

THESE

présentée à

L'UNIVERSITE D'EVRY-VAL D'ESSONNE

par

Nadjet TALBI

pour obtenir le diplôme de

**DOCTORAT DE L'UNIVERSITE
D'EVRY VAL D'ESSONNE**

Spécialité: Robotique

**RÉSOLUTION DU CONTACT FROTTANT ENTRE OBJETS
DÉFORMABLES EN TEMPS RÉEL ET AVEC RETOUR
HAPTIQUE**

*Thèse soutenue le
11 Décembre 2008*

JURY

M. Philippe Meseure	Professeur à l'université de Poitiers	Rapporteur
M. François Peyraut	Professeur à l'université de Belfort-Montbéliard	Rapporteur
M. Pierre Villon	Professeur à l'université de Compiègne	Examineur
M. Christian Duriez	Chargé de Recherche à l'INRIA Lille	Examineur
M. François Faure	Maître de Conférences à l'université de Grenoble I	Examineur
M. Zhi-Qiang Feng	Professeur à l'université d'Evry Val d'Essonne	Examineur
M. Pierre Joli	Maître de Conférences à l'université d'Evry Val d'Essonne	Co-directeur
M. Abderrahmane Kheddar	Directeur de Recherche CNRS	Directeur thèse

Remerciements

"A ma Maman, mon Papa, Sabrina, Nadjib, khirreddine, Sarah, Hamoud, Souhila"

"A Sébastien mon Grand Amour"

Je tiens à remercier les membres du jury (Meseure Philippe, Peyraut François, Duriez Christian, Villon Pierre, et Faure François) pour avoir accepté de rapporter et d'examiner mon manuscrit.

Je tiens à remercier chaleureusement

Abderrahmane Kheddar, mon directeur de thèse, pour m'avoir recruté dans son équipe et permis de réaliser cette thèse.

Pierre Joli, mon co-directeur de thèse, pour son aide précieuse, sa présence, son soutien et ses conseils tout au long de ces années.

Zhi-Qiang FENG, professeur à l'université d'Evry pour ses conseils, son aide, et ses explications qui m'ont souvent éclairées dans mon travail.

Etienne Colle, co-directeur du laboratoire IBISC pour ses conseils et ses encouragements

Un grand merci pour toute l'équipe RVH pour cette bonne ambiance qui régnait

Je remercie également l'équipe Evasion de l'INRIA de Grenoble pour leur accueil chaleureux lors de mon séjour chez eux.

Je remercie ensuite tout le personnel de l'université et de l'IUP d'Evry (administratif et enseignant) qui m'a permis d'effectuer cette thèse dans de bonnes conditions (Annie, Sylviane, Florent, ...).

Un infini merci à mes parents, à mes frères et soeurs

Un grand merci chaleureux à ma tante Dalila et mon oncle Samir et sa femme Yamina pour leur soutien et leurs conseils.

A tous mes amis, que je remercie pour leurs encouragements, Amel, Thierry, Annick, Jocelyn, Franck, Belkacem, Asma, (Pardon pour ceux dont le prénom n'apparaît pas)

Mes derniers remerciements plus qu'affectueux pour Sébastien Delarue avec qui j'ai vécu les plus beaux moments de ma vie ... pour cette merveilleuse histoire des milles et une nuits que j'espère éternelle... .

Résumé

Cette thèse traite de la résolution de contacts multiples frottants entre objets déformables dans le cadre de la simulation interactive avec retour d'effort.

Le contexte de ces travaux est l'apprentissage des gestes médicaux-chirurgicaux par l'intermédiaire d'une interface haptique. Dans ce type de contexte, il est nécessaire de prendre en compte la déformation des organes, les interactions organes/organes et les interactions outils/organes.

Ces interactions sont du type contact avec frottements (secs ou visqueux). On utilise la Méthode des Eléments Finis appliquée aux équations de la Mécanique des Milieux Continus qui est certainement la méthode la plus rigoureuse pour modéliser le comportement linéaire ou non linéaire des tissus mous.

On considère que les objets se déforment dans un état d'équilibre quasi-statique car les gestes opératoires sont lents et les objets manipulés sont de faible masse.

Afin de résoudre ces équations d'équilibre quasi-statique dont les inconnues sont les forces de contact et les déplacements élastiques, on utilise la méthode incrémentale de Newton-Raphson.

Plusieurs formulations, basées principalement sur le lagrangien augmentée (pseudo-potentiels, bi-potentiel), sont présentées pour modéliser les lois de contact frottant (Signorini et frottement de Coulomb) entre objets déformables dans un cadre très général (statique/dynamique, grandes déformations, schémas numériques : explicite, implicite).

Afin de séparer le calcul des forces de contact du calcul des déplacements élastiques, on utilise la méthode de flexibilité laquelle permet de considérer un solveur de force de contact indépendamment du solveur des déplacements.

Le solveur de forces de contact utilise des techniques de résolution numérique type «Gauss-Seidel» et «Uzawa» et permet un contrôle robuste de la solution en fonction de la précision demandée. Une version de ce solveur a été développée dans laquelle, il n'est pas nécessaire de connaître les modèles utilisés au niveau des déformations.

Un simulateur temps réel avec interface haptique appelé «HapCo» a été mis au point et a permis de valider les concepts introduits dans cette thèse.

Abstract

This thesis deals with the solving of multiple friction contact problems between deformable bodies in the specific area of interactive simulation with force feedback.

The general context of this work is the training of medical-surgical gestures via an haptic device. In this type of context, it is often necessary to take into account the global deformations of the organs and the local contact forces due to interactions soft tissues/soft tissues or to interactions tools/soft tissues.

These interactions are the consequences of the friction contact force (dry or viscous). We use the Finite Element Method based on the mechanics theory of continuous media which is certainly the most rigorous method to model linear or non linear behaviours of soft tissues.

We also consider that the objects are deformed following the quasi-static equilibrium hypothesis because the operational gestures are slow and the handled objects have low masses.

In order to solve these quasi-static equilibrium equations whose unknowns are the forces of contact and the elastic displacements, we use the incremental method of Newton-Raphson.

We present various formulations, mainly based on the augmented lagrangian method (pseudo-potentials, bi-potential) to model the contacts laws (Signorini and friction of Coulomb) between deformable objects within a general framework (dynamic, large deformations, explicit or implicit numerical resolution).

In order to disconnect the computation of the contact forces from the computation of the elastic displacements, we use the flexibility method which gives the possibility to consider a contact solver independently of the displacement solver.

The contact solver is based on «Gauss-Seidel» and «Uzawa» techniques and allows a robust control of the contact forces relative to a given numerical precision. A version of this solver has been realised in which it is not necessary to know the models used for the deformations.

A real-time simulator with haptic device has been produced and allowed to test the concepts introduced in this thesis.

Table des matières

Remerciements	3
Résumé	5
Introduction	17
1. Etat de l'art	23
1.1 Introduction	23
1.2 Quelques exemples de simulateurs interactifs basés sur les lois de la physique	24
1.3 Schéma général de fonctionnement d'un simulateur interactif avec retour haptique	28
1.4 Modélisation d'objets déformables	29
1.4.1 Modèle masses-ressorts	31
1.4.2 Méthode des éléments finis	35
1.5 Détection de collision	50
1.5.1 Recherche de proximité (broad-phase)	51
1.5.2 Volumes englobants	52
1.5.3 Détection de collisions pour les objets déformables	55
1.6 Modélisation des contacts multiples	61
1.7 Conclusion	62
2. Modélisation et formulation du contact frottant entre objets déformables	65
2.1 Introduction	65
2.2 Principes généraux de résolution d'un modèle discret	67
2.2.1 Discrétisation spatiale	67
2.2.2 Discrétisation temporelle	68
2.2.3 Conclusion	71
2.3 Formulation classique des lois de contact de frottements secs	71
2.3.1 Définition du repère local de contact	71
2.3.2 Conditions de Signorini	72
2.3.3 Loi de Coulomb	73
2.3.4 Conclusion	74
2.4 Autres formulations des lois de contact de frottements secs	75
2.4.1 Formulation par des fonctions de pénalité	75
2.4.2 Formulations par des multiplicateurs de Lagrange	77
2.4.3 Formulations lagrangiennes augmentées	80
2.4.4 Conclusion	83
2.5 Mise en place des équations d'équilibre avec contacts multiples	85
2.6 Principe de la méthode de flexibilité	87

2.7	Résolution par équilibres successifs des forces de contact: algorithme de Gauss-Seidel	90
2.8	Résolution locale des forces de contact	91
2.8.1	Méthode de Newton	92
2.8.2	Méthode d'Uzawa	93
2.9	Conclusion	94
3.	Résolution du contact frottant entre objets déformables: Expérimentations et Comparaisons	97
3.1	Introduction	97
3.2	Présentation du banc de test	99
3.3	Le calcul de la raideur locale ρ	101
3.4	Comparaison entre les deux formulations «pseudo-potentielles» et «bi-potentielle»	102
3.5	Recherche d'un ρ optimal	107
3.6	Approche Uzawa locale	110
3.7	Expérimentations en fonction des modules d'Young	111
3.7.1	Contact frottant entre aluminium et tissu-mou	112
3.7.2	Contact frottant entre aluminium et caoutchouc	113
3.7.3	Contact frottant entre deux tissus mous	116
3.8	Expérimentations en fonction du coefficient de frottement	119
3.9	Expérimentation en fonction du nombre de points en contact	121
3.10	Conclusion	123
4.	HapCo: un prototype de simulateur temps réel avec retour haptique traitant des contacts multiples entre objets déformables.	125
4.1	Introduction	125
4.2	Solveur de force de contact générique	126
4.2.1	Principe algorithmique	126
4.2.2	Expérimentations numériques	129
4.3	SOFA: exemple d'intégration du solveur de contact frottant	136
4.4	Description du simulateur HAPCO intégrant le retour d'effort	142
4.4.1	Architecture et fonctionnement du simulateur HapCo	143
4.4.2	Interfaces graphiques du simulateur HapCo	146
4.4.3	Expérimentations haptiques	149
4.5	Conclusion	154
	Conclusion	157
	Références bibliographiques	159

Table des figures

1.1	Exemples de films ayant eu recours à l'animation basée sur la physique. . .	25
1.2	La chirurgie abdominale en laparoscopie (KISMET Medical.)	26
1.3	A droite: Simulateur de conduite développé par INRETS et IBISC. A gauche: Simulateur de vol développé par la NASA.	27
1.4	Plateforme Perf-RV (INRIA Rennes/CEA List): Test de montage d'un moteur lève vitre à l'intérieur d'une portière. Test d'ouverture des portières (voiture Renault).	28
1.5	Schéma fonctionnel d'une simulation interactive avec retour haptique. . .	29
1.6	Balle sur un solide déformable d'après le modèle de Terzopoulos.	31
1.7	A droite: Discrétisation d'un objet 2D à l'aide de masses-ressorts. A gauche: un maillage en 3D modèle masse ressort, tiré de [Aul01].	32
1.8	Simulation de laparoscopie.	34
1.9	simulation masse ressort.	35
1.10	Exemples d'éléments finis surfaciques et volumiques.	36
1.11	En haut: Interpolation linéaire ou quadratique sur un élément fini, En bas: plus le degré d'interpolation est élevé, plus l'élément peut se déformer. . .	37
1.12	La transformation iso-paramétrique d'un élément cubique.	38
1.13	Approximation linéaire en rouge d'une loi de comportement (ici élasto-plastique).	40
1.14	Déformation d'un modèle de foie composé de 1500 noeuds avec une fréquence visuelle de 50 Hz et une fréquence de 300 Hz pour le retour de force [CDA99].	44
1.15	Dilatation d'un matériau élastique linéaire (fil de fer) sous l'effet d'une rotation globale [Pic01].	45
1.16	En haut: Deux poutres en grandes déformations d'après [Nes04] (la poutre rouge est en élasticité linéaire), En bas: le modèle linéaire situé à gauche devient faux d'après [ZC00]	46
1.17	L'espace est divisé par une grille uniforme.	51
1.18	A droite: Un arbre octree correspondant à l'espace divisé. A Gauche: Un arbre BSP correspondant à l'espace divisé.	52
1.19	sweep and prune en 2D.	52
1.20	Différents types de volumes englobants d'après [Eri04].	53
1.21	D'après [Lin04] : Hiérarchie des tests d'intersection entre les volumes englobants. A gauche: Après un test positif sur les boîtes englobantes principales, on recommence un test en utilisant les boîtes du second niveau d'un des objets. A droite: la descente dans la hiérarchie permet de localiser précisément la collision par un nombre de tests réduit.	55
1.22	La mise à jour des trois types de volumes englobants en se basant sur leurs fils [Hat06].	57
1.23	La mise à jour peut produire une hiérarchie non optimale [Hat06].	57

1.24	Les résultats à différents niveaux de populations de la stratégie ESPIONS [Hat06].	59
1.25	Exemple de détection de collision selon la méthode à base de séparation et hiérarchie de volumes englobants partielle [Hat06].	61
2.1	Diagramme de la méthode de flexibilité.	67
2.2	Repère local de contact.	72
2.3	Graphe des conditions de Signorini.	73
2.4	Graphe de la loi de Coulomb.	74
2.5	Bloc conditionnel des 3 statuts de contact	75
2.6	cône de Coulomb.	75
2.7	Régularisation des lois de contact	76
2.8	Cône de Coulomb à huit facettes	78
2.9	Projection sur le disque de Coulomb	83
2.10	Projection sur le cône de Coulomb	83
2.11	Chaque contact relie deux points que l'on interpole sur les triangles de la surface des objets déformables.	86
2.12	L'algorithme générale de la méthode de Flexibilité.	90
3.1	Stratégies utilisées pour le calcul des forces de contact avec frottement.	99
3.2	Test du solveur de contact frottant sec entre deux objets déformables, l'objet en rouge est fixe, l'objet en vert est mobile et soumis à des contraintes en déplacements imposées.	100
3.3	A gauche: définition du parcours pénétration, initialement, les deux objets déformables se touchent. A droite: définition du parcours glissement, initialement, l'objet mobile pénètre de 0.5 cm dans l'objet encastré. 1 pas = 0.1 cm.	101
3.4	Comparaison en temps CPU entre les solveurs Uzawa global bi-potentiel et Uzawa global* bi-potentiel pendant le parcours de pénétration (interaction aluminium/tissu mou).	103
3.5	Comparaison du nombre d'itérations Gauss-Seidel entre les solveurs Uzawa global bi-potentiel et Uzawa global* bi-potentiel pendant le parcours de pénétration (interaction aluminium/tissu mou).	103
3.6	Norme du vecteur force de contact issu du solveur Uzawa global* bi-potentiel pendant le parcours de pénétration (interaction aluminium/tissu mou).	104
3.7	Ecart entre $\mathbf{r}_{\mathbf{GB}}$ issu du solveur Uzawa global bi-potentiel et $\mathbf{r}_{\mathbf{G}^*\mathbf{B}}$ issu du solveur Uzawa global* bi-potentiel pendant le parcours de pénétration (interaction aluminium/tissu mou).	104
3.8	Nombre de contacts glissants issu des solveurs Uzawa global pseudo-potentiel, Uzawa global bi-potentiel et Uzawa global* bi-potentiel pendant le parcours de pénétration (interaction aluminium/tissu mou).	105

3.9	Comparaison en temps CPU entre les solveurs Uzawa global pseudo-potentiel, Uzawa global bi-potentiel et Uzawa global* bi-potentiel pendant la parcours de glissement (interaction aluminium/tissu mou).	106
3.10	Norme des forces de contact avec frottement issue du solveur Uzawa global* bi-potentiel pendant le parcours de glissement (interaction aluminium/tissu mou).	106
3.11	Ecart entre $\mathbf{r}_{\mathbf{GB}}$ issu du solveur Uzawa global bi-potentiel, $\mathbf{r}_{\mathbf{GP}}$ issu du solveur Uzawa global pseudo-potentiel et $\mathbf{r}_{\mathbf{G*B}}$ issu du solveur Uzawa global* bi-potentiel pendant le parcours de glissement (interaction aluminium/tissu mou).	106
3.12	Nombre de contacts glissants issu des solveurs Uzawa global bi-potentiel et Uzawa global* bi-potentiel pendant le parcours de glissement (interaction aluminium/tissu mou).	107
3.13	Comparaison en temps CPU entre les différents solveurs pendant le parcours de pénétration (interaction aluminium/tissu mou).	108
3.14	Comparaison en nombre d'itérations Gauss-Seidel entre les différents solveurs pendant le parcours de pénétration (interaction aluminium/tissu mou).	108
3.15	Norme des forces de contact avec frottement issus du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/tissu mou).	109
3.16	Ecart entre le vecteur force de réaction avec frottement issu du solveur Uzawa global avec $\rho_{min} \times 2$ et celui issu du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/tissu mou).	109
3.17	Ecart entre le vecteur force de contact avec frottement issu du solveur Uzawa global* avec $\rho^\alpha/2$ et celui issu du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/tissu mou).	109
3.18	Evolution du nombre de contacts glissants issu des différents solveurs pendant le parcours de pénétration (interaction aluminium/tissu mou).	110
3.19	Comparaison entre les deux solveurs Uzawa global et Uzawa global* en fonction de la valeur de ρ	110
3.20	Temps CPU moyen en ms des différents solveurs pendant le parcours de pénétration (interaction aluminium/tissu mou).	112
3.21	Ecarts entre les différents vecteurs force de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/tissu mou).	112
3.22	Temps CPU moyen en ms des différents solveurs pendant le parcours de glissement (interaction aluminium/tissu mou).	112
3.23	Ecarts entre les différents vecteurs force de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de glissement (interaction aluminium/tissu mou).	113
3.24	Temps CPU moyen en ms des différents solveurs pendant le parcours de pénétration (interaction aluminium/caoutchouc).	114

3.25	Ecarts entre les différents vecteurs force de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/caoutchouc).	114
3.26	Norme du vecteur force de contact avec frottement issu du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/caoutchouc).114	
3.27	Temps CPU moyen en ms des différents solveurs pendant le parcours de glissement (interaction aluminium/caoutchouc).	115
3.28	Ecart entre les différents vecteurs forces de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de glissement (interaction aluminium/caoutchouc).	115
3.29	Norme du vecteur force de contact avec frottement issu du solveur Uzawa global* pendant le parcours de glissement (interaction aluminium/caoutchouc).115	
3.30	Temps CPU moyen en ms des différents solveurs pendant le parcours de pénétration (interaction tissu mou/tissu mou).	116
3.31	Ecarts entre les différents vecteurs force de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de pénétration (interaction tissu mou/tissu mou).	117
3.32	Norme du vecteur force de contact avec frottement pendant le parcours de pénétration (interaction tissu mou/tissu mou).	117
3.33	Temps CPU moyen en ms des différents solveurs pendant le parcours de glissement (interaction tissu mou/tissu mou).	117
3.34	Ecart entre les différents vecteurs force de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de glissement (interaction tissu mou/tissu mou).	118
3.35	Norme du vecteur force de contact avec frottement pendant le parcours glissement.	118
3.36	Temps CPU moyen (ms) des différentes solveurs en fonction du coefficient de frottement pendant le parcours pénétration (interaction aluminium/tissu mou).	119
3.37	Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours glissement (interaction aluminium/tissu mou).	119
3.38	Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours pénétration (interaction aluminium/caoutchouc).	119
3.39	Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours glissement (interaction aluminium/caoutchouc).	120
3.40	Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours pénétration (interaction tissu mou/tissu mou).	120

3.41	Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours glissement (interaction tissu mou/tissu mou).	120
3.42	Norme des vecteurs forces de contact issus du solveur Uzawa global* en fonction du coefficient de frottement pendant le parcours pénétration et glissement (interaction aluminium/tissu-mou).	121
3.43	Evolution du temps de calcul de l'algorithme Uzawa global* en fonction du nombre de contacts (interaction aluminium/tissu mou).	122
3.44	Evolution du temps de calcul de l'algorithme Uzawa globale en fonction du nombre de contacts (interaction aluminium/caoutchouc).	122
3.45	Evolution du temps de calcul de l'algorithme Uzawa global en fonction du nombre de contacts (interaction tissu mou/tissu mou).	122
3.46	Banc d'essai impliquant 66 contact, à gauche : parcours pénétration, à droite : parcours glissement.	123
4.1	Détection de collision entre un objet impacteur et un objet obstacle. . . .	128
4.2	L'algorithme général de résolution des équations d'équilibre sans construction de l'opérateur de Delassus.	129
4.3	Banc d'essai impliquant 36 contacts, à gauche : parcours pénétration, à droite : parcours glissement.	130
4.4	Comparaison en temps CPU entre le solveur de force de contact avec et sans Delassus (interaction aluminium/tissu mou).	131
4.5	Comparaison en nombre de contacts glissants (en haut) et en nombre de contacts adhérents (en bas) issus des deux solveurs de forces de contact avec et sans Delassus (interaction aluminium/tissu mou).	131
4.6	Norme du vecteur force de contact issue du solveur R avec et sans Delassus (interaction aluminium/tissu mou).	132
4.7	Comparaison en temps CPU entre le solveur de force de contact avec et sans Delassus (interaction aluminium/caoutchouc).	132
4.8	Comparaison en nombre de contacts glissants (en haut) et en nombre de contacts adhérents (en bas) issus des deux solveurs de forces de contact avec et sans Delassus (interaction aluminium/caoutchouc).	133
4.9	Norme du vecteur force de contact issue du solveur R avec et sans Delassus (interaction aluminium/caoutchouc).	133
4.10	Comparaison en temps CPU entre le solveur de force de contact avec et sans Delassus (interaction tissu mou/tissu mou).	133
4.11	Comparaison en nombre de contacts glissants (en haut) et en nombre de contacts adhérents (en bas) issus des deux solveurs de forces de contact avec et sans Delassus (interaction tissu mou/tissu mou).	134
4.12	Norme du vecteur force de contact issue du solveur R avec et sans Delassus (interaction tissu mou/tissu mou).	134
4.13	Temps CPU moyen (ms) des différents modules dans le cas du calcul avec et sans Delassus pour 36 contacts.	135

4.14	Temps CPU moyen (ms) des différents modules dans le cas du calcul avec et sans Delassus pour 16 contacts.	135
4.15	En haut, deux solides en interaction. En bas, la représentation dans SOFA de cette scène sous forme de graphe.	138
4.16	Représentation multimodale d'un objet dans la plate forme SOFA.	138
4.17	Contact frottant entre une barre soumise à son poids et un foie dans SOFA.	139
4.18	Le graphe SOFA associé à la scène barre foie	140
4.19	Le dispositif haptique PHANTOM Desktop à gauche et PHANTOM Omni à droite de SensAble technologies.	142
4.20	Architecture de la simulation HapCo.	144
4.21	Couplage entre le dispositif haptique et la simulation	144
4.22	la classe Solid du simulateur HapCo.	145
4.23	Fichier descriptif de l'objet déformable décrit par la méthode des éléments finis en élasticité linéaire.	146
4.24	La scène virtuelle 3D et l'interface des paramètres.	147
4.25	Tableau récapitulatif des différents critères et leurs valeurs, testés dans les expérimentations haptiques	150
4.26	Forces haptiques restituées à l'interface haptique pendant un parcours pénétration, interaction aluminium/tissu mou $E = 45$ Mpa avec 36 contacts.	153
4.27	Forces haptiques restituées à l'interface haptique pendant un parcours pénétration, interaction aluminium/tissu mou $E = 45$ Mpa avec 16 contacts.	154

Liste des tableaux

Introduction

De nos jours, la demande en simulateurs réalistes et interactifs grandit considérablement : la mécanique, la robotique, l'informatique, la croissance de la puissance des ordinateurs et les nouvelles techniques d'acquisition participent de plus en plus au développement de ces simulateurs. Ces derniers doivent être réalistes (rendu visuel 3D, sonore, tactile et haptique), précis et basés sur les lois de la physique. Les utilisateurs doivent aussi pouvoir interagir avec la simulation à travers des dispositifs mécaniques (les interfaces haptiques par exemple) et obtenir une réponse en un temps très limité (temps réel) afin de reproduire le plus fidèlement possible des phénomènes réels. L'interaction entre objets déformables représente une partie non négligeable du monde réel que les applications de la réalité virtuelle tentent de reproduire.

Nous trouvons ce type d'objet dans les simulateurs chirurgicaux ayant pour objectif la formation et l'entraînement aux gestes opératoires, l'acquisition des réflexes correspondants ainsi que la planification chirurgicale sans passer par des opérations sur des animaux ou sur des cadavres. Nous les trouvons aussi dans les effets spéciaux cinématographiques et les jeux vidéo ainsi que dans différents problèmes de mécanique lié à la Conception Assistée par Ordinateur (CAO).

Dans le cadre de cette thèse, nous nous intéressons principalement à l'apprentissage des gestes médicaux-chirurgicaux par l'intermédiaire d'une interface haptique. Une interface haptique étant un dispositif mécanique restituant en temps réel à l'opérateur un effort calculé lors de la simulation. Dans ce type de contexte, il est souvent nécessaire de prendre en compte la déformation des organes, les interactions organes/organes et les interactions outils/organes. Ces interactions sont du type contact avec frottements (secs ou visqueux). Il existe de nombreuses applications médicales imposant ce type de cahier des charges. Prenons par exemple la chirurgie mini-invasive endovasculaire où il s'agit d'introduire un long cathéter flexible par voie artérielle. L'opération est rendue complexe par la présence de frottements sur des zones importantes de contact entre le cathéter et les parois artérielles (et aussi par un retour visuel très dégradé). Citons également la manipulation du fœtus au moment de la délivrance (contact entre la tête du fœtus et les muscles pelviens), ou encore la chirurgie coelioscopie (chirurgie mini-invasive abdominale) dans laquelle on manipule des outils longs et rigides à travers des trocards ce qui nécessite de créer un chemin d'accès en écartant ou en déplaçant des organes pour atteindre la zone à traiter (création de surfaces de compression et d'adhérence/glisement entre organes). Le retour d'effort restitué via l'interface haptique sera très différent selon le nombre de points de contact et le type de frottement (secs ou visqueux). Le cas le plus sévère est celui de contacts multiples avec frottements secs car c'est celui qui introduit le plus d'instabilité numérique.

De nombreux travaux existent sur la modélisation des corps déformables. Certains modèles non physique sont basés uniquement sur des mécanismes cinématiques comme les «cubes actifs» ou les «chain-mail». La précision de ces modèles n'étant pas prouvée on préfère souvent utiliser des modèles plus complexes basés sur des lois de comportement. La définition d'une loi de comportement est généralement issue d'observations, de mesures rhéologiques et de connaissances d'experts. Le problème majeur réside dans le domaine de validité des lois supposées. Si l'on choisit une loi de comportement élastique linéaire (déformations réversibles) souvent utilisée en raison des faibles temps de calculs, alors, le domaine de validité est limité aux petites déformations. Une approche plus raisonnable pour modéliser les tissus mous est de considérer les grandes déformations. Dans ce cas il est nécessaire d'utiliser des lois de comportement non linéaires d'hyperélasticité compressibles ou incompressibles (Blatz-Ko, Gent, Ogden, Mooney-Rivlin par exemple) ou de visco-élasticité. L'incompressibilité de certains organes vient de leur caractère biphasique (fluide/solide). Il faut également prendre en compte l'anisotropie des tissus organiques. Une loi classiquement utilisée dans les simulateurs chirurgicaux est la loi de St Venant-Kirchhoff qui n'est autre qu'une extension de l'élasticité linéaire appliquée aux grandes déformations et qui représente là encore une approximation forte du comportement des tissus mous.

La géométrie des organes étant complexe et continue, pour construire un modèle numérique, il est nécessaire de procéder par une méthode de discrétisation spatiale. La méthode masse-ressort est souvent utilisée dans les simulateurs chirurgicaux pour sa simplicité de mise en oeuvre. Elle consiste à utiliser un modèle géométrique discret surfacique ou volumique dans lequel les sommets sont considérés comme étant des masses et les arêtes comme des ressorts. La contrainte d'incompressibilité et les valeurs des constantes paramétriques des ressorts restent difficiles à mettre en oeuvre si l'on veut respecter les lois de comportement des tissus. Ce modèle a permis de créer des simulateurs chirurgicaux fonctionnant en temps réel, mais peu de données sur leur précision sont fournies.

La Méthode des Eléments Finis appliquée aux équations de la mécanique des milieux continus est certainement la méthode la plus rigoureuse pour modéliser le comportement linéaire ou non linéaire des tissus mous car les lois de comportement sont directement intégrées dans la constitution des modèles. Le principe est de décomposer l'organe étudié en une grande quantité de petits éléments finis, typiquement des tétraèdres pour les objets 3D, puis de calculer les déformations et les contraintes dans le volume tout entier en fonctions des déplacements aux noeuds du maillage. L'inconvénient est le nombre important de degrés de liberté (ddl) générés dans ce type de modélisation. Citons également la méthode des éléments de frontière basée sur les équations de la MMC et sur le calcul des équations intégrales. Dans ce cas, la discrétisation est uniquement surfacique ce qui réduit considérablement le nombre de degré de liberté du modèle. Cependant cette méthode nécessite la connaissance au préalable des fonctions de Green non définies dans le cas non linéaire des lois de comportement et des grandes déformations. Plus récemment un compromis, qui limite la discrétisation volumique coûteuse en ddl, consiste à modéliser le comportement élastique uniquement à la surface suivant une certaine profondeur et à

considérer un fluide incompressible à l'intérieur. Il peut s'agir là d'une approximation suffisante pour certains tissus mous comme dans l'exemple de la cuisse. Citons également les méthodes sans maillage, comme la méthode des sphères finies associée à des techniques de collocation. Cette méthode à l'avantage d'enrichir un maillage au niveau de l'interaction entre l'outil et l'organe sans qu'il soit nécessaire de remailler la zone de contact.

Il est difficile de trancher de manière objective sur l'efficacité de tel ou tel modèle si l'on doit prendre en compte simultanément la précision des résultats numériques et la rapidité des calculs. Bien souvent ces deux critères s'opposent mais l'approche éléments finis est certainement celle qui conduit à une meilleure prise en compte des phénomènes physiques. Rien qu'en partant de cette approche, il est possible de faire des choix multiples de modèles différents afin de réduire le coût numérique suivant le contexte (approche quasi-statique ou dynamique, schéma numérique d'intégration implicite ou explicite, méthode de résolution type Newton-Raphson ou du type gradient conjugué, matrices pleines ou condensées, réduction de modèle ou non). Citons le modèle masse tenseur comme un de ces modèles dérivés de la méthode des éléments finis où la matrice de masse est condensée et les forces internes aux noeuds sont dérivées directement à partir de l'expression de l'énergie interne élément par élément sans assemblage matriciel.

Quelque soit le modèle choisi, il doit prendre en compte la déformation des corps en contact de manière réaliste afin de restituer un rendu haptique cohérent entre la perception mentale que l'on a des objets manipulés (ou que l'on touche) et la perception réelle ressentie à travers l'interface haptique utilisée. Le challenge est considérable car il est admis qu'il faut une fréquence de calcul supérieure à 300 Hz (voir 1kHz pour certaine application que nous qualifierons de «raides»). Il est donc nécessaire d'optimiser les trois phases suivantes de calculs intervenant dans le processus de simulation:

- le calcul des déformations et des déplacements induits,
- le calcul de détection de collision,
- le calcul des forces de contact.

L'objectif de cette thèse est d'optimiser cette dernière phase dans le cas de contacts multiples et simultanés avec frottement. L'idéal étant de mettre au point un «solveur» de force de contact indépendant des techniques utilisées dans les autres phases de calcul et dont on peut contrôler la stabilité numérique en fonction de la puissance de calcul disponible, des propriétés de l'interface haptique (force maximale, impédance mécanique de l'interface, ...) et de la contrainte temps réel lié au rendu haptique.

Pour cela un banc de test de simulation a été développé, il est composé de deux objets déformables discrétisés par la méthode des éléments finis dont l'un peut être manipulé via une interface haptique type «Phantom» (SensAble Technologies) et peut ainsi entrer en collision avec le second qui est fixe. La modélisation des déformations est physiquement

«réaliste» car elle est basée sur la théorie des milieux continus. La formulation des équations d'équilibre est déduite de la méthode classique des éléments finis en élasticité linéaire. Un module de détection de collision a été mis au point afin de détecter toute collision potentielle entre les deux objets déformables. Il permet également de localiser tous les points de contact (contact multiples et simultanés) ainsi que toutes les données nécessaires pour résoudre les forces de contact (les normales, les coefficients barycentriques des points de projection). Parce que nous considérons les effets inertiels comme négligeables, nous avons considéré que les objets se déforment dans un état d'équilibre quasi-statique. En effet, on se place dans le cadre de la conception de simulateur médicaux dans lesquels les gestes opératoires sont lents et les objets manipulés de faible masse. Dans notre simulateur, on impose un mouvement de corps rigide à l'un des deux corps. Ce mouvement sert de mouvement de référence autour duquel les déformations de ce même corps sont calculées. Cette approche qualifiée par certains chercheurs de corotationnelle globale a l'avantage de pouvoir séparer les grands déplacements dus au mouvement de corps rigide (dont le mouvement est ici imposé), des petits déplacements dus à la déformation.

Afin de résoudre les équations d'équilibres quasi-statiques dont les inconnues sont les forces de contact et les déplacements élastiques (dus aux déformations), on utilise le cadre général de la méthode incrémentale de Newton-Raphson pour résoudre des systèmes d'équations algébriques. Cette formulation générale permet de formaliser n'importe quel problème de mécanique qu'il soit linéaire (petits déplacements, petites déformations) ou non linéaire (grands déplacements, grandes déformations) en dynamique ou en statique. Il faut pour cela calculer les matrices tangentes correspondant au type de problème à traiter. Dans notre cas (petites déformations, élasticité linéaire, grands déplacements imposés), notre problème de mécanique est linéaire relativement aux déplacements élastiques. La résolution s'effectue donc en un seul incrément de déplacement pour chaque pas de chargement (pas de force extérieure ou de déplacement imposé). La matrice tangente est constituée des matrices de raideur des deux corps qui sont supposées constantes (élasticité linéaire).

Dans le cas où il y a des points de contact, les forces de contact induites sont calculées de manière implicite par un processus itératif afin de respecter les conditions de non inter-pénétration (condition de Signorini) et les conditions d'adhérence ou de glissement (Lois de Coulomb). Ce processus itératif nécessite de construire une matrice de projection appelé matrice de Delassus (ou matrice de flexibilité) permettant d'établir le système minimal d'équations (dérivant des équations d'équilibre quasi-statique) que doivent satisfaire les forces de contact. Cette matrice va nous permettre d'en déduire un coefficient de raideur «fictif» nécessaire dans la résolution du contact (avec et sans frottement) et d'assurer la convergence du processus itératif. Les équations du contact sont formulées à partir des conditions de Signorini et de lois de frottement de Coulomb en cas de contact avec frottement. Ces équations représentent des contraintes que doivent respecter aux points de contact les déplacements élastiques et les forces de contact. Ceci implique une résolution à partir de méthodes numériques d'optimisation sous contraintes : LCP, Lagrangien augmentée (pseudo-potentiels, bi-potentiel).

Cette thèse se décline en quatre chapitres :

- Le premier chapitre présente un état de l’art autour du thème de la simulation interactive et des problématiques associées comme la modélisation des corps déformables, la détection de collision et la résolution de contacts multiples.
- Le deuxième chapitre concerne la formulation d’un problème de contact entre objets déformables. Plusieurs méthodes sont envisagées conduisant à la mise en forme de plusieurs algorithmes de résolution en contrainte.
- Le troisième chapitre présente les tests numériques des différents algorithmes définis dans le chapitre 2. Ces essais numériques ont été réalisés sur le simulateur HAPtique COntact (HAPCO) que nous avons développé dans notre laboratoire. Ils ont été effectués en «boucle ouverte» c’est à dire à partir de parcours prédéfinis et activés par clavier sans retour Haptique.
- Le quatrième chapitre présente le logiciel HAPCO, le solveur modulaire introduit dans la plateforme SOFA développée par l’équipe Evasion de l’INRIA Grenoble. Il traite également du rendu haptique à partir d’une évaluation effectuée auprès de 7 personnes étrangères à nos travaux.

Nous terminerons par une conclusion générale qui synthétisera les résultats obtenus et présentera des perspectives de recherche.

Etat de l'art

1.1 Introduction

Les modèles déformables existants reposent sur un modèle géométrique surfacique ou volumique. L'évolution du modèle géométrique dépend ensuite des propriétés de déformation qu'on lui confère. On distingue les modèles déformables physiques et non physiques selon qu'ils soient basés ou non sur les équations de la physique définissant leur comportement. Dans cette étude, nous nous intéressons à l'interaction d'objets déformables basés sur des modèles physiques discrets comme les systèmes «masse-ressort» ou des modèles basés sur la Mécanique des Milieux Continus (MMC) et discrétisés ensuite par la Méthode des Eléments Finis (MEF). Des objets déformables en interaction impliquent des collisions à détecter, des déformations à calculer ainsi que la réponse adéquate liée à la force de contact pour le dispositif de retour d'effort. Une cadence élevée de cette réponse est de grande importance pour une interaction haptique réaliste entre l'opérateur et l'environnement virtuel.

La prise en compte du contact frottant entre objets déformables (avec plusieurs régions de contact) est importante afin de savoir comment ils interagissent. Il est évident que la force de contact joue un rôle fondamental dans le comportement de la structure: sa déformation, son mouvement, la distribution des efforts, etc. La modélisation du phénomène du contact frottant entre des objets «déformables ou pas» est en soit non linéaire et pose de sérieuses difficultés: conceptuelles, mathématiques et informatiques. Malheureusement, très peu de problèmes peuvent être traités de manière analytique. Il est donc nécessaire de développer des méthodes numériques adaptées à ce type de problèmes.

Plusieurs applications de Réalité Virtuelle (R.V.) impliquent des interfaces haptiques et un retour d'effort: simulateurs chirurgicaux, simulateur d'entraînement au pilotage, simulateur d'assemblage pour prototypage virtuel... Un retour haptique de qualité (réaliste) nécessite l'implémentation de modèles physiques ce qui va bien souvent à l'encontre d'une performance « temps réel » (fréquence de calcul optimale $> 300\text{Hz}$ pour des objets mous, $> 1\text{KHz}$ pour des objets durs). Pour cette raison des méthodes d'accélération de calcul sont indispensables.

Un autre domaine où il est nécessaire de développer des méthodes numériques adaptées se situe au niveau de la détection de collisions qui est un goulet numérique d'étranglement bien connu dans les applications de R.V.

Nous allons commencer par présenter, dans cet état de l'art, quelques exemples de simulateurs interactifs basés sur les lois de la physique, nous définirons ensuite un schéma général de fonctionnement d'une simulation interactive avec interface haptique. Nous passerons également en revue différents types de modélisation des corps déformables utilisés pour des simulations réalistes ainsi que différentes techniques de détection de collisions entre objets déformables. Enfin nous terminerons par un bref état de l'art des méthodes utilisées pour traiter les contacts frottants.

1.2 Quelques exemples de simulateurs interactifs basés sur les lois de la physique

L'animation par des modèles physiques consiste à générer automatiquement le mouvement et le comportement de différents objets de façon réaliste en simulant les lois physiques qui gouvernent son mouvement (translation, rotation) et/ou son comportement (forces internes, déformations). C'est une technique qui est de plus en plus populaire. Nous la trouvons désormais dans les jeux vidéo, dans la production des effets spéciaux des films (auparavant, l'industrie audiovisuelle générait le mouvement/comportement des objets de la simulation manuellement) ainsi que dans les simulateurs d'applications médicales. Citons quelques exemples d'utilisation de simulations physiquement réalistes : modélisation de corps rigides (The Hulk, Grand Turismo, Medal of Honor, Half-Life), modélisation de la déformation des muscles et de la peau (Shrek, the Nutty Professor, jurassic Park, the Mummy), modélisation d'éclaboussement d'eau (Shrek, Titanic, Finding Nemo). La figure 1.1 illustre ces exemples d'animations physiquement réalistes, de plus en plus répandues [ESH05].



FIG. 1.1 – Exemples de films ayant eu recours à l'animation basée sur la physique.

De nombreux simulateurs médicaux/chirurgicaux ont vu le jour. Initialement, ces derniers étaient fondés sur des modèles géométriques de structures anatomiques, sans tenir compte de leur réalité physique. La représentation de phénomènes physiques et, plus spécifiquement, la modélisation réaliste des tissus mous, a nettement amélioré la qualité de ces simulateurs. En effet, nous trouvons dans [BNC96], [MC00] et [MDH⁺03] des modélisations de tissus mous basées sur des modèles physiques. Le champ d'applications potentielles de la simulation interactive a pu ainsi s'élargir allant de la planification en neurochirurgie à toute forme de chirurgie minimalement invasive (laparoscopie, cholécystectomie, endovasculaire ...). Afin de simuler de manière réaliste la déformation des tissus mous, il est nécessaire de modéliser avec précision ces déformations mais aussi de les calculer efficacement. La biomécanique permet d'étudier des modèles mathématiques complexes de comportement des tissus vivants. Plusieurs algorithmes pour le calcul temps-réel de corps déformables intégrant un retour d'effort ont été proposés : [ALC99], [CDA00], [LMS03].

La figure 1.2 montre un exemple de simulateur d'entraînement à la chirurgie laparoscopique avec interface haptique proposé par la société Kisnet Medical. La chirurgie laparoscopique consiste à introduire au niveau de la cavité abdominale des instruments rigides de forme allongée à travers de petites incisions (1 cm environ) sur lesquelles sont mis en place des petites canules appelées trocarts. L'abdomen est gonflé par l'injection d'un gaz inerte afin d'augmenter l'espace opérationnel. Un endoscope est également introduit dans un des trocarts afin de restituer sur un moniteur vidéo la vision du mouvement des instruments à

l'intérieur de l'abdomen. L'avantage de ce type d'opération dite mini-invasive est de réduire considérablement les traumatismes du patient et d'accélérer ainsi sa récupération. L'inconvénient est que cela nécessite une parfaite coordination entre la vision et les gestes manuels qui dans ce cas n'est pas naturelle. Comme on peut le voir, l'interface haptique reproduit physiquement le mode opératoire. Elle est composée de deux instruments et d'un endoscope placés à travers des trocards fictifs. Le mouvement des instruments dans l'abdomen est reproduit dans un environnement virtuel lui même restitué sur un écran par endoscopie virtuelle (vision de l'endoscope). Un dispositif à retour d'effort sur les instruments permet de donner l'illusion d'un contact physique avec l'organe traité.



FIG. 1.2 – La chirurgie abdominale en laparoscopie (KISMET Medical.)

Outre son intérêt éthique, un tel simulateur a l'avantage de pouvoir être enrichi de fonctionnalités pédagogiques en proposant par exemple une série d'exercices de difficulté croissante, voir la simulation de situation d'urgence afin de mesurer l'efficacité du chirurgien à les gérer. Il est également avantageux sur le plan économique car utilisable en permanence et ne nécessitant pas d'équipements particuliers. Enfin, utilisé avec des données provenant d'un patient réel, il permettrait un entraînement pour une opération difficile et spécifique. Toutefois, nous nous plaçons ici dans un contexte d'applications temps-réel, ce qui exige des concessions sur la qualité de la simulation produite : [SHT⁺98], [SBB⁺00], [SBH⁺98], [SBH⁺00], [Pic01].

En effet, on constate que les simulateurs actuels se limitent à des opérations simples comme l'ablation de la vésicule biliaire dans lesquelles les zones de contact ont lieu uniquement entre l'extrémité d'un objet rigide (l'instrument) et d'un objet déformable (l'organe). Nous sommes bien loin du réalisme attendu par un chirurgien qui voudrait que les organes interagissent, nécessitant de gérer des zones de contact entre objets déformables.

L'industrie: l'aérospatiale, le naval, l'automobile et le bâtiment s'intéressent également

aux simulations physiques interactives. En effet, le simulateur de vol ou le simulateur de conduite auto/moto représentent des outils importants d'apprentissage pour le pilotage. Dans ce cas il n'y a pas d'interface haptique mais un rendu d'effets inertiels induits par le mouvement de la cabine de pilotage montée sur une architecture robotisée. L'attention pour restituer ce rendu est portée sur la conception mécanique et sur le contrôle dynamique basé sur des modèles physiques. Il est possible d'élaborer des lois de contrôle fondées sur des concepts psycho-physiques comme l'interaction visio-vestibulaire pour donner l'illusion de grands déplacements tout en disposant d'actionneurs à faible course [Moh05], [NMA⁺08]. Il n'y a pas dans ce type d'application d'interaction entre objets déformables.

Le développement des interfaces haptiques a permis l'intégration du retour haptique dans certaines simulations interactives de prototypage virtuel (figure 1.4). Ainsi l'utilisateur peut manipuler des objets conçus sur l'écran via une interface haptique. Il peut aussi reproduire les différents gestes du montage via une interface de capture de mouvement et vérifier que toutes les opérations de montage de la pièce manipulée seront possibles. La prise en compte des frottements et des déformations de certaines pièces (les câbles, les pièces enclipsées et les pièces en mousse) dans les applications du prototypage virtuel est indispensable pour le réalisme et la précision de ces applications [Dur04]. Le prototypage avec retour haptique permettrait de réduire le temps de développement, de diminuer le nombre d'essais réels, les coûts, d'accéder à des grandeurs non mesurables et de diminuer les risques d'erreurs de conception [Dur04], [Wan02], [dSZ99].

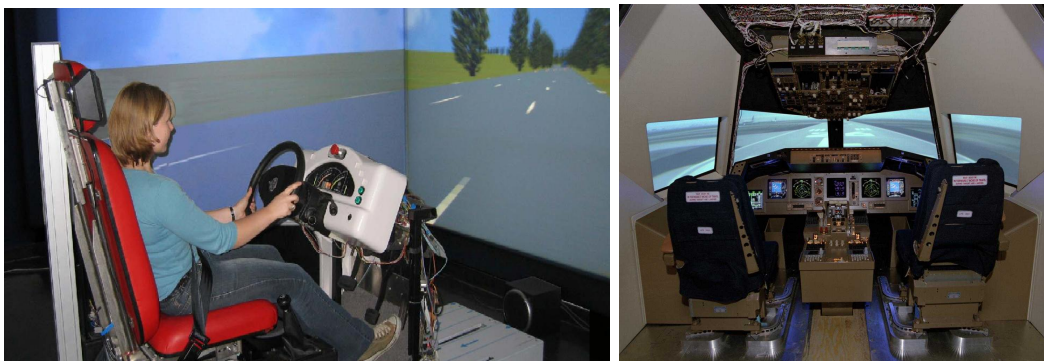


FIG. 1.3 – A droite: Simulateur de conduite développé par INRETS et IBISC. A gauche: Simulateur de vol développé par la NASA.



FIG. 1.4 – Plateforme Perf-RV (INRIA Rennes/CEA List): Test de montage d'un moteur lève vitre à l'intérieur d'une portière. Test d'ouverture des portières (voiture Renault).

1.3 Schéma général de fonctionnement d'un simulateur interactif avec retour haptique

La figure 1.5 décrit le schéma fonctionnel d'une simulation interactive avec retour haptique. Le processus de simulation numérique d'un système physique est un processus incrémental c'est à dire que les inconnues du problème sont calculées à partir de données imposées de manière échantillonnée. Ici, la donnée imposée est un incrément en déplacement d'un objet virtuel pouvant entraîner une collision avec un autre objet virtuel, l'inconnue du problème est alors la force de retour haptique directement liée au calcul des forces de contact appliquées sur l'objet virtuel déplacé. La durée réelle de l'incrément correspond à la durée de la simulation pour le calcul de la force haptique. Pour que la perception haptique soit de qualité, il faut que cette durée soit faible sinon l'opérateur sera à même de percevoir cette discontinuité incrémentale sous la forme d'une vibration. Cette vibration non physique dégrade la qualité du rendu haptique et peut entraîner une instabilité du bras haptique qui ne peut plus être contrôlé par l'opérateur. De plus il faut s'assurer une perception réaliste du phénomène simulé ce qui veut dire que le temps de calcul d'un incrément doit être inférieur ou égal au pas de temps nécessaire pour simuler le phénomène désiré. Dans le temps de calcul d'un incrément du processus de simulation, il faut considérer :

- le calcul des déformations et des déplacements induits,
- le calcul de détection de collision,
- le calcul des forces de contact.

Le challenge est considérable car il est admis qu'il faut une fréquence de calcul supérieure à 300 Hz pour restituer un bon « toucher » (rendu kinesthésique) avec des objets

mous (1kHz avec des objets durs). Il est donc nécessaire d'optimiser ces trois phases de calcul au niveau algorithmique car il a été démontré que l'immersion de l'opérateur et donc sa capacité d'apprentissage sont directement liées à la fréquence de mise à jour du simulateur [DA03], [Dur04]. Pour un bon rendu visio-haptique, il est nécessaire de prendre en compte la déformation des corps en contact de manière réaliste (Modèles physiques d'objets déformables) afin de restituer un rendu haptique cohérent avec la perception visuelle que l'on a en général de la déformation des objets mous. Le rendu visuel est moins contraignant sur le temps d'exécution de la simulation que le rendu haptique, les fréquences acceptables d'échantillonnage sont de l'ordre de 30 Hz.

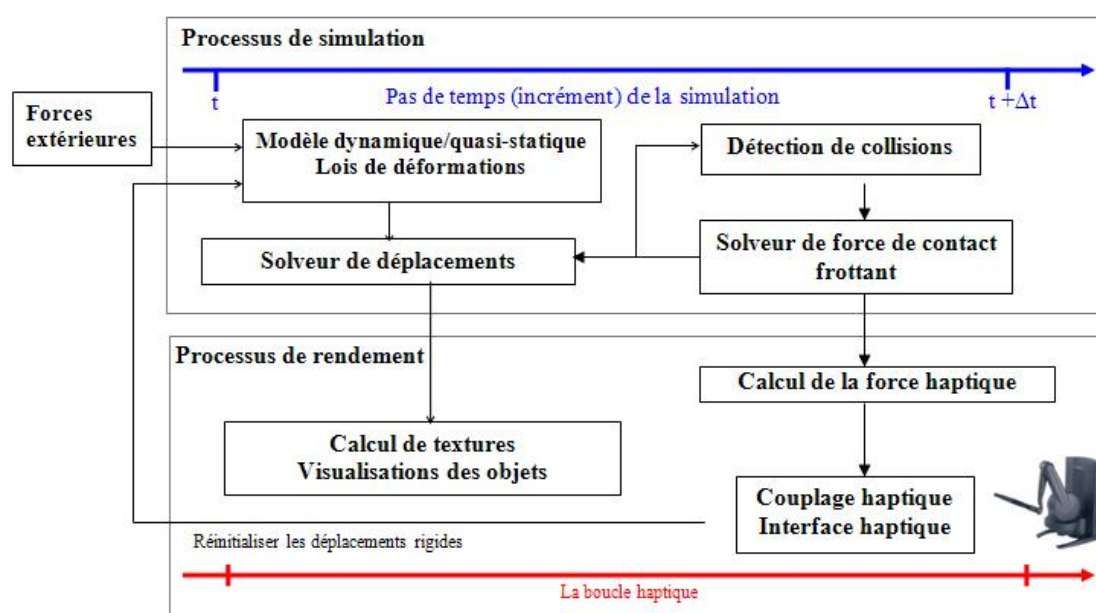


FIG. 1.5 – Schéma fonctionnel d'une simulation interactive avec retour haptique.

1.4 Modélisation d'objets déformables

Les objets déformables regroupent une large gamme de solides non rigides avec différentes propriétés physiques. Citons comme exemples les muscles, les tissus biologiques, les vêtements, la gelée et l'eau, etc. Leur modélisation est plus complexe que celle des objets rigides pour de multiples raisons :

- Premièrement par la complexité des variables décrivant le mouvement de l'objet déformable qui est supérieure à celle du mouvement d'objet rigide. En effet, la position du centre de gravité et l'orientation d'un repère attaché à ce point suffisent à décrire le mouvement d'un objet rigide alors que pour un objet déformable, le mouvement de toutes les particules qui le composent doit être décrit.
- Deuxièmement parce que les objets déformables peuvent entrer en auto-collision. Ainsi la résolution de la détection de collision devient plus complexe.

- Enfin par la prise en compte de la physique des objets déformables décrite par une loi physique constitutive mettant en relation les efforts et les contraintes internes.

Les premières animations cherchant à reproduire des déformations mécaniques ont été basées sur des modèles non-physiques. Dans cette approche, les objets sont de nature géométrique, représentés par leur enveloppe. Celle-ci est constituée d'une densité surfacique de points reliés entre eux par des fonctions mathématiques explicites (fonctions d'interpolation, courbes d'interpolation, B-splines ...) ou par des équations scalaires (surfaces implicites). Ces modes de représentation donnent accès à des points de contrôle permettant à l'opérateur de créer tout type de mouvement ou de déformation modifiant ainsi position, orientation et forme de l'objet initial. On peut également modifier la forme de l'objet en déformant l'espace qui le contient c'est le principe même de la méthode FFD (Free Form Deformation). Nous renvoyons le lecteur à la thèse de Debonne [Deb00] pour plus de détails sur les différentes techniques de modélisation géométrique des déformations.

Jusqu'à la fin des années 1980, les modèles non-physiques étaient les seuls à être utilisés en animation graphique et en simulation interactive et ceci pour différentes raisons : les possibilités limitées en puissance de calcul rendaient les autres solutions inaccessibles. De plus la flexibilité de ces modèles non-physiques permettait de reproduire de nombreuses situations du monde réel sous une apparence suffisamment réaliste. Les procédés d'animation utilisaient des modèles cinématiques, construits à partir de primitives géométriques ou algébriques se déplaçant le long de trajectoires prescrites et décrites par des fonctions mathématiques [Par82]. Cela rendait longue et ardue, la création d'animations d'apparence naturelle.

Terzopoulos a rompu avec cette habitude en créant des modèles actifs, qui répondent à l'action de forces, de contraintes ou de contacts, comme les objets du monde réel (figure 1.6). Les modèles de Terzopoulos ne visaient pas à réaliser des applications en temps réel. Dans ce nouveau formalisme, une courbe, une surface ou un objet 3D se déforme afin de minimiser son énergie de déformation. Celle-ci est mesurée à partir de la norme d'une matrice appelée tenseur métrique et qui n'est rien d'autre que le tenseur des déformations de Green-Lagrange défini en Mécanique des Milieux Continus (MMC) en élasticité non linéaire [TPBF87], [TF88]. Terzopoulos a été ainsi le premier à concevoir des modèles de déformations basés sur des lois de la physique. Il a réussi à établir le lien entre le monde de la mécanique et celui de l'animation graphique en empruntant les théories et les méthodes de l'ingénierie mécanique pour les appliquer à la simulation d'objets déformables. Avec l'émergence des besoins de simulation dans le domaine biomédical, ces travaux ont ouvert la voie à toute une série de recherches visant à trouver des algorithmes et des techniques de calcul appropriés à la simulation physique de déformations en temps réel.

On peut distinguer deux grandes classes de modèles physiques basés sur une description Lagrangienne du mouvement. On rappelle qu'une description Lagrangienne consiste à décrire le mouvement d'un ensemble de particules matérielles au cours du temps par rapport

à leur configuration initiale :

- Les modèles particuliers dans lesquels le solide est représenté par un ensemble de particules auxquelles on associe un voisinage d'interaction. Citons le modèle masse-ressorts, le modèle dynamique moléculaire, le modèle SPH (Smoothed particule Hydrodynamics) et toutes les variétés des modèles issus des méthodes sans maillages (Meshless, MeshFree methods).
- Les modèles continus dans lesquels le solide est décomposé en petits sous domaines appelés éléments ou cellules et connectés les uns aux autres. Citons les modèles MMC : la méthode des éléments finis, la méthode des volumes finis et la méthode des différences finies.

Ainsi dans l'esprit initié par Terzopoulos, un grand nombre de méthodes de simulation interactive basées sur des modèles physiques ont été développées. Bien que la plupart de ces modèles soient utilisés depuis longtemps en calcul des structures et en interaction fluide/structure, le challenge vient du contexte de la simulation interactive qui nécessite des compromis entre précision et rapidité des calculs tout en conservant la stabilité numérique. Une description de ces modèles peut être trouvée dans l'état de l'art réalisé en 1997 par Sarah Gibson et Brian Mirtich [GM97] ainsi que dans les thèses de Mathieu Desbrun [Des97] et Stéphane Cotin [Cot97].

Nous développons respectivement dans les 2 paragraphes suivants, le modèle masses-ressorts et le modèle MMC par la méthode des éléments finis qui sont les deux modèles les plus couramment rencontrés dans les simulateurs chirurgicaux. Ils représentent chacun, une des deux classes des modèles physiques citées ci-dessus. Le modèle MMC par la méthode des éléments finis sera, par la suite, appelé modèle éléments finis par abus de langage.

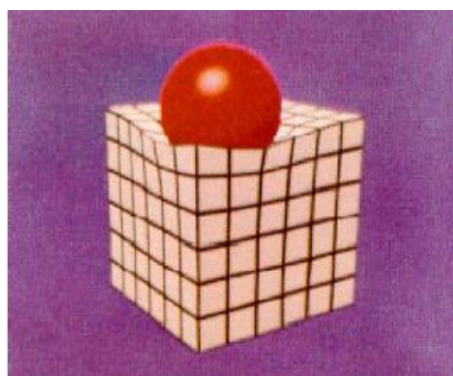


FIG. 1.6 – Balle sur un solide déformable d'après le modèle de Terzopoulos.

1.4.1 Modèle masses-ressorts

De part leur simplicité et leur aspect intuitif, les modèles masses-ressorts ont eu un grand succès dans le domaine de l'image de synthèse. Une vision continue de l'espace

étant difficilement compatible avec la modélisation informatique, on va le discrétiser pour ne plus le considérer que comme un ensemble de valeurs ponctuelles. En effet, l'objet va être représenté par un nombre fini de points, si possible équi-répartis pour une meilleure représentation. Leur comportement les uns vis-à-vis des autres est intuitivement celui de petites masses ponctuelles reliées à leurs voisines par des relations de type ressorts, figure 1.7, [Pro95], [LPC95], [GW97], [NT98], [TGG00], [BC00].

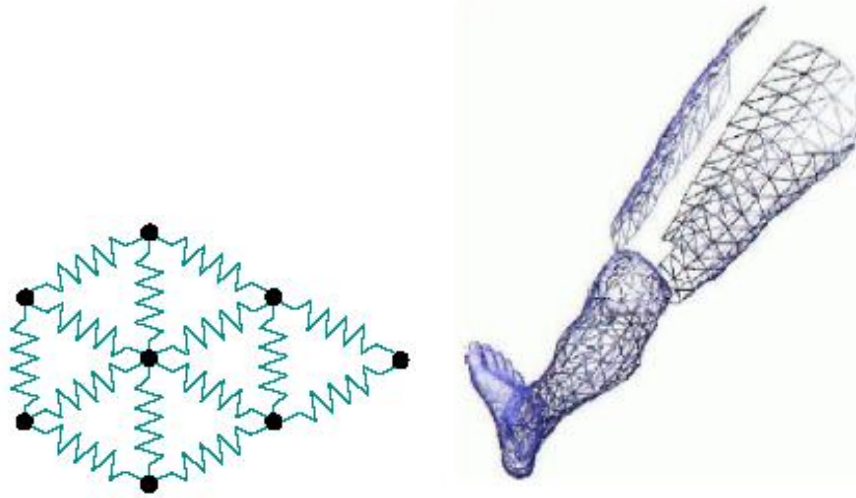


FIG. 1.7 – A droite : Discretisation d'un objet 2D à l'aide de masses-ressorts. A gauche : un maillage en 3D modèle masse ressort, tiré de [Aul01].

Ce réseau Masse-ressort sera utilisé pour discrétiser les équations du mouvement. Le ressort qui relie une paire de noeuds définit une élasticité locale, il sera ainsi facile de calculer la force interne subie par chaque masse à chaque instant en fonction des distances à ses voisines : $\sum_{j \in V(i)} k^{ij} (\|\vec{p_i p_j}\| - l_0) \frac{\vec{p_i p_j}}{\|\vec{p_i p_j}\|}$. $V(i)$ représente les points voisins du point i et k^{ij} la raideur entre les points p_i et p_j . Ces forces internes sont souvent linéaires; cependant, l'utilisation de ressorts non linéaires permet d'obtenir des comportements élastiques non linéaires (grandes déformations). Nedel et Thalmann ont utilisé cette technique pour modéliser des tissus mous à comportement non élastique [NT98].

Dans le système dynamique, la seconde loi de Newton gouverne le mouvement de chaque point-masse p_i de l'objet modélisé par un réseau masse-ressort :

$$m_i \ddot{p}_i + d_i \dot{p}_i + k_i p_i + \sum_{j \in V(i)} k^{ij} (\|\vec{p_i p_j}\| - l_0) \frac{\vec{p_i p_j}}{\|\vec{p_i p_j}\|} = f_{ext}$$

m_i , d_i et k_i sont trois scalaires désignant respectivement la masse, le coefficient d'amortissement et la raideur du point i . f_{ext} représente les forces externes appliquées à l'objet comme la force de gravité, la force appliquée par un utilisateur ou les frottements visqueux

etc. L'algorithme consistera donc à intégrer ces équations au cours du temps, et ce pour chaque point séparément.

Les ressorts permettent d'insuffler au modèle des énergies de déformation qui ont pour tâche de définir le comportement de l'objet. Ainsi, le choix du type de ressort est des plus importants dans l'élaboration d'un modèle masses-ressorts. Pour cela, voici une liste non exhaustive de quelques types de ressorts utiles [Len04] : ressort d'élongation, ressort d'élongation amorti et ressorts angulaires.

Le comportement d'un modèle masses-ressorts est directement lié au type de ressort employé. Ainsi, puisque les ressorts décrits ci-dessus ne permettent pas de décrire certaines déformations, de nombreux travaux ont été menés à ce sujet. Provot [Pro95] propose de définir une contrainte sur l'élongation des ressorts. En effet, les ressorts sont munis d'une élongation maximale qui permet de limiter le comportement trop élastique des simulations masses-ressorts sans pour autant augmenter de manière démesurée le coefficient de raideur des ressorts. Ainsi le modèle proposé permet de simuler des tissus très peu élastiques sans soulever de problèmes d'intégration numérique (problème raide). Bourguignon et Cani [BC00] proposent de contrôler l'anisotropie des déformations en exprimant des propriétés élastiques du matériau en tout point de l'objet sur différents axes. Boux de Casson et al [dCL99] ont mis au point une technique qui permet d'augmenter le réalisme de la simulation de déformation du rein, en modélisant le caractère hétérogène de ce matériau, ce dernier est recouvert d'une fine peau qui sert de protection (capsule de Glisson dans le cas du foie). Le comportement du matériau global est obtenu par la combinaison du comportement de ces deux modèles : l'élasticité de la peau et la viscosité de l'intérieur.

L'un des premiers modèles temps réel fut développé par [RL96]; Il s'agit d'un maillage 2D déformable entourant un corps rigide et les résultats obtenus sont qualitativement satisfaisants. Cependant, le maillage 3D est essentiel pour se rapprocher de la réalité et avoir des résultats exploitables.

Ce modèle masses-ressorts est largement utilisé : nous citons [BHG92], [BHW94], [BHW94], [Pro95], [VCMT95], [LPC95], [EWW96], [BW98], [HB00], [VMT00] et [BWK03] pour la simulation de vêtements ou tissus textiles. Zara et al [ZfV02] simulent un modèle masses-ressorts de vêtements sur une architecture distribuée. [TGG99] pour la simulation chirurgicale, Chadwick [CHP89] pour animer la chair de personnages (sans coefficient d'amortissement) [PB81], [PNT98] et [ZPS01] pour l'animation faciale. Nous pouvons aussi noter les travaux de Montgomery et al [MHBW01] qui ont mis au point un simulateur d'hystérocopie, ou ceux de Dower et al [DCG⁺98] dont le simulateur est dédié à l'entraînement d'opérations en laparoscopie. Une autre application particulièrement avancée utilisant des modèles de masses-ressorts est le «Karlsruhe Endoscopic Surgery Trainer», un simulateur de chirurgie laparoscopique (endoscopie à travers l'abdomen) adapté notamment à l'enlèvement de la vésicule biliaire et à des opérations gynécologiques [KKH⁺97], [KaM00], [CK00]. Ce simulateur (figure 1.8) est sans doute le plus avancé dans ce domaine.

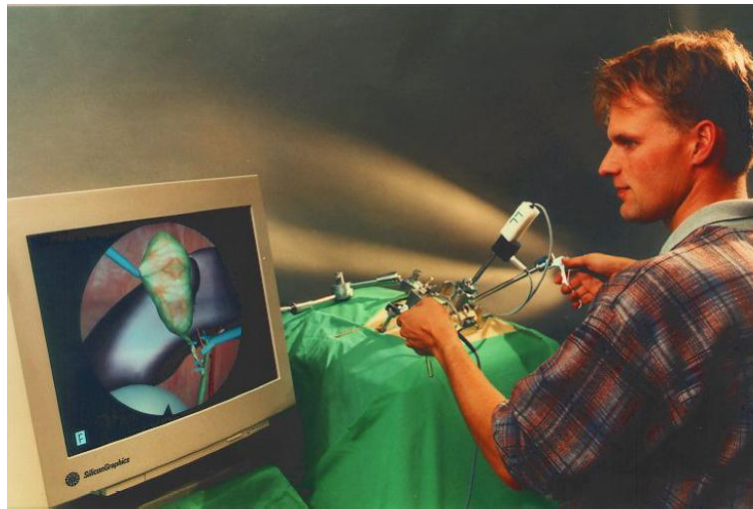


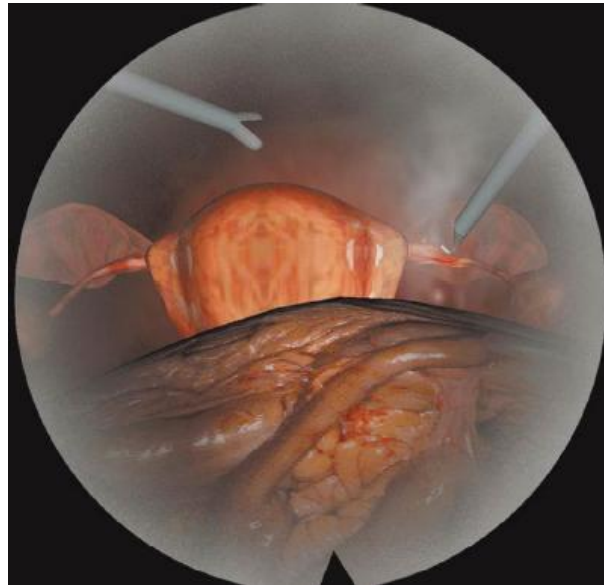
FIG. 1.8 – *Simulation de laparoscopie.*

En effet, la simulation produit une vue virtuelle telle que la produirait en réalité la caméra endoscopique et offre au chirurgien un ensemble d'outils identiques à ceux dont il dispose au cours d'une opération réelle, lui permettant de saisir, couper ou coudre des organes. Les organes ont été modélisés par des maillages surfaciques ou volumiques de masses-ressorts et se comportent comme des matériaux élastiques linéaires. La simulation se fait avec retour de force et permet :

- La manipulation de la caméra et le changement de vues selon la position de l'endoscope.
- La manipulation d'instruments variés tels que les pinces, scalpels, etc.
- La simulation du saignement et de la pression artérielle (ressentie via le système haptique).
- La simulation de la fumée dégagée lors de l'opération.
- La suture (pose de clips pour arrêter le saignement).
- L'irrigation pour nettoyer la cavité puis aspiration.

Ce simulateur permet ainsi de modéliser plusieurs gestes caractéristiques d'une opération chirurgicale de type laparoscopie. Néanmoins, de part le modèle et des différentes simplifications nécessaires au temps réel, il reste peu réaliste [Pac].

La méthode masses-ressorts a l'avantage d'être facile à implémenter. De plus, elle permet une résolution temps réel, rendant possible le couplage avec un système de retour de force. Par conséquent, elle devient un outil puissant dans l'apprentissage de gestes opératoires (figure 1.9).

FIG. 1.9 – *simulation masse ressort.*

Par contre, les résultats obtenus en terme de comportement mécanique sont peu réalistes et l'identification des paramètres masses, raideurs et coefficients d'amortissement est très délicate. Par ailleurs, le temps réel nécessaire au couplage avec un système haptique limite la taille du maillage. Le temps réel a donc un prix, celui de la précision. Ne se basant pas sur des principes continus, les résultats sont moins fiables que pour les modèles à éléments finis.

1.4.2 Méthode des éléments finis

Les modèles physiques issus de la MMC fournissent des équations qui en général sont sans solution analytique. Cependant, l'analyse numérique apporte des solutions numériques (qui convergent vers la solution exacte) obtenues par l'intermédiaire de différentes méthodes parmi lesquelles : la méthode des différences finies, la méthode des volumes finis et la méthode des éléments finis qui est la plus utilisée dans le cas de calcul de structure et à laquelle nous allons nous intéresser. La méthode des éléments finis s'applique aussi pour l'analyse des déformations et des contraintes dans des structures mécaniques comme les ponts, les immeubles et les avions. Elle est aussi utilisée dans des problèmes de conduction thermique, d'écoulement de fluides et de flux gazeux ou magnétiques, etc. Plus généralement, cette technique permet de résoudre des équations différentielles en temps et aux dérivées partielles en espace avec conditions aux limites [Nes04].

Pour une étude plus approfondie de la méthode des éléments finis, nous recommandons les cours de J. Garrigues (Garrigues, 1999)¹ et de P.J. Hunter [HP98]², ainsi que le chapitre 2

1. La méthode des éléments finis : <http://esm2.imt-mrs.fr/gar/ef.html>

2. <http://www.esc.auckland.ac.nz/Academic/Texts/FEM-BEM-notes.html>

de la thèse de S. Cotin [Cot97] ou bien encore le chapitre 3 du livre de W. Maurel et al [MWTT98] et les ouvrages de références de O. Zienkiewicz [Zie77], de K-L. Bathe [Bat82] et de T. Hughes [Hug87].

Son Principe

La méthode des éléments finis est un outil de discrétisation. Elle permet de diviser un milieu continu complexe (un objet) en un certain nombre fini d'éléments géométriques (un maillage) relativement simples appelés éléments finis, comme des tétraèdres ou des hexaèdres. Les champs physiques sont interpolés sur chaque élément en fonction de leur valeur en certains points donnés nommés noeuds ou points de contrôle. Dans les cas les plus simples ces noeuds sont situés aux sommets de l'élément. A chaque élément est défini un système d'équations obtenu à partir d'une formulation intégrale variationnelle des équations de la MMC (formulation faible). Puis un processus d'assemblage prenant en compte la connectivité entre tous les éléments mène à la construction d'un ensemble d'équations à résoudre. La résolution de ce système d'équations, en tenant compte des conditions imposées aux limites du système, permet d'obtenir une approximation continue des propriétés physiques [CB97].

Le Maillage

Les éléments finis peuvent être de différentes formes : en dimension 2 les formes les plus couramment utilisées sont les triangles ou les quadrilatères. En dimension 3 ce sont les tétraèdres ou les hexaèdres. Il est possible d'utiliser différents types d'éléments dans un même maillage mais cela complique significativement l'écriture du système. Dans la majorité des cas, un seul type d'éléments est utilisé pour un maillage donné, (figure 1.10).

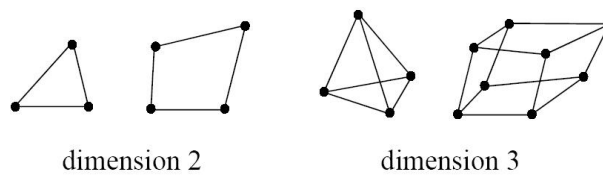


FIG. 1.10 – Exemples d'éléments finis surfaciques et volumiques.

Plus le maillage est fin, plus le calcul est précis, mais coûteux. Cependant, un compromis s'impose entre la finesse de la discrétisation des objets et le coût du calcul pour les simulations interactives. Nous pouvons aussi réaliser un maillage grossier dans certaines régions et plus fin dans d'autres, en fonction de l'importance des déformations et afin d'augmenter la précision là où le calcul s'en fait ressentir. Ainsi nous pouvons adapter la densité d'éléments selon les nécessités de la simulation.

Un avantage supplémentaire est apporté par cette représentation locale des propriétés mécaniques. Comme les paramètres mécaniques sont associés à chaque élément d'un

maillage, il est possible de construire des modèles hétérogènes composés de différentes structures imbriquées et de propriétés mécaniques différentes.

À chaque élément fini est associé un certain nombre de noeuds. Les noeuds forment l'ensemble discret des points du système en lesquels les propriétés physiques sont calculées. Comme vu plus haut, l'ensemble des noeuds ne se limite pas forcément aux sommets des éléments finis. Il est possible de définir des noeuds additionnels, par exemple un quadrilatère à 8 noeuds ou un tétraèdre à 10 noeuds (figure 1.11) afin d'assurer une meilleure continuité à la frontière entre les sous domaines et d'approcher au mieux la solution cherchée.

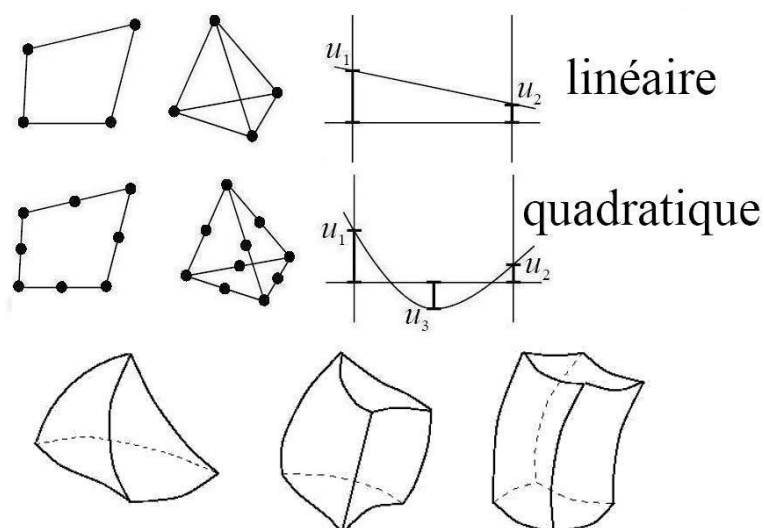


FIG. 1.11 – *En haut : Interpolation linéaire ou quadratique sur un élément fini, En bas : plus le degré d'interpolation est élevé, plus l'élément peut se déformer.*

Nous avons adopté dans notre mise en oeuvre des éléments finis tétraédriques et hexaédriques. Le tétraèdre est la figure tridimensionnelle la plus simple et permet de mailler des formes quelconques sans introduire d'anisotropie topologique, contrairement à ce qui se produit avec un maillage de parallélépipèdes.

Les Fonctions d'interpolation

Elles sont aussi appelées les fonctions de forme. Le choix du nombre de noeuds de chaque élément est lié au choix d'un schéma d'interpolation. Le principe de la méthode des éléments finis étant d'interpoler les champs physiques en tout point du continuum en fonction de leur valeur aux noeuds, diverses fonctions d'interpolation peuvent être choisies. La plus simple est une fonction d'interpolation linéaire. Dans ce cas les seuls noeuds utilisés

sont les sommets des éléments finis. En choisissant une fonction d'interpolation quadratique, un noeud additionnel est nécessaire sur chacune des arêtes. Ce cas correspond au cas du quadrilatère à 8 noeuds ou au tétraèdre à 10 noeuds (figure 1.11). La précision du résultat est améliorée lorsqu'une fonction d'interpolation de degré plus élevé est utilisée. Il est possible d'augmenter librement le degré de la fonction d'interpolation mais en pratique des fonctions de degré supérieur à 3 sont rarement utilisées, étant donné la lourdeur des calculs que cela implique. Nous adoptons alors une interpolation linéaire sur les éléments finis car notre objectif est de limiter au maximum le temps de calcul et la recherche d'une précision inférieure à la taille d'un élément fini n'est pas une priorité.

Il est aussi possible de construire un espace de fonctions d'interpolation directement sur chaque maille réelle, mais généralement, nous procédons en deux temps : nous construisons d'abord un espace de fonctions d'interpolation N_i sur une maille de référence standard topologiquement équivalente à la maille réelle, puis nous le transformons pour qu'il devienne un espace de fonctions d'interpolation sur les mailles réelles. $J(3 \times 3)$ est la matrice jacobienne qui correspond à cette transformation. Ce procédé à l'avantage de faire gagner du temps de calcul (figure 1.12).

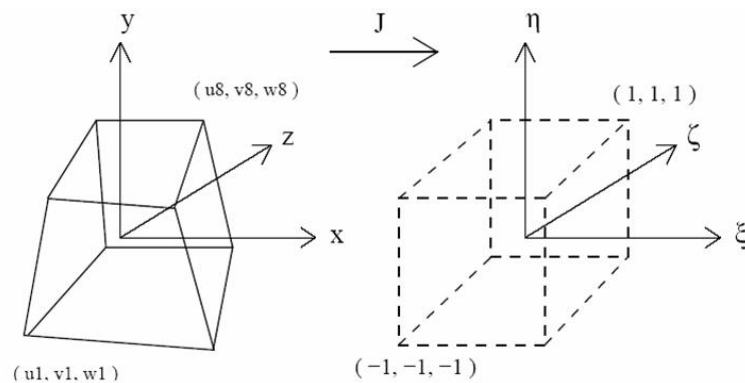


FIG. 1.12 – La transformation iso-paramétrique d'un élément cubique.

La formule suivante représente un exemple de fonctions d'interpolations utilisées dans le cas d'un élément cubique de 8 sommets défini dans un espace local (ξ, η, ζ) :

$$\left\{ \begin{array}{l} N_1 = 0.125 (1 - \xi)(1 - \eta)(1 - \zeta) \\ N_2 = 0.125 (1 + \xi)(1 - \eta)(1 - \zeta) \\ N_3 = 0.125 (1 + \xi)(1 + \eta)(1 - \zeta) \\ N_4 = 0.125 (1 - \xi)(1 + \eta)(1 - \zeta) \\ N_5 = 0.125 (1 - \xi)(1 - \eta)(1 + \zeta) \\ N_6 = 0.125 (1 + \xi)(1 - \eta)(1 + \zeta) \\ N_7 = 0.125 (1 + \xi)(1 + \eta)(1 + \zeta) \\ N_8 = 0.125 (1 - \xi)(1 + \eta)(1 + \zeta) \end{array} \right.$$

Pour un élément fini e composé de s noeuds, on en déduit la relation :

$$\mathbf{u} = \mathbf{N}_e \mathbf{u}^{(e)} \quad (1.1)$$

$$\text{avec: } \mathbf{N}_e = \left[\begin{array}{ccc|ccc|c|ccc} N_1 & 0 & 0 & N_2 & 0 & 0 & \dots & N_s & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \dots & 0 & N_s & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \dots & 0 & 0 & N_s \end{array} \right]$$

$$\mathbf{u}^{(e)} = \left\{ \mathbf{u}_1^{(e)} \ \mathbf{u}_2^{(e)} \ \dots \ \mathbf{u}_s^{(e)} \right\}^T, \quad \mathbf{u}_i^{(e)} = \{u_i \ v_i \ w_i\}^T$$

$\mathbf{u}_i^{(e)}$ représente le vecteur déplacement du noeud i appartenant à l'élément e , de même on peut lui associer son vecteur position $\mathbf{x}_i^{(e)} = \{x_i, y_i, z_i\}^T$.

L'Assemblage

Dans un maillage d'éléments finis, un noeud donné appartient en général à plusieurs éléments et la déformation de chacun d'entre eux induit une force en ce noeud. Le calcul des forces se fait ainsi élément par élément avec des systèmes locaux. Les sommets communs sont des noeuds de connexion qui permettent l'assemblage de tous les systèmes locaux en un grand système global contenant les équations de tous les noeuds. Cela nécessite une renumérotation des noeuds à travers une table de connectivité. Pour l'ensemble des n noeuds qui discrétisent l'objet déformable S , on définit le vecteur de position global $\mathbf{X} = \{x_1, x_2, x_3, \dots, x_n\}^T$ et le vecteur de déplacement globale $\mathbf{U} = [u_1, u_2, u_3, \dots, u_n]^T$. Le processus d'assemblage est relativement coûteux numériquement. Dans ce qui suit nous aborderons uniquement la formulation de modèles physiques simples qui peuvent être résolus de manière rapide par la méthode des éléments finis.

1.4.2.1 Élasticité linéaire

Un matériau élastique se déforme sous l'action de certaines forces et regagne son état de repos, en passant par le même chemin, une fois que ces forces disparaissent. Il restitue ainsi toute l'énergie qu'il avait absorbée. L'élasticité est de plus linéaire lorsqu'il y a proportionnalité entre déformations et efforts imposés. Ce cas particulier n'est constaté qu'en petites déformations (voir figure 1.13). En général, on considère qu'un matériau élastique n'est plus linéaire lorsque les déplacements élastiques dépassent 10% de la taille de l'objet considéré. Au delà, il faut utiliser la théorie de la MMC en grandes déformations [PDA03].

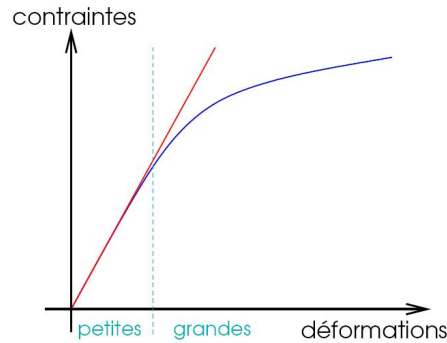


FIG. 1.13 – Approximation linéaire en rouge d'une loi de comportement (ici élasto-plastique).

Nous allons aborder dans la suite quelques éléments de la théorie de la MMC en élasticité linéaire. Les petites déformations peuvent être écrites d'une manière condensée comme suit (écriture tensorielle) :

$$\varepsilon = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{xy} \\ 2\varepsilon_{xz} \\ 2\varepsilon_{yz} \end{Bmatrix} \quad \text{tel que} \quad \left\{ \begin{array}{l} \varepsilon_{xx} = \frac{\partial u}{\partial x} \quad , \quad \varepsilon_{xy} = \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ \varepsilon_{yy} = \frac{\partial v}{\partial y} \quad , \quad \varepsilon_{xz} = \frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\ \varepsilon_{zz} = \frac{\partial w}{\partial z} \quad , \quad \varepsilon_{yz} = \frac{1}{2} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \end{array} \right\}$$

On définit ainsi un opérateur de dérivation \mathbf{S} entre les petites déformations ε et les déplacements $\mathbf{u} = \{u, v, w\}^T$ tel que :

$$\varepsilon = \mathbf{S}\mathbf{u} \quad (1.2)$$

$$\text{avec } \mathbf{S} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix}$$

Loi de Hooke : La loi de Hooke est une loi de comportement du matériau qui lie les petites déformations ε aux contraintes σ et qui s'applique aux matériaux isotropes (propriétés du matériau identiques dans toutes les directions), homogènes. Cette loi est linéaire

et peut se mettre sous la forme suivante :

$$\sigma = \mathbf{D}\varepsilon \quad (1.3)$$

avec

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

\mathbf{D} est le tenseur d'élasticité avec μ le coefficient de résistance au cisaillement et λ le coefficient de compression, appelés également coefficients de Lamé et qui sont définis par :

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

Où E et ν sont deux paramètres caractéristiques du matériau appelé respectivement module de Young (exprimé en MPa) et coefficient de Poisson (sans unité). E caractérise la raideur du matériau, il peut varier de 0,02 Mpa pour des tissus mous à plus de 200 000 Mpa pour des aciers. ν caractérise la compressibilité du matériau, sa valeur est comprise entre 0 et 0,5. $\nu=0,5$ pour un matériau dit «incompressible».

Pour plus de détails sur les concepts abordés dans ce paragraphe, nous conseillons les ouvrages de D.François [DPZ90], de Y.C.Fung [Fun65] et de A.J.M.Spencer [Spe84]; ainsi que les cours de J.Garrigues³ et de T.G.Shawki [Sha96]⁴ sur la mécanique des milieux continus. En ce qui concerne la théorie d'élasticité, le cours très pédagogique de J.Garrigues⁵ pourra être approfondi par l'étude des ouvrages de P.G.Ciarlet [Cia88], [Cia90], [Cia97], de J.J.Oden [Ode67] et de R.w.Ogden [Ogd84].

Dans le cas d'un modèle statique ou quasi-statique, on ne prend pas en compte les forces inertielles liées aux accélérations. La méthode des éléments finis basée sur la théorie de la MMC en élasticité linéaire permet de construire une matrice de rigidité constante,

3. mécanique des milieux continus 1999 : <http://esm2.imt-mrs.fr/gar/mmc.html>

4. <http://www.tam.uiuc.edu/Faculty/Shawki.html>

5. Elasticité classique 1999 : <http://esm2.imt-mrs.fr/gar/elas.html>

qu'il suffit d'inverser une seule fois, hors ligne, de manière à optimiser les calculs.

Les équations du modèle statique sont obtenues par application du principe des travaux virtuels appliqué à chaque élément finis :

$$\delta W_{int}^{(e)} + \delta W_{ext}^{(e)} = 0 \quad (1.4)$$

avec

$$\delta W_{int}^{(e)} = - \int_{\Omega^e} \delta \varepsilon^T \sigma \, dv$$

$$\delta W_{ext}^{(e)} = \delta \mathbf{u}^{(e)T} \mathbf{f}_{ext}^{(e)}$$

En utilisant les relations 1.1, 1.2, 1.3, on obtient :

$$\delta W_{int}^{(e)} = -\delta \mathbf{u}^{(e)T} \mathbf{K}^{(e)} \mathbf{u}^{(e)} = \delta \mathbf{u}^{(e)T} \mathbf{f}_{int}^{(e)} \quad (1.5)$$

avec

$$\mathbf{K}^{(e)} = \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} \, dv, \quad \mathbf{B} = \mathbf{S} \mathbf{N}_{(e)}$$

En utilisant l'équation 1.4, la relation 1.5 et par application du principe des travaux virtuels, on obtient les équations d'équilibre :

$$\mathbf{K}^{(e)} \mathbf{u}^{(e)} = \mathbf{f}_{ext}^{(e)} \quad (1.6)$$

Le calcul de la matrice de raideur $\mathbf{K}^{(e)}$ est élémentaire lorsque les fonctions de forme sont linéaires car la matrice \mathbf{B} est constante et indépendante des variables d'intégration. Dans le cas contraire, le calcul de $\mathbf{K}^{(e)}$ est discrétisé en des points particuliers appelés «points de Gauss» [Zie77]. Les matrices $\mathbf{K}^{(e)}$ sont indépendantes de $\mathbf{u}^{(e)}$ et donc des déformations. Elles peuvent être calculées «hors ligne» (avant le début de la simulation).

La table de connectivité associée au maillage, nous permet de construire une matrice de raideur globale $K(3n \times 3n)$ où n représente le nombre de noeud du maillage. Le système

linéaire global s'écrit alors sous la forme matricielle :

$$\mathbf{K}\mathbf{U} = \mathbf{F}_{ext} \quad (1.7)$$

Cette matrice \mathbf{K} née du regroupement des $\mathbf{K}^{(e)}$ est creuse puisque un noeud ne peut appartenir qu'à un nombre très limité d'éléments. Elle est par contre symétrique, définie positive (si elle tient compte des conditions de Dirichlet) et il est possible de la mettre sous la forme d'une matrice bande afin d'appliquer des méthodes optimisées de résolution du système d'équations 1.7.

Bro-Nielsen et Cotin [BNC96] ont utilisé les éléments finis en élasticité linéaire dans des simulations temps-réel. Le calcul de la matrice de rigidité est effectué hors ligne qui est condensée de la manière suivante afin d'optimiser les temps de calcul :

$$\begin{bmatrix} \mathbf{K}_{ss} & \mathbf{K}_{si} \\ \mathbf{K}_{is} & \mathbf{K}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{U}_s \\ \mathbf{U}_i \end{bmatrix} = \begin{bmatrix} \mathbf{F}_s \\ \mathbf{F}_i \end{bmatrix} \Rightarrow [\mathbf{K}_{ss} - \mathbf{K}_{si}\mathbf{K}_{ii}^{-1}\mathbf{K}_{is}] [\mathbf{U}_s] = [\mathbf{F}_s] - [\mathbf{K}_{si}\mathbf{K}_{ii}^{-1}] [\mathbf{F}_i]$$

\mathbf{U}_s , \mathbf{U}_i représentent respectivement les déplacements des noeuds à la surface et à l'intérieur du solide considéré. Les forces qu'ils cherchent à imposer sur leurs modèles s'appliquent uniquement sur les noeuds à la surface des objets, soit $[\mathbf{F}_i] = 0$. On réduit le nombre de ddl mais la matrice à inverser est beaucoup plus dense.

Cotin [Cot97] propose toujours dans le cas de l'élasticité linéaire de stocker des modes de déformations associés à chaque noeud de la surface du solide. Un mode de déformation est obtenu en appliquant en un noeud de la surface une force unité dans une des 3 directions du repère global. En appliquant le principe de superposition valable en élasticité linéaire, il est possible de calculer rapidement la déformation de l'objet pour un ensemble de forces imposées. Dans le cas de déplacements imposés, les modes de déformations permettent de construire une matrice d'influence mutuelle des noeuds déplacés que nous appellerons plus tard matrice de flexibilité ou matrice de Delassus. Afin d'optimiser la place mémoire, les modes de déformation sont filtrés c'est à dire qu'au delà d'un certain seuil (ou au delà d'une zone d'influence) on considère les déformations comme étant nulles.

Bro-Nielsen [BN98] et Cotin [CDA99] ont réussi à faire évoluer des maillages de l'ordre de 250 éléments avec une résolution dynamique et 1500 éléments et une résolution quasi-statique (entre chaque pas de temps est appliqué un déplacement résolu par une méthode statique). Cette méthode a permis des animations temps-réel malgré l'importance des temps de pré-calculs et de la place mémoire utilisée (figure 1.14).

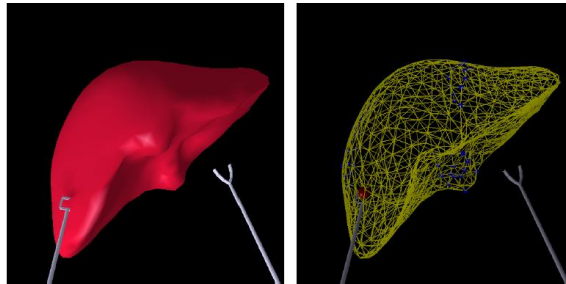


FIG. 1.14 – Déformation d'un modèle de foie composé de 1500 noeuds avec une fréquence visuelle de 50 Hz et une fréquence de 300 Hz pour le retour de force [CDA99].

L'inconvénient de l'approche quasi statique est qu'elle donne une suite d'états d'équilibre indépendants du temps ce qui implique qu'il n'y a pas d'effet de retard. Si nous relâchons un point que nous avons déplacé, celui-ci revient instantanément à sa position initiale, sans oscillation et quelque soit la distance de déplacement. Ainsi le réalisme visuel est affecté du fait de ces retours brusques.

1.4.2.2 Prise en compte des grandes rotations en élasticité linéaire

En grandes rotations, la linéarisation des déformations entraîne des phénomènes bien connus de distorsion et de non conservation du volume (voir figures 1.16). Le comportement élastique n'est plus réaliste. En effet les déformations calculées par ε appelé tenseur de Cauchy ne prend pas en compte les non linéarités dues aux grandes rotations. Ainsi, une grande rotation peut engendrer des déformations parasites qui s'ajoutent aux déformations élastiques. Pour éviter cela, il faut séparer le mouvement de rotation dit de «corps rigide» du mouvement d'ensemble afin de ne considérer que les déplacements élastiques au niveau de la loi de Hooke. Conformément à la classification introduite dans la thèse de Duriez [Dur04], nous allons considérer deux approches appelées respectivement «corotationnelle globale» et «corotationnelle locale».

Corotationnelle globale

Dans cette approche les déformations sont réactualisées par rapport à un unique repère en mouvement attaché au solide. C'est le cas classique où l'on impose un mouvement de corps rigide à des noeuds bloqués en déformation par des conditions de Dirichlet. Dans ce cas on attache un repère mobile à ces noeuds qui seront donc considérés comme fixes dans ce repère. Le mouvement de ce repère étant connu, on en déduit facilement, à partir des coordonnées locales, le déplacement rigide \mathbf{U}_{rigide} de tous les noeuds. On calcule ensuite les déplacements élastiques des noeuds non bloqués $\tilde{\mathbf{U}}$ dans ce repère local ce qui revient à transformer le système 1.7 de la manière suivante :

$$\mathbf{K}\tilde{\mathbf{U}} = \mathbf{R}^T \mathbf{F}_{ext} \quad (1.8)$$

\mathbf{R} représente la matrice de rotation ou de changement de repère du repère global vers le repère local (sous forme de matrice diagonale par blocs).

$$\mathbf{U}_{total} = \mathbf{U}_{rigide} + \mathbf{R}\tilde{\mathbf{U}}$$

Cette approche utilisée par certains auteurs ([TF88] ou [Fel00]) est celle que nous avons adoptée dans le simulateur HapCo présenté dans le Chapitre 4. Elle n'est justifiée que si les déformations restent petites relativement au repère local.

Corotationnelle locale

Dans un contexte de simulation mécanique classique, Felippa [Fel00] présente une méthode plus générale, nommée corotationnelle locale, qui permet d'extraire les rotations de chaque élément du modèle de façon explicite et de la réintroduire dans le calcul des forces élastiques.

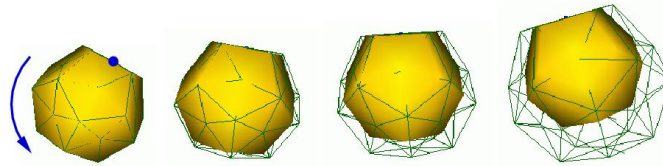


FIG. 1.15 – Dilatation d'un matériau élastique linéaire (fil de fer) sous l'effet d'une rotation globale [Pic01].

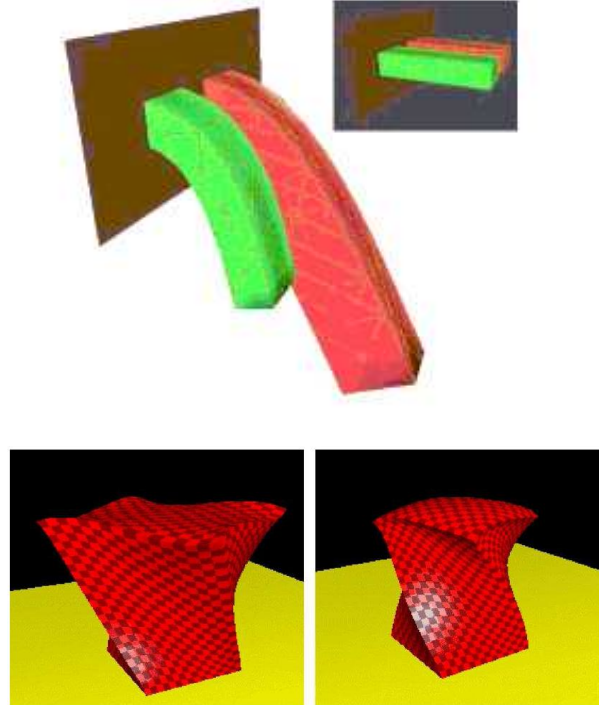


FIG. 1.16 – En haut : Deux poutres en grandes déformations d'après [Nes04] (la poutre rouge est en élasticité linéaire), En bas : le modèle linéaire situé à gauche devient faux d'après [ZC00]

Le cas de la poutre encastrée en grandes déformations présenté par Nesme [Nes04] nécessite de considérer de grandes rotations qui ne sont pas constantes mais qui varient d'un élément fini à l'autre. Il faut donc associer à chaque élément une matrice de rotation $\mathbf{R}^{(e)}$. La méthode la plus élégante pour trouver cette matrice est d'utiliser les coordonnées barycentriques w_i . Prenons l'exemple d'un tétraèdre :

$$\mathbf{x}^{(e)} = \begin{bmatrix} \mathbf{x}_1^{(e)} & \mathbf{x}_2^{(e)} & \mathbf{x}_3^{(e)} & \mathbf{x}_4^{(e)} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{Bmatrix} = \mathbf{X}\mathbf{w} \quad (1.9)$$

avec $\mathbf{x}^{(e)}$ qui représente le vecteur des positions des noeuds du tétraèdre à l'état déformé.

$$\bar{\mathbf{x}}^{(e)} = \begin{bmatrix} \bar{\mathbf{x}}_1^{(e)} & \bar{\mathbf{x}}_2^{(e)} & \bar{\mathbf{x}}_3^{(e)} & \bar{\mathbf{x}}_4^{(e)} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{Bmatrix} = \mathbf{P}\mathbf{w} \quad (1.10)$$

avec $\bar{\mathbf{x}}^{(e)}$ qui représente le vecteur des positions des noeuds du tétraèdre à l'état non déformé. On peut donc d'après ce qui précède établir une relation entre $\mathbf{x}^{(e)}$ et $\bar{\mathbf{x}}^{(e)}$ de la forme :

$$\mathbf{x}^{(e)} = \mathbf{X}\mathbf{P}^{-1}\bar{\mathbf{x}}^{(e)} \quad (1.11)$$

La matrice $\mathbf{X}\mathbf{P}^{-1}$ est une matrice (4×4) qui a la forme d'une matrice homogène :

$$\mathbf{X}\mathbf{P}^{-1} = \begin{bmatrix} \mathbf{G} & \mathbf{a} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (1.12)$$

Le vecteur \mathbf{a} désigne le mouvement de translation et la matrice \mathbf{G} désigne le mouvement de rotation en tenant compte de la déformation. Il faut donc ensuite extraire la matrice $\mathbf{R}^{(e)}$ de la matrice \mathbf{G} par décomposition polaire ou par une orthonormalisation Gram-Schmidt. La construction de la matrice \mathbf{G} présuppose la connaissance de l'état déformé. La résolution ne peut être qu'implicite et est incompatible avec le temps réel. C'est pourquoi cette approche est utilisée uniquement à partir d'une connaissance explicite de cette matrice. L'application de cette approche dans un contexte interactif a été introduite par [MMD⁺02] puis [MG04] et [HS04]. Une méthode similaire, destinée spécialement à des objets de type cylindrique (tubes, câbles, ...), est présentée dans [NNF⁺02]. Citons une autre approche par Capell et al [CGC⁺02] qui proposent d'introduire manuellement un squelette à l'intérieur du maillage et de répartir les éléments autour des segments du squelette. Ainsi, les éléments utilisent les coordonnées locales des segments (qui prennent en compte les rotations) pour calculer les forces linéaires élastiques.

1.4.2.3 Elasticité non linéaire

Lorsque les déplacements élastiques dépassent 10% de la taille de l'objet considéré, on est en élasticité non linéaire. Dans ce cas les déformations ne peuvent plus être représentées par le tenseur de Cauchy ε . Il faut alors considérer le tenseur de Green-Lagrange \mathbf{E} :

$$\mathbf{E} = \varepsilon + \frac{1}{2} (\nabla \mathbf{u}^T \nabla \mathbf{u})$$

∇ représente l'opérateur gradient.

Le tenseur de Green-Lagrange est invariant en rotation, il permet donc de prendre en compte les grandes rotations sans qu'il soit nécessaire de faire appel à des approches corotationnelles. Dans le cadre de la méthode des éléments finis, le tenseur de Green-Lagrange inclut formellement les termes linéaires ($\mathbf{B}_L \mathbf{u}$) et non linéaires par rapport aux déplacements nodaux ($\mathbf{B}_{NL} \mathbf{u}$).

$$\mathbf{E} = \left(\mathbf{B}_L + \frac{1}{2} \mathbf{B}_{NL}(\mathbf{u}) \right) \mathbf{u} \quad (1.13)$$

Dans le cas des lois élastiques ou hyper-élastiques, il existe un potentiel élastique W (ou densité d'énergie) qui est une fonction scalaire du tenseur des déformations. Les contraintes représentent la dérivée par rapport aux déformations de cette fonction scalaire. Dans le cas particulier des modèles isotropes de Saint-Venant-Kirchhoff, on obtient une relation entre contraintes et déformations similaire à la loi de Hooke :

$$\mathbf{S} = \mathbf{D} \mathbf{E} \quad (1.14)$$

\mathbf{D} est le tenseur d'élasticité et \mathbf{S} le second tenseur des contraintes de Piola-Kirchhoff. Comme dans le cas linéaire, par application du principe des travaux virtuels, on obtient le vecteur des forces internes :

$$\mathbf{f}_{int}^{(e)} = - \int_{\Omega^e} (\mathbf{B}_L + \mathbf{B}_{NL}(\mathbf{u}))^T \mathbf{S} dV \quad (1.15)$$

On peut montrer qu'après calculs de dérivation [FCM05], on obtient :

$$\mathbf{K}^{(e)} = \frac{\partial \mathbf{f}_{int}^{(e)}}{\partial \mathbf{u}} = \mathbf{K}_e^{(e)} + \mathbf{K}_\sigma^{(e)} + \mathbf{K}_u^{(e)} \quad (1.16)$$

avec

$$\mathbf{K}_e^{(e)} = \int_{V_0} \mathbf{B}_L^T \mathbf{D} \mathbf{B}_L dV \quad (1.17)$$

$$\mathbf{K}_\sigma^{(e)} = \int_{V_0} \mathbf{S} \frac{\partial \mathbf{B}_{NL}}{\partial \mathbf{u}} dV \quad (1.18)$$

$$\mathbf{K}_u^{(e)} = \int_{V_0} (\mathbf{B}_L^T \mathbf{D} \mathbf{B}_{NL} + \mathbf{B}_{NL}^T \mathbf{D} \mathbf{B}_L + \mathbf{B}_{NL}^T \mathbf{D} \mathbf{B}_{NL}) dV \quad (1.19)$$

On remarque que la matrice de raideur tangente $\mathbf{K}^{(e)}$ associée à un élément \mathbf{e} se décompose en trois matrices, la matrice de rigidité élastique $\mathbf{K}_e^{(e)}$ qui est constante et identique au cas linéaire, la matrice de rigidité $\mathbf{K}_\sigma^{(e)}$ des contraintes initiales et la matrice de rigidité $\mathbf{K}_u^{(e)}$ des déplacements initiaux.

Selon la loi de comportement, il est possible de tenir compte selon le même principe des non linéarité comme l'hyper-élasticité. Cependant le modèle de comportement le plus utilisé dans le contexte de la simulation interactive est le modèle de St Venant-Kirschhoff qui est une extension de la loi de Hooke en grand déplacement.

Suivant la même technique d'assemblage que dans le cas linéaire, il est possible de construire un système avec une matrice tangente globale. Le calcul explicite de la matrice tangente est trop prohibitif dans un contexte de simulation temps réel c'est pourquoi on se contente de sommer les contributions des forces internes en chaque noeud, et de résoudre les équations d'équilibre en utilisant une méthode de résolution itérative type gradient conjugué. Même dans ces conditions la résolution est très coûteuse et un calcul explicite en dynamique est souvent préférable surtout dans le cas de tissus mous où les pas de temps peuvent être importants (de l'ordre de la ms). Citons l'approche masse tenseur proposée par Pincinbono dans laquelle les forces internes sont estimées à l'aide d'un schéma numérique explicite. La matrice de masse \mathbf{M} étant considérée comme diagonale, le système d'équations est entièrement découplé au moment de la résolution, le couplage entre les noeuds se faisant par effet de retard sur le calcul des efforts internes. La résolution s'apparente alors à celle utilisée pour les masses ressort.

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} = \mathbf{F}_{\text{ext}} + \mathbf{F}_{\text{int}}(\mathbf{u})$$

Afin d'assurer une bonne stabilité numérique, il est nécessaire d'introduire une matrice d'amortissement \mathbf{C} suivant la méthode de Rayleigh avec $\beta = 0$ ($\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$). Pour pouvoir résoudre ce système d'équations différentielles, on procède par intégration numérique. Il existe plusieurs schémas d'intégration numérique. La méthode de Newmark est très souvent utilisée en éléments finis. On la retrouve par exemple en version explicite pour le temps réel dans le travail de Zhuang [ZC00]. Irving et al. [ITF04] ont présenté une méthode, adaptée à des objets mous et peu élastiques, qui utilise aussi un schéma de Newmark qui rend explicites les termes de raideur et implicites les termes d'amortissement. Dans le cas d'un matériau plus raide, il est nécessaire d'utiliser un schéma implicite pour ne pas avoir des pas de temps trop faibles et des risques d'instabilités numériques.

1.5 Détection de collision

Cet état de l'art est largement inspiré des travaux récents de Hatab [Hat06] effectués dans notre laboratoire.

Les applications avec rendus haptiques sont très exigeantes : elles nécessitent robustesse et interactivité. Cela veut dire un algorithme assez rapide pour la bande passante du rendu haptique (centaines de Hertz) et robuste aux fluctuations des périodes d'échantillonnage et au comportement aléatoire de l'opérateur.

Malgré l'importance de la détection de collisions pour les différents domaines et l'abondance des publications sur ce sujet, il n'existe finalement que très peu de livres dédiés au sujet. Notons toutefois le cinquième chapitre de [MK06] qui présente un état de l'art de la détection de collisions en français. A notre connaissance, Van Den Bergen [vdB04] et Ericson [Eri04] sont les seuls livres entièrement dédiés à la détection de collisions.

Il n'existe pas aujourd'hui un algorithme universel. Chaque application ou domaine de recherche possède ses exigences. On peut trouver dans la littérature beaucoup d'algorithmes pointus et très spécifiques.

Enumérons les critères permettant de choisir un algorithme de détection de collisions :

- La complexité qui décrit le rapport entre la quantité de primitives dans la scène, et le nombre d'opérations nécessaires avant l'obtention d'un résultat. Très peu d'algorithmes proposent un rapport linéaire, sauf pour des cas particuliers.

- La mémoire exigée par l’algorithme pour stocker des résultats intermédiaires ou les données utiles.
- La robustesse et la stabilité.
- La précision considérée comme un compromis avec le temps de calcul.

1.5.1 Recherche de proximité (broad-phase)

Certaines méthodes accélératrices se basent sur la subdivision ou le partitionnement de l’espace et prennent en compte la disposition des objets dans la scène virtuelle. Un premier groupe de méthodes, divise l’espace en une grille uniforme ou variable en taille (figure 1.17). Pour plus de détails voir [Eri04]. Une autre façon est de représenter l’espace par des arbres hiérarchiques. Les « octrees » et les arbres binaires de recherche (BSP « binary space partition ») sont souvent utilisés (figure 1.18) [Hat06]. D’autres méthodes, classent les objets les uns par rapport aux autres pour détecter les paires d’objets proches. « Sweep and prune » est une méthode introduite par Baraff [Bar92]. Elle se base sur l’idée suivante : si deux objets sont en collision, alors leurs projections sur les trois axes ont obligatoirement des parties communes. Autrement dit, si les projections sur un des trois axes n’ont pas de parties communes, alors les objets ne sont pas en collision (figure 1.19). Cette dernière a été utilisée dans plusieurs logiciels de détection de collisions tels I-Collide [CLMP95] et SOLID [vdB04].

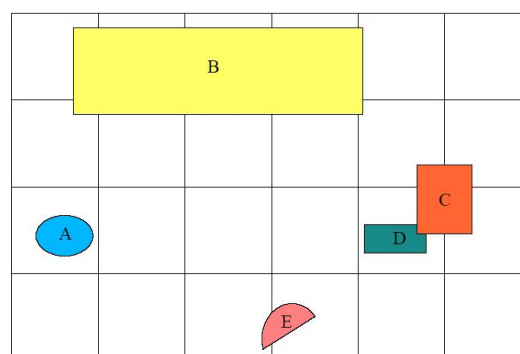


FIG. 1.17 – L’espace est divisé par une grille uniforme.

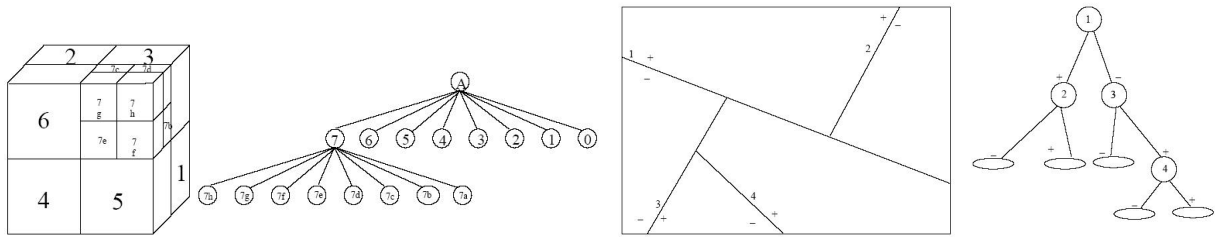


FIG. 1.18 – A droite : Un arbre octree correspondant à l'espace divisé. A Gauche : Un arbre BSP correspondant à l'espace divisé.

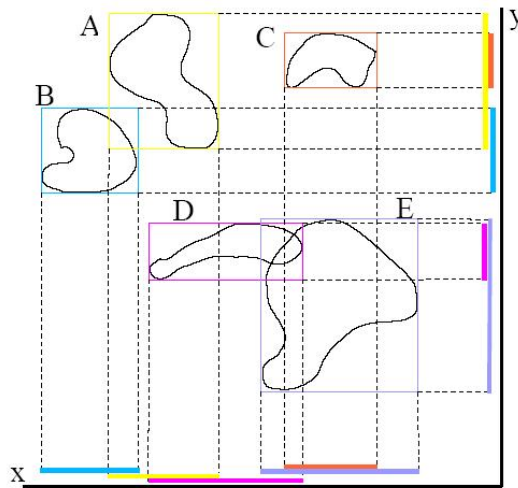


FIG. 1.19 – sweep and prune en 2D.

1.5.2 Volumes englobants

Pour accélérer la détection de collisions, il est possible d'englober les objets dans des volumes pour lesquels le test de détection de collisions est très rapide. Les volumes sont définis avant le début de la simulation. Lorsque les volumes englobants (VE) sont en collision, on ne peut pas conclure à la collision des objets, une recherche plus avancée est nécessaire. Pour être efficace, un VE doit permettre un test rapide de détection de collision, de maximiser le rapport entre volume englobé et englobant, d'être défini rapidement à partir d'un ensemble de sommets, d'être mis à jour facilement en cas de rotation et/ou de translation et ne pas être trop gourmand en mémoire.

Plusieurs types de VE ont été proposés dans la littérature, la figure 1.20 présente les plus courants: les sphères, les boîtes isothétiques, les boîtes orientées par rapport à l'objet, les k-DOPs et les enveloppes convexes. Pour plus de détails voir [Eri04]. Chacun de ces types respecte plus ou moins les propriétés citées plus haut. Par exemple, le test de

chevauchement entre deux sphères est extrêmement rapide ($Dist(O_1, O_2) > (R_1 + R_2)$) et ne requière pas beaucoup de mémoire mais n'englobe pas de façon optimale les objets longs.

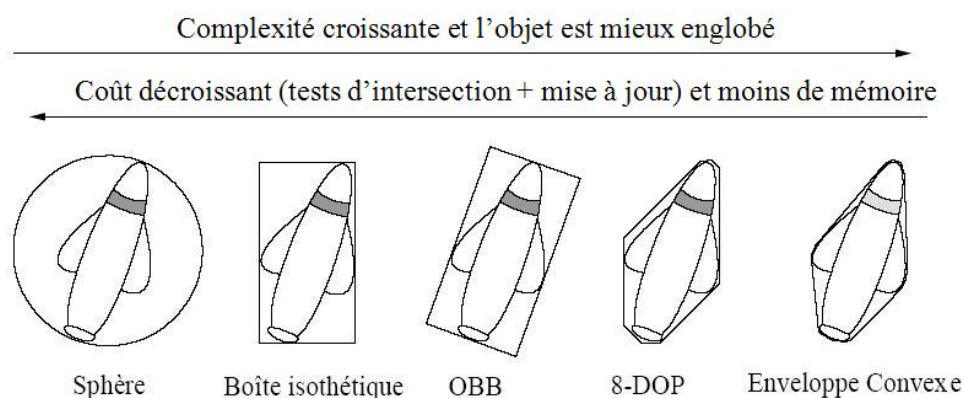


FIG. 1.20 – Différents types de volumes englobants d'après [Eri04].

Les Axes Aligned Bounding Boxes (AABB) qui sont des boîtes alignées sur les axes du repère global (isothétiques) sont très efficaces pour des objets déformables. Van den Bergen [vdB97] les utilise pour gérer la collision discrète entre tous types d'objets. Son code Solid est disponible en ligne⁶. L'avantage majeur des AABBs est la rapidité du test d'intersection, puisqu'il s'agit d'une comparaison entre les valeurs des intervalles sur les axes x , y et z . Pour construire une AABB dans le repère local de l'objet, il suffit de chercher les valeurs minimales et maximales des points de l'objet sur tous les axes. Cependant, une AABB doit rester alignée aux mêmes axes suite aux mouvements. Pour cela, elle doit être mise à jour suite aux rotations.

Il existe d'autres volumes englobants qui englobent de manière plus efficace la forme des objets, mais qui ont des tests d'intersection plus lents. Les Oriented Bounding Boxes (OBB) par exemples sont des boîtes orientées par rapport à l'objet. Elles représentent bien l'objet, mais requiert beaucoup de calcul et de mémoire. Comme une OBB est représentée par un centre, une matrice d'orientation et les extensions dans 3 directions suivant les axes, il faut 60 octets pour stocker une OBB en mémoire. Les enveloppes convexes OBBs, utilisées avec des objets rigides, semblent beaucoup moins performantes avec des objets déformables car leur mise à jour, suite aux déformations, est complexe.

Les k-Dop ou «Discrete oriented polytopes» [MKE03], [KHM⁺98], [Zac98] sont basés sur le même principe que les boîtes AABB (une boîte AABB est une 6-Dop) mais avec des géométries plus complètes. En effet ce sont des enveloppes convexes dont les faces sont déterminées par leurs normales fixées selon un ensemble de k orientations. Le calcul d'un

6. <http://www.win.tue.nl/~gino/solid/>

k-DOP est alors aussi similaire au calcul d'une AABB, puisque il s'agit de chercher les points les plus éloignés dans les directions prédéterminées selon le degré k . L'approximation du volume est plus ou moins bonne et les temps de calcul sont plus ou moins rapides. D'après [Zac98], pour k allant de 6 à 14, il y a très peu de gain vis-à-vis des OBBs, il faudrait donc aller au delà, ce qui entraînerait beaucoup de calculs.

Il est possible d'utiliser la stratégie de VE à l'intérieur des objets d'une manière récursive en construisant des hiérarchies de volumes englobants (HVE). Il s'agit d'une détection approximative qui cherche à diminuer le coût de calculs en réalisant une série de test entre les VEs les plus grossiers vers les plus fin. Ainsi un test rapide de chevauchement élimine les objets qui ne s'interpénètrent pas et la détection d'intersection entre primitives ne se fera que sur un nombre réduit de paires de triangles. Une hiérarchie peut être construite de bas en haut ou de haut en bas. La deuxième façon est la plus rapide, mais produit des hiérarchies avec plus de volume libre [Lin04], (figure 1.21).

Pour construire un arbre hiérarchique de sphères englobantes, une approche du bas vers le haut (Bottom-up) est décrite dans [RKK97]. L'objet est recouvert de petites sphères, qui seront les feuilles de la future hiérarchie. Une fois les feuilles créées, une méthode de type diviser pour régner [Dum98] permet de créer les sphères plus grossières de la hiérarchie interne. Devine [Dev03] a proposé une approche par sphères englobantes pour accélérer le calcul d'intersection continu de Provot [Pro97]. Les sphères sont rapides à remettre à jour après déformation, si la topologie des objets reste inchangée. D'après l'article [JP04], la complexité est en $O(PN)$ avec N le nombre de volumes englobants et P le nombre de triangles à l'intérieur du volume. Cependant, dans cet article, les auteurs proposent de changer uniquement la taille des volumes englobants en gardant intacte la hiérarchie initiale. Ceci n'est valable que pour des objets déformables en petits déplacements.

L'hypothèse cruciale qui est faite ici est basée sur le non-changement de topologie de l'objet durant la déformation. Si l'on ne change pas de topologie, on n'est pas obligé de reconstruire la structure complète de l'arbre à chaque instant. On peut alors optimiser une remise à jour de l'arbre. En bottom-up (du précis vers le grossier), on remet l'arbre à jour dans son ensemble. Cette démarche est plus performante lorsque beaucoup de noeuds sont testés pour la collision (multiples collisions ou surfaces de contact). En top-down (du grossier vers le précis), on remet à jour uniquement les branches nécessaires au fur et à mesure de la progression de la collision dans l'arbre. Cette méthode est plus performante dans le cas où il n'y a pas beaucoup de noeuds visités. Larson et Möller [LM01] proposent une méthode hybride de remise à jour : une partie de l'arbre est remise à jour à la manière bottom-up et l'autre partie est remise à jour à la manière top-down.

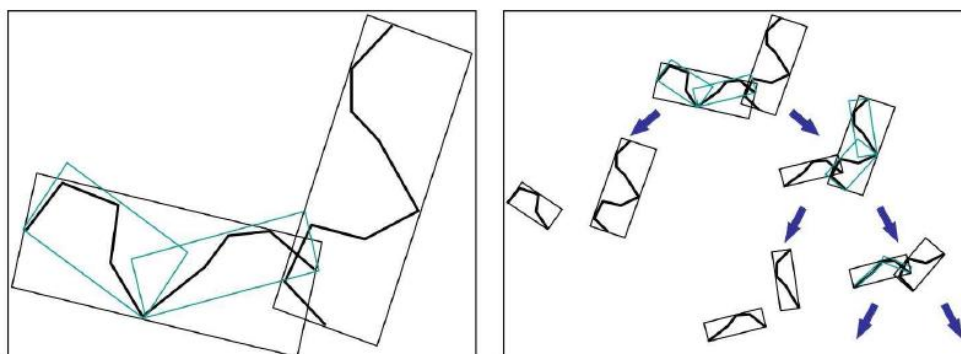


FIG. 1.21 – D’après [Lin04] : Hiérarchie des tests d’intersection entre les volumes englobants. A gauche : Après un test positif sur les boîtes englobantes principales, on recommence un test en utilisant les boîtes du second niveau d’un des objets. A droite : la descente dans la hiérarchie permet de localiser précisément la collision par un nombre de tests réduit.

1.5.3 Détection de collisions pour les objets déformables

Les méthodes de détection de collisions conçues initialement pour des objets solides ne peuvent pas être utilisées directement sur les objets déformables. En effet, les objets solides permettent de calculer des données descriptives hors-ligne et de les réutiliser pour accélérer la détection. Suite aux déformations, ces données ne sont plus valables et elles doivent être recalculées, ou remises à jour lorsque c’est possible. Une adaptation simpliste de ces méthodes aux objets déformables, si elle ne remet pas en cause la stabilité ou la robustesse, peut présenter une dégradation dans la vitesse de convergence telle qu’elle deviendrait inexploitable pour les applications temps réel. Par exemple, les méthodes se basant sur les enveloppes convexes perdent leur complexité d’ordre constant, puisque les déformations peuvent altérer les enveloppes qu’elles utilisent. En revanche, quelques méthodes d’objets solides peuvent être subtilement transformées pour pouvoir s’appliquer aux déformations, comme certaines hiérarchies de volumes englobants par exemple. D’autres méthodes ont été conçues pour les objets déformables et nous allons en présenter quelques unes dans ce qui suit. L’auto-collision est définie quand deux groupes de triangles du même objet peuvent se toucher. Ceci est du à la déformation de l’objet [Hat06].

1.5.3.1 Adaptation des hiérarchies de volumes englobants

Les méthodes se basant sur les hiérarchies de volumes englobants forment le groupe le plus utilisé par les applications virtuelles, puisqu’elles offrent plus de flexibilités et d’options. Ces méthodes exploitent, grâce à des structures appropriées, des données devant être préparées hors-ligne. Ces structures décrivent les différents niveaux des volumes englobants

et sont pour la plupart assez rapides à mettre à jour lors des déformations. Elles permettent par ailleurs d'interrompre le processus de détection si l'intervalle de temps consacré pour cette opération est épuisé, tout en présentant les résultats avec un seuil connu de précision. Si le processus arrive à son terme, ces méthodes sont capables de trouver toutes les paires de triangles qui sont en collision entre les deux objets. Leurs principales limitations sont les suivantes :

- La nécessité de la mise à jour des hiérarchies lorsque les objets enveloppés changent de forme. Le temps de mise à jour peut être négligeable lorsque les déformations sont rares mais devient rapidement considérable lorsque les objets sont en déformations continues.
- Les hiérarchies qui ont subies une mise à jour ne sont pas aussi performantes que celles qui sont reconstruites. Cela prend plus de temps, puisqu'elles ne sont pas aussi optimales.
- Les structures peuvent être mises à jour seulement pour un nombre fixe de triangles. Les changements importants de topologies, par exemple, l'addition ou la suppression de triangles lors des fissures, nécessiteront la reconstruction complète de la hiérarchie correspondante.

Les différentes hiérarchies de volumes englobants n'ont pas les mêmes propriétés et ne s'adaptent pas de la même façon aux objets déformables. Il faut alors trouver des stratégies pour les mettre à jour rapidement, afin qu'elles décrivent les déformations que les objets ont subies depuis leur construction. Une fois les mises à jour faites, la détection s'effectue de la même façon que pour les hiérarchies des objets solides. Van Den Bergen [vdB97] affirme que seul les hiérarchies des k-DOPs, et par conséquent les AABBs, puisque ce sont un cas particuliers des k-DOPs, permettent de faire une mise à jour des hiérarchies d'une manière optimale. En effet, si nous supposons que les boîtes $B1$ et $B2$ englobent les groupes de triangles $Ts1$ et $Ts2$ respectivement, si Ts est l'union des deux groupes, la plus petite k-DOP englobant Ts est la boîte capable de contenir $B1 \cup B2$ sans avoir à parcourir tous les triangles qu'elle contient. Dans le cas des OBBs et des sphères, le volume englobant ses deux volumes fils est plus grand ou égal à celui qui englobe les triangles qui s'y trouvent. Les volumes ainsi obtenus ne sont pas optimaux. La figure 1.22 démontre cette propriété pour les trois types de volumes englobants.

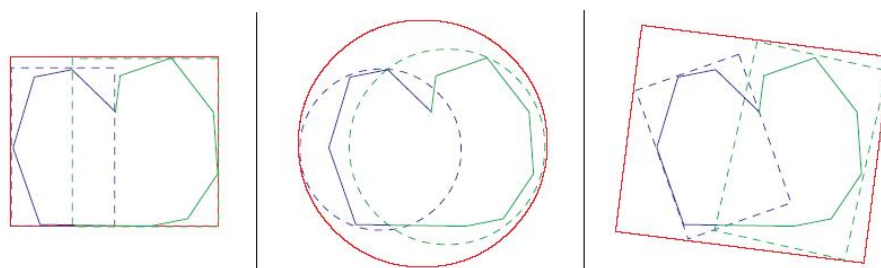


FIG. 1.22 – La mise à jour des trois types de volumes englobants en se basant sur leurs fils [Hat06].

Van Den Bergen [vdB97] propose de mettre à jour chaque hiérarchie de bas en haut. Il utilise dans son article des AABBs pour les tests expérimentaux. Il commence par recadrer les feuilles de l'arbre qui sont les boîtes contenant un triangle chacune. Une fois toutes les feuilles traitées, il suffit de recadrer leurs parents en utilisant les boîtes obtenues au niveau inférieur. La mise à jour d'une hiérarchie contenant n_t triangles effectue $2 \times n_t - 1$ recadrages, elle est donc de complexité linéaire. Il faut souligner qu'une mise à jour de la hiérarchie ainsi effectuée ne change pas la distribution des triangles contenus dans les différents volumes. La hiérarchie est recadrée mais pas reconstruite. La distribution de triangles qui produisait des hiérarchies optimales avant la déformation peut alors produire des hiérarchies qui ne le sont pas après les déformations. Une hiérarchie optimale signifie une hiérarchie qui possède la plus petite somme de tous les volumes qu'elle contient [Hat06].

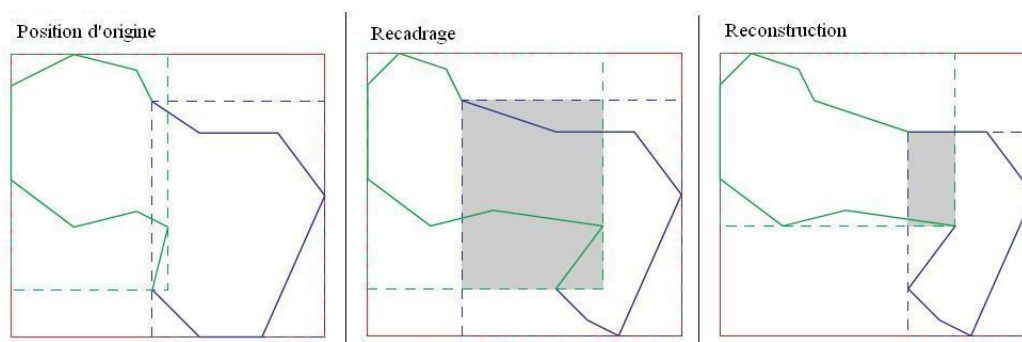


FIG. 1.23 – La mise à jour peut produire une hiérarchie non optimale [Hat06].

La figure 1.23 illustre que la mise à jour peut produire des hiérarchies non optimales et ceci malgré la création de boîtes optimales à chaque niveau. La première partie de la figure 1.23 montre deux boîtes et leur mère avant la déformation. La deuxième partie montre ces boîtes recadrées après la déformation, la troisième partie montre les boîtes optimales qui auraient été obtenues si la hiérarchie était reconstruite et non recadrée. Dans ce cas, les triangles appartenant aux deux boîtes ont été changés. Van Den Bergen affirme que malgré cet inconvénient, la mise à jour des AABBs est dix fois plus rapide que leur reconstruction.

Elle est donc mieux adaptée aux objets déformables.

James et Pai [JP04] proposent de mettre à jour les hiérarchies de sphères englobantes pour les objets déformables dont les déformations sont exprimées par des vecteurs de déplacement de tous les sommets. Chaque sphère dans la hiérarchie est alors mise à jour en utilisant ces vecteurs déplacements pour chacune d'entre elles. Cette méthode est assez rapide mais présente un inconvénient majeur important : le volume de chaque sphère ne peut qu'augmenter, ainsi les hiérarchies ne peuvent que gagner en volume réduisant ainsi la qualité de leur construction.

1.5.3.2 Méthodes stochastiques

La détection de collision dans le cas des simulations interactives présente un compromis entre la vitesse et la précision. Par conséquent, les méthodes stochastiques; qui sont basées sur les probabilités ou les statistiques pour trouver les collisions, ne garantissent pas d'inclure toutes les collisions. Par contre, la vitesse de convergence est assez rapide. Les résultats imprécis sont défendus par le fait que la description surfacique n'est qu'une approximation de la géométrie de l'objet, et qu'il est plus important d'obtenir des résultats rapides qu'exactes.

Klein et Zachmann ont introduit dans [KZ03] une approche probabiliste pour détecter des collisions dans des applications où le temps de calcul est critique. Ils proposent de contrôler le rapport entre temps de calcul et fiabilité des résultats en se basant toujours sur des volumes englobants (d'après l'article, quel que soit leur type). Leur approche dépend de deux paramètres qui jouent sur le rapport précision/temps de calcul. D'après les auteurs, cette approche peut être intégrée dans les codes existants en ajoutant simplement les paramètres de probabilité au sein des noeuds des structures hiérarchiques.

Joussemet et al [JCA06] proposent un algorithme « ESPIONS », qui utilise le concept de l'évolution des populations sur les paires des sommets des deux objets qui vont subir la détection de collisions. Plusieurs procédés sont impliqués dans la sélection des générations suivantes à partir de la génération actuelle comme :

- Le mélange génétique : prendre deux paires (s_{11}, s_{21}) et (s_{12}, s_{22}) puis les mélanger pour obtenir les paires (s_{11}, s_{22}) et (s_{12}, s_{21}) .
- Les compétitions : ce sont les comparaisons qui servent à éliminer les paires possédant les plus grandes distances afin de les remplacer par des paires plus intéressantes.
- Le tirage au hasard des paires de sommets à partir du groupe initial.

Cet article propose d'implémenter ces critères de sélection et d'appliquer la stratégie

de l'évolution sur des objets virtuels solides ou déformables. Dans ce cas, chaque individu est le plus souvent constitué d'une paire de sommets, chacun se trouvant sur un des deux objets impliqués. L'évaluation de chaque individu est le calcul de la distance qui sépare les deux sommets qu'il contient. Un «meilleur» individu est celui qui présente une plus petite distance. Le but est de minimiser ces distances en sélectionnant les paires qui en possèdent les plus petites. Plusieurs paramètres peuvent affecter les résultats de la détection comme la taille des populations des sommets, les critères de sélection ou le nombre de générations (itérations internes de l'algorithme).

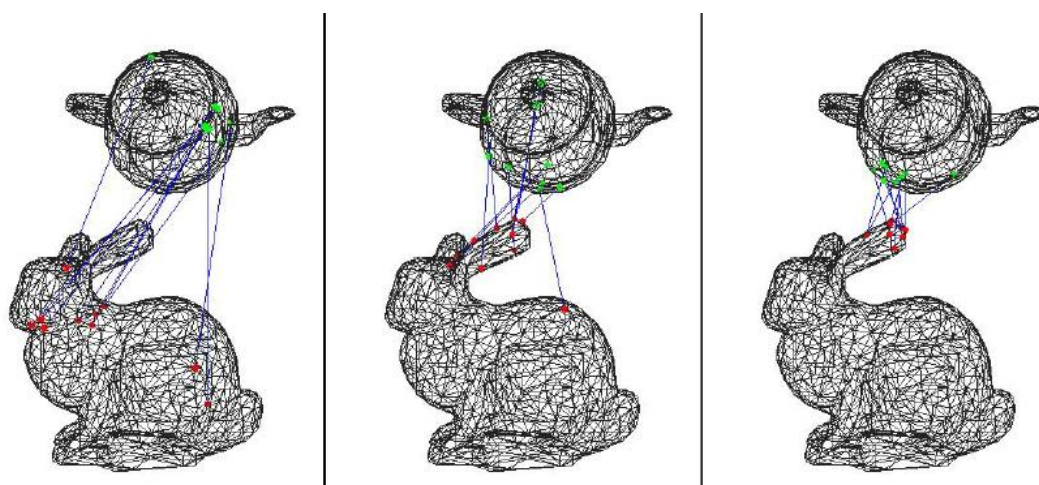


FIG. 1.24 – Les résultats à différents niveaux de populations de la stratégie ESPIONS [Hat06].

La figure 1.24 montre trois itérations de ESPIONS avec les individus trouvés. Cette méthode aboutit finalement à la liste de paires de sommets qui présentent les plus petites distances trouvées et qui sont les individus de la dernière génération. Il faut souligner que malgré sa rapidité, cette méthode, comme toutes les méthodes stochastiques, ne garantit pas de trouver la plus petite distance. Il faut bien choisir les paramètres (le nombre d'individus, le nombre de générations ou encore l'importance des lois de sélection) pour augmenter la probabilité de trouver les résultats complets.

1.5.3.3 Champs de distances

Le principe de cette technique est de calculer pour un certain objet, les champs de distance qui l'entourent. Pour chaque point de R^3 correspond une valeur dans R représentant la plus petite distance vers l'objet. Fisher et Lin [FL01] ont proposé d'utiliser ces champs de distance pour effectuer la détection de collision. Il existe plusieurs façons de calculer les champs de distance. Si un point du deuxième objet se trouve dans la partie du champ présentant une valeur négative d du premier objet, alors il est en collision avec ce dernier, et sa distance d'interpénétration est de d . Au lieu de chercher à calculer la distribution des distances dans l'espace, on peut utiliser une méthode d'exploration de l'ensemble des distances. Ainsi, Johnson [JW03], [JW04] propose une méthode qui extrait les distances

minimales entre deux objets concaves. L'intérêt de cette méthode est d'extraire les zones des différents objets, qui sont « proches » du contact sans être en collision, autrement dit, c'est quand les distances minimales entre les objets deviennent très petites, cela permet une certaine anticipation dans le traitement des contacts. En revanche, elles nécessitent un traitement des contacts qui évite de façon très rigoureuse toute interpénétration entre les objets.

1.5.3.4 Méthode récursive à base de séparation

Maher Hatab [Hat06] s'est intéressé au développement d'une approche de détection de collisions qui favorise la généricité (objets solides et déformables). Sa méthode consiste à recevoir en entrée une soupe de triangles pour chaque objet afin de traiter les diverses transformations : déplacements rigides, déformations, déchirures, cassures, etc. Sa stratégie est basée sur un algorithme récursif à base de séparation qui lui permet d'arriver à une détection de collision efficace en procédant par découpes successives et récursives en éliminant à chaque itération un ensemble de triangles jusqu'au résultat escompté. En effet, le principe de séparation découle d'une simple observation : si on peut identifier une région qui contiendra certainement tous les triangles en collision d'une paire d'objets, on peut alors éliminer ceux qui se trouvent entièrement à l'extérieur de cette zone, sans risque de rater une collision. L'identification d'une telle région peut se faire en fonction d'une direction de séparation *DIR*, qui quant à elle, sera établie par différentes méthodes : [Hat06], [KLM02], [vdB01] et selon le type de volumes englobants utilisé. Le choix de *DIR* permet de minimiser le volume de la zone de collision possible, et permettra donc d'améliorer les performances de la méthode en terme de temps d'exécution. Cette direction de séparation minimale est définie généralement pour des enveloppes convexes comme la direction portant la plus petite translation qui permet de séparer les deux objets. Afin d'accélérer le processus, Maher Hatab [Hat06] a proposé une construction en ligne mais partielle de la hiérarchie de volumes englobants. Il a aussi étudié les différents types de volumes englobants et a testé les effets que peut avoir chaque type de volume sur les résultats finaux de la méthode ainsi que les options en terme de description de la collision qu'ils peuvent fournir (les triangles en collision, les distances d'interpénétration, les normales, etc.). Différentes structures de données ont été conçues pour permettre une implémentation rapide et efficace du code. La robustesse du calcul de cette technique reste encore à améliorer car les contraintes fortes de généricité que Hatab s'est imposé, rendent le problème de calcul de distance difficile à traiter. Sa stratégie est extensible aux objets déformables et prend en compte les auto-collisions moyennant quelques ajouts. Son développement intègre d'une manière optionnelle des interruptions qui permettent d'imposer une borne supérieure sur le temps de calcul au prix d'une précision moindre. La figure 1.25 montre un exemple de détection de collision entre un lapin (de 1500 triangles et 752 sommets) et une théière (de 992 triangles et 530 sommets).

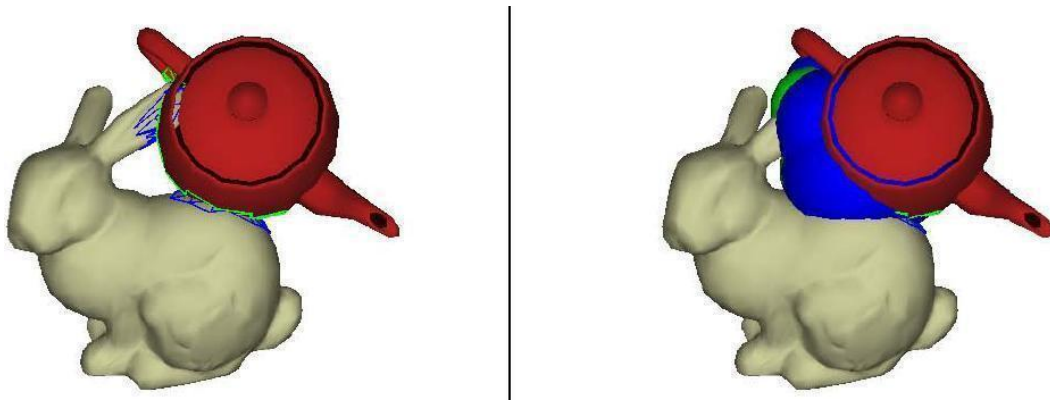


FIG. 1.25 – Exemple de détection de collision selon la méthode à base de séparation et hiérarchie de volumes englobants partielle [Hat06].

1.6 Modélisation des contacts multiples

On peut distinguer 3 types de méthode pour traiter des contacts multiples :

- les méthodes avec fonction de pénalité explicite
- les méthodes des collisions avec contacts successifs
- les méthodes des contraintes avec contacts simultanés

Les méthodes avec fonction de pénalité explicite sont des méthodes qui modélisent la force de contact par un ressort et un amortisseur virtuels. La force de contact est alors proportionnelle à la distance et la vitesse d'interpénétration des corps en contact. Il est donc très facile de traiter des contacts multiples par cette méthode. Cependant elle engendre quelques difficultés. La première est l'existence de l'interpénétration qui n'est pas très satisfaisante visuellement et qui peut engendrer des difficultés au niveau du détecteur de collisions. En effet, une fois qu'il y a interpénétration en un point de contact sur une facette, il faut s'assurer de ressortir par cette même facette non pas par une autre facette plus proche et orientée différemment. La deuxième difficulté est le choix de la valeur des paramètres de raideur et d'amortissement. Une raideur trop forte conduit à des instabilités numériques propres à la résolution d'Equations Différentielles Ordinaires «raides». Enfin la prise en compte du frottement sec augmente les risques d'instabilité numérique. En général on ignore les phases d'adhérence aux points de contact et on ne considère que les phases de glissement avec frottement. On peut dire que la méthode de pénalité explicite a beaucoup de points communs avec la méthode masse-ressort définie précédemment et peut s'appliquer aux corps déformables [Pro95].

Les méthodes des collisions avec contact successifs traitent un seul contact à la fois comme un impact. Ainsi un objet sur une table peut être considéré par des impacts multiples successifs à très haute fréquence. On itère jusqu'à qu'il n'y ait plus de points en contact. Cette approche est plus adaptée à des problèmes d'impact entre plusieurs corps rigides avec des «temps de vol» entre deux impacts. Ces méthodes sont simples et peu coûteuses en temps de calculs dans le cas de contacts intermittants. Cependant, elle peut conduire à des boucles de calculs infinis dans certaines configurations comme le cas d'une chaîne fermée de solides en contact. La difficulté de cette méthode est d'estimer ce temps de vol afin de trouver l'instant du choc. Stéphane Redon propose une méthode continue pour des corps rigides [RKC02]. Les méthodes des collisions ne s'appliquent pas à des contacts entre corps déformables qui sont des contacts persistants.

La méthode des contraintes avec contacts simultanés traite tous les contacts à un instant donné, on n'avance pas dans la solution tant que toutes les forces de contact ne sont pas calculées. C'est un problème d'optimisation qui est formulé au niveau des forces et des accélérations mais qui peut être également formulé au niveau des vitesses et des impulsions (méthode impulsionnelle). Dans la littérature, de nombreuses méthodes ont été développées pour traiter ce genre de problèmes conjointement avec la méthode des éléments finis. Cette approche permet de traiter aussi bien des contacts persistants que des contacts intermittants. Elle est parfaitement adaptée aux contacts entre corps déformables. Dans le cas des corps rigides compte tenu des discontinuités en cas de choc une formulation impulsionnelle est préférable [ESHD05]. C'est dans ce cadre que seront abordées dans le chapitre 2 la modélisation et la formulation d'un problème de contact entre objets déformables.

1.7 Conclusion

Comme nous l'avons vu, il existe de nombreuses méthodes pour modéliser les objets déformables et, il est de plus en plus nécessaire de mettre au point une stratégie de calcul permettant de résoudre les forces de contact entre ces objets. Il existe déjà des plateformes de simulation comme SOFA, développée par l'INRIA, dans lesquelles les corps sont modélisés de différentes manières. Un solveur de contact, qui soit indépendant des modèles de déformation constitue donc un premier challenge. Un autre challenge consiste à ce que ce solveur de contact restitue avec précision les forces de contact pour un rendu haptique réaliste ce qui dans le cas des contacts frottants entre objets déformables peut paraître ambitieux. Pour des raisons de stabilité numérique, ce solveur ne doit pas être basé sur des méthodes de résolution avec fonctions de pénalité ainsi que l'a déjà fait remarquer Christian Duriez dans sa thèse [Dur04]. En effet la simulation avec retour haptique doit tenir compte de l'opérateur humain dans la boucle de simulation, il semble difficile de demander à cet opérateur de contrôler une position visuellement si la force retournée dépend d'une interpénétration. Il est nécessaire de s'orienter vers des méthodes de résolution par

contraintes.

Dans la simulation avec interface haptique, le plus important est d'avoir un bon rendu kinesthésique ce qui nécessite un calcul des forces de contact précis et rapide. La modularité, la précision, la rapidité sont de propriétés qui semblent s'opposer ou tout du moins faire appel à des compromis. Même si il existe des simulateurs chirurgicaux avec des rendus visuels particulièrement attractifs, ils ne seront jamais réalistes tant que l'on ne pourra pas simuler en temps réel des contacts entre tissus mous et tant que l'on ne pourra pas ressentir à travers l'interface haptique les zones d'adhérence ou de glissement entre les organes manipulés par l'opérateur. L'objectif du chapitre 2 est de mettre en place des algorithmes de résolution par contrainte qui pourraient réaliser ces compromis en partant du principe de respecter rigoureusement les équations du problème. Ces algorithmes sont nécessairement itératifs et soumis à des critères de convergence afin de contrôler la solution.

Modélisation et formulation du contact frottant entre objets déformables

2.1 Introduction

L'utilisation d'une interface haptique augmente le réalisme d'une simulation. En effet en plus du retour visuel 3D, le retour d'effort à partir du calcul des forces de contact via une interface haptique permet une meilleure immersion de l'opérateur dans la simulation. De nombreuses applications en réalité virtuelle sont concernées par ce type de travaux comme les simulateurs d'applications médicales mais également les nouvelles plates-formes CAO où l'on s'intéresse à simuler des tâches d'assemblage et de démontage de pièces mécaniques. Dans l'approche médicale, il est indispensable de prendre en compte la déformation des corps en contact de manière réaliste afin de restituer un rendu haptique cohérent entre la perception mentale que l'on a des objets que l'on manipule (ou que l'on touche) et la perception réelle ressentie à travers l'interface haptique utilisée. Le challenge est considérable car il est admis qu'il faut une fréquence de calcul supérieure à 300 Hz pour des objets mous (1KHz pour des objets durs) afin de restituer un bon «toucher» (rendu kinesthésique). Il est donc nécessaire d'optimiser les trois phases suivantes de calculs intervenant dans le processus de simulation :

- Le calcul des déformations et des déplacements induits.
- La détection de collision.
- Le calcul des forces de contact avec frottement.

Dans le cadre de cette thèse, l'objectif est d'optimiser la phase du calcul des forces de contact dans le cas de contacts multiples et simultanés avec frottement. L'idéal étant de mettre au point un «solveur» de force de contact indépendant des techniques utilisées dans les autres phases de calcul et dont on peut contrôler la stabilité numérique et la précision en fonction de la puissance de calcul disponible, des propriétés de l'interface haptique (force maximale, impédance mécanique de l'interface, etc.) et de la contrainte temps réel lié au rendu haptique.

L'analyse des problèmes de contact avec frottement est d'une grande importance dans de multiples domaines d'ingénierie. Dans la littérature, de nombreuses méthodes ont été développées pour traiter ce genre de problèmes conjointement avec la méthode des éléments finis :

- La méthode des multiplicateurs de Lagrange : Dans cette méthode, les contraintes liées au contact unilatéral et au frottement sont introduites dans la formulation par le biais d'inconnues supplémentaires (les multiplicateurs de Lagrange). Cette méthode permet de vérifier exactement les conditions de contact mais elle a le désavantage d'augmenter la taille du système à résoudre. Nous trouvons quelques exemples d'utilisation de cette méthode dans [HTS⁺76], [NW86], [BN91], [CTK91], [PS98], [SP98].
- La méthode de fonction de pénalisation : Dans cette autre méthode, la loi de contact unilatéral ainsi que la loi de frottement sont régularisées par l'intermédiaire d'un ou deux coefficients de pénalité suivant que l'on traite un problème de statique ou de dynamique [CT71], [TY73], [KS81], [KO84], [CB86], [Par89]. Cette méthode a pour principal avantage sa simplicité d'implantation. Cependant, il faut souligner que cette méthode ne permet pas de vérifier avec exactitude les conditions de contact. L'importance de la pénétration autorisée dépend du choix (difficile) des coefficients de pénalité [Hun92] qui n'ont pas de signification physique et qui doivent être déterminés pour chaque cas étudié. De plus on modifie la structure bande de la matrice de rigidité des corps en contact.
- La méthode du Lagrangien augmenté : Dans cette dernière méthode, les déplacements et les réactions de contact sont déterminés simultanément [[JT88], [AC91], [SL92], [dF91], [Kla92], [HC93], [dF98], [CKPS98]]. Cette méthode présente l'avantage de fournir un résultat qui vérifie exactement les conditions de contact et d'éviter les problèmes de singularité constatés lorsque les contraintes algébriques ne sont pas totalement indépendantes.

La formulation du problème de contact avec la méthode de fonction de pénalisation conduit à une seule résolution globale en déplacements car les forces de contact sont explicitement définies en fonction des déplacements. Par contre, dans les deux autres méthodes, les forces de contact sont définies d'une manière implicite et constituent des inconnues supplémentaires qui s'ajoutent aux inconnus en déplacement dans le système d'équations d'équilibre global des corps en contact. Deux approches sont alors possibles pour déterminer ces deux inconnus. La première consiste à les résoudre simultanément d'une manière globale. La deuxième approche dite «mixte» consiste à les résoudre séparément de manière successive. Les forces de contact sont tout d'abord déterminées d'une manière locale par réduction des équations d'équilibre aux points de contact. Les déplacements sont alors obtenus par une résolution globale du système d'équations d'équilibre en considérant les forces de contact comme des forces externes. Cette seconde approche permet de séparer la

non linéarité du comportement mécanique due aux grands déplacements, de la non linéarité due aux lois de contact frottant. En effet, le problème de contact est intrinsèquement non-linéaire car la surface de contact et les forces de contact sont, a priori, inconnues et changent progressivement lorsqu'on applique des forces externes. Elle permet aussi de créer une approche modulaire en introduisant dans le processus de simulation un solveur des forces de contact indépendant du solveur global des déplacements.

Dans ce chapitre nous allons développer une méthode dite de «flexibilité» [FZ75] conforme à cette approche «mixte» comme le montre la figure 2.1. Deux formulations des lois de contact frottant basées sur une formulation lagrangienne augmentée seront aussi présentées. Ces formulations conduisent à la construction d'opérateurs de projection permettant de contrôler la solution améliorant ainsi la stabilité et la précision numérique et en particulier pour tout ce qui touche à la simulation des phénomènes de frottement.

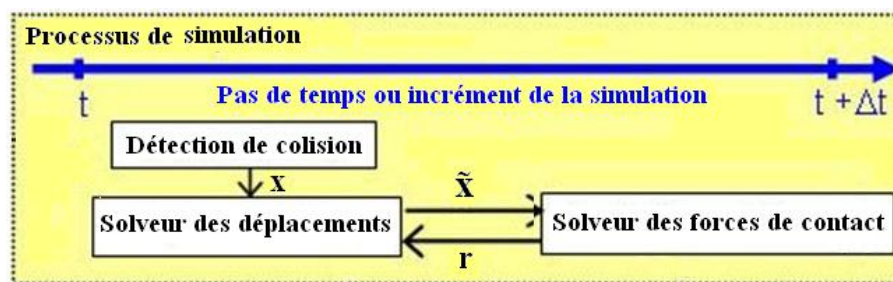


FIG. 2.1 – Diagramme de la méthode de flexibilité.

2.2 Principes généraux de résolution d'un modèle discret

2.2.1 Discrétisation spatiale

Quelque soit la méthode (Particulière, Eléments Finis, Différences Finis ...) pour discrétiser spatialement un modèle physique, nous sommes amenés à établir un système d'équations du type :

$$\mathbf{F}_{\text{int}} + \mathbf{F}_{\text{ext}} + \mathbf{R} = \mathbf{0} \quad (2.1)$$

\mathbf{F}_{int} et \mathbf{F}_{ext} représentent respectivement les forces internes et les forces externes aux solides en contact, s'exerçant sur les particules (modèles particulières) ou sur les noeuds de contrôle (méthode avec grille ou maillage). \mathbf{R} représente les forces de contact. \mathbf{F}_{int} , \mathbf{F}_{ext} , et \mathbf{R} sont sous une forme algébrique de tableaux uni-colonnes composés de coordonnées et définis dans un repère global \mathcal{R}_0 . La dimension de ces tableaux est $3 \times n$ dans le cas d'un problème 3D ($2 \times n$ cas 2D) où n représente le nombre de particules (ou noeuds/points de contrôle). Les forces internes peuvent se décomposer de la manière suivante :

$$\mathbf{F}_{\text{int}} = \mathbf{F}_{\text{in}}(\ddot{\mathbf{X}}) + \mathbf{F}_{\text{d}}(\mathbf{X}, \dot{\mathbf{X}}) \quad (2.2)$$

\mathbf{X} , $\dot{\mathbf{X}}$, $\ddot{\mathbf{X}}$ sont respectivement les positions, vitesses et accélérations des particules ou des noeuds sous leur forme algébrique. \mathbf{F}_{in} représente le vecteur des forces inertielles fonction des accélérations et \mathbf{F}_{d} le vecteur des forces liées aux déformations qui est fonction des positions et/ou des vitesses (dans le cas de lois de comportement du matériau avec vitesses de déformation).

Le système 2.1 est un système d'équations différentielles du second ordre qui est, sauf exception, non linéaire. Les 4 principales causes de non linéarités sont :

- les grands mouvements,
- les grandes déformations,
- les lois de comportement (plasticité, hyper élasticité...),
- les lois de contact avec frottement.

Comme nous avons pu le constater dans le chapitre précédent avec l'étude de la méthode masse-ressort et de la méthode des éléments finis, les méthodes de discrétisation spatiale diffèrent principalement dans la manière de calculer les forces internes et de prendre en compte le voisinage d'interaction autour de chaque particule ou noeud de contrôle.

2.2.2 Discrétisation temporelle

Il n'existe pas, sauf cas particulier, de solution analytique au système 2.1. On réalise donc une discrétisation temporelle. On ne résout le système qu'à une suite discrète d'instant t_{n+1} tel que $t_{n+1} = t_n + \Delta t$ où Δt représente le pas de temps de discrétisation. Les opérateurs différentiels en temps sont alors approximés à partir d'un schéma d'intégration numérique explicite, semi-implicite ou implicite. On est ainsi ramené à la résolution d'un système algébrique non linéaire sous contraintes dont la forme générale dépend du schéma utilisé.

2.2.2.1 Discrétisation temporelle par un schéma numérique d'intégration implicite

Dans ce cas la forme générale devient :

$$\mathbf{F}_{\text{int}}(\mathbf{X}_{n+1}) + \mathbf{F}_{\text{ext}}(t_{n+1}) + \mathbf{R}(t_{n+1}) = \mathbf{0} \quad (2.3)$$

t_{n+1} et Δt étant considérés comme des paramètres de calcul, la seule inconnue est $\mathbf{X}_{n+1} = \mathbf{X}(t_{n+1})$ car les variables $\mathbf{X}_n = \mathbf{X}(t_n)$, $\dot{\mathbf{X}}_n = \dot{\mathbf{X}}(t_n)$, $\ddot{\mathbf{X}}_n = \ddot{\mathbf{X}}(t_n)$ de l'état précédent sont supposées connues.

Prenons l'exemple simple du schéma d'Euler implicite :

$$\dot{\mathbf{X}}_{n+1} = \frac{\mathbf{X}_{n+1} - \mathbf{X}_n}{\Delta t}, \quad \ddot{\mathbf{X}}_{n+1} = \frac{\dot{\mathbf{X}}_{n+1} - \dot{\mathbf{X}}_n}{\Delta t} \quad (2.4)$$

Soient les équations 2.1 exprimées à l'instant t_{n+1} (avec $\mathbf{X}_{n+1} = \mathbf{X}(t_{n+1})$, $\dot{\mathbf{X}}_{n+1} = \dot{\mathbf{X}}(t_{n+1})$, $\ddot{\mathbf{X}}_{n+1} = \ddot{\mathbf{X}}(t_{n+1})$), en y substituant les expressions 2.4, on obtient bien la forme générale 2.3.

Pour résoudre 2.3, on utilise un processus itératif dérivé de la méthode Newton Raphson :

$$\begin{cases} \mathbf{K}_T \delta \mathbf{X} - \mathbf{R}(t_{n+1})^{i+1} = \mathbf{F}_{\text{int}}(\mathbf{X}_{n+1}^i) + \mathbf{F}_{\text{ext}}(t_{n+1}) \\ \mathbf{X}_{n+1}^{i+1} = \mathbf{X}_{n+1}^i + \delta \mathbf{X} \\ \mathbf{X}_{n+1}^0 = \mathbf{X}_n \end{cases} \quad (2.5)$$

$\delta \mathbf{X}$ est un incrément en position. \mathbf{K}_T représente la matrice tangente de la résolution avec $\mathbf{K}_T = -\frac{\partial \mathbf{F}_{\text{int}}(\mathbf{X}_{n+1}^{i+1})}{\partial \mathbf{X}_{n+1}^i}$. On pose $\Delta \mathbf{X}$ l'incrément total en position qui représente le déplacement effectué entre la configuration définie à l'instant t_n et la configuration définie à l'instant t_{n+1} .

Le principe de base de la résolution consiste donc à calculer $\Delta \mathbf{X} = \mathbf{X}_{n+1}^{i+1} - \mathbf{X}_n$ et $\mathbf{R}(t_{n+1})^{i+1}$ à chaque itération afin de vérifier les équations d'équilibre aux instants t_{n+1} en connaissant les variables d'états à l'instant t_n . La convergence est obtenue lorsque le résidu des équations d'équilibre est nul, le résidu étant défini par :

$$\text{Rés}(\mathbf{X}_{n+1}^{i+1}) = \mathbf{F}_{\text{int}}(\mathbf{X}_{n+1}^{i+1}) + \mathbf{R}(t_{n+1})^{i+1} + \mathbf{F}_{\text{ext}}(t_{n+1}) \quad (2.6)$$

L'incrément $\delta \mathbf{X}$ est contraint par la présence des forces de contact $\mathbf{R}(t_{n+1})$, on appellera par la suite $\delta \tilde{\mathbf{X}}$, l'incrément libre c'est-à-dire l'incrément calculé en l'absence des forces de contact.

Remarque 1: Si l'on effectue une seule itération alors la résolution est dite semi-implicite et $\delta \mathbf{X} = \Delta \mathbf{X}$. Dans ce cas le résidu n'est pas nul sauf dans le cas où la matrice tangente est constante (cas linéaire).

Remarque 2: Lorsque l'on néglige les effets inertiels ($\mathbf{F}_{\text{in}} \rightarrow 0$) avec l'absence de considération des vitesses au niveau des déformations (comportement élastique, hyper élastique, élasto-plastique) et de frottements visqueux, la notion de temps disparaît. Cependant la résolution demeure un problème évolutif décrivant une transformation d'un état de référence

\mathbf{X}_0 (état non déformé) à un état d'équilibre \mathbf{X}_1 (avec déformation) et $\Delta\mathbf{X} = \mathbf{X}_1 - \mathbf{X}_0$. Le processus de résolution devient :

$$\begin{cases} \mathbf{K}_T \delta\mathbf{X} - \mathbf{R}^{i+1} = \mathbf{F}_{\text{int}}(\mathbf{X}_1^i) + \mathbf{F}_{\text{ext}} \\ \mathbf{X}_1^{i+1} = \mathbf{X}_1^i + \delta\mathbf{X} \\ \mathbf{X}_1^0 = \mathbf{X}_0 \end{cases} \quad (2.7)$$

Avec $\mathbf{K}_T = -\frac{\partial \mathbf{F}_d(\mathbf{X}_1^i)}{\partial \mathbf{X}_1}$. Dans le cas de fortes non linéarités, on peut passer par des équilibres intermédiaires successifs en faisant évoluer la fonction de charge de manière incrémentale :

$$\mathbf{F}_{\text{ext}}(n+1) = \mathbf{F}_{\text{ext}}(n) + \Delta\mathbf{F}_{\text{ext}} \quad \Rightarrow \quad \mathbf{X}_{n+1} = \mathbf{X}_n + \Delta\mathbf{X} \quad (2.8)$$

Le processus de résolution d'un équilibre statique devient ainsi équivalent au cas général défini dans le cas de la dynamique.

2.2.2.2 Discrétisation temporelle par un schéma numérique d'intégration explicite

Dans le cas particulier d'un schéma d'intégration numérique explicite, les variables \mathbf{X}_{n+1} et $\dot{\mathbf{X}}_{n+1}$ sont calculées comme des fonctions explicites des variables d'états définies à l'instant t_n . Les seules inconnues qui demeurent sont $\ddot{\mathbf{X}}_{n+1}$ et $\mathbf{R}(t_{n+1})$, le système à résoudre devient :

$$\mathbf{M} \ddot{\mathbf{X}}_{n+1} - \mathbf{R}(t_{n+1}) = \mathbf{F}_d(\mathbf{X}_{n+1}, \dot{\mathbf{X}}_{n+1}) + \mathbf{F}_{\text{ext}}(t_{n+1}) \quad (2.9)$$

Avec $\mathbf{F}_{\text{in}} = -\mathbf{M} \ddot{\mathbf{X}}_{n+1}$ où \mathbf{M} représente la matrice de masse du système qui est constante, définie positive. Les forces de contact $\mathbf{R}(t_{n+1})$ contraignent le mouvement sur l'incrément total $\Delta\mathbf{X} = \mathbf{X}_{n+2} - \mathbf{X}_{n+1}$ (à condition d'utiliser un schéma au moins d'ordre 2). Comme précédemment $\Delta\tilde{\mathbf{X}}$ représente l'incrément libre c'est-à-dire celui calculé en l'absence de force de contact. Prenons l'exemple du schéma de Newmark explicite :

$$\begin{cases} \mathbf{X}_{n+1} = \mathbf{X}_n + \Delta t \dot{\mathbf{X}}_n + \frac{\Delta t^2}{2} \ddot{\mathbf{X}}_n \\ \dot{\mathbf{X}}_{n+1} = \dot{\mathbf{X}}_n + \frac{\Delta t}{2} \ddot{\mathbf{X}}_n \end{cases} \quad (2.10)$$

Après résolution du système 2.9, on en déduit l'incrément contraint :

$$\Delta\mathbf{X} = \frac{\Delta t^2}{2} \ddot{\mathbf{X}}_{n+1} + \Delta t \dot{\mathbf{X}}_{n+1} \quad (2.11)$$

Dans cette approche les équations 2.9 ne sont pas exactes car les forces internes liées à la déformation sont calculées à partir d'une approximation. Ceci induit des problèmes d'instabilité numérique bien connus lorsque le pas de temps est trop grand. Le schéma est nécessairement conditionnellement stable. L'avantage de cette approche par rapport à l'utilisation d'un schéma numérique d'intégration implicite est la rapidité de calcul car la résolution est directe (non itérative) et la matrice de masse est facilement inversible. Dans la méthode par éléments finis, la matrice masse est condensée sur la diagonale en négligeant ainsi les couplages inertiels entre les noeuds ce qui rend la méthode encore plus efficace.

2.2.3 Conclusion

On vient de voir que quelque soit la méthode utilisée au niveau de la discrétisation spatiale ou temporelle, on est amené à résoudre des forces de contact ainsi qu'un incrément en position (ou en déplacement) contraint induit par ces forces de contact. Ce calcul se fait au sein d'un processus itératif (formulation implicite) ou direct (formulation explicite). Pour mener à bien cette résolution, il nous faut maintenant considérer les lois de contact. Nous nous limitons ici aux conditions de Signorini et à la loi de Coulomb qui régissent le frottement sec. Ces lois sont dites «raides» et constituent toujours un «challenge» au niveau de la résolution numérique.

2.3 Formulation classique des lois de contact de frottements secs

2.3.1 Définition du repère local de contact

Pour des raisons de simplicité et de clarté, nous considérons uniquement deux corps déformables Ω_1 et Ω_2 en contact. Sur la figure 2.2, le point P appartenant à Ω_1 est un point de contact potentiel et Q est le point le plus proche de P sur Ω_2 . En effet Q est le point projeté de P sur la surface de Ω_2 qui sera considéré par la suite comme l'origine du repère local représenté sur figure 2.2. Le vecteur unitaire normal au point de contact \vec{n} est dirigé vers Ω_1 et désigne un vecteur de base de ce repère local.

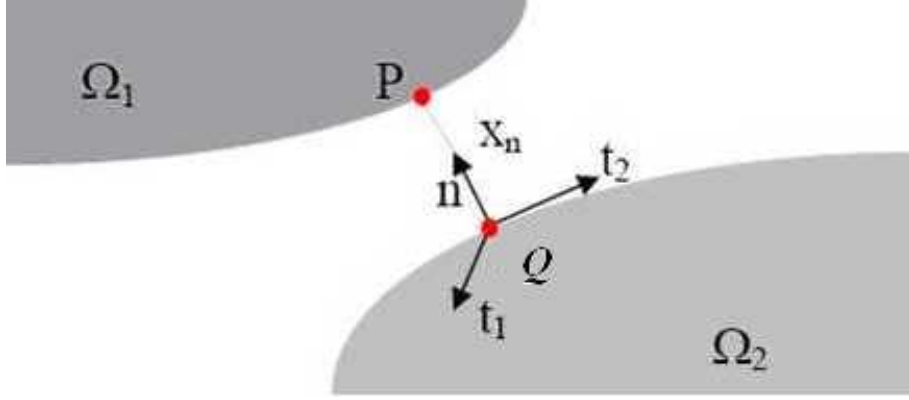


FIG. 2.2 – Repère local de contact.

On définit les notations suivantes :

- $\vec{x} = x_n \vec{n} + \vec{x}_t = \vec{X}(P) - \vec{X}(Q)$ vecteur position de P par rapport au repère local.
- $\vec{r} = r_n \vec{n} + \vec{r}_t$ vecteur force de contact qu'exerce le corps Ω_2 sur le point P.
- $\mathbf{r} = [r_n, r_{t_1}, r_{t_2}]^T$ forme algébrique de \vec{r} exprimée dans le repère local.
- $\mathbf{x} = [x_n, x_{t_1}, x_{t_2}]^T$ forme algébrique de \vec{x} exprimée dans le repère local.
- $\mathbf{r}_t = [0, r_{t_1}, r_{t_2}]^T$ et $\mathbf{x}_t = [0, x_{t_1}, x_{t_2}]^T$ forment les parties tangentielles exprimées dans le repère local.
- $\dot{\mathbf{x}} = [\dot{x}_n, \dot{x}_{t_1}, \dot{x}_{t_2}]^T$ représente la vitesse relative de P par rapport au repère local et exprimée dans ce même repère.

2.3.2 Conditions de Signorini

$$x_n = \vec{n} \cdot (\vec{X}(P) - \vec{X}(Q)) \geq 0 \quad (2.12)$$

Cette inégalité traduit la condition de non interpénétration du point P dans Ω_2 . Cette condition est définie dans la configuration d'équilibre c'est-à-dire à l'instant t_{n+1} . On peut en déduire sa forme incrémentale :

$$x_n = g + \Delta x_n \geq 0 \quad (2.13)$$

avec

$$\begin{cases} g = \vec{n} \cdot (\vec{X}_n(P) - \vec{X}_n(Q)) \geq 0 \\ \Delta x_n = \vec{n} \cdot (\Delta \vec{X}(P) - \Delta \vec{X}(Q)) \end{cases} \quad (2.14)$$

Où g représente le «gap» défini à l'instant t_n par le détecteur de collisions et $\tilde{\mathbf{n}}$ la normale à l'instant t_{n+1} . $\tilde{\mathbf{X}}_n(P)$ et $\tilde{\mathbf{X}}_n(Q)$ sont les positions des deux points à l'instant t_n .

Les conditions de Signorini sont définies par les trois conditions :

$$\text{Signorini}(x_n, r_n) \Leftrightarrow x_n \geq 0, r_n \geq 0 \text{ et } x_n r_n = 0 \quad (2.15)$$

La première condition traduit la non interpénétration, la deuxième l'unilatéralité de la force de contact (force non adhésive), la troisième la nécessaire complémentarité entre la distance et la force de contact. Ces conditions peuvent être représentées par le graphe de la figure 2.3.

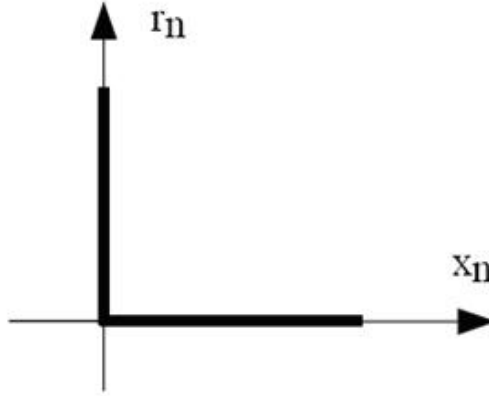


FIG. 2.3 – Graphe des conditions de Signorini.

On s'aperçoit que ce n'est pas un graphe d'application, c'est à dire que l'on ne peut écrire la relation ni sous la forme $x_n = g(r_n)$, ni sous la forme $r_n = g(x_n)$. Cette particularité constitue une difficulté au niveau de la résolution numérique.

2.3.3 Loi de Coulomb

Dans le cas du contact-frottement isotrope, le modèle de Coulomb s'écrit :

$$\text{Coulomb}(\dot{\mathbf{x}}_t, \mathbf{r}_t) \Leftrightarrow \begin{cases} \|r_t\| < \mu r_n & \text{si } \|\dot{\mathbf{x}}_t\| = 0 \quad (\text{adhérence}) \\ r_t = -\mu r_n \frac{\dot{\mathbf{x}}_t}{\|\dot{\mathbf{x}}_t\|} & \text{si } \|\dot{\mathbf{x}}_t\| \neq 0 \quad (\text{glissement}) \end{cases} \quad (2.16)$$

μ représente le coefficient de frottement qui dépend de la nature des matériaux en contact. La forme incrémentale devient :

$$\text{Coulomb}(\Delta \mathbf{x}_t, \mathbf{r}_t) \Leftrightarrow \begin{cases} \|r_t\| < \mu r_n & \text{si } \|\Delta \mathbf{x}_t\| = 0 \\ r_t = -\mu r_n \frac{\Delta \mathbf{x}_t}{\|\Delta \mathbf{x}_t\|} & \text{si } \|\Delta \mathbf{x}_t\| \neq 0 \end{cases} \quad (2.17)$$

$\Delta \mathbf{x}_t = \mathbf{x}_t(t_{n+1}) - \mathbf{x}_t(t_n)$ mais du fait que le point Q soit le point projeté orthogonalement de P, on a $\mathbf{x}_t(t_n) = \mathbf{0}$ ce qui implique que $\Delta \mathbf{x}_t = \mathbf{x}_t(t_{n+1}) = \mathbf{x}_t$. On a donc :

$$\text{Coulomb}(\mathbf{x}_t, \mathbf{r}_t) \Leftrightarrow \begin{cases} \|r_t\| < \mu r_n & \text{si } \|\mathbf{x}_t\| = 0 \\ r_t = -\mu r_n \frac{\mathbf{x}_t}{\|\mathbf{x}_t\|} & \text{si } \|\mathbf{x}_t\| \neq 0 \end{cases} \quad (2.18)$$

Ces conditions peuvent être représentées par le graphe de la figure 2.4. On remarque que l'on n'a toujours pas de graphe d'application dans le cas de non glissement. De plus, il y a une forte discontinuité au voisinage de la vitesse de glissement nulle ce qui pose également des problèmes de résolution numérique. Une autre difficulté est que le graphe présuppose la connaissance de la direction de glissement ce qui n'est possible qu'en 2D.

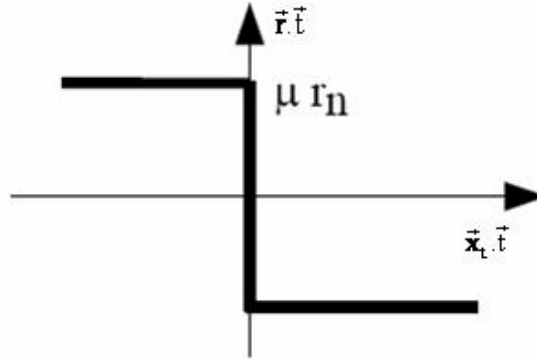


FIG. 2.4 – Graphe de la loi de Coulomb.

Le domaine de définition de la force de contact est un cône K_μ appelé «cône de Coulomb» représenté sur la figure 2.6 et qui met en évidence le couplage entre l'effort normal et la norme de la force tangentielle. On définira $\text{Int}(K_\mu)$ et $\text{Bd}(K_\mu)$ respectivement l'intérieur et le bord du cône de Coulomb. $K_\mu = \{\mathbf{r} \in \mathbf{R}^3, \|\mathbf{r}_t\| \leq \mu r_n\}$, $\text{Bd}(K_\mu) = \{\mathbf{r} \in \mathbf{R}^3, \|\mathbf{r}_t\| = \mu r_n\}$, $\text{Int}(K_\mu) = \{\mathbf{r} \in \mathbf{R}^3, \|\mathbf{r}_t\| < \mu r_n\}$.

2.3.4 Conclusion

Les conditions de Signorini et la loi de Coulomb définissent une loi de comportement du point de contact qui est dissipative, complexe et non régulière (non smooth) incluant 3 statuts de contact : non contact, contact avec adhérence et contact avec glissement. Cette loi de comportement peut être traduite par le bloc conditionnel définie par :

Si $r_n = 0$ alors $x_n \geq 0$! Non contact
sinon si $r \in \text{int}(K_\mu)$ alors $x = 0$! Adhérence
sinon $r \in \text{Bd}(K_\mu)$ $\left\{ \exists \lambda > 0, x_t = -\lambda \frac{r_t}{\ r_t\ } \right\}$,	! Glissement
Finsi	

FIG. 2.5 – Bloc conditionnel des 3 statuts de contact

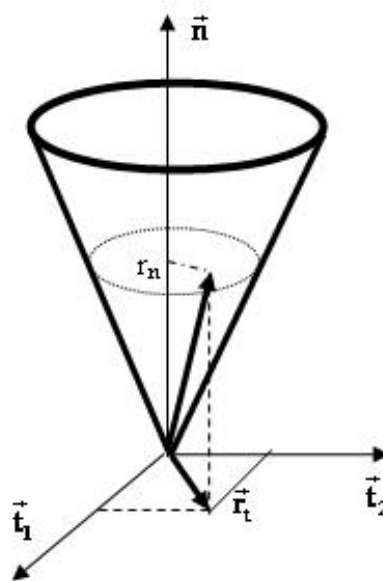


FIG. 2.6 – cône de Coulomb.

2.4 Autres formulations des lois de contact de frottements secs

2.4.1 Formulation par des fonctions de pénalité

Le principe de la méthode consiste à régulariser les fonctions associées aux lois de contact de manière à obtenir des graphes d'application.

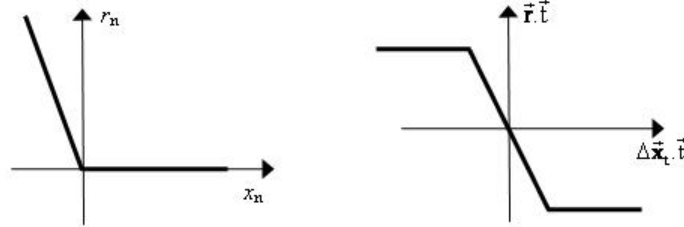


FIG. 2.7 – Régularisation des lois de contact

2.4.1.1 Formulation implicite

Les conditions de Signorini et les conditions de la loi de Coulomb peuvent ainsi être reformulées par :

$$\text{Signorini}(x_n, r_n) \Rightarrow r_n = \text{Proj}_{\mathbb{R}^+}(-\rho_n x_n) \text{ avec } \rho_n > 0 \quad (2.19)$$

$$\text{Coulomb}(\mathbf{x}_t, \mathbf{r}_t) \Rightarrow \mathbf{r}_t = \text{Proj}_{D_\mu}(-\rho_t \mathbf{x}_t) \text{ avec } \rho_t > 0 \quad (2.20)$$

Avec $\text{Proj}_{\mathbb{R}^+}$ et Proj_{D_μ} sont des opérateurs de projection définis par :

$$\text{Proj}_{\mathbb{R}^+}(z \in \mathbb{R}) = \begin{cases} 0 & \text{si } z < 0 \\ z & \text{si } z \geq 0 \end{cases}, \text{Proj}_{D_\mu}(\mathbf{z} \in \mathbb{R}^2) = \begin{cases} h \frac{\mathbf{z}}{\|\mathbf{z}\|} & \text{si } \mathbf{z} \notin D_\mu \\ \mathbf{z} & \text{si } \mathbf{z} \in \text{int}(D_\mu) \end{cases}$$

D_μ représente le disque de Coulomb défini dans \mathbb{R}^2 et de rayon h ($h = \mu r_n$).

Cette formulation constitue une approximation des lois de contact dans laquelle les paramètres ρ_n , ρ_t représentent des raideurs ajoutées respectivement suivant la direction normal et tangentielle au point de contact considéré. En effet, pour avoir une force de contact, il faut nécessairement que $x_n < 0$ c'est-à-dire qu'il y ait interpénétration. Dans cette approche les forces de contact agissent comme des forces de rappel pour compenser l'interpénétration et le glissement. En augmentant la valeur de ces raideurs, on augmente la pente des courbes représentées sur les graphes de 2.7 et on se rapproche ainsi des lois de contact. Cependant, on ne peut pas excéder une certaine limite qui entraînerait un mauvais conditionnement numérique de la matrice tangente. En effet dans cette formulation les forces de contact sont fonctions des déplacements contraints et contribuent au calcul des forces internes modifiant ainsi le calcul de la matrice tangente K_T du système.

Cette formulation se prête bien à la programmation mais elle nécessite la modification de la matrice tangente qui change de forme suivant le statut de chaque point de contact (Non contact, adhérence ou glissement) ce qui dans une résolution numérique itérative est très coûteux.

2.4.1.2 Formulation explicite

Les conditions de Signorini et la loi de Coulomb peuvent être aussi reformulées par :

$$\text{Signorini}(\tilde{x}_n, r_n) \Rightarrow r_n = \text{Proj}_{\mathbb{R}^+}(-\rho_n \tilde{x}_n) \text{ avec } \rho_n > 0 \quad (2.21)$$

$$\text{Coulomb}(\tilde{\mathbf{x}}_t, \mathbf{r}_t) \Rightarrow \mathbf{r}_t = \text{Proj}_{D_\mu}(-\rho_t \tilde{\mathbf{x}}_t) \text{ avec } \rho_t > 0 \quad (2.22)$$

Cette formulation diffère de la précédente car elle ne prend pas en compte les déplacements contraints mais les déplacements libres uniquement. Dans cette formulation les forces de contact s'ajoutent aux calculs des forces externes, la matrice tangente reste inchangée. Pour calculer les déplacements libres, on peut procéder de manière locale par projection des équations d'équilibre au point de contact sans qu'il soit nécessaire de résoudre globalement le système. Les paramètres ρ_n , ρ_t doivent être choisis avec précaution en relation avec les raideurs des deux solides au point de contact. Des valeurs trop élevées entraînent une réactivité trop forte du point de contact et une instabilité numérique. Des valeurs trop faibles entraînent des violations de contraintes (interpénétration, adhérence) trop importantes.

2.4.2 Formulations par des multiplicateurs de Lagrange

2.4.2.1 Formulation par adhérence

On bloque le point de contact dans les 3 directions normales et tangentielles ($x_n = 0$, $\mathbf{x}_t = 0$), on en déduit une force de blocage $\vec{\lambda} = \lambda_n \vec{\mathbf{n}} + \vec{\lambda}_t$ dans le repère local (calculée en prenant en compte les équations d'équilibre). On projette ensuite cette force de manière à satisfaire les conditions de contact. Les conditions de Signorini et la loi de Coulomb peuvent ainsi être reformulées par :

$$\text{Signorini}(x_n, r_n) \Rightarrow r_n = \text{Proj}_{\mathbb{R}^+}(\lambda_n) \quad (2.23)$$

$$\text{Coulomb}(\mathbf{x}_t, \mathbf{r}_t) \Rightarrow \mathbf{r}_t = \text{Proj}_{D_\mu}(\lambda_t) \quad (2.24)$$

2.4.2.2 Formulation par glissement sans frottement

On bloque le point de contact uniquement dans la direction normale ($x_n = 0$), on en déduit une force de blocage $\vec{\lambda} = \lambda_n \vec{\mathbf{n}}$ dans la direction normale (calculée en prenant en compte les équations d'équilibre) et $\tilde{\mathbf{x}}_t$ le déplacement libre suivant les directions tangentielles.

Les conditions de Signorini et la loi de Coulomb peuvent ainsi être reformulées par :

$$\text{Signorini}(x_n, r_n) \Rightarrow r_n = \text{Proj}_{\mathbb{R}^+}(\lambda_n) \quad (2.25)$$

$$\text{Coulomb}(\Delta \mathbf{x}_t, \mathbf{r}_t) \Rightarrow \mathbf{r}_t = -\mu r_n \frac{\tilde{\mathbf{x}}_t}{\|\tilde{\mathbf{x}}_t\|} \quad (2.26)$$

Le choix de l'une ou l'autre formulation dépend du problème considéré. Si c'est un problème dans lequel il y a plus d'adhérence que de glissement alors c'est la première méthode qui est préférable sinon c'est la seconde. D'une manière générale la seconde méthode est plus stable que la première. En effet dans la seconde méthode, l'effet dissipatif lié aux frottements est garanti car la force tangentielle s'oppose aux mouvements tangentiels relatifs, tandis que la première méthode relaxe la force tangentielle au risque de libérer de l'énergie de déformation de manière trop brutale et d'entraîner des problèmes de stabilité numérique. Ces deux méthodes ne simulent pas correctement les changements de statuts (glissement, adhérence).

2.4.2.3 Formulation par un cône de Coulomb à facettes

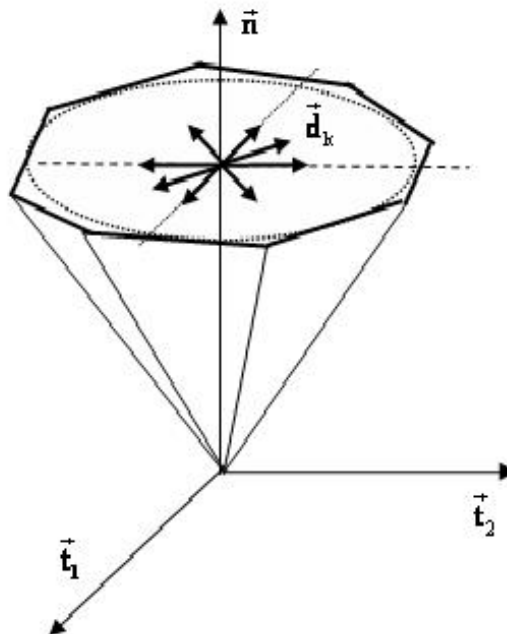


FIG. 2.8 – Cône de Coulomb à huit facettes

On remplace le cône continu par un cône à facettes en considérant h directions privilégiées définies dans le plan tangent par :

$$\mathbf{d}_k = \cos\left(\frac{2(k-1)\pi}{h}\right)\mathbf{t}_1 + \sin\left(\frac{2(k-1)\pi}{h}\right)\mathbf{t}_2 \quad k = 1, h \quad (2.27)$$

On exprime la force tangentielle de contact de manière générale relativement à ces nouveaux vecteurs :

$$\mathbf{r}_t = \sum_{k=1, h} \beta_k \mathbf{d}_k \quad \text{avec} \quad \beta_k \geq 0 \quad (2.28)$$

On montre que les lois de contact peuvent se mettre sous la forme d'inégalité avec des relations complémentaires.

$$\left\{ \begin{array}{l} \text{(a)} \quad \varphi_k = \lambda + \mathbf{d}_k^T \mathbf{x}_t \geq 0, \beta_k \geq 0 \quad (k = 1, h), \text{ rel. compl. : } \varphi_k \cdot \beta_k = 0 \\ \text{(b)} \quad \phi = \mu r_n - \sum_{k=1, h} \beta_k \geq 0, \lambda \geq 0, \text{ rel. compl. : } \phi \cdot \lambda = 0 \\ \text{(c)} \quad x_n \geq 0, r_n \geq 0, \text{ rel. compl. : } x_n \cdot r_n = 0 \end{array} \right. \quad (2.29)$$

On retrouve dans cette formulation les conditions de Signorini et les conditions de la loi de Coulomb mises sous la forme d'inégalités complémentaires c'est-à-dire sous la même forme que les conditions de Signorini. La variable λ est un multiplicateur de Lagrange qui peut être considéré comme une approximation du déplacement relatif au point de contact. Cette formulation prend bien en compte les trois statuts de contact :

Non contact: Dans ce cas $x_n > 0$, (c) implique $r_n = 0$. Après substitution dans (b), on obtient $\beta_k = 0$ ce qui implique qu'il n'y a pas d'effort tangentiel. D'après (a), il existe une infinité de valeurs $\lambda > 0$ possibles.

Glissement: Dans ce cas $\mathbf{x}_t \neq \mathbf{0}$, il existe au moins une direction privilégiée telle que $\mathbf{d}_k^T \mathbf{x}_t < 0$, on pose $\lambda = -\mathbf{d}_k^T \mathbf{x}_t$ ce qui implique $\beta_k = \mu r_n$ d'après (b).

Adhérence: Dans ce cas $\mathbf{x}_t = \mathbf{0}$ ce qui implique $\lambda \geq 0$, deux cas se présentent :

- si $\lambda = 0$ alors d'après (b) la force tangentielle se trouve dans le cône à facettes.
- Si $\lambda > 0$ alors $\beta_k = 0$ d'après (a) et donc d'après (b) on a $\mu r_n = 0$ ce qui constitue un cas non générique.

Cette formulation se prête bien à la programmation mathématique à partir d'une formulation LCP (Linear Complementarity Problem). La formulation LCP s'obtient en rajoutant au système d'inégalités complémentaires défini ci-dessus, un système d'équations reliant linéairement les déplacements contraints et la force de contact dans les trois directions (normale et tangentielles). Ces équations sont obtenues par réduction (ou projection) des équations d'équilibre dans le repère local. La précision de cette méthode dépend du

nombre de facettes considérées sur le cône de Coulomb. Plus il y a de facettes, plus la méthode est précise. L'inconvénient est que l'on rajoute des inconnues ($\lambda > 0, \beta_k$) au problème. La taille du système à résoudre est donc augmentée ce qui peut être pénalisant du point de vue temps de calcul.

2.4.3 Formulations lagrangiennes augmentées

Nous allons voir maintenant deux méthodes qui considèrent la force de contact et les déplacements contraints comme étant deux extremums de potentiels convexes non différentiables. La première méthode introduite par Jean et Moreau [JM92], définit deux potentiels non différentiables appelés «pseudo-potentiels». Le premier est associé aux conditions de Signorini et le deuxième aux conditions de la loi de Coulomb. La deuxième méthode introduite par DeSaxé et Feng [dF91], [dF98], définit un unique potentiel non différentiable appelé «bi-potential» associé à la fois aux conditions de Signorini et à la loi de Coulomb. L'avantage de cette dernière méthode permet de mieux prendre en compte le couplage entre la force normale et la force tangentielle au niveau des lois du contact lors de la résolution numérique.

Ces deux méthodes conduisent à l'élaboration d'opérateurs de projection à partir de considérations issues de l'analyse convexe et de la technique du lagrangien augmenté.

2.4.3.1 Pseudo-potentiels

Conditions de Signorini

$$\text{Signorini}(x_n, r_n) \Leftrightarrow r_n \geq 0 \text{ et } \forall r_n' \geq 0 \quad (r_n' - r_n) x_n \geq 0 \quad (2.30)$$

Montrons qu'il y a bien équivalence :

Si $r_n > 0$ alors il existe au moins un $r_n' > 0$ tel que $r_n' < r_n$ ce qui implique $x_n = 0$ et on vérifie donc bien Signorini.

Si $r_n = 0$ alors $r_n' \cdot x_n \geq 0$ ce qui implique $x_n \geq 0$ et on vérifie donc bien de nouveau Signorini.

On peut montrer que 2.30 définit $-x_n$ comme un sous gradient d'une fonction potentielle non différentiable construite à partir d'une fonction indicatrice, pour plus de précision voir [CJ01].

On peut remplacer l'inégalité dans 2.30 par :

$$\begin{aligned}
(r_n' - r_n) x_n \geq 0 &\Leftrightarrow (r_n' - r_n) \rho x_n \geq 0 \text{ avec } \rho > 0 \\
&\Leftrightarrow (r_n' - r_n) (r_n - r_n + \rho x_n) \geq 0 \text{ avec } \rho > 0 \\
&\Leftrightarrow (r_n' - r_n) (r_n - r_n^*) \geq 0 \text{ avec } r_n^* = r_n - \rho x_n
\end{aligned} \tag{2.31}$$

r_n^* constitue la force de contact normale augmentée.

$$r_n \geq 0 \text{ et } \forall r_n' \geq 0 \quad (r_n' - r_n) (r_n - r_n^*) \geq 0 \Leftrightarrow r_n = \text{Proj}_{\mathbb{R}^+}(r_n^*) \tag{2.32}$$

En effet $r_n > 0$ alors $(r_n' - r_n)$ peut être de signe quelconque ce qui implique $r_n = r_n^*$ par contre si $r_n = 0$ alors $-r_n^* r_n' \geq 0$ ce qui implique $0 \geq r_n^*$.

Loi de Coulomb

$$\text{Coulomb}(\mathbf{x}_t, \mathbf{r}_t) \Leftrightarrow \mathbf{r}_t \in D_\mu, \forall \mathbf{r}_t' \in D_\mu, (\mathbf{r}_t' - \mathbf{r}_t) \mathbf{x}_t \geq 0 \tag{2.33}$$

Montrons qu'il y a bien équivalence :

Si \mathbf{r}_t est à l'intérieur du cône de Coulomb alors il existe un disque centré sur \mathbf{r}_t inclus dans ce cône et on peut avoir toutes les directions $(\mathbf{r}_t' - \mathbf{r}_t)$ possibles ce qui implique que $\mathbf{x}_t = \mathbf{0}$, ce qui définit bien l'adhérence. Si \mathbf{r}_t est au bord du disque de Coulomb (de rayon non nul) alors l'inégalité traduit que $-\mathbf{x}_t$ est dirigé suivant la normale extérieure impliquant $\exists \lambda > 0, \mathbf{x}_t = -\lambda \frac{\mathbf{r}_t}{\|\mathbf{r}_t\|}$, ce qui définit bien le glissement. On peut montrer que 2.33 définit $-\mathbf{x}_t$ comme un sous gradient d'une fonction potentielle non différentiable construite à partir d'une fonction indicatrice, pour plus de précision voir [CJ01].

On modifie l'inégalité dans 2.33 par :

$$\begin{aligned}
(\mathbf{r}_t' - \mathbf{r}_t) \mathbf{x}_t \geq 0 &\Leftrightarrow -(\mathbf{r}_t' - \mathbf{r}_t) \rho \mathbf{x}_t \leq 0 \\
&\Leftrightarrow (\mathbf{r}_t' - \mathbf{r}_t) (\mathbf{r}_t^* - \mathbf{r}_t) \leq 0 \text{ avec } \mathbf{r}_t^* = \mathbf{r}_t - \rho \mathbf{x}_t
\end{aligned} \tag{2.34}$$

En raisonnant de la même manière que précédemment, il est évident que lorsque \mathbf{r}_t est à l'intérieur du disque de coulomb alors nécessairement on a $\mathbf{r}_t^* = \mathbf{r}_t$. Lorsque \mathbf{r}_t est sur le bord du disque cela veut dire que $(\mathbf{r}_t' - \mathbf{r}_t)$ est orienté suivant la normale extérieure du disque. \mathbf{r}_t est donc le projeté de \mathbf{r}_t^* sur le bord du disque lorsque \mathbf{r}_t^* est à l'extérieur (voir figure 2.9).

2.4.3.2 Bi-potentiel

$$\left. \begin{array}{l} \text{Signorini}(x_n, r_n) \\ + \\ \text{Coulomb}(\mathbf{x}_t, \mathbf{r}_t) \end{array} \right\} \Leftrightarrow \left. \begin{array}{l} \mathbf{r} \in K_\mu, \forall \mathbf{r}_t' \in K_\mu, (\mathbf{r}_t' - \mathbf{r}) \mathbf{v} \geq 0 \\ \text{avec } \mathbf{v} = (\mu \|\mathbf{x}_t\| + x_n) \mathbf{n} + \mathbf{x}_t \end{array} \right\} \tag{2.35}$$

Montrons qu'il y a bien équivalence :

Si $\mathbf{r} \in \text{Int}(K_\mu)$ alors $(\mathbf{r}' - \mathbf{r})$ peut prendre n'importe quelle direction ce qui implique que :

$$\mathbf{v} = \mathbf{0} \Leftrightarrow \begin{cases} \mu \|\mathbf{x}_t\| + x_n = 0 \\ \mathbf{x}_t = \mathbf{0} \end{cases} \Leftrightarrow \begin{cases} x_n = 0 \\ \mathbf{x}_t = \mathbf{0} \end{cases} \quad (2.36)$$

Cela définit bien les conditions d'adhérence. Si $\mathbf{r} \in \text{Bd}(K_\mu) - \{\mathbf{0}\}$ alors $-\mathbf{x}_t$ est orthogonal au cône de coulomb impliquant :

$$\begin{cases} \mu \|\mathbf{x}_t\| + x_n = \lambda \mu \\ -\mathbf{x}_t = \lambda \frac{\mathbf{r}_t}{\|\mathbf{r}_t\|} \end{cases} \text{ avec } \lambda > 0 \quad (2.37)$$

λ représente la norme de \mathbf{x}_t ce qui entraîne $x_n=0$. On retrouve les conditions de contact avec glissement. Si $\mathbf{r} = \{\mathbf{0}\}$ alors on a $\forall \mathbf{r}'_t \in K_\mu, \mathbf{r}'_t \cdot (-\mathbf{v}) \leq 0$ ce qui entraîne $-\mathbf{v} \in K_\mu^*$ où K_μ^* représente le cône polaire (ou cône dual) dont les bords sont orthogonaux aux bords du cône de Coulomb (voir Figure 2.10) et qui est défini par $K_\mu^* = \{\mathbf{z} \in \mathbf{R}^3, \mu \|\mathbf{z}_t\| + z_n \leq 0\}$.

$$\begin{aligned} -\mathbf{v} \in K_\mu^* &\Leftrightarrow \mu \|\mathbf{x}_t\| - x_n - \mu \|\mathbf{x}_t\| \leq 0 \\ &\Leftrightarrow 0 \leq x_n \end{aligned} \quad (2.38)$$

Ceci est conforme avec la perte de contact. On modifie l'inégalité dans 2.35 par la même technique du lagrangien augmenté :

$$\begin{aligned} (\mathbf{r}' - \mathbf{r}) \cdot \mathbf{v} \geq 0 &\Leftrightarrow -(\mathbf{r}' - \mathbf{r}_t) \cdot \rho \mathbf{v} \leq 0 \text{ avec } \rho > 0 \\ &\Leftrightarrow (\mathbf{r}' - \mathbf{r}) \cdot (\mathbf{r}^* - \mathbf{r}) \leq 0 \text{ avec } \mathbf{r}^* = \mathbf{r} - \rho \mathbf{v} \end{aligned} \quad (2.39)$$

Si $\mathbf{r} \in \text{Bd}(K_\mu)$ alors $\mathbf{r} = \mathbf{r}^*$, si $\mathbf{r} \in \text{Bd}(K_\mu) - \{\mathbf{0}\}$ alors $(\mathbf{r}^* - \mathbf{r})$ est orienté suivant la normale extérieure du cône de coulomb. \mathbf{r}_t est donc le projeté orthogonal de \mathbf{r}'_t sur le bord du cône de Coulomb lorsque celui-ci est à l'extérieur de K_μ et de K_μ^* (voir Figure 2.10). On a donc :

$$\left. \begin{array}{l} \text{Signorini}(x_n, r_n) \\ + \\ \text{Coulomb}(\mathbf{x}_t, \mathbf{r}_t) \end{array} \right\} \Leftrightarrow \mathbf{r} = \text{Proj}_{K_\mu}(\mathbf{r}^*) \text{ avec } \mathbf{r}^* = \mathbf{r} - \rho \mathbf{v} \quad (2.40)$$

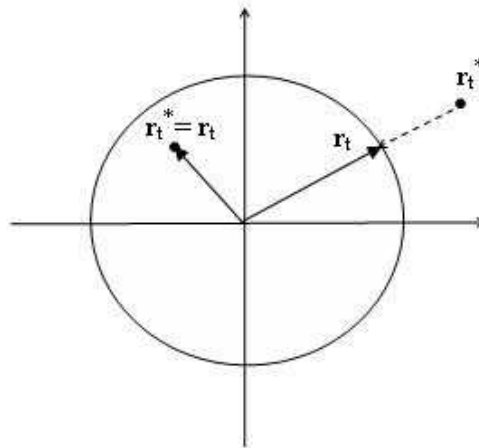


FIG. 2.9 – Projection sur le disque de Coulomb

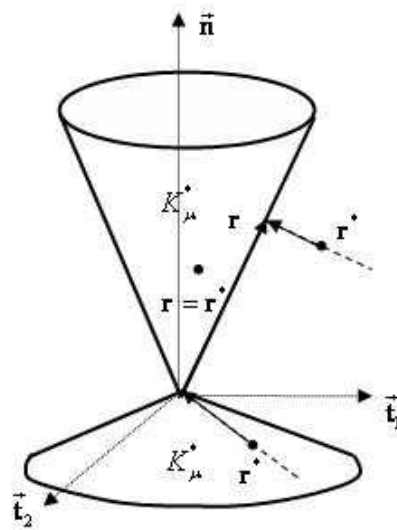


FIG. 2.10 – Projection sur le cône de Coulomb

2.4.4 Conclusion

Nous venons de voir qu'il existe plusieurs formulations des lois de contact frottant sec, ces dernières pouvant être formulées pour chaque couple de points de contact (P^α, Q^α) de la manière suivante :

$$\text{Signorini}(x_n^\alpha, r_n^\alpha) \Leftrightarrow x_n^\alpha \geq 0, r_n^\alpha \geq 0 \text{ et } x_n^\alpha \cdot r_n^\alpha = 0 \quad (2.41)$$

$$\text{Coulomb}(\mathbf{x}_t^\alpha, \mathbf{r}_t^\alpha) \Leftrightarrow \begin{cases} r_n^\alpha > 0, \\ \text{Si } \|\mathbf{x}_t^\alpha\| = 0 \text{ alors } \|\mathbf{r}_t^\alpha\| \leq \mu r_n^\alpha \\ \text{Sinon } \mathbf{r}_t^\alpha = -\mu r_n^\alpha \frac{\mathbf{x}_t^\alpha}{\|\mathbf{x}_t^\alpha\|} \end{cases} \quad (2.42)$$

Nous avons montré que seules les formulations pseudo-potentielle ou bi-potentielle sont du point de vue mathématique strictement équivalentes et conduisent à la construction d'opérateurs de projection particulièrement bien adaptés à la programmation.

Dans l'écriture «Pseudo-potentielle» des lois de contact détaillé précédemment et, présentée par [JT88], [AC91], [SL92] et [Kla92], chaque composante normale r_n^α et tangentielle \mathbf{r}_t^α de la force de réaction \mathbf{r}^α dérivent de deux pseudo-potentiels indépendants. La formulation «Pseudo-potentielle» des lois de contact frottant est alors basée sur deux opérateurs de projection distincts, le premier pour contrôler la composante normale de la force de contact et le second pour contrôler sa composante tangentielle :

$$\begin{aligned} \text{Signorini}(x_n^\alpha, r_n^\alpha) &\Leftrightarrow r_n^\alpha = \text{Proj}_{\mathbf{R}^+}(\underbrace{r_n^\alpha - \rho_n x_n^\alpha}_{r_n^{*\alpha}}) \\ &\Leftrightarrow Z_n = r_n^\alpha - \text{Proj}_{\mathbf{R}^+}(r_n^{*\alpha}) = 0 \end{aligned} \quad (2.43)$$

Si $r_n^{*\alpha} \geq 0$ alors $\text{Proj}_{\mathbf{R}^+}(r_n^{*\alpha}) = r_n^{*\alpha}$ et il y a contact sinon $\text{Proj}_{\mathbf{R}^+}(r_n^{*\alpha}) = 0$ et dans ce cas il n'y a pas contact.

$$\begin{aligned} \text{Coulomb}(\mathbf{x}_t^\alpha, \mathbf{r}_t^\alpha) &\Leftrightarrow r_n^\alpha > 0, \mathbf{r}_t^\alpha = \text{Proj}_{\mathbf{K}_\mu}(\underbrace{\mathbf{r}_t^\alpha - \rho_t \mathbf{x}_t^\alpha}_{\mathbf{r}_t^{*\alpha}}) \\ &\Leftrightarrow r_n^\alpha > 0, \mathbf{Z}_t = \mathbf{r}_t^\alpha - \text{Proj}_{\mathbf{K}_\mu}(\mathbf{r}_t^{*\alpha}) = 0 \end{aligned} \quad (2.44)$$

si $\mathbf{r}_t^{*\alpha} \leq \mu r_n^\alpha$ alors $\text{Proj}_{\mathbf{K}_\mu}(\mathbf{r}_t^{*\alpha}) = \mathbf{r}_t^{*\alpha}$ et il y a contact avec adhérence sinon

$\text{Proj}_{\mathbf{K}_\mu}(\mathbf{r}_t^{*\alpha}) = -\mu r_n^\alpha \frac{\mathbf{r}_t^{*\alpha}}{\|\mathbf{r}_t^{*\alpha}\|}$ et dans ce cas il y a contact avec glissement.

Dans la formulation «bi-potentielle» détaillée précédemment et introduite par G. De Saxcé et Z.Q. Feng [dF91], [dF98], la force de contact dérive d'une seule fonction appelée «bi-potentielle» qui couple le contact unilatéral et le frottement. L'avantage est qu'un seul opérateur de projection est nécessaire pour contrôler la force de contact permettant ainsi une meilleure robustesse algorithmique par rapport à la formulation «pseudo-potentielle». L'opérateur de projection peut être défini explicitement par :

$$\left. \begin{array}{l} \text{Signorini}(x_n^\alpha, r_n^\alpha) \\ + \\ \text{Coulomb}(\mathbf{x}_t^\alpha, \mathbf{r}_t^\alpha) \end{array} \right\} \Leftrightarrow \begin{cases} \mathbf{r}^\alpha = \text{Proj}_{\mathbf{K}_\mu}(\mathbf{r}^\alpha - \rho(\mathbf{x}^\alpha + \mu \|\mathbf{x}_t^\alpha\| \mathbf{n})) \\ \mathbf{r}^\alpha = \text{Proj}_{\mathbf{K}_\mu}(\mathbf{r}^{\alpha*}) \\ \mathbf{Z}^\alpha = \mathbf{r}^\alpha - \text{Proj}_{\mathbf{K}_\mu}(\mathbf{r}^{\alpha*}) = \mathbf{0} \end{cases} \quad (2.45)$$

avec $\mathbf{n} = (1,0,0)^T$, les trois statuts de contact étant définis par :

$$\left\{ \begin{array}{ll} \text{Proj}_{\mathbf{K}_\mu}({}^*\mathbf{r}^\alpha) = {}^*\mathbf{r}^\alpha & \text{si } \|{}^*\mathbf{r}_t^\alpha\| \leq \mu {}^*r_n^\alpha & \text{Contact avec adhérence} \\ \text{Proj}_{\mathbf{K}_\mu}({}^*\mathbf{r}^\alpha) = 0 & \text{si } \mu \|{}^*\mathbf{r}_t^\alpha\| \leq -{}^*r_n^\alpha & \text{Pas de contact} \\ \text{Proj}_{\mathbf{K}_\mu}({}^*\mathbf{r}^\alpha) = {}^*\mathbf{r}^\alpha - \left(\frac{\|{}^*\mathbf{r}_t^\alpha\| - \mu {}^*r_n^\alpha}{1 + \mu^2} \right) \left(\frac{{}^*\mathbf{r}_t^\alpha}{\|{}^*\mathbf{r}_t^\alpha\|} - \mu \mathbf{n} \right) & & \text{Contact avec glissement} \end{array} \right.$$

Des tests numériques réalisés par Feng et DeSaxcé [dF98] ont montré que cette méthode permet de réduire les temps de calcul, par rapport à la méthode Pseudo-potentielle, de moitié pour les problèmes 2D et du tiers pour les problèmes 3D dans le calcul de la force de contact avec frottement. Comme nous le verrons plus tard dans le chapitre 3, cette formulation permet d'obtenir des résultats stables et précis.

Quelque soit la formulation choisie, bi-potentielle ou pseudo-potentielle, nous aurons pour chaque couple de points de contact (P^α, Q^α) le système $\mathbf{Z}^\alpha = \mathbf{0}$ à résoudre où les inconnues sont \mathbf{r}^α et \mathbf{x}^α . Il y a donc deux fois plus d'inconnues que d'équations, les équations supplémentaires seront obtenues par projection des équations d'équilibre dans les repères locaux aux points de contact. Nous appellerons méthode de flexibilité ce principe de projection qui permet de séparer le calcul des forces de contact de la résolution globale en déplacement des équations d'équilibre.

2.5 Mise en place des équations d'équilibre avec contacts multiples

Nous traitons uniquement la formulation d'un problème de deux objets Ω_1, Ω_2 en contact par soucis de simplification sans que cela induise de perte de généralité par rapport à plusieurs objets en contact. Les notations sont celles déjà définies dans le paragraphe 2.3.1.

Nous avons vu dans le paragraphe 2.2.2 que les équations d'équilibre pouvaient se mettre sous la forme générale :

$$\left\{ \begin{array}{l} \mathbf{K}_T \delta \mathbf{X} + \mathbf{R}(t_{n+1})^{i+1} = \mathbf{F}_{\text{int}}(\mathbf{X}_{n+1}^i) + \mathbf{F}_{\text{ext}}(t_{n+1}) \\ \mathbf{X}_{n+1}^{i+1} = \mathbf{X}_{n+1}^i + \delta \mathbf{X} \\ \mathbf{X}_{n+1}^0 = \mathbf{X}_n \end{array} \right. \quad (2.46)$$

Dans le cas de deux solides, on a : $\mathbf{K}_T = \begin{bmatrix} \mathbf{K}_{T1} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{T2} \end{bmatrix}$, $\mathbf{X} = \left\{ \begin{array}{l} \mathbf{X}_1 \\ \mathbf{X}_2 \end{array} \right\}$

$\mathbf{X}_1, \mathbf{X}_2$ représentent respectivement les positions des noeuds des solides Ω_1, Ω_2 . On s'intéresse ici à la résolution d'un seul incrément $\Delta\mathbf{X}$. Afin d'alléger les notations, on omettra toute référence à l'instant t_{n+1} (cas d'un problème de dynamique) ou au numéro de pas de chargement (cas quasi-statique). Les équations d'équilibre prennent donc la forme simplifiée :

$$\begin{cases} \mathbf{K}_T \delta\mathbf{X} + \mathbf{R}_{i+1} = \mathbf{F}_{\text{int}}(\mathbf{X}_i) + \mathbf{F}_{\text{ext}} \\ \mathbf{X}_{i+1} = \mathbf{X}_i + \delta\mathbf{X} \\ \mathbf{X}_0 \text{ est une donnée du problème} \end{cases} \quad (2.47)$$

D'une manière générale, on peut déterminer les positions de P^α (resp. Q^α) à partir d'une matrice d'interpolation \mathbf{B}_1^α (resp. \mathbf{B}_2^α) :

$$\mathbf{X}(P^\alpha) = \mathbf{B}_1^\alpha \mathbf{X}_1, \quad \mathbf{X}(Q^\alpha) = \mathbf{B}_2^\alpha \mathbf{X}_2 \quad (2.48)$$

Soit \mathbf{X}^α le vecteur position relatif de P^α par rapport à Q^α par :

$$\mathbf{X}^\alpha = \mathbf{X}(P^\alpha) - \mathbf{X}(Q^\alpha) \quad (2.49)$$

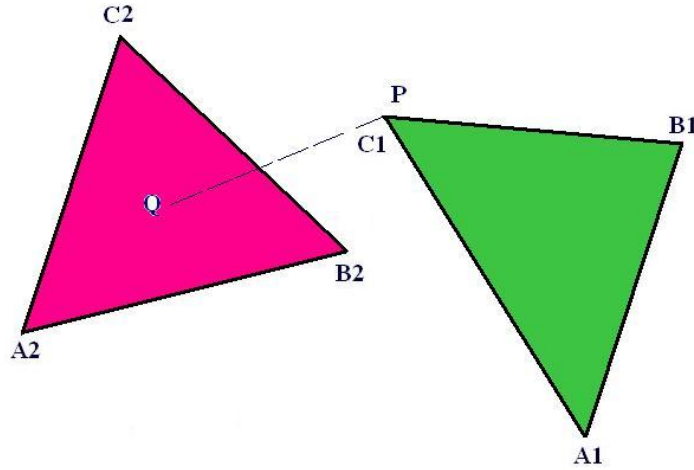


FIG. 2.11 – Chaque contact relie deux points que l'on interpole sur les triangles de la surface des objets déformables.

On en déduit d'après les notations définies dans le paragraphe 2.3.1 que :

$$\begin{cases} x_n = \mathbf{n}^T \cdot \mathbf{X}^\alpha \\ x_{t_1} = \mathbf{t}_1^T \cdot \mathbf{X}^\alpha \\ x_{t_2} = \mathbf{t}_2^T \cdot \mathbf{X}^\alpha \end{cases} \quad (2.50)$$

Finalement en combinant 2.50 et 2.48, on obtient :

$$\mathbf{x}^\alpha = \mathbf{H}_\alpha \mathbf{X} \quad (2.51)$$

avec

$$\mathbf{H}_\alpha = \begin{bmatrix} \mathbf{t}_1^T \mathbf{B}^\alpha \\ \mathbf{t}_2^T \mathbf{B}^\alpha \\ \mathbf{n}^T \mathbf{B}^\alpha \end{bmatrix}, \mathbf{B}_\alpha = [-\mathbf{B}_\alpha^1 | \mathbf{B}_\alpha^2], \mathbf{X} = \begin{Bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{Bmatrix} \quad (2.52)$$

La forme incrémentale de 2.51 est :

$$\mathbf{x}_{i+1}^\alpha = \mathbf{H}_\alpha \delta \mathbf{X} + \mathbf{x}_i^\alpha \quad (2.53)$$

On fait l'hypothèse que pendant toute la longueur de l'incrément $\Delta \mathbf{X}$ la matrice \mathbf{H}_α est constante. D'autre part on remarque $\mathbf{x}_0^\alpha = g^\alpha \mathbf{n}$ où g^α représente le gap entre P^α et Q^α donné par le détecteur de collisions. Afin de prendre en compte N_c couples de points de contact simultanés, on considère les notations suivantes :

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_\alpha \end{bmatrix}, \mathbf{x} = \begin{Bmatrix} \mathbf{x}^1 \\ \vdots \\ \mathbf{x}^{N_c} \end{Bmatrix}, \mathbf{r} = \begin{Bmatrix} \mathbf{r}^1 \\ \vdots \\ \mathbf{r}^{N_c} \end{Bmatrix} \quad (2.54)$$

Ce qui donne en combinant 2.53 et 2.54 :

$$\mathbf{x}_{i+1} = \mathbf{H} \delta \mathbf{X} + \mathbf{x}_i \quad (2.55)$$

D'autre part en utilisant le calcul du travail virtuel de la force de contact en chaque point de contact, on obtient la transformation de passage entre les coordonnées locales et globales :

$$\mathbf{r}^{\alpha T} \delta \mathbf{x}^\alpha = \mathbf{R}^{\alpha T} \delta \mathbf{X}^\alpha \quad (2.56)$$

D'après 2.51, on obtient :

$$\mathbf{R}^\alpha = \mathbf{H}_\alpha^T \mathbf{r}^\alpha \quad (2.57)$$

Ce qui nous donne: $\mathbf{R} = \mathbf{H}^T \mathbf{r}$

2.6 Principe de la méthode de flexibilité

Le système que l'on doit résoudre à chaque itération i de la boucle Newton Raphson est le suivant :

$$\begin{cases} \mathbf{K}_T \delta \mathbf{X} + \mathbf{H}^T \mathbf{r}_{i+1} = \mathbf{F}_{\text{int}}(\mathbf{X}_i) + \mathbf{F}_{\text{ext}} \\ \mathbf{x}_{i+1} = \mathbf{H} \delta \mathbf{X} + \mathbf{x}_i \\ \mathbf{Z} = \mathbf{0} \end{cases} \quad (2.58)$$

avec $\mathbf{Z} = \begin{bmatrix} \mathbf{Z}^1 \\ \vdots \\ \mathbf{Z}^{N_c} \end{bmatrix}$ le système d'équations introduit par la formulation pseudo-potentielle ou bi-potentielle pour les N_c points de contact.

Après chaque résolution les positions sont réactualisées ($\mathbf{X}_{i+1} = \mathbf{X}_i + \delta\mathbf{X}$) jusqu'à obtenir la convergence définie par le critère suivant :

$$|\delta\mathbf{X}| = \sum_{j=1, ddl} |\delta X_j| \leq \varepsilon_{Newton-Raphson} \quad (2.59)$$

La procédure itérative de Newton-Raphson permet de surmonter la complexité des calculs et d'améliorer la stabilité numérique pour ce type de problème non linéaire [BLM00], [SH98].

Il existe trois types de méthode pour résoudre le système 2.58. La première est de calculer simultanément les deux vecteurs inconnus : les forces de contact \mathbf{r} et les déplacements nodaux $\delta\mathbf{X}$ en respectant les contraintes associées aux points de contact par des techniques mathématiques d'optimisation. La deuxième technique est basée sur une formulation explicite des forces de contact en fonction des déplacements nodaux, il peut s'agir d'une fonction de pénalité par exemple. Ainsi, on réduit les inconnus du système d'équilibre 2.58 uniquement aux déplacements nodaux $\delta\mathbf{X}$. La dernière méthode consiste à calculer les déplacements nodaux $\delta\mathbf{X}$ après avoir calculer en premier les forces de contact \mathbf{r} par une formulation implicite. Cette dernière se nomme la « méthode de flexibilité », développée par Francavilla et Zienkiewicz [FZ75] pour des problèmes de contact sans frottement puis par Sachdeva et al. [SR81], [SRN81]. Cette approche a pour but de séparer les non-linéarités liées aux grands déplacements et aux grandes déformations des non linéarités liées aux forces de contact.

Toujours par soucis de simplification, nous omettons par la suite les indices i liés aux itérations de Newton Raphson, ce qui nous donne le système à résoudre sous sa forme simplifiée :

$$\begin{cases} \mathbf{K}_T \delta\mathbf{X} + \mathbf{H}^T \mathbf{r} = \mathbf{F}_{\text{int}}(\mathbf{X}) + \mathbf{F}_{\text{ext}} \\ \mathbf{x} = \mathbf{H} \delta\mathbf{X} + \mathbf{g} \\ \mathbf{Z} = \mathbf{0} \end{cases} \quad (2.60)$$

avec $\mathbf{g} = \mathbf{x}_i$.

Le principe de la méthode consiste à éliminer $\delta\mathbf{X}$ afin de conduire à la résolution d'un système réduit où les inconnues sont les déplacements contraints \mathbf{x} et les forces de contacts \mathbf{r} . Cela revient à projeter les équations d'équilibre dans les repères locaux liés aux points de contact.

$$\mathbf{x} = \mathbf{W}\mathbf{r} + \tilde{\mathbf{x}} \quad (2.61)$$

avec

$$\mathbf{W} = \mathbf{H}\mathbf{K}_T^{-1}\mathbf{H}^T \quad \text{and} \quad \tilde{\mathbf{x}} = \mathbf{H}\mathbf{K}_T^{-1}(\mathbf{F}_{\text{int}} + \mathbf{F}_{\text{ext}}) + \mathbf{g} \quad (2.62)$$

La matrice \mathbf{W} représente la matrice de flexibilité, appelée également matrice de Delassus, c'est une matrice carrée de taille (N_c, N_c) . $\tilde{\mathbf{x}}$ représente le vecteur des déplacements libres aux points de contact. Résoudre séparément les forces de contact, revient à résoudre le système suivant :

$$\text{Trouver } \boldsymbol{\chi} \text{ tel que } \mathbf{f}(\boldsymbol{\chi}) = \mathbf{0} \quad (2.63)$$

Avec

$$\boldsymbol{\chi} = \begin{Bmatrix} \chi_1 \\ \vdots \\ \chi_{N_c} \end{Bmatrix} \quad \text{and} \quad \boldsymbol{\chi}_\alpha = \begin{Bmatrix} \mathbf{r}^\alpha \\ \mathbf{x}^\alpha \end{Bmatrix} \quad (2.64)$$

$$\mathbf{f}(\boldsymbol{\chi}) = \begin{Bmatrix} \mathbf{f}_1(\boldsymbol{\chi}) \\ \vdots \\ \mathbf{f}_{N_c}(\boldsymbol{\chi}) \end{Bmatrix} \quad \text{and} \quad \mathbf{f}_\alpha(\boldsymbol{\chi}) = \begin{Bmatrix} \mathbf{x}^\alpha - \sum_{\beta=1}^{N_c} \mathbf{W}_{\alpha\beta} \mathbf{r}^\beta - \tilde{\mathbf{x}}^\alpha \\ \mathbf{Z}^\alpha \end{Bmatrix} \quad (2.65)$$

$$\tilde{\mathbf{x}}^\alpha = \mathbf{H}_\alpha \mathbf{K}_T^{-1} (\mathbf{F}_{\text{int}} + \mathbf{F}_{\text{ext}}) + \mathbf{g}^\alpha \quad (2.66)$$

$$\mathbf{W}_{\alpha\beta} = \mathbf{H}_\alpha \mathbf{K}_T^{-1} \mathbf{H}_\beta^T \quad (2.67)$$

Comme nous le verrons par la suite, le système 2.63 est résolu de manière itérative en parcourant tous les points de contact. Une fois celui-ci résolu, alors les forces de contact sont connues et peuvent être ajoutées aux forces externes du système 2.46 à chaque itération de la boucle Newton Raphson. La méthode de flexibilité est synthétisée par le schéma de la figure 2.12. On remarque que le détecteur de collisions est lancé au début de chaque pas de chargement. Une autre option, représentée en pointillé sur le schéma, est de le lancer à chaque boucle d'itération du calcul des forces de contact. Cette option est beaucoup plus coûteuse numériquement mais cependant beaucoup plus correcte dans le cas de grands glissements.

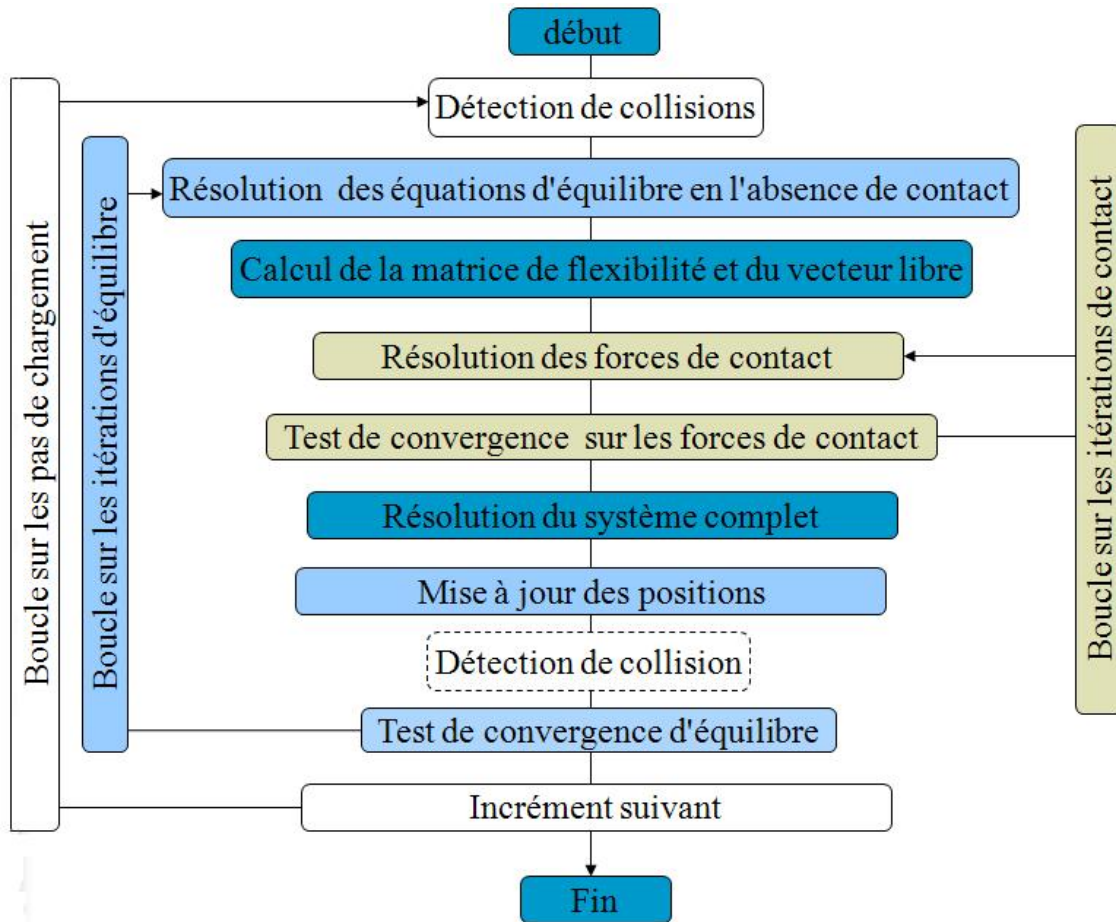


FIG. 2.12 – L'algorithme générale de la méthode de Flexibilité.

2.7 Résolution par équilibres successifs des forces de contact: algorithme de Gauss-Seidel

Dans le domaine de la mécanique granulaire [CJ01], un algorithme de résolution de type Gauss-Seidel à été proposé pour résoudre de manière itérative les forces de contact. Ce principe de résolution à été utilisé également dans la thèse de Duriez [Dur04] dans un contexte «temps réel» avec retour haptique. Dans les deux cas la formulation choisie des lois de contact était celle du pseudo-potentielle.

Le principe de cet algorithme est de décomposer la résolution globale des $6 \times N_c$ équations du système 2.63 en N_c résolutions successives de systèmes à 6 équations définis en chaque point de contact α de la manière suivante :

$$\mathbf{f}_\alpha(\boldsymbol{\chi}) = \left\{ \begin{array}{c} \mathbf{x}^\alpha - \mathbf{W}_{\alpha\alpha}\mathbf{r}^\alpha - \mathbf{x}^{\alpha\beta} \\ \mathbf{Z}_\alpha \end{array} \right\} = \mathbf{0} \quad (2.68)$$

avec

$$\mathbf{x}^{\alpha\beta} = \sum_{\beta=1, \beta \neq \alpha}^{N_c} \mathbf{W}_{\alpha\beta}\mathbf{r}^\beta + \tilde{\mathbf{x}}^\alpha \quad (2.69)$$

Où $\mathbf{x}^{\alpha\beta}$ représente la contribution des déplacements relatifs au point de contact α due au vecteur \mathbf{g} , aux forces externes et aux forces de contact des $N_c - 1$ autres points de contact β . Cette contribution est «figée» pendant la résolution locale. Une série de N_c résolutions locales correspond à une iteration k de l'algorithme. Le processus iteratif est défini de la manière suivante :

$$\mathbf{f}_\alpha(\boldsymbol{\chi})_{k+1} = \left\{ \begin{array}{c} \mathbf{x}_{k+1}^\alpha - \mathbf{W}_{\alpha\alpha}\mathbf{r}_{k+1}^\alpha - \mathbf{x}^{\alpha\beta} \\ \mathbf{Z}^\alpha \end{array} \right\} = \mathbf{0} \quad (2.70)$$

avec

$$\mathbf{x}^{\alpha\beta} = \sum_{\beta=\alpha+1, N_c} \mathbf{W}_{\alpha\beta}\mathbf{r}_k^\beta + \sum_{\beta=1, \alpha-1} \mathbf{W}_{\alpha\beta}\mathbf{r}_{k+1}^\beta + \tilde{\mathbf{x}}^\alpha \quad (2.71)$$

La résolution locale est obtenue lorsque le critère suivant est vérifié :

$$\frac{\|\mathbf{r}^{(k+1)} - \mathbf{r}^{(k)}\|}{\|\mathbf{r}^{(k+1)}\|} \leq \varepsilon_{Gauss-Seidel} \quad (2.72)$$

Où $\varepsilon_{Gauss-Seidel}$ est une tolérance numérique définie par l'utilisateur. L'algorithme démarre avec des forces de contact nulles ($\mathbf{r}^{(0)} = \mathbf{0}$).

2.8 Résolution locale des forces de contact

Il s'agit maintenant de proposer une méthode de résolution du système 2.70 à chaque itération de l'algorithme de Gauss-Seidel. Pour cela, nous proposons la méthode de Newton utilisée par Jean [CJ01], Duriez [Dur04], Joli [JF08], et la méthode Uzawa qui utilisée par Feng et al. [Fen95], [FPL03] ...

Toujours afin d'alléger les écritures, on omettra l'écriture de l'indice de boucle k , on sous entend ainsi que la résolution est effectuée dans une itération de Gauss-Seidel, ce qui donne le système à résoudre suivant :

$$\mathbf{f}_\alpha(\boldsymbol{\chi}) = \left\{ \begin{array}{c} \mathbf{x}^\alpha - \mathbf{W}_{\alpha\alpha}\mathbf{r}^\alpha - \mathbf{x}^{\alpha\beta} \\ \mathbf{Z}^\alpha \end{array} \right\} = \mathbf{0} \quad (2.73)$$

2.8.1 Méthode de Newton

L'algorithme de Newton appliqué à la résolution des équations implicites 2.73 donne la procédure itérative suivante :

- Etape 1 : initialiser

$$\boldsymbol{\chi}_\alpha^0 = \left\{ \begin{array}{l} \mathbf{r}_0^\alpha = \mathbf{0} \\ \mathbf{x}_0^\alpha = \mathbf{x}_k^{\alpha\beta} \end{array} \right\}, j = 0$$

- Etape 2 : augmenter les forces de contact :

$$\mathbf{r}_j^\alpha = \mathbf{r}_j^\alpha - \rho \mathbf{x}_j^{\alpha*} \text{ avec } \mathbf{x}_j^{\alpha*} = \mathbf{x}_j^\alpha + \mu \|(\mathbf{x}_t^\alpha)_j\| \mathbf{n}$$

- Etape 3: Résoudre

$$\left[\frac{\partial \mathbf{f}_\alpha(\boldsymbol{\chi}^j)}{\partial \boldsymbol{\chi}_\alpha} \right] \Delta \boldsymbol{\chi}_\alpha = -\mathbf{f}_\alpha(\boldsymbol{\chi}_j)$$

- Etape 4: Réactualiser

$$\boldsymbol{\chi}_\alpha^{j+1} = \boldsymbol{\chi}_\alpha^j + \Delta \boldsymbol{\chi}_\alpha$$

- Etape 5: Vérifier la convergence:

$$\begin{array}{l} \text{si } \|\Delta \boldsymbol{\chi}_\alpha\| / \|\boldsymbol{\chi}_\alpha^j\| > \varepsilon \text{ alors } j = j + 1 \text{ retour à Etape 2} \\ \text{sinon } \boldsymbol{\chi}_\alpha^{k+1} = \boldsymbol{\chi}_\alpha^{j+1} \end{array}$$

Où $\boldsymbol{\chi}_\alpha^{k+1}$ représente les variables au point de contact α à l'itération $k + 1$ de la résolution globale de l'algorithme de Gauss Seidel.

$\left[\frac{\partial \mathbf{f}_\alpha(\boldsymbol{\chi}^j)}{\partial \boldsymbol{\chi}_\alpha} \right]$ représente la matrice tangente des équations locales au point de contact α .

Cette matrice est de dimension 6×6 et a pour forme générale :

$$\left[\frac{\partial \mathbf{f}_\alpha(\boldsymbol{\chi}^j)}{\partial \boldsymbol{\chi}_\alpha} \right] = \begin{bmatrix} -\mathbf{H}_{\alpha\alpha} & \mathbf{Id}_{3 \times 3} \\ \mathbf{A}_\alpha & \mathbf{B}_\alpha \end{bmatrix}$$

Avec

$$\mathbf{A}_\alpha = \left[\frac{\partial \mathbf{Z}_\alpha}{\partial r_{t_1}^\alpha} \mid \frac{\partial \mathbf{Z}_\alpha}{\partial r_{t_2}^\alpha} \mid \frac{\partial \mathbf{Z}_\alpha}{\partial r_n^\alpha} \right], \mathbf{B}_\alpha = \left[\frac{\partial \mathbf{Z}_\alpha}{\partial x_{t_1}^\alpha} \mid \frac{\partial \mathbf{Z}_\alpha}{\partial x_{t_2}^\alpha} \mid \frac{\partial \mathbf{Z}_\alpha}{\partial x_n^\alpha} \right]$$

Il y a deux formes explicites des matrices A et B selon le statut de contact glissant ou adhérent. Dans le cas d'une formulation bipotentielle, les expressions de ces matrices ont été déterminées dans [JF08]. Dans le cas d'une formulation pseudo-potentielle, on peut trouver les expressions dans [CJ01], Curnier [CHK95], Duriez [Dur04]. Lorsqu'il n'y a pas de contact, on ne calcule pas de matrice tangente, la solution est directement donnée par :

$$\boldsymbol{\chi}_\alpha^{k+1} = \left\{ \begin{array}{l} \mathbf{0} \\ \mathbf{x}^{\alpha\beta(k)} \end{array} \right\}$$

Remarque: La solution à chaque itération peut être obtenue à partir de la technique de condensation suivante qui optimise les calculs :

$$\boldsymbol{\chi}_\alpha^{j+1} = \left\{ \begin{array}{l} \Delta \mathbf{r}^\alpha = [\mathbf{A}_\alpha + \mathbf{B}_\alpha \mathbf{H}_{\alpha\alpha}]^{-1} (-\mathbf{Z}^{\alpha(j)} + \mathbf{B}_\alpha \mathbf{x}^{\alpha(j)}) \\ \Delta \mathbf{x}^\alpha = -\mathbf{x}^{\alpha(j)} + \mathbf{H}_{\alpha\alpha} \Delta \mathbf{r}^\alpha \end{array} \right\}$$

2.8.2 Méthode d'Uzawa

La résolution des équations implicites 2.73 peut être effectuée au moyen de l'algorithme d'Uzawa qui consiste en un schéma itératif prédiction/correcteur défini de la façon suivante :

- Etape 1 : initialisation

$$\boldsymbol{\chi}_\alpha^0 = \left\{ \begin{array}{l} \mathbf{r}_0^\alpha = \mathbf{0} \\ \mathbf{x}_0^\alpha = \mathbf{x}_k^{\alpha\beta} \end{array} \right\}, j = 0$$

- Etape 2 : phase de prédiction

$$\mathbf{r}_{j+1}^\alpha = \mathbf{r}_j^\alpha + \Delta \mathbf{r}^\alpha$$

avec

$$\left\{ \begin{array}{l} \Delta \mathbf{r}^\alpha = -\rho(\mathbf{x}_j^\alpha + \mu \|(\mathbf{x}_t^\alpha)_j\| \mathbf{n}) \text{ (formulation bi-potentielle)} \\ \Delta \mathbf{r}^\alpha = - \left\{ \begin{array}{l} \rho_n(x_n^\alpha)_j \\ \rho_t(\mathbf{x}_t^\alpha)_j \end{array} \right\} \text{ (formulation pseudo-potentielle)} \end{array} \right.$$

- Etape 3 : phase de correction

$$\mathbf{r}_{j+1}^\alpha = \text{Proj}_{\mathbf{K}_\mu}(\mathbf{r}_{j+1}^\alpha) \text{ (formulation bipotentielle)}$$

$$\mathbf{r}_{j+1}^\alpha = \left\{ \begin{array}{l} (r_n^\alpha)_{j+1} = \text{Proj}_{\mathbf{R}^+}((r_n^\alpha)_{j+1}) \\ (\mathbf{r}_t^\alpha)_{j+1} = \text{Proj}_{\mathbf{D}_\mu}((\mathbf{r}_t^\alpha)_{j+1}) \end{array} \right\} \text{ (formulation pseudo-potentielle)}$$

- Etape 5: Vérifier la convergence:

$$\text{si } \|\Delta \mathbf{r}^\alpha\| / \|\mathbf{r}_{j+1}^\alpha\| > \epsilon \text{ alors } \mathbf{x}_{j+1}^\alpha = \mathbf{W}_{\alpha\alpha} \mathbf{r}_{j+1}^\alpha + \mathbf{x}^{\alpha\beta}, j = j + 1 \text{ retour à Etape 2}$$

$$\text{sinon } \mathbf{r}_{k+1}^\alpha = \mathbf{r}_{j+1}^\alpha$$

Uzawa Locale

Où \mathbf{r}_{k+1}^α représente la force au point de contact α à l'itération $k + 1$ de la résolution globale de l'algorithme de Gauss Seidel.

Nous appellerons cette méthode, la méthode Uzawa locale, car elle assure une convergence locale au point de contact au même titre que la méthode de Newton. Feng et al. [FHdM06] utilisent une approche légèrement différent en ne cherchant pas la convergence locale mais uniquement la convergence globale sur l'ensemble des points de contact. C'est pourquoi ils n'utilisent qu'une seule prédiction/correction par point de contact et par itération de Gauss-Seidel. Nous appellerons cette méthode la méthode Uzawa globale, car

la convergence est obtenue uniquement par l'algorithme Gauss-Seidel. Cet algorithme peut être défini de la manière suivante :

- Etape 1: phase de prédiction

$$\mathbf{r}_{k+1}^\alpha = \mathbf{r}_k^\alpha + \Delta \mathbf{r}^\alpha$$

Avec

$$\begin{cases} \Delta \mathbf{r}^\alpha = -\rho(\mathbf{x}_k^\alpha + \mu \|(\mathbf{x}_t^\alpha)_k\| \mathbf{n}) & \text{(formulation bi-potentielle)} \\ \Delta \mathbf{r}^\alpha = - \begin{cases} \rho_n(x_n^\alpha)_k \\ \rho_t(\mathbf{x}_t^\alpha)_k \end{cases} & \text{(formulation pseudo-potentielle)} \end{cases}$$

- Etape 2: phase de correction

$$\mathbf{r}_{k+1}^\alpha = \text{Proj}_{\mathbf{K}_\mu}(\mathbf{r}_{k+1}^\alpha) \text{ (formulation bi-potentielle)}$$

$$\mathbf{r}_{k+1}^\alpha =$$

$$\left\{ \begin{array}{l} (r_n^\alpha)_{k+1} = \text{Proj}_{\mathbf{R}^+}((r_n^\alpha)_{k+1}) \\ (\mathbf{r}_t^\alpha)_{k+1} = \text{Proj}_{\mathbf{D}_\mu}((\mathbf{r}_t^\alpha)_{k+1}) \end{array} \right\} \text{ (formulation pseudo-potentielle)}$$

Où k et $k + 1$ représentent les indices des iterations Gauss Seidel. Le déplacement contraint est réactualisé par :

$$\mathbf{x}_{k+1}^\alpha = \mathbf{W}_{\alpha\alpha} \mathbf{r}_{k+1}^\alpha + \mathbf{x}^{\alpha\beta} \quad (2.74)$$

2.9 Conclusion

Dans ce chapitre, nous avons présenté différentes formulations pour modéliser les lois de contact frottant entre objets déformables. Les formulations pseudo-potentielle et bi-potentielle sont du point de vue mathématique équivalentes dans le sens où elles traduisent toutes les deux exactement les conditions de Signorini et les conditions de la loi de Coulomb. Elles conduisent toutes les deux à la construction d'opérateurs de projection parfaitement adaptés à la programmation. La formulation bipotentielle considère un unique opérateur de projection qui couple l'ensemble des conditions. Le contrôle de la solution est ainsi meilleur que dans le cas de la formulation pseudo-potentielle qui nécessite deux opérateurs de projection, un pour les conditions de Signorini et un autre pour les conditions de la loi de Coulomb. Bien que par la suite dans le chapitre 3 et 4, nos résultats numériques sont obtenus en quasi-statique et en élasticité linéaire, nous nous sommes attachés à présenter la mise en oeuvre de ces deux formulations dans un cadre général (dynamique, grandes déformations, schémas numériques : explicite, implicite). Cela a été possible grâce au principe de la méthode de flexibilité qui sépare les non linéarités liés au contact des autres types de non linéarités. En effet, la méthode de flexibilité permet d'isoler dans la résolution globale des déplacements un solveur de forces de contact. Nous avons ainsi présenté

trois types de solveur: Newton, Uzawa local et Uzawa global. Ces trois solveurs résolvent de manière itérative l'ensemble des forces de contact selon le principe de Gauss-Seidel. Ils se distinguent par leur résolution locale en chaque point de contact. En fait, on pourrait considérer qu'il y a six solveurs possibles selon le choix de la formulation des lois de contact (pseudo potentielle, bi-potentielle). Du fait de la complexité (et de son coût numérique) du solveur de Newton dans le cas de la formulation bipotentielle, nous nous limiterons par la suite aux solveurs Uzawa.

Résolution du contact frottant entre objets déformables: Expérimentations et Comparaisons

3.1 Introduction

Dans le chapitre 2, nous avons élaboré plusieurs stratégies possibles pour résoudre des problèmes de contact frottant entre objets déformables. Une stratégie de résolution se distingue par :

- le choix de la formulation des lois de contact,
- le choix de la formulation des équations d'équilibre,
- le choix de la méthode numérique pour résoudre les forces de contact et les déplacements contraints.

Nous avons retenus les stratégies où les lois de contact sont formulées à partir d'opérateurs de projection qui vérifient de manière exacte les conditions de Signorini et la loi de Coulomb. L'utilisation d'opérateurs de projection permet une implémentation simple des lois de contact, sans ajout d'inconnues supplémentaires (multiplicateurs de Lagrange). Deux choix sont alors possibles :

- La formulation « pseudo-potentielle » de Jean et Moreau ([JM92], [Jea95], [Jea96]) qui utilise deux opérateurs de projection indépendants associés respectivement aux conditions de Signorini et la loi de Coulomb.
- La formulation appelée bi-potentielle par Desaxé et Feng ([dF91], [dF98]) qui utilise un unique opérateur de projection associé simultanément aux conditions de Signorini et la loi de Coulomb.

D'autre part, dans nos stratégies, nous optons pour une formulation incrémentale des équations d'équilibre qui s'adapte aussi bien à des problèmes de statique que de dynamique. Nous utiliserons également la méthode de flexibilité qui consiste à projeter les équations d'équilibre dans des repères locaux liés aux points de contact et qui permet de ne pas modifier la matrice tangente de résolution contrairement à la méthode de pénalité

ou à la méthode des multiplicateurs de Lagrange. Enfin nous utilisons une méthode numérique basée sur les algorithmes itératifs «Gauss Seidel» et «Uzawa» pour résoudre les forces de contact et les déplacements contraints. Nous avons défini dans le chapitre 2, deux algorithmes «Uzawa», le premier appelé «Uzawa Global » qui utilise une seule itération prédiction/correction et le deuxième appelé «Uzawa Local» qui en utilise plusieurs. Un autre choix possible est d'utiliser l'algorithme itératif de «Newton» à la place de l'algorithme «Uzawa» qui à l'avantage d'une vitesse de convergence plus rapide (converge avec moins d'itération). Cependant les travaux de Feng et Joli [JF08] ont montré, dans le cas du Bi-potentiel, que la résolution «Newton» est plus lente que la résolution Uzawa car elle nécessite la construction et l'inversion à chaque itération d'une matrice jacobienne multi-forme fonction du statut du contact. Nous considérons donc dans ce chapitre, comme le montre la figure 3.1, uniquement des stratégies de calculs basées sur :

- une formulation incrémentale des équations d'équilibre combinée à la méthode de flexibilité
- une formulation bi-potentielle ou pseudo potentielle des lois de contact
- des techniques de résolution basées sur «Gauss-Seidel» et «Uzawa»

Dans nos différentes stratégies, les algorithmes de résolution de forces de contact seront évalués en fonction de différentes formulations de calcul des raideurs locales fictives et des différentes propriétés mécaniques (modules de Young et coefficient de frottement). Dans cette étude, nous avons opté pour trois types de matériaux différents, l'aluminium pour représenter un objet rigide, le tissu mou pour représenter un organe et le caoutchouc.

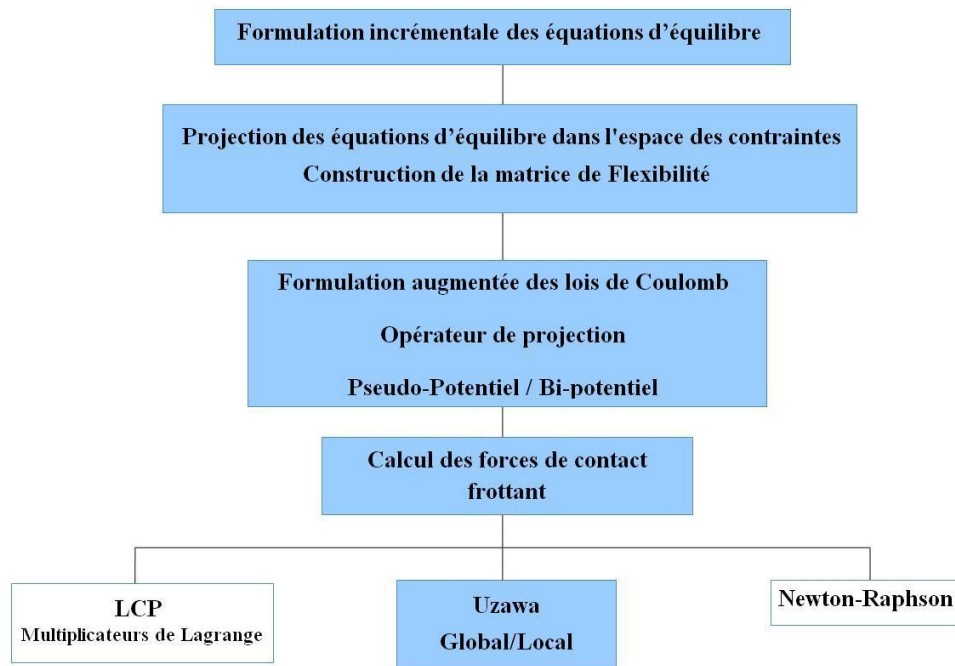


FIG. 3.1 – Stratégies utilisées pour le calcul des forces de contact avec frottement.

3.2 Présentation du banc de test

Nous allons dans cette section, tester et comparer les différentes approches de résolution des lois de contact frottant sec entre des objets élastiques en interaction dans le cadre d'une résolution par la méthode de flexibilité des équations d'équilibre du système. La simulation est composée de deux objets en élasticité linéaire en éléments finis linéaires cubiques ou tétraédriques soumis à des hypothèses d'équilibre quasi-statiques. Ainsi les effets inertiels ne sont pas considérés. La première structure est mobile car elle est soumise à des contraintes en déplacement imposés par l'opérateur humain, tandis que la seconde structure est fixe. Les déformations obtenues peuvent dépasser les 10% de la longueur de l'objet. La matrice de raideur de l'objet étant constante, elle est calculée et inversée une seule fois hors ligne afin d'obtenir une simulation temps réel. Nous rappelons que le module de Young (exprimé en Méga Pascal $Mpa = \frac{Newton}{mm^2}$) et le coefficient de Poisson (sans unité) sont deux paramètres du modèle élastique linéaire qui caractérisent respectivement la rigidité et la compressibilité du matériau. Un matériau parfaitement incompressible admet un coefficient de Poisson de valeur 0.5. Les expérimentations qui suivent sont réalisées en faisant varier plusieurs paramètres : module de Young, coefficient de frottement, nombre de points en contact, raideur locale fictive ρ et tolérance numérique (erreur Gauss-Seidel, erreur Newton-Raphson, erreur Uzawa).

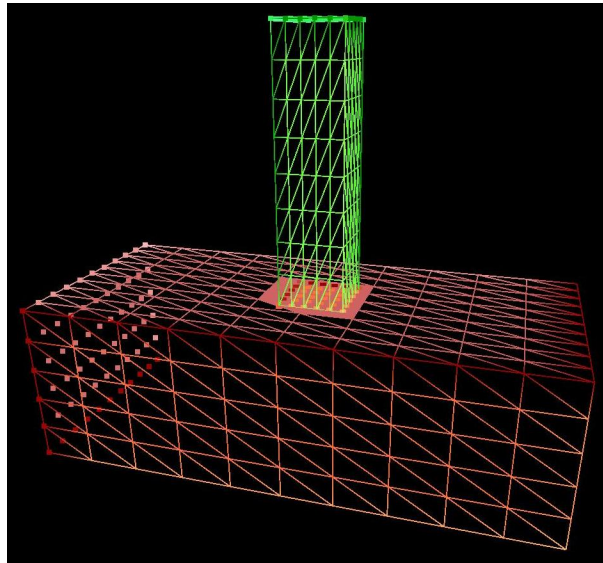


FIG. 3.2 – Test du solveur de contact frottant sec entre deux objets déformables, l'objet en rouge est fixe, l'objet en vert est mobile et soumis à des contraintes en déplacements imposées.

La figure 3.2 présente deux parallélépipèdes en interaction, les noeuds en surbrillance sur la structure rouge sont bloqués représentant ainsi un encastrement. Cet objet est composé de 726 noeuds, 500 éléments cubiques et admet les dimensions $L = 8$ cm (longueur), $H = 2.5$ cm (hauteur) et $l = 3.5$ cm (largeur). Le volume de chaque élément représente 0.2% du volume total. La déformation de l'objet vert est déterminée autour d'une configuration de référence réactualisée suivant un mouvement de corps rigide imposé (approche corotationnelle globale). Ce solide est composé de 324 noeuds, 200 éléments cubiques et admet les dimensions $L = 1$ cm, $H = 4$ cm et $l = 1$ cm. Le volume de chaque élément représente 0.5% du volume total. Quand les deux objets entrent en contact, la procédure de résolution du contact frottant est lancée.

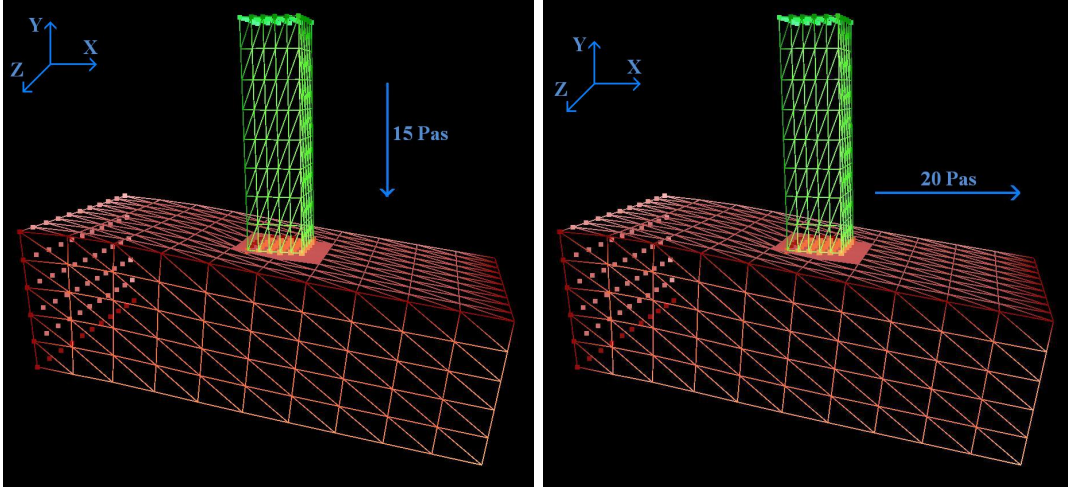


FIG. 3.3 – A gauche : définition du parcours pénétration, initialement, les deux objets déformables se touchent. A droite : définition du parcours glissement, initialement, l'objet mobile pénètre de 0.5 cm dans l'objet encastré. 1 pas = 0.1 cm.

L'objet mobile est initialement positionné au milieu de l'objet encastré, puis est soumis à une série d'incrément en déplacement de 0.1 cm suivant deux parcours différents : pénétration et glissement (figure 3.3). Le parcours «pénétration» correspond à 15 incréments en déplacements selon l'axe $-Y$. Initialement les deux objets se touchent sans aucune déformation. Pour le parcours «glissement», l'objet mobile se déplace de 20 incréments suivant l'axe X . Initialement, les deux objets sont déformés suite à une pénétration de 0.5 cm.

3.3 Le calcul de la raideur locale ρ

Dans la littérature, Feng et deSaxcé [dF91], [HFdM04] ont proposé une résolution des forces de contact avec frottement en simulation «temps différé», basée sur la résolution Uzawa globale des lois de contact dérivées de la formulation «bi-potentielle». Dans ces travaux on cherche avant tout la précision numérique et c'est pourquoi ils considèrent l'erreur de Gauss-Seidel ε_{GS} très faible, en dessous de 10^{-10} . Feng et al. calculent le paramètre ρ , qui apparaît dans l'écriture augmentée de la force de contact \mathbf{r}^* , à partir des valeurs diagonales de l'opérateur de Delassus afin d'éviter tout problème d'instabilité numérique.

$$\rho = \rho_{min} = \text{Inf} \left(\frac{1}{W_{ii}} \right) \quad i = 1, N_c \times 3 \quad (3.1)$$

Jean [Jea96], [Jea98] propose un algorithme dont les lois de contact sont formulées à partir de deux pseudo-potentiels où le paramètre ρ est propre à chaque point de contact ρ^α . Ce n'est plus une raideur isotrope et constante à un ensemble de points de contact (ρ_{min}) mais plutôt une raideur locale établie selon la normale ρ_n^α ou selon le plan tangentiel ρ_t^α . Ces ρ locaux figurent aussi dans les travaux de Christian Duriez [Dur04] et sont calculés de la façon suivante :

$$\begin{aligned}\rho_n^\alpha &= \left(\frac{1}{W_{nn}^\alpha} \right) \\ \rho_t^\alpha &= \left(\frac{\lambda_{min}}{\lambda_{max}^2} \right)\end{aligned}\tag{3.2}$$

λ_{min} et λ_{max} sont les valeurs propres de \mathbf{W}_{tt} , la matrice de Delassus tangentielle de taille 2×2 associé au point de contact \mathbf{P}^α .

$$\mathbf{W}_{tt}^\alpha = \begin{bmatrix} W_{t_1 t_1}^\alpha & W_{t_1 t_2}^\alpha \\ W_{t_2 t_1}^\alpha & W_{t_2 t_2}^\alpha \end{bmatrix}\tag{3.3}$$

3.4 Comparaison entre les deux formulations «pseudo-potentielle» et «bi-potentielle»

Nous allons définir trois algorithmes basés sur la résolution Uzawa globale, le premier appelé « Uzawa global bi-potentiel » dans lequel on utilise ρ_{min} , le deuxième appelé « Uzawa global pseudo-potentiel » utilisant ρ_n^α et ρ_t^α , et enfin « Uzawa global* bi-potentiel » dans lequel nous utilisons également ρ_n^α et ρ_t^α .

Nous allons tester ces trois algorithmes en terme de temps de calcul, nombre d'itérations Gauss-Seidel et norme de la force de contact. Cette phase d'expérimentation numérique nous permettra ensuite de procéder à des comparaisons au niveau de la formulation (bi-potentielle et pseudo-potentielle) et du calcul des raideurs locales fictives (ρ_{min} , ρ_n^α et ρ_t^α).

Dans cette expérimentation, l'objet mobile est en aluminium et l'objet encastré est un tissu mou. Les caractéristiques mécaniques de l'aluminium sont $E_1 = 69000$ Mpa pour le module de Young et $\nu = 0.33$ pour le coefficient de Poisson. Pour le tissu mou, nous avons $E_2 = 0.02$ Mpa pour le module de Young et $\nu = 0.4$ pour le coefficient de Poisson. Ces dernières correspondent aux caractéristiques mécaniques d'un rein d'agneau [PTDRC03]. L'aluminium est un matériau rigide pouvant entrer dans la conception d'instruments chirurgicaux. En l'absence de données bibliographiques à notre disposition, la valeur du coefficient de frottement a été fixé arbitrairement à $\mu = 0.3$. Dans la recherche d'une solution de référence, c'est à dire précise du point de vu numérique, nous avons fixé dans un premier temps l'erreur de Gauss-Seidel à 10^{-8} et le nombre maximum d'itérations Gauss-Seidel Nb-iterGS-max à 10 000. Dans le cas où cette limite est atteinte, nous concluons que l'algorithme n'a pas convergé. Ces tests ont été réalisés sur un processeur Intel(R) double coeur CPU 6600 d'une fréquence de 2.4 GHz et équipé d'une RAM de 2 Go.

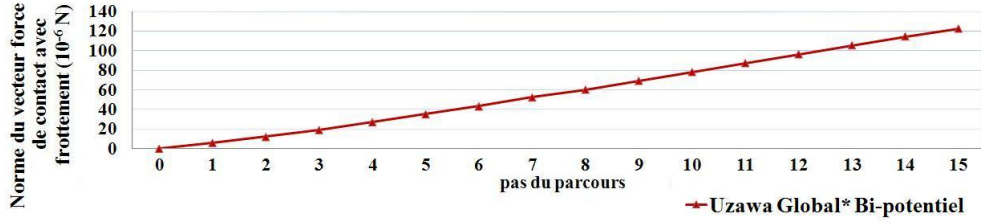


FIG. 3.6 – Norme du vecteur force de contact issu du solveur Uzawa global* bi-potentiel pendant le parcours de pénétration (interaction aluminium/tissu mou).

L'équation 3.4 mesure l'écart relatif en % entre le vecteur force de contact issu de l'algorithme Uzawa global bi-potentiel $\mathbf{r}_{\mathbf{G}\mathbf{B}}$ et celui issu de l'algorithme Uzawa global* bi-potentiel $\mathbf{r}_{\mathbf{G}^*\mathbf{B}}$. De même, l'équation 3.5 mesure l'écart relatif en % entre le vecteur force de contact issu de l'algorithme Uzawa global pseudo-potentiel $\mathbf{r}_{\mathbf{G}\mathbf{P}}$ et $\mathbf{r}_{\mathbf{G}^*\mathbf{B}}$.

$$\text{Ecart}(\%) = 100 \times \frac{\|\mathbf{r}_{\mathbf{G}\mathbf{B}} - \mathbf{r}_{\mathbf{G}^*\mathbf{B}}\|}{\|\mathbf{r}_{\mathbf{G}^*\mathbf{B}}\|} \quad (3.4)$$

$$\text{Ecart}(\%) = 100 \times \frac{\|\mathbf{r}_{\mathbf{G}\mathbf{P}} - \mathbf{r}_{\mathbf{G}^*\mathbf{B}}\|}{\|\mathbf{r}_{\mathbf{G}^*\mathbf{B}}\|} \quad (3.5)$$

Nous avons constaté un écart important, supérieur à 12%, entre $\mathbf{r}_{\mathbf{G}\mathbf{P}}$ et $\mathbf{r}_{\mathbf{G}^*\mathbf{B}}$. Par ailleurs, la figure 3.7 montre des écarts entre $\mathbf{r}_{\mathbf{G}\mathbf{B}}$ et $\mathbf{r}_{\mathbf{G}^*\mathbf{B}}$ inférieurs à 6.7%. La figure 3.8 montre une différence entre le nombre de contacts glissants issus des deux algorithmes Uzawa global pseudo-potentiel et Uzawa global bi-potentiel. Par contre, on obtient le même nombre de contacts glissants issus des deux algorithmes Uzawa global et Uzawa global* en bi-potentiel. La résolution Uzawa globale bi-potentielle affiche à la fin du parcours pénétration une difficulté à converger (1239 itérations) alors que la résolution Uzawa globale* bi-potentielle converge plus facilement (233 itérations).

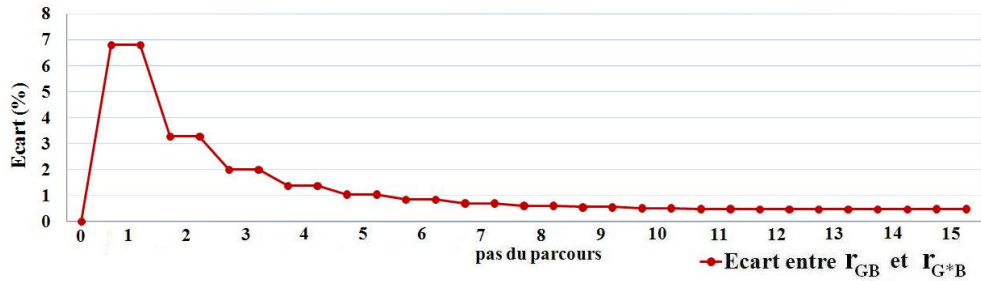


FIG. 3.7 – Ecart entre $\mathbf{r}_{\mathbf{G}\mathbf{B}}$ issu du solveur Uzawa global bi-potentiel et $\mathbf{r}_{\mathbf{G}^*\mathbf{B}}$ issu du solveur Uzawa global* bi-potentiel pendant le parcours de pénétration (interaction aluminium/tissu mou).

3.4. Comparaison entre les deux formulations «pseudo-potentielles» et «bi-potentielle» 105

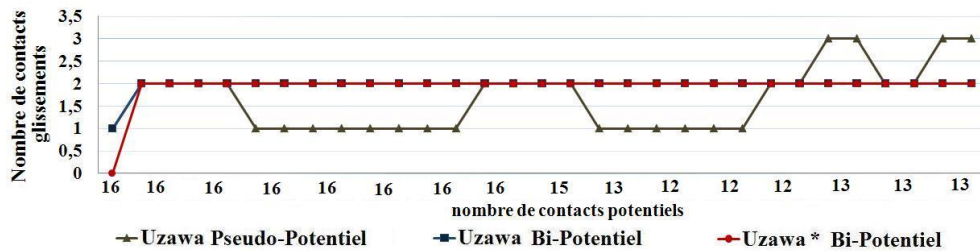


FIG. 3.8 – Nombre de contacts glissants issu des solveurs Uzawa global pseudo-potentiel, Uzawa global bi-potentiel et Uzawa global* bi-potentiel pendant le parcours de pénétration (interaction aluminium/tissu mou).

A partir des figures 3.4, 3.5, 3.7 et 3.8 nous en déduisons, pour un contact frottant aluminium/tissu mou et dans le cas du parcours en pénétration, que l’approche de résolution «Uzawa global* bi-potentielle» est la plus performante. En effet, elle converge plus rapidement et semble plus robuste. La robustesse est liée à la formulation bi-potentielle des lois de contact frottant qui tient compte du couplage entre les conditions de Signorini et la loi de frottement sec (Coulomb), contrairement à la formulation pseudo-potentielle qui ne prend pas en considération ce couplage.

Parcours glissement

Afin de confirmer ces résultats, on considère cette fois-ci le parcours glissement avec 36 contacts potentiels. La valeur de ρ_{min} décroît tout au long de ce parcours de 25×10^{-6} (N/cm) à 11×10^{-6} (N/cm) ce qui semble logique car le déplacement s’effectue de l’extrémité encastree vers l’extrémité libre. En conséquence, on observe bien une diminution de la norme de la force de contact comme le montre la figure 3.10. La figure 3.9 montre de nouveau que l’approche Uzawa global* est la plus rapide. En effet, cette dernière affiche un temps de calcul moyen égale à 46.35 ms pour un nombre moyen d’itérations de l’ordre de 969 alors que les deux algorithmes Uzawa global bi-potentiel et Uzawa global pseudo-potentiel affichent respectivement des temps de calcul moyens égaux à 63.65 ms et 332 ms pour un nombre moyen d’itérations respectifs de 1325 et 7264. Dans la figure 3.11 nous observons des écarts inférieurs à 5% entre r_{GB} et r_{G*B} . Ce maximum est atteint lorsqu’on observe une différence de nombre de contacts glissants entre les deux algorithmes comme le montre la figure 3.12. L’écart entre r_{GP} et r_{G*B} est souvent supérieur à 7% car la résolution Uzawa globale pseudo-potentielle ne converge pas souvent.

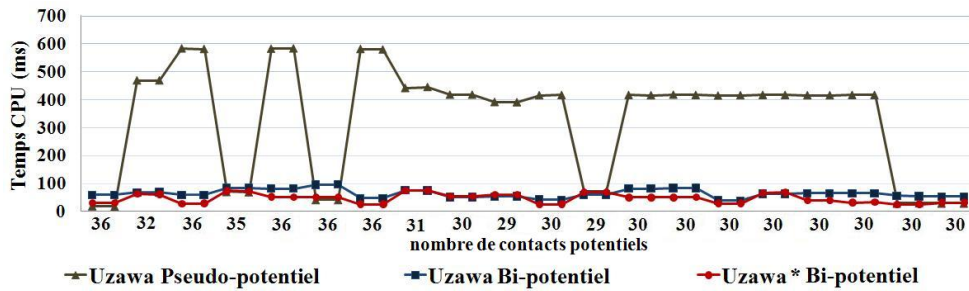


FIG. 3.9 – Comparaison en temps CPU entre les solveurs Uzawa global pseudo-potential, Uzawa global bi-potential et Uzawa global* bi-potential pendant la parcour de glissement (interaction aluminium/tissu mou).

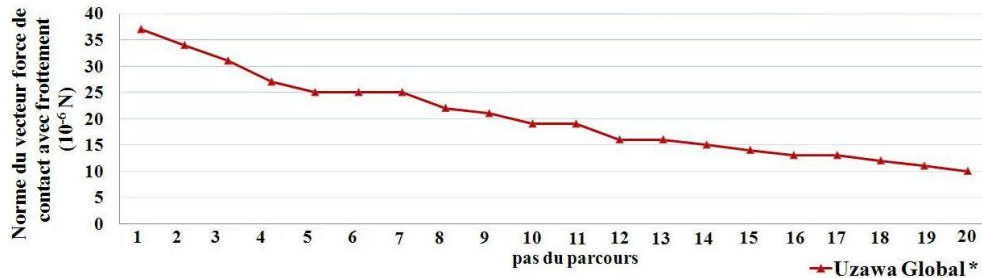


FIG. 3.10 – Norme des forces de contact avec frottement issue du solveur Uzawa global* bi-potential pendant le parcour de glissement (interaction aluminium/tissu mou).

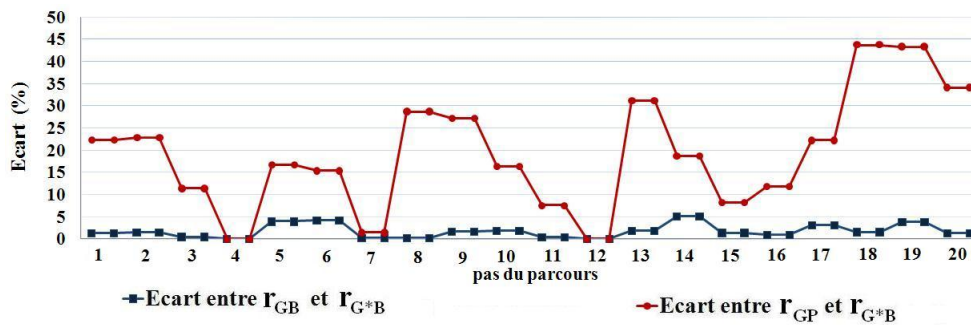


FIG. 3.11 – Ecart entre \mathbf{r}_{GB} issu du solveur Uzawa global bi-potential, \mathbf{r}_{GP} issu du solveur Uzawa global pseudo-potential et \mathbf{r}_{G*B} issu du solveur Uzawa global* bi-potential pendant le parcour de glissement (interaction aluminium/tissu mou).

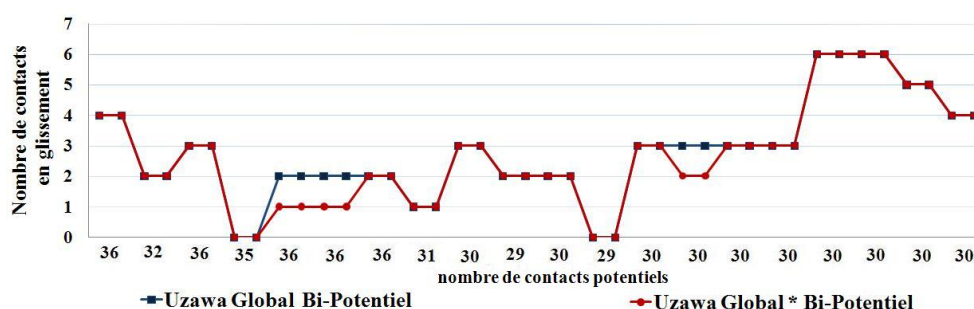


FIG. 3.12 – Nombre de contacts glissants issu des solveurs Uzawa global bi-potentiel et Uzawa global* bi-potentiel pendant le parcours de glissement (interaction aluminium/tissu mou).

Conclusion

A partir de ces expérimentations nous pouvons conclure que, pour un contact frottant aluminium/tissu mou dans les deux types de parcours considérés (pénétration, glissement), la résolution «Uzawa globale* bi-potentielle» est la plus performante et constituera par la suite notre solution de référence. La résolution «Uzawa globale pseudo-potentielle» telle qu'elle a été programmée n'est pas suffisamment robuste. En effet, cette formulation ne tient pas compte du couplage entre les conditions de Signorini et la loi de Coulomb. Une meilleure programmation consisterait à itérer d'abord sur l'opérateur de projection lié aux conditions de Signorini jusqu'à convergence, pour ensuite itérer à nouveau sur l'opérateur de projection lié à loi de Coulomb. La formulation pseudo-potentielle nécessite donc au moins une boucle itérative supplémentaire par rapport à la formulation bi-potentielle où la seule boucle itérative globale est suffisante. Nous ne poursuivrons pas d'avantage nos expériences numériques sur la formulation pseudo-potentielle, pour concentrer nos efforts, à partir de maintenant, sur la formulation bi-potentielle.

3.5 Recherche d'un ρ optimal

Nous aimerions maintenant connaître les raisons pour lesquelles la résolution Uzawa globale* est meilleure que la résolution Uzawa globale. En effet, les deux algorithmes sont les mêmes, ils ne présentent de différence qu'au niveau du calcul des raideurs fictives. Par conséquent, nous souhaitons connaître l'influence du changement de la valeur de ces derniers sur les temps de calcul et la stabilité numérique.

Nous savons que $\rho_{min} \leq \text{Inf}(\rho_n^\alpha)$, est ce une contrainte trop forte ?. En d'autres termes, est-il possible d'augmenter la valeur de ρ_{min} ?.

En effet, nous avons observé un rapport de 2 entre ρ_{min} et $Max(\rho_n^\alpha)$, nous allons donc envisager l'algorithme Uzawa global avec $\rho = \rho_{min} \times 2$. Si on augmente la valeur des raideurs fictives, on augmente en principe la vitesse de convergence, par contre on risque de rencontrer des problèmes d'instabilité numérique. Afin de montrer l'influence des raideurs fictives sur la vitesse de convergence, nous avons aussi testé deux autres variantes, Uzawa globale, avec $\rho = \rho_{min}/2$ et Uzawa globale* avec $\rho^\alpha = (\rho_n^\alpha, \rho_t^\alpha)/2$.

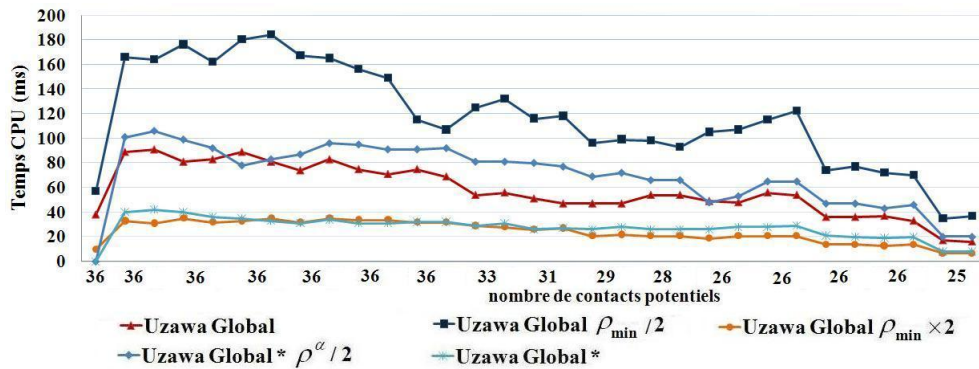


FIG. 3.13 – Comparaison en temps CPU entre les différents solveurs pendant le parcours de pénétration (interaction aluminium/tissu mou).

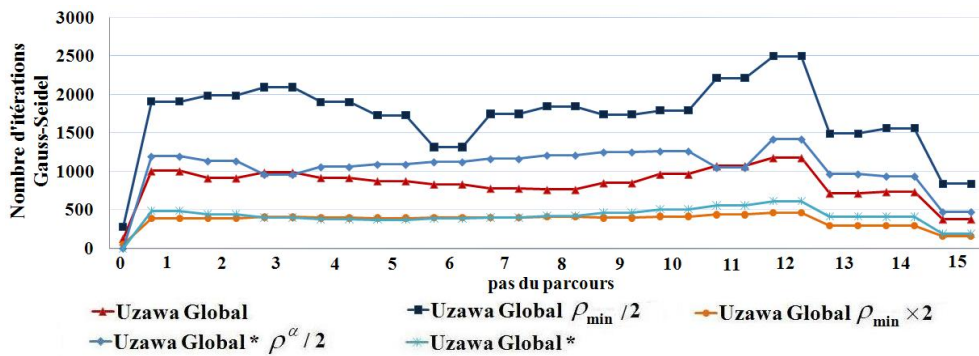


FIG. 3.14 – Comparaison en nombre d'itérations Gauss-Seidel entre les différents solveurs pendant le parcours de pénétration (interaction aluminium/tissu mou).

Les figures 3.13 et 3.14 montrent une même tendance. En effet, diviser la valeur de ρ_{min} par 2 dans la résolution Uzawa global implique un temps de calcul et un nombre d'itérations Gauss-Seidel multiplié par 2. Par contre, multiplier la valeur de ρ_{min} par 2 dans cette même résolution implique de meilleures performances dépassant même celles données par la résolution Uzawa globale*, le temps de calcul et le nombre d'itérations Gauss-Seidel sont divisés par 2. Le même constat est observé sur l'approche Uzawa globale*, diviser les raideurs locales par deux double le temps de calcul ainsi que le nombre d'itérations Gauss-Seidel. Par contre, dans cette dernière approche, multiplier par 2 les raideurs locales fictives donne des instabilités numériques.

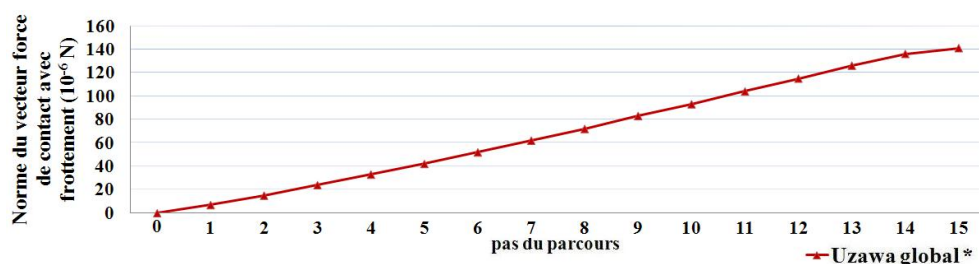


FIG. 3.15 – Norme des forces de contact avec frottement issues du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/tissu mou).

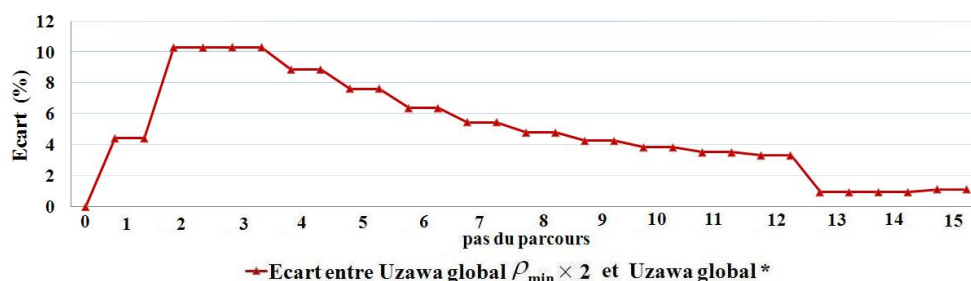


FIG. 3.16 – Écart entre le vecteur force de réaction avec frottement issu du solveur Uzawa global avec $\rho_{min} \times 2$ et celui issu du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/tissu mou).

La valeur moyenne de la raideur locale ρ_{min} est estimée à 26×10^{-6} (N/cm). La figure 3.15 montre l'évolution de la norme de la force de contact en fonction du déplacement vertical dans le parcours de pénétration. On constate respectivement sur les deux figures 3.16 et 3.17, un écart moyen au niveau du calcul des forces de contact de 5.07% entre \mathbf{r}_{GB} et \mathbf{r}_{G^*B} et de $10^{-40}\%$ entre \mathbf{r}_{G^*B} avec $\rho^\alpha/2$ et \mathbf{r}_{G^*B} . La figure 3.18 affiche le même nombre de contacts glissants issus des deux algorithmes Uzawa global et Uzawa global*. Dans ces essais numériques, il n'y a aucun contact adhérent.

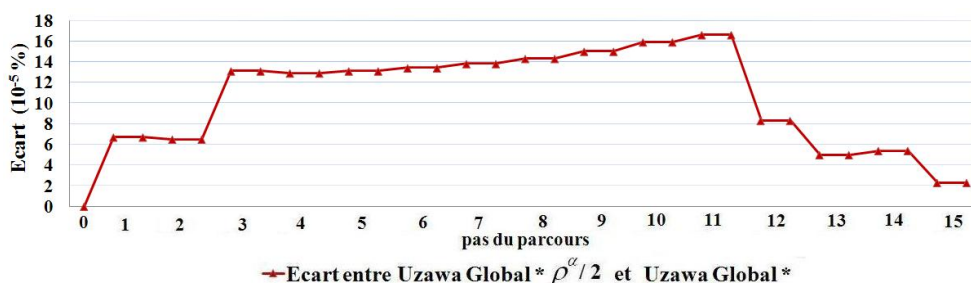


FIG. 3.17 – Écart entre le vecteur force de contact avec frottement issu du solveur Uzawa global* avec $\rho^\alpha/2$ et celui issu du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/tissu mou).

une longueur caractéristique de la scène virtuelle et ε_{Uzawa} une tolérance numérique sans dimension. On définit 4 algorithmes supplémentaires basés sur l'algorithme Uzawa local :

- «Uzawa local» considère ρ_{min} comme raideur locale fictive et l'équation 3.6 comme critère d'arrêt, \mathbf{r}_L désigne le vecteur force de contact associé.
- «Uzawa local*» considère ρ_n^α et ρ_t^α comme raideurs locales fictives et l'équation 3.6 comme critère d'arrêt, \mathbf{r}_{L^*} désigne le vecteur force de contact associé.
- «Uzawa local Signorini» considère ρ_{min} comme raideur locale fictive et l'équation 3.7 comme critère d'arrêt, \mathbf{r}_{LS} désigne le vecteur force de contact associé.
- «Uzawa local Signorini*» considère ρ_n^α et ρ_t^α comme raideurs locales fictives et l'équation 3.7 comme critère d'arrêt, \mathbf{r}_{LS^*} désigne le vecteur force de contact associé.

$$\frac{\|\mathbf{r}^\alpha - \mathbf{r}^{(j-1)\alpha}\|}{\rho} \leq \varepsilon_{Uzawa} \times L \quad (3.6)$$

$$x_n^\alpha \geq -\varepsilon_{Uzawa} \times L \quad (3.7)$$

Dans ce qui suit, nous allons comparer ces différents algorithmes en fonction des propriétés mécaniques des objets en contact (module de Young et coefficient de frottement). On garde la même solution de référence c'est à dire celle issue de l'algorithme Uzawa global* avec une erreur de Gauss-Seidel de l'ordre de 10^{-8} et 36 points en contact. Le nombre maximal d'itérations Gauss-Seidel est réduit à 5000. Les expérimentations qui suivent ont été réalisées avec un coefficient de frottement égal à 0.3, une erreur Gauss-Seidel égale à 10^{-4} et une erreur d'Uzawa (tolérance de pénétration) de 10^{-3} . Le processeur utilisé est un Intel(R) Xeon(R) CPU 5110 double coeur d'une fréquence de 1.60 GHz. Ceci entraîne des temps de calcul plus élevés que ceux constatés plus haut et réalisés sur un processeur Intel(R) double coeur CPU 6600 cadensé à 2.4 GHz.

3.7 Expérimentations en fonction des modules d'Young

Nous aimerions savoir si la nature des matériaux en contact influence les résultats au niveau de la vitesse de convergence et des écarts constatés. Nous proposons trois cas de contact différents, le premier est un contact frottant entre un objet en aluminium et un tissu mou impliquant une très grande différence entre les deux modules de Young $E_1 \gg E_2$. Le deuxième cas concerne le contact frottant entre un objet en aluminium et un caoutchouc, impliquant une différence moins importante entre les deux modules de Young $E_1 > E_2$. Le dernier cas est celui du contact frottant entre deux tissus mous de même module de Young faible $E_1 = E_2$.

3.7.1 Contact frottant entre aluminium et tissu-mou

Dans ce cas $E_1 = 69000$ Mpa et $E_2 = 0.02$ Mpa d'où des raideurs locales fictives très faibles. En effet, la raideur locale fictive peut être considérée comme l'association de deux raideurs locales en série (voir chapitre 2). Les figures 3.20 et 3.21 montrent les résultats en temps de calcul et en écarts obtenus pendant le parcours de pénétration. Les figures 3.22 et 3.23 montrent les mêmes types de résultat pour le parcours de glissement.

Résolution Uzawa	Temps moyens (ms) parcours de pénétration
Globale * ρ^α	7.74
Locale * ρ^α	9.41
Globale ρ_{min}	13.29
Locale ρ_{min}	14.22
Locale Signorini ρ_{min}	22.41
Locale * Signorini ρ^α	24.48

FIG. 3.20 – Temps CPU moyen en ms des différents solveurs pendant le parcours de pénétration (interaction aluminium/tissu mou).

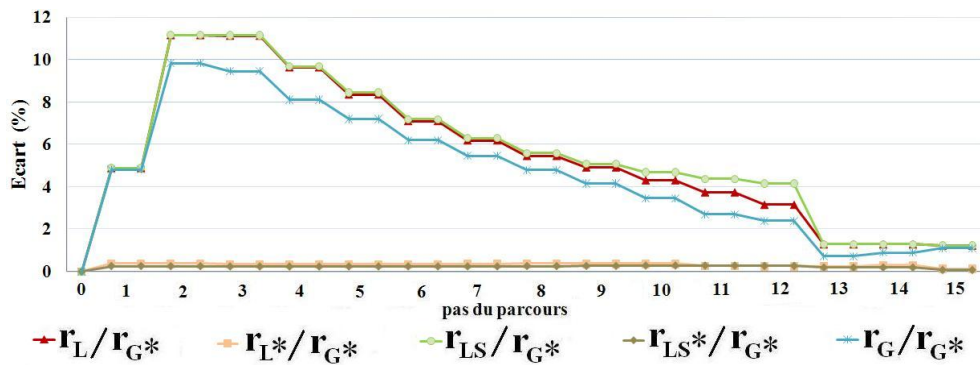


FIG. 3.21 – Ecart entre les différents vecteurs force de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/tissu mou).

Résolution Uzawa	Temps moyens (ms) parcours de glissement
Globale * ρ^α	18.02
Locale * ρ^α	19.57
Globale ρ_{min}	19.85
Locale * Signorini ρ^α	20.32
Locale ρ_{min}	26.77
Locale Signorini ρ_{min}	29.1

FIG. 3.22 – Temps CPU moyen en ms des différents solveurs pendant le parcours de glissement (interaction aluminium/tissu mou).

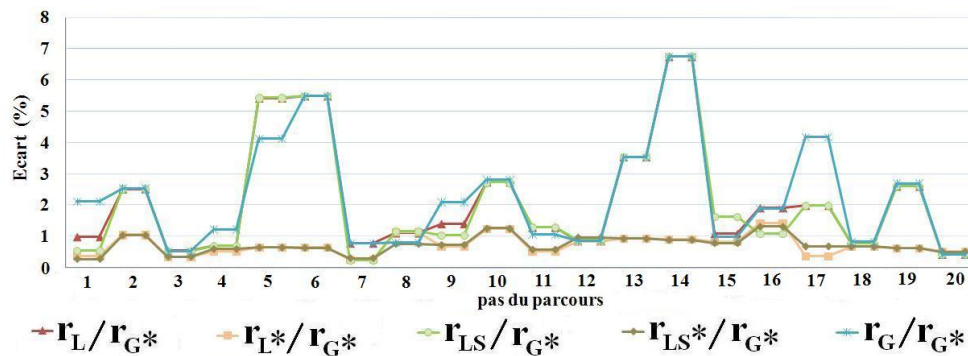


FIG. 3.23 – *Écarts entre les différents vecteurs force de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de glissement (interaction aluminium/tissu mou).*

Les deux figures 3.20 et 3.22 montrent que les algorithmes Uzawa global* et Uzawa local* sont les meilleurs en temps de calcul. En effet, nous retrouvons la même tendance que précédemment. L'utilisation de ρ_n^α et ρ_t^α confère une meilleure vitesse de convergence que l'utilisation de ρ_{min} et ceci quelque soit l'approche, locale ou globale. Par contre, on constate que l'approche locale n'apporte pas d'amélioration par rapport à l'approche globale et semble même plus lente surtout dans le parcours glissement. De même la résolution locale basée sur le critère d'arrêt 3.7 n'apporte pas d'amélioration notable.

D'après les figures 3.21 et 3.23, ce n'est pas l'approche globale ou locale qui influence les écarts mais la façon de calculer les raideurs locales fictives. En effet, les algorithmes qui utilisent ρ_n^α et ρ_t^α donnent de plus petits écarts (écart moyen de 0.27% pour le parcours pénétration et 0.71% pour le parcours glissement) que les algorithmes qui utilisent ρ_{min} (écart moyen de 5.36% pour le parcours pénétration et 2.16% pour le parcours glissement). Cela est cohérent car la référence (Uzawa global*) utilise ρ_n^α et ρ_t^α .

3.7.2 Contact frottant entre aluminium et caoutchouc

Dans ce cas $E_1 = 69000$ Mpa et $E_2 = 2350$ Mpa d'où des raideurs locales fictives plus importantes que celles constatées dans le cas précédent. En effet, ρ_{min} admet une valeur moyenne égale à 2.17 (N/cm) pendant le parcours de pénétration, et varie de 2.16 (N/cm) à 2.06 (N/cm) pendant le parcours de glissement. Les figures 3.26 et 3.29 montrent respectivement l'évolution de la norme de la force de contact pendant les parcours pénétration et glissement, calculée à partir de l'algorithme de référence. Les figures 3.24 et 3.25 montrent les résultats en temps de calcul et en écart obtenus pendant le parcours de pénétration. Les figures 3.27 et 3.28 montrent le même type de résultats pour le parcours de glissement.

Résolution Uzawa	Temps moyens (ms) parcours de pénétration
Globale ρ_{min}	19.64
Globale * ρ^α	22.35
Locale Signorini ρ_{min}	23
Locale * ρ^α	24.90
Locale ρ_{min}	25.25
Locale * Signorini ρ^α	36.45

FIG. 3.24 – Temps CPU moyen en ms des différents solveurs pendant le parcours de pénétration (interaction aluminium/caoutchouc).

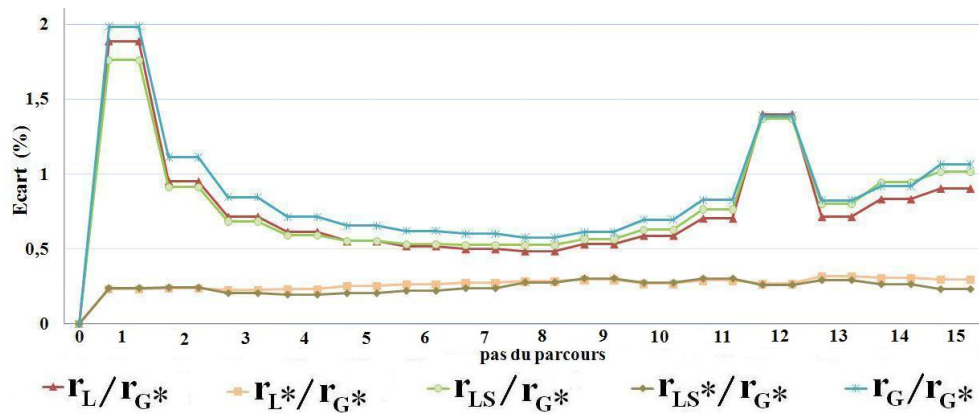


FIG. 3.25 – Ecart entre les différents vecteurs force de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/caoutchouc).

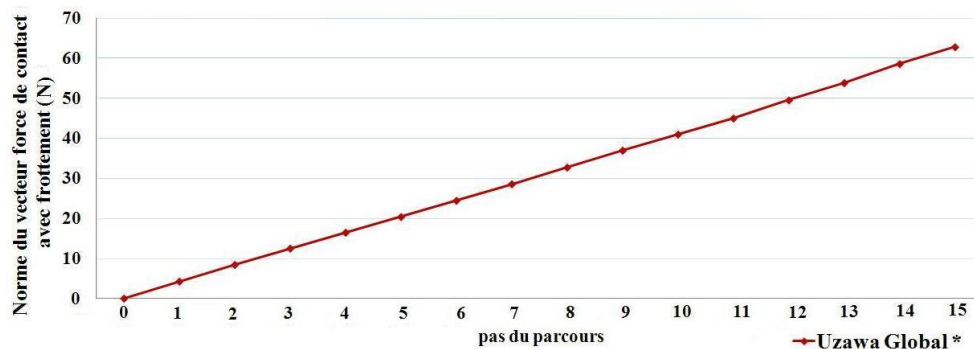


FIG. 3.26 – Norme du vecteur force de contact avec frottement issu du solveur Uzawa global* pendant le parcours de pénétration (interaction aluminium/caoutchouc).

Résolution Uzawa	Temps moyens (ms) parcours de glissement
Globale ρ_{min}	23.22
Globale * ρ^α	23.27
Locale Signorini ρ_{min}	25.57
Locale * Signorini ρ^α	25.67
Locale ρ_{min}	25.72
Locale * ρ^α	26.02

FIG. 3.27 – Temps CPU moyen en ms des différents solveurs pendant le parcours de glissement (interaction aluminium/caoutchouc).

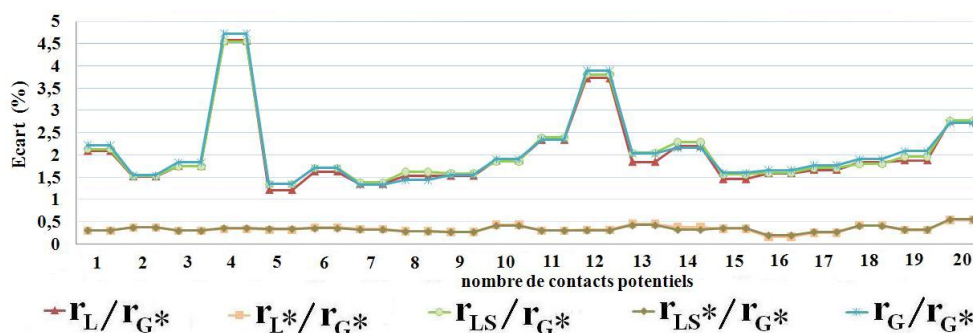


FIG. 3.28 – Ecart entre les différents vecteurs forces de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de glissement (interaction aluminium/caoutchouc).

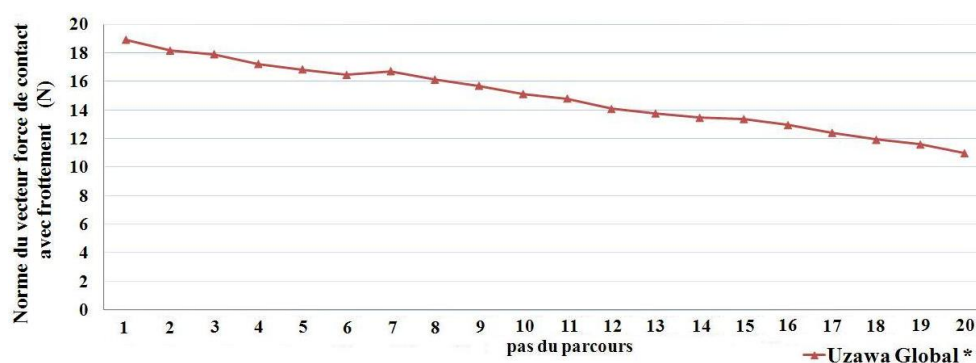


FIG. 3.29 – Norme du vecteur force de contact avec frottement issu du solveur Uzawa global* pendant le parcours de glissement (interaction aluminium/caoutchouc).

Dans ce cas, la tendance observée pour le contact aluminium/tissu mou n'est plus valable. En effet, considérer les raideurs locales fictives ρ_n^α et ρ_t^α n'implique plus de meilleures performances. On constate que l'algorithme Uzawa global est le plus rapide dans les deux

parcours pénétration et glissement. Nous expliquons ceci par le fait que considérer un ρ_n^α différent de ρ_t^α dans la formulation bi-potentielle modifie en réalité l'opérateur de projection qui utilise un unique ρ^α . Ceci entraîne une modification du couplage entre la composante normale et la composante tangentielle dans la formulation augmentée de la force de contact.

En ce qui concerne les écarts, les figures 3.25 et 3.28 confirment le fait que ce n'est pas l'approche globale ou locale qui influence les écarts mais la façon de calculer les raideurs locales fictives. En effet, les algorithmes qui utilisent ρ_n^α et ρ_t^α donnent de plus petits écarts (écart moyen de 0.24% pour le parcours pénétration et 0.33% pour le parcours glissement) que les algorithmes qui utilisent ρ_{min} (écart moyen de 0.8% pour le parcours pénétration et 2.06% pour le parcours glissement). On a remarqué que les écarts maximum se produisent lors de transitions adhérence/glissement au niveau des points de contact.

3.7.3 Contact frottant entre deux tissus mous

Dans ce cas, $E_1 = 0.02$ Mpa et $E_2 = 0.02$ Mpa d'où des raideurs locales fictives faibles. En effet, ρ_{min} admet une valeur moyenne égale à de 6.9×10^{-7} (N/cm) pendant le parcours de pénétration, et varie de 6.93×10^{-7} (N/cm) à 6.75×10^{-7} (N/cm) pendant le parcours de glissement. Les figures 3.32 et 3.35 montrent respectivement l'évolution de la norme de la force de contact pendant les parcours pénétration et glissement, calculée à partir de l'algorithme de référence. Les figures 3.30 et 3.31 montrent les résultats en temps de calcul et en écarts obtenus pendant le parcours de pénétration. Les figures 3.33 et 3.34 montrent les mêmes types de résultat pour le parcours de glissement.

Résolution Uzawa	Temps moyens (ms) parcours de pénétration
Locale Signorini ρ_{min}	50,13
Globale ρ_{min}	52,77
Locale ρ_{min}	60
Globale * ρ^α	503,25 (Diverge)
Locale * Signorini ρ^α	672,96(Diverge)
Locale * ρ^α	815,54 (Diverge)

FIG. 3.30 – Temps CPU moyen en ms des différents solveurs pendant le parcours de pénétration (interaction tissu mou/tissu mou).

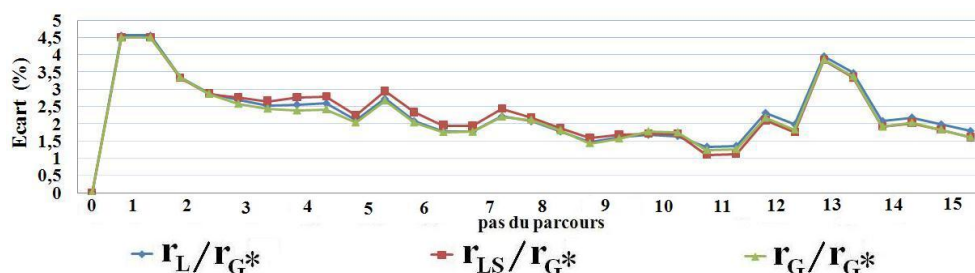


FIG. 3.31 – *Ecart entre les différents vecteurs force de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de pénétration (interaction tissu mou/tissu mou).*

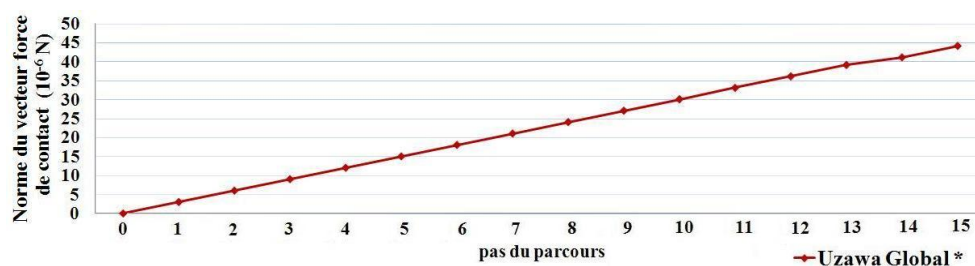


FIG. 3.32 – *Norme du vecteur force de contact avec frottement pendant le parcours de pénétration (interaction tissu mou/tissu mou).*

Résolution Uzawa	Temps moyens (ms) parcours de glissement
Globale ρ_{min}	72.02
Locale Signorini ρ_{min}	75.35
Locale ρ_{min}	77.75
Globale * ρ^α	104.1 (Diverge)
Locale * Signorini ρ^α	105.55 (Diverge)
Locale * ρ^α	109,45 (Diverge)

FIG. 3.33 – *Temps CPU moyen en ms des différents solveurs pendant le parcours de glissement (interaction tissu mou/tissu mou).*

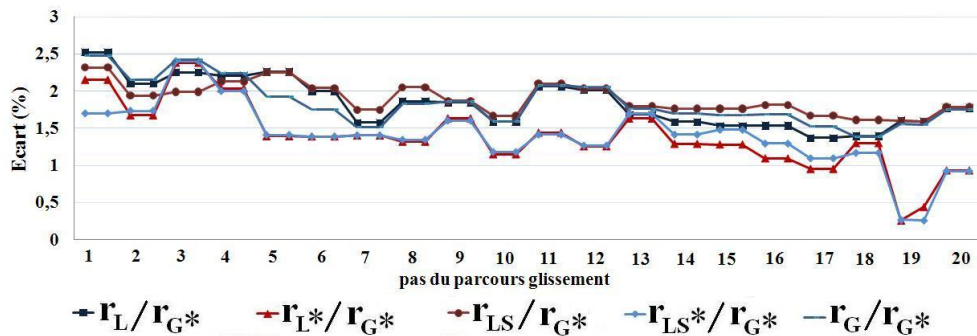


FIG. 3.34 – Ecart entre les différents vecteurs force de contact avec frottement issus des différents solveurs et celui issu du solveur Uzawa global* pendant le parcours de glissement (interaction tissu mou/tissu mou).

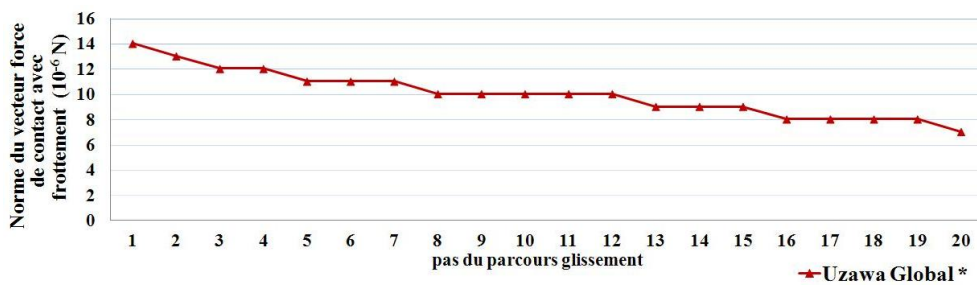


FIG. 3.35 – Norme du vecteur force de contact avec frottement pendant le parcours glissement.

Dans ce cas (contact tissu mou/tissu mou), la tendance observée pour le contact aluminium/caoutchouc reste valable. En effet, considérer les raideurs locales fictives ρ_n^α et ρ_t^α n'implique pas toujours de meilleures performances. On constate que les algorithmes Uzawa local Signorini et Uzawa global sont les plus rapides dans les deux parcours pénétration et glissement. Dans ce cas, ce n'est pas l'approche globale ou locale qui influence la vitesse de convergence mais le choix de la raideur locale fictive. On remarque que les algorithmes qui utilisent les ρ_n^α et ρ_t^α converge rarement. Toutefois, les écarts sur les forces de contact entre les méthodes (figures 3.31 et 3.34) possèdent une moyenne de 2.28% pour le parcours pénétration et de 1.7% pour le parcours glissement.

3.8 Expérimentations en fonction du coefficient de frottement

Résolution Uzawa	Temps moyens (ms) parcours de pénétration			
	$\mu = 0.1$	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$
Globale * ρ^α	10	7.74	11.25	11.80
Locale * ρ^α	11.22	9.41	9.71	11.54
Globale ρ_{min}	15.87	13.29	17.35	20.5
Locale ρ_{min}	17.38	14.22	14.19	17.16
Locale * Signorini ρ^α	24	24.48	16.09	14.13
Locale Signorini ρ_{min}	21.16	22.41	23.8	19

FIG. 3.36 – Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours pénétration (interaction aluminium/tissu mou).

Résolution Uzawa	Temps moyens (ms) parcours de glissement			
	$\mu = 0.1$	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$
Globale * ρ^α	13.05	18.02	25.82	32.77
Globale ρ_{min}	18.17	19.85	28.27	29.12
Locale * ρ^α	14.3	19.57	31.1	41.52
Locale * Signorini ρ^α	13.92	20.32	31.7	42.82
Locale ρ_{min}	22.2	26.77	36.57	43.55
Locale Signorini ρ_{min}	26.75	29.1	41	48.7

FIG. 3.37 – Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours glissement (interaction aluminium/tissu mou).

Résolution Uzawa	Temps moyens (ms) parcours de pénétration			
	$\mu = 0.1$	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$
Globale ρ_{min}	15.8	19.64	25.43	28.36
Globale * ρ^α	9.53	22.35	26.3	30.96
Locale Signorini ρ_{min}	19.4	23	27.7	35.36
Locale ρ_{min}	14.03	25.25	29.7	35.9
Locale * ρ^α	10.86	24.90	690 (Div)	722 (Div)
Locale * Signorini ρ^α	27.03	36.45	571 (Div)	644 (Div)

FIG. 3.38 – Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours pénétration (interaction aluminium/caoutchouc).

Résolution Uzawa	Temps moyens (ms) parcours de glissement			
	$\mu = 0.1$	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$
Globale ρ_{min}	18.45	23.22	27.22	30.82
Globale * ρ^α	13.82	23.27	27.15	55.2 (Div)
Locale ρ_{min}	16.82	25.72	30.37	33.65
Locale Signorini ρ_{min}	20.85	25.57	29.15	33.1
Locale * Signorini ρ^α	16.27	25.67	216.9 (Div)	545,9 (Div)
Locale * ρ^α	14.8	26.02	427.7 (Div)	656 (Div)

FIG. 3.39 – Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours glissement (interaction aluminium/caoutchouc).

Résolution Uzawa	Temps moyens (ms) parcours de pénétration			
	$\mu = 0.1$	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$
Globale ρ_{min}	25.38	52.77	81.16	120
Locale Signorini ρ_{min}	30.22	50.13	85	134.25
Locale ρ_{min}	29.03	60	126.06	139.31
Globale * ρ^α	29.25	503.25 (Div)	492 (Div)	246.5 (Div)
Locale * Signorini ρ^α	51.42	672.9 (Div)	702 (Div)	446.82 (Div)
Locale * ρ^α	28.71	815 (Div)	994 (Div)	835 (Div)

FIG. 3.40 – Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours pénétration (interaction tissu mou/tissu mou).

Résolution Uzawa	Temps moyens (ms) parcours de glissement			
	$\mu = 0.1$	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$
Globale ρ_{min}	45.7	72.02	96.75	88.6
Locale Signorini ρ_{min}	50.67	75.35	98.42	90.03
Locale ρ_{min}	54.52	77.75	103.87	99.66
Globale * ρ^α	55.55	104.1 (Div)	140.92 (Div)	341.3 (Div)
Locale * Signorini ρ^α	58.6	105.55 (Div)	241.67 (Div)	593.17 (Div)
Locale * ρ^α	58.6	109.45 (Div)	554.52 (Div)	643.7 (Div)

FIG. 3.41 – Temps CPU moyen (ms) des différents solveurs en fonction du coefficient de frottement pendant le parcours glissement (interaction tissu mou/tissu mou).

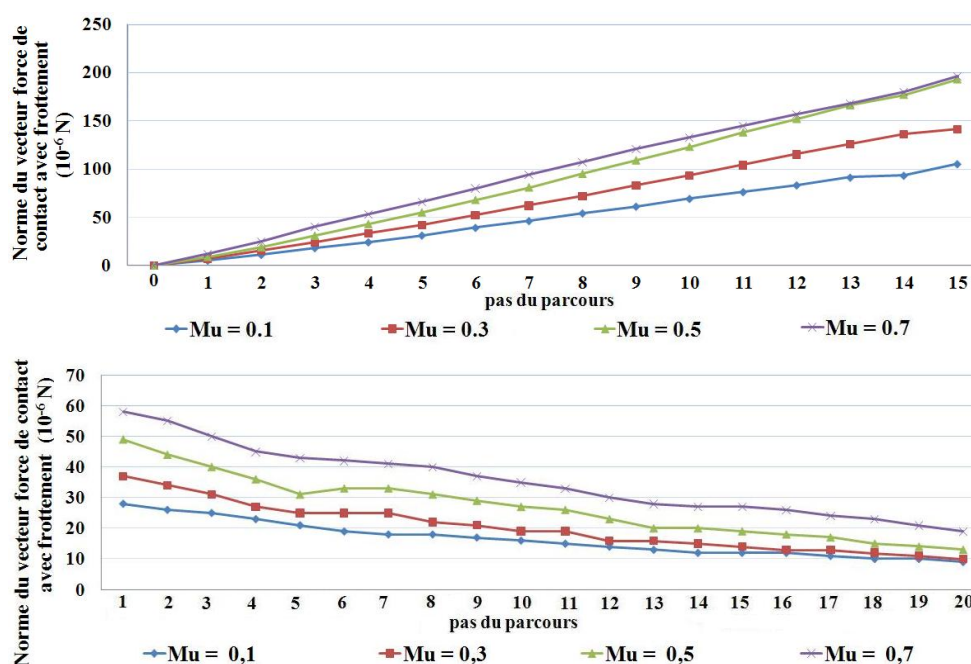


FIG. 3.42 – Norme des vecteurs forces de contact issus du solveur Uzawa global* en fonction du coefficient de frottement pendant le parcours pénétration et glissement (interaction aluminium/tissu-mou).

Les tableaux 3.36 et 3.37 donnent les temps moyens de calcul obtenus pendant les parcours pénétration et glissement dans le cas du contact frottant aluminium/tissu mou en fonction du coefficient de frottement. Les tendances observées avec un coefficient de frottement égal à 0.3 restent valables, à savoir que l'algorithme Uzawa global* est le plus rapide. Les tableaux 3.38 et 3.39 donnent le temps de calcul moyen dans le cas du contact frottant entre aluminium et caoutchouc. Pour ce cas, c'est plutôt les deux algorithmes Uzawa global et Uzawa global* qui sont les plus rapides. Enfin les tableaux 3.40 et 3.41 montrent que dans le cas du contact frottant entre deux tissus mous, l'algorithme Uzawa globale est le plus adapté pour le parcours pénétration et glissement. Les résultats montrent que d'une manière générale les temps de calcul augmentent avec le coefficient de frottement. La figure 3.42 montre des normes des forces de contact qui augmentent en fonction du coefficient de frottement, pendant les parcours pénétration et glissement.

3.9 Expérimentation en fonction du nombre de points en contact

Nous allons maintenant nous intéresser à l'évolution du temps de calcul en fonction du nombre de contacts afin d'estimer la complexité des algorithmes de résolution des forces de contact frottant. En effet, le nombre de contacts est resté autour de 36 pendant les deux

parcours de pénétration et de glissement. Nous avons alors effectué d'avantages expérimentations en considérant 16 et 66 contacts. Ces tests ont été réalisés sur un PC portable Dell Latitude D830, possédant une RAM de 512 Mo et un processeur Intel(R) Core(TM) 2 Duo T7100 d'une fréquence de 1.80 GHz. Nous avons considéré un coefficient de frottement égal à 0.3, une erreur Gauss-Seidel égale à 10^{-4} et une erreur d'Uzawa de valeur 10^{-3} .

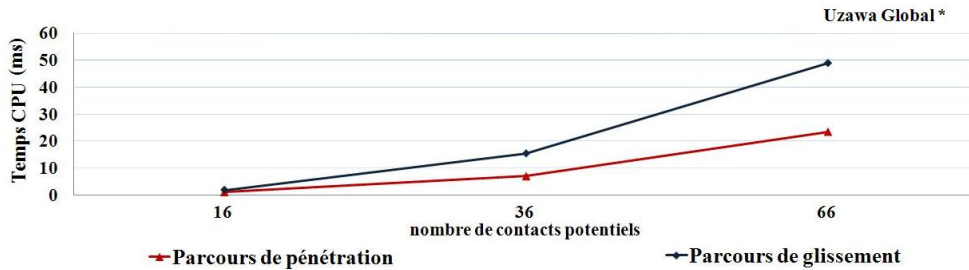


FIG. 3.43 – Evolution du temps de calcul de l'algorithme Uzawa global* en fonction du nombre de contacts (interaction aluminium/tissu mou).

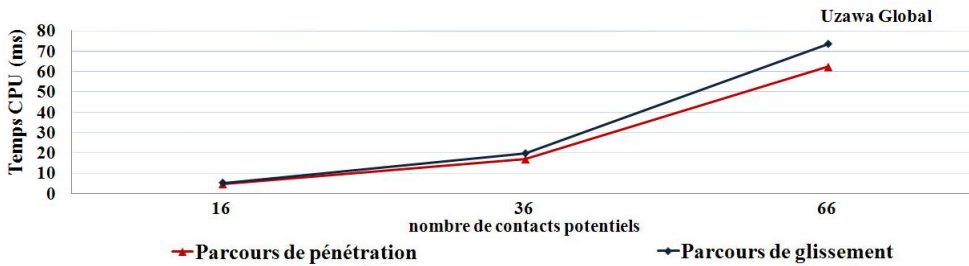


FIG. 3.44 – Evolution du temps de calcul de l'algorithme Uzawa globale en fonction du nombre de contacts (interaction aluminium/caoutchouc).

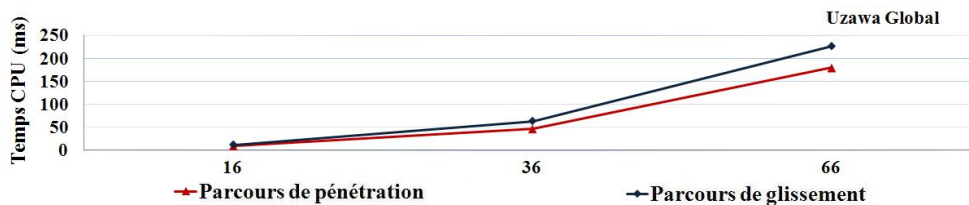


FIG. 3.45 – Evolution du temps de calcul de l'algorithme Uzawa global en fonction du nombre de contacts (interaction tissu mou/tissu mou).

On remarque que les graphes 3.43, 3.44 et 3.45 tendent vers une courbe d'équation $temps = a \times n^{ordre}$ tel que a représente un coefficient réel et n le nombre de points en contact. Les expérimentations effectuées sur le banc d'essai 3.3 implique une complexité égale à $O(n^{2.07})$ pour l'algorithme Uzawa global* et de $O(n^{2.05})$ pour l'algorithme Uzawa

global pour les différents contacts frottant aluminium/tissu mou, aluminium/caoutchouc et tissu mou/tissu mou et ceci pour les deux parcours pénétration et glissement. Nous avons aussi réalisé des expérimentations sur un deuxième banc d'essai montré dans la figure 3.46.

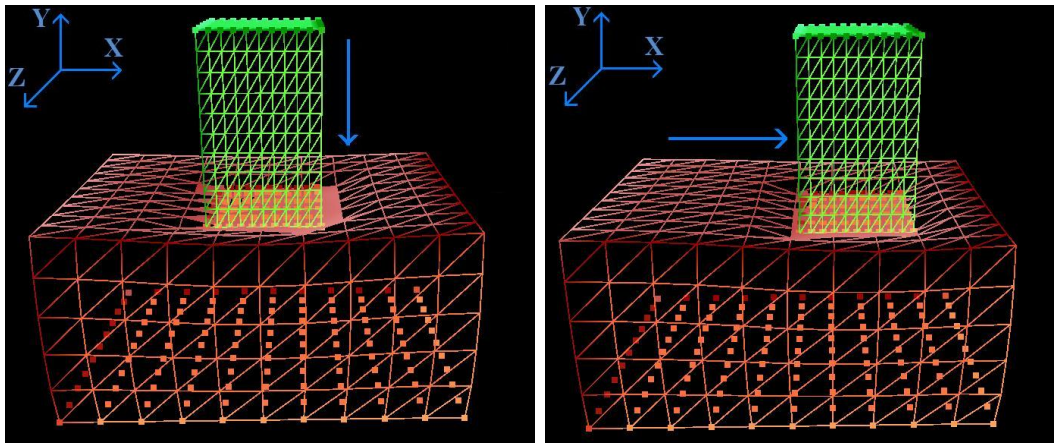


FIG. 3.46 – Banc d'essai impliquant 66 contact, à gauche : parcours pénétration, à droite : parcours glissement.

Ce banc d'essai (figure 3.46) représente également deux parallélépipèdes en interaction, mais les noeuds bloqués qui sont en surbrillance ne sont plus les mêmes. L'objet encastré en rouge est composé de 726 noeuds, 500 éléments cubiques et admet les dimensions $L = 5$ cm, $H = 2.5$ cm et $l = 3.2$ cm. Le volume de chaque élément représente 0.2% du volume total. La déformation de l'objet vert est déterminée autour d'une configuration de référence réactualisée suivant un mouvement de corps rigide imposé (approche corotationnelle globale). Ce solide est aussi composé de 726 noeuds, 500 éléments cubiques afin de donner 66 contacts. Il admet les dimensions $L = 1.425$ cm, $H = 2.4$ cm et $l = 1.425$ cm. Le volume de chaque élément représente 0.2% du volume total.

Les expérimentations effectuées sur ce banc d'essai 3.46 implique une complexité égale à $O(n^{2.83})$ pour l'algorithme Uzawa global* et de $O(n^{2.46})$ pour l'algorithme Uzawa global pour les différents contacts frottant aluminium/tissu mou, aluminium/caoutchouc et tissu mou/tissu mou et ceci pour les deux parcours pénétration et glissement.

3.10 Conclusion

Nous avons évalué et testé dans ce chapitre différentes stratégies pour résoudre des problèmes de contact frottant entre objets déformables en considérant trois types de matériau (caoutchouc, aluminium, tissu mou). Ces stratégies diffèrent essentiellement sur 3 points :

- résolution Uzawa globale ou locale,

- raideurs locales fictives ρ_{min} ou $(\rho_n^\alpha, \rho_t^\alpha)$,
- formulation pseudo-potentielle ou bi-potentielle des lois de contact.

D'après les résultats obtenus, il ne semble pas que la résolution Uzawa locale apporte une amélioration au niveau du temps de calcul, elle est même moins performante dans la plupart des cas. C'est donc l'approche Uzawa globale qui est la meilleure c'est à dire celle utilisée par Zhi Qang Feng dans ses travaux.

La formulation Bi-potentielle est plus robuste que la formulation Pseudo-potentielle. En effet cette dernière ne converge plus dans le cas d'interaction entre tissus mous, et converge difficilement dans le cas d'interaction aluminium/caoutchouc. Il faut toutefois noter que cette formulation n'est sans doute pas appropriée à une résolution type Uzawa globale (une seule itération prédiction/correction) mais plutôt adaptée à une résolution type Newton que nous n'avons pas testé, nous renvoyons le lecteur intéressé à la thèse de Christian Duriez [Dur04].

Le choix des raideurs fictives s'avère plus difficile à analyser. Dans certain cas il est possible d'augmenter la valeur de ρ^α pour accélérer la vitesse de convergence et les temps de calcul, mais pas dans tous les cas. En effet, nous avons constaté que cela dépendait des raideurs relatives entre les objets en contact. Ainsi dans le cas aluminium/tissu mou, on augmente nettement la vitesse de convergence lorsque l'on augmente ρ^α . Le choix de $(\rho_n^\alpha, \rho_t^\alpha)$ semble alors judicieux car non aléatoire. Cependant dans le cas aluminium/caoutchouc on ne constate plus la même tendance. Dans le cas tissu mou/tissu mou, les algorithmes ont du mal à converger avec $(\rho_n^\alpha, \rho_t^\alpha)$ alors qu'avec ρ_{min} on converge dans tous les cas. D'autre part choisir $(\rho_n^\alpha, \rho_t^\alpha)$ dans une formulation bi-potentielle ne semble pas cohérent car elle est formulée avec une raideur fictive constante dans toutes les directions. Le choix de ρ_{min} semble s'imposer si l'on veut traiter tout type de matériau.

En conclusion, la méthode globale bi-potentielle avec ρ_{min} est la méthode la plus robuste et s'avère la plus rapide dans le cas d'interaction entre tissus mous. Cette méthode contrôle parfaitement la solution dans tous les cas. Augmenter la raideur fictive est un art difficile qu'il faut déconseiller. Pour diminuer les temps de calculs on peut diminuer le nombre d'itérations, en stoppant la solution avant convergence.

HapCo : un prototype de simulateur temps réel avec retour haptique traitant des contacts multiples entre objets déformables.

4.1 Introduction

Les simulations interactives temps-réel réalistes basées sur les lois de la physique sont de plus en plus populaires. Elles sont dorénavant utilisées dans des domaines aussi divers que les jeux vidéo, les effets spéciaux des films, l'industrie manufacturière ou bien la chirurgie.

Dans ce chapitre nous présenterons «HapCo» qui est un logiciel de simulation interactive avec rendus multimodaux, visuel et haptique. Il permet de traiter et de résoudre le multi-contact frottant entre objets déformables.

L'objectif de ce simulateur est de proposer une plate forme logicielle intégrant la méthode de flexibilité qui permet de séparer le calcul des déplacements de celui des forces de contact. En outre, la possibilité de tester et de comparer les différents algorithmes de résolution du contact frottant entre objets déformables (Uzawa globale, Uzawa locale, Newton-Raphson) va nous permettre de valider ou non la précision et la stabilité numérique de ces méthodes.

Ce simulateur permet de manipuler et de faire interagir entre eux des objets déformables, ou même rigides, à l'aide d'une interface haptique de type PHANTOM desktop ou Omni de chez SensAble. Dans le cadre de notre étude, nous nous baserons sur des hypothèses d'équilibre quasi-statique. Les forces haptiques calculées dans la simulation «HapCO» prennent en considération les lois de l'élasticité linéaire et du contact frottant (Signorini et Coulomb).

Avant de décrire l'architecture et le fonctionnement de «HapCo», nous présenterons le «Multiple Contact Friction Solver» qui est un solveur de forces de contact avec frottement entre objets déformables indépendant des modèles de déformation utilisés. Nous validerons ensuite ce modèle générique sur la plateforme logicielle «SOFA» développée par l'INRIA dont le but est de favoriser la recherche dans le domaine de la simulation temps réel et tout particulièrement pour des applications à caractère médical.

Enfin, nous évaluerons qualitativement le rendu haptique restitué (stabilité, fluidité et transition adhérence/glisserment) par des expériences réalisées avec un ensemble de personnes.

4.2 Solveur de force de contact générique

4.2.1 Principe algorithmique

La méthode de flexibilité est basée sur une formulation mixte des équations d'équilibre du système en déplacements élastiques et en forces de contact avec frottement. Son principe consiste à séparer la résolution des forces de contact des déplacements. Il s'agit d'une méthode facile à implémenter, elle permet aussi de tester plusieurs solveurs de forces de contact frottant, de tester leur robustesse (convergence) et leur stabilité numérique. La modularité de cette méthode permet également d'envisager du calcul parallèle afin d'améliorer le temps de calcul et de tester des problèmes de contact très raides avec retour haptique (à haute fréquence).

Elle est appelée méthode de «flexibilité» car elle construit, comme nous l'avons vu dans le chapitre 2, une matrice \mathbf{W} symétrique carrée de taille égale au nombre de contacts $(N_c) \times 3$ et définie positive appelée matrice de flexibilité. Cette dernière est exprimée en fonction des matrices de raideur inverses (matrices de flexibilité) des objets en contact et nous informe sur la raideur ou la flexibilité de la zone de contact. Le challenge dans l'obtention d'une simulation temps réel est de construire cette matrice d'une manière très optimisée en évitant au maximum les multiplications par zéro et de trouver des procédures rapides de calcul des forces de contact frottant.

La méthode de flexibilité consiste à projeter les équations en déplacements sur les repères locaux associés aux points de contact. Le résultat établit une relation linéaire entre les déplacements relatifs contraints et les forces de contact avec frottement via l'opérateur de Delassus ([Wri02], [DDKA05]) comme le montre l'équation 4.1 :

$$\begin{cases} \mathbf{x} = \mathbf{W} \mathbf{r} + \tilde{\mathbf{x}} \\ \mathbf{r}^\alpha = \text{Proj}_{\mathbf{K}_\mu}(\mathbf{r}^{*\alpha}) \end{cases} \quad (4.1)$$

Avec

$$\left\{ \begin{array}{l} \mathbf{W} = \mathbf{H}_1 \mathbf{K}_1^{-1} \mathbf{H}_1^T + \mathbf{H}_2 \mathbf{K}_2^{-1} \mathbf{H}_2^T \\ \mathbf{r}^T = (\mathbf{r}_1^T, \dots, \mathbf{r}_\alpha^T, \dots, \mathbf{r}_{N_c}^T) \\ \mathbf{r}_\alpha^T = (\mathbf{r}_n^\alpha, \mathbf{r}_{t_1}^\alpha, \mathbf{r}_{t_2}^\alpha) \end{array} \right. \quad (4.2)$$

Nous rappelons que \mathbf{x} et $\tilde{\mathbf{x}}$ représentent respectivement le vecteur des déplacements relatifs contraints en considérant les forces de contact et celui des déplacements relatifs libres c'est à dire en l'absence des forces de contact. \mathbf{r} et \mathbf{r}^* représentent respectivement les vecteurs des forces de contact réelles et augmentées de taille égale au $N_c \times 3$, \mathbf{H}_1 et \mathbf{H}_2 sont les matrices de transformation de l'espace global vers l'espace des contraintes liées aux points de contact. \mathbf{K}_1^{-1} et \mathbf{K}_2^{-1} sont les matrices de raideur inverses. Nous pouvons en déduire que la matrice de flexibilité telle qu'elle est calculée dans l'équation 4.1 dépend du modèle de déformation.

Dans le cas de la simulation HapCo, les objets élastiques sont modélisés par la méthode des éléments finis en élasticité linéaire, les matrices \mathbf{K}_1^{-1} et \mathbf{K}_2^{-1} sont alors accessibles et peuvent être calculées et inversées une seule fois «hors ligne» car elles sont constantes. Dans le cas général, ces matrices ne sont pas toujours accessibles et ne sont pas constantes. Elles ne sont pas accessibles lorsque les modèles de déformation des objets en interaction sont intégrés dans la simulation comme des «boîtes noires» ou bien lorsque la méthode de résolution des équations d'équilibre ne prévoit pas la construction d'une matrice de raideur. Elles ne sont pas constantes dans le cas de l'élasticité non linéaire et des grands déplacements, ce qui nécessite un calcul «en ligne» très coûteux.

Le solveur de contact tel qu'on l'a défini jusqu'à maintenant utilise la matrice de flexibilité \mathbf{W} afin d'évaluer le vecteur déplacement relatif contraint \mathbf{x} (équation 4.1). Il est donc dépendant du modèle de déformation des corps en contact car il nécessite la connaissance de \mathbf{K}_1^{-1} et \mathbf{K}_2^{-1} . Notre but est de présenter un solveur de forces de contact avec frottement sans construction de la matrice de Delassus afin qu'il soit indépendant des modèles de déformation utilisés dans la simulation. Pour d'atteindre cet objectif, deux questions doivent être posées :

- La première est comment réévaluer dans notre solveur (Uzawa global) le déplacement relatif contraint entre deux objets (impacteur/obstacle) au niveau des points de contact?
- La deuxième question concerne l'estimation de la raideur locale fictive ρ^α qui figure dans la formulation lagrangienne augmentée des forces de contact. En effet, cette raideur locale fictive est calculée à partir de la matrice de Delassus.

En réponse à la première question, nous proposons que le vecteur de déplacements relatifs \mathbf{x} entre les deux objets soit réévalué à chaque itération du solveur de contact à partir de la réactualisation des positions fournies par les solveurs de déplacements élastiques

associés à chaque objet, comme le montre l'équation 4.3. Nous rappelons que $(\vec{\mathbf{n}}^\alpha, \vec{\mathbf{t}}_1^\alpha, \vec{\mathbf{t}}_2^\alpha)$ représente le repère local au contact (P^α, Q^α) avec $\alpha = 1, N_c$, ces données sont fournies par le module de détection de collision (figure 4.1).

$$\mathbf{x} = \mathbf{W} \mathbf{r} + \tilde{\mathbf{x}} \Leftrightarrow \begin{cases} \mathbf{X}_1(P^\alpha) = \text{Appel Solveur Déplacement Solide 1 } (\mathbf{r}^\alpha) \\ \mathbf{X}_2(Q^\alpha) = \text{Appel Solveur Déplacement Solide 2 } (\mathbf{r}^\alpha) \\ \mathbf{x}_\alpha = \begin{pmatrix} \vec{\mathbf{n}}^\alpha \cdot (\mathbf{X}_1(P^\alpha) - \mathbf{X}_2(Q^\alpha)) \\ \vec{\mathbf{t}}_1^\alpha \cdot (\mathbf{X}_1(P^\alpha) - \mathbf{X}_2(Q^\alpha)) \\ \vec{\mathbf{t}}_2^\alpha \cdot (\mathbf{X}_1(P^\alpha) - \mathbf{X}_2(Q^\alpha)) \end{pmatrix} \\ \mathbf{x}^T = (\mathbf{x}_1^T, \dots, \mathbf{x}_\alpha^T, \dots, \mathbf{x}_{N_c}^T) \end{cases} \quad (4.3)$$

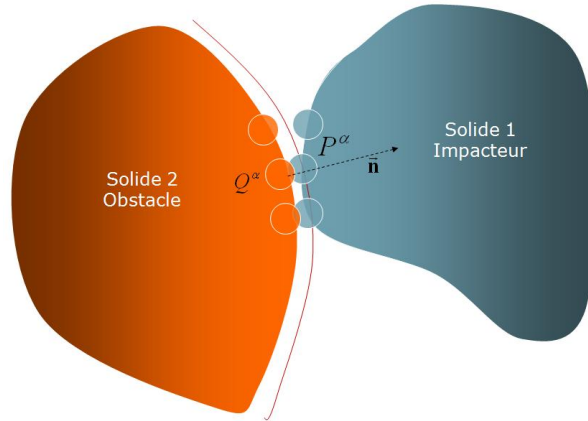


FIG. 4.1 – Détection de collision entre un objet impacteur et un objet obstacle.

En réponse à la deuxième question concernant le calcul de la raideur locale fictive, elle sera désormais estimée à partir des paramètres ρ_{S_1} et ρ_{S_2} liés respectivement aux objets en contact S_1 et S_2 . Une estimation robuste de ρ_{min} serait de considérer $\rho_{min} = \text{Inf}(\rho_{S_1}, \rho_{S_2})$. Cette estimation peut conduire à une valeur très faible de ρ_{min} risquant ainsi de ralentir la convergence du solveur des forces de contact. Dans ce cas, l'utilisateur pourrait augmenter la valeur de ρ_{min} au risque de créer des instabilités numériques.

La figure 4.2 schématise le principe de la nouvelle version de l'algorithme général de résolution des équations d'équilibre sans construction de l'opérateur de Delassus. Dans cet algorithme, il n'est plus nécessaire d'avoir accès à la modélisation des objets déformables permettant ainsi d'avoir un solveur de forces de contact générique. Cependant, appeler le solveur des déplacements pour calculer les nouvelles positions des contacts à chaque itération du solveur des forces de contact augmente les temps de calculs et peut mettre en péril l'objectif du temps réel.

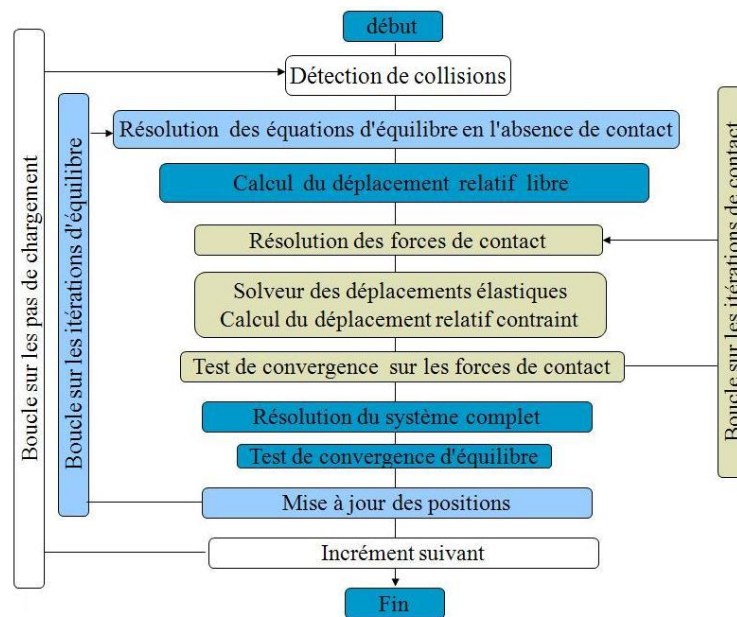


FIG. 4.2 – L’algorithme général de résolution des équations d’équilibre sans construction de l’opérateur de Delassus.

4.2.2 Expérimentations numériques

Nous avons intégré dans la simulation HapCo deux types de solveur de forces de contact, le premier appelé «solveur avec Delassus» calcule la matrice de Delassus \mathbf{W} en se basant sur l’équation 4.1. Le deuxième appelé «solveur sans Delassus» ne calcule pas la matrice de Delassus et fait appel au solveur des déplacements élastiques selon l’équation 4.3.

Dans l’algorithme de résolution des équations d’équilibre, on souhaite comparer les temps de calcul et les solutions obtenus avec les deux solveurs (avec et sans Delassus). On considère alors le banc d’essai présenté sur la figure 4.3. Le solveur de force de contact est basé sur l’algorithme Uzawa global en formulation bi-potentielle avec un coefficient de frottement égal à $\mu = 0.3$, et une erreur Gauss-Seidel égale à 10^{-4} . L’objet mobile est soumis à deux parcours, le premier dit de pénétration consiste à faire 5 pas selon la direction (-Y), 1 pas = 0.1 cm. Initialement les deux objets se touchent sans aucune interpénétration. Le second parcours dit de glissement consiste à faire 20 pas selon la direction (X). Initialement les deux objets sont déformés suite à un enfoncement de 2 cm. Ces tests ont été réalisés sur un PC portable Dell Latitude D830, possédant une RAM de 512 Mo et un processeur Intel(R) Coeur(TM) 2 Duo T7100 cadencé à 1.80 GHz.

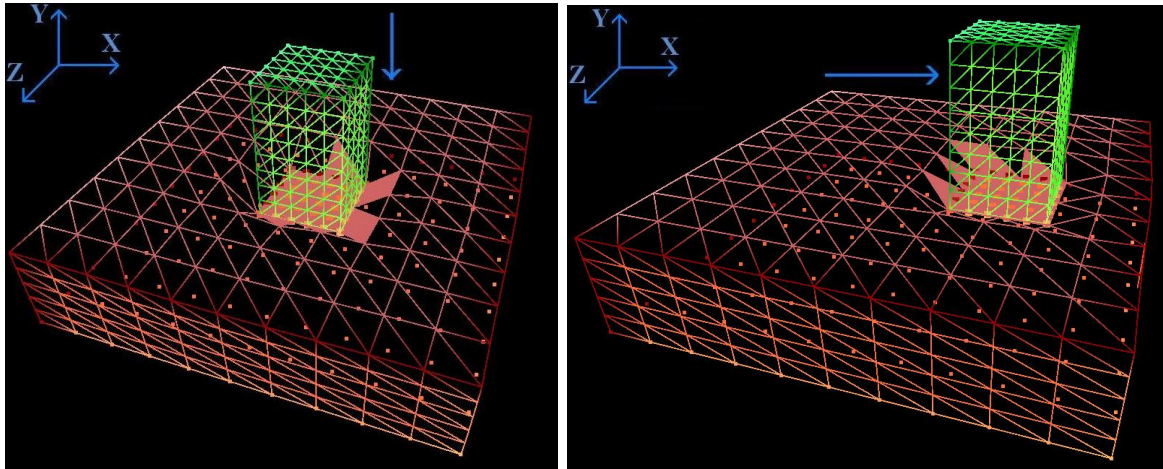


FIG. 4.3 – Banc d'essai impliquant 36 contacts, à gauche : parcours pénétration, à droite : parcours glissement.

L'objet en rouge encastré est composé de 726 noeuds, 500 éléments cubiques et de dimensions $L = 6.7$ cm, $H = 1.5$ cm et $l = 6.7$ cm. Le volume de chaque élément représente 0.2% du volume total. La déformation de l'objet vert est déterminée autour d'une configuration de référence réactualisée suivant un mouvement de corps rigide imposé (approche corotationnelle globale). Ce solide est composé de 324 noeuds, 200 éléments cubiques afin de donner 36 contacts. Ses dimensions sont $L = 1.425$ cm, $H = 2.4$ cm et $l = 1.425$ cm. Le volume de chaque élément représente 0.5% du volume total. Si l'on désire une zone de contact composée uniquement de 16 noeuds, on pourra considérer un solide de 176 noeuds et 90 éléments cubiques.

Afin de réaliser une comparaison objective, la valeur de la raideur locale ρ_{min} dans le solveur de force de contact «sans Delassus» est identique à la valeur calculée dans le solveur «avec Delassus». L'équation 4.4 mesure l'écart relatif en % entre le vecteur force de contact issu du solveur sans Delassus $\mathbf{r}_{\text{sans-W}}$ et celui du solveur avec Delassus $\mathbf{r}_{\text{avec-W}}$.

$$\text{Ecart (\%)} = 100 \times \frac{\|\mathbf{r}_{\text{sans-W}} - \mathbf{r}_{\text{avec-W}}\|}{\|\mathbf{r}_{\text{avec-W}}\|} \quad (4.4)$$

Nous allons dans ce qui suit comparer les temps de calcul et les solutions obtenues des deux solveurs. Le temps CPU du solveur sans Delassus inclut les appels des solveurs de déplacements des points de contact à chaque itération de l'algorithme Uzawa global. Pour le solveur avec Delassus, le temps CPU inclut le temps de construction de la matrice de flexibilité, le temps de calcul du vecteur de déplacement relatif libre $\tilde{\mathbf{x}}$ ainsi que le temps de résolution Uzawa global. Nous désignons par «calcul géométrique» le calcul effectué par la détection de collision afin de fournir les couples de contact et les repères attachés aux points de contact. Nous désignons par «calcul physique» tous les calculs nécessaires à rétablir l'équilibre du système, c'est à dire le calcul des forces de contact avec frottement

(Solveur R) et des déplacements élastiques (Solveur U). On désigne par «Temps R» et «Temps U» les temps de calcul respectifs des solveurs R et U.

Interaction aluminium/tissu mou

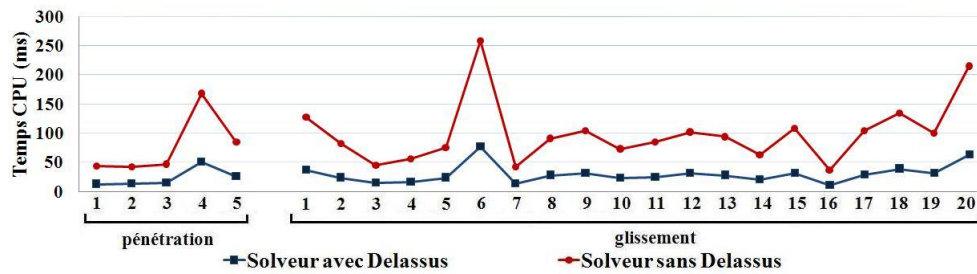


FIG. 4.4 – Comparaison en temps CPU entre le solveur de force de contact avec et sans Delassus (interaction aluminium/tissu mou).

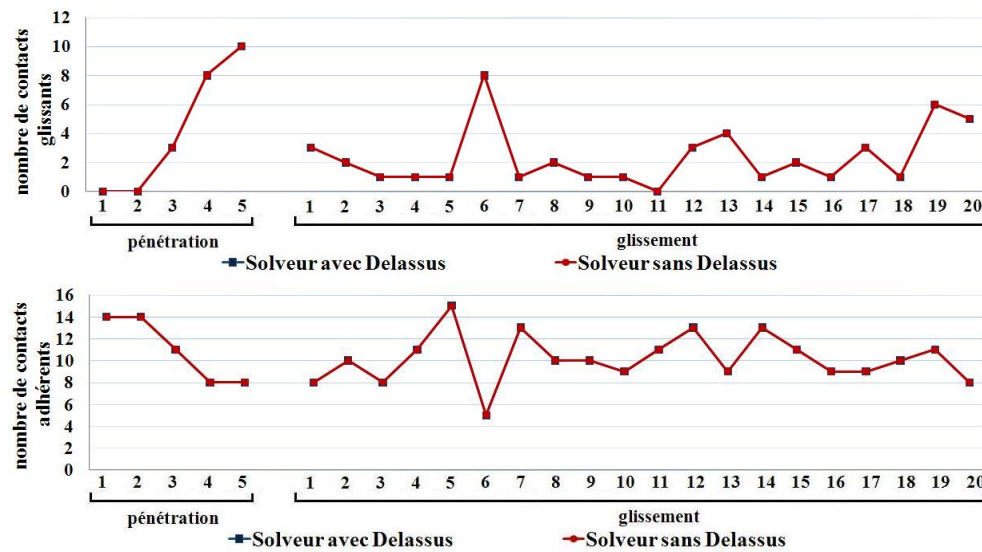


FIG. 4.5 – Comparaison en nombre de contacts glissants (en haut) et en nombre de contacts adhérents (en bas) issus des deux solveurs de forces de contact avec et sans Delassus (interaction aluminium/tissu mou).

4. HapCo : un prototype de simulateur temps réel avec retour haptique traitant des contacts multiples entre objets déformables.

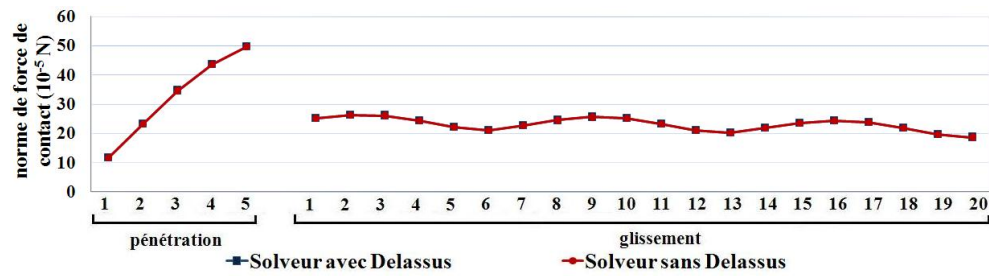


FIG. 4.6 – Norme du vecteur force de contact issue du solveur R avec et sans Delassus (interaction aluminium/tissu mou).

Interaction aluminium/caoutchouc

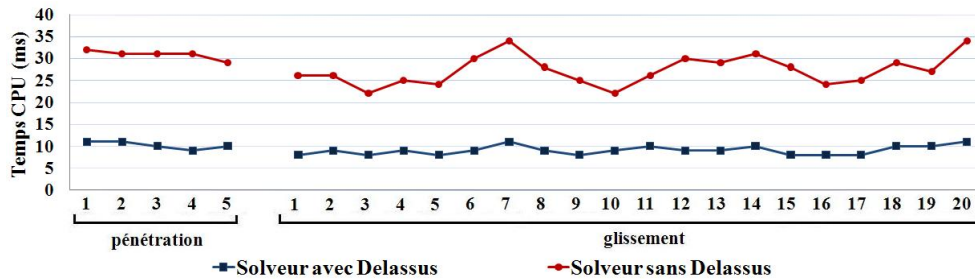


FIG. 4.7 – Comparaison en temps CPU entre le solveur de force de contact avec et sans Delassus (interaction aluminium/caoutchouc).

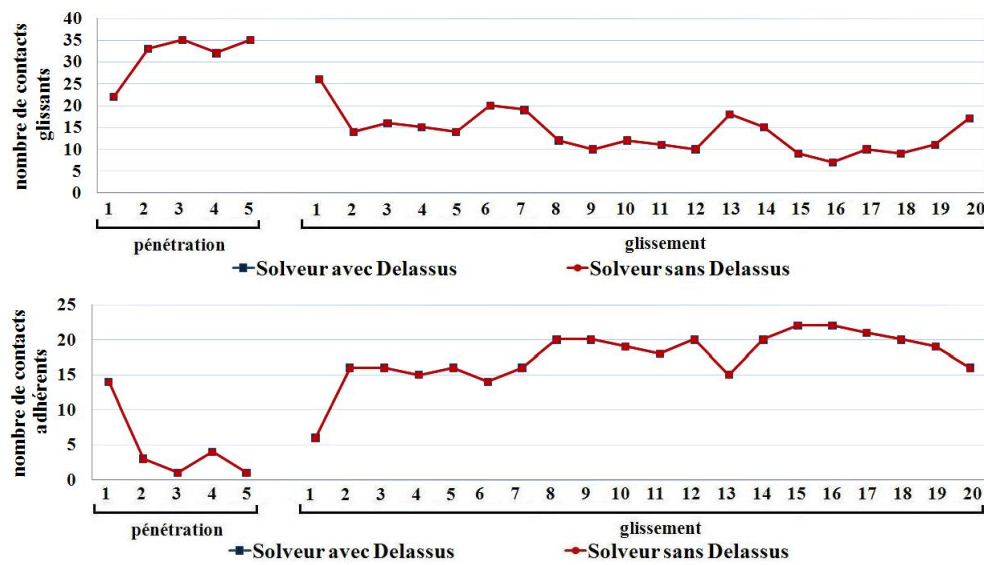


FIG. 4.8 – Comparaison en nombre de contacts glissants (en haut) et en nombre de contacts adhérents (en bas) issus des deux solveurs de forces de contact avec et sans Delassus (interaction aluminium/caoutchouc).

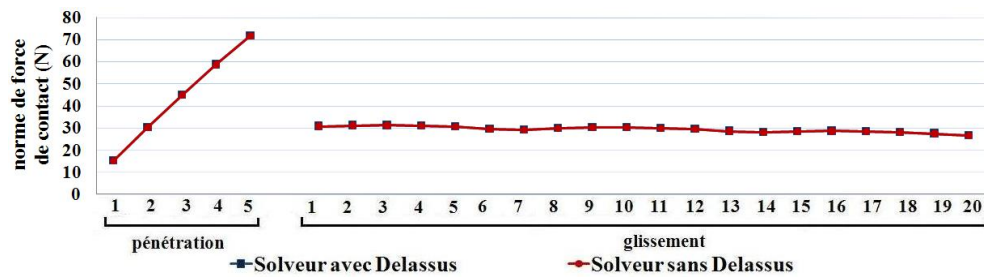


FIG. 4.9 – Norme du vecteur force de contact issue du solveur R avec et sans Delassus (interaction aluminium/caoutchouc).

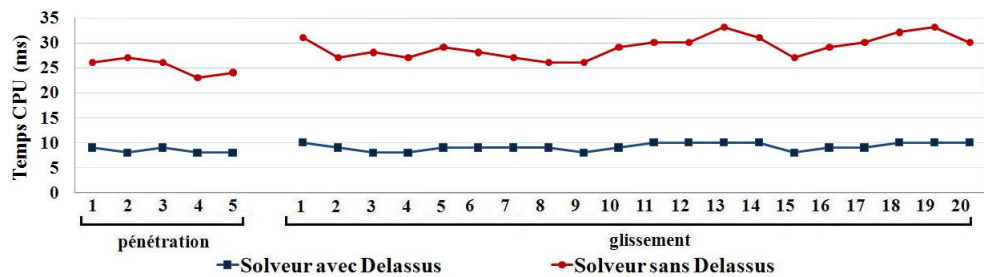


FIG. 4.10 – Comparaison en temps CPU entre le solveur de force de contact avec et sans Delassus (interaction tissu mou/tissu mou).

Interaction tissu mou/tissu mou

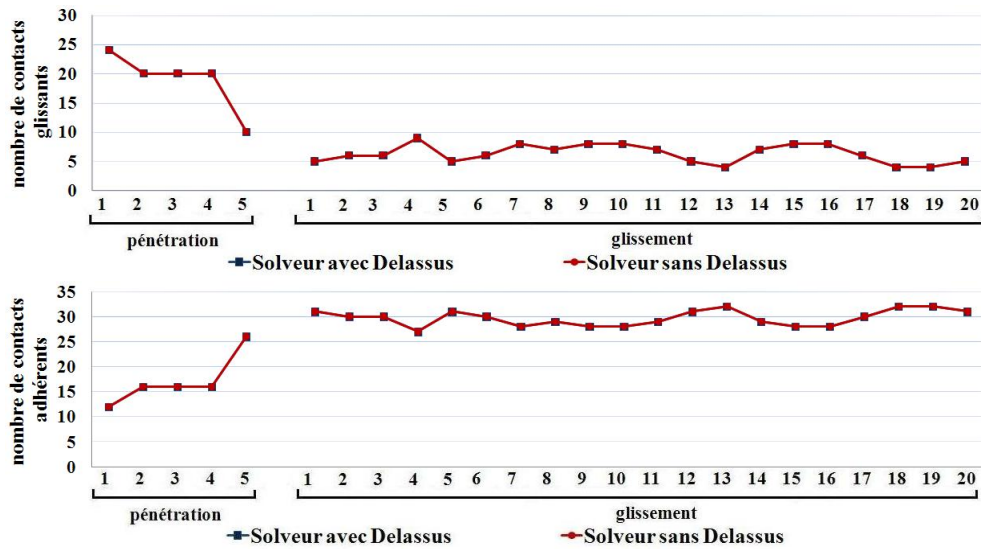


FIG. 4.11 – Comparaison en nombre de contacts glissants (en haut) et en nombre de contacts adhérents (en bas) issus des deux solveurs de forces de contact avec et sans Delassus (interaction tissu mou/tissu mou).

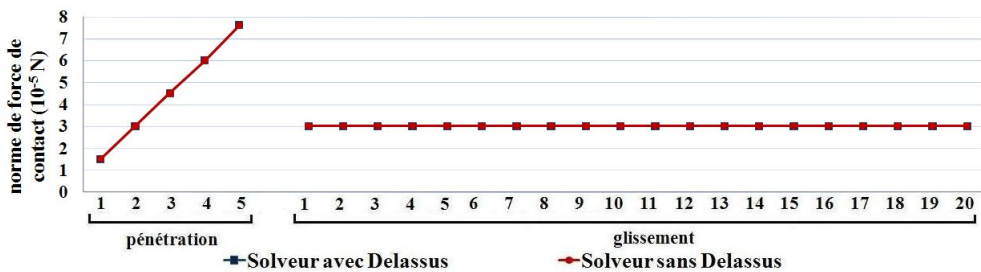


FIG. 4.12 – Norme du vecteur force de contact issue du solveur R avec et sans Delassus (interaction tissu mou/tissu mou).

Les graphes 4.4, 4.7 et 4.10 montrent les temps de calcul des deux solveurs. Les tableaux 4.13 et 4.14 présentent un récapitulatif des temps de calcul moyens des forces de contact (temps R), des temps de calcul moyens des déplacements élastiques (temps U) et enfin les temps de calcul moyens du calcul physique. Ces résultats montrent que le temps R du solveur avec Delassus est trois fois plus petit que le temps R du solveur sans Delassus. Le temps U, quand à lui, reste similaire.

Les graphes 4.5, 4.8 et 4.11 représentent respectivement le nombre de contacts glissants et adhérents issus des solveurs Uzawa global avec et sans Delassus dans le cas de

contact frottant aluminium/tissu-mou, aluminium/caoutchouc et tissu-mou/tissu-mou. Les graphes 4.6, 4.9 et 4.12 montrent les normes des forces de contact calculées pendant les parcours pénétration et glissement pour ces différents cas.

Comme on pouvait s'y attendre, on remarque qu'il n'y a aucune différence entre les solutions issues des deux solveurs. Nous observons le même nombre d'itérations Gauss-Seidel et les mêmes statuts des points de contact. De plus, nous ne constatons aucun écart entre $\mathbf{r}_{\text{sans-W}}$ et $\mathbf{r}_{\text{avec-W}}$. Nous en déduisons que les solveurs de contacts «avec Delassus» et «sans Delassus» convergent vers les mêmes solutions.

contact frottant	Calcul	36 contacts		
		Temps R	Temps U	Temps calcul physique
aluminium/tissu mou	avec Delassus	28.8	2.92	31.8
	sans Delassus	95.28	2.52	97.8
aluminium/caoutchouc	avec Delassus	9.28	2.88	12.16
	sans Delassus	28.88	2.76	31.64
tissu mou/tissu mou	avec Delassus	9.04	2.88	11.92
	sans Delassus	27.56	2.68	30.24

FIG. 4.13 – Temps CPU moyen (ms) des différents modules dans le cas du calcul avec et sans Delassus pour 36 contacts.

contact frottant	Calcul	16 contacts		
		Temps R	Temps U	Temps calcul physique
aluminium/tissu mou	avec Delassus	3.16	2.04	5.2
	sans Delassus	9.88	2	11.88
aluminium/caoutchouc	avec Delassus	2.16	2.12	4.24
	sans Delassus	6.4	2.08	8.48
tissu mou/tissu mou	avec Delassus	1.84	2.12	3.96
	sans Delassus	5.72	2.08	7.8

FIG. 4.14 – Temps CPU moyen (ms) des différents modules dans le cas du calcul avec et sans Delassus pour 16 contacts.

Dans le solveur avec Delassus, la construction de la matrice \mathbf{W} est optimisée. Etant carrée de taille $N_c \times 3$ et symétrique, nous ne calculons que sa partie triangulaire supérieure. Le déplacement relatif libre est calculé une seule fois avant de procéder au calcul des forces de contact.

Comme le montre l'équation 4.1, le calcul du déplacement relatif contraint se fait à l'aide d'un produit matrice/vecteur et d'une addition de deux vecteurs. Ceci revient à effectuer $(3 \times N_c)^2$ multiplications et additions par itération Gauss-Seidel dans la résolution Uzawa globale.

Le solveur de déplacement élastique pour mettre à jour les positions des noeuds dans la méthode de flexibilité n'est appelé qu'une seule fois après la résolution des forces de contact. Le solveur des forces de contact sans Delassus appelle quatre fois le solveur U par couple de contact (P^α, Q^α) dans chaque itération Gauss-Seidel. En effet, il faut mettre à jour les positions des points P^α mais aussi S_1^α , S_2^α et S_3^α points d'interpolation de Q^α .

Dans la simulation HapCo, le modèle de déformation correspond à un modèle élastique linéaire en élément finis avec des matrices de raideurs inverses constantes et calculées hors ligne. Le solveur U peut être réduit à un produit matrice/vecteur et une addition de deux vecteurs $\Delta U^\alpha = \mathbf{K}^{-1} (\mathbf{R} + \mathbf{F}_{\text{int}})$. Si le modèle de déformation utilisé est plus complexe, la mise à jours des contacts sera ralentie augmentant ainsi les temps de calcul du solveur de contact.

Dans la simulation HapCo, ce solveur de forces de contact sans Delassus sera nommé «Multiple Contact Friction Solver». Alors que ce dernier augmente le temps CPU de la résolution des forces de contact et nécessite une estimation du paramètre ρ_{\min} , il permet tout de même :

- D'être indépendant des modèles de déformation.
- De paralléliser les calculs.
- D'avoir un solveur de contact générique pouvant être intégré dans n'importe quelle plateforme de simulation.

4.3 SOFA: exemple d'intégration du solveur de contact frottant

«SOFA» pour «Simulation Open Framework Architecture»¹ est un projet collaboratif international (INRIA France, CIMIT USA, CSIRO Australie, Chine, Royaume-Uni et Allemagne), impliquant différentes équipes INRIA dont l'objectif est de développer une plateforme logicielle favorisant la recherche dans le domaine de la simulation temps réel et particulièrement pour les applications médicales. Les principaux objectifs du projet SOFA sont :

- Développer une librairie open-source pour la communauté de simulation médicale,
- Favoriser l'échange de modèles et d'algorithmes,
- Optimiser les temps de développement en évitant de réinventer la roue,
- Promouvoir les collaborations entre groupes de recherche,
- faciliter le transfert de technologie entre recherche et industrie,
- Standardiser certains éléments d'une simulation (modèles anatomiques par exemple),

1. SOFA: www.sofa-framework.org

- Permettre l'évaluation, la validation et la comparaison de nouveaux algorithmes.

Les objets simulés sur SOFA sont modélisés à l'aide de composants indépendants attachés à une structure de données arborescente appelée graphe de scène. Les scènes sont décrites en langage XML à l'aide de composants appartenant à des catégories telles que champs de force, masses ou solveurs différentiels. Il est facile d'utiliser plusieurs modèles synchronisés d'un même objet afin d'optimiser des tâches différentes comme le calcul de forces internes, la détection des collisions ou l'affichage. Cette souplesse permet de modéliser une grande variété d'objets. Des solveurs génériques d'équations différentielles sont utilisés pour piloter la simulation permettant à des objets régis par des lois de comportement différentes d'interagir. SOFA peut, par exemple, simuler les interactions de structures déformables avec des fluides ou des objets rigides (grandes raideurs, grand pas de temps ...). Même si de nombreuses propriétés restent à implémenter, la version actuelle de SOFA contient déjà de nombreux modèles de déformation (masses ressorts, éléments finis), des méthodes d'intégration numérique (intégrateurs implicites et explicites), des méthodes de détection et de réponse aux collisions, ainsi que leur implémentation sur GPU.

Notre solveur générique de forces de contact frottant a ainsi été testé sur la plateforme SOFA pour des interactions entre objets utilisant des modèles de déformation différents. SOFA intègre un algorithme de résolution des forces de contact sans frottement fondé sur la méthode de pénalité. Dans cette dernière, des ressorts sont créés entre deux objets lorsque la distance les séparant passe en dessous d'un seuil prédéfini. Une force de contact est alors calculée afin d'éviter toute interpénétration. Une collaboration entre l'équipe Evasion (INRIA Grenoble) et notre équipe RVH (IBISC Evry) a été initiée afin d'intégrer et tester dans SOFA notre solveur de forces de contact avec frottement.

La figure 4.15 nous montre un exemple de scène contenant un objet déformable composé de 3 noeuds et un objet rigide composé de 4 noeuds. Les degrés de liberté sont représentés sous forme de cercle et les forces sous forme de lignes. La plate forme SOFA permet une représentation multimodale des objets composant la scène comme le montre la figure 4.16. La scène peut aussi être représentée sous forme d'un graphe comme le montre la figure 4.15. Les noeuds de ce graphe sont représentés sous forme d'étoiles qui représentent des groupes structurés de composants. Quand une collision est détectée, un noeud «Interaction force» est ajouté au graphe. Celui-ci permet de calculer les forces de contact par une méthode de pénalité.

4. HapCo : un prototype de simulateur temps réel avec retour haptique traitant des contacts multiples entre objets déformables.

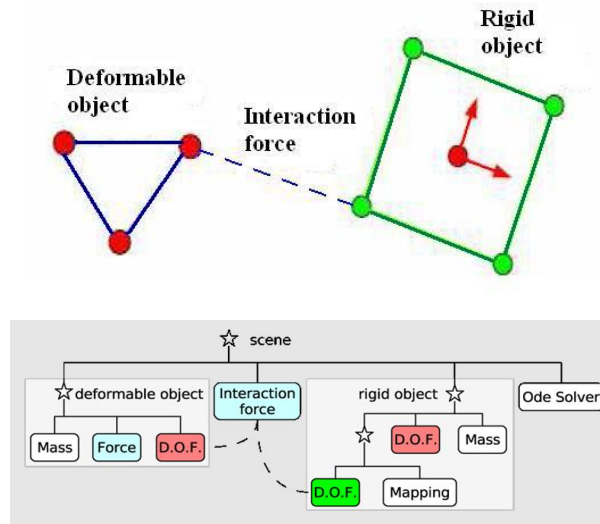


FIG. 4.15 – En haut, deux solides en interaction. En bas, la représentation dans SOFA de cette scène sous forme de graphe.

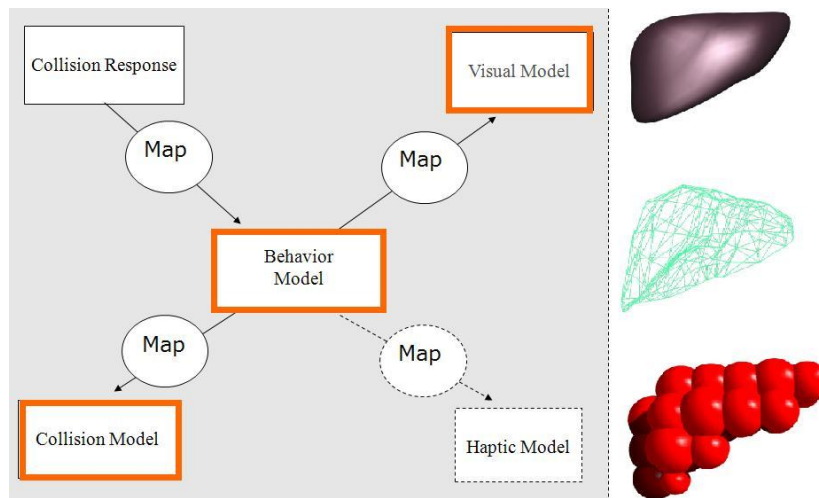


FIG. 4.16 – Représentation multimodale d'un objet dans la plate forme SOFA.

La figure 4.17 montre l'exemple étudié afin de tester le solveur «Multiple Contact Friction Solver». Il s'agit d'une barre modélisée en éléments finis tétraédriques et composée de 125 noeuds. Elle est encastree d'un côté avec 25 noeuds bloqués (conditions de Dirichlet) et soumise à son propre poids. Elle admet un module de Young égal à 200 Mpa et un coefficient de Poisson de 0.49 (très peu compressible). Cet objet vient en contact avec un foie composé de 181 noeuds (dont 3 bloqués) avec un module de Young égal à 5000 Mpa et un coefficient de Poisson de 0.3. Ce système est supposé en équilibre quasi-statique, l'accélération n'est donc pas considérée. Le coefficient de frottement entre les deux objets est de 0.3.

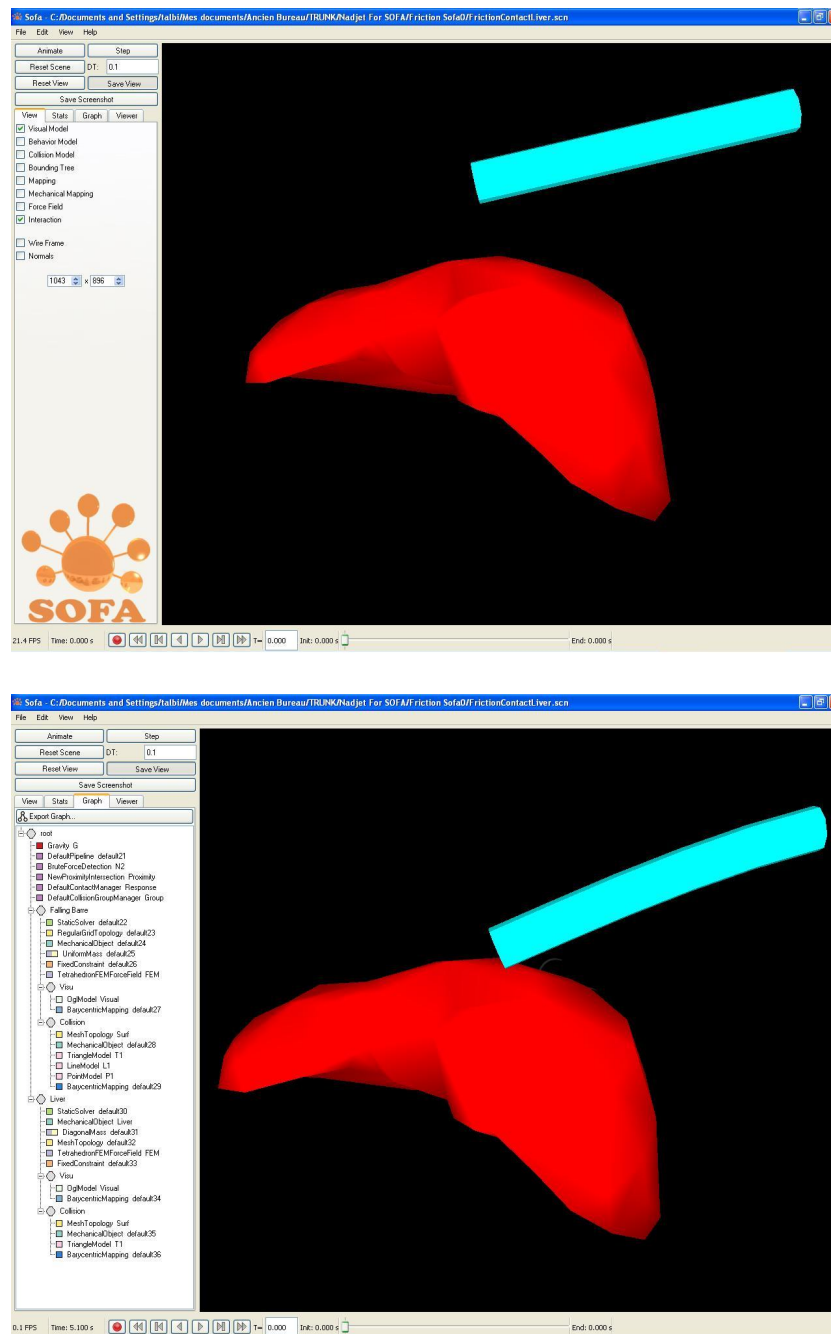


FIG. 4.17 – Contact frottant entre une barre soumise à son poids et un foie dans SOFA.

La figure 4.18 représente le graphe simplifié de cette expérience dans SOFA (figure 4.17). Notre solveur, baptisé dans SOFA «Static Friction Solver» est alors ajouté au graphe de la scène à la place du solveur «Interaction Force» afin de calculer les forces de contact frottant entre les deux objets. Le noeud « Static solver» représente le solveur de résolution du déplacement élastique. Basé sur la méthode du gradient conjugué, il permet de calculer

les nouvelles positions des noeuds de l'objet auquel il est rattaché. Dans cette expérience, ce même solveur est utilisé pour les deux objets. Nous aurions pu aussi considérer des solveurs de déplacements différents pour chacun des objets.

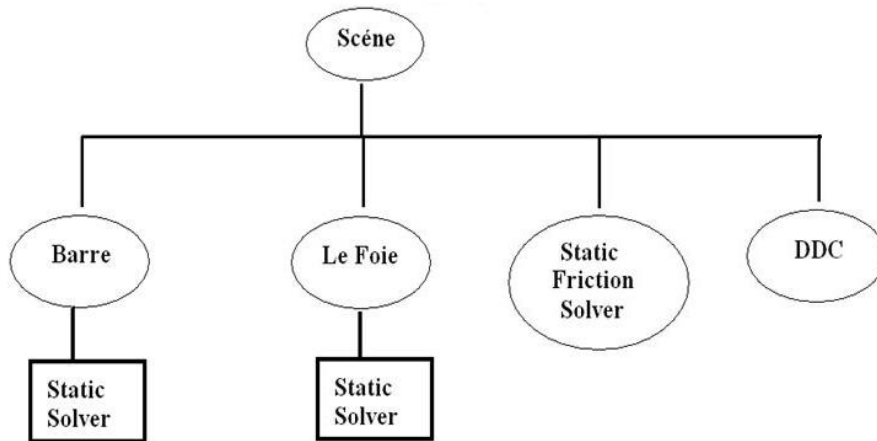


FIG. 4.18 – Le graphe SOFA associé à la scène barre foie .

L'algorithme 4.1 défini ci-après, décrit le contenu du solveur «Static Friction Solver» adapté à la plateforme logicielle SOFA. La valeur du paramètre ρ_{min} fournie par la simulation est égale au minimum des raideurs locales aux points de contact.

L'intégration du «Static Friction Solver» dans SOFA a mis en évidence les difficultés suivantes:

- Concernant le calcul des positions libres des noeuds, la plateforme de SOFA ne permettait pas d'appeler le solveur de déplacement avant le «static Friction solver» (c'est à dire en l'absence des forces de contact). Des modifications ont ainsi été apportées pour remédier à ce problème.
- De plus, l'appel du solveur de déplacement uniquement pour les points en contact n'est pas possible dans SOFA. Cela entraîne ainsi le calcul des positions de tous les noeuds du modèle mécanique de l'objet. Il est alors très coûteux en temps de calcul de faire appel au solveur U à chaque incrément de l'algorithme de résolution des forces de contact frottant.
- De ce fait, nous n'avons pas pu calculer dans la boucle globale d'Uzawa uniquement la position du point en cours de traitement. Comme le montre l'algorithme 4.1, les positions des points en contact sont mises à jour à la sortie de la boucle globale après le traitement successif et indépendant des points en contact (itération de Jacobi). En effet, le traitement d'un contact ne considère pas les dernières forces associées aux points de contact précédents, contrairement à la version de base d'Uzawa globale (itération de Gauss Seidel). Ces restrictions peuvent amener à converger moins rapidement (voir pas du tout) et éventuellement conduire à des instabilités numériques.

Algorithme 4.1 : Static Friction Solver

- 1: Force de contact frottant $\mathbf{r} = \mathbf{0}$, $Nb - iter = 0$
- 2: Appeler le static solver (gradient conjugué) pour chaque objet avec des forces de contact nulles.
- 3: **while** $\varepsilon_{Globale} > \varepsilon_{Gauss-Seidel}$ et $Nb - iter < Nb - iter - max$ **do**
- 4: $\mathbf{r}^{(k-1)} = \mathbf{r}$
- 5: $Nb - iter + = 1$
- 6: **for all** α $1 \leq \alpha \leq N_c$ **do**
- 7: Calcul du déplacement relatif contraint en ce contact \mathbf{x}^α

$$\mathbf{x}^\alpha = \begin{pmatrix} \vec{\mathbf{n}} \cdot (\mathbf{X}(\mathbf{P}^\alpha) - \mathbf{X}(\mathbf{Q}^\alpha)) \\ \vec{\mathbf{t}}_1 \cdot (\mathbf{X}(\mathbf{P}^\alpha) - \mathbf{X}(\mathbf{Q}^\alpha)) \\ \vec{\mathbf{t}}_2 \cdot (\mathbf{X}(\mathbf{P}^\alpha) - \mathbf{X}(\mathbf{Q}^\alpha)) \end{pmatrix}$$

- 8: Phase de prédiction-correction des forces de réaction selon Uzawa bi-potentiel
- 9: **end for**
- 10: // Calcul de l'erreur globale

$$\varepsilon_{Globale} = \frac{\|\mathbf{r} - \mathbf{r}^{(k-1)}\|}{\|\mathbf{r}\|}$$

- 11: Appeler le static solver (gradient conjugué) pour chaque objet en contact avec les forces de contact $\mathbf{R} = \mathbf{H}^T \mathbf{r}$
 - 12: **end while**
-

Malgré ces difficultés et le manque de temps (séjour de deux semaines), nous avons réussi à intégrer dans SOFA le solveur «Multiple Contact Friction Solver», basé sur l'itération de Jacobi. Nous avons pu le tester dans le cadre d'une interaction avec frottement entre une barre soumise à son poids et un foie. Il faudrait tout de même réaliser d'avantages d'essais et valider la méthode pour d'autres applications. Par ailleurs, il reste à modifier SOFA afin de pouvoir intégrer ce même solveur, cette fois ci basé sur l'itération Gauss-Seidel.

4.4 Description du simulateur HAPCO intégrant le retour d'effort

«HAPCO» pour «HAPTic COntact» est un simulateur interactif temps réel avec retour haptique. Il permet de résoudre des problèmes de multi-contacts frottants entre objets déformables. Ce premier prototype développé par notre équipe permet de tester et de comparer différents algorithmes de résolution de contact avec frottement entre objets déformables dans un contexte de simulation interactive avec retour d'effort. Les dispositifs haptiques considérés sont de type Phantom Desktop ou Phantom Omni de SensAble technologies (figure 4.19).



FIG. 4.19 – Le dispositif haptique PHANTOM Desktop à gauche et PHANTOM Omni à droite de SensAble technologies.

Dans notre simulateur, nous utilisons l'hypothèse d'équilibre quasi-statique. Aucune force inertielle due aux accélérations n'est alors considérée dans le déplacement ou la déformation des objets. Dans un contexte médical, cette hypothèse est justifiée car les gestes sont lents. Pour des raisons de temps de calcul, nous utilisons l'hypothèse des déformations linéaires. Les déformations au delà de 10% ne sont alors plus vraiment réalistes, ce qui est en désaccord avec le contexte chirurgical où celles ci peuvent aller au delà de 100%. Rappelons tout de même que HapCo a pour objectif de tester des algorithmes de contact. Ceux que nous avons développés peuvent tout aussi bien s'adapter à des modèles de déformation non linéaires (grandes déformations, grands déplacements).

4.4.1 Architecture et fonctionnement du simulateur HapCo

Le simulateur interactif HapCo avec retour d'effort, décrit sur la figure 4.20, est composé principalement de deux boucles. La première (processus de calcul) assure les calculs géométriques (DDC), physiques (rétablir l'équilibre) et le rendu visuel de la scène virtuelle. La seconde (processus haptique) mesure le mouvement du dispositif haptique couplé à la simulation (figure 4.19) et lui envoie les forces de contact avec frottement calculées afin qu'elles puissent être restituées à l'utilisateur (rendu haptique).

L'algorithme 4.2 décrit le module d'initialisation du simulateur. Cette phase permet de charger deux objets déformables en élasticité linéaire modélisés par des éléments tétraédriques ou cubiques. Ces deux objets sont affichés respectivement en vert pour l'objet impacteur (mobile) et en rouge pour l'objet obstacle (encastré). Le solide impacteur est manipulé par l'opérateur humain à travers le dispositif haptique.

Le processus haptique interroge régulièrement le dispositif à retour d'effort afin d'obtenir la position du stylet manipulé par l'utilisateur. Celle-ci est stockée en mémoire et rendue disponible à tout moment au processus de calcul. A chaque pas de chargement t , ce dernier vient lire cette position et en déduit le déplacement rigide par une approche corotationnelle globale (voir chapitre 1). Le module de détection de collision calcule alors la distance séparant les deux objets. Lorsque celle-ci passe en dessous d'un certain seuil (objets potentiellement en contact) ou devient négative (pénétration), une requête (incluant l'ensemble des couples de contact et les espaces de contraintes) est envoyée au solveur des forces (avec ou sans Delassus). Ce dernier sollicite tout d'abord le solveur des déplacements en l'absence du contact afin de calculer le mouvement libre puis détermine les forces de contact avec frottement annulant toute interpénétration (condition de Signorini) et satisfaisant les lois de frottement de Coulomb (adhérence / glissement). La force à retourner à l'interface haptique ($\mathbf{R}_{\text{Haptic}}$) est alors déduite du vecteur des forces de contact frottant ($\mathbf{R}_{\text{Haptic}} = \mathbf{H}^T \mathbf{r}$) et mise à disposition du processus haptique. Enfin, le solveur de déplacement calcule les positions nodales en considérant les forces de contact avec frottement comme des forces extérieures ou données permettant ainsi de rétablir l'équilibre du système.

La simulation HapCo se comporte en impédance car elle reçoit des positions et restitue des forces. L'interface haptique quand à elle, se comporte en admittance car elle accepte des forces et renvoie des positions. Le couplage entre le dispositif haptique et la simulation est décrit dans la figure 4.21. Le mouvement imposé par l'interface haptique est projeté de l'espace de travail haptique vers l'espace de travail virtuel. Si la force calculée est supérieure à la force maximale Max que l'interface haptique peut restituer, celle-ci est alors limitée à Max .

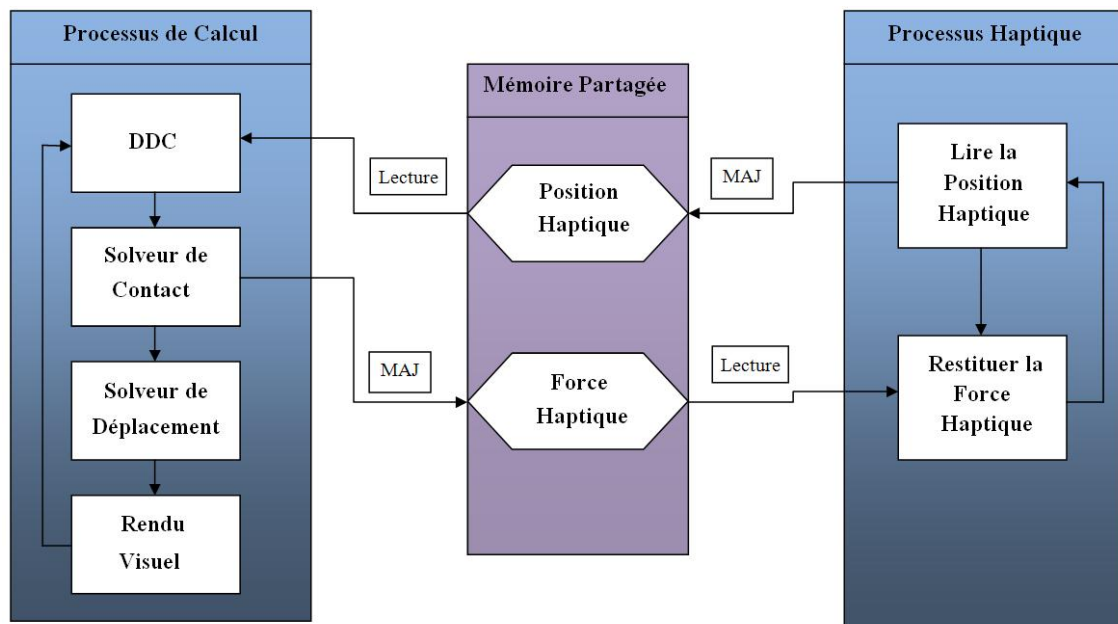


FIG. 4.20 – Architecture de la simulation HapCo.

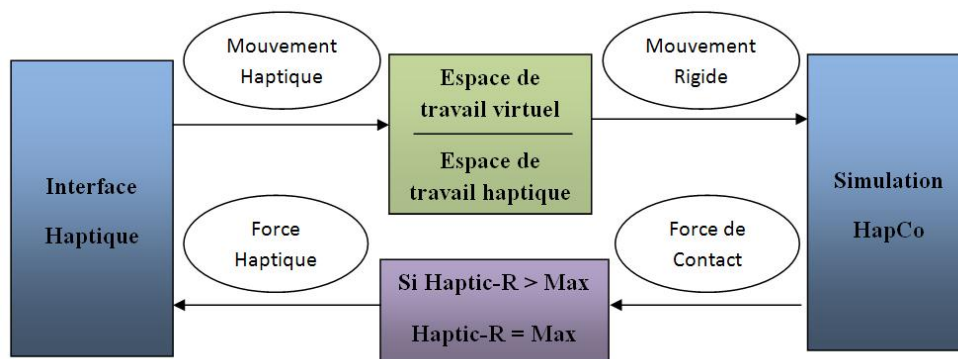


FIG. 4.21 – Couplage entre le dispositif haptique et la simulation

HapCo est basé sur les concepts d'une architecture orientée objet modulaire et évolutive. La classe principale «Solid» décrite par la figure 4.22 est la représentation informatique des objets déformables utilisés dans la simulation. Elle est composée de données membres (Propriétés mécaniques, modèle mécanique, modèle visuel ...) et de fonctions associées (Chargement, Calculs, Rendu ...).

Algorithme 4.2 : Module d'initialisation d'HapCo

- Initialiser les paramètres du simulateur HapCo : coefficient de frottement $\mu = 0.3$, erreur Gauss-Seidel $\varepsilon_{GS} = 10^{-4}$, ... etc.
- 2: Charger dans la simulation l'objet impacteur mobile **Sm**. Si nécessaire calculer sa matrice de raideur puis l'inverser.
Charger dans la simulation l'objet obstacle encastré **Sf**. Si nécessaire calculer sa matrice de raideur puis l'inverser.
 - 4: Initialiser le dispositif haptique.
Lancer le processus de calcul.
 - 6: Lancer le processus haptique.

Classe Solid
<p>E, ν module de Young et coefficient de Poisson</p> <p>L, K, K^{-1} matrices de connectivité, de raideur et de flexibilité</p> <p>X, U vecteur des positions nodales et des déplacements élastiques</p> <p>nbr_pnt, nbr_dll nombre des nœuds et nombre des degrés de liberté</p>
<p>Load (File),</p> <p>K (), K-inverse ()</p> <p>DDC (Solid S2),</p> <p>Flexibility Multiple Point Friction Solver (Solid S2),</p> <p>Multiple Point Friction Solver (Solid S2)</p> <p>Draw (mode, R, G, B)</p>

FIG. 4.22 – la classe Solid du simulateur HapCo.

La figure 4.23 présente le contenu d'un fichier décrivant le modèle mécanique de l'objet élastique défini dans la simulation HapCo. Ce fichier est le plus souvent issu d'un logiciel de discrétisation de structure en éléments finis type «ANSYS» ou d'autres logiciels standards de CAD. Ce fichier est chargé dans la simulation à l'aide de la fonction membre «Load (FILE)» de la classe Solid. Il doit nécessairement contenir les propriétés mécaniques du matériau, les coordonnées 3D des noeuds du maillage volumique de l'objet et la liste des éléments finis. En option, il peut aussi fournir la liste des triangles du modèle visuel, les conditions de Dirichlet (liste des noeuds bloqués) et la matrice de raideur (et son inverse). Dans le cas contraire, le modèle visuel est automatiquement calculé à partir du modèle mécanique, les conditions de Dirichlet sont imposées d'une manière arbitraire et la matrice de raideur ainsi que son inverse sont construites lors de l'initialisation de l'objet.

Fichier descriptif du modèle mécanique de l'objet déformable
Propriétés mécaniques E, ν
Coordonnées des nœuds du maillage volumique
Liste des triangles du modèle visuel
Table de connectivité (liste des éléments finis)
Conditions de Dirichlet
Matrice de raideur K , matrice de flexibilité K^{-1} de l'objet déformable

FIG. 4.23 – Fichier descriptif de l'objet déformable décrit par la méthode des éléments finis en élasticité linéaire.

Dans le cadre d'une simulation temps réel, une fréquence minimale de 25 images par seconde est nécessaire pour garantir une bonne interactivité visuelle. En ce qui concerne le rendu haptique, la fréquence de rafraîchissement des forces transmises à l'utilisateur doit être supérieure à 300 Hz. Pour cela, il est nécessaire que les modules détection de collision, solveur de déplacements et solveur des forces de contacts soient optimisés (en rouge sur la figure 4.20).

4.4.2 Interfaces graphiques du simulateur HapCo

HapCo a été développé sous l'environnement «Microsoft Visual C++ 6.0» et utilise les bibliothèques GLUT et GLUI d'OpenGL. La bibliothèque GLUT pour OpenGL Utility Toolkit a été développée par Mark Kilgard ([Kil96], [KF03]) lorsqu'il travaillait pour Silicon Graphics dans le but de produire du code portable entre les différents systèmes d'exploitations. Sa simplicité permet de ne pas se préoccuper de la partie système, relativement complexe, et de se concentrer sur le fond du programme OpenGL lui-même. Elle offre des fonctions de base d'un gestionnaire de fenêtre, et propose des routines de base de rendu graphique 3D. GLUI pour OpenGL User Interface Library lancé par Paul Rademacher en 1998 est une bibliothèque en C++ qui se combine avec GLUT et qui fournit des routines pour créer une interface graphique entièrement avec OpenGL. Cette dernière est alors indépendante du système d'exploitation et du gestionnaire de fenêtres. Elle vient compléter la bibliothèque de gestion de fenêtres OpenGL (GLUT) en apportant divers éléments de contrôle supplémentaires tels que les boutons, les cases à cocher, les zones de texte, les listes déroulantes, etc.

Les supports Ghost SDK pour le PHANTOM Desktop et hd/hdud pour le PHANTOM Omni permettent de coupler la simulation avec l'interface haptique. Ces bibliothèques permettent d'obtenir la position dans l'espace cartésien du stylet manipulé par l'utilisateur. Elles permettent aussi à la simulation de fournir à l'interface haptique les forces que l'on

souhaite faire ressentir à l'utilisateur.

Comme le montre la figure 4.24, le simulateur HapCo est composé de deux fenêtres graphiques. La première est dédiée à l'affichage 3D de la scène virtuelle et à la visualisation des interactions entre objets. La deuxième est utilisée pour configurer la simulation : mode de visualisation (filaire, ...), paramètres mécaniques (module de Young et coefficient de Poisson), coefficient de frottement, erreurs numériques, raideur locale ρ_{min} , norme de la force haptique, Elle permet en outre de réinitialiser le point de vue, les objets et les paramètres de la simulation. Nous allons maintenant décrire plus en détails les différents contrôles présents sur cette interface.

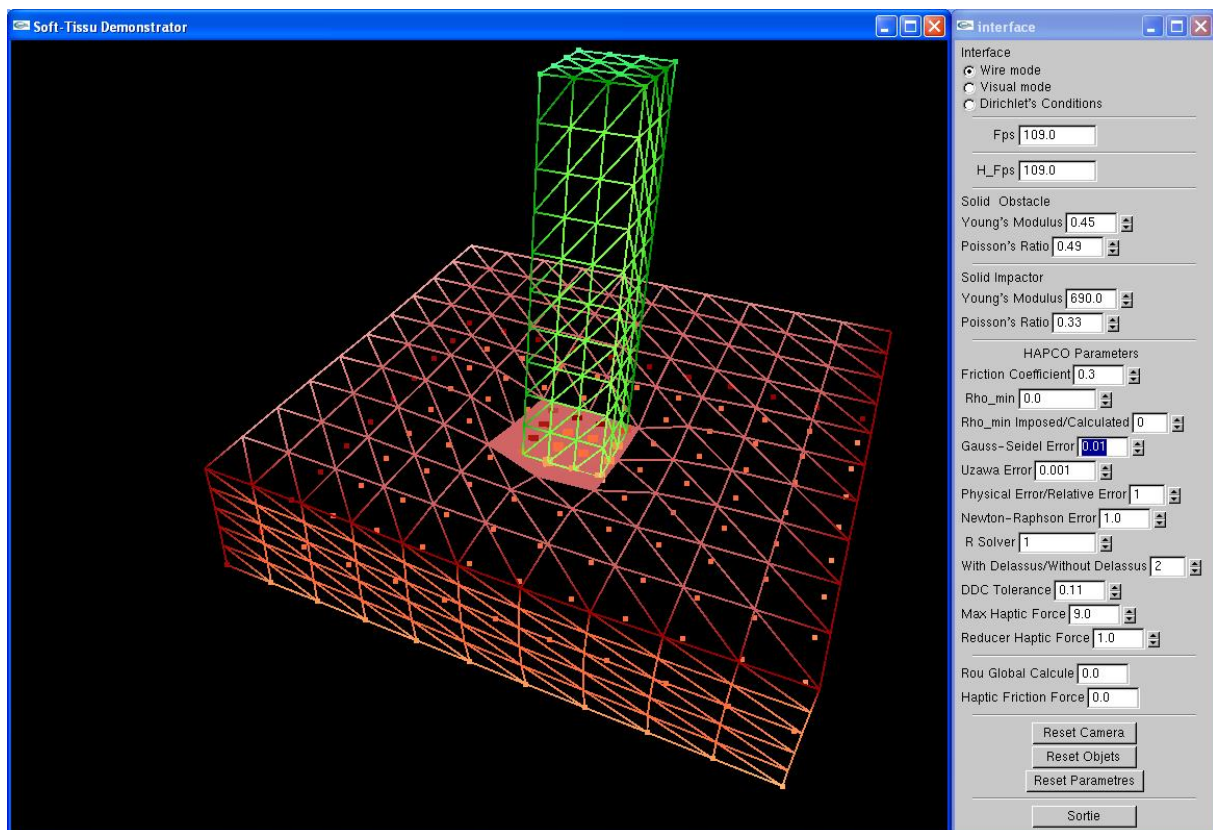


FIG. 4.24 – La scène virtuelle 3D et l'interface des paramètres.

- «Fps» et «H_Fps» : Ces valeurs représentent respectivement la fréquence d'appel du processus de calcul et du processus haptique.
- «Friction Coefficient» : Ce paramètre, représentatif du frottement de Coulomb, peut être changé au cours de la simulation afin de modifier le ressenti des phénomènes d'adhérence/glissement via le rendu haptique.

- « ρ_{min} » : Ce contrôle permet d'imposer une valeur à ρ_{min} . Si cette valeur est très petite par rapport à celle calculée à partir de la matrice de Delassus, la convergence est ralentie. Par contre, si cette valeur est très grande, des instabilités numériques peuvent apparaître.
- « ρ_{min} Calculated/Imposed» : Quand ce paramètre vaut 0 ρ_{min} est calculé à partir de la matrice de Delassus. Quand il vaut 1, ρ_{min} est imposé et sa valeur est donnée par le contrôle « ρ_{min} ».
- «Gauss-Seidel Error» : Ce paramètre permet de contrôler la boucle globale de résolution des forces de contact. Elle peut représenter une erreur relative sur les forces haptiques calculées ou bien une tolérance physique d'interpénétration.
- «Uzawa Error» : Elle est utilisée principalement par le solveur Uzawa locale afin de contrôler la boucle de résolution locale au point de contact. Elle peut être associée à une tolérance physique d'interpénétration au niveau de chaque contact ou bien à une erreur relative.
- «Physical Error/Relative Error» : Sa valeur permet de déterminer la formule de calcul de l'erreur globale. Quand elle est égale à 1, nous considérons une erreur relative $\varepsilon_{Globale} = \frac{\|\mathbf{r}-\mathbf{r}^{(k-1)}\|}{\|\mathbf{r}\|}$. Quand sa valeur est égale à 0, nous considérons une erreur physique d'interpénétration de type $\varepsilon_{Globale} = \frac{\|\mathbf{r}-\mathbf{r}^{(k-1)}\|}{\rho_{min}}$. Nous pouvons aussi envisager d'autres formules de calcul de l'erreur globale (non changement de statut par exemple) dans le but de tester l'influence de l'erreur globale sur la force haptique ressentie.
- «Newton-Raphson Error» : Elle permet de contrôler la boucle de résolution Newton-Raphson des équations d'équilibre du système.
- «DDC Tolerance» : C'est la distance maximale séparant l'objet impacteur de l'objet obstacle en deça de laquelle on considère qu'il y a collision.
- «Haptic Force» : Elle représente en temps réel la norme de la force haptique (N) transmise à l'interface haptique.
- «Max Haptic Force» : C'est la force haptique maximale (N) que l'interface haptique peut transmettre à l'utilisateur.
- «Reducer Haptic Force» : Ce paramètre est un coefficient réducteur de la force transmise à l'interface haptique afin d'éviter les effets de rebonds.

4.4.3 Expérimentations haptiques

HapCo est un simulateur qui combine interaction visuelle et retour haptique afin d'offrir à l'utilisateur une interaction plus riche et une plus grande immersion. Le rendu temps réel haptique devant être plus soutenu que le rendu visuel, il est nécessaire de réduire au maximum les temps de calcul. Plus, l'interaction est rigide, plus la fréquence de rendu doit être élevée: l'idéal étant d'avoir une fréquence de rendu haptique comprise entre 300 Hz et 1 KHz. Plusieurs critères participent alors à l'obtention d'une simulation temps réel ayant un bon rendu haptique :

- Le temps de la détection des collisions entre l'objet manipulé et les autres objets de la scène: Ce temps reste tout de même trop important. De multiples algorithmes de détection de collision ou de proximité ont été mis au point afin de réduire au maximum ce temps (hiérarchies englobantes, utilisation optimisée du processeur graphique GPU ...). La méthode que nous utiliserons dans notre étude est basée sur la recherche de couples potentiellement en contact. Afin de ne pas trop ralentir notre simulation, nous avons réduit partiellement les zones de contact entre les objets en éventuelle interaction, divisant ainsi par trois les temps de calcul. Nous avons atteint un temps moyen pour la détection de collision de l'ordre 5.5 ms dans le cas de 16 points de contacts et de 10 ms pour 36. La zone de contact de l'objet obstacle est composée de 180 triangles.
- Le temps de calcul du moteur physique (solveur des forces de contact et solveur de déplacement) : les différents modules utilisés pour le calcul physique ont été optimisés au niveau des algorithmes (éviter les multiplications par zéro, fonctions inlines, matrices construite hors ligne,...). Bien sur, la complexité des maillages des objets en interaction et le nombre de contacts influencent aussi très largement ces temps.
- Le compromis temps-précision : comme nous l'avons vu dans la section 4.4.2, la simulation «HapCo» intègre des critères portant sur la précision numérique. Dès lors se pose une question importante, à laquelle nous tenterons de répondre par le biais de nos évaluation : la précision numérique influe-t-elle sur le ressenti haptique de l'utilisateur?
- La transparence de l'interface haptique [Law93] : il existe un compromis entre la rigidité simulée et la stabilité de l'interaction. En considérant le caractère discret de la simulation, si les forces sont trop importantes ou renvoyées au mauvais moment, des oscillations peuvent apparaître. Cette instabilité peut, par exemple, être résolue partiellement en utilisant une interface proposant de meilleures caractéristiques (plage de forces plus importante). Une autre solution, présentée dans [AH99], montre l'analogie que l'on peut faire entre systèmes haptiques et électriques et propose un couplage bilatéral entre l'interface haptique et la simulation. Des facteurs supplémentaires d'amortissement et de rigidité permettent ainsi de contenir les surplus

d'énergie. Il est par ailleurs important de noter que l'utilisateur, dans son interaction avec la simulation, ne doit pas être perturbé par les imperfections mécaniques du dispositif à retour d'effort (équilibrage, poids, frottements, inerties ...). Il est alors primordial de pallier à ces éventuels défauts en ajoutant, par exemple, des forces de compensation.

Les interfaces haptiques dont nous disposons acceptent une force haptique maximale de 7.9 N pour le Phantom Desktop et 3.3 N pour l'Omni ce qui nous a contraint à réaliser des expérimentations entre objets de type aluminium/tissu mou ou tissu mou/tissu mou générant des forces de contact dans la plage des forces admises par nos interfaces. Les interactions entre objets plus rigides comme, par exemple aluminium/caoutchouc, génèrent, quand à elles, des forces dépassant les 7 N.

Les expérimentations haptiques ont été réalisées sur un PC équipé d'un processeur Intel(R) Xeon(R) 5110 double coeur cadencé à une fréquence de 1.60 GHz et de 2.0 Go de RAM. Des processeurs plus récents tels les E4300 cadencés à 3Ghz permettraient sans doute d'obtenir des temps de calcul plus petits et de tester la simulation avec des maillages plus complexes.

Nous avons opté pour l'interface «Phantom Desktop» pour réaliser les expérimentations haptiques, car il offre une plus large plage de forces admises. Cette interface est d'abord présentée à l'utilisateur. A chaque étape de l'expérience haptique, l'utilisateur note qualitativement le ressenti qu'il a, du rendu haptique. L'objectif étant ici de qualifier ce ressenti en fonction des différents paramètres figurant dans le tableau 4.25 :

Paramètres	Valeurs Testées
Erreur de Newton-Raphson ε_{NR}	$\varepsilon_{NR} = 10^{-3}$ $\varepsilon_{NR} = 1$
Erreur de Gauss-Seidel ε_{GS}	$\varepsilon_{GS} = 10^{-8}$ $\varepsilon_{GS} = 10^{-4}$ $\varepsilon_{GS} = 10^{-2}$
Coefficient de frottement μ	$\mu = 0.3$ $\mu = 0.7$
Interaction impacteur/obstacle	Aluminium/Tissu mou ($E = 45$ Mpa) Aluminium/Tissu mou ($E = 90$ Mpa) Caoutchouc ($E = 800$ Mpa)/Tissu mou ($E = 90$ Mpa)
Nombre de contacts N_c	16 36

FIG. 4.25 – Tableau récapitulatif des différents critères et leurs valeurs, testés dans les expérimentations haptiques .

Nous avons souhaité faire ressentir à des utilisateurs les forces de contact issues de l'interaction aluminium/tissu mou dans la simulation «HapCo» (figure 4.24). Nous avons

considéré deux «tissus mous» avec deux modules de Young différents ($E = 45$ Mpa pour le premier et $E = 90$ Mpa pour le second) et un coefficient de Poisson égal à 0.49. L'objet en aluminium possède un module de Young égal à $E = 69000$ Mpa.

Ces expérimentations haptiques entre un objet en aluminium et un tissu mou sont représentatives d'une interaction objet rigide/objet déformable. Dans la littérature, les simulations d'interaction rigide/déformable considèrent un corps solide indéformable à 6 degrés de liberté simplifiant ainsi les modèles de résolution utilisés. Tout au long de notre étude, nous avons proposés des algorithmes solutions permettant de traiter le contact entre deux objets déformables discrétisés de matériaux quelconques. Nous avons donc souhaité réalisé d'autres expérimentations mais cette fois ci entre deux objets qui se déforment. Alors que l'objet obstacle est toujours un «tissu mou» (module de Young égal à 0.9 Mpa) l'objet manipulé par l'utilisateur est maintenant en caoutchouc (module de Young égal à 800 Mpa).

L'erreur de Newton-Raphson permet de contrôler la résolution des équations d'équilibre du système. En hypothèse d'équilibre quasi-statique, la résolution Newton-Raphson réalise deux itérations pour $\varepsilon_{NR} = 10^{-3}$. La deuxième itération permet de vérifier que l'équilibre du système est bien rétabli. Ceci a été constaté dans le chapitre 3 où le solveur R recalcule à chaque itération de Newton-Raphson les mêmes forces de contact. Considérer une erreur de Newton-Raphson égale à $\varepsilon_{NR} = 1$, implique une seule itération Newton-Raphson (un seul appel du solveur des forces de contact).

L'erreur de Gauss-Seidel permet de contrôler la résolution Uzawa globale dans le solveur des forces de contact avec frottement. Considérer $\varepsilon_{GS} = 10^{-8}$, une valeur très faible, implique une précision numérique sur les forces de contact calculées. Afin de connaître l'influence de cette précision sur le rendu haptique, nous avons imposé à ε_{GS} des valeurs plus importantes (moins de précision).

Considérer différentes valeurs du coefficient μ , permet d'étudier l'influence de ce paramètre de frottement de Coulomb sur le rendu haptique. De plus, ça permet de tester l'efficacité des algorithmes de résolution du contact avec frottement entre deux objets déformables (transition adhérence/glissement).

Considérer une zone de contact composée de 16 et 36 points, permet d'étudier l'influence du nombre de contacts sur la qualité du ressenti haptique. Il permet aussi d'étudier l'influence du maillage des objets discrétisés en interaction sur ce retour d'effort.

L'utilisateur déplace via l'interface haptique l'objet impacteur de dimensions $L = 1$ cm, $H = 4$ cm et $l = 1$ cm. Il teste dans un premier temps le ressenti sur un parcours de «pénétration» puis sur un parcours de «glissement». Pour chacun des parcours, il évaluera le ressenti haptique pour des zones de contacts composées de 16 ou de 36 points. Nous allons dans ce qui suit décrire les différentes étapes de l'expérimentation haptique :

- Etape1: Initialiser les valeurs des paramètres suivants $\varepsilon_{GS} = 10^{-8}$, $\varepsilon_{NR} = 10^{-3}$, coefficient de frottement $\mu = 0.3$.

- Etape2 : Augmenter la valeur de ε_{NR} à 1.
- Etape3 : Augmenter la valeur de ε_{GS} à 10^{-4} ($\mu = 0.3$)
- Etape4 : Augmenter le coefficient de frottement à $\mu = 0.7$ ($\varepsilon_{GS} = 10^{-4}$)
- Etape5 : Considérer l'erreur ε_{GS} égale à 10^{-2} et le coefficient de frottement μ à 0.3
- Etape6 : Augmenter le coefficient de frottement à $\mu = 0.7$ ($\varepsilon_{GS} = 10^{-2}$)

Sept personnes ont participé à ces expérimentations. Nous avons pour chacune relevé leurs appréciations sur le ressenti haptique.

Le rendu haptique de la première étape a été jugé par tous instable et saccadé. Ceci s'explique par le fait que ε_{GS} est très petite, et la résolution Newton-Raphson itère deux fois (deux appels au solveur R). Avec de tels précisions numériques, les temps de calcul sont grands et ne permettent pas d'atteindre une fréquence de rendu haptique correct (en dessous de 10 Hz). Ceci explique le ressenti haptique saccadé.

Pour la seconde étape, augmenter la valeur de ε_{NR} permet d'effectuer qu'une seule itération Newton-Raphson. Le temps de détection de collision quant à lui reste constant. Le temps de calcul n'est cependant pas encore suffisamment réduit pour avoir un bon rendu haptique.

A l'étape 3, quand la valeur de ε_{GS} augmente à 10^{-4} , tous ont constaté une amélioration par rapport aux deux étapes précédentes. Cependant, des oscillations haptiques étaient toujours ressenties dans le cas d'une zone de contact avec 36 points. En effet, les temps de calcul obtenus restent critiques pour une simulation avec retour d'effort correct (fréquence haptique en dessous de 50 Hz).

A l'étape 5 où ε_{GS} prend la valeur 10^{-2} , chaque personne a estimé le rendu haptique beaucoup plus souple et stable. Le ressenti a été jugé plus agréable et naturel. En outre, le contact a été jugé légèrement moins rigide que précédemment. Ceci s'explique par le fait que le flux des forces de contact obtenu avec $\varepsilon_{GS} = 10^{-2}$ permet à l'utilisateur de suivre plus facilement la force haptique, créant ainsi cette sensation d'un contact plus mou. Notons qu'en terme de temps de calcul, considérer une erreur de Gauss-Seidel 100 fois plus grande permet d'accélérer la convergence du solveur de force de contact d'un facteur d'ordre 2.

Lors des étapes 4 et 6, pendant le parcours de glissement (avec légère pénétration), la modification du coefficient de frottement a été nettement ressentie. La différence a tout de même été mieux perçue dans le cas d'interaction aluminium/tissu mou avec $E = 90$ Mpa (forces haptiques de l'ordre de 1.5 N) que pour $E = 45$ Mpa (forces haptiques de l'ordre de 0.5 N). Toutefois en parcours de glissement, on constate des phénomènes d'accrochage dus à des déformations en cuvette importantes de la zone de contact qui nécessiterait une discrétisation en éléments finis plus fine.

Les personnes ont pu constater à travers le retour d'effort que l'interaction aluminium/tissu mou avec $E = 90$ Mpa était plus rigide qu'avec $E = 45$ Mpa. Les forces

pouvaient atteindre dans le premier cas les valeurs 4.5 et 7 N pour les coefficients de frottement respectifs 0.3 et 0.7. Dans le second cas, les forces pouvaient atteindre les valeurs 3 et 4.5 N pour les mêmes coefficients de frottement (0.3 et 0.7).

Diminuer le nombre de points en contact de 36 à 16, a créé une sensation de contact moins rigide. En effet, même si les forces calculées ne sont pas les mêmes (variation de l'ordre de 0.3 N), c'est plutôt l'augmentation du flux des forces haptiques restituées qui donne à l'utilisateur l'impression d'un contact plus mou (le solveur des forces de contact devient plus rapide).

Ces observations peuvent être retrouvées sur les figures 4.26 et 4.27. Elles montrent les normes des forces haptiques calculées dans le cas d'interaction aluminium/tissu mou ($E = 45$ Mpa) pendant un parcours de pénétration avec respectivement 36 et 16 contacts (figure 4.24).

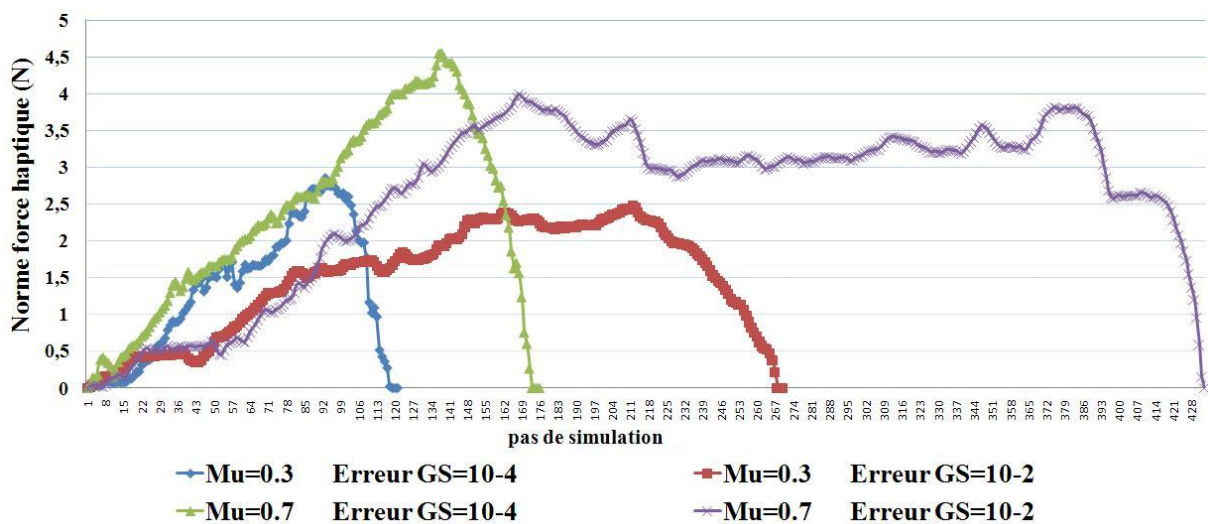


FIG. 4.26 – Forces haptiques restituées à l'interface haptique pendant un parcours pénétration, interaction aluminium/tissu mou $E = 45$ Mpa avec 36 contacts.

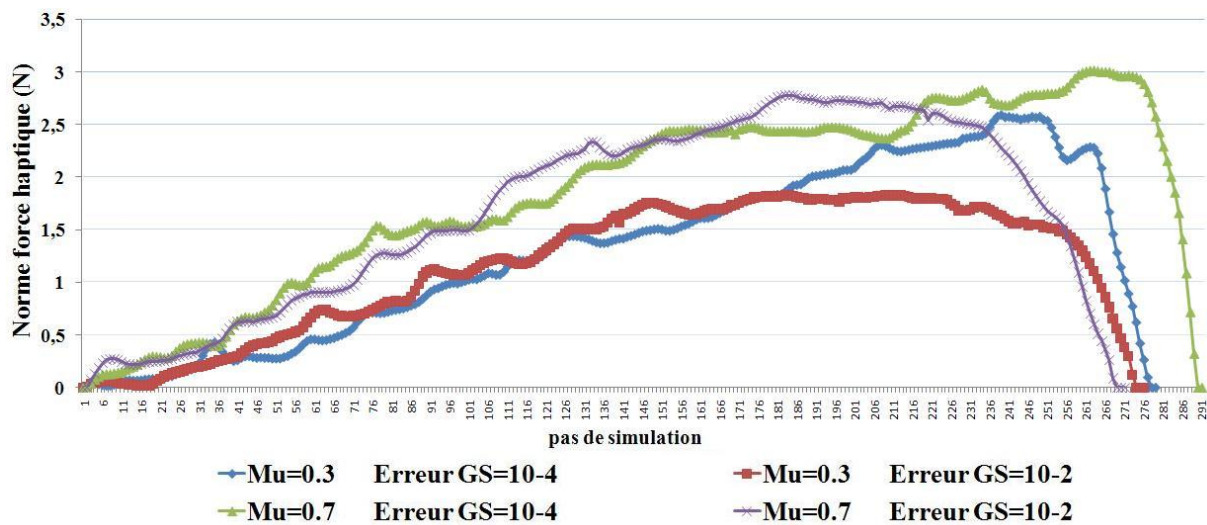


FIG. 4.27 – Forces haptiques restituées à l’interface haptique pendant un parcours pénétration, interaction aluminium/tissu mou $E = 45 \text{ Mpa}$ avec 16 contacts.

A travers ces différentes expériences haptiques (aluminium/tissu mou), les utilisateurs ont constaté un rendu haptique réaliste et stable lorsque le contact était composé de 16 points et que la précision Gauss-Seidel du solveur des forces de contact était égale à 10^{-2} . En effet, la fréquence haptique atteint alors la valeur de 100 Hz. Le ressenti a été jugé bon bien que les imperfections mécaniques liées au Phantom (l’effet de gravité, forces de frottement, forces inertielles) n’ont pas été compensées dans la simulation.

Les expérimentations haptiques réalisées entre l’objet impacteur en caoutchouc et le tissu mou obstacle ont montré qu’avec 16 points de contact et une précision Gauss-Seidel égal à 10^{-4} , nous obtenons de meilleures sensations haptiques. Augmenter, dans ce cas, la tolérance numérique de Gauss-Seidel (10^{-2}) dégrade la qualité du rendu haptique. On a constaté l’effet inverse obtenu lors d’interaction aluminium/tissu mou. Augmenter cette tolérance ne conduit donc pas forcément à un meilleur rendu haptique. Les fréquences haptiques obtenues varient entre 60 Hz et 100 Hz.

4.5 Conclusion

Nous avons décrit dans ce chapitre «HapCo» un simulateur interactif permettant de traiter et résoudre du multi-contact frottants entre objets déformables en élasticité linéaire et intégrant du retour haptique.

La simulation HapCo a servi de plateforme d’essai afin de tester et comparer les divers algorithmes de résolution des forces de contact présentés dans le chapitre 2.

Nous avons aussi mis au point «Multiple Contact Friction Solver», un solveur générique de forces de contact avec frottement indépendant des modèles de déformation utilisés. Nous avons montré comment nous l'avons adapté et intégré dans la plate forme logicielle «SOFA» développée par l'équipe Evasion (INRIA Grenoble). Cette collaboration initiée entre l'équipe Evasion et notre équipe (RVH IBISC) a permis de valider le principe général de ce solveur générique.

La simulation HapCo est munie d'une interface permettant de contrôler la force haptique en fonction des propriétés de l'interface haptique (force maximale) et la contrainte temps réel. A travers les différentes expérimentations haptiques réalisées (interaction aluminium/tissu mou et caoutchouc/tissu mou), nous avons obtenu des ressentis haptiques d'une qualité plutôt satisfaisante avec des objets simulés peu raides.

Dans le cas de contact raide entre l'objet mobile et l'objet obstacle, des instabilités haptiques sont fortement ressenties (des rebonds, des accouts). Ces difficultés peuvent avoir des origines diverses :

- Lorsque l'opérateur ne tient pas fermement le stylet haptique, il est surpris par la force haptique qui augmente quand le contact devient de plus en plus rigide.
- Les imperfections liées au dispositif à retour d'effort (équilibrage, poids, frottement, ...) qui ne sont pas pris en compte.
- Ces instabilités peuvent également survenir quand l'opérateur humain impose un grand mouvement rapide avec le stylet, c'est pour cette raison que l'utilisateur doit arriver avec une vitesse réduite surtout si le contact avec l'objet obstacle est rigide.

Afin de corriger ces instabilités, une fonction Zoom sur le contact permet à l'opérateur humain de mieux maîtriser et contrôler sa gesture et son mouvement afin d'obtenir une meilleure qualité du rendu haptique. Une autre solution à envisager permettant d'améliorer le ressenti haptique serait de considérer un couplage mécanique type ressort-amortisseur entre l'interface haptique et la simulation HapCo.

Les différentes expérimentations réalisées sur la simulation HapCo ont permis aussi de constater l'efficacité des algorithmes de résolution du contact avec frottement entre deux objets déformables. La condition de Signorini est respectée, et la transition adhérence/glisement est bien ressentie. D'autres part, nous avons tout de même réussi à provoquer de grandes déformations même avec un modèle de déformation en élasticité linéaire. Rappelons que ces expériences ont été effectuées sur un microprocesseur Intel(R) de fréquence 1.60 GHz. La considération d'un microprocesseur plus puissant, de fréquence 3 Ghz par exemple, augmentera les performances de la simulation HapCo et améliorera ainsi la qualité du rendu haptique.

4. HapCo : un prototype de simulateur temps réel avec retour haptique traitant des contacts multiples entre objets déformables.
-

Conclusion et Perspectives

Le travail qui a été présenté dans cette thèse s'inscrit dans la continuité des travaux initiés par Christian Duriez qui ont mis en évidence la nécessité de s'orienter vers des algorithmes de résolution par contraintes de type LCP ou de type Lagrangien augmenté pour traiter des problèmes de contact frottant en simulation interactive avec retour haptique. En effet, les méthodes avec fonctions de pénalité très populaires en animation graphique interactive posent des problèmes de stabilité lorsqu'un retour haptique est mis en place.

Nous avons proposé et testé plusieurs algorithmes basés sur deux types de formulation, pseudo-potentielle et bi-potentielle des lois de contact avec frottement. La formulation bi-potentielle développée par Zhi Qiang Feng n'avait jamais été testée dans un contexte temps réel et encore moins dans un contexte avec retour haptique. Suite à de nombreux tests numériques en faisant varier un certain nombre de paramètres comme l'élasticité (Module d'Young) des objets en contact, les précisions numériques des différentes boucles itératives, le coefficient de frottement, nous avons montré que cette formulation permet d'avoir un bon contrôle du calcul des forces de contact. Ce contrôle peut permettre à terme de réaliser le compromis optimal entre précision et rapidité de la boucle de calculs en fonction des ressources disponibles au niveau de la machine de calcul et de l'interface haptique. Le compromis optimal étant d'avoir un bon rendu haptique c'est à dire suffisamment réaliste, stable et robuste.

Les résultats que nous avons obtenus sont prometteurs dans le cas d'interaction entre objets déformables pour 16 points ou 36 points de contact. Toutefois lorsque la raideur des objets en contact augmente, des problèmes d'instabilité apparaissent lors de simulations avec retour haptique. Les raisons qui ont été identifiées sont dues à :

- une bande passante de la boucle de calcul trop faible (<100 Hz). Il faudrait augmenter la puissance de calculs et paralléliser les tâches. En effet l'implémentation qui a été faite tourne sur un seul processeur (Intel(R) de fréquence 1.60 GHz) en multi-threading sans affectation de tâches prioritaires. Dans nos applications, le goulet d'étranglement numérique n'est pas le solveur de contact mais plutôt le solveur de déplacement et le détecteur de collision. On peut également ajouter un couplage ressort amortisseur entre l'interface haptique et la simulation afin de filtrer les hautes fréquences.
- une interface haptique qui ne permet pas de déformer des objets élastiques à partir d'une certaine raideur car les forces de contact générées dépassent très vite les capacités disponibles (>5 N). D'autre part le retour d'effort n'est effectué que sur 3 degrés de liberté motorisés, on ne peut donc restituer qu'une force ou qu'un couple mais pas

les deux à la fois ce qui perturbe la perception haptique.

- un détecteur de collision de proximité trop rudimentaire. Les discrétisations des surfaces en contact étant grossières, il serait nécessaire de prévoir un lissage des normales aux points de contact dans le cas de déformation en cuvette. Le contact arête/arête n'a pas été implémenté.

Outre le travail sur le calcul des forces de contact et le rendu haptique, nous avons développé «Multiple Contacts Friction Solver», un solveur générique de forces de contact avec frottement indépendant des modèles de déformation utilisés. Nous avons montré comment nous l'avons adapté et intégré dans la plate forme logicielle «SOFA» développée par l'équipe Evasion (INRIA Grenoble). Cette collaboration initiée entre l'équipe Evasion et notre équipe (RVH IBISC) a permis de valider le principe général de ce solveur générique. Cependant il reste à être amélioré car les temps de calculs sont beaucoup trop longs pour les deux raisons suivantes :

- les appels des solveurs de déplacement des objets déformables ne sont pas optimisés. Il faudrait définir un standard de communication afin que ces appels ne concernent que la réactualisation des points nécessaires aux calculs des points de contact.
- «Multiple Contacts Friction Solver» est basé sur l'algorithme de Gauss Seidel ce qui nécessite de nombreux appels des solveurs de déplacement. Nous proposons comme solution de modifier «Multiple Contacts Friction Solver» en le basant sur l'algorithme de Jacobi ce qui diminuerait ces appels.

Les perspectives sont nombreuses qui dépassent largement le cadre de cette thèse et nécessiteraient un travail collaboratif soutenu avec d'autres équipes de recherche afin de développer une plateforme de simulation chirurgicale. Citons par exemple le développement de modèles de corps déformables hyper-élastiques, visco-plastiques, de détecteurs de collisions rapides, ... C'est pourquoi l'intégration de notre solveur de contact dans une plate-forme comme SOFA permettrait de multiplier rapidement des tests de validation tout en bénéficiant des apports d'autres équipes de recherche.

Références bibliographiques

- [AC91] P. Alart and A. Curnier. A mixed formulation for frictional contact problems prone to newton like solution methods. *Comput. Methods Appl. Mech. Engng.* 92, pages 353–375, 1991.
- [AH99] R. J. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3), pages 465–474, 1999. (pp 113, 114, 116, 117).
- [ALC99] D. Aulignac, C. Laugier, and M. C. Cavusoglu. Towards a realistic echographic simulator with force feedback. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2):727–732, 1999.
- [Aul01] D. Aulignac. *Modélisation de l'interaction avec objets déformables en temps-réel pour des simulateurs médicaux*. PhD thesis, Institut National Polytechnique de Grenoble, 2001.
- [Bar92] D. Baraff. *Dynamic simulation on non-penetrating rigid bodies*. PhD thesis, Computer Science Department, Cornell University, 1992.
- [Bat82] K. L. Bathe, editor. *Finite Element Procedures In Engineering Analysis*. Prentice-Hall, 1982.
- [BC00] D. Bourguignon and M. P. Cani. Controlling anisotropy in mass-spring systems. *the 11th Eurographics Workshop, Computer Animation and Simulation*, pages 113–123, 2000.
- [BHG92] D. E. Breen, D. H. House, and P. H. Getto. A physically-based particle model of woven cloth. *The Visual Computer*, pages 264–277, 1992.
- [BHW94] D. E. Breen, D. H. House, and M. J. Wozny. Predicting the drape of woven cloth using interacting particles. *ACM SIGGRAPH Orlando Florida (USA)*, pages 365–372, 1994.
- [BLM00] T. Belytschko, W. K. Liu, and B. Moran, editors. *Nonlinear Finite Elements for Continua and Structures*. Wiley, Chichester, UK, 2000.
- [BN91] T. Belytschko and M. O. Neal. Contact-impact by the pinball algorithm with penalty and lagrangian methods. *Int. J. Num. Meth. Engng.*, 31, pages 547–572, 1991.
- [BN98] M. Bro-Nielsen. Finite element modeling in surgery simulation. *Proceedings of the IEEE*, 86(3):490–503, Mar 1998.
- [BNC96] M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *EACG Conference on Eurographics*, 1996.
- [BW98] D. Baraff and A. Witkin. Large steps in cloth simulation. *Proceedings of SIGGRAPH*, pages 43–54, 1998.
- [BWK03] D. Baraff, A. Witkin, and M. Kass. Untangling cloth. *SIGGRAPH (ACM Transactions on Graphics) San Diego California (USA)*, pages 862–870, 2003.
- [CB86] A. B. Chaudhary and K. J. Bathe. A solution method for static and dynamic analysis of three dimensional contact problems with friction. *Computers and Structures*, 24(6), pages 855–873, 1986.

- [CB97] T. R. Chandrupatla and A. D. Belegundu, editors. *Introduction to Finite Elements in Engineering*. Prentice-Hall, Upper Saddle River, 1997.
- [CDA99] S. Cotin, H. Delingette, and N. Ayache. Real time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics*, 1999.
- [CDA00] S. Cotin, H. Delingette, and N. Ayache. A hybrid elastic model allowing real-time cutting, deformations and forcefeedback for surgery training and simulation. *Visual Computer*, (8):437–452, 2000.
- [CGC⁺02] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popovic. Interactive skeleton-driven dynamic deformations. *Proceedings of ACM Siggraph'02, Juillet, Juillet 2002*.
- [CHK95] A. Curnier, Q. C. He, and A. Klarbring. Continuum mechanics modelling of large deformation contact with friction. *in: M. Raous, M. Jean and J. J. Morau, eds. Contact Mechanics, Plenum, New York*, pages 145–158, 1995.
- [CHP89] J. E. Chadwick, D. R. Haumann, and R. E. Parent. Layered construction for deformable animated characters. *Computer Graphics*, pages 243–252, 1989.
- [Cia88] P. G. Ciarlet, editor. *Mathematical elasticity vol1: Three-dimensional elasticity*. Elsevier Science Publishers B.V., 1988.
- [Cia90] P. G. Ciarlet, editor. *Plates and Junctions in Elastic multi-structures*. Springer-Verlag, 1990.
- [Cia97] P. G. Ciarlet, editor. *Mathematical elasticity vol2: Theory of plates*. Elsevier Science Publishers B.V., 1997.
- [CJ01] B. Cambou and M. Jean. *Micromécanique des matériaux granulaires*. Hermès-Lavoisier, 2001, 2001.
- [CK00] H. K. Cakmak and U. Kühnapfel. Animation and simulation techniques for vr-training systems in endoscopic surgery. *Eurographics Workshop on Animation and Simulation EGAS*, pages 173–185, 2000.
- [CKPS98] P. W. Chritensen, A. Klarbring, S. Pang, and N. Strömberg. Formulation and comparison of algorithms for frictional contact problems. *Int. J. Num. Meth. Engng.*, 42, pages 145–173, 1998.
- [CLMP95] J. D. Cohen, M. C. Lin, D. Manocha, and M. K. Panamgi. I-collide: An interactive and exact collision detection system for large-scale environments. *ACM Interactive 3D Graphics Conference*, pages 189–196, 1995.
- [Cot97] S. Cotin. *Modèles anatomiques déformables en temps-réel: Application à la simulation de chirurgie avec retour d'effort*. PhD thesis, Epidaure-Sophia Antipolis, 1997.
- [CT71] S. H. Chan and I. S. Tuba. A finite element method for contact problems of solid bodies. *Int. J. Mech. Sci.*, 13, pages 615–639, 1971.
- [CTK91] N. J. Carpenter, R. L. Taylor, and M. G. Katona. Lagrange constraints for transient finite element surface contact. *Int. J. Num. Meth. Engng.*, 32, pages 103–128, 1991.

- [Cur84] A. Curnier. A theory of friction. *Int. J. Solids Structures*, 20(7), pages 637–647, 1984.
- [DA03] H. Delingette and N. Ayache. Soft tissue modeling for surgery simulation. *rapport*, 2003.
- [DCG⁺98] M. Downes, M. C. Cavusoglu, W. Gantert, L. W. Way, and F. Tendick. Virtual environments for training critical skills in laparoscopic surgery. In *In Proceedings of Medicine Meets Virtual Reality VI (MMVR'98)*, pages 316–322. IOS Press, 1998.
- [dCL99] F. Boux de Casson and C. Laugier. Modeling the dynamics of a human liver for a minimally invasive surgery simulator. In *MICCAI '99: Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1156–1165, London, UK, 1999. Springer-Verlag.
- [DDKA05] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic haptic rendering of interacting deformable objects in virtual. *IEEE trans. on visualization and computer Graphics*, Nov/Dec 2005.
- [Deb00] G. Debunne. *Animation multirésolution d'objets déformables en temps-réel Application à la simulation chirurgicale*. PhD thesis, Institut National Polytechnique de Grenoble, 2000.
- [Des97] M. Desbrun. *Modélisation et animation d'objets hautement déformables à l'aide de surfaces implicites*. PhD thesis, Institut National Polytechnique, Grenoble, 1997.
- [Dev03] C. Devine. temps réel d'objets déformables à base d'éléments finis. Technical report, Institut d'Informatique d'Entreprise, 2003.
- [dF91] G. deSaxcé and Z. Q. Feng. New inequality and functional for contact with friction: The implicit standard material approach. *Mech. Struct. and Mach.*, 19, pages 301–325, 1991.
- [dF98] G. deSaxcé and Z. Q. Feng. The bi-potential method: a constructive approach to design the complete contact law with friction and improved numerical algorithms. *Mathematical and Computer Modeling*, 28(4-8). special issue *Recent Advances in Contact Mechanics.*, pages 225–245, 1998.
- [DPZ90] F. Dominique, A. Pineau, and A. Zaoui, editors. *Comportement mécanique des matériaux*. Hermes, 1990.
- [dSZ99] A. Gomes de Sa and G. Zachmann. Virtual reality as a tool for verification of assembly and maintenance processes. *Computer and Graphics*, 23(3), (p 6), pages 389–403, 1999.
- [Dum98] P. Dumas. Diviser pour régner. <http://pauillac.inria.fr/algo/dumas/DC/>, page 29, 1998.
- [Dur04] C. Duriez. *Contact frottant entre objets déformables dans des simulations temps-réel avec retour haptique*. PhD thesis, Université d'Evry Val d'Essonne, 2004.
- [Eri04] C. Ericson. *Real Time Collision Detection*. Morgan Kaufmann Publishers, 2004.
- [ESHD05] K. Erleben, J. Sparring, K. Henriksen, and H. Dohlmann. *Physics-Based Animation*. edition charles river Media INC, CANADA, 2005.

- [EWW96] B. Eberhardt, A. Weber, and S. Wolfgang. A fast, flexible, particle-system model for cloth draping. *IEEE Computer Graphics and Applications*, pages 52–59, 1996.
- [FCM05] Z. Q. Feng, J. M. Cros, and B. Magnain. Un algorithme efficace pour les problèmes d'impact avec frottement. *Revue Européenne des Eléments Finis*, (14):65–86, 2005.
- [Fel00] C. A. Felippa. A systematic approach to the element-independent corotational dynamics of finite elements. *Technical report, Center for Aerospace Structures*, Janvier 2000.
- [Fen91] Z. Q. Feng. *Contribution à la modélisation des problèmes non linéaires: contact, plasticité et endommagement*. PhD thesis, Université de Technologie de Compiègne, 1991.
- [Fen95] Z. Q. Feng. 2d or 3d frictional contact algorithms and applications in a large deformation context. *Comm. Numer. Meth. Engng.*, 11, pages 409–416, 1995.
- [FHdM06] Z. Q. Feng, M. Hjiiaj, G. deSaxcé, and Z. Mróz. Effect of frictional anisotropy on the quasistatic motion of a deformable solid sliding on a planar surface. *Computational Mechanics - Special Issue Valliappan*, 37, pages 349–361, 2006.
- [FL01] S. Fisher and M. C. Lin. Fast penetration depth estimation for elastic bodies using deformed distance fields. *International Conference on Intelligent Robots and Systems*, 2001.
- [FPL03] Z. Q. Feng, F. Peyraut, and N. Laped. Solution of large deformation contact problems with friction between blatz-ko hyperelastic bodies. *International Journal of Engineering Science* 41, pages 2213–2225, 2003.
- [Fun65] Y. Fung, editor. *Foundations of solid mechanics*. Prentice-Hall, 1965.
- [FZ75] A. Francavilla and O. C. Zienkiewicz. A note on numerical computation of elastic contact problems. *Int. Num. Meth. Engng.* 9, (9):913–924, 1975.
- [GM97] S. F. F. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. Technical report, Mitsubishi Electric Research Laboratories, 1997.
- [GW97] A. Van Gelder and J. Wilhelms. Simulation of elastic membranes with triangulated spring meshes. *Technical Report UCSC-CRL-97-12, University of California, Santa Cruz, California (USA)*, 1997.
- [Hat06] M. Hatab. *Vers une approche unifiée de détection de collision pour objets de natures diverses*. PhD thesis, Université D'Evry Val D'Essonne, 2006.
- [HB00] D. H. House and D. E. Breen. Cloth modeling and animation. *A.K. Peters, Ltd*, 2000.
- [HC93] J. H. Heegaard and A. Curnier. An augmented lagrangian method for discrete large slip problems. *Int. J. Numer. Methods Engng.* 36, pages 569–593, 1993.
- [HFdM04] M. Hjiiaj, Z. Q. Feng, G. deSaxcé, and Z. Mróz. Three-dimensional finite element computations for frictional contact problems with non-associated sliding rule. *Int. J. Num. Meth. Engng.*, 60, pages 2045–2076, 2004.
- [HP98] P. Hunter and A. Pullan, editors. *FEM/BEM notes*. Bioengineering Research Group, Departement of Engineering Science, University of Auckland, New Zealand, 1998.

- [HS04] M. Hauts and W. Strasser. Corotational simulation of deformable solids. *Proceedings of WSCG*, pages 137–145, 2004.
- [HTS⁺76] T. J. R. Hughes, R. L. Taylor, J. L. Sackman, A. Curnier, and W. Kanoknukulchai. A finite element method for a class of contact-impact problems. *Comp. Meth. Appl. Mech. Engng.*, 8, pages 249–276, 1976.
- [Hug87] T. J. Hughes, editor. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Inc., Englewood cliffs, New Jersey, 1987.
- [Hun92] I. Hunék. On a penalty formulation for contact-impact problems. *Computers and Structures*, 48(2), pages 192–203, 1992.
- [ITF04] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, (8):20, 2004.
- [JCA06] L. Joussemet, A. Crosnier, and C. Andriot. Espions: A novel algorithm based on evolution strategy for fast collision detection. *In Proceedings of EuroHaptics 2006, Paris*, pages 159–166, 2006.
- [Jea95] M. Jean. Frictional contact in rigid or deformable bodies: numerical simulation of geomaterials. *A.P.S. Salvadurai J.M. Boulon, Elsevier Science Publisher, Amsterdam.*, pages 463–486, 1995.
- [Jea96] M. Jean. Documentation sur lmgc: Logiciel de mécanique gérant le contact. *Université de la Méditerranée, LMA, Marseilles.*, pages 47, 50, 72, 78, 102, janvier 1996.
- [Jea98] M. Jean. The non smooth contact dynamics method. *Computer Method in Applied Mechanics and Engineering, Special issue on contact and friction, eds Klarbung and Martins.*, pages 39, 50, 94, 1998.
- [JF08] P. Joli and Z. Q. Feng. Uzawa and newton algorithms to solve frictional contact problems within the bi-potential framework. *Int. J. Numer. Meth. Engng.*, (73):317–330, 2008.
- [JM92] M. Jean and J. J. Moreau. Unilaterality and dry friction in the dynamics of rigid bodies collections. *In A. Curnier, (ed), Proc. of Contact Mech. Int. Symp.*, pages 31–48, 1992.
- [JP04] D. L. James and D. K. Pai. Bd-tree: Output-sensitive collision detection for reduced deformable models. *ACM Transactions on Graphics, SIGGRAPH Conference*, 23(3), pages 393–398, Août 2004.
- [JT88] M. Jean and G. Touzot. Implementation of unilateral contact and dry friction in computer codes dealing with large deformation problems. *J. Theor. Appl. Mech.* 7, pages 145–160, 1988.
- [JW03] D. Johnson and P. Willemsen. Six degree-of-freedom haptic rendering of complex polygonal models. *Proceedings of Haptics Symposium, (p 32)*, 2003.
- [JW04] D. Johnson and P. Willemsen. Accelerated haptic rendering of polygonal models through local descent. *IEEE editor, Proceedings of Haptics Symposium, (p 32)*, 2004.
- [KaM00] U. Kühnapfel, H. K. Çakmak, and H. MaaSS. Endoscopic surgery training using

- virtual reality and deformable tissue simulation. *Computer and Graphics*, pages 671–682, 2000.
- [KF03] M. J. Kilgard and R. Fernando, editors. *The CG Tutorial: The Definitive Guide to Programmable Real-Time Graphics*. 2003.
- [KHM⁺98] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Transactions on Visualization and Computer Graphics*, 4(1), (p 29), pages 21–36, janvier 1998.
- [Kil96] M. J. Kilgard, editor. *OpenGL Programming for the X Window System*. 1996.
- [KKH⁺97] U. Kühnapfel, C. Kuhn, M. Hübner, H. G. Krumm, H. MaaSS, and B. Neisius. The karlsruhe endoscopic surgery trainer as an example for virtual reality in medical education. *Minimally Invasive Therapy and Allied Technologies*, pages 122–125, 1997.
- [Kla92] A. Klarbring. Mathematical programming and augmented lagrangian methods for frictional contact problems, in: A. curnier (ed.). *Proc. Contact Mechanics Int. Symp., PPUR*, pages 369–390, 1992.
- [KLM02] Y. Kim, M. Lin, and D. Manocha. Deep: Dual-space expansion for estimating penetration depth between convex polytopes. *IEEE International Conference on Robotics and Automation, Washington*, May 2002.
- [KO84] N. Kikuchi and J. T. Oden, editors. *Contact problems in elastostatics, in: J. T. Oden, G.F.Carey(Eds.), Finite Elements*. Prentice-Hall, Englewood Cliffs, NJ, 1984.
- [KS81] N. Kikuchi and Y. J. Song. Penalty/finite element approximations of a class of unilateral problems in linear elasticity. *Quarterly of Applied Mathematics*, 39, pages 1–22, 1981.
- [KZ03] J. Klein and G. Zachmann. Adb-trees: Controlling the error of time-critical collision detection. *8th International Fall Workshop Vision, Modeling, and Visualization, Munich, Allemagne, (p 33)*, 2003.
- [Law93] D. A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Trans. Robotics and Automation*, pages 624–637, Octobre 1993. (p 113).
- [Len04] J. Lenoir. *Modèle déformable 1D pour la simulation physique temps réel*. PhD thesis, l’Université des Sciences et Technologies de Lille, 2004.
- [Lin04] M. Lin. Physically-based modeling, simulation and animation. *Notes de cours, Université de Caroline du Nord à Chapel Hill*, 2004.
- [LM01] T. Larson and T. A. Möller. Collision detection for continuously deforming bodies. *Eurographics, short presentation, (p 30)*, 2001.
- [LMS03] C. Laugier, C. Mendoza, and K. Sundaraj. *Towards a realistic medical simulator using virtual environments and haptic interaction*. Robotics Research, volume 6 of Springer Tracts in Advanced Robotics (STAR). Springer-Verlag, 2003.
- [LPC95] J. Louchet, X. Provot, and D. Crochemore. Evolutionary identification of cloth animation models. *Eurographics Workshop on Computer Animation and Simulation, Maastricht (Netherlands)*, pages 44–54, 1995.
- [MC00] P. Meseure and C. Chaillou. A deformable body model for surgical simulation. *Visualization and Computer Animation*, (11):197–208, 2000.

- [MDH⁺03] P. Meseure, J. Davanne, L. Hilde, J. Lenoir, L. France, F. Triquet, and C. Chaillou. A physically-based virtual environment dedicated to surgical simulation. *International Symposium on Surgery Simulation and Soft Tissue Modeling*, 2003.
- [MG04] M. Müller and M. Gross. Interactive virtual materials. *Proceedings of Graphics Interface*, pages 239–246, Mai 2004.
- [MHBW01] K. Montgomery, L. Heinrichs, C. Bruyns, and S. Wildermuth. Surgical simulator for hysteroscopy: a case study of visualization in surgical training. *Vizualisation conference San Diego*, pages 449–452, 2001.
- [MK06] P. Meseure and A. Kheddar. *Traité de la Réalité Virtuelle, chapitre Détection de collisions*. Morgan Kaufmann Publishers, 2006.
- [MKE03] S. Mezger, J. Kimmerle, and O. Etmuss. Hierarchical techniques in collision detection for cloth animation. *Journal of WSCG, (p 29)*, pages 322–329, 2003.
- [MMD⁺02] M. Müller, L. McMillan, J. Dorsey, R. Jagnow, and B. Cutler. Stable real-time deformations. *Proceedings of ACM Siggraph'02, Juillet*, Juillet 2002.
- [Moh05] H. Mohellebi. *Conception et réalisation de systèmes de restitution de mouvement et de retour haptique pour un simulateur de conduite à faible coût dédié à l'étude comportementale du conducteur*. PhD thesis, Université d'Evry Val d'Essonne, 2005.
- [MS94] Z. Mróz and S. Stupkiewicz. An anisotropic friction and wear model. *Int. J. Solids and Structures*, 31(8), pages 1113–1131, 1994.
- [MWTT98] W. Maurel, Y. Wu, N. M. Thalmann, and D. Thalmann, editors. *Biomechanical Models for Soft Tissue Simulation*. Springer-Verlag, 1998.
- [Nes04] M. Nesme. *Modèle déformable par éléments finis élasticité linéaire et grands déplacements, pour application à la simulation chirurgicale*. Technical report, Institut National Polytechnique de Grenoble, 2004.
- [NMA⁺08] L. Nehaoua, H. Mohellebi, A. Amouri, H. Arioui, S. Espié, and A. Kheddar. Design and control of a small-clearance driving simulator. *IEEE Transactions on Vehicular Technology*, 57(1), 1(57):736–746, 2008.
- [NNF⁺02] I. Nikitin, L. Nikitina, P. Frolov, G. Goebbels, M. Göbel, S. Klimenko, and G. Nielson. Real-time simulation of elastic objects in virtual environments using finite element method and precomputed green's functions. *Eurographics Workshop on Virtual Environments*, 2002.
- [NT98] L. P. Nedel and D. Thalmann. Real time muscle deformations using massspring systems. *CGI, IEEE Computer Society Press*, 1998.
- [NW86] B. Nouromid and P. Wriggers. A two level iteration method for solution of contact problems. *Comput. Methods. Appl. Mech. Engng.* 54, pages 131–144, 1986.
- [Ode67] J. Oden, editor. *Mechanics of elastic structures*. Mc Graw-Hill, 1967.
- [Ogd84] R. Ogden, editor. *Non-linear elastic deformation*. Ellis Horwood limited, 1984.
- [Pac] A. Paccini. *Développement d'un modèle éléments finis 3D appliqué à la simulation d'opérations chirurgicales des tissus mous*. PhD thesis.
- [Par82] F. I. Parke. Parameterized models for facial animation. *IEEE Computer Graphics and Application*, pages 61–68, 1982.

- [Par89] H. A. Parisch. Consistent tangent stiffness matrix for three-dimensional nonlinear contact analysis. *Int. J. Num. Meth. Engng.*, 28, pages 1803–1812, 1989.
- [PB81] S. Platt and N. Badler. Animating facial expressions. *Proceedings of SIGGRAPH*, pages 245–252, 1981.
- [PDA03] G. Picinbono, H. Delingette, and N. Ayache. Non-linear anisotropic elasticity for real-time surgery simulation. *Graphical Models*, pages 305–321, 2003.
- [Pic01] G. Picinbono. *Modèles géométriques et physiques pour la simulation d'interventions chirurgicales*. PhD thesis, Université de Nice INRIA Sophia Antipolis, 2001.
- [PNT98] L. Porcher-Nedel and D. Thalmann. Real time muscle deformations using mass-spring systems. *CGI'98, IEEE Computer Society Press, Hannover (Germany)*, 1998.
- [Pro95] X. Provat. Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Canadian Human-Computer Communications Society, Graphics Interface*, pages 147–154, 1995.
- [Pro97] X. Provat. Collision and self-collision handling in cloth model dedicated to design garments. *Graphics Interface. (pp 27,29)*, 1997.
- [PS98] P. Papadopoulos and J. M. Solberg. A lagrange multiplier method for the finite element solution of frictionless contact problems. *Mathematical and Computer Modelling*, 28, pages 373–384, 1998.
- [PTDRC03] A. Paccini, Y. Tillier, M. Durand-Reville, and J. L. Chenot. Modèle éléments finis 3d pour la simulation de chirurgie, application : indentation d'un rein. *16ème Congrès Français de Mécanique Nice, 1-5 septembre 2003*, 2003.
- [RKC02] S. Redon, A. Kheddar, and S. Coquillart. Fast continuous collision detection between rigid boies. *Eurographics 2002 Proceedings*, 2002.
- [RKK97] D. C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. *Proceedings of ACM Siggraph'97, Los Angeles*, pages 34–352, Août 1997.
- [RL96] D. Reznik and C. Laugier. Dynamic simulation and virtual control of a deformable fingertip. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1669–1674, 1996.
- [SBB⁺00] G. Székely, M. Bajka, C. Brechbuhler, J. Dual, R. Enzler, U. Haller, J. Hug, R. Hutter, N. Ironmonger, M. Kauer, V. Meier, P. Niederer, A. Rhomberg, P. Schmid, G. Schweitzer, M. Thaler, V. Vuskovic, and G. Troster. Virtual reality based simulation of endoscopic surgery. *in: Presence MIT press*, pages 310–333, 2000.
- [SBH⁺98] G. Székely, C. Brechbuhler, R. Hutter, A. Rhomberg, and P. Schmid. Modeling of soft tissue deformation for laparoscopic surgery simulation. *First International Conference on Medical Image Computing and Computer-Assisted Intervention MICCAI*, pages 550–561, 1998.
- [SBH⁺00] G. Székely, C. Brechbuhler, R. Hutter, A. Rhomberg, and P. Schmid. Modelling of soft tissue deformation for laparoscopic surgery simulation. *Medical Image Analysis*, pages 57–66, 2000.
- [SH98] J. C. Simo and T. J. R. Hughes, editors. *Computational Inelasticity*. Springer-Verlag, New York, 1998.

- [Sha96] T. Shawki, editor. *Lecture Notes on Fundamentals of Solid Mechanics*. Department of Theoretical and Applied Mechanics, University of Illinois, 1996.
- [SHT⁺98] N. Suzuki, A. Hattori, A. Takatsu, T. Kumano, A. Ikemoto, Y. Adachi, and A. Uchiyama. Virtual surgery system using deformable organ models and force feedback system with three fingers. *First International Conference on Medical Image Computing and Computer-Assisted Intervention MICCAI*, pages 397–403, 1998.
- [SL92] J. C. Simo and T. A. Laursen. An augmented lagrangian treatment of contact problems involving friction. *Comput. Struct*, 42, pages 97–116, 1992.
- [SP98] A. M. Solberg and P. Papadopoulos. A finite element method for contact/impact. *Finite Elements in Analysis and Design*, 30, pages 297–311, 1998.
- [Spe84] A. J. M. Spencer, editor. *Continuum Theory of Fiber-Reinforced Composites*. Springer-Verlag, New York, 1984.
- [SR81] T. D. Sachdeva and C. V. Ramarkrishnan. A finite element solution for the two-dimensional elastic contact problems with friction. *Int. J. Numer. Methods Engng.* 17, pages 1257–1271, 1981.
- [SRN81] T. D. Sachdeva, C. V. Ramarkrishnan, and R. Natarajan. A finite element method for the elastic contact problems. *Transactions of the ASME*, 103, pages 456–461, 1981.
- [TF88] D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. *SIGGRAPH Atlanta, USA*, pages 269–278, 1988.
- [TGG99] M. Teschner, S. Girod, and B. Girod. Optimization approaches for soft-tissue prediction in craniofacial surgery simulation. In *In: 2nd Int. Conf. on Medical Image Computing and Computer Assisted Intervention, MICCAI'99. Volume 1679 of LNCS*, pages 1183–1190. Springer, 1999.
- [TGG00] M. Teschner, S. Girod, and B. Girod. Direct computation of nonlinear softtissue deformation. *Proceedings of Vision, Modeling, and Visualization VMV'00*, pages 383–390, 2000.
- [TJF07] N. Talbi, P. Joli, and Z. Q. Feng. A contact solver for surface-based haptic rendering. *Surgetica 2007*, 2007.
- [TJFK07] N. Talbi, P. Joli, Z. Q. Feng, and A. Kheddar. Hapco: a real-time simulator with haptic device for problems with multiple contacts. *huitième colloque national en calcul des structures, Giens, Mai 2007*.
- [TJFK08] N. Talbi, P. Joli, Z. Q. Feng, and A. Kheddar. Hapco: real time simulation of interaction between deformable objects with haptic feedback for solving friction multiple contacts. *GRAPP2008, 22-25 January, Funchal, Madeira Island, PORTUGAL*, pages 55–61, 2008.
- [TPBF87] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *SIGGRAPH USA*, pages 205–214, 1987.
- [TY73] T. Tsuta and S. Yamaji. Finite element analysis of contact problem. *Tokyo seminar on element analysis, editor, Theory and practice in finite element structural analysis*, pages 178–194, 1973.

- [VCMT95] P. Volino, M. Courchesne, and N. Magnenat-Thalmann. Versatile and efficient techniques for simulating cloth and other deformable objects. *SIGGRAPH, Annual Conference Series , Los Angeles, California (USA)*, pages 137–144, 1995.
- [vdB97] G. van den Bergen. Efficient collision detection of complex deformable models using aabb trees. *Journal of Graphics Tools, (p 29)*, 1997.
- [vdB01] G. van den Bergen. Proximity queries and penetration depth computation on 3d game objects. *Game Developer Conference*, 2001.
- [vdB04] G. van den Bergen. *Collision Detection in Interactive 3D Environments*. Morgan Kaufmann Publishers, 2004.
- [VMT00] P. Volino and N. Magnenat-Thalmann. Implementing fast cloth simulation with collision response. *In Computer Graphics International (IEEE Computer Society DL Geneva (Switzerland))*, pages 257–268, 2000.
- [Wan02] G. G. Wang. Definition and review of virtual prototyping. *Journal of Computing and Information Science in Engineering, 2(3), (p 6)*, pages 232–236, 2002.
- [Wri02] P. Wriggers, editor. *Computational contact mechanics*. John Wiley and Sons, 2002.
- [Zac98] G. Zachmann. Rapid collision detection by dynamically aligned dop-trees. *Proceedings of IEEE VRAIS, (p 29)*, 1998.
- [ZC00] Y. Zhuang and J. Canny. Haptic interaction with global deformations. *Proceedings of IEEE ICRA, San Francisco, (pp20, 21, 88)*, 2000.
- [ZfV02] F. Zara, F. Faure, and J. M. Vincent. Physical cloth simulation on a pc cluster. *Parallel Graphics and Visualisation, EG Workshop Proceedings*, pages 105–112, 2002.
- [Zie77] O. Zienkiewicz, editor. *The finite element method*. Mc Graw-Hill, London, 1977.
- [ZPS01] Y. Zhang, E. C. Prakash, and E. Sung. Real-time physically-based facial expression animation using mass-spring system. *Computer Graphics International Conference*, 0:347, 2001.