stages.

La table A.2 récapitule ensuite les charges d'enseignement effectuées année après année depuis mon arrivée en 2005 à l'Université Lyon 1. A noter que j'essaie de limiter au maximum mes heures supplémentaires pour avoir du temps pour effectuer de la recherche. En 2005-2006, j'ai bénéficié d'une décharge de 42 heures en tant que maître de conférences nouvellement recrutée ; en 2008-2009 et 2011-2012 ma charge d'enseignement a été réduite à 96 heures pour cause de congés de maternité ; en 2014-2015 ma charge d'enseignement est nulle pour cause de délégation CNRS de 6 mois cumulée à un CRCT de l'Université Lyon 1 de 6 mois ; et en 2017-2018 ma charge d'enseignement a été réduite à 38,4 heures pour cause de congé de maternité (durée de 6 mois pour un troisième enfant).

Année	CM	TD	TP	Autre	Total	Total équivalent TD
2019-2020	18,5	55,5	128,5	1,5	204	208,98
2018-2019	28	58,5	94,5	10,5	191,5	203,43
2017-2018	20	31,5	60	5	116,5	112,57
2016-2017	24,5	54,5	120,5	19	218,5	224,75
2015-2016	34	65	113,5	34,5	247	253,68
2014-2015	0	0	0	0	0	0
2013-2014	30	70,5	65	22,5	188	201,83
2012-2013	30	51	80	21,5	182,5	196,76
2011-2012	22,5	34	57,5	34	148	151,64
2010-2011	34	42	73,5	53,5	203	216,88
2009-2010	35,5	60	52,5	50,75	198,75	214,52
2008-2009	37	41	75	9,5	162,5	156
2007-2008	27	40,5	106,5	56	230	208,1
2006-2007	21	18	147	44,4	230,4	191,9
2005-2006	0	36	174	42	252	194
Total	362	658	1348	405,65	2772,65	2735,06

Table A.2 – Récapitulatif du nombre d'heures effectuées chaque année à Lyon 1 depuis 2005.

Enfin, la table A.3 présente les heures déclarées pour l'année 2019-2020.

Niveau	Enseignement	CM	TD	TP	Autre	Total	Total équiv. TD
L1 Informatique	Algo. prog. impérative		24	25	1,5	50,5	50,5
L2 Informatique	Informatique graphique	7,5	7,5	36		51	54,75
L2 Informatique	Système d'exploitation		6	18		24	24
L3 Informatique	Programmation concurrente		9	15		24	24
L3 Informatique	Suivi stage			3		3	3
M1 Informatique	Algorithmique distribuée		9	9		18	18
M1 Informatique	Animation	4,5		13,5		18	20,5
M1 Informatique	Projet Orientation Master						
M2 Informatique	Animation	6,5		9		15,5	18,75
	Total	18,5	55,5	128,5	1,5	204	208,98

Table A.3 – Récapitulatif des heures effectuées en 2019-2020.

Les enseignements que j'effectue actuellement concernent ainsi :

- L1 Informatique - Unité d'Enseignement LIFAP1 - Algorithmique Programmation Impérative, Initiation. TDs, TP.
- L2 Informatique - Unité d'Enseignement LIFGraphique - Informatique Graphique. Cours, TDs, TP. **Responsable de l'enseignement** au semestre d'automne.
- L2 Informatique - Unité d'Enseignement LIFASR5 - Système d'exploitation. TDs, TP.
- L3 Informatique - Unité d'Enseignement LIFASR7 - Programmation concurrente. TDs, TP.
- Master 1 Informatique - Unité d'Enseignement M1if37 - Animation en synthèse d'image. Cours, TP. **Responsable de l'enseignement**. Enseignement en lien avec mes problématiques de recherche.
- Master 1 Informatique - Unité d'Enseignement M1if12 - Algorithmique distribuée. TDs, TP.
- Master 2 Image Développement et Technologies 3D - Unité d'Enseignement Animation, Corps Articulés et Moteurs Physiques. Cours, TP. **Responsable de l'enseignement**. Enseignement en lien avec mes problématiques de recherche.

Mes enseignements passés ont concerné :

- L1 - Unité TRansversale TR1 PCI - Pratiques et Compétences Informatiques. **Co-responsable de l'enseignement. 1500 étudiants** de L1 au premier semestre (100 groupes de TPs, 10 amphis, 50 intervenants) et 160 étudiants au second semestre. Cours, TPs.
- L1 Informatique - Unité d'Enseignement LIFAP2 - Algorithmique et programmation récursive. Cours, TDs, TPs. **Responsable de l'enseignement** au semestre de printemps.
- L3 Informatique - Unité d'Enseignement IF7 - Algorithmique et programmation complexe. TPs.
- Master 1 Informatique médicale et technologies de communication - Département de Biologie Humaine - Unité d'Enseignement Recherche Biomédicale. Cours (directement déclarés en heures TD). *Cette UE n'a pas ouverte cette année mais devrait reprendre à la rentrée 2020.*
- Master 1 Informatique - Unité d'Enseignement Parallélisme. TDs, TPs.
- Master 2 Ecole du jeu vidéo - Gamagora - Unité d'Enseignement Simulation. Cours, TPs (directement déclarés en heures TD). **Responsable de l'enseignement.** Enseignement en lien avec mes problématiques de recherche.
- Master 2 Recherche Informatique - Spécialité Informatique Graphique et Images - Unité d'enseignement MRI-IGI3 - Reconstruction géométrique et photométrie pour la réalité augmentée et virtuelle. Cours. Enseignement en lien avec mes problématiques de recherche.



Annexe **B**

Liste des publications

Mes publications et leurs résumés sont disponibles en ligne à l'adresse suivante :

<http://scholar.google.fr/citations?hl=fr&user=TlnUeTOAAAAJ>.

Les publications effectuées au LIRIS sont disponibles sur le site Web du LIRIS :

<https://liris.cnrs.fr/page-membre/florence-zara>.

Revue internationale avec comité de sélection

- [J8] Merge-and-simplify Operation for Compact Combinatorial Pyramid Definition. G. Damiand, **F. Zara**. In *Pattern Recognition Letters*, vol. 129, pp. 48-55. 2020.
- [J7] Hybrid 3D Mass-Spring System for simulation of isotropic materials with any Poisson's ratio. K. Golec, J.-F. Palierne, **F. Zara**, S. Nicolle, G. Damiand. In *Visual Computer*, 2019.
- [J6] Distributed Combinatorial Maps for Parallel Mesh Processing. G. Damiand, A. Gonzalez-Lorenzo, **F. Zara**, F. Dupont. *Open Access Revue Algorithms* 11(7), 105, 2018.
- [J5] Shell finite element model for interactive fetal head deformation during childbirth. M. Bailet, **F. Zara**, E. Promayon. *CMBBE - Computer Methods in Biomechanics and Biomedical Engineering* 16() :312-314, 2013.
- [J4] Biomechanical simulation of the fetal descent without imposed theoretical trajectory. R. Buttin, **F. Zara**, B. Shariat, T. Redarce, G. Grangé. *Computer Methods and Programs in Biomedicine* 111(2) :389-401, Elsevier, ISSN 0169-2607, 2013.
- [J3] Theme F "Medical Robotics for training and guidance" : results and future works. **F. Zara**, T. Redarce, P. Poignet. *IRBM BioMedical Engineering and Research* 34(1) :16-17, ISSN 1959-0318, 2013.
- [J2] Real-time structured texture synthesis and editing using image-mesh analogies. J.M. Dischler, **F. Zara**. *The Visual Computer* 22(9-11) :926-935, ISSN 0178-2789, 2006.
- [J1] Parallel Simulation of Large Dynamic System on a PCs Cluster : Application to Cloth Simulation. **F. Zara**, F. Faure, J.-M. Vincent. *International Journal of Computers and Applications*, volume 26, number 3, pages 173-180, 2004.

Conférences internationales avec comité de lecture et actes

- [C8] A real-time ultrasound rendering with model-based tissue deformation for needle insertion. C. Barnouin, **F. Zara**, F. Jaillet. Dans 15th International Conference on Computer Graphics Theory and Applications, GRAPP 2020, Malta, february 2020.
- [C7] Improvement of a Topological-Physical Model to manage different Physical Simulations. K. Golec, M. Coquet, **F. Zara**, G. Damiand. Dans WSCG 2015 - International Conference in Central Europe on Computer Graphics and Visualization, Plzen, Czech Republic. 2015.
- [C6] Biomechanical model of the fetal head for interactive childbirth simulation. M. Bailet, **F. Zara**, E. Promayon. Dans SURGETICA, Chambéry, 2014.
- [C5] Accurate thickness computation of a B-Rep model on the GPU. G. Lemasson, JC Iehl, **F. Zara**, B. Shariat, V Baudet, P. Arthaud. Dans International Conference on Computer Graphics, Visualization and Computer Vision (WSCG 2014), Pilsen, Czech Republic, 2014.
- [C4] A generic topological framework for physical simulation. E. Flechon, **F. Zara**, G. Damiand, F. Jaillet. Dans WSCG 2013 - International Conference in Central Europe on Computer Graphics and Visualization, Plzen, Czech Republic. pp. 104-113. ISBN 978-80-86943-74-9. ISSN 1213-6972, 2013.
- [C3] A Biomechanical Model of the Female Reproductive System and the Fetus for the realization of a Childbirth Virtual Simulator. R. Buttin, **F. Zara**, B. Shariat, T. Redarce. Dans IEEE Engineering in Medicine and Biology Society (EMBC'09), Minneapolis, Minnesota, USA, 2009.
- [C2] Integrating Tensile Parameters in Hexahedral Mass-Spring System for Simulation. V Baudet, M. Beuve, F. Jaillet, B. Shariat, **F. Zara**. Dans International Conference on Computer Graphics, Visualization and Computer Vision'2009 - WSCG'2009, Plzen, Czech Republic, 2009.
- [C1] Coupling Parallel Simulation and Multi-Display Visualization on a PC Cluster. J. Allard, B. Raffin, **F. Zara**. International Conference on Parallel and Distributed Computation (Euro-Par 2003), Klagenfurt, Austria, August 2003.

Articles de workshops internationaux avec comité de lecture

- [W7] A unified topological-physical model for adaptive refinement. E. Flechon, **F. Zara**, G. Damiand, F. Jaillet. Dans 11th Eurographics Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS 2014), Bremen, Allemagne, pp. 39-48, 2014.
 - [W6] Multi-criteria adaptation of physical simulations. F. Caillaud, X. Faure, **F. Zara**, F. Jaillet, JM. Moreau. Dans Eurographics Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS 2013), "Work in Progress" session, J. Bender, J. Dequidt, C. Duriez, and G. Zachmann ed. Lille, France. pp. 1-4, 2013.
 - [W5] An Implicit Tensor-Mass solver on the GPU for soft bodies simulation. X. Faure, **F. Zara**, F. Jaillet, JM. Moreau. Dans Eurographics Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS 2012), Darmstadt, Germany. pp. 1-10, 2012.
 - [W4] A dedicated Compression Scheme for Large Multidimensional Functions Visualization. M. Haefele, **F. Zara**, G. Latu, J.M. Dischler. Dans The 1st International Workshop on Super Visualization (IWSV08), Ile de Kos - Grèce, 2008.
 - [W3] Efficient and Easy Parallel Implementation of Large Numerical Simulations. R. Revire, **F. Zara**, T. Gauthier. ParSim 2003 (Special Session of EuroPVM/MPI 2003), Venice, Italy, september 2003.
 - [W2] Coupling Parallel Simulation and Parallel Visualization on PC Clusters. **F. Zara**, J.-M. Vincent, F. Faure. Commodity Clusters For Virtual Reality 2003, VR 2003 Workshop, Los Angeles, USA, March 2003.
 - [W1] Physical cloth simulation on a PC cluster. **F. Zara**, F. Faure, J.-M. Vincent. Parallel Graphics and Visualisation 2002, Eurographics Workshop, Blaubeuren, Germany, September 2002
-

Contributions à un ouvrage

- [BC2] Biomechanics of Living Organs : Hyperelastic Constitutive Laws for Finite Element Modeling. Chap. 15 : Biomechanical modeling of uterus. Application to a childbirth simulation. **F. Zara**, O. Dupuis, Elsevier, 2017.
 - [BC1] Développement de nouvelles technologies pour la formation aux gestes médico-chirurgicaux. **F. Zara**, L. Vadcard, T. Redarce. Annales des Mines. Réalités Industrielles. Connaissances et systèmes technologiques pour la santé. Novembre, 2014.
-

Reuves nationales avec comité de lecture

- [j2] Simuler en temps réel la descente du fœtus. **F. Zara**. Quadrature, 107, EDP Sciences, 2018.
 - [j1] Simulation biomécanique de la descente foetale sans trajectoire théorique imposée. R. Buttin, **F. Zara**, B. Shariat, T. Redarce, G. Grangé. Revue Electronique Francophone d'Informatique Graphique 5(2) :1-13, 2011.
-

Conférences nationales avec actes

- [c3] Real-time Ultrasound Rendering for Ultrasound-Guided Puncture Training. C. Barnouin, **F. Zara**, F. Jaillet. Dans Journées Françaises d'Informatique Graphique (JFIG 2018), Poitiers, France, 2018.
- [c2] Acquisition multi-modale en temps réel pour le suivi du mouvement respiratoire. X. Faure, F. Jaillet, **F. Zara**, JM. Moreau. Dans ORASIS 2011, Praz-sur-Arly, France, 2011.
- [c1] Simulation physique de textiles sur grappe de processeurs. **F. Zara**. AFIG 2001, Limoges, France, November 2001.

Éditions scientifique d'ouvrages

- [A5] Proceedings du numéro spécial "Foreword to the special section on VRIPHYS 2017", F. Jaillet, **F. Zara**, Computers & Graphics, Volume 69, A1-A2, 2017.
- [A4] Proceedings du workshop VRIPHYS 2017 - 13th Workshop on Virtual Reality Interactions and Physical Simulations, Lyon, France, 2017. Fabrice Jaillet, Florence Zara, Dieter W. Fellner (TU Darmstadt & Fraunhofer IGD, Germany).
- [A3] Proceedings du numéro spécial "Foreword to the special section on VRIPHYS 2015", F. Jaillet, **F. Zara**, G. Zachmann, Computers & Graphics, Volume 57, June 2016.
- [A2] Proceedings du workshop VRIPHYS 2015 - 12th Workshop on Virtual Reality Interactions and Physical Simulations, Lyon, France, 2015. Eurographics Association. F. Jaillet, **F. Zara**, G. Zachmann.
- [A1] Actes des 14èmes journées du Groupe de Travail Animation et Simulation de l'Association Française d'Informatique Graphique - GTAS'07. **F. Zara**. (Groupe de Travail Animation et Simulation de l'Association Française d'Informatique Graphique - GTAS'07), Lyon. 114 pages, 2007.

Posters dans conférences internationales

- ✉ 2 New Mass Spring System formulation to model the behavior of soft tissues. K. Golec, **F. Zara**, S. Nicolle, J-F. Palierne, G. Damiand. Dans ESB 2016 - 22nd Congress of the European Society of Biomechanics, July, 2016, Lyon, France.
- ✉ 1 Implicit Tensor-Mass solver on the GPU. X. Faure, **F. Zara**, F. Jaillet, JM. Moreau. Dans ACM/Eurographics Symposium on Computer Animation (SCA'2012), M. Lau ed. Lausanne, Switzerland, 2012.

Autres conférences

- ✓ 3 Hybrid 3D Mass-Spring System to simulate isotropic materials. K. Golec, J.-F. Palierne, **F. Zara**, S. Nicolle, G. Damiand. Dans Work In Progress Session, VRIPHYS 2017 : 13th Workshop on Virtual Reality Interaction and Physical Simulation, 23-24 April 2017, Lyon, France.
- ✓ 2 Tensor-Mass/FEM real-time simulation for medical applications. X. Faure, **F. Zara**, F. Jaillet, JM. Moreau. Dans Journée des thèses du LIRIS, Villeurbanne, France, 2012.
- ✓ 1 Integrating Tensile Parameters in 3D Mass-Spring System. V Baudet, M. Beuve, F. Jaillet, B. Shariat, **F. Zara**. Dans SURGETICA, Chambéry, France, 2007.

Rapports de recherche INRIA et LIRIS

- [RR15] Thickness computation of trimmed B-Rep model using GPU ray tracing. G. Lemasson, JC Iehl, **F. Zara**, B. Shariat, V Baudet, P. Arthaud. Rapport de recherche RR-LIRIS-2014-006, 2014.
- [RR14] Implicit Tensor-Mass solver on the GPU X. Faure, **F. Zara**, F. Jaillet, JM. Moreau. Rapport de recherche RR-LIRIS-2012-010, 2012.
- [RR13] Reconstruction et complétion de maillages sous contraintes. Z. Jiang, F. Jaillet, **F. Zara**. Rapport de recherche RR-LIRIS-2011-017, 2011.
- [RR12] Biomechanical Simulation of the Fetal descent without Imposed Theoretical Trajectory. R. Buttin, **F. Zara**, B. Shariat, T. Redarce, G. Grangé. Rapport de recherche RR-LIRIS-2011-008, 2011.

- [RR11] Acquisition multi-modale en temps réel pour le suivi du mouvement respiratoire. X. Faure, F. Jaillet, **F. Zara**, JM. Moreau. Rapport de recherche RR-LIRIS-2011-002, 2011.
- [RR10] Bio-mechanical Simulation of the Fetal descent without Imposed Theoretical Trajectory. R. Buttin, **F. Zara**, B. Shariat, T. Redarce, G. Grangé. Rapport de recherche RR-LIRIS-2010-012, 2010.
- [RR9] Simulation bio-mécanique de la descente foetale sans trajectoire théorique imposée. R. Buttin, **F. Zara**, B. Shariat, T. Redarce, G. Grangé. Rapport de recherche RR-LIRIS-2010-009, 2010.
- [RR8] Integrating Tensile Parameters in Mass-Spring System for Deformable Object Simulation. V Baudet, M. Beuve, F. Jaillet, B. Shariat, **F. Zara**. Rapport de recherche RR-LIRIS-2009-034, 2009.
- [RR7] Integrating Tensile Parameters in 3D Mass-Spring System. V Baudet, M. Beuve, F. Jaillet, B. Shariat, **F. Zara**. Rapport de recherche RR-LIRIS-2007-004, 2007.
- [RR6] New Mass-Spring System Integrating Elasticity Parameters in 2D. V Baudet, M. Beuve, F. Jaillet, B. Shariat, **F. Zara**. Rapport de recherche RR-LIRIS-2007-003, 2007.
- [RR5] Adaptive Raymarching and Multi-Sampled Pre-integration on Graphics Hardware : Application to Gyrokinetic Simulations Visualization. J.F. El Hajjar, **F. Zara**, J.M. Dischler, V. Grandgirard. Etablissement : RR INRIA Lorraine, projet CALVI, 2006.
- [RR4] Hypervolumetric Plasma-data Visualization. **F. Zara**, M. Haefele, C. Mion, J.M. Dischler. Etablissement : RR INRIA Lorraine, projet CALVI, 2006.
- [RR3] Implantation d'un raymarching adaptatif pour la visualisation de données volumiques issues de simulations gyrocinétiques. J.F. El Hajjar, **F. Zara**, J.M. Dischler, V. Grandgirard. Etablissement : RT INRIA Lorraine, projet CALVI, 2006.
- [RR2] Exploration interactive de données 4-D+t issues de simulations de plasma. Christophe Mion, Florence Zara, Jean-Michel Dischler. Research Report RR-5444, INRIA Lorraine, projet CALVI, 28 pages, December 2004.
- [RR1] Étude et développement d'un module de contrôle pour une plate-forme de simulation numérique. M. Haefele, D. Vigier, E. Violard, **F. Zara**. Technical Report RT-0299, INRIA Lorraine, projet CALVI, 56 pages, September 2004.

Mémoires

- ✎ 2 Algorithmes parallèles de simulation physique pour la synthèse d'images : application à l'animation de textiles. **F. Zara**. Thèse de l'Institut National Polytechnique de Grenoble (INPG) - Décembre 2003.
- ✎ 1 Serveur interactif de cartes géographiques parallélisé sur grappe de processeurs. **F. Zara**. DEA de l'Université Joseph Fourier (UJF - Grenoble I) - Juin 2000.



Simulation interactive d'objets déformables pour la conception de simulateurs d'apprentissage aux gestes médicaux-chirurgicaux

La formation médicale repose majoritairement sur l'observation plutôt que sur de la pratique car l'apprentissage réalisé directement auprès du patient pose des problèmes éthiques voire médicaux-légaux. Or la dextérité nécessaire au maniement des outils chirurgicaux nécessite de la pratique. L'emploi de systèmes d'entraînement basés sur de la simulation peut alors constituer une réponse à ce problème de formation. L'idée est alors d'apporter, par le biais d'une simulation numérique, des informations (comme par exemple la position des organes sur lesquels ils interagissent ou encore les forces exercées sur eux) à des dispositifs physiques sur lesquels l'apprentis retrouve les sensations et situations perçues dans la réalité, ainsi qu'un rendu visuel des mécanismes physiologiques engendrés lors du geste permettant d'améliorer leur compréhension par l'apprentis.

Le challenge que nous essayons ainsi de relever consiste à proposer des simulations restituant un comportement réaliste des organes et cela en temps interactif. C'est-à-dire que nous ne recherchons pas une précision absolue du comportement biomécanique des tissus mous, mais à partir d'une connaissance théorique nous essayons de simplifier les simulations pour diminuer leur temps de calcul tout en conservant un comportement global pertinent pour une intégration dans un simulateur d'apprentissage de gestes médicaux.

Pour diminuer le temps de calcul des simulations, nous avons abordé au fil des années le problème sous différents aspects. Nous avons ainsi travaillé sur la proposition de modèles physiques permettant de gérer des objets mixtes, c'est-à-dire décomposés en éléments de géométries et de lois de comportement différents ; sur l'apport d'un modèle topologique complexe permettant d'optimiser les changements topologiques subis par l'objet durant sa simulation ; sur la définition des interactions exercées au sein de l'objet qui modélisent son comportement physique avec notamment la mise en place d'un modèle surfacique permettant de gagner en complexité par rapport à un modèle volumique ; et sur la parallélisation des traitements effectués durant la simulation pour gagner encore un peu plus en temps de calculs. L'idée sous-jacente est de jouer sur ces différents aspects au cours de la simulation par l'emploi de critères mis en place en fonction de l'application.

Interactive simulation of deformable objects for the design of training simulators of medical-surgical gestures

Currently the medical training mainly based on observation rather than practice because learning done directly from the patient poses ethical problems or medical-legal. Or dexterity required in the use of surgical tools requires practice. The use of training systems based on the simulation can then be an answer to this training problem. The idea is then to provide through a numerical simulation information (such as the position of the organs in which they interact, or the forces exerted upon them) to physical devices on which the apprentice found sensations and situations perceived in reality and a visual rendering of the physiological mechanisms generated during the gesture to improve their understanding by the trainees.

The challenge is to provide realistic simulations of organs in interactive time. That is to say, we do not seek absolute accuracy of the biomechanical behavior of soft tissue, but from a theoretical knowledge we try to simplify the simulations to reduce their computation time while maintaining a relevant global behavior integration into a learning simulator of medical procedures.

To reduce the computation time of simulations, we tackled over the years the problem from different aspects. We have worked on the proposal of new physical models to manage mixed objects, that is to say broken down into elements geometries and different behavior laws ; the contribution of a complex topological model to optimize the topological changes undergone by the subject during his simulation ; on defining interactions held within the object that model the physical behavior including the setting up of a surface model to gain complexity over a solid model ; and the parallelization of processing performed during the simulation to still gain a little more time calculations. The underlying idea is then to play on these aspects during the simulation by using criteria established according to the intended application.