Suture en chirurgie virtuelle : simulation interactive et modèles hétérogènes

THÈSE

présentée et soutenue publiquement le 1er Juillet 2010

pour l'obtention du

Doctorat de l'Université des Sciences et Technologies de Lille

(spécialité informatique)

par

Christophe Guébert

Composition du jury

Président :	Nouredine Melab, Professeur	Université de Lille I
Rapporteurs :	Georges Dumont, Maître de Conférences	École Normale Supérieure de Cachan
	Emmanuel Promayon, Maître de Conférences	Université Joseph Fourier à Grenoble
Examinateurs :	Christian Duriez, Chargé de recherche	INRIA
Invités :	Stephane Cotin, Directeur de recherche	INRIA
	Clément Forest, Chef d'entreprise	Digital Trainers
Directeur :	Laurent Grisoni, Professeur	Université de Lille I

UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE
Laboratoire d'Informatique Fondamentale de Lille — UPRESA 8022
U.F.R. d'I.E.E.A. – Bât. M3 – 59655 VILLENEUVE D'ASCQ CEDEX
Tél. : +33 (0)3 28 77 85 41 – Télécopie : +33 (0)3 28 77 85 37 – email : direction@lifl.fr

Mis en page et composé avec LaTeX et la classe thlifl.

Remerciements

Je tiens à remercier tout d'abord à remercier Georges Dumont et Emmanuel Promayon qui m'ont fait l'honneur de rapporter cette thèse ; leurs remarques et commentaires contribuant d'autant à son amélioration. Je remercie également les autres membres du jury, Stephane Cotin et Clément Forest pour l'intérêt porté sur mon travail tout au long de ma thèse. Merci aussi à Nouredine Melab qui a accepté d'être président du jury.

Un grand merci à Laurent Grisoni, mon directeur de thèse, pour son soutien et ses conseils. Je remercie également Christian Duriez, mon encadrant dont les nombreuses idées sont à l'origine de ce travail.

Ces trois années de thèse n'auraient pas été les mêmes sans les échanges avec les membres de l'équipe Graphix/Alcove. Un grand merci aux thésards qui ont fini avant moi et qui m'ont montré le chemin à suivre : Julien, Damien, Jérémie et Adrien. Une pensée pour mes compagnons de thèse Jérémy, Hadrien, Olivier et Philippe. Un grand merci également aux ingénieurs et chercheurs toujours disponibles pour donner un coup de main au niveau de la programmation : Pierre-Jean, Frédérick et Juan-Pablo, sans oublier Jérémie Allard. Je remercie également tous les membres de l'IRCICA et de l'INRIA avec qui j'ai passé du temps, pour le cadre de travail convivial et les nombreuses pauses café productives.

Un grand merci à l'entreprise Digital-Trainers et tout particulièrement à Clément Forest qui m'a fait confiance, à la fois pour finir ma thèse et pour continuer mon travail de recherche dans le cadre d'un nouvel emploi.

Enfin, je tiens à remercier toute ma famille, pour le soutien qui a été sans faille malgré la distance, les relectures ainsi que l'intérêt montré à tenter de comprendre puis de vulgariser mon travail auprès des amis.

Résumé

La médecine a été une grande motivation dans la recherche en informatique, et nous pensons que la simulation de procédures médicales sera un élément majeur de la médecine du 21e siècle. Pour ces applications, la simulation physique doit être interactive et proposer un comportement et un rendu visuel des objets simulés qui soient réalistes tout en faisant avec un temps de calcul limité. Dans ce travail, nous nous sommes intéressés à la simulation du geste chirurgical de la suture, faisant interagir des modèles hétérogènes (rigides, déformables 1D et 3D) dans la reproduction d'un geste complexe. Nous proposons de modéliser ces interactions par des contraintes de complémentarité, avec une méthode générique et une résolution indépendante des types de contraintes utilisées. Cette approche a permis la mise au point de contraintes spécifiques pour créer des modèles d'interaction nouveaux. Ces contraintes ont permis une simulation complète de l'insertion d'aiguille, validée par les mesures expérimentales des travaux antérieurs, qui suit les lois physiques et s'adapte à la manipulation de l'utilisateur. Enfin, une simulation interactive très réaliste a été réalisée pour l'entraînement au geste de suture dans le cadre d'opérations laparoscopiques.

Abstract

Medicine has been a great motivation in computer science, and we think the simulation of medical gestures will be an important part of the medicine of the 21st century. For these applications, the physical simulation has to be interactive and use realistic models for the behavior and the visualization of the simulated objects, all in a limited computation time. In this work, we have looked at the medical simulation of the complex gesture of the suture, where heterogeneous models interact (rigids, 1D and 3D deformable models). We propose a generic method based on complementarity constraints for the modelisation of these interactions, using a resolution independent of the types of constraints used in the simulation. This approach enabled us to create specific constraints for new interaction models. Using these constraints following the physical laws, a complete simulation of the insertion of needles has been proposed. The user has the control over this interactive simulation, and the interaction model has been validated using previous experimental measures. A realistic interactive simulation of the suture has been created for the training of surgeons during laparoscopic operations.

Table des matières

1 Introduction **9**

2 État de l'art : vers la simulation de suture chirurgicale **11**

2.1 Du point de vue médical 11

2.1.1 Enjeux de l'insertion d'aiguille 12

2.1.2 Étude des forces mises en jeu 13

2.1.3 Généralités sur la suture 19

2.2 Simulation physique 21

2.2.1 Objets rigides 22

2.2.2 Objets déformables 23

2.3 Interactions entre objets 28

2.3.1 Méthodes de résolutions des contraintes 28

2.3.2 Couplage entre objets déformables et rigides 36

2.4 Simulation d'insertion d'aiguille 37

2.4.1 Interaction de l'aiguille avec un tissu mou 37

2.4.2 Déflexion de l'aiguille durant son insertion 42

2.4.3 Conclusion sur les approches présentées 45

3 Une approche générique de l'interaction entre modèles hétérogènes 47

3.1 Quelques travaux antérieurs . . . 48

3.2 Contraintes généralisées . . . 50

3.2.1 Contraintes bilatérales . . . 50

3.2.2 Contraintes unilatérales . . . 51

3.2.3 Contraintes de frottement . . . 52

3.3 Espace des contraintes . . . 53

3.3.1 Contraintes interpolées . . . 54

3.3.2 Contraintes et représentations multi-modèles des objets . . . 56

3.4 Résolution . . . 57

3.4.1 Présentation de la méthode . . . 57

3.4.2 Traitement spécifique à la contrainte . . . 58

3.4.3 Gauss-Seidel modifié . . . 59

3.4.4 Une note sur l'architecture logicielle . . . 61

3.4.5 Integration implicite . . . 63

3.4.6 Convergence de l'algorithme . . . 63

3.5 Exemples de contraintes . . . 68

3.5.1 Point fixe . . . 69

3.5.2 Contact . . . 71

3.5.3 Frottement . . . 74

3.5.4 Point glissant . . . 74

3.5.5 Simulations utilisant des contraintes hétérogènes . . . 77

3.6 Conclusion . . . 80

4 Un modèle d'interaction pour l'insertion d'aiguille **81**

4.1 Contraintes à la pointe de l'aiguille . . . 83

4.1.1 Détection de contact . . . 84

4.1.2 Contact avec frottement . . . 86

4.1.3 Contrainte de pénétration . . . 87

4.1.4 Contrainte de découpe . . . 89

4.1.5 Incidence de la forme de l'aiguille . . . 91

4.2 Contraintes sur le corps de l'aiguille . . . 92

4.2.1 Maintenir l'aiguille dans son chemin . . . 92

4.2.2 Frottement dans un matériau homogène . . . 93

4.2.3 Prise en compte de la contrainte dans le matériau . . . 96

4.2.4 Matériau hétérogène . . . 98

4.3 Discrétisation des contraintes . . . 99

4.3.1 Création dynamique de contraintes . . . 99

4.3.2 Passage d'un objet à l'autre . . . 101

4.3.3 Influence sur les performances . . . 103

4.4 Paramétrisation et validation du modèle d'interaction . . . 103

4.4.1 Test d'insertion . . . 105

4.4.2 Évitement d'un obstacle . . . 105

4.4.3 Déflexion de l'aiguille . . . 107

4.4.4 Insertion de plusieurs aiguilles simultanément . . . 108

4.5 Conclusion . . . 108

5 Vers la simulation du protocole de suture **111**

5.1 Le modèle de fil de suture . . . 112

5.1.1 Les modèles déformables 1D . . . 112

5.1.2 Réalisation de noeuds . . . 118

5.1.3 Modèle unifié d'aiguille et de fil . . . 124
5.2 Interactions lors du geste de suture . . . 127
5.2.1 Les modèles d'interaction existants . . . 127
5.2.2 Collision et autocollision du fil . . . 131
5.2.3 Contacts . . . 133
5.2.4 Pénétration . . . 135
5.2.5 Frottement . . . 136
5.2.6 Préhension de l'aiguille par l'utilisateur . . . 137
5.3 Simulation d'un geste simple de suture . . . 139
5.4 Conclusion . . . 141

6 Conclusion **143**

Bibliographie **149**

A La méthode des contraintes dans SOFA **167**

Chapitre 1

Introduction

La variété et la complexité de la médecine, ainsi que son importance éthique dans la société d'aujourd'hui, ont été une grande motivation dans beaucoup de disciplines techniques et scientifiques. Le médical a déjà été un domaine d'application de la recherche en informatique, et plusieurs outils, tels que le traitement d'images, font maintenant partie intégrante de la médecine moderne. Il n'y a aucun doute que l'intégration de nouvelles technologies dans la médecine va continuer dans le futur. Dans ce contexte, nous pensons que la simulation de procédures médicales, quelle soit orientée vers l'éducation, l'entraînement, la préparation des interventions, ou même l'aide au chirurgien durant l'opération, sera un élément majeur de la médecine du vingt-et-unième siècle.

Pour ces applications, la simulation physique doit relever plusieurs problématiques intéressantes. Le chirurgien ayant maintenant sa place au sein de la simulation, celle-ci se doit d'être interactive : des périphériques physiques sont utilisés pour manipuler des objets virtuels et fournir un retour haptique, le temps de calcul est limité pour visualiser instantanément les conséquences de ses actions, le comportement et le rendu visuel des objets simulés doivent être vraisemblants pour donner l'illusion.

Dans ce travail de thèse, nous avons voulu nous intéresser à la simulation du geste chirurgical de la suture. Cette opération très fréquente est caractéristique des problématiques actuelles dans le domaine de la recherche en simulation médicale. Nous désirons reproduire un geste complexe, dans une simulation temps réel faisant intervenir des modèles hétérogènes. Ainsi, un simulateur de suture fera appel à un modèle déformable 1D pour le fil chirurgical, des modèles déformables 3D pour les différents organes et à des modèles rigides articulés pour les instruments.

Nous proposons une méthode pour gérer les interactions entre modèles hétérogènes qui dépend très peu des formalismes et algorithmes choisis pour représenter chaque modèle. Ces interactions sont gérées par des contraintes de complémentarité, dont la définition a été rendue générique. Ainsi, leur résolution par l'algorithme du Gauss-Seidel est elle-même indépendante des types de contraintes utilisées dans la simulation. Cette approche permet de faciliter grandement la création de contraintes dans le but de simuler des modèles d'interaction nouveaux.

L'approche que nous présentons nous a permis de réaliser une simulation très complète et réaliste de l'insertion d'aiguille, geste opératoire qui a fait l'objet de nombreuses recherches

à la fois dans le domaine bio-médical ainsi que celui de la simulation physique. Alors que les travaux antérieurs utilisent dans l'ensemble une interaction basée sur des profils de forces très peu dépendants de l'action de l'utilisateur, nous montrons que l'association de plusieurs contraintes de complémentarité permet de reproduire tous les comportements établis dans la littérature, en suivant les lois physiques et laissant une plus grande liberté pour la manipulation de l'aiguille. Le modèle d'interaction pour la simulation d'insertion d'aiguille que nous proposons a été validé avec les mesures expérimentales à notre disposition.

Enfin, un premier simulateur interactif du geste de suture a été réalisé, permettant à l'utilisateur de s'entraîner à cette procédure complexe dans le cadre d'opérations laparoscopiques. L'utilisation de modèles physiques au comportement validé en relation avec des contraintes suivant les lois physiques et des mesures expérimentales permet d'atteindre un réalisme encore inégalé dans une simulation temps réel de suture.

Ce travail de thèse sera divisé en 4 chapitres. Nous ferons un état de l'art des simulations d'insertion d'aiguille et de suture au chapitre 2, aussi bien d'un point de vue médical que de la simulation physique. Le chapitre 3 présente notre méthode pour l'interaction de modèles hétérogènes basée sur des contraintes généralisées. Nous présenterons un modèle d'interaction basé sur cette méthode pour la simulation de l'insertion d'aiguille au chapitre 4. Enfin, nous présentons au chapitre 5 les problématiques posées par la simulation réaliste d'un fil chirurgical, et utiliserons nos contraintes de complémentarité pour un simulateur du geste de suture.

Chapitre 2

État de l'art : vers la simulation de suture chirurgicale

Dans ce travail de thèse, nous désirons proposer une méthode basée sur des contraintes de complémentarité permettant de simuler des interactions complexes entre modèles hétérogènes. Nous voulons illustrer cette méthode par la mise au point de deux modèles d'interaction dans le cadre de simulateurs médicaux, l'un pour l'insertion d'aiguille flexible, un second pour le geste de suture.

Dans un premier lieu, nous nous intéresserons à ces simulations d'un point de vue médical et biomécanique. Nous aborderons ainsi les enjeux de ces opérations chirurgicales ainsi que de leur simulation, et présenterons les expériences qui ont été réalisées pour comprendre plus précisement l'insertion d'aiguille.

Nous regarderons alors les besoins de ces simulations du point de vue de l'informatique graphique. Nous présenterons tout d'abord les approches les plus courantes permettant de simuler le comportement des objets rigides et déformables. Ensuite, nous nous intéresserons à l'interaction entre ces objets, principalement via les méthodes de contraintes mais également par quelques travaux qui cherchent à coupler des modèles hétérogènes dans une même simulation. La détection de collision, bien qu'indispensable à toute simulation physique, n'a pas fait l'objet de recherches pour ce travail de thèse, et ne sera donc pas abordée en détails dans cet état de l'art. Enfin, nous aborderons les approches proposées pour simuler l'insertion d'aiguille dans des tissus mous, ainsi que la déflexion de cette aiguille lors de son insertion. Alors que nous aurons vu dans la première partie de l'état de l'art les aspects médicaux de ces opérations, nous nous concentrerons alors sur les problématiques de leurs modèles d'interaction.

2.1 Du point de vue médical

Nous voulons présenter dans cette partie les enjeux des gestes chirurgicaux que nous voulons reproduire et aborder certains aspects importants au niveau biomédical qui vont avoir un impact sur nos simulations. Nous allons donc nous intéresser tout d'abord à l'insertion

d'aiguille, puis aux mesures expérimentales qui ont été faites pour mieux comprendre cette opération. Nous présenterons alors la suture d'un point de vue médical, et montrerons l'intérêt d'en réaliser la simulation.

2.1.1 Enjeux de l'insertion d'aiguille

Un grand nombre de pratiques médicales font appel de nos jours à des diagnostiques ou des traitements percutanés. De fins appareils tubulaires tels que aiguilles, electrodes, cathéters ou sondes sont inserés profondément dans le corps humain à travers des tissus déformables et inhomogènes pour atteindre une cible.

Les applications sont de divers types :
- La biopsie, qui correspond à un prélèvement d'un échantillon de tissus dans le but de réaliser un examen. L'aiguille utilisée présente un tube creux dans lequel va s'accumuler le tissu au fur et à mesure qu'elle s'y enfonce. Lorsque l'aiguille est retirée, l'échantillon reste dans le tube et peut y être récupéré.
- L'anesthésie loco-régionale consiste à injecter un anesthésique local très proche d'un nerf afin d'agir sur tout le territoire desservi par celui-ci. On fait préalablement appel à un stimulateur nerveux afin de reconnaître le nerf à bloquer et ainsi de localiser précisement l'endroit où le produit sera déposé.
- En neurochirurgie, des aiguilles ou des cathéters peuvent être employées pour atteindre des cibles précises et y délivrer des produits. L'électro-stimulation profonde du cerveau consiste à implanter des electrodes à l'intérieur du cerveau afin d'y envoyer des signaux electriques pour interférer avec l'activité nerveuse du site. Un guide est d'abord inséré, dans lequel une sonde permet de repérer précisement l'emplacement à stimuler, grâce aux réactions du patient. La sonde est alors retirée du guide, et on y glisse l'électrode souple à la place, avant de retirer le guide à son tour.
- La brachythérapie, ou curiethérapie en français, consiste à introduire une substance radioactive dans le corps. L'opération la plus courante est le traitement du cancer de la prostate, où le praticien implante des grains d'iode radioactifs (entre 60 et 100) à l'aide d'aiguilles dans tout le volume de la tumeur.

Il a été montré dans des essais cliniques que la précision des insertions étaient à mettre en relation avec l'efficacité du traitement [NCY+00] ou l'exactitude du diagnostic [AMS00]. La précision souhaitée est souvent inférieure au millimètre, alors que le praticien est confronté aux limitations de l'imagerie médicale, à la déformation des tissus ainsi qu'à la flexion de l'aiguille.

Les praticiens bénéficient des avancées dans le domaine des simulateurs en réalité virtuelle pour leur entraînement, le développement de nouvelles techniques opératoires et le contrôle de systèmes robotisés. Des simulateurs ont été proposés pour de nombreux gestes opératoires tels que la biopsie [AMS00] [KKK+01] [ZKX+98], la curiethérapie [MJK99] [WF04], l'insertion de cathéter [GTZZ00], la ponction lombaire [GKW+00] [DAS01], l'insertion de sonde [SK94], etc.

Dans tout le reste du chapitre et dans un soucis de lisibilité nous employerons le terme d'aiguille pour désigner en fait tous les instruments medicaux fins et flexibles qui peuvent être introduits dans des tissus et organes.

2.1.2 Étude des forces mises en jeu

La nature des interactions entre l'aiguille et les tissus rend complexe la compréhension des méchanismes de déformation, découpe et frottement. Une bonne connaissance des forces intervenant dans ces interactions est requise pour être le plus près possible du comportement observé, à la fois pour reproduire précisement la déformation des tissus, mais également pour rendre correctement la perception de ces forces par le praticien au moment de la conception d'une interface haptique.

Des études ont porté spécifiquement sur la mesure de ces forces, que ce soit dans un cadre clinique pour améliorer la compréhension du comportement des tissus, ou pour la simulation médicale. Lors d'opération percutanées, l'aiguille perce plusieurs couches de tissus comme la peau, les tissus musculaires, graisseux et conjonctifs. La force requise pour pénétrer et traverser ces couches est différente pour chacun d'entre eux, et varie d'un individu à l'autre selon l'âge, le sexe, la masse corporelle, les traitements médicaux antérieurs, etc. Abolhassani *et al.* [APM07] ont réalisé récemment un état de l'art très complet concernant l'insertion d'aiguille dans des tissus mous, qui englobe l'ensemble des problématiques de la communauté.

L'ensemble des travaux antérieurs s'accordent pour distinguer, lors des interactions entre l'aiguille et un tissu, trois principales forces :

- **Pénétration**. Lorsque l'aiguille entre en contact avec la membrane du tissu, celle-ci résiste d'abord et se déforme. L'aiguille va percer cette membrane lorsque la force qu'elle applique au niveau de la pointe atteint une certaine valeur, qui dépend à la fois du matériau constitutif de la membrane et de la forme de la pointe de l'aiguille. Cette valeur peut être mesurée, et correspond à la force de pénétration.
- **Découpe**. Lorsque elle progresse à l'intérieur d'un tissu, la pointe de l'aiguille découpe petit à petit le matériau. Le tissu y résiste et on observe une force au niveau de l'aiguille que l'on appelle la force de découpe.
- **Frottement**. Ayant été découpé par la pointe, le tissu est déplacé latéralement pour laisser passer le corps de l'aiguille. Il applique une pression sur toute la longueur de l'aiguille, et s'oppose au mouvement de celle-ci. On peut ramener ce phénomène à une force de frottement.

La somme des forces de découpe et de frottement peut être définie comme étant la force axiale. Elle est identifiée dans certains articles [HSS98, WUSC94, BHT00] qui ne la séparent pas encore en ses deux composantes. Westbrook *et al.* [WUSC94] repèrent l'instant où l'aiguille pénètre la membrane grâce au pic observé dans la mesure de la force axiale. Brett *et al.* [BHT00] mesurent cette force pour repérer les différents tissus rencontrés par l'aiguille lors de son insertion au niveau de la colonne vertébrale.

Simone et Okamura [SO02] ont été les premiers à séparer les mesures en plusieurs forces, afin de les modéliser. Leur banc de mesure consiste en une cellule de mesure de force fixée à une aiguille chirurgicale, le tout actionné par un robot à 1 degré de liberté. Les mesures ont été faites en insérant et retirant l'aiguille dans des foies bovins. Ils séparent leur étude en une phase avant la pénétration et une seconde une fois que l'aiguille a pénétré le tissu. Avant la rupture de la membrane, la force mesurée augmente progressivement avant de chuter brusquement lors de la pénétration de l'aiguille. Les variations de la force après cela sont la conséquence du frottement, de la force de découpe, de la raideur interne ainsi que la collision et la pénétration des structures internes.

Ils séparent ainsi la force mesurée à la base de l'aiguille en la somme des forces de raideur, de frottement et de découpe. Ils proposent un modèle permettant de prédire ces forces en fonction de la distance parcourue par l'aiguille :

$$f_{needle}(z) = f_{stiffness}(z) + f_{frottement}(z) + f_{cutting}(z) \tag{2.1}$$

La force de raideur correspond à la force de réaction due au contact de l'aiguille avec le tissu. Elle dépend des propriétés élastiques de l'organe et à celle de sa membrane. Simone et Okamura modélisent cette force à l'aide d'un ressort non-linéaire, et elle peut être entièrement décrite par :

$$f_{stiffness} = \begin{cases} 0 & z < z_1 \\ a_1 z + a_2 z^2 & z_1 \leq z \leq z_2 \\ 0 & z > z_3 \end{cases} \tag{2.2}$$

avec z la position de la pointe de l'aiguille et z_1, z_2 et z_3 définis tels que sur la figure 2.1. Les paramètres a_1 et a_2 sont extraits des mesures, et les auteurs remarquent qu'ils peuvent changer significativement d'un foie à l'autre.

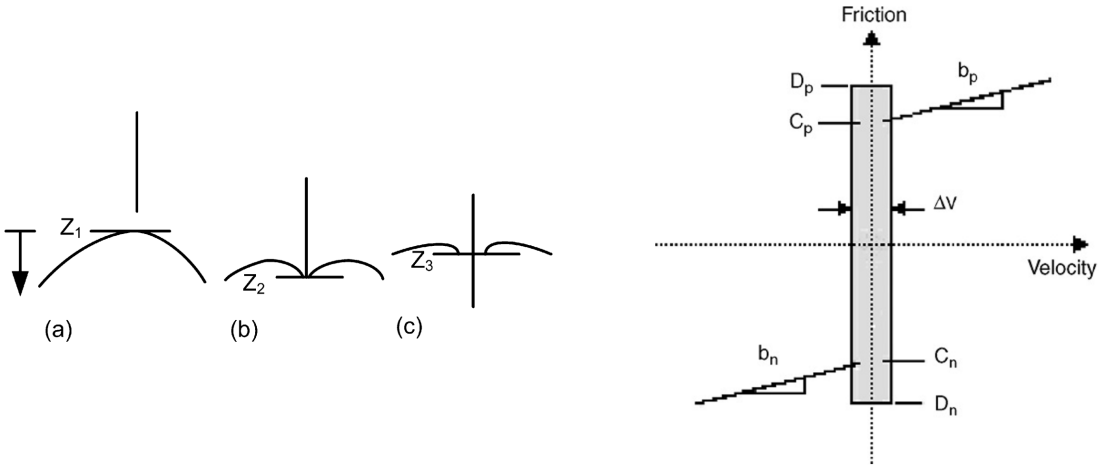


FIGURE 2.1 – (à gauche) Localisation de la surface du tissu avant, pendant et après la pénétration de l'aiguille. (à droite) Modèle de frottement de Karnopp modifié. [SO02]

La force de frottement apparaît tout le long de l'aiguille, et est la conséquence de l'adhérence des tissus ansi que de l'amortissement. Les auteurs choisissent le modèle de frottement de Karnopp [Kar85], qui inclut le frottement dynamique ainsi que le frottement statique avec une "zone morte" lorsque la vitesse est proche de zéro ; le frottement étant dans cette zone uniquement statique (voir figure 2.1).

$$f_{frottement} = \begin{cases} C_n \operatorname{sgn}(\dot{z}) + b_n \dot{z} & \dot{z} \leq -\Delta v/2 \\ \max(D_n, F_a) & -\Delta v/2 < \dot{z} \leq 0 \\ \min(D_p, F_a) & 0 < \dot{z} \leq -\Delta v/2 \\ C_p \operatorname{sgn}(\dot{z}) + b_p \dot{z} & \dot{z} \geq -\Delta v/2 \end{cases} \tag{2.3}$$

où C_n et C_p sont les valeurs négative et positive du frottement dynamique, b_n et b_p sont les coefficients d'amortissement, D_n et D_p les valeurs du frottement statique, $\dot{z}$ la vitesse relative entre l'aiguille et le tissu, $\Delta v/2$ est la valeur en dessous de laquelle la vitesse est considérée comme étant nulle, et F_a est la somme des forces autres que le frottement appliqué au système.

Enfin, la force de découpe correspond à la résistance qu'offre le tissu lorsque la pointe de l'aiguille coupe le tissu, et elle dépend donc en partie de la raideur du tissu. Dans l'idéal, la force de découpe est constante et ne varie pas en fonction de la profondeur de l'aiguille. Son amplitude est donnée par :

$$f_{cutting} = \begin{cases} 0 & z_{tip} \leq z_2, t < t_p \\ a & z_{tip} > z_3, t \geq t_p \end{cases} \tag{2.4}$$

où z, z_2 et $z3$ sont les mêmes que pour 2.2, t est le temps et t_p l'instant où l'aiguille a pénétré le tissu.

Leur modèle a été validé en le comparant aux mesures expérimentales (voir figure 2.2). La forme globale est similaire, mais on observe de grandes divergences au moment de la pénétration, qui peuvent être dues aux variations de géométrie du foie et à ses structures internes. Il est intéressant de remarquer également que la force requise pour la pénétration de l'aiguille varie du simple au double lorsque l'on passe d'un foie à un autre, montrant là toute la difficulté de l'identification de paramètres pour des modèles d'interaction. La force d'interaction dépendant donc des propriétés mécaniques du tissu traversé, la paramétrisation de l'interaction en utilisant une force n'est peut-être pas la méthode la plus adaptée. Ainsi, nous pensons qu'un modèle d'interaction basé sur des contraintes permettrait de calculer les forces d'interaction exactes selon les paramètres actuels de la simulation.

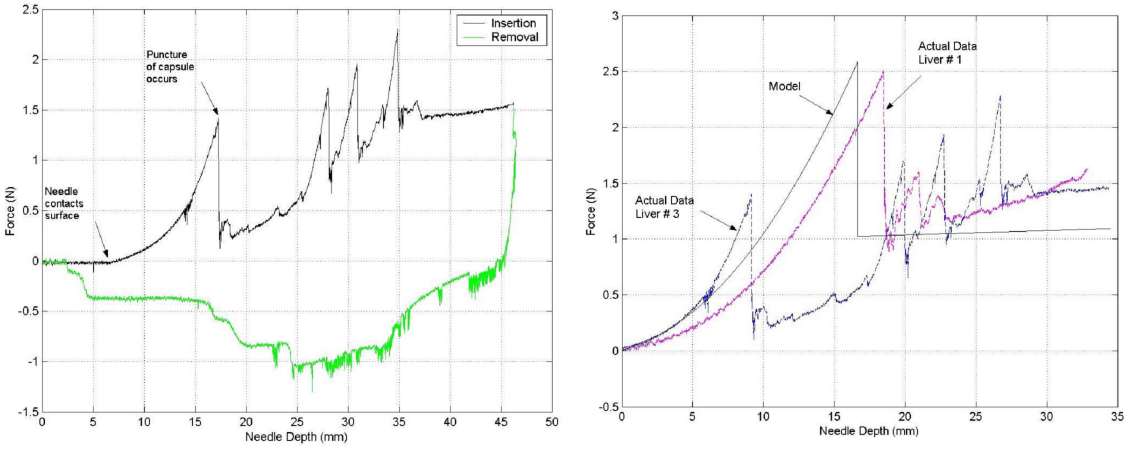


FIGURE 2.2 – (à gauche) Mesure de la force à la base de l'aiguille lors de l'insertion et le retrait dans un foie bovin. (à droite) Comparaison du modèle avec deux mesures expérimentales. [SO02]

Dans des travaux ultérieurs, Okamura *et al.* [OSO04] se sont intéressés à l'impact du diamètre de l'aiguille ainsi qu'à la forme de la pointe sur les forces mesurées lors de l'insertion. Ainsi, l'augmentation du diamètre de l'aiguille entraîne une augmentation des forces. De

la même façon, les forces augmentent si l'on passe d'une pointe triangulaire à une pointe biseautée, et encore plus pour une pointe conique. Cependant, l'angle du biseau n'a pas d'effet significatif sur la force dans le sens axial.

Kataoka *et al.* [KWAM01, KWC+02] ont réalisé des mesures sur une prostate d'un cadavre de chien, avec accès direct, les tissus environnants ayant été écartés. En utilisant une cellule de mesure de forces spécialement conçue (voir figure 2.3), ils ont été capables de mesurer indépendemment les forces agissant sur la pointe de l'aiguille, les forces de frottement et les forces de pincement. Cette dernière, mesurée dans la direction normale, est attribuée à la résistance du tissu qui est repoussé par le corps de l'aiguille. Elle dépend du diamètre de l'aiguille ainsi que de la forme de l'incision, et elle a une influence sur la force de frottement. La force mesurée au niveau de la pointe et dans la direction axiale est attribuée principalement à la découpe, et est affectée par la forme de la pointe de l'aiguille. Lorsque l'aiguille entre en contact avec l'organe, elle compresse sa surface jusqu'à ce que l'amplitude de la force atteigne le seuil nécessaire pour percer la membrane. Le comportement de la prostate comprimée par un point n'est pas linéaire, puisque la force mesurée augmente de façon exponentielle. Cette force décroit fortement une fois que la capsule de la prostate est percée, car celle-ci est plus résitante que le tissu interne. Ils considèrent que la force de découpe est constante une fois l'aiguille à l'intérieur de l'organe.

La force de frottement agissant sur le corps de l'aiguille et dans la direction axiale est considérée comme étant la somme d'un frottement de Coulomb et d'un frottement visqueux. Ce dernier peut être modélisé par le produit d'un coefficient d'amortissement et de la vitesse relative des deux matériaux. Celle-ci est mise en évidence lorsque l'aiguille s'immobilise dans le tissu et que la force mesurée décroit, alors que la surface de la prostate reste également immobile. La force de frottement augmente proportionnellement à la profondeur d'insertion de l'aiguille, et au diamètre de celle-ci.

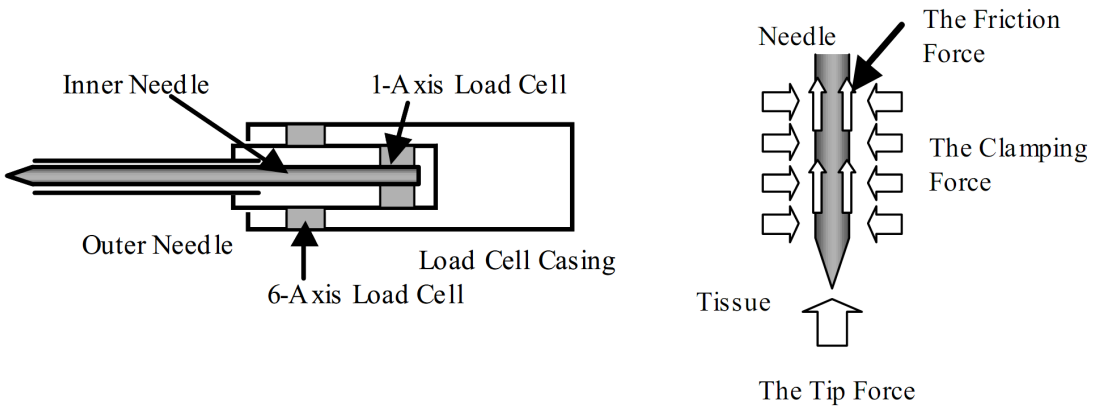


FIGURE 2.3 – La structure des capteurs de force, et les forces mises en évidence. [KWC+02]

Maurin *et al.* [MbB+04] ont étudié les forces d'insertion lors d'expériences *in vivo* sur des foies de porc. Leurs mesures sont particulièrement intéressantes car elles se démarquent de celles effectuées sur des fantômes de tissus artificiels ou des tissus issus de cadavres d'animaux, leurs propriétés bio-méchaniques étant très différentes. Ils comparent des insertions effectuées

par un robot ou par un praticien, et par une méthode par accès direct où les autres tissus ont été enlevés ou par accès au travers de la peau (voir figure 2.4). On observe facilement lors de l'insertion avec accès direct les différentes phases et leur impact sur la force mesurée :

- L'aiguille en contact avec la capsule du foie comprime celle-ci, la non-linéarité de l'elasticité des tissus entraînant une augmentation exponentielle de la force en fonction de la distance.
- Lorsque l'amplitude de la force atteint un certain seuil, la capsule est percée, et la force mesurée chute brusquement pour atteindre la valeur de la force de découpe, qui restera constante à partir de cet instant.
- La force de frottement augmente proportionnellement avec la profondeur de l'aiguille.
- Lorsque l'aiguille est immobilisée, on oberve une phase de relaxation, et la force d'insertion diminue progressivement. Durant cette phase la force de frottement visqueuse décroit alors que l'on n'observe pas de déplacement du tissu.
- Lors du retrait de l'aiguille, la force de frottement s'opposant au déplacement diminue rapidement et devient négative un bref instant avant que l'aiguille ne sorte complètement du tissu.

Les mesures montrent qu'un accès direct à l'organe permet une force d'insertion plus faible que dans le cas d'un accès percutané. On observe de grandes disparités au niveau de la force mesurée entre les insertions à la main et celles effectuées par un robot, qui montrent bien la difficulté de créer un modèle qui soit valide dans une majorité de cas de figure. Les auteurs ont comparé leurs mesures expérimentales avec le modèle de Simone et Okamura [SO02] ainsi que le modèle de Maurel [Mau99] basé sur le travail de Fung [Fun93], et obtiennent une erreur faible pour les deux.

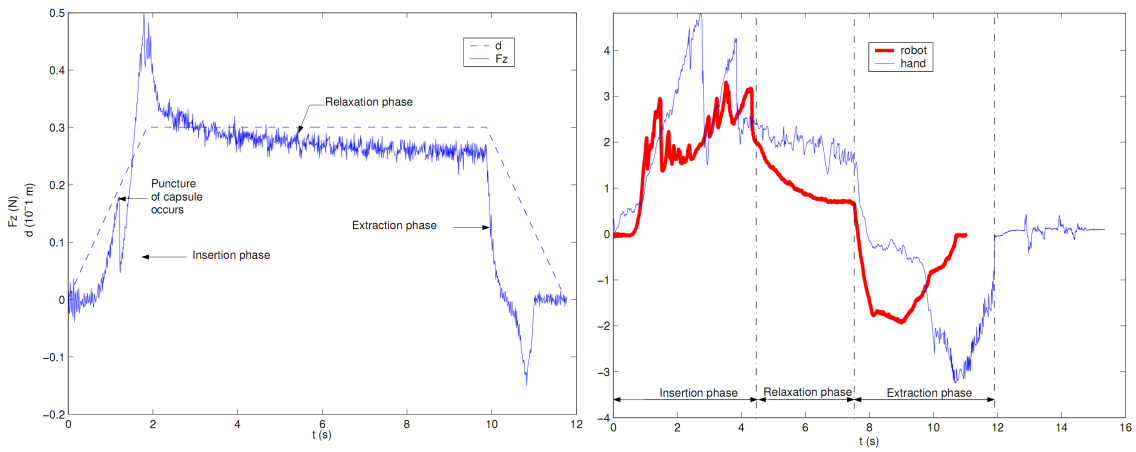


FIGURE 2.4 – (à gauche) Force mesurée à la base de l'aiguille lors d'une insertion *in vivo* dans un foie avec accès direct. (à droite) Comparaison de l'insertion par un radiologiste et par un robot. [MbB+04]

Podder *et al.* [PCS+06] ont été les premiers à réaliser des mesures *in vivo* sur des tissus humains. A l'aide d'une sonde équipée de capteurs 6D (forces et moments) et d'un capteur électromagnétique de position 6D, ils ont mesuré les forces, moments et position lors d'une insertion manuelle durant une procédure de curiethérapie. Ils montrent que la force d'insertion

maximale est de l'ordre de 15N pour le périnée, 7N pour la prostate, avec une force transverse d'environ 1,6N.

DiMaio et Salcudean [DS03a] se sont intéressés à la relation entre les forces d'insertion et les déformations en 2D du tissu. Ils effectuent des mesures sur un fantôme d'un tissu artificiel dans lequel des marqueurs ont été insérés pour être suivis par une caméra. Ils relèvent une distribution des forces le long de l'aiguille (voir figure 2.5) qui met en évidence deux forces : une force de frottement dont l'amplitude est uniforme sur tout le corps de l'aiguille et une force de découpe qui s'applique uniquement au niveau de la pointe, et dont l'amplitude est environ 30% supérieure à la force de frottement. Ils observent que la force de frottement augmente avec la vitesse d'insertion, alors que la force de découpe y est plutôt insensible. Les résultats de cette étude est compatible avec le modèle présenté par Simone et Okamura [SO02].

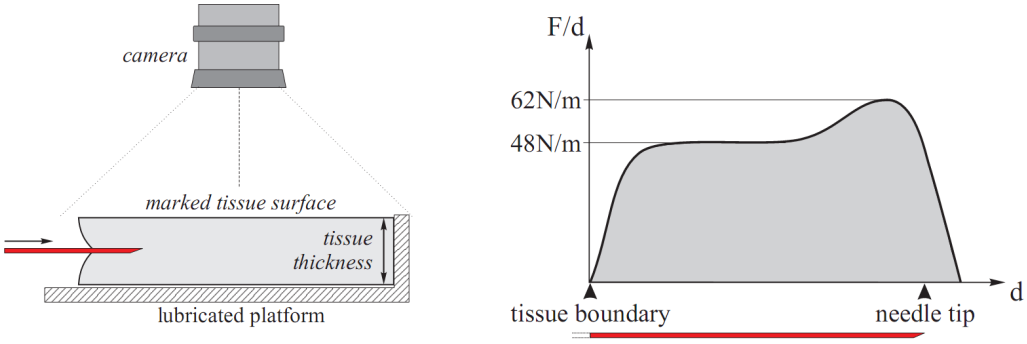


FIGURE 2.5 – (à gauche) Vue du profil du banc de mesure. (à droite) Densité de force estimée le long de l'aiguille. [DS03a]

Crouch *et al* [CSWO05] reprennent le principe du suivi par caméra de marqueurs implantés dans un fantôme transparent énoncé par DiMaio et Salcudean [DS03a]. En utilisant cette fois-ci plusieurs couches de marqueurs et deux caméras, à savoir une de profil en plus de celle du dessus, ils sont capables d'enregistrer les déformations en 3D du tissu. Ils se sont intéressés à la composante dynamique des forces agissant lors de l'insertion d'aiguille.

Hing *et al.* [HBD05, HBD07] utilisent pour leur part deux C-Arm fluoroscopes pour mesurer les mouvements de grains marqueurs dans le cadre de la curiethérapie de la prostate. Ils proposent de mesurer la force de découpe en soustrayant la force mesurée lors de la phase de retrait à la mesure de la force totale, puisque l'aiguille ne découpe plus les tissus lorsqu'elle est retirée. Les auteurs calculent, à partir de leurs mesures des forces et des déplacements des tissus, un module de Young local au moment de la ponction pour quantifier la résistance du tissu.

Dehghan *et al.* [DWZA+07, DWZA+08] utilisent une sonde à ultrasons pour mesurer les déformations en 3D, sans avoir à insérer des marqueurs dans le fantôme. Un modèle de profil de forces à 3 paramètres similaire à celui de DiMaio [DS03a] est proposé pour l'interaction entre l'aiguille et les tissus.

2.1.3 Généralités sur la suture

L'opération de suture, peut-être l'un des actes chirurgicaux les plus anciens [1], a de nombreuses applications ; la plus connue est certainement celle consistant à rapprocher puis à maintenir ensemble deux parties de peau ou d'un organe interne après qu'ils aient été séparés par une incision ou une blessure. La suture est également employée sur les vaisseaux sanguins et les conduits du corps humains, que ce soit pour restreindre le flux en les nouant ou pour les joindre ensemble. On se servira également de la suture pour fermer des trous ou des défauts dans des tissus, pour renforcer localement des tissus abimés ou aider à leur réparation en formant un échafaudage, ou encore à plisser les tissus pour réduire leur taille.

Selon les cas, les chirurgiens ont à leur disposition le fil de suture ou d'autres matériels tels que agrafes et adhésifs. Dans ce travail, nous nous intéresserons exclusivement à la suture à fils.

Nous relevons deux principales techniques pour la réalisation d'une suture : la suture interrompue et la suture continue. On appelle suture interrompue ou à points séparés une suture où le fil est passé un seule fois avant d'être noué, chaque point de suture étant indépendant des autres. Elle est facile à executer, permet une très bonne approximation des bords de l'incision, mais nécessite un grand nombre de nœuds. La suture continue est réalisée avec une seule longueur de fil, et n'est nouée qu'aux extremités. Elle est plus rapide à executer, leur inconvénient étant le rélachement de la suture si un nœud se défait. Une variante de la suture continue est le surjet à points passés, dont la configuration empêche les tensions latérales pouvant résulter d'un surjet simple. On appelle surjet la réunion des deux lèvres d'une plaie réalisée avec un seul fil.

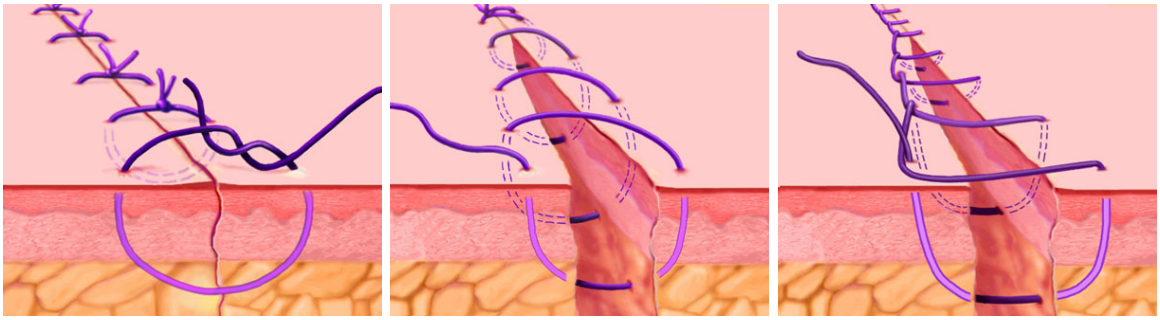


FIGURE 2.6 – Principaux types de suture, de gauche à droite : points séparés, suture continue, points passés. Images ©Ciné-Med

Les aiguilles employées pour les opération de suture ont une grande variété de tailles, de formes et de types de pointe. Le choix d'une aiguille en particulier se fera bien évidemment en fonction de l'application recherchée.

Contrairement aux aiguilles présentées précédemment, la flexibilité des aiguilles de suture n'est pas un paramètre ayant un impact sur le déroulement de l'opération. Dans un usage courant, l'épaisseur de l'aiguille est uniquement choisie afin que celle ci ne se brise pas durant

1. http://home.nordnet.fr/~acapon/Page2.html

la manipulation, mais est souvent assez faible pour qu'elle puisse se déformer légèrement. Pour les sutures vasculaires, on s'assure que le trou formé par l'aiguille est entièrement rempli par le fil de suture afin de minimiser les fuites, l'aiguille étant alors à peine plus épaisse que le fil.

Une aiguille de suture est en général courbée et forme un arc de cercle, souvent 1/4, 3/8, 1/2 ou 5/8 de cercle. Il existe cependant des aiguilles droites, ainsi que des aiguilles courbées uniquement sur la pointe (voir figure 2.7.a).

Les aiguilles de suture peuvent présenter plusieurs types de pointes. On utilisera une aiguille de découpe pour les tissus denses, solides, fibreux ou organisés, tels que la peau, le cartilage, les tendons et les ligaments. la pointe est alors triangulaire, vers l'intérieur ou vers l'extérieur de la coubure de l'aiguille, ou en forme de spatule, principalement pour les opérations ophtalmologiques. Pour les tissus moins denses ainsi que les vaisseaux on choisira une pointe conique, et on pourra éventuellement se servir d'une pointe non tranchante pour des tissus friables.

Il est à noter que pour éviter de devoir faire un nœud au niveau de l'aiguille, les praticiens utilisent souvent des aiguilles à usage unique, dans lesquelles est déjà serti le fil de suture (comme montré dans la figure 2.7.c). Ces aiguilles sont appellés atraumatiques, et ont un impact plus petit sur les tissus traversés que les aiguilles traumatiques à chas qui sont conditionnées séparement de leur fil.

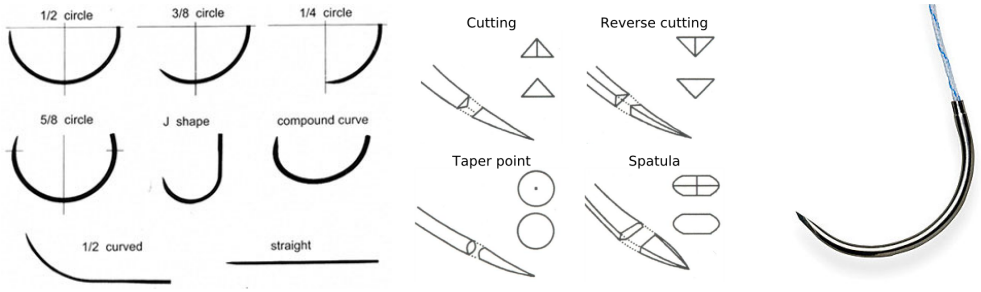


FIGURE 2.7 – Caractéristiques d'une aiguille de suture : sa forme, sa pointe, le système de fixation du fil. Images ©Shandong Sinorgmed, mrcophth.com et wikisurgery.com

Dans l'objectif de réaliser la simulation d'une opération de suture, nous voulons nous intéresser plus particulièrement aux propriétés du fil de suture. Nous pouvons distinguer trois caractéristiques : le matériau, le tressage et la taille. Les matériaux naturels tels que la soie et l'intestin de porc entraînant d'intenses réactions des tissus, ils ont été largement remplacés par des matériaux synthétiques. Ces fibres sont de longues chaînes formées par des molécules assez simples liées entre elles, telles que le nylon, le polyester et le polypropylène. En polymérisant des glucides, on obtient des fibres qui sont résorbables par les tissus. Puisque la suture est un moyen de fournir aux tissus une aide extérieure jusqu'à ce qu'ils soient suffisament solides pour prendre le relais, on emploiera des fils résorbables si la guérison totale des tissus est inférieure à trois semaines. Dans le cas contraire, et que la résistance du fil est requise pour plusieurs semaines ou mois, on privilégiera un fil non résorbable.

Un fil de suture peut être fait d'un seul filament, ou bien être tressé. Un fil monofilament est très lisse et présente ainsi un frottement faible à l'intérieur des tissus. Il est donc aisé de réaliser plusieurs points de suture, et le passage de ce fil endommagera très peu les tissus. Cependant, il est difficile à nouer et les nœuds formés peuvent facilement se défaire. Le fil monofilament est également assez élastique, et est sensible à la torsion après plusieurs points de suture. Le fil tressé présente quand à lui un frottement plus important ainsi qu'une meilleure souplesse. Il est plus facile à manipuler et à nouer, les nœuds ainsi formés tenant mieux. Par contre, son frottement le rend plus difficile à tirer à travers les tissus, qu'il peut endommager.

La taille des fils de suture varie également beaucoup, depuis le fil tressé frolant le millimètre de diamètre pour les opérations orthopédiques, jusqu'au fil monofilament d'une dizaine de microns utilisé pour les opérations de la cornée.

Delingette [Del98] avait déjà fait remarquer la necessité de prendre en compte la suture dans un simulateur médical. O'Toole *et al.* [OPK+99] ont évalué l'intérêt des simulateurs pour l'évaluation et l'entraînement des chirurgiens. Ils ont ainsi fait passer sur un simulateur de suture, qu'ils ne décrivent malheureusement pas en détails, des chirurgiens expérimentés en chirurgie vasculaire, ainsi que des étudiants en médecine. On voit que la performance moyenne des chirurgiens est significativement meilleure que celle des étudiants, principalement sur quatre paramètres mesurés que sont les dégats aux tissus, l'erreur totale, le temps mis à compléter les tâches ainsi que la distance parcourue par la pointe de l'outil. Après entraînement sur le simulateur, et avec des retours sur leurs performances, les utilisateurs montrent un certain progrès, celui des étudiants étant légèrement supérieur à celui des chirurgiens.

Depuis, de nombreux simulateurs médicaux ont intégré des tâches de suture, à des degrés de réalismes différents. Ainsi, des travaux ont porté sur la suture dans des simulateurs de chirurgie mini invasive laparoscopique [KcM00], [MPD05]. La suture de vaisseaux sanguins est une tâche complexe qui a également été étudiée et pour laquelle plusieurs simulations ont été créés [BMcLS01], [HBB+04], [R04], [ZGH+07]. Enfin, pour l'apprentissage du geste de suture en lui-même, Shinar *et al.* [SP08] a proposé la simulation d'une tâche arbitraire de suture simple pour la fermeture d'une plaie au niveau de la peau. Des modules de suture virtuelle sont proposés par plusieurs simulateurs commerciaux, principalement dans des simulateurs de laparoscopie tel que *LapSim*[2] et *LaparoscopyVR*[3]. Ils utilisent en général des modèles de fil uniquement géométriques, et simplifient grandement les interactions (voir chapitre 5). La réalisation de nœuds fait intervenir des configurations préenregistrées, et la simulation du serrage du fil pour fermer la plaie n'est pas physique.

2.2 Simulation physique

Nous allons maintenant et jusqu'à la fin de ce chapitre nous intéresser aux problématiques de la simulation médicale, d'un point de vue de l'informatique graphique. Nous aborderons ainsi les méthodes employées pour la simulation des objets rigides et des objets déformables, l'interaction entre objets et les méthodes de contraintes, et terminerons par l'étude de la simulation d'insertion d'aiguille.

2. http://www.surgical-science.com
3. http://www.cae.com/fr/healthcare/laparoscopy.asp

L'algorithme utilisé pour simuler le comportement d'un objet dépend de ses caractéristiques physiques. Dans la réalité, tous les matériaux sont déformables à un certain point ; cependant lorsque les propriétés d'un matériau le contraignent à des déformations d'amplitudes très faibles, on considérera qu'il est *rigide*. En simulation physique on pourra alors traiter ces objets différement. Habibi [Hab97] propose une classification des corps selon leur degré de déformabilité. Dans l'ordre : les cristaux, les corps solides déformables, les corps très déformables mais structurés, les corps non structurés (tel que le sable), les fluides, la fumée, et les flammes.

Il existe alors plusieurs approches pour aborder les milieux continus. D'une part, on utilisera l'*approche Lagrangienne* si on peut munir les objets d'une structure interne, les décomposer en particules dont on étudiera les positions, la vitesse et l'accélération. Les objets déformables auront généralement une configuration de référence à laquelle on se rapportera pour calculer les déformations courantes. D'autres objets déformables ne peuvent pas être dotés d'une structure interne, et leur étude se basera sur des informations locales propres à chaque élément infinitésimal constituant l'objet. L'*aproche Eulérienne* étudie alors la vitesse et l'accélération de l'objet en des points fixes de l'espace. Celle-ci est plutôt adaptée à l'étude des fluides ; dans le reste de cet exposé nous nous concentrerons sur les modèles lagrangiens possédant une structure interne.

Nous dinstinguons les objets rigides, qui feront l'objet du premier paragraphe, des objets déformables pour lesquelles nous détaillerons divers travaux les plus en relation avec notre simulateur. Les modèles déformables 1D, utilisés en particulier pour simuler aiguille et fil de suture, seront présentés plus en détails et pour les applications qui nous intéressent au chapitre 5.

2.2.1 Objets rigides

Les objets rigides, de par leurs propriétés physiques, possèdent des contraintes internes qui les empêchent de se déformer. Leur géométrie est ainsi fixée au cours du temps, et on les représentera à l'aide du formalisme de Newton-Euler en utilisant six degrés de liberté, trois en translation et trois en rotation. On sépare l'étude des translations et des rotations, en faisant intervenir le centre de masse $\mathbf{G}$ ainsi qu'un vecteur orientation $\mathbf{\Omega}$. La position du centre de masse est définie par le barycentre des points du solide, pondérés par leur masse dans le cas discret ou leur densité volumique de masse dans le cas continu. On concentre sur $\mathbf{G}$ la somme de toutes les forces qui sont appliquées à l'objet.

L'étude du mouvement en rotation fait intervenir le calcul du moment des forces. Une force $\mathbf{f}$ appliquée en un point $\mathbf{A}$ de l'objet engendre un moment définit par : $\mathbf{m} = \mathbf{GA} \wedge \mathbf{f}$.

La dynamique des corps rigides est alors régie par l'équation :

$$\begin{cases} \sum_{i=1}^{n} \mathbf{f_i} = m\ddot{\mathbf{G}} \\ \sum_{i=1}^{n} \mathbf{m_i} = \sum_{i=1}^{n} \mathbf{GA_i} \wedge \mathbf{f_i} = \dfrac{I\omega}{dt} \end{cases} \tag{2.5}$$

avec ω le vecteur vitesse de rotation instantanée du solide, et I la matrice d'inertie qui représente la distribution de masse dans le volume de l'objet. Pour un modèle discret, on la calcule à partir des masses m_i des points :

$$I = \begin{pmatrix} \sum_i m_i(y_i^2 + z_i^2) & -\sum_i m_i x_i y_i & -\sum_i m_i x_i z_i \\ -\sum_i m_i x_i z_i & \sum_i m_i(x_i^2 + z_i^2) & -\sum_i m_i y_i z_i \\ -\sum_i m_i x_i z_i & -\sum_i m_i y_i z_i & \sum_i m_i(x_i^2 + y_i^2) \end{pmatrix} \tag{2.6}$$

Pour un objet continu, d'un volume V et d'une densité de masse ρ :

$$I = \begin{pmatrix} \int_V \rho(y^2 + z^2)\ dv & -\int_V \rho xy\ dv & -\int_V \rho xz\ dv \\ -\int_V \rho xy\ dv & \int_V \rho(x^2 + z^2)\ dv & -\int_V \rho yz\ dv \\ -\int_V \rho xy\ dv & -\int_V \rho yz\ dv & \int_V \rho(x^2 + y^2)\ dv \end{pmatrix} \tag{2.7}$$

Dès lors, l'application d'une force $\mathbf{f}$ sur un objet rigide induit à la fois un déplacement de l'objet via la translation de son centre de masse $\mathbf{G}$, mais également un changement d'orientation selon le point d'application de la force ainsi que de l'inertie de l'objet.

À noter que l'on peut représenter la rotation sous la forme d'un quaternion $\mathbf{q}$, et que son évolution est donnée par : $\dot{\mathbf{q}} = \frac{1}{2}\omega\mathbf{q}$.

2.2.2 Objets déformables

Les objets déformables ont été étudiés depuis longtemps en mécanique et dans le domaine de l'ingénierie. La simulation des objets déformables est le sujet de nombreux travaux dans la communauté graphique comme le rappellent [GM97, NMK+06], ainsi que dans le domaine de la simulation médicale [MLM+05].

La réalisation d'un modèle informatique d'un corps déformable nécessite de passer par une étape de discrétisation. La différence entre les modèles tient alors principalement dans la façon de réduire le nombre infini de leurs degrés de liberté. Ainsi, lorsque la discrétisation est réalisée dès la conception du modèle, on parlera de modèles *discrets*. D'un autre côté, si la discrétisation n'intervient que pour mener à bien le calcul, en réalisant une sommation infinitésimale d'équations continues, alors on dira que ces modèles sont *continus*. Enfin, d'autres modèles se basent sur les équations continues de l'objet, mais ne les résolvent qu'en certains points de la structure, ce sont des modèles *semi-continus*.

Nous présenterons dans cette section les méthodes les plus couramment utilisées de chaque famille de modèles.

Les modèles discrets

Les modèles discrets ne présentent pas une structure continue, et se basent sur une modélisation par assemblage d'éléments discrets. Dans le cas le plus général, c'est un nuage de

particules qui interagissent localement entre elles. La méthode la plus connue est la modélisation par masse-ressort, qui prend compte de ces couplages locaux en liant des points voisins munis de masse par des ressorts ainsi que des amortisseurs.

Ces modèles sont largement utilisés dans la communauté graphique pour la simulation de vêtements [BHG92, BHW94, Pro95, VCMT95, LPC95, EWS96, BW98, HB00, VT00, BWK03], la simulation médicale [TGG00], l'animation faciale [TW90, LTW93, NT98, ZPS01]. Les systèmes masse-ressort ont également été intégrés dans des modèles d'animation de personnages [CHP89], de serpents [Mil88] ou encore de poissons [TT94].

Le principal avantage de cette méthode est sa facilité d'implantation. L'objet à simuler est défini comme un ensemble de points dotés d'une masse, lesquels sont reliés à leurs voisins les plus proches par des ressorts. Le choix des ressorts permet de définir le comportement de l'objet, au travers de la modélisation des énergies de déformation. Nous présentons ici trois des ressorts les plus couramment utilisés :

– **Les ressorts d'élongation**
Ces ressorts exercent une force directement proportionnelle à l'élongation mesurée. Pour un ressort liant deux points **A** et **B**, de longueur au repos l_0 et de raideur k, la force exercée par **B** sur **A** est donnée par :

$$\mathbf{f_{B\to A}} = k(|\mathbf{AB}| - l_0).\frac{\mathbf{AB}}{|\mathbf{AB}|} \tag{2.8}$$

Chaque ressort vérifie le principe physique de conservation de l'énergie : la somme des forces qu'il induit est nulle, il en est de même pour les moments. Pour les ressorts binaires reliant deux points, cela revient au principe d'action-réaction, et donc la force exercée par **A** sur **B** est de même intensité mais de sens opposé :

$$\mathbf{f_{A\to B}} = -\mathbf{f_{B\to A}} \tag{2.9}$$

– **Les ressorts amortis**
Les ressorts d'élongation amortis présentent les mêmes caractéristiques, plus un coefficient d'amortissement a, la force étant amortie proportionnellement à la vitesse $\dot{\mathbf{AB}}$ relative des deux points :

$$\mathbf{f_{B\to A}} = \left(k(|\mathbf{AB}| - l_0) + a\frac{\dot{\mathbf{AB}}.\mathbf{AB}}{|\mathbf{AB}|}\right).\frac{\mathbf{AB}}{|\mathbf{AB}|} \tag{2.10}$$

– **Les ressorts angulaires**
Ce type de ressort permet de contraindre trois points **A**, **B** et **C** à garder un certain angle α_0 :

$$\begin{aligned}
\mathbf{f_A} &= -k\left[\frac{\mathbf{BA}.\mathbf{BC}}{|\mathbf{BA}||\mathbf{BC}|} - cos(\alpha_0)\right]\frac{\mathbf{BC}}{|\mathbf{BC}|} \\
\mathbf{f_B} &= k\left[\frac{\mathbf{BA}.\mathbf{BC}}{|\mathbf{BA}||\mathbf{BC}|} - cos(\alpha_0)\right]\left(\frac{\mathbf{BC}}{|\mathbf{BC}|} + \frac{\mathbf{BA}}{|\mathbf{BA}|}\right) \\
\mathbf{f_C} &= -k\left[\frac{\mathbf{BA}.\mathbf{BC}}{|\mathbf{BA}||\mathbf{BC}|} - cos(\alpha_0)\right]\frac{\mathbf{BA}}{|\mathbf{BA}|}
\end{aligned} \tag{2.11}$$

Des travaux ont ajouté des contraintes supplémentaires sur les ressorts décrits ci-dessus. Ainsi, [Pro95] définit une contrainte sur l'élongation des ressorts, afin de limiter le comportement trop élastique des objets, sans avoir à augmenter le coefficient de raideur. [DRG03]

proposent d'ajouter un terme d'hystérésis afin de modéliser des comportements plastiques dans des simulations textiles. L'anisotropie des matériaux a été intégrée dans un modèle masse-ressort par [BC00].

Cependant, les modèles masse-ressort présentent plusieurs inconvénients. Tout d'abord, la résolution passe itérativement sur chaque point, en fixant ses voisins. Cette résolution explicite souffre d'un faible taux de convergence, en particulier lorsque le maillage est raffiné. De plus, les ressorts introduisent souvent des problèmes d'oscillation autour de la position d'équilibre. Des travaux ont toutefois utilisé une résolution implicite pour des modèles masse-ressort [KCC+00], augmentant de ce fait la complexité du calcul. Enfin, il est difficile d'identifier ces modèles avec le réel comme l'ont montré [VG98]. Le choix des paramètres est alors délicat, ceux-ci sont souvent obtenus par tâtonnements ou par l'utilisation d'algorithmes d'optimisation ou d'algorithmes génétiques [Lou94, DKT95].

Les modèles continus

Contrairement aux modèles masses-ressorts, le modèle éléments finis (FEM) est basé sur les équations de la mécanique des milieux continus. La faisabilité et le potentiel de cette technique ont été démontrés pour le domaine de l'animation par Terzopoulos [TPBF87]. Les premières performances temps réel pour la simulation chirurgicale ont été réalisées par Cotin *et al.* [CDA99], au prix d'un long précalcul.

Les méthodes par éléments finis sont une famille d'outil mathématiques permettant de résoudre les équations différentielles partielles. Ces techniques ont été définies précisement pour les sciences de l'ingénieur au cours des années 1960, et sont particulièrement utilisées pour l'étude des matériaux déformables. Leur principe est de discrétiser le domaine de continuité en un ensemble de sous-domaines discrets, ainsi que les fonctions d'interpolation associées permettant d'évaluer les intégrales de l'équation d'après les propriétés aux sommets des éléments. Les équations de la mécanique suivent une formulation faible, car elles ne sont pas définies localement mais pour la globalité de l'objet.

Cas statique

La simulation statique, souvent utilisée pour l'animation [Gou94], ne s'intéresse qu'à l'état d'équilibre du système, qui peut s'exprimer ainsi :

$$K(\mathbf{x})\mathbf{u} = \mathbf{f} \tag{2.12}$$

avec $K(\mathbf{u})$ la matrice de raideur de l'objet, $\mathbf{x}$ le vecteur position de l'ensemble des nœuds, $\mathbf{u}$ leur déplacement et $\mathbf{f}$ le bilan des forces.

Si on considère les forces comme étant proportionnelles aux déformations, celles-ci sont donc linéaires et K ne dépend pas du vecteur déplacement $\mathbf{u}$. Le système linéaire d'équations peut alors être résolu aussi bien par des méthodes directes via l'inversion de la matrice K, par décomposition LU ou QR, que par des méthodes itératives telles que gradient conjugué, Gauss-Seidel ou Jacobi.

Au contraire, lorsque les déformations sont non-linéaires, la matrice K doit être mise à jour en fonction du vecteur $\mathbf{u}$. Différentes méthodes itératives ont été proposées pour résoudre ce

système d'équations non-linéaire, comme les méthodes de la sécante, du point fixe, de Newton-Raphson, d'Aitken, de Broyden.

Dans le domaine des simulations médicales, des travaux [CDA99] ont montré que le comportement visco-élastique des organes demande des lois constitutives non-linéaires. Pour simuler un comportement quasi non-linéaire, les auteurs partent de la formulation linéaire, puis décomposent le déplacement en une partie normale et une partie tangentielle. Une approximation polynomiale du déplacement radial par rapport au déplacement axial permet de reconstruire un déplacement relatif à un comportement quasi non-linéaire.

La simulation statique des éléments finis a été utilisée par [GMTT89] pour la déformation de corps humain, et par [CDA96] pour la simulation chirurgicale. Une technique de condensation a été proposée par [BnC96] afin ne conserver dans les équations que les nœuds de surface, et ainsi réduire sensiblement la taille du système. Cette technique a été étendue par [WH04] pour séparer des zones de l'objet déformable.

Cas dynamique

Pour les simulations interactives, on désire garder les phénomènes transitoires, et l'équation du mouvement devient donc :

$$M\ddot{\mathbf{u}} + C\dot{\mathbf{u}} + K(\mathbf{u})\mathbf{u} = \mathbf{f} \tag{2.13}$$

avec M la matrice de masse et C la matrice d'amortissement.

Une grande partie des déformations non-linéaires sont dues aux rotations de l'objet ou des éléments qui le constituent. Dès les premiers modèles déformables [TPBF87], il a été proposé de séparer les déformations de l'objet dans son propre repère et son mouvement rigide qui ne provoque pas de déformations. Plus récemment a été proposée la méthode du corotationnel [CGC+02, MDM+02, HS04]. Celle-ci consiste à extraire les rotations de chaque élément afin de l'introduire dans le calcul des forces élastiques. Les modèles actuels [NPF05] peuvent donc être utilisés à grands déplacements ou grandes rotations, et avec une intégration implicite.

Les modèles semi-continus

Plusieurs approches essayent de tirer partie à la fois du réalisme des modèles continus ainsi que de la rapidité de calcul des modèles discrets. Ces modèles semi-continus se basent sur les équations continues de l'objet, mais ne les évaluent qu'en certains points de celui-ci, de telle façon à ce qu'elles restent locales.

La méthode des différences finies

De la même façon que les modèles éléments finis, les méthodes par différences finies cherchent à discrétiser une équation différentielle de telle façon à ce que son calcul soit réalisable par des méthodes numériques. Les équations continues sont ainsi approximées par leurs développements limités.

Terzopoulos [TPBF87] propose d'appliquer la relation fondamentale de la dynamique sur les nœuds d'une grille régulière dans laquel est plongé l'objet à simuler. Chaque nœud possède les informations physiques relatives au matériau ainsi qu'aux fonctions continues du modèle. L'énergie de déformation est ainsi approximée à l'aide de différences finies d'ordres un et deux, au niveau de chaque nœud. De la même façon, les équations physiques du mouvement doivent également être discrétisées en ces points.

Le modèle employé par [TPBF87] pour la simulation d'objets déformables a été enrichi pour prendre en compte la plasticité ainsi que la fracture dans [TF88, TWR88]. Ce modèle a été exploité dans la simulation chirurgicale par [BN95]. La méthode à été portée sur des grilles irrégulières [DDBC99], nécessitant alors des opérateurs discrets adaptés [DDCB00].

Les masses-tenseurs

Cette méthode est l'alliance des modèles masse-ressort et éléments finis. L'objet est représenté par un maillage tétrahédrique, sa masse étant répartie sur les nœuds. L'énergie de déformation est elle calculée en chaque tétraèdre, dont dérive la force élastique produite par ce tétraèdre sur un nœud. La force résultante sur ce nœud est alors la somme des contributions de tous les tétraèdres ayant ce sommet en commun.

Dans les modèles masses-tenseurs, l'énergie de déformation est continue contrairement aux masses-ressorts où son expression dépend de la discrétisation. De plus, le calcul de l'élasticité est réalisé grâce aux même fonctions d'interpolation que pour la méthode des éléments finis. Cependant, comme pour les masses-ressorts, l'intégration en temps est gérée par un schéma explicite de telle façon qu'on calcule le déplacement d'un nœud en bloquant tous ses voisins.

Le modèle a été étendu aux grands déplacements [PDA00] en prenant en compte les rotations. L'anisotropie a été étudiée, en petits déplacements [PLDA02] ainsi qu'en grand déplacements [PDA03].

Les Smoothed-Particle Hydrodynamics

Très utilisée en astrophysique, l'approche SPH (Smoothed-Particle Hydrodynamics) a été proposée dans la communauté graphique par Desbrun [DG96]. Son principal intérêt est la possibilité de créer une représentation multi-résolution du milieu [Des97], propriété qu'elle a hérité lors de sa transformation depuis un modèle continu.

Cette méthode concentre en un ensemble de points les champs scalaires et vectoriels physiques (densité de masse, force, vitesse, etc.), et se classe difficilement dans les approches présentées ci-dessus. Elle est plutôt adaptée aux fluides, mais des travaux s'intéressent à son application à la mécanique des solides [Hoo06].

Les modèles Point-Based

Dans le même esprit que les SPH, [MKN+04] présente un modèle adapté à la déformation d'objets solides. Les objets sont réprésentés par un nuage de points appelés phyxels, auxquels sont associés masse, volume et densité. Aucune information topologique n'existe entre les phyxels, on s'intéresse uniquement aux voisins les plus proches.

Le gradient de déplacement est calculé pour chaque point, en se basant sur les déplacements des points voisins, selon une regression de type *Moving Least Square*. L'énergie potentielle élastique est ensuite calculée depuis ce gradient, puis on peut en déduire les forces s'appliquant aux points.

[MKN+04] montre la possibilité d'utiliser un schéma d'intégration implicite, et propose de modéliser un comportement plastique. La simulation de fracture est proposée par [PKA+05]. Ce modèle a été reconstruit en partant des contraintes en déplacement pour calculer les déformations [AOW+08].

2.3 Interactions entre objets

Les objets que l'ont désire simuler sont rarement isolés, leur comportement doit généralement prendre en compte celui des objets qui leur sont voisins. Plusieurs objets peuvent ainsi entrer en collision et subir des chocs, ou être liés entre eux, par exemple.

Nous allons dans cette section présenter les techniques de contraintes permettant de répondre au contact ainsi qu'aux liaisons entre objets. La problématique de couplage entre objets sera également illustrée par quelques travaux récents.

Avant de répondre au contact, il faut détecter les collisions éventuelles par l'utilisation d'un algorithme de détection de collision. Pour ce travail de thèse, nous utilisons un algorithme existant, que nous présentons à la section 3.5.2. Pour un aperçu des différentes méthodes proposées pour la détection de collision, nous conseillons les états de l'art de Lin [LG98] et de Teschner *et al.* [TKZ+04].

2.3.1 Méthodes de résolutions des contraintes

Les objets d'une simulation peuvent donc entrer en collision, et être liés entre eux. Le concept de contraintes rend compte de ces collisions et de ces liaisons.

D'une manière générale, les contraintes enrichissent un modèle en lui imposant certaines conditions, qui peuvent être physiques, géométriques, ou purement formelles et mathématiques. Un exemple de contrainte géométrique est le point fixe : on contraint un nœud d'un objet à rester fixe dans l'espace, ou à suivre la position d'un autre nœud appartenant à un second objet. Les conditions limites dans les simulations sont un exemple des contraintes physiques. Enfin, une contrainte obligeant un quaternion à rester unitaire est une contrainte purement mathématique. On peut différencier également les contraintes holonomes, qui ne font intervenir que les degrés de liberté de l'objet, des contraintes cinématiques qui peuvent également faire appel aux dérivées premières par rapport au temps des degrés de liberté.

La simulation d'un système dynamique cherche à minimiser une fonction $f(\mathbf{x})$ de telle façon à trouver le minimum de l'énergie, avec $\mathbf{x}$ l'état du système. On exprimera une contrainte bilatérale par une égalité $C_b(\mathbf{x}) = 0$ alors qu'une contrainte unilatérale sera écrite sous la forme d'une inégalité $C_u(\mathbf{x}) \geq 0$. Les contraintes bilatérales servent en général à exprimer une liaison, tel que le point fixe : on désire que la différence entre la position des deux points contraints soit nulle. Les contraintes unilatérales sont utiles pour représenter les collisions

[MW88]. En effet, on veut empêcher l'interpénétration qui vérifierait $C_u(\mathbf{x}) \leq 0$, mais autoriser le contact ($C_u(\mathbf{x}) = 0$) et permettre aux deux objets de se séparer sans introduire d'adhérence ($C_u(\mathbf{x}) > 0$). Il faut noter que les contraintes unilatérales peuvent être transformées en contrainte bilatérales. Il s'agit alors de les activer uniquement lorsqu'elles sont violées et qu'il y a donc pénétration, puis de les désactiver dès lors que les objets se séparent.

On peut donc poser le système contraint à résoudre, avec f la fonction objectif qui doit être minimisée :

$$\begin{cases} \text{minimiser} & f(\mathbf{x}) \\ \text{sujette à} & \begin{cases} C_{b_i}(\mathbf{x}) = 0 & \forall i \in \{0..m\} \\ C_{u_j}(\mathbf{x}) \geq 0 & \forall j \in \{0..l\} \end{cases} \end{cases} \tag{2.14}$$

Nous allons par la suite présenter certaines des approches proposées pour résoudre ce système et intégrer les contraintes dans une simulation physique.

Pénalité

Witkin *et al.* [WFB87] ont proposé d'exprimer des contraintes géométriques via une formulation énergétique. Cette approche est plus connue sous le nom de méthode des pénalités [Coe99]. Une contrainte est représentée par une fonction $E(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ qui devient $E(\mathbf{x}) = 0$ si la contrainte est satisfaite. En dérivant cette fonction d'énergie selon $\mathbf{x}$ on obtient la force de la contrainte $\mathbf{f}$ qui accélère le système vers la configuration désirée.

L'avantage de la méthode des pénalités est son efficacité : les contraintes n'apparaissent pas comme des équations supplémentaires dans l'équation Lagrangienne du mouvement, mais agissent comme des forces additionnelles sur le système. Ainsi le système peut s'écrire sous la forme :

$$\text{minimiser} f(\mathbf{x}) + k.c(\mathbf{x}) \tag{2.15}$$

le terme c dépendant donc des contraintes unilatérales et bilatérales, k étant un facteur pondérant les effets des contraintes qui augmente l'énergie totale du système tant que les contraintes ne sont pas respectées. La fonction c peut être exprimée selon deux approches (voir [BC98] et le cours d'optimisation de Georgia Tech [4]) :

- **Les approches séquentielles**
  Cette technique est connue sous le nom de *Sequential Unconstrained Minimization Technique*, d'après les travaux de Fiacco et McCormick [FM68]. Le terme c sera calculé et mis à jour au cours des itérations afin de s'approcher de la solution, sans forcement réaliser les contraintes. On peut alors soit utiliser des *fonctions de pénalité* qui pousseront la solution vers un cas où les contraintes sont respectées, ou des *fonctions de barrière* qui empêcheront une solution valide de ne plus respecter une contrainte.
  Les fonctions de pénalité sont les plus utilisées, elles expriment typiquement le terme c avec une fonction au carré :

$$c(\mathbf{x}) = \sum \max(0, C_u)^2 + \sum C_b^2 \tag{2.16}$$

4. http://www.srl.gatech.edu/education/ME6103/Penalty-Barrier.ppt

Les termes sont donc décroissants au fur et à mesure que l'on s'approche de la contrainte. La solution peut passer par des états où les contraintes ne sont pas vérifiées, et sont aussi plus robustes que les fonctions de barrière puisqu'en pratique l'état initial n'est pas toujours valide. Cette méthode a été introduite dans la communauté graphique pour les contraintes bilatérales par [PB88a, PB88b], ainsi que pour la réponse à la collision [MW88].
On utilise souvent les fonctions inverses ou logarithmes pour les fonctions barrière :

$$-\sum C_b^{-1} \text{ ou } -\sum \ln(-C_b) \tag{2.17}$$

Les termes sont alors plus importants lorsqu'on arrive sur la contrainte. Les fonctions de barrière sont parfois préférées parce que bien qu'elles ne convergent pas, la solution proposée est toujours valide et elles nécessitent moins d'évaluations de fonctions que pour les pénalités. Cependant, la solution peut être bloquée dans un minimum local.

– **Les approches exactes**
Il s'agit de construire le terme c de telle façon à réaliser toutes les contraintes. En pratique, les fonctions de pénalité non différentiables sont les plus courantes, il faut alors développer des méthodes de programmation quadratique séquentielle pour reformuler le problème [5].

Quand dans la communauté graphique il est question de *méthodes par pénalité*, c'est en général pour désigner *les méthodes séquentielles à fonctions de pénalité*. Les termes ajoutés pour chaque contrainte sont dans ce cas leurs équations au carré, ce qui est équivalent à une énergie potentielle créée par un ressort. Cette méthode revient donc à modéliser chaque contrainte par un ressort, qui peut être introduit dans le système physique. Les contraintes multiples sont gérées intrinsèquement et il est garanti de trouver une solution ; celle-ci sera la meilleure solution au sens des moindres carrés, mais sans garantir pour autant que toutes les contraintes seront vérifiées.

Les méthodes par pénalité sont facile à implanter, mais présentent certains défauts. La constante de raideur du ressort est difficile à régler : trop faible la contrainte n'est pas respectée pendant longtemps ; trop grande on risque de créer un problème raide et des oscillations dans le système, qui demandent alors un amortissement. Un autre inconvénient est la perte de la précision, car elles violent le principe des puissances virtuelles [GPS02].

Projection

Une seconde méthode, classique dans les problèmes d'optimisation et souvent employée dans les simulations physiques est la projection. On effectue dans un premier lieu la minimisation simple de la fonction objectif, comme s'il n'y avait pas de contrainte dans le système. Une fois la solution obtenue, on vérifie l'état des contraintes : si l'une n'est pas respectée, on obtiendra par projection l'état le plus proche vérifiant cette contrainte. Chaque contrainte définissant un espace vectoriel en fonction de son équation, la solution est projetée tour à tour sur les espaces vectoriels de chacune des contraintes. Pendant la méthode de la projection, la vérification d'une contrainte peut entraîner la violation d'une autre contrainte. Cette méthode est donc préférée pour les cas simples où il y a peu de dépendances entre les contraintes. De

5. http://roso.epfl.ch/cours/optimisation_B/2002-2003/cours/cours06.ppt

plus, la projection provoque des changements brutaux pouvant poser problème avec l'intégration du système physique.

Une façon d'éviter ces changements brusques consiste à corriger les contraintes en modifiant l'accélération des points contraints, ce qui revient à introduire des forces permettant de vérifier ces contraintes. Ainsi, Platt et Barr [PB88b] proposent au travers de contraintes de réaction de corriger au niveau des forces la violation d'une contrainte. On réalise d'abord le bilan des forces au niveau du point qui viole cette contrainte. La force de réaction est alors calculée de telle façon à ramener le point sur la contrainte, mais prend également en compte la composante des forces qui pousse le point à violer encore plus la contrainte, de telle façon à la compenser. Le calcul fait intervenir l'écart entre le point et la contrainte, ainsi que sa dérivée par rapport au temps. La méthode permet de vérifier avec exactitude la contrainte, mais est limitée dans le sens où une seule contrainte peut être fixée par point de l'objet.

Gascuel [GCG94] utilise une méthode par projection pour simuler des corps articulés. Un premier bilan de forces ne prend pas en compte les contraintes, et l'intégration numérique calcule le nouvel état du système. Les contraintes n'étant plus vérifiées, une correction est réalisée à la fois au niveau des positions mais également de la vitesse afin d'assurer une dynamique homogène. L'algorithme itératif arrive à détecter si la solution ne converge pas, et s'arrête alors.

Promayon *et al.* [PBP96] se servent de la projection pour contraindre les déformations d'un objet, en déplacement ainsi qu'en volume. La position des particules est corrigée en calculant par projection le vecteur déplacement. L'algorithme est itératif, avec intégration numérique puis correction, en utilisant un critère d'arrêt de cohérence temporelle. Une fois que les positions des points ont été corrigées, les vitesses sont ajustées.

Les contraintes par projection ont été utilisées pour des systèmes multi-objets contraints par Yu et Chen [YC00]. Après intégration numérique, une méthode itérative de type Newton-Raphson corrige les coordonnées généralisées. En se servant de l'équation dérivée des contraintes holonomes, les vitesses sont également corrigées.

Méthodes de post-stabilisation

Les méthodes de post-stabilisation, de la même façon que les méthodes projectives, corrigent le système pour qu'il vérifie les contraintes, après l'intégration numérique. Cependant, elles ont été mises au point de telle façon à ne pas introduire d'instabilités dans le système dynamique.

Ascher *et al.* [ACR95] montrent que la différentiation des équations des contraintes transforme une équation différentielle algébrique en une équation différentielle ordinaire. Les auteurs proposent une méthode post-stablisation qui se base sur l'introduction d'invariants de telle manière à conserver le même problème. La technique est présentée avec plus de détails dans la thèse de Chin [Chi95].

Cline et Pai [Cli02, CP03] proposent une démarche pratique pour la simulation physique utilisant la définition plus large des travaux d'Ascher. Soit $\mathbf{p}$ la position d'un point après l'étape d'intégration numérique, g la fonction qui définit une contrainte. La contrainte n'ayant pas été prise en compte lors du bilan des forces, elle ne sera en général pas vérifiée : $g(\mathbf{p}) \neq 0$.

L'étape de stabilisation cherche un déplacement $\mathbf{dp}$ de telle façon que $g(\mathbf{p}+\mathbf{dp}) = 0$. Si $\mathbf{dp}$ est petit, on peut réaliser l'approximation suivante :

$$g(\mathbf{p}+\mathbf{dp}) \approx g(\mathbf{p}) + G(\mathbf{p})\mathbf{dp} \tag{2.18}$$

avec la matrice $G = \frac{\delta g}{\delta p}$. On voit que le déplacement doit satisfaire $G\mathbf{dp} = -g(\mathbf{p})$. La matrice G n'étant en général pas carrée, les auteurs proposent d'utiliser une pseudo-inverse :

$$\mathbf{dp} = -(G^T(GG^T)^{-1})g(\mathbf{p}) \tag{2.19}$$

Cette idée fonctionne dès lors que GG^T est inversible, ce qui est le cas pour les systèmes ne faisant appel qu'à des contraintes bilatérales. Les contraintes redondantes pouvant être créées après détection de collision rendent parfois la matrice singulière, ce qui nécessite une décomposition supplémentaire. La méthode proposée est efficace pour les petits déplacements, mais le système peut diverger s'ils deviennent plus importants et engendrent des erreurs de linéarisation [CP03].

Faure [Fau98] propose une technique basée sur la linéarisation des contraintes de déplacement. Il réalise une correction sur les accélérations puis une étape de post-stabilisation pour les vitesses et les positions. Un processus itératif utilisant la linéarisation de l'équation des contraintes corrige les positions jusqu'à ce que les contraintes soient vérifiées. Une méthode d'intégration numérique est proposée afin de s'affranchir de la post-stabilisation des vitesses.

Réduction cinématique

La méthode de la réduction cinématique propose de supprimer les degrés de liberté du point qui lui permettrait de violer la contrainte auquel il est soumis. Ainsi, tous les objets dans une simulation physique sont représentés mécaniquement par un ensemble fini de degrés de libertés. Les corps rigides en possèdent six, trois pour la translation afin de représenter les trois directions de l'espace, ainsi que trois pour la rotation. Les corps déformables sont eux définis par trois degrés de libertés pour la position de chacune des particules qui les composent.

Cette technique part de l'observation que les contraintes limitent le déplacement d'un point ou d'un objet dans une direction de l'espace. Le degré de liberté correspondant à cette direction, pour ce point donné, sera donc supprimé du système physique. L'ajout d'une contrainte dans le système allège donc celui-ci, tout en assurant que cette contrainte soit vérifiée, puisque les objets n'ont plus les dégrés de libertés nécessaires pour la violer.

L'application la plus commune de cette méthode concerne les contraintes de point fixe dans l'espace, pour lesquels on supprimera les trois degrés de libertés en position. Le point existe toujours pour la simulation des objets déformables, mais les forces qui lui sont appliquées n'ont plus d'effet sur lui. En supprimant uniquement l'une ou l'autre des directions de l'espace, on peut également créer des contraintes de plan fixe et de point coulissant sur une droite. Pour les objets rigides, il est possible de réduire de la même façon les rotations.

Cependant, en dehors des contraintes de point fixe et des rigides articulés, cette méthode est en général très délicate à appliquer. Si la direction d'une contrainte ne correspond pas exactement à un axe de l'espace global, on peut réaliser une transformation des degrés de libertés de l'objet que l'ont veut contraindre ; un ou plusieurs degrés de liberté transformés seront alors supprimés.

Multiplicateurs de Lagrange

La méthode des multiplicateurs de Lagrange est une des méthodes d'intégrations de contraintes les plus connues ainsi qu'une des plus efficaces. Il s'agit, contrairement à la technique de réduction cinématique, d'ajouter de nouveaux degrés de libertés au système. Chacun d'entre eux correspondra à l'intensité de la force permettant de respecter la contrainte qui lui est associée [Bar96]. La résolution devient semblable à celle d'un système non contraint, les contraintes étant assimilées aux degrés de libertés du système.

La méthode des statuts permet de généraliser les multiplicateurs de Lagrange aux contraintes unilatérales. On utilise un processus itératif : une première résolution est faite sans prendre en compte les contraintes unilatérales. On vérifie ensuite si certaine d'entre elles ont été violées, dans quel cas on les ajoute au système et l'on refait une résolution. On continue jusqu'à ce que l'ensemble des contraintes vérifie le théorème de Kühn-Tucker.

Dans le cas d'une résolution statique, l'équation de contrainte est directement intégrée au système. On ajoute ainsi à la fonction objectif $f(\mathbf{x})$ le terme $\lambda C(\mathbf{x})$, avec λ le multiplicateur de Lagrange. Puisque le système n'est exprimé qu'avec les positions des particules, il ne peut exister que des contraintes holonomes dans le cas statique. Platt et Barr [PB88a, PB88b] proposent une méthode dérivée de la descente de gradient pour calculer les minima de la fonction d'énergie. L'algorithme permet de gérer des contraintes multiples, dont des contraintes par inégalité. Les auteurs montrent qu'il est possible de combiner les multiplicateurs de Lagrange avec les méthodes par pénalité. Cotin *et al.* [CDA99] utilisent la méthode des multiplicateurs de Lagrange dans une simulation médicale en résolution statique.

Les multiplicateurs de Lagrange sont également utilisés dans le cas dynamique. On pose l'équation dynamique du système sous la forme $M\mathbf{a} = \mathbf{f}$, avec m la matrice de masse, $\mathbf{a}$ le vecteur des accélérations et $\mathbf{f}$ les forces. L'ajout des multiplicateurs de Lagrange transforme alors le système :

$$\begin{bmatrix} M & -L \\ L & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{e} \end{bmatrix} \tag{2.20}$$

les contraintes s'exprimant alors sous la forme : $L\mathbf{a} = \mathbf{e}$.

Dans le cas dynamique, les contraintes dépendent de la position des particules (contraintes holonomes), et potentiellement de leur vitesse (contraintes cinématiques). Cependant, les inconnues du systèmes sont les accélérations, et il va donc falloir modifier les équations des contraintes. Nous présentons deux techniques :

- **Différentiation**
  La méthode la plus simple est de différencier l'équation de contrainte par rapport au temps jusqu'à ce que les accélérations apparaissent dans l'équation [Bar96]. Ainsi, on dérivera deux fois les contraintes holonomes, une seule fois les contraintes cinématiques : $\ddot{C}_h$ et $\dot{C}_c$. L'inconvénient de la technique est que l'on perd la contrainte sur les positions et les vitesses. L'erreur numérique se propage donc et s'accumule au long de la simulation, puisqu'aucune correction n'est faite en dehors de l'accélération. La convergence de la solution ne peut donc pas être garantie avec cette technique. De plus, les équations des contraintes résolues ne sont pas équivalentes aux équations de contraintes posées.

– **Schéma de Baumgarte**
Baumgarte [Bau72] propose donc, pour les contraintes holonomes, d'utiliser conjointement les différentes différentiations dans une équation différentielle de contrainte :

$$\ddot{C}_h + \alpha\dot{C}_h + \beta C_h = 0 \tag{2.21}$$

Cette méthode corrige la dérive numérique que l'on peut rencontrer avec la différentiation simple, mais le problème résolu n'est une fois de plus pas le problème posé.
Ascher [Asc97] montre que pour un amortissement minimal sans phénomènes oscillatoires, il faut que les racines de l'équation caractéristiques soient négatives. Il propose donc que les coefficients soient définis par $\alpha = 2\gamma$ et $\beta = \gamma^2$ avec $\gamma > 0$.
Barzel et Barr [BB88] utilisent le schéma de Baumgarte pour un système de modélisation utilisant des contraintes dynamiques. Ils proposent les coefficients suivants : $\alpha = \frac{2}{\delta_t}$ et $\beta = \frac{1}{\delta_t^2}$ avec δ_t le pas de temps de la simulation.

Contraintes de complémentarité

Ce type de contraintes est basé sur une formulation des lois de contact selon un problème de complémentarité linéaire (LCP). Murty [Mur88] pose le LCP sous la forme :

$$\mathbf{w} - A\mathbf{z} = q \tag{2.22}$$

$$\mathbf{w} > 0 \perp \mathbf{z} > 0 \tag{2.23}$$

$\mathbf{w}$ et $\mathbf{z}$ étant les inconnues du système, A une matrice carrée et $\mathbf{q}$ un vecteur donné. L'équation 2.23 impose l'orthogonalité entre les contraintes $\mathbf{w} > 0$ et $\mathbf{z} > 0$, de la même manière que la loi de Signorini qui établit une relation d'orthogonalité entre la force de réponse au contact f et la distance de pénétration δ :

$$0 \leq f \perp \delta \geq 0 \tag{2.24}$$

Ainsi, lorsqu'il y a contact il faut une force de contact pour empêcher l'interpénétration ($f > 0$, $\delta = 0$), alors que quand les objets sont séparés, la force de réponse est nulle ($\delta > 0$, $f = 0$).

L'application du LCP aux contraintes unilatérales de contact est directe : $\mathbf{q}$ est le vecteur des distances d'interpénétration après une étape d'intégration du système sans prendre en compte les contraintes. Après résolution du LCP, $\mathbf{w}$ représente le vecteur des distances d'interpénétration, et $\mathbf{z}$ le vecteur des forces de contact. La matrice A doit exprimer une relation entre les forces et les déplacements induits par ces forces. Pour la réponse au contact, cette matrice, appelée opérateur de Delassus, est calculée à partir de l'opérateur $\mathbf{h}$ permettant de passer de l'espace des mouvements à celui des contacts, ainsi que de la matrice de compliance C :

$$A = \mathbf{hCh}^T \tag{2.25}$$

La matrice de compliance permet de prendre en compte le couplage mécanique existant entre différents points de contact. Dans le cas statique, C est l'inverse de la matrice de raideur du système $\mathbf{K}$: $\mathbf{C} = \mathbf{K}^-1$. En dynamique, elle dépend du schéma d'intégration :

$$\mathbf{C} = \left[\ddot{M} + \dot{B} + K\right]^{-1} \tag{2.26}$$

Baraff [Bar89] [Bar91] [Bar94] a proposé des contraintes de complémentarité pour le calcul des forces de réponse à la collision pour des objets rigides, en tenant compte du frottement. Des formulations à base de LCP ont également été utilisées pour résoudre les collisions d'objets volumétriques déformables [DDK06] et de nuages de points quasi-rigides [PPG04].

Plusieurs algorithmes, directs ou itératifs, ont été proposés pour la résolution des LCP. Nous citerons par exemple l'algorithme de Lemke [Lem65] qui est basé sur une méthode de pivot, et le Gauss-Seidel très fréquent (pas seulement pour le LCP) dans les simulations d'objets déformables. Duriez [DDK06] a montré qu'une simulation temps réel était possible avec une résolution utilisant un Gauss-Seidel, et les améliorations sur le calcul de la compliance apportées par Saupin [SDCG08, Sau08] ont permis de faire interagir simultanément un grand nombre d'objets, rigides et déformables.

Cependant, la résolution d'un LCP est complexe et comme l'a montré Kaufman *et al.* [KEP05], la solution résultante n'est pas nécessairement plus correcte qu'une autre approche.

Bilan de la résolution des contraintes

Les méthodes que nous venons de présenter ont toutes des avatantages et des inconvénients qui les rendent attrayantes pour certaines applications et peu pratiques dans d'autres cas. Nous proposons de revenir rapidement sur les points importants de chacune de ces techniques.

Les méthodes de pénalité permettent de s'approcher de la solution, quelque soit l'état de départ, mais n'assurent pas que les contraintes seront vérifiées. Elles sont faciles à implanter, la résolution est simple car le système contraint est transformé en un système non contraint par l'ajout de termes directement à la fonction objectif. Cette méthode est souvent employée pour la réponse à la collision car elle est rapide et gère automatiquement les systèmes multi-contraints. Cependant, elle n'est absolument pas robuste et ne convient pas forcement dans le cas général.

Les méthodes par projection vérifient exactement les contraintes. Cependant, elles peuvent se révéler instables car elles perturbent le système dynamique et l'intégration numérique. Les contraintes multiples sont difficilement prises en compte, principalement lorsqu'il existe des dépendances entre certaines contraintes. En effet, la résolution d'une contrainte peut violer une contrainte voisine, et il faudra itérer pour essayer de converger vers une solution médiane.

Les contraintes sont également vérifiées exactement par les méthodes post-stabilisation. Les calculs, qui peuvent être relativement complexes, font intervenir des linéarisations qui rendent la méthode inefficace pour les grands déplacements.

La réduction cinématique réalise toutes les contraintes posées, en supprimant les degrés de libertés qui pourraient les violer. Cependant, cette méthode n'est souvent utilisée que pour certains cas particuliers, en dehors desquels elle demande une redéfinition des degrés de liberté qui peut être très lourde, d'autant plus que le nombre de contraintes est grand.

Les multiplicateurs de Lagrange vérifient toutes les contraintes, sans processus itératif. Cette méthode est largement employée, aussi bien dans les simulations statiques que dynamique. Elle peut être assez coûteuse en temps de calcul, et la définition des contraintes peut poser des problèmes dans le cas dynamique. L'utilisation du schéma de Baumgarte permet de gérer des contraintes multiples, et contradictoires.

Enfin, les contraintes de complémentarité résolues par LCP permettent de trouver les forces de contact exactes, au prix d'un calcul plus long que les autres approches. Des travaux récents [DDK06] ont intégré le cône de frottement sans avoir à le facétiser, et des optimisations ont permis de réaliser des simulations interactives.

2.3.2 Couplage entre objets déformables et rigides

Dans nos simulateurs médicaux, nous désirons faire interagir des objets de type différents : objets rigides pour les instruments médicaux, objets déformables 3D pour les organes, objets 1D pour les fils chirurgicaux, coques élastiques pour les vaisseaux sanguins par exemple. Cependant, il est délicat de faire cohabiter des modèles dont le formalisme est différent. En effet, là où on représente en général les corps déformables par un ensemble de particules ayant chacune trois degrés de liberté, les rigides n'ont que six degrés de liberté pour tout l'objet. De plus, la simulation physique utilise des algorithmes très différents, et la résolution numérique ainsi que l'intégration peuvent être optimisés pour certains types d'objets. Nous allons présenter ici quelques-uns des travaux qui se sont interessés à faire cohabiter au sein de la même simulation des objets mécaniques hétérogènes, et à gérer leurs interactions.

O'Brien *et al.* [OZH00] s'intéressent à l'interaction de modèles hétérogènes dans l'animation, à un niveau de description élevé. Ils étudient les *mouvements secondaires* des objets dont le comportement dépend fortement d'un autre objet animé (cheveux suivant la tête d'un personnage, trampoline se déformant sous le poid du gymnaste, robe d'une danseuse, etc.). Les auteurs proposent de choisir, selon le réalisme demandé et les interactions en question, entre des approches unidirectionnelles, bidirectionnelles ou hybrides. L'interaction en elle-même utilise la méthode des pénalités.

Baraff [BW97] propose une technique pour utiliser dans une même scène différents simulateurs et placer des contraintes entre eux. Un simulateur de particules, un simulateur de tissu déformable et un troisième simulateur d'objets rigides offrent une interface commune pour accéder aux positions, vitesses et forces des points internes des objets simulés. Une approche basée sur les multiplicateurs de Lagrange évaluent les violations des contraintes et fournissent les distributions d'énergie nécessaires pour vérifier les contraintes. Pour prendre en compte les multiples systèmes indépendants, le calcul est conduit de manière itérative en considérant statiques tous les systèmes autre que celui-ci sur lequel est réalisé le calcul.

Cette technique de simulations entrelacées, où chaque simulation utilise les résultats de la précédente en alternant alors les interactions unidirectionnelles peut amener à des problèmes de stabilité. En effet, un système ne peut pas prévoir avec précision ce que les autres vont faire, un peu à la manière d'une intégration explicite. Plusieurs astuces permettent d'accroître la stabilité en rendant le système semi-implicite. Ainsi, Sifakis *et al.* [SSIF07] prennent en compte implicitement l'effet des forces d'amortissement entre les modèles déformables et rigides, puis réappliquent l'inertie au simulateur d'objets rigides.

Une autre approche est d'utiliser le même type de simulation pour tous les phénomènes que l'on désire simuler, au risque que l'un d'eux souffre d'une simulation sous-optimale. Ainsi, Jansson et Vergeest [JV03] utilisent le formalisme des masses-ressorts pour représenter à la fois objets déformables ainsi qu'objets rigides. Les interactions intervenant uniquement entre les points de masse, les cas rigide-rigide ou rigide-déformable n'ont pas besoin d'être traités

différement. Les ressorts utilisés pour la réponse au contact peuvent générer des problèmes raides dans le cas de contraintes rigides, telles que les articulations d'objets rigides. D'autres systèmes entièrement couplés ont été réalisés de cette manière, citons l'utilisation de la méthode des SPH pour la simulation des solides et des fluides [MST+04], et la représentation des tissus déformables comme un réseau articulé d'objets rigides [Ben07].

Meseure *et al.* [MDH+03] ont proposé un simulateur physique pouvant rendre compte du comportement d'un grand nombre d'objets, en combinant plusieurs formalismes. Les collisions sont alors résolues par la méthode des pénalités. Lenoir *et al.* [LF04] enrichissent le simulateur en représentant les interactions par des contraintes par multiplicateurs de Lagrange.

Shinar *et al.* [SSF08] proposent une méthode de couplage qui conserve les avantages et les spécificités de chaque formalisme. Les auteurs décrivent un schéma d'intégration unifié, où les objets rigides et déformables évoluent de manière synchrone en utilisant des algorithmes entièrement couplés à chaque instant. La méthode permet un très grand réalisme, mais sa compléxité ne la rend pas adaptée à une simulation interactive.

La méthode utilisée par Saupin [SDCG08, Sau08] pour représenter le contact par des contraintes de complémentarité gère indifférement les objets 3D rigides ou déformables. Une linéarisation d'une fonction de "mapping" permet de poser le problème à résoudre dans l'espace des contraintes, indépendemment des degrés de libertés dans l'espace des mouvements. Cet opérateur est redéfini pour chaque type de modèle à simuler, afin d'appliquer les forces calculées par la méthode des contraintes sur les degrés de libertés effectifs des objets.

La simulation de fluides fait intervenir de manière générale des modèles non maillés, parfois selon une approche Eulerienne. Les modèles de fluides étant hors du domaine d'application de ce travail, nous ne détaillerons pas les travaux s'y rattachant. L'interaction des fluides avec d'autres objets a fait l'objet de nombreuses recherches, que ce soit pour les objets rigides [AGHmD03, CMT04, KAG+05, KFCO06, BBB07, RMSG+08], les modèles déformables [MST+04, CGFO06], les coques [GSLF05, RMSG+08], ainsi que l'interaction de plusieurs fluides entre eux [LSSF06].

2.4 Simulation d'insertion d'aiguille

Une simulation médicale utilise des modèles d'objets déformables qui peuvent rendre compte du comportement des organes simulés, les faisant interagir entre eux dans le but de reproduire les gestes du chirugien en opération. Nous aborderons maintenant la simulation d'insertion d'aiguille, en nous intéressant principalement aux interactions entre une aiguille, rigide dans un premier cas, déformable pour certaines applications, et les organes dans lesquels elle est insérée qui sont eux le plus souvent déformables.

2.4.1 Interaction de l'aiguille avec un tissu mou

Les premiers simulateurs d'insertion d'aiguille ne se sont pas basés sur des mesures expérimentales et ont dû proposer des modèles heuristiques pour estimer la force ressentie par l'utilisateur [XWS+98, ZKX+98]. Plusieurs travaux ont utilisé des tables mettant en relation

une force mesurée par rapport à une distance de pénétration donnée. Pour ajouter à leurs simulateurs un retour d'effort haptique, Wang et Fenster [WF04] ainsi que Hing *et al.* [HBD05] ont proposé des modèles 1D pour calculer la force ressentie par l'utilisateur en fonction de la position de l'aiguille, en se basant sur des études préalable de la déformation du tissu et des forces mises en œuvre. Les modèles que nous allons étudier par la suite ont pris en compte l'interaction de l'aiguille avec les tissus ainsi que leur déformation pour estimer le plus exactement possible les forces intervenant dans le système.

DiMaio et Salcudean [DS02a] ont été les précurseurs de la simulation d'insertion d'aiguille en 2D. Ils ont modélisé le tissu par un modèle éléments finis triangulaires quasi-statique en élasticité linéaire avec une méthode de condensation, et ont considéré dans un premier temps l'aiguille comme étant rigide. Pour faire interagir l'aiguille avec le tissu, les nœuds les plus proches appartenant au tissu sont simplement contraints le long de l'aiguille. Cela est fait sans mettre à jour le modèle mécanique, ce qui peut engendrer des déformations locales non réalistes. Pour augmenter la précision, ils raffinent donc localement le modèle géométrique en appliquant un niveau de subdivision avant d'adapter le maillage à la configuration de l'aiguille. Ils appliquent ensuite directement à chaque nœud du tissu contraint sur l'aiguille une force dont la valeur dépend de sa distance par rapport à la pointe. La force appliquée aux nœuds près de la pointe est plus importante que sur le reste du corps de l'aiguille, pour rendre compte de la force de découpe qui y est exclusivement localisée. Ils atteignent une grande précision, au prix d'une phase de calibration où le profil de force le long du corps de l'aiguille est estimé en fonction des déformations observées sur le tissu lors d'une expérience d'insertion dans un fantôme. Cette mesure est effectuée à l'aide d'une surface paramétrée où des marqueurs doivent être insérés pour être suivis par des capteurs, et n'est pas transposable pour des opérations *in vivo*.

Dans les version suivantes de leur méthode [DS02b, DS02c, DS03a], DiMaio et Salcudean intègrent un automate pour gérer les états de frottement statique et dynamique des nœuds du tissu contraints sur le long de l'aiguille. Le seuil de force déterminant le changement d'état est donné par la lecture du profil de force estimé expérimentalement. Lorsqu'un nœud « glisse », la force seuil lui est appliquée et une condition limite en force est créée dans la direction de l'aiguille. Si il est dans l'état « collant », le nœud est contraint en déplacement à rester à un point fixe sur l'aiguille. Le même profil de forces est utilisé quelque soit la vitesse d'insertion de l'aiguille, alors qu'il a été mesuré expérimentalement pour une vitesse constante de $1 mms^{-1}$. Cela est d'autant plus étonnant que les mêmes auteurs ont montré que la densité de force mesurée augmentait avec la vitesse d'insertion.

On retrouve des éléments clés de leur méthode dans un grand nombre de travaux ultérieurs, à savoir la subdivision [NvdS04, CAR+09], le repositionnement des nœuds du tissu sur l'aiguille [APT+03b, NvdS04, GSD+05, DWZA+08], le profil de force [DWZA+07] ou encore le modèle « glissant-collant » de frottement [AGO05, GSD+05].

Alterovitz *et al.* [APT+03b, APT+03a] proposent, pour une simulation de curiethérapie de la prostate, un modèle alternatif reposant sur un nombre très réduit de paramètres, tels que le frottement de l'aiguille, le tranchant et la vitesse d'insertion. Ces paramètres peuvent être en partie selectionnés par le praticien et correspondent aux données recueillies par Simone et Okamura [SO02]. Le tissu est modélisé par un modèle éléments finis linéaire et l'aiguille considérée comme rigide n'est pas modélisée par un objet maillé séparemment. A tout instant durant l'insertion de l'aiguille, un nœud du tissu est contraint en position à la pointe de l'ai-

guille. Celle-ci avançant, ce nœud devient un nœud gérant le frottement pouvant glisser le long de l'aiguille, et un autre nœud du tissu est alors déplacé pour être à la pointe où s'applique la force de découpe. Le modèle mécanique n'est pas mis à jour durant ces opérations, engendrant des déformations non réalistes et éventuellement des inversions d'éléments. Lorsque la norme de la force à la pointe dépasse la force requise pour couper le tissu, l'aiguille peut avancer. Le frottement est modélisé avec la méthode de Baraff et Witkin [BW98], en considérant la force normale comme étant constante puisque l'épaisseur de l'aiguille est constante et que le tissu étudié est uniforme. La membrane entourant la prostate est modélisée séparément par un polygone, utilisé alors pour gérer la pénétration de l'aiguille. Les auteurs se servent de la même méthode que pour la découpe mais en prenant en compte une force qui est plus importante pour découper la membrane que pour les tissus. La méthode gère le retrait de l'aiguille, en ne maintenant pas de nœud à la pointe de l'aiguille, et en libérant progressivement les nœuds de frottement lorsqu'ils glissent au delà de l'aiguille.

Goksel *et al.* [GSD+05] ont repris en 3D les travaux de DiMaio et Salcudean [DS03a]. Les auteurs utilisent deux techniques de remaillage, le repositionnement et l'ajout de nœuds dans le tissu sur le chemin créé par l'aiguille. La déformation de l'aiguille est modélisée par des éléments finis symétriques, mais l'interaction avec le tissu est basée sur la ligne de segments qui forment sa fibre moyenne. L'aiguille pose des conditions limites sur le déplacement du tissu via les nœuds de contact qui sont le long de celle-ci. Les forces de réaction du tissu sont ensuite appliquées au modèle de l'aiguille pour tenir compte de sa déformation et pour trouver les forces haptiques transmises à l'utilisateur. Il est à noter que le couplage tissu - aiguille est amorti afin de garantir une bonne stabilité. En utilisant l'algorithme de Woodbury pour la mise à jour de la matrice K^{-1}, les auteurs obtiennent un pas de temps de simulation compatible avec l'haptique (c'est à dire entre 300Hz et 1000Hz). Les auteurs présentent un simulateur de curiethérapie de la prostate (voir figure 2.8).

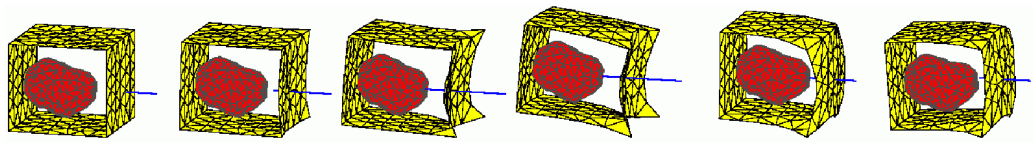


FIGURE 2.8 – Simulation 3D de l'insertion d'une aiguille dans la prostate. [GSD+05]

Nienhuys et van der Stappen [Nie03, NvdS04] ont réalisé la simulation en 3D de l'insertion d'aiguille dans un matériau non-linéaire. Pour éviter de manipuler une matrice de flexibilité trop lourde résultant d'un maillage uniforme nécessaire pour la méthode de condensation, les auteurs décident d'utiliser des algorithmes itératifs qui n'ont pas besoin de structures de données précalculées. Ils utilisent une méthode de subdivision adaptative simple car ils n'ont pas besoin de représenter exactement la surface de contact entre l'aiguille et le tissu, qui est prise en compte pour le frottement. Ainsi, les tétraèdres proches de la fibre moyenne de l'aiguille sont considérés comme faisant partie de celle-ci et la frontière de cet ensemble de tétraèdres est projetée sur l'aiguille. Son aire est alors utilisée pour calculer les forces de frottement qui sont intégrées à un modèle de frottement « glissant-collant » quasi-statique. Bien que leur prototype 2D atteigne des performances permettant un simulateur haptique, les coûts de calcul de la simulation 3D sont trop importants.

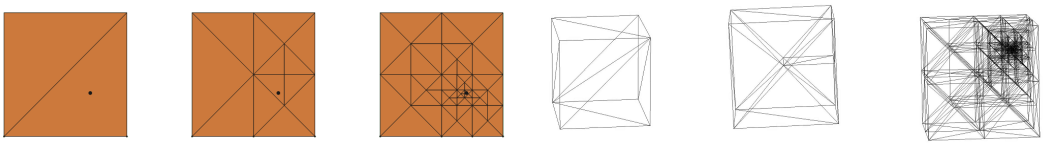


FIGURE 2.9 – Subdivision récursive des éléments. De gauche à droite : maillage initial en 2D et avec 5 et 10 subdivisions, puis maillage en 3D avec 0, 3 et 20 subdivisions. [NvdS04]

Glozman et Shoham [GS04] proposent l'utilisation d'un ensemble de ressorts virtuels distribuées le long de la courbe de l'aiguille pour modéliser la réaction du tissu dans la direction normale, ainsi que des forces de frottement dans la direction tangente (voir figure 2.10). Le système est linéarisé à chaque pas de temps afin de mettre à jour les coefficients des ressorts qui sont fonction de la déformation du tissu, comme un module de Young local représentant la raideur du tissu. Ces coefficients sont déterminés expérimentalement ou en fonction d'images médicales préopératoires, en estimant les valeurs pour les organes en fonction des connaissances antérieures. Les auteurs ne précisent pas leur calcul des forces de frottement.

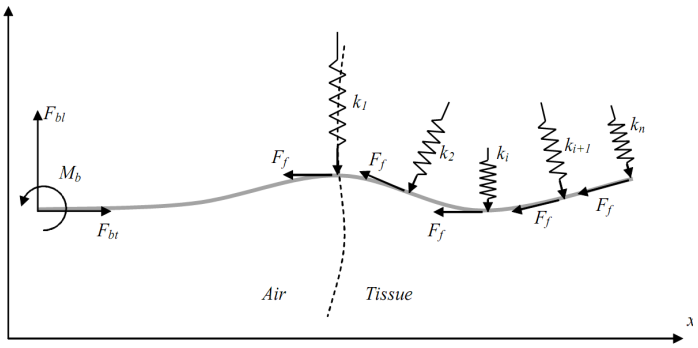


FIGURE 2.10 – Modèle d'interaction par ressorts virtuels : la réaction du tissu est modélisée par un ensemble de ressorts latéraux virtuels distribués le long de l'aiguille et des forces de frottement tangentielles. [GS04]

Crouch *et al* [CSWO05] ont étudié la déformation des tissus par rapport au temps et à la vitesse d'insertion de l'aiguille. Les forces d'interaction de l'aiguille sont donc modélisées de manière dynamique en fonction de la vitesse d'insertion relative de l'aiguille. Le tissu est représenté par un modèle éléments finis avec un matériau élastique linéaire, l'aiguille étant rigide. Des nœuds répartis le long de l'aiguille (espacés de 3 mm au niveau de la pointe, puis 10 mm pour le reste du corps) appliquent des forces calculées par un modèle de spline cubique. Le chemin emprunté par l'aiguille est connu à l'avance, et la résolution du modèle 3D représentant le tissu, spécialement créé, augmente lorsqu'on s'approche de l'aiguille. Les auteurs montrent via une série d'expérimentations que leur modèle reproduit de manière réaliste les forces mesurées lors du mouvement de l'aiguille ainsi que lors de la phase de relaxation des tissus lorsque celle-ci ne bouge plus. Cependant, ils conseillent l'utilisation d'un modèle viscoélastique pour représenter fidèlement la déformation des tissus lors de cette relaxation.

Marchal *et al.* [MPT06, Mar06] sont à notre connaissance les seuls qui ont simulé l'insertion d'aiguille dans un modèle 3D sans maillage. Ce modèle discret est basé sur le principe de la mémoire de forme, et des contraintes placées sur les nœuds de surface assurent que le volume des objets soit constant. Contrairement aux approches que nous avons présentées précédemment, les forces d'interaction ne sont pas appliquées comme des conditions limites pour les éléments formant le tissu. Un des avantages de la méthode est donc qu'elle ne nécessite pas d'étape de remaillage et permet que l'aiguille soit inserée à n'importe quelle position. L'aiguille est modélisée par une série de nœuds, et une zone d'influence est définie autour de chacun (voir figure 2.11). Les nœuds du tissu se trouvant dans ces zones sont directement sujets aux forces provenant de l'insertion de l'aiguille. Les paramètres du modèle sont récupérés des mesures de DiMaio *et al.* [DS03a] et à chaque particule du tissu sont attribués deux poids correspondant à la résistance du tissu aux forces de frottement et de découpe. Il est possible de donner des paramètres différents à chaque particule, pour pouvoir représenter des changements de matériau et ainsi simuler par exemple une tumeur à l'intérieur d'un organe. La valeur des forces appliquées par l'aiguille est une moyenne pondérée des poids des nœuds du tissu présents dans la zone d'influence. La force de pénétration est gérée de la même façon que la force de découpe, en attribuant alors aux nœuds de la surface un paramètre de découpe plus élevé. Le modèle de frottement gère les frottements statique et dynamique et est basé sur celui de Baraff *et al.* [BW98]. Les particules du tissu n'étant pas nécessairement le long de l'aiguille, les auteurs créent des particules de tissu virtuelles qui pouvent être contraintes en déplacement lorsque le frottement est statique. Une simulation de curiethérapie est proposée, avec plusieurs organes modélisés et en interaction, et la possibilité d'insérer plusieurs aiguilles simultanément (représentée sur la figure 2.11).

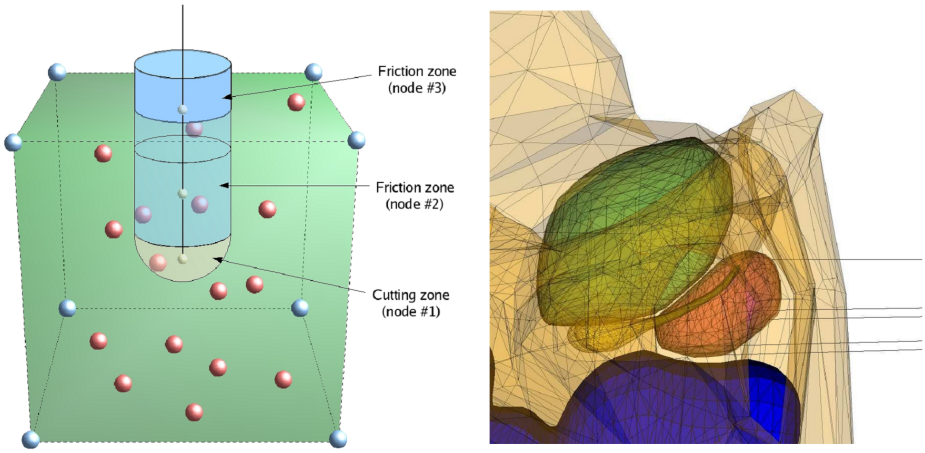


FIGURE 2.11 – (à gauche) Zones d'intérêt créées autour des nœuds de l'aiguille. (à droite) Insertion de plusieurs aiguilles dans la prostate pour une simulation de curiethérapie. [MPT06]

Récemment, Chentanez *et al.* [CAR+09] ont proposé une simulation d'insertion d'aiguille en 3D où l'interaction avec le tissu est géré par des contraintes. Les organes sont représentés par un modèle éléments finis sur des maillages tétrahédriques, et l'aiguille dont le diamètre est considéré négligeable est modélisée comme une tige élastique 1D à la manière de DiMaio et Salcudean [DS05b]. Les auteurs choisissent une intégration du système implicite en utilisant

le schéma de Newmark. Le maillage du tissu est remaillé et mis à jour à chaque pas de temps pour conformer à l'aiguille, de telle sorte que les nœuds de l'aiguille coincident exactement avec des nœuds du tissu. L'aiguille pouvant s'étirer légèrement durant la simulation, les auteurs la reparamétrisent et interpolent à nouveau ses paramètres physiques. Chaque nœud de l'aiguille a un état de frottement qui lui est propre, parmi les valeurs « libre » si ce nœud est en dehors du tissu, « statique » et deux états « dynamique » selon la direction du mouvement de l'aiguille à cet endroit. Les nœuds en commun avec le tissu permettent le couplage, dont les équations sont resolues par multiplicateur de Lagrange. Le système couplé obtenu est symétrique mais indéfini car on ne connait pas avant la résolution l'état de frottement de chaque nœud. Ce type de système ne peut normalement pas être résolu par la méthode du gradient conjugué, et demande un algorithme du type MINRES ou SYMMLQ [PS75], mais qui sont deux fois plus lourds en calculs par itération. Les auteurs choisissent d'utiliser tout de même un gradient conjugué parce qu'en pratique il résout le système avec la tolérance requise. Marcia [Mar08] a fait la même observation pour des systèmes linéaires et en donne une explication. Pour la résolution, ils doivent donc deviner tous les états de frottement, et les deviner à nouveau si le premier essai se rélève être faux. En parallélisant leur méthode, ils atteignent des temps de calcul interactifs sur une simulation de curiethérapie de la prostate mettant en jeu plusieurs organes et un très grand nombre de tétraèdres. Les auteurs proposent également de gérer l'insertion d'aiguilles biseautées, sur lesquelles le tissu exerce une force non symétrique entraînant une flexion de l'aiguille. Cet effet est simulé par l'ajout d'un déplacement au niveau de la pointe, perpendiculaire et proportionnel à la vitesse d'insertion. Nous allons voir par la suite d'autres études qui se sont intéressés tout particulièrement à la simulation d'aiguilles déformables.

2.4.2 Déflexion de l'aiguille durant son insertion

Récemment des travaux [APA07, TcH08, OA08] ont étudié l'utilisation d'aiguilles très flexibles munies de pointes biseautées pour la navigation à l'intérieur des tissus. Les praticiens qui se servent du biseau de l'aiguille accomplissent celà par expérience ce qui rend cette pratique difficile à enseigner. Les simulations s'intéressent à la déformation de l'aiguille en relation avec les tissus, un des buts recherchés étant de pouvoir prévoir le chemin emprunté par l'aiguille et de contrôler entièrement l'insertion de celle-ci par des robots.

Okamura *et al.* [OSO04] se sont également intéressé durant leurs mesures aux effets de la géométrie de l'aiguille sur sa déflexion. Ils montrent que les aiguilles fines se déforment plus, et que les pointes biseautées entrainent une plus grande flexion que les pointes coniques ou triangulaires. Les aiguilles biseautées fléchissent parce qu'elles sont asymétriques et que les forces de réaction du tissu sont plus importantes d'un côté que de l'autre. Cependant, les aiguilles symétriques peuvent également se déformer suite aux variations mécaniques à l'intérieur des matériaux dans lesquels est insérée l'aiguille.

Kataoka *et al.* [KWAM01] proposent un modèle pour représenter la déflexion d'une aiguille biseautée durant son insertion, et introduisent en particulier le concept de force infinitésimale de déflexion par unité de longueur. Ils posent une formule permettant de calculer le déplacement latéral de la pointe de l'aiguille en fonction de cette force, de la distance d'insertion et des paramètres physiques de l'aiguille. En comparant leur modèle à des mesures expérimentales, ils observent que la forme des courbes de déplacement sont similaires, mais que la déflexion mesurée est plus importante que celle prévue. Ils reconnaissent que leur modèle est

incomplet, car ils n'ont pas considéré le biseau de l'aiguille ni les propriétés mécaniques du tissu.

Webster *et al.* [WMO05] ont étudié l'impact du biseau de la pointe de l'aiguille sur sa déflexion lors d'une manipulation d'insertion dans un tissu. Leurs mesures ont porté sur l'insertion d'aiguilles dans un fantôme peu déformable, à l'aide d'un robot pouvant contrôler l'orientation et la translation du corps de l'aiguille, avec des capteurs de forces et de moments ainsi qu'une aquisition d'images stéréo. Les auteurs ont testé 5 angles de biseaux différents, à savoir 5°, 25°, 40°, 60°et 80°. Ils observent que la déflexion de l'aiguille lors de l'insertion augmente de façon monotone avec l'inclinaison du biseau. Dans leur position finale, on peut mesurer une distance de 2,5 cm entre la pointe des différentes aiguilles, soit 10% de la distance d'insertion (voir figure 2.12). La vitesse d'insertion n'a pas d'influence dans la déflexion de l'aiguille lors de l'insertion dans le fantôme relativement dur qui est utilisé, mais les auteurs émettent l'hypothèse que cela est différent pour les tissus déformables.

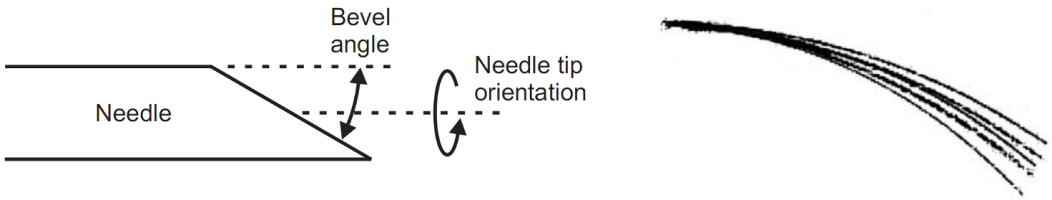


FIGURE 2.12 – (à gauche) Définition du biseau de l'aiguille. (à droite) Configuration finale des aiguilles après le test d'insertion. [WMO05]

Webster *et al.* [WKC+06] ont développé un modèle non holonomique pour conduire une aiguille flexible dont la pointe est en biseau. Ils rapprochent l'insertion d'une aiguille biseautée au déplacement d'un vélo dont la roue avant est fixée avec un certain angle. La résolution de ce modèle est inspiré des travaux sur les systèmes non-holonomiques de Murray *et al.* [MSZ94] dans la communauté robotique. Les auteurs ont choisi d'utiliser un tissu très rigide leur permettant de simplifier les interactions en ignorant les déformations du tissu. Cette hypothèse n'est pas réaliste pour les simulateurs médicaux, et de plus les déformations du tissu peuvent jouer un grand rôle sur la déflexion de l'aiguille car elles modifient les directions et les valeurs des forces qui s'appliquent sur elle.

Les modèles d'interaction entre l'aiguille et le tissu déformable ont été en partie utilisés pour réaliser des simulateurs d'insertion d'aiguille temps réel [DS03a, APT+03b, GSD+05]. Dans une première approche, ces simulateurs ne prenaient pas en compte la complexité des organes et les différents tissus que l'ont peut rencontrer lors d'une opération chirurgicale. Ces modèles sont également de plus en plus employés pour réaliser le planning préopératoire de l'insertion d'aiguilles déformables dans des environnements complexes. Les recherches portent ainsi sur des algorithmes de recherche de chemin qui doivent en général connecter le point d'insertion avec la cible tout en évitant les obstacles et en réduisant la courbure de l'aiguille afin de minimiser la pression sur les tissus. Le problème posé pour les modèles d'interaction est de pouvoir déterminer la translation et l'orientation à imposer à la base de l'aiguille en fonction d'une position et d'une orientation voulues de la pointe. L'application de ces travaux est la manipulation des aiguilles par des robots, qui peuvent reproduire exactement les gestes

calculés, tout en mesurant les forces que les tissus appliquent à l'aiguille afin de mettre à jour les modèles.

DiMaio et Salcudean [DS03b, DS05b] ont réalisé le planning de l'insertion d'aiguilles flexibles symétriques. Ils ont proposé le calcul d'une jacobienne de l'aiguille qui met en relation le déplacement et l'orientation de la pointe de l'aiguille en fonction des déplacements imposés à la base (voir figure 2.13). Glozman et Shoham [GS04] se sont également intéressés

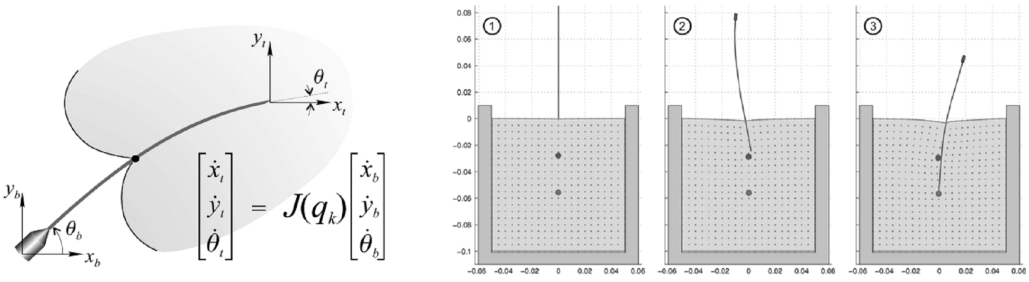


FIGURE 2.13 – (à gauche) Déplacement de la pointe en fonction du déplacement imposé à la base de l'aiguille. (à droite) Trajectoire simulée. [DS03b, DS05b]

aux aiguilles dont la pointe est triangulaire ou conique, et ont utilisé des algorithmes de cinématique inverse pour établir la trajectoire de l'aiguille permettant de minimiser la pression sur les tissus. Les auteurs ont validé leur méthode en faisant reproduire ce chemin à un robot qui a inséré une aiguille flexible dans un fantôme, en évitant un obstacle (voir figure 2.14). Park *et*

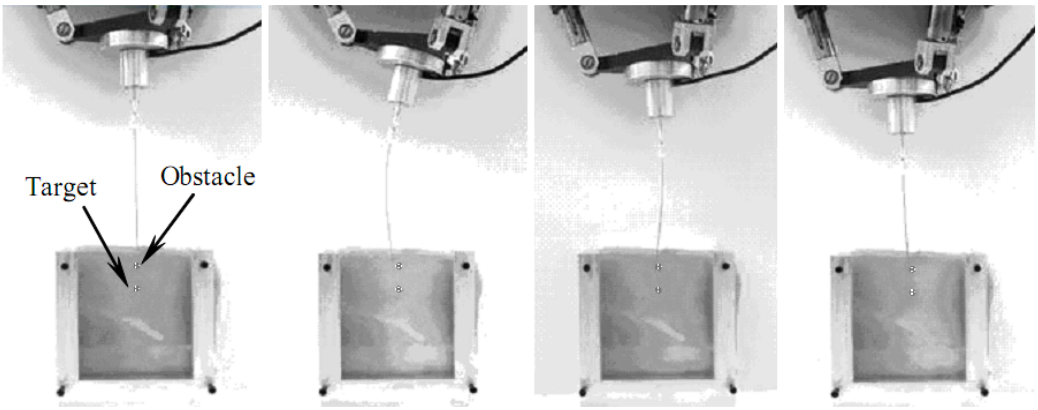


FIGURE 2.14 – Insertion robotisée suivant une trajectoire optimale pour éviter un obstacle en minimisant la pression sur le tissu. [GS04]

al. [PKZ+05] ont modélisé le biseau de l'aiguille comme une contrainte non-holonomique. Ils ont ensuite utilisé un modèle déterministe et des méthodes cinématiques inverses utilisant les probabilités de densité d'information pour proposer un chemin pour l'aiguille. Alterovitz *et al.* [AGO05, ABG08] posent le problème de l'insertion d'aiguilles biseautées comme un problème d'optimisation non-linéaire contraint. Ils prennent en compte la déformation des tissus, et

calculent les chemins d'insertion pour des cibles inatteignables par des aiguilles rigides (voir figure 2.15). Duidam *et al.* [DXA+08] utilisent une approche purement géométrique du pro-

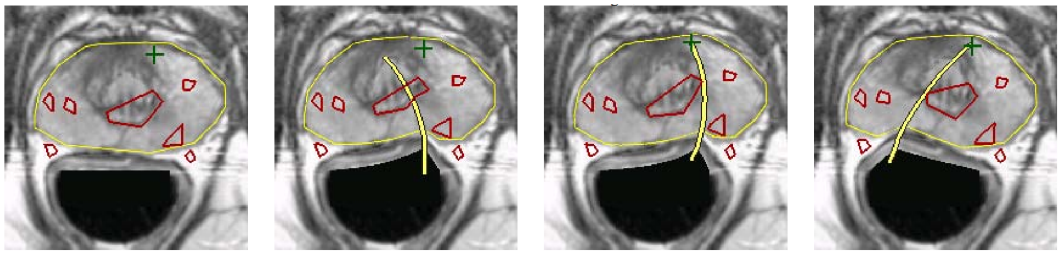


FIGURE 2.15 – Prostate humaine, tumeur cible et multiples obstacles. Trajectoire calculée pour un biseau à gauche. Planning pour un biseau à gauche. Planning pour un biseau à droite. [AGO05, ABG08]

blème, et interprêtent l'insertion d'aiguille biseautée comme la trajectoire d'un avion dont la vitesse est constante et l'utilisateur contrôle le roulis. Chentanez *et al.* [CAR+09] montrent dans leurs travaux récents un algorithme de planning se basant sur leur modèle d'insertion d'aiguilles symétriques ou biseautées, sans donner de précisions.

2.4.3 Conclusion sur les approches présentées

Avant de présenter notre modèle de contraintes utilisé pour modéliser les interactions entre aiguille et objet déformable, nous voulons récapituler les principales caractéristiques des travaux que nous avons présenté dans cette section. Dans la majorité des travaux l'aiguille agit sur le tissu en posant des conditions limites sur les éléments qui le constituent. Dans le cas d'aiguille rigide, certains ont décidé d'imposer le chemin qu'empruntera l'aiguille. On voit le plus souvent des techniques de remaillage ou de repositionnement afin que des nœuds du tissu soient placés le long de l'aiguille. Enfin, nous avons vu un exemple d'utilisation d'un modèle de tissu sans maillage où les forces d'interaction sont réparties sur les nœuds les plus proches de l'aiguille. Lorsque l'aiguille est insérée dans un tissu, elle exerce une pression sur lui et le déforme. Dans le même temps, l'utilisateur peut ressentir la résistance du tissu. Les travaux pionniers ont directement appliqué aux nœuds du tissu proches de l'aiguile des forces dont l'intensité est lue à partir d'une densité de force obtenue expérimentalement. D'autres calculent ces forces à partir de modèles mathématiques comportant un nombre réduit de paramètres (par exemple : coefficient de frottement, vitesse d'insertion, diamètre de l'aiguille). Lorsque la vitesse d'insertion n'est pas constante durant la simulation, par exemple si l'aiguille est contrôlée par l'utilisateur, il faut utiliser un modèle de frottement gérant frottements statique et dynamique. Pour modéliser la déflexion de l'aiguille dont la pointe est biseautée, une contrainte non-holonomique peut être utilisée. Une autre méthode est d'appliquer une force ou un déplacement normal afin de faire dévier l'aiguille, dont l'amplitude dépend du déplacement axial. Certains travaux récents gèrent le fait que l'utilisateur peut retirer l'aiguille du tissu et éventuellement la réinsérer. Pour la simulation de la curiethérapie telle qu'elle est pratiquée par une partie des hôpitaux français, certains simulateurs permettent l'insertion de plusieurs aiguilles dans le même tissu. Enfin, alors que les premiers travaux ont porté sur la simulation temps réel de l'insertion d'aiguille, potentiellement avec retour haptique pour

l'utilisateur, des travaux récents s'intéressent au planning et calcul de trajectoire optimale, avec manipulation de l'aiguille flexible par un robot.

Chapitre 3

Une approche générique de l'interaction entre modèles hétérogènes

L'utilisation de lois physiques dans des simulations informatiques est de plus en plus fréquente ; elle permet l'inclusion dans les scènes d'objets qui seraient autrement difficiles à animer (par exemple, des liquides qui se déplacent de façon réaliste, des objets déformables ou encore des objets de types variés interagissant les uns avec les autres), et offre également plus de liberté que les techniques d'animations classiques, où le comportement des objets est scripté par un spécialiste en animation. Dans le cas spécifique des applications interactives, la simulation basée sur les lois physiques doit prendre en compte l'interaction de l'utilisateur sans se servir de lois d'animation scriptées. Cette interaction est généralement intégrée au sein des scènes virtuelles à l'aide de certains objets spécialisés, dont le mouvement est contraint par l'utilisateur, via l'utilisation de périphériques adaptés. Cet objet, aussi bien que les autres, peut interagir avec le reste de la scène au travers de la simulation numérique des lois physiques. Lorsqu'un périphérique haptique est utilisé, les forces calculées et appliquées à l'objet virtuel de l'utilisateur sont utilisées comme une base pour le retour d'effort. Chaque objet peut potentiellement entrer en collision avec d'autres objets, auquel cas un traitement précis des contacts est nécessaire. Un objet peut également être inclus dans une structure plus complexe, l'animation de l'ensemble devant alors combiner le comportement de chaque objet en prenant en compte les contraintes additionnelles. Dans la majorité des cas pratiques, les scènes virtuelles demandent la définition d'un grand nombre de contraintes et leur résolution numérique.

Dans les applications de Réalité Virtuelle, nous pouvons être amenés à rencontrer des cas pratiques de simulations où des contraintes changent dynamiquement, et peuvent être de types variés. La majorité des contraintes sont *non holonomiques*, c'est à dire qu'elles font intervenir des dérivés des degrés de liberté. Une contrainte est dite *holonomique* lorsque l'équation qui lui correspond ne fait apparaître que des dérivés à l'ordre 0 des degrés de liberté : elle doit être exprimable sous la forme d'une fonction $f(x_1, x_2, ..., x_N, t) = 0$, qui ne dépend que des coordonnées x_j et du temps t, pas des vitesses. La combinaison des deux types de contraintes dans la même simulation implique habituellement l'utilisation d'une intégration temporelle

explicite [WGW90], qui est connue pour offrir une mauvaise stabilité. Ce manque de flexibilité rend la combinaison de contraintes hétérogènes difficile à atteindre dans le contexte de la simulation interactive et de la Réalité Virtuelle. De plus, toujours dans ce contexte, on peut relever le fait que peu de travaux se sont penchés sur le problème des objets déformables contraints, là où plusieurs approches sont disponibles pour les contraintes appliquées aux objets rigides.

Dans ce chapitre nous proposons une méthode pour combiner et traiter à la fois des contraintes holonomiques (telles que point fixe, point glissant) et non-holonomiques (contact, frottement), en restant compatible avec les schémas d'intégration implicite. Nous montrons que la méthode proposée gère aussi bien les objets déformables que les objets rigides, dans un contexte de simulation interactive. La méthode présentée utilise et sera intégrée dans le framework *SOFA* (Simulation Open Framework Architecture). Le travail sur les contraintes généralisées a fait l'objet d'une publication à VRIPHYS'08 : *Unified Processing of Constraints for Interactive Simulation* [GDG08]

Nous organisons le chapitre comme suit : la section 3.1 présente les travaux antérieurs sur lesquels nous nous basons. Nous proposons à la section 3.2 une définition unifiée des contraintes classiques, en particulier les contraintes bilatérales, unilatérales et de frottement. La section 3.3 définit l'espace dans lequel nous définissons les contraintes, et nous introduisons la méthode de résolution à la section 3.4. Enfin, des exemples et des mesures de performance sont donnés à la section 3.5, montrant que la méthode proposée est utilisable pour réaliser des simulations interactives.

3.1 Quelques travaux antérieurs

Le définition de contraintes et leur résolution est un processus bien connu dans le domaine des simulations interactives et de l'informatique graphique. Plusieurs solutions ont été proposées, que nous décrivons dans la suite de cette section.

Premièrement, un grand nombre de travaux ont réutilisé et optimisé la méthode des multiplicateurs de Lagrange. C'est un outil pour trouver les extrèmes d'une fonction de mouvement dans un système mécanique soumis à une ou plusieurs contraintes. La méthode que nous utilisons consiste à ajouter une nouvelle équation au système de matrices formé par la mécanique. Lorsque ce système de matrices est résolu, la contrainte est automatiquement respectée et le multiplicateur de Lagrange fournit la force de réaction associée à la contrainte. Cependant, la méthode requiert l'utilisation de solveurs spécifiques à cause de la présence de termes nuls dans la diagonale de la matrice obtenue. De plus, le choix de la place de l'équation de la contrainte dans le système de matrices peut parfois être délicate pour garder de bonnes propriétés (matrice bande, matrice creuse) pour un processus de résolution efficace [LF04]. Les multiplicateurs de Lagrange sont très efficaces pour exprimer des contraintes dans un ensemble d'objets articulés [Bar96]. En les triant de telle façon à avoir n contraintes primaires et k contraintes auxiliaires, des matrices creuses sont obtenues, qui peuvent alors être résolues en $O(n)+O(nk)$. Cette méthode est rapide et exacte, mais les systèmes simulés doivent nécéssairement suivre une structure linéaire, potentiellement avec quelques embranchements. Cette méthode est inefficace pour les contraintes sur des objets déformables, puisqu'alors $k > n$.

Lenoir *et al.* [LF04] ont présenté un système de contraintes basé sur des multiplicateurs de Lagrange pour la simulation hétérogène d'objets rigides et déformables. Ils décomposent le système en trois équations : mouvement libre, résolution des contraintes dans l'espace des contraintes et un mouvement de correction. Les matrices sont mises à jour de manière efficace entre chaque pas de temps. Leur technique peut être étendue aux contraintes unilatérales en utilisant l'algorithme des status [DLC07] qui ajoute ou supprime une équation de contrainte durant un processus itératif de telle façon à obtenir uniquement une réaction positive sur le multiplicateur de Lagrange. Cependant, cela augmente le nombre de calculs et rend cette méthode peut adaptée aux contraintes non-holonomiques telles que le frottement.

Les contraintes holonomiques réduisent le nombre de degrés de liberté d'un système ; une seconde méthode de gestion des contraintes consiste alors à transformer le système de coordonnées de telle façon à travailler avec un nombre réduit de variables. En mécanique lagrangienne, l'utilisation de coordonnées généralisées permet l'élimination des coordonnées qui ne sont pas indépendantes parce que liées par une expression de contrainte. Cette méthode est souvent mise en œuvre pour la simulation efficace d'objets articulés [Fea99].

Une troisième méthode consiste à projeter les contraintes durant un processus itératif, tel qu'une méthode de gradient conjugué. Une façon très simple de procéder, en présence de contraintes bilatérales, est de forcer la vitesse des points contraints à chaque itération [WGW90]. La méthode de la projection a été étendue aux contraintes unilatérales avec une méthode de descente du gradient en utilisant une stratégie d'ensemble actif (méthode de Rosen) [RA04].

Une autre solution basée sur la programmation quadratique [6] (QP) permet de mixer les solveurs bilatéraux et unilatéraux. Redon [RKC02] utilise le principe des moindres contraintes de Gauss pour la simulation d'objets rigides. Les contraintes sont résolues très efficacement avec un solveur QP. Cependant, l'extension d'une telle méthode pour son application aux objets déformables posera des problèmes avec la méthode QP, et ralentira la résolution. De plus, le principe de Gauss ne s'applique pas aux contraintes de frottement.

Murty *et al.* [Mur97] ont montré que la formulation QP est équivalente à une formulation d'un problème de complémentarité linéaire (LCP). Les contraintes de contact peuvent aisément être décrites et résolues comme un LCP. Cette formulation a été introduite dans la communauté graphique par Baraff [Bar89] qui a proposé un algorithme pour le calcul de contacts sans frottement entre des objets solides. Il est basé sur l'algorithme de Dantzig, en relation avec les méthodes de résolution des LCP par pivot. Ruspini *et al.* [RK00] décrivent les collisions dans l'espace des contacts, et expriment les équations de contact et les forces de réponse dans cet espace. Ils utilisent une matrice inversée de l'inertie dans l'espace des contacts pour décrire la relation dynamique entre les points de contact. Ils identifient dans leur équation un terme de mouvement libre, représentant l'accélération dans l'espace des contacts qui aurait lieu si aucun contact n'existait. Les forces de contact sont obtenues à travers la résolution d'un problème LCP.

La formulation LCP, basée sur les contraintes unilatérales a été étendue aux contacts avec frottement entre des objets rigides [APS98]. Le cône de frottement de Coulomb est approximé par un cône pyramidal afin de maintenir une formulation LCP. Cependant, cela augmente

6. La programmation quadratique cherche à résoudre des problèmes d'optimisation où la fonction objectif est quadratique, les contraintes étant linéaires ou plus rarement quadratiques.

grandement la taille du LCP et empêche cette méthode d'être utilisée dans les applications temps réel.

Moreau et Jean [MJ96] introduisent un solveur spécifique pour gérer la non-linéarité du problème de complémentarité du frottement entre des objets rigides. Leur approche est basée sur le cône de frottement exact et un solveur itératif de type Gauss-Seidel. L'utilisation de cette approche dans le contexte de la simulation interactive de modèle déformable a été proposée dans [DDKA06]. Ces dernières approches sont efficaces mais sont limitées aux contraintes de contacts avec frottement. Ce travail cherche à présenter une méthode plus flexible permettant d'utiliser et de combiner différents types de contraintes à la fois sur des objets rigides et déformables.

3.2 Contraintes généralisées

Dans cette section nous décrivons le formalisme unifié que nous utilisons pour gérer, au sein de la même simulation, les différentes contraintes qui nous intéressent. Nous présentons ici les familles de contraintes à partir desquelles nous déclinerons toutes celles que nous utiliserons dans nos simulations. Nous reviendrons plus en détail sur chaque type de contrainte dans la suite pour présenter leur résolution ainsi que leurs applications, mais nous voulons montrer ici que toutes ces contraintes partagent un ensemble commun de paramètres.

Ces paramètres sont les suivants :
- la **dimension** de la contrainte : nous introduisons un peu abusivement ce terme dans ce travail pour décrire le nombre de degrés de liberté concernés par la contrainte.
- la ou les **directions** de la contrainte.
- les **violations** de la contrainte suivant chacune des directions.
- la **loi** de la contrainte, qui donne le comportement, issu des lois physiques, que l'on cherche à faire appliquer par la contrainte.

Nous détaillons rapidement dans cette section chaque type de contrainte et montrons qu'elles peuvent toutes être entièrement définies en spécifiant leur loi, dimension, direction(s) et violation(s).

3.2.1 Contraintes bilatérales

Les contraintes bilatérales sont souvent utilisées pour simuler les points fixes ou les articulations. Elles peuvent être exprimées en projetant la contrainte suivant des direction prédéfinies (en utilisant la technique de la réduction de coordonnées) : pour chaque dimension, le mouvement est contraint selon une direction spécifique. En fonction du nombre de dimensions de la contrainte et du nombre de degrés de liberté des nœuds, des applications variées peuvent être créées.

Ainsi, une contrainte bilatérale de dimension 1 pourra représenter un point contraint à se déplacer sur un plan. En effet, on empêche tout mouvement de l'objet suivant la normale à ce plan, tout en laissant libre les déplacements tangents au plan. Une contrainte de dimension 2 correspond à l'assemblage de deux contraintes de dimension 1, chacune limitant le déplacement à un plan. Les deux contraintes étant tout le temps vérifiées, l'objet contraint est restraint

à l'intersection des deux plans, c'est à dire une ligne. Enfin, un objet peut être contraint à un point fixe dans l'espace ou sur un autre objet grâce à l'utilisation d'une contrainte de dimension 3. On peut envisager de créer des contraintes de dimensions supérieures, mais dont l'usage est moins courant. On peut néanmoins citer la contrainte de dimension 6 qui consiste à contraindre un objet rigide à la fois en position et en rotation. Des exemples de ces différentes contraintes sont présentés dans la section 3.5.

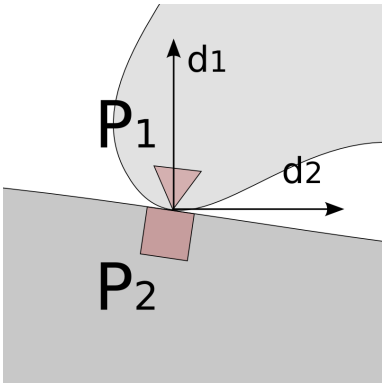


FIGURE 3.1 – Les contraintes bilatérales peuvent utiliser plusieurs directions selon l'application désirée. Pour un point fixe en 2D, n'importe quelle combinaison de deux vecteurs orthogonaux est valide.

Pour définir une contrainte bilatérale, nous devons en premier lieu choisir sa dimension. Pour chaque degré de liberté d'un nœud que nous voulons contraindre, il faudra préciser une direction (voir figure 3.1). Dans le cas particulier où l'ensemble des degrés de liberté d'un nœud sont contraints, nous pouvons utiliser le repère orthonormé de l'espace pour définir la contrainte. En plus de cela, pour chaque dimension la violation δ doit être calculée. Elle est associée à un vecteur dans la direction de la contrainte, orienté de tel façon que, si on la considère comme une force, elle tend à vérifier la contrainte. De façon générale, δ est la distance entre les deux objets avant la résolution du système, et qui doit idéalement être nulle après la résolution de ce pas de temps.

3.2.2 Contraintes unilatérales

Les contraintes unilatérales peuvent sembler au premier abord plus simple puiqu'elles ne font intervenir qu'une seule dimension (voir figure 3.2), mais la loi qui régit leur comportement est non lisse. Elles sont utilisées le plus souvent pour la réponse à la collision, car leur comportement signifie que l'interpénétration de deux objets sera corrigée lors de la résolution de la contrainte, mais la force entre les deux objets ne peut pas être attractive. On peut ainsi parfaitement modéliser des contacts sans frottement.

Une différence marquante par rapport aux contraintes bilatérales est le fait que pour ces contraintes unilatérales le signe de la violation est d'une importance majeure pour la résolution. En effet, il dicte si la contrainte est effectivement active ou non. Ainsi, résoudre une contrainte simple de contact lorsque la violation est négative est équivalent à la résolution

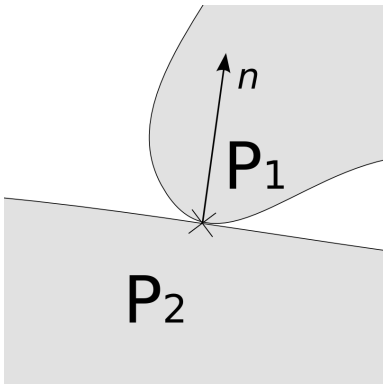


FIGURE 3.2 – Les contraintes unilatérales définissent une direction **n** perpendiculaire à la surface de contact.

d'une contrainte bilatérale. Une violation positive signifie que les deux objets ne sont plus en contact, et la force calculée durant la résolution doit alors être nulle. Lors de la résolution des contraintes, le couplage mécanique existant entre celles-ci compliquera le problème, puisque la résolution d'une contrainte peut modifier le signe de la violation d'une seconde.

3.2.3 Contraintes de frottement

Les contraintes de frottement sont rarement utilisées seules, car elles apportent le comportement dissipatif suivant les directions tangentielles aux contraintes bilatérales ou unilatérales auxquelles elles sont combinées. On peut vouloir introduire du frottement dès lors que l'on modélise le déplacement d'un objet en contact avec un autre, que ce soit un point glissant le long d'une courbe ou le contact frottant entre les objets. Ajouter du frottement nécessite de connaître la valeur de la force suivant la direction normale, le plus souvant la ou les directions de la contrainte bilatérale ou unilatérale qui est associée avec cette contrainte de frottement (voir figure 3.3).

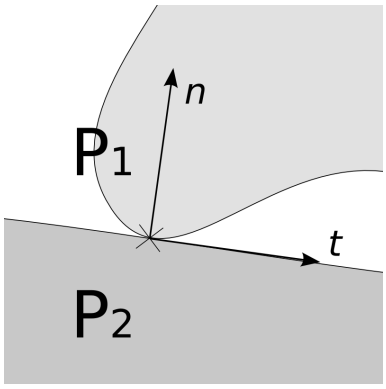


FIGURE 3.3 – Les contraintes de frottement utilisent, dans le cas des contraintes de contact, la normale **n** et les directions **t** tangentielles à la surface de contact.

Pour modéliser le frottement, nous proposons d'utiliser la loi de frottement de Coulomb (voir figure 3.4). Celle-ci dicte que lorsque les forces de réponse dans les directions tangetielles $f_{\vec{T}}$ sont plus basses que la force normale $f_{\vec{n}}$ multipliée par le coefficient de frottement μ, les deux objets sont en adhérence et que le déplacement tangentiel $\delta_{\vec{T}}$ est ainsi nul (on l'appelle le frottement statique). Dans le cas contraint, les objets glissent l'un contre l'autre ($\delta_{\vec{T}} \neq \vec{0}$), et les forces de réponse sont réduites à un comportement dissipatif $f_{\vec{T}} = -\mu \, ||f_{\vec{n}}|| \, \vec{T}$ (c'est le frottement dynamique). Puisque nous utilisons un modèle de Coulomb intégré, nous travaillons en déplacement et non en vitesse.

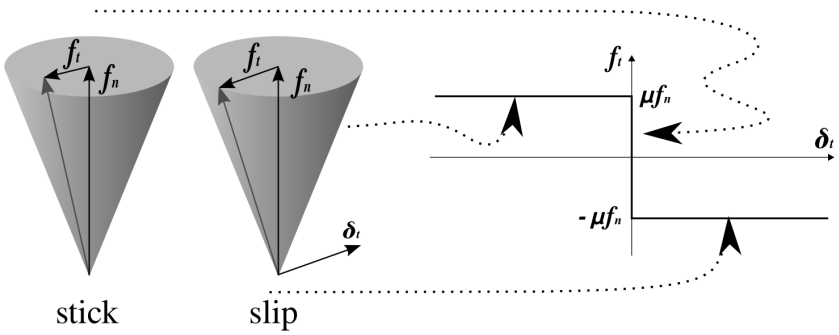


FIGURE 3.4 – Représentation de la loi de frottement de Coulomb. La force de réaction est strictement à l'intérieur du cône de frottement lorsque les objets sont en adhérence, et est sur la bordure du cône lorsqu'ils sont en glissement.

3.3 Espace des contraintes

Nous avons vu que toutes les contraintes qui nous intéressent peuvent être définies par une direction. Une force sera appliquée suivant cette direction afin de faire respecter la contrainte, son amplitude étant le résultat de la résolution des contraintes.

Des paramètres sont définis selon ces directions suivant le type des contraintes :
- la mesure de la violation pour les contraintes bilatérales
- la mesure de l'interpénétration pour les contraintes unilatérales de contact
- la force normale et le déplacement tangentiel pour les contraintes de frottement.

Les lois de contraintes utilisent ces paramètres, nous alors donc faire en sorte d'avoir un accès direct à ceux-ci lors de la phase de résolution.

De plus, exprimer le comportement des objets contraints suivant les directions des contraintes nous permettra de prendre en compte le couplage entre ces contraintes.

3.3.1 Contraintes interpolées

Dans [RK00], Ruspini *et al.* ont introduit la notion d'*espace des contacts* comme l'espace local de solution dans lequel les contraintes unilatérales sont résolues, à travers un exemble de paramètres qui décrivent localement le déplacement relatif des objets durant le contact et la collision. Pour une paire d'objets en interpénétration, un algorithme de détection de collision identifie un ensemble de points potentiellement en contact, et fournit pour chacun d'eux la normale $\mathbf{n}$ du contact, perpendiculaire à la géométrie au point de contact. En utilisant la direction $\mathbf{n}$, nous pouvons exprimer la distance relative δ^{free} entre les deux points de contact.

En généralisant cette idée aux autres contraintes et à leurs paramètres, nous introduisons la notion d'*espace des contraintes*. Dans cet espace, il est possible d'exprimer des contraintes avec une durée de vie finie (tel que les contraintes de contact, voir section 3.2.2). Ecrire les lois de contraintes dans cet espace est souvent plus aisé que dans l'espace des mouvements, et ouvre la porte à des contraintes plus complexes.

Nous voulons que nos contraintes puissent être créées n'importe où sur la surface ou dans le volume des objets que nous manipulons. Nous associons à chaque point de contact un repère, et c'est dans cet espace que nous mesurerons les déplacements relatifs entre les objets. Nous lions les positions de chaque contrainte P à ceux des degrés de liberté $\mathbf{x}$ de l'objet à l'aide de la fonction de "mapping" : $P = \Phi(\mathbf{x})$. La relation cinématique entre l'espace des contraintes et l'espace des degrés de liberté nécessite une linéarisation de l'équation précédente. On pose ainsi $H = \frac{\partial \Phi}{\partial \mathbf{x}}$ afin d'exprimer la relation entre les déplacements : $dP = H\mathbf{dx}$. Pour les objets déformables, nous linéarisons les fonctions de "mapping" en utilisant simplement les coordonnées barycentriques des contraintes dans les éléments. Pour les objets rigides et quelques objets très déformables, la fonction $\Phi(\mathbf{x})$ n'est pas linéaire et nous construisons une matrice jacobienne à l'emplacement de la contrainte.

Une fois que nous aurons résolu les contraintes dans l'espace des contraintes, il faudra transposer les forces pour les appliquer aux degrés de liberté réels de l'objet. Nous nous basons sur le théorème des travaux virtuels, qui stipule que le travail de la force f_c appliquée au point de la contrainte correspond à la somme des forces $\mathbf{f}_{ddl}$ appliquées aux degrés de liberté : $dP^T f_c = \mathbf{dx}^T \mathbf{f}_{ddl}$. Nous obtenons donc le vecteur de forces sur les degrés de liberté : $\mathbf{f}_{ddl} = H^T f_c$

Le positionnement des contraintes est défini par deux points P et Q, un sur chaque objet devant être contraint. Pour les simulation médicales telles que l'insertion d'aiguille, nous aurons souvent à placer des contraintes entre un point compris dans le volume des tissus et un autre sur la courbe de l'aiguille.

Pour chaque contrainte, δ mesure la distance entre ces deux points suivant la direction $\mathbf{n}$ de la contrainte. Les points peuvent être placés n'importe où dans le volume des objets, leur déplacement sera mis à jour avec la même interpolation que les modèles déformables (voir figure 3.5).

La variation de δ peut être mappée sur le déplacement des nœuds du modèle déformable, prenons l'exemple d'une contrainte se basant sur les déplacements d'une aiguille $\Delta\mathbf{q}_a$ et d'un organe déformable $\Delta\mathbf{q}_o$:

$$\Delta\delta = \mathbf{n}^T(\mathbf{u}_a - \mathbf{u}_o) = \mathbf{n}^T(\mathbf{H}_a\Delta\mathbf{q}_a - \mathbf{H}_o\Delta\mathbf{q}_o) = \mathbf{N}_a\Delta\mathbf{q}_a + \mathbf{N}_o\Delta\mathbf{q}_o \tag{3.1}$$

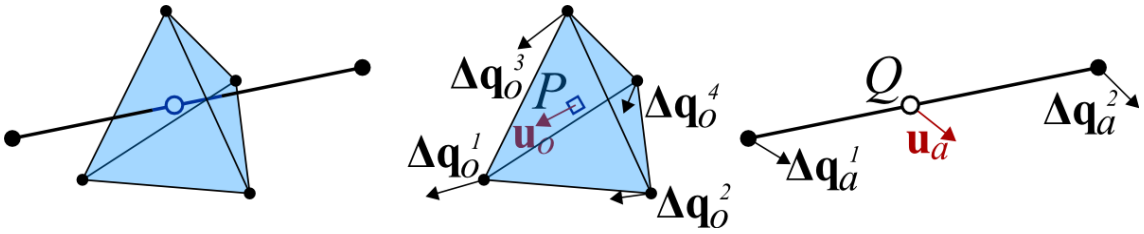


FIGURE 3.5 – **Exemple de "mapping" pour un point de contrainte**. *Une interpolation linéraire est utilisée sur les tétraèdres du modèle déformable, le déplacement* $\mathbf{u}_o$ *d'un point de contrainte placé dans un tétraèdre est obtenu par les coordonnées barycentriques de ce point et les déplacements* $\Delta\mathbf{x}_o$ *des quatres sommets* $\mathbf{u}_o = H_o\Delta\mathbf{x}_o$. *Le déplacement d'un point de contrainte sur un modèle 1d est obtenu de la même façon à partir de l'interpolation du modèle déformable* $\mathbf{u}_a = H_a\Delta\mathbf{q}_a$.

Dans la suite de ce travail, λ représente la force de réponse qui vérifie la contrainte.

Pour mapper cette force sur les nœuds du modèle déformable, nous utilisons le principe des travaux virtuels ($\lambda^T\Delta\delta = \mathbf{f}_o^T\Delta\mathbf{q}_o + \mathbf{f}_a^T\Delta\mathbf{q}_a$). Nous obtenons :

$$\mathbf{f}_o = \mathbf{N}_o^T\lambda \qquad et \qquad \mathbf{f}_a = \mathbf{N}_a^T\lambda \tag{3.2}$$

Les paramètres de l'espace des contraintes sont souvent interdépendants, et nous devons exprimer les relations dynamiques entre eux. La fonction de "mapping" $\Phi(X)$ reliant les positions dans l'espace des mouvements à celles dans l'espace des contraintes est le plus souvent non-linéaire. Dans le cas général, le problème peut être réduit à un problème linéaire en utilisant un Jacobien $H = \frac{\partial\Phi}{\partial x}$ évalué au temps t autour du point de contrainte. Cette matrice est utilisée pour construire l'opérateur de Delassus W, qui exprime le couplage mécanique dans l'espace des contraintes.

$$W = HCH^T \tag{3.3}$$

Dans cette expression, C est la matrice de compliance, calculée comme suit :

$$C = (\ddot{M} + \dot{B} + K)^{-1} \tag{3.4}$$

Ses propriétés sont présentées plus en détails dans [SDCG08]. M, B et K sont respectivement les matrices de masse, d'amortissement et de rigidité. Selon la nature de l'objet, cette matrice peut être obtenue facilement (pour les objets rigides), à l'aide d'un solveur spécifique de matrice bande (pour les objets déformables 1d) ou durant une étape de précalcul (objets déformables 3d linéaires ou corotationnels).

Pour les simulations temps réel d'objets déformables contraints, le nombre de degrés de liberté est généralement bien plus grand que le nombre de contraintes qui leur sont appliquées. Dans ce contexte, résoudre le problème dans l'espace des contraintes présente l'avantage de réduire significativement la taille du système à résoudre. Cependant, dans le cadre des simulations d'objets rigides purs, et si de nombreuses contraintes sont créées, une stratégie de résolution dans l'espace des mouvements sera plus appropriée, puisque chaque objet n'a pas plus de 6 degrés de liberté. Ce cas particulier ne sera pas étudié dans ce travail, plus axé vers la simulation simultanée d'objets rigides et déformables contraints.

Une stratégie locale consiste à résoudre en premier lieu le déplacement des objets sans leur appliquer de contraintes (étape de mouvement libre), puis à résoudre les contraintes dans l'espace des contraintes $\delta = W\mathbf{f} + \delta^{\mathbf{free}}$, et finalement calculer un mouvement de correction. Nous décrivons la résolution des contraintes à la section 3.4.

3.3.2 Contraintes et représentations multi-modèles des objets

La technique que nous venons d'aborder nous permet donc de placer des contraintes n'importe où dans le volume ou sur la surface d'un objet. Cependant, le framework *SOFA* dans lequel nous intégrons notre méthode de résolution des contraintes va plus loin dans l'utilisation des fonctions de "mapping". En effet, les auteurs ont intégré la possibilité de gérer simultanément plusieurs représentations différentes d'un même objet, qui sont ensuite mises à jour quand nécessaire. La figure 3.6 présente un exemple de ce concept. Le modèle mécanique intervenant sur les degrés de liberté de l'objet est indépendant du modèle de collision qui utilise ici une hiérarchie de sphère, mais également du modèle visuel qui peut alors être plus détaillé sans surcharger les calculs mécaniques. Plusieurs techniques de "mapping" sont donc employées pour mettre à jour à chaque pas de temps les différentes représentations de l'objet suivant les déplacements de ses degrés de liberté.

Dans ce travail, nous nous servons régulièrement de ce concept de multireprésentation. Ainsi, l'algorithme de détection de collision nous fournit des points de contacts sur la représentation surfacique triangulée des objets. Nous créons alors des contraintes de contact pour répondre aux collisions, en nous basant sur les triangles du modèle de collision. Le framework se charge alors de remonter les informations des contraintes à travers les différentes représentations, jusqu'aux degrés de liberté de l'objet, comme si elles avaient été directement créées sur eux. L'addition de la contribution de plusieurs contraintes est automatique, et très utile par exemple lorsque plusieurs points de contact sont situés sur un objet rigide ; ce dernier n'ayant que 6 degrés de liberté.

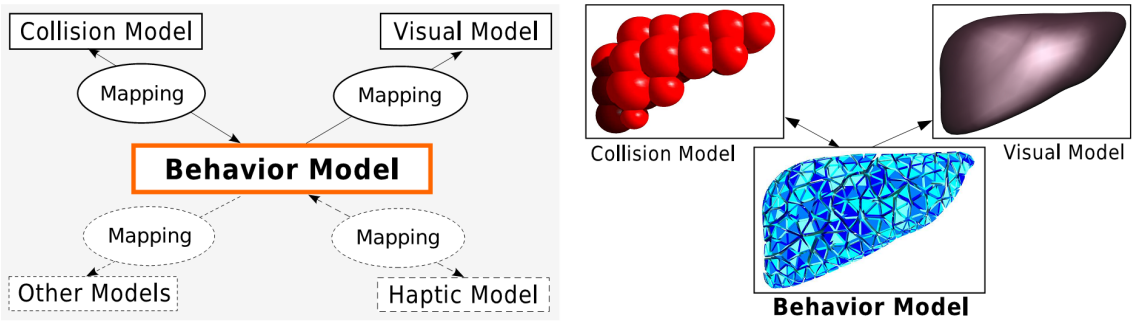


FIGURE 3.6 – Illustration de la représentation multi-modèles dans *SOFA*. (à gauche) Possibles représentations pour un objet simulé, avec le modèle comportemental (en général les degrés de liberté de l'objet) contrôlant les autres représentations à travers une série de "mappings". (à droite) Exemples de ces représentations sur un modèle de foie, où le modèle visuel est plus détaillé et que le modèle de collision se base sur une représentation totalement différente.

Un des inconvénients de la méthode de détection de collisions par proximité que nous utilisons (et que nous présentons succintement à la section 3.5.2), est que les objets en contact sont à tout moment séparés par un léger interstice dont la valeur est définie pour la scène. En séparant le modèle de collision de la représentation visuelle, il est ainsi possible de passer outre ce problème, de telle façon que l'objet soit visuellement un peu plus large pour corriger la distance de contact.

3.4 Résolution

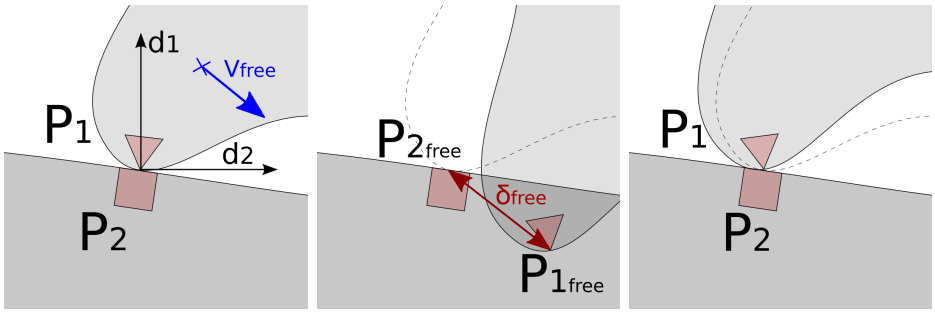
Avant couplage, une contrainte n'a besoin que de calculer la force de correction correspondant à la violation de la contrainte (et de la force normale dans le cas des contraintes de frottement). Cette description est applicable à l'ensemble des contraintes, les spécificités de cette relation décrivant leur comportement. Dans un premier temps, nous nous intéresserons au comportement spécifique des contraintes et proposerons ensuite une solution pour tenir compte du couplage, basé sur un algorithme de type Gauss-Seidel.

3.4.1 Présentation de la méthode

L'intégration de contraintes dans une simulation nécessite de réaliser quelques calculs supplémentaires en préparation de leur résolution ; et le résultat doit ensuite être réinjecté dans le système avant intégration. Nous présenterons ici les différentes étapes de notre méthode.

1. **Déplacement libre.** Nous commençons par simuler un pas de temps du système mécanique, dans le but de savoir comment se déplaceraient les objets s'il n'y avait aucune contrainte dans la scène. Nous désactivons ici toutes les contraintes, et ne tenons pas compte des interactions passées ou futures entre les objets ; c'est ce que nous appelons l'étape de mouvement libre. Elle est nécessaire pour pouvoir calculer les différents paramètres des contraintes, tels que l'interpénétration pour les contraintes de contact ou la violation des contraintes bilatérales. Nous représentons à la figure 3.7.a la configuration initiale de deux objets liés par une contrainte bilatérale, avant l'étape du mouvement libre.
2. **Définition des contraintes**. Les objets peuvent déjà être en contact lors du pas de temps précédent, ou ils ont pu se déplacer et entrer en collision lors du mouvement libre ; il peut donc potentiellement y avoir des collisions à rechercher. Dans ce travail, nous partons de l'hypothèse qu'un algorithme de détection de collision est disponible. Nous créons alors les contraintes de contact en réponse à la sortie de l'algorithme de détection de collision, ainsi que les autres contraintes définies dans la scène, que ce soit statiquement lors du design de la scène, ou dynamiquement suite à certains évènements prédéfinis. Si besoin, des calculs supplémentaires sont réalisés pour obtenir l'ensemble des informations requises par les contraintes pour leur résolution. En particulier, nous calculons la violation δ_{free} (comme montré à la figure 3.7.b), dont la signification peut varier selon le type de la contrainte. On définit cette étape comme la définition des contraintes. Une fois que les directions des contraintes et leurs paramètres sont exprimés, nous pouvons définir l'espace des contraintes, duquel est déduit l'opérateur W.

3. **Résolution des contraintes**. Une fois que les prérequis sont calculés, nous utilisons l'algorithme de résolution des contraintes (présenté à la section 3.4.3), en tenant compte du couplage mécanique qui peut exister entre elles. D'un pas de temps à l'autre, il est possible d'utiliser la cohérence temporelle afin d'améliorer les calculs : en stockant les résultats de la résolution précédente, il est possible de s'en servir comme des paramètres initiaux lors du pas de temps suivant (nous reviendrons sur cet aspect à la section 3.4.6).
4. **Intégration**. A partir des forces calculées comme étant les contributions des contraintes au système, nous définissons des déplacements de corrections pour les points où les contraintes sont appliquées. Enfin, le système mécanique est intégré. Nous utilisons un solveur Euler implicite (voir section 3.4.5), comme nous avons déjà calculé les matrices de rigidité des objets déformables pour le couplage mécanique des contraintes. La figure 3.7.c montre le résultat de la résolution des contraintes et des mouvements de correction pour une scène simple.



(a) mouvement libre (b) définition des contraintes (c) résolution

FIGURE 3.7 – Les différentes étapes de la résolution d'une contrainte bilatérale. La silhouette en pointillés représente la position initiale de l'objet.

3.4.2 Traitement spécifique à la contrainte

Toutes les contraintes présentent des caractéristiques communes (voir section 3.2), nous permettant d'utiliser un même algorithme générique pour toutes les résoudre. Cependant, chacune réalise un rôle différent dans la scène ; ce rôle est exprimé sous la forme d'une loi de comportement qui est propre à chacune. L'algorithme de résolution des contraintes devant rester indépendant de ces lois de comportement, le traitement spécifique à chaque contrainte sera réalisé dans un objet de résolution séparé. Celui-ci permet de stocker des données propres à certaines contraintes, et autorise la création de comportements complexes. Nous y reviendrons à la section 3.4.4.

La résolution d'une contrainte, lorsque l'on ne prend pas en compte le couplage mécanique avec les contraintes voisines, consiste à calculer la force de réponse f qui correspond aux paramètres de la contrainte et à sa loi de comportement. La violation d'une contrainte est notée δ_{free} avant résolution et δ_a durant la résolution, cette dernière pouvant changer de

nombreuses fois puisque nous utilisons une résolution itérative. Nous présentons ici les lois de comportement des contraintes les plus simples que nous voulons simuler :

– Contraintes bilatérales. Résoudre une contraintes bilatérale consiste à ajouter une force proportionnelle à la violation, et de sens opposé. On pose w comme étant le terme, provenant de la diagonale de l'opérateur de Delassus W, qui correspond à cette contrainte précise.

$$f = -W^{-1}\delta_a \tag{3.5}$$

– Contraintes unilatérales. Lorsque la violation est négative la force y est proportionnelle ; une violation positive entraîne une force de réponse nulle.

if $\delta_a < 0$ **then**
  $f = -\delta_a / W_{ij}$
else
  $f = 0$

– Contraintes de frottement. La contrainte est liée à une autre contrainte bilatérale ou unilatérale. Lorsque celle-ci est résolue, la contrainte de frottement récupère la force de réponse, qui est considérée comme étant la force *normale* pour le frottement. Les forces tangentielles sont alors calculées selon la loi de frottement de Coulomb, comme décrit à la section 3.2.3.

3.4.3 Gauss-Seidel modifié

Pour la simulation complète d'un système complexe, nous avons besoin de coupler les contraintes entre elles, puisque la résolution de l'une change le plus souvent les paramètres de ses voisines. Cela sera fait par l'algorithme de résolution des contraintes, en utilisant le couplage mécanique exprimé dans l'espace des contraintes.

Nous utilisons un algorithme de Gauss-Seidel modifié, mais un autre solveur itératif aurait pu être testé (Jacoi, relaxation, etc.), du moment que la résolution d'une contrainte en particulier peut être isolée du couplage mécanique. Notre Gauss-Seidel est très proche des modifications réalisées pour les contacts par Moreau et Jean [MJ96] ; nous généralisons l'algorithme pour tous les types de contraintes.

L'algorithme de Gauss-Seidel présente l'avantage de bien se comporter en présence de contraintes redondantes telles que des contacts multiples entre deux objets, et converge assez bien (il converge dès que le système est bien posé, même si ce n'est pas l'algorithme le plus rapide). Nous illustrons à la figure 3.8.b un cas surcontraint ; l'algorithme converge dans tous les cas, la solution et le nombre d'itérations avant la convergence ne dépendant que de l'ordre dans lequel les contraintes ont été définies.

Durant une itération, nous calculons la violation de chaque dimension de contraintes en bloquant toutes les autres contraintes, et en sommant leur contributions relatives à la dimension qui nous intéresse :

$$\delta_{a_i}^{(k)} = \delta_{free_i} + \sum_{j<i} W_{ij} f_j^{(k)} + \sum_{j>i} W_{ij} f_j^{(k-1)} \tag{3.6}$$

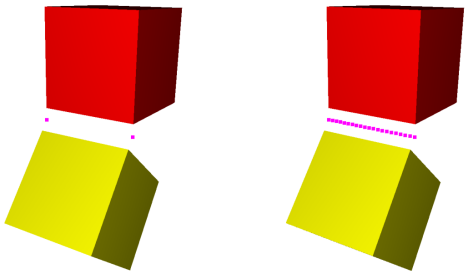


FIGURE 3.8 – Deux cubes rigides connectés par (a) 2 contraintes bilatérales (points violets) ; (b) 20 contraintes largement redondates. La vitesse de convergence est la même pour les deux scènes.

Le calcul de δ_{ai} utilise les $f_j^{(k)}$ dimensions qui ont déjà été calculées avant celle-ci, ainsi que les $f_j^{(k-1)}$ dimensions qui seront mises à jour par la suite.

Après le calcul de la violation courante δ_{ai}, nous pouvons réaliser le traitement spécifique de cette contrainte tel que présenté à la section précédente. Nous utilisons comme paramètres la violation δ_{ai} ainsi que la force calculée à l'itération précédente $f_i^{(k-1)}$; celle-ci peut être utile dans le calcul des contraintes avec frottement. Le résultat de la résolution des contraintes est une nouvelle force de réponse $f_i^{(k)}$, qui remplace immédiatement la précédente. A noter que W_{ij} est constante tout au long de la résolution du système (pour un pas de temps) et n'est donc pas fournie à chaque itération du Gauss-Seidel.

Nous calculons l'erreur par rapport à l'itération précédente comme $err = \sum_i W_{ij}(f_i^{(k)} - f_i^{(k-1)})$ et arrêtons l'algorithme du Gauss-Seidel lorsque celle-ci passe sous un certain seuil, ou qu'un nombre d'itérations est atteint. Nous présentons plus en détails les considérations de calcul de l'erreur et de la convergence de l'algorithme à la section 3.4.6.

Le Gauss-Seidel modifié peut être synthétisé de la façon suivante :

---

répéter
  mise à zéro de l'erreur
  pour toutes contraintes indépendantes **faire**
    stocker la force précédente pour le calcul de l'erreur
    calculer δ_a à partir de l'équation (3.6)
    réaliser le traitement spécifique à cette contrainte
    sauver la nouvelle force
    calculer et ajouter l'erreur
  fin pour
jusqu'à erreur $\leq$ tolérance

Cette méthode peut gérer l'ensemble des scènes courantes, ainsi que des cas plus complexes (la figure 3.9 montre ainsi une configuration de plusieurs contraintes formant une boucle fermée). On a constaté, même si c'est délicat de le prouver théoriquement, que l'algorithme

converge à partir du moment où il n'y a pas de contraintes conflictuelles dans la scène (vouloir tirer et pousser en même temps sur un point, par exemple), ce qui ne devrait pas arriver si l'on fait attention lors de la définition des contraintes.

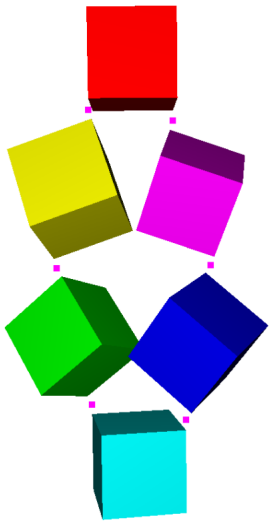


FIGURE 3.9 – Une scène où des objets rigides forment une boucle fermée. Les cubes sont liés l'un à l'autre par des contraintes de point fixe, et des contraintes de contact avec frottement sont créées pour répondre aux collisions pouvant intervenir.

Adapter la méthode que nous proposons à d'autres algorithmes itératifs de résolution des contraintes ne devrait pas poser de problèmes particuliers.

3.4.4 Une note sur l'architecture logicielle

Du côté de l'implantation, nous utilisons le paradigme de programmation orientée objet, qui nous permet de définir une contrainte générale que l'algorithme de résolution peut utiliser, et dont nous dériverons toute une famille de contraintes, chacune avec son comportement et ses données propres. Cette méthode présente certains avantages par rapport à un algorithme global de résolution des contraintes.

Chaque objet de contrainte peut stocker individuellement des données, ses propres paramètres. Lorsqu'à chaque itération, l'algorithme du Gauss-Seidel demande à la contrainte de calculer sa force de réponse, il ne lui fournit que δ_a et f dont les valeurs sont définies à ce moment de la résolution. L'opérateur W ne change pas et est connu au début de la phase de résolution des contraintes.

Des paramètres additionnels peuvent être stockés localement par les objets pour leur usage interne. Par exemple, les contraintes de contact avec frottement enregistrent leur coefficient de frottement, qui peut alors être différent pour chaque contrainte. Il est ainsi possible de simuler le contact entre des objets dont le frottement à la surface varie de l'un à l'autre.

Le plus souvent, un algorithme de Gauss-Seidel réalise le calcul une ligne à la fois, chaque ligne représentant alors une dimension d'une contrainte. Cette méthode est parfaitement adaptée aux contraintes unilatérales simples, et ne pose pas de problème pour les contraintes bilatérales de dimensions multiples qui peuvent être transformées en plusieurs contraintes unidimensionnelles. Cependant, les contraintes de frottement ou de point glissant avec frottement nécessitent de travailler avec au moins 3 dimensions simultanément, et l'on désire que l'algorithme de résolution s'adapte automatiquement.

Nous proposons que la décision d'utiliser soit une seule ligne à la fois, soit plusieurs lignes en même temps ne soit pas de la responsabilité de l'algorithme de résolution. Nous désirons que chaque contrainte puisse informer l'algorithme de ses besoins, et que celui-ci s'adapte en fonction. De cette façon, les contraintes de contact simple pourront utiliser une ligne tandis que des contraintes plus complexes seront créées et feront appel lors de leur résolution aux dimensions additionnelles dont elles ont besoin.

Un aspect résultant de ceci est qu'il est possible dans certains cas d'optimiser la résolution des contraintes pour une convergence plus rapide de l'algorithme du Gauss-Seidel. Ainsi, pour les contraintes comportant plusieurs dimensions tels que les contraintes de point fixe 3d, nous choisirons une résolution par bloc (3×3) plutôt que de séparer la contrainte en 3 contraintes unidimmensionnelles résolues chaque ligne indépendemment des autres. Nous calculons ainsi $\mathbf{f} = W_b^{-1}\delta_{\mathbf{a}}$, avec W_b la matrice bloc 3×3 et en précalculant une seule fois avant la résolution la matrice inverse W_b^{-1}.

Cette aproche permet la simulation de contraintes nonholonomiques ou nonlinéaires. Nous ne sommes pas non plus limités aux contraintes pouvant être exprimées comme une simple relation entre le déplacement et la force de réponse. Les contraintes de contact peuvent être désactivées durant la résolution si les objets ont été séparés par d'autres contraintes, les contraintes de frottement peuvent être adhérentes ou glissantes selon les valeurs de leurs paramètres propres.

Cela ouvre un ensemble de possibilités, et nous pouvons imaginer des contraintes dont le comportement change radicalement durant la résolution des contraintes. Nous présenterons ainsi à la section 4.1.3 une contrainte dont la direction de la force de réponse change selon l'amplitude de celle-ci et certains paramètres de la scène. Il faut cependant rester prudent dans la conception de telles contraintes, en vérifiant que le système converge bien quelque soit le comportement actuel de celles-ci.

La conception de cette méthode a été pensée afin de promouvoir le prototypage rapide et la création de nouveaux types de contraintes. En effet, ajouter une nouvelle contrainte dans le framework se limite souvent à décrire sa loi de comportement. L'algorithme de résolution des contraintes n'a pas besoin d'être modifié, car il ne manipule les contraintes que via une interface générique et n'a pas besoin de connaître les spécificités de chacune. Nous n'imposons pas de règles strictes sur la conception des contraintes, car elles peuvent choisir leurs dimensions, et les données dont elles ont besoin pour leur résolution.

Actuellement, les objets de résolution des contraintes sont demandés à chaque pas de temps par l'algorithme du Gauss-Seidel, et sont détruits après résolution. Ceux-ci ne peuvent donc pas être utilisés pour stocker des informations d'un pas de temps à l'autre. Il est alors nécessaire de passer par une autre classe si des données doivent être conservées tout le long de la simulation, celles-ci pouvant alors être accédées lors de la création ou de la destruction des objets de résolution.

3.4.5 Integration implicite

Une fois les forces dues aux déformations et aux contraintes mises à jour, nous réalisons l'intégration du système mécanique avec un schéma d'Euler implicite. Le choix du schéma d'intégration n'est pas fixé par l'algorithme du Gauss-Seidel ou de la méthode de définition des contraintes que nous utilisons, mais est la conséquence des propriétés des objets que l'on désire manipuler dans les simulations qui nous intéressent.

L'intégration explicite est un schéma facile à mettre en œuvre et dont la parallélisation est possible. La méthode d'Euler explicite utilise les forces du pas de temps n pour calculer les vitesses du pas de temps $n+1$. D'autres schémas explicites tels que Runge-Kutta à l'ordre 2 ou 4 offrent une meilleure précision [But03]. La *condition de Courant* [CFL67] précise que le pas de temps d'intégration Δt doit être inversement proportionnel à la racine de la raideur du système. Si ce critère n'est pas respecté, le système s'éloignera rapidement de la solution car les forces seront considérées comme étant constantes sur un trop grand interval de temps. Le système divergera car il sera soumis à des variations trop brusques. Face à ce problème d'équation raide, la stabilité n'est assurée que pour des objets massifs ou à des pas de temps très faibles.

Dans notre cas, nous voulons simuler des objets tels que une aiguille de biopsie ou encore un fil de suture, qui sont tous deux très légers et très raides. De plus, ces objets seront en interaction avec d'autres objets, qui sont eux plutôt mous. Plutôt que d'utiliser alors une intégration explicite nécessitant un pas de temps très faible (de l'ordre de 10^-4 seconde), nous allons utiliser le schéma d'Euler implicite. Terzopoulos [TPBF87] a utilisé ce schéma dès les premiers travaux sur les simulations d'objets déformables, mais il a fallu attendre dix ans avant que ses avantages soient mis en valeur, avec la simulation de vêtements par Baraff et Witkin [BW98] ou d'objets masses-ressorts 3D par Dresbrun [DSB99].

L'idée du schéma d'Euler implicite est de remplacer les forces du pas de temps n par celles du pas de temps $n+1$. On obtient alors une équation non linéaire, et plutôt que de la résoudre exactement (ce qui nécessiterait un processus itératif), nous utilisons la méthode de Newton-Raphson qui consiste à réaliser une approximation linéaire par pas de temps avec un développement de Taylor Young à l'ordre 1. Ce shéma se révèle être très stable, et l'amortissement inhérent à ce schéma est utile dans nos simulations puisqu'il y a intervention de l'utilisateur. Enfin, le pas de temps d'intégration qui est permis par ce schéma est assez large (nous choisissons en général 20 ms) et permet ainsi de garder un rafraîchissement raisonnable de la compliance et des contraintes. Nous utilisons alors une méthode de "time-stepping" pour gérer toutes les contraintes en même temps, quelque soit leur ordre d'arrivée dans le pas de temps.

3.4.6 Convergence de l'algorithme

La résolution des contraintes suit donc un processus itératif, dont le résultat à chaque itération doit normalement se rapprocher de la solution théorique. Bien évidemment nous ne connaissons pas cette solution théorique, et ne pouvons pas savoir exactement à quelle distance s'en trouve la solution calculée par l'algorithme à une itération donnée. Nous nous basons alors sur l'erreur d'une itération à l'autre pour déterminer lorsque le Gauss-Seidel aura

convergé. Elle correspond à la distance parcourue par le point de contrainte durant l'itération. Pour une contrainte donnée, située à la ligne l et de dimension n, nous calculons l'erreur de la façon suivante :

$$Erreur = \sqrt{\sum_{i=0}^{n}\sum_{j=0}^{n}\left[W_{l+i,l+j}(f_i^{(t)} - f_i^{(t-1)})\right]^2} \tag{3.7}$$

Pour une contrainte de dimensions 1, le calcul se simplifie : $Erreur = \left\|W_{l,l}(f_l^{(t)} - f_l^{(t-1)})\right\|$. D'autres travaux préconisent de calculer une erreur relative (en pourcentage) en calculant $\frac{\left\|f_l^{(t)} - f_l^{(t-1)}\right\|}{\left\|f_l^{(t)}\right\|}$. Nous préférons cependant avoir une contrainte homogène à une distance, qui est souvent plus facile à paramétrer.

Nous établissons alors un critère sur l'erreur pour choisir le moment d'arrêter l'algorithme. Trois choix s'offrent à nous :

- Le total des erreurs de chaque contrainte est inférieur à une tolérance prédéfinie. Lorsque le nombre de contraintes augmente, l'algorithme prendra plus de temps à converger.
- La moyenne des erreurs est inférieure à la tolérance. Cela revient à augmenter la tolérance au fur et à mesure que l'on ajoute des contraintes. Ce calcul de l'errer présente le même soucis que la méthode précédente, à savoir qu'une contrainte peut ne pas être vérifiée si les autres ont déjà convergé.
- L'erreur de chaque contrainte est inférieure à la tolérance. La méthode la plus précise qui s'assure que chaque contrainte est vérifiée, qui offre le plus de garantie de ne pas arrêter la résolution sur un minimum local, mais également la plus longue à converger dans la majorité des cas.

Nous utilisons en général la moyenne de l'erreur pour les scènes faisant intervenir principalement des objets solides et des contraintes de contact car la convergence est rapide et les erreurs ont une influence minime et sont souvent corrigées au pas de temps suivant. Cependant, pour la suite de ce travail portant sur l'insertion d'aiguille et la suture, et faisant intervenir des contraintes dans des objets déformables en même temps que de l'autocollision de modèles fins, nous cherchons à garder une erreur par contrainte qui soit stable. En effet, autoriser ne serait-ce qu'une contrainte à ne pas être vérifiée peut avoir des conséquences importantes sur la simulation, tel qu'un fil de suture qui passe à travers lui-même au niveau d'un nœud car la contrainte de contact gérant son autocollision présente une erreur ponctuellement supérieure à la distance de contact.

L'algorithme du Gauss-Seidel convergera toujours lorsque le système est bien posé (dès lors qu'il n'y a pas de contraintes antagonistes), mais sa vitesse de convergence est souvent plus lente que celle d'autres algorithmes. Plusieurs techniques peuvent être mises en place selon les cas pour améliorer la vitesse de convergence de la résolution des contraintes. Nous avons testé principalement deux méthodes, à savoir le départ à chaud du Gauss-Seidel et la sur-relaxation successive.

Départ à chaud

Cette méthode consiste à corriger le fait que le Gauss-Seidel recommence à zéro à chaque pas de temps. Plutôt que d'initialiser les forces à 0 pour toutes les contraintes, on veut repartir de leur valeur au pas de temps précédent. On se base sur le principe de la cohérence temporelle, dans le sens où on peut considérer que la simulation change peu d'un pas de temps à l'autre. Lorsque deux objets sont en contact, la force permettant de résoudre la collision une fois sera certainement très proche de celle calculée pour la fois suivante, d'autant plus si les objets sont immobiles. L'algorithme de résolution des contraintes repart donc d'une position plus proche de la solution théorique, et nécessite alors en général moins d'itérations pour converger. En pratique, on fera en sorte que chaque contrainte enregistre les forces qui lui sont appliquées après résolution, et on les lui redemandera au pas de temps suivant.

Pour les contraintes statiques (définies au début de la simulation), cette méthode est efficace et facile à mettre en œuvre. Il faut cependant que la direction des contraintes reste la même ou change très peu d'un pas de temps à l'autre afin que les valeurs des forces que l'on enregistre gardent leur cohérence. On gardera donc également en mémoire les directions de ces contraintes afin de pouvoir les mettre à jour selon la rotation la plus petite possible.

Le cas des contraintes de contact est plus délicat à gérer. En effet, elles sont créées directement par l'algorithme de détection de collision, et sont supprimées à chaque fin de pas de temps. De plus, même en considèrant un point de contact en particulier, celui-ci peut se déplacer sur la surface des objets et ainsi changer de primitive de collision. Détecter que la nouvelle contrainte correspond à celle qui a été supprimée précédemment sur le triangle voisin n'a rien de direct et rallongerait les calculs, ce qui est à l'opposé de ce que l'on cherche à faire. La normale du contact elle même peut changer, et initialiser cette contrainte avec la valeur calculée pour une direction différente peut être contre-productive. Nous n'avons pas à ce jour de méthode correcte pour initialiser les contraintes de contact, l'approche naïve mais rapide que nous avons choisie (consistant à choisir des directions latérales pseudo-aléatoires [7]) est suffisante dans les cas où les contacts ne changent pas trop d'un pas de temps à l'autre, mais est inefficace pour le départ à chaud lorsque les objets sont en mouvement. Comme les mesures le montrent, il peut arriver qu'en initialisant les contraintes avec des valeurs faussées, le nombre d'itérations nécessaires pour que le Gauss-Seidel converge soit plus grand que si on avait initialisé les forces à 0.

Relaxation

Le principe de la relaxation consiste à exagérer artificiellement la correction qui est faite lors d'une itération du Gauss-Seidel. En partant de la valeur de la force à l'itération précédente f_i^t et du résultat courant de l'algorithme $f_i^{(t+1)}$, on calcule la valeur de cette itération en appliquant un coefficient ω de relaxation : $\tilde{f}_i^{t+1} = (1-\omega) f_i^t + \omega f_i^{t+1}$. Lorsque $\omega = 1$ on garde le résultat du Gauss-Seidel ; en choisissant $\omega < 1$ on est en sous-relaxation (convergence plus lente) ; la sur-relaxation correspond à $\omega > 1$.

7. Les directions latérales sont obtenues par un calcul simple se basant sur la direction normale. Lorsque celle-ci change, les deux vecteurs tangentiels calculés n'ont aucune cohérence temporelle.

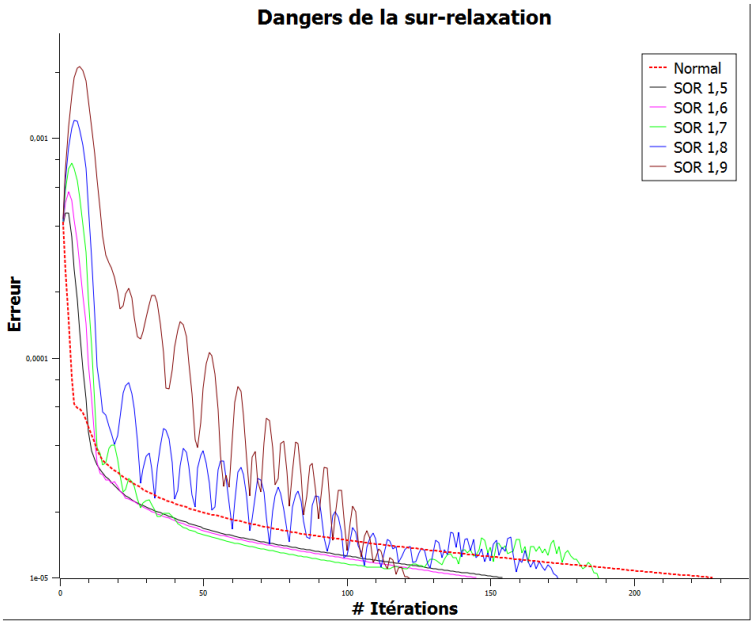


FIGURE 3.10 – Influence de ω sur la vitesse de convergence pour une simulation d'insertion d'aiguille similaire à la figure 3.13. Pour des valeurs > 1.6, des oscillations commencent à apparaître.

La sur-relaxation successive (SOR) permet pour des valeurs de ω supérieures à 1 d'obtenir une convergence plus rapide. Cependant, il faut également éviter les valeurs trop élevées, car on observerait rapidement des oscillations de l'erreur. Celles-ci sont dûes au fait que les contraintes ne peuvent pas atteindre leur solution, la dépassant d'un côté puis de l'autre. Les valeurs optimales de ω dépendent certainement du type des contraintes utilisés ainsi que du rapport masse/raideur. Suite aux mesures que l'on présente à la section suivante, on remarque des oscillations à partir de $\omega = 1.7$ tout en ayant encore une amélioration de la vitesse de convergence (voir figure 3.10), qui vont en augmentant jusqu'à ce que la solution diverge pour $\omega = 2$. En pratique, nous choisissons une valeur de 1.5, pour laquelle on obtient de bons résultats pour l'ensemble des scènes testées, tout en ne voyant pas apparaître d'oscillations.

Mesures

Pour mettre en évidence l'intérêt des méthodes que nous venons de présenter, nous avons réalisé des mesures sur des simulations assez simples en activant sélectivement certaines d'entre elles. Trois scènes ont été retenues : un collier de cubes rigides liés par des contraintes de point fixe (figure 3.11) ; une pile d'objets rigides où des contraintes de contact sont créées automatiquement (figure 3.12) ; l'insertion d'une aiguille souple dans un objet déformable, avec traction latérale de celle-ci une fois qu'elle a été insérée (figure 3.13).

Afin que le nombre d'itérations de l'algorithme de résolution soit significatif pour observer l'impact des méthodes d'accélération, nous avons délibérement choisi une valeur d'erreur tolérée très basse. Pour chaque scène, on mesure le nombre d'itérations requises pour que l'erreur moyenne des contraintes devienne inférieure à une valeur de tolérance définie. Nous

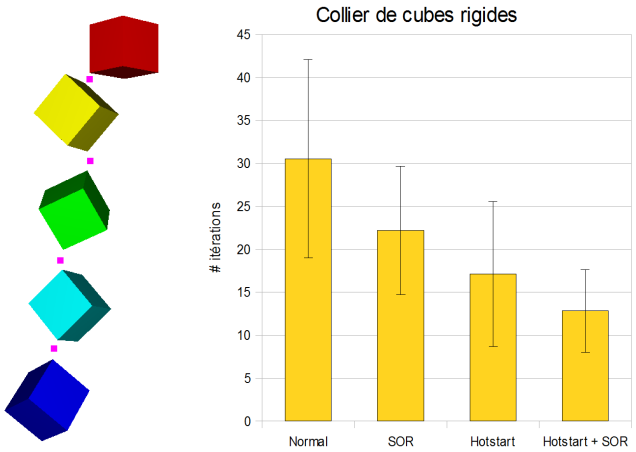


FIGURE 3.11 – Mesures pour un ensemble d'objets rigides liés par des contraintes de point fixe, représentant 12 contraintes bilatérales.

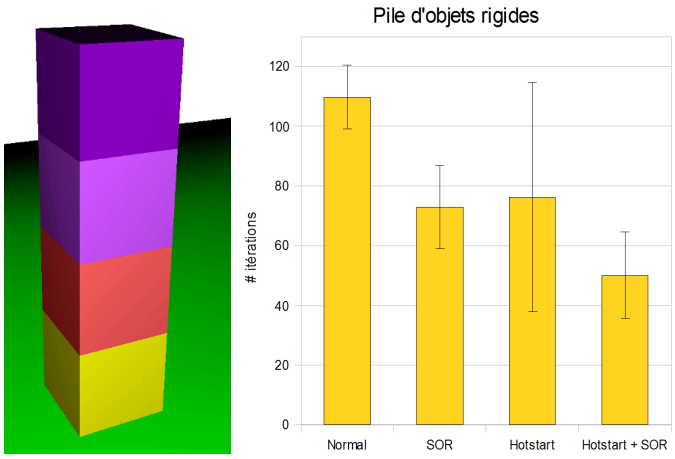


FIGURE 3.12 – Mesures pour une pile d'objets rigides en contact permanent. Le nombre de contraintes lors des mesures est d'environ 250. Cette scène requiert une erreur très basse pour que l'empilement ne se mette pas à osciller.

réalisons une moyenne sur 1000 pas de temps lorsque le nombre de contraintes reste stable, sur le même intervalle de temps quelque soit la méthode testée. On choisi arbitrairement $\omega = 1.5$ pour les tests de sur-relaxation. Les schémas suivant présentent donc pour chaque scène : la valeur **normale** sans méthode d'accélération de la convergence ; avec de la sur-relaxation (**SOR** pour *successive over-relaxation*) ; avec la méthode du départ à chaud (**hotstart**) ; et enfin avec les deux méthodes actives simultanément.

Chaque méthode présentée précédemment améliore donc sensiblement la vitesse de convergence du Gauss-Seidel, et donc le temps nécessaire pour résoudre nos contraintes. De plus, l'utilisation des deux méthodes conjointement donne les meilleurs résultats pour chaque scène testée.

Cependant, on remarque que le départ à chaud peut être inefficace dans le cas des contraintes de contact, pour les raisons évoquées plus haut. Bien qu'elle soit peut utile dans cette scène en particulier, on observe des temps de calcul au moins similaires dans la plupart

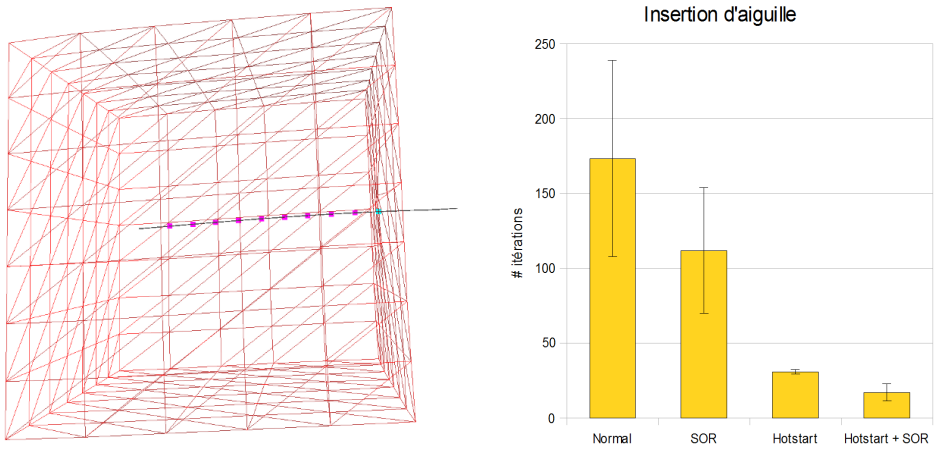


FIGURE 3.13 – Mesures pour une insertion d'aiguille souple dans un objet déformable. On réalise les mesures à la fin de l'insertion, lorsque l'aiguille est immobile et que 34 contraintes ont été créées.

des cas. Au défaut d'avoir une méthode de mémorisation des forces exacte, on désactivera le départ à chaud dans certains scènes, et ce uniquement pour les contraintes de contact.

Pourtant, la meilleure façon pour obtenir un faible temps de calcul reste de choisir une tolérance adaptée à la scène. En effet, comme on peut l'observer sur le schéma de la figure 3.14, la vitesse de convergence n'est pas linéaire. Dès lors que les contraintes sont interdépendantes et connectées à des matériaux déformables, on se rapproche plus d'une courbe en $y = 1/x$. Ainsi, on remarque pour cet exemple précis qu'il faut multiplier par 10 le nombre d'itérations si on veut diviser par 10 l'erreur sur la violation des contraintes. Cependant, le temps de calcul par itération du Gauss-Seidel reste sensiblement constant. Il faut donc trouver pour chaque scène un compromis entre l'erreur acceptée, et le temps de calcul. Les contraintes de contact requièrent dans certains cas une très faible erreur, par exemple lorsque l'on manipule des objets fins en collision qui pourraient facilement passer au travers l'un de l'autre si la contrainte n'est pas exactement vérifiée. La solution sera alors d'utiliser deux méthodes de tolérance simultanément : on demandera que l'erreur de chaque contrainte de contact pouvant poser problème soit inférieure à la tolérance, mais on calculera la moyenne de l'erreur pour toutes les autres contraintes afin que le problème ne soit pas plus long à résoudre au fur et à mesure qu'il gagne en taille.

3.5 Exemples de contraintes

Nous allons récapituler dans cette partie les principales contraintes que nous utilisons et présenterons plusieurs applications de notre méthode d'interaction. Nos travaux sont intégrés dans le framework *SOFA*, et nous employons les modèles déformables, les méthodes de détection de collision de même que les schémas d'intégration qu'il propose. Les mesures sont réalisées sur un Core 2 Quad 2.66 Ghz avec 4 Go RAM.

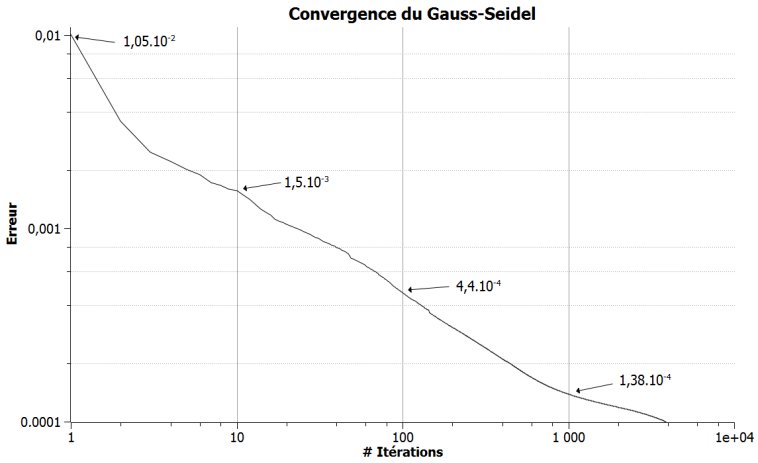


FIGURE 3.14 – Mesure de la convergence du Gauss-Seidel sur une scène complexe de suture, sans méthode d'accélération de la convergence.

3.5.1 Point fixe

La contrainte de point fixe consiste à empêcher tout mouvement relatif pour les deux particules concernées par la contrainte. Elle correspond à une contrainte bilatérale de dimension 3, dans le sens où l'on contraint chaque direction de l'espace 3D. Rappelons qu'une contrainte bilatérale simple ne contraint le déplacement relatif que suivant une seule direction. Ainsi une contrainte de dimension 1 limitera le déplacement d'une particule sur un plan, une contrainte de dimension 2 permettra le déplacement uniquement le long d'une ligne.

Nous utilisons généralement les contraintes de point fixe pour assembler entre eux plusieurs objets, qui vont rester collés l'un à l'autre aussi longtemps que les contraintes sont actives. Le nombre de degrés de liberté de chaque objet concernés par la contrainte gouverne le déplacement que pourront avoir les objets entre eux. En effet, bien que l'on contraigne les positions des particules, les objets sont libres de tourner autour de celles-ci. Ainsi, si l'on fixe un seul point de chaque objet, chacun pourra librement tourner selon les 3 axes. Une fois que l'on fixe deux paires de particules, on élimine la rotation suivant deux axes, et les objets ne peuvent plus que pivoter autour de l'axe qui traverse les deux points de contraintes. Enfin, à partir de 3 contraintes de point fixe non alignées, la rotation globale des objets est contrainte.

La figure 3.15 montre un ensemble de cube rigides, chacun relié au suivant via l'utilisation d'une contrainte de point fixe. Les cubes ne sont donc pas fixés en rotation et peuvent librement tourner sur eux-mêmes sans entraîner leurs voisins. Il faut noter que les points de contraintes que nous avons utilisés pour relier ces cubes ne sont pas placés directement sur les degrés de liberté de ceux-ci, mais légèrement en dehors de leur modèle de collision. En effet, puisque nous utilisons une détection de collision par proximité, nous ne pouvons pas réunir exactement deux cubes par leurs sommets sans que des contraintes de réponse à la collision soient créées. Celles-ci seraient antagonistes avec les contraintes de point fixe, et la résolution des contraintes n'aurait pas de solution. Dans la figure 3.16 nous avons utilisé deux contraintes de point fixe pour que le mouvement des deux cubes soit limité à une sorte de charnière.

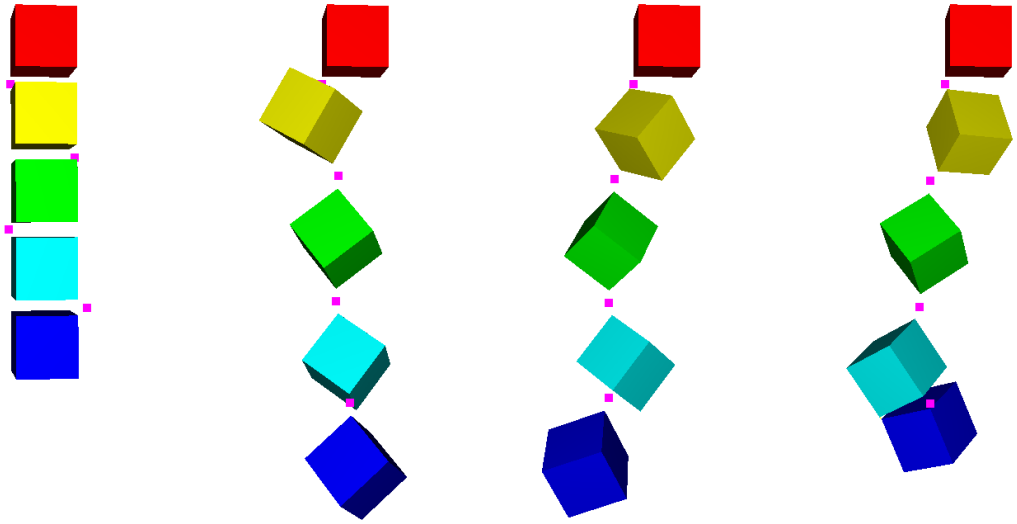


FIGURE 3.15 – Ensemble de cubes rigides reliés entre eux par des contraintes de point fixe, le cube du haut n'étant pas simulé. Ils peuvent librement tourner autour d'eux mêmes car leur rotation n'est pas fixée.

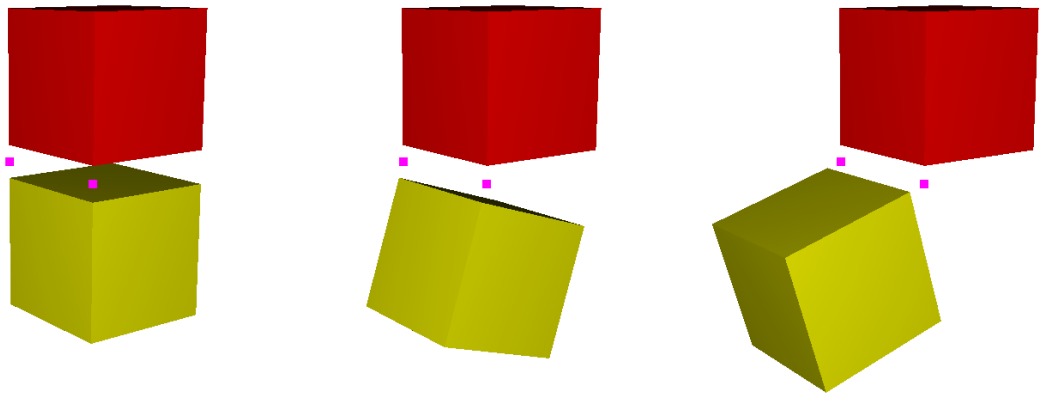


FIGURE 3.16 – Deux cubes rigides reliés par deux contraintes de point fixe. Le cube du haut est fixe, celui du bas étant alors contraint à tourner autour de l'axe passant par les deux points de contraintes.

Bien que l'on utilise le plus généralement des contraintes bilatérales de dimension 3 (les contraintes de point fixe), il est tout à fait envisageable de choisir ponctuellement une contrainte avec un nombre de dimensions différent, pour s'adapter au besoins particuliers d'une situation. Nous avons ainsi présenté les contraintes limitant le déplacement d'une particule à un plan ou à une droite. Cependant, il est également concevable dans certains cas de créer des contraintes bilatérales de dimension supérieures à 3. En effet, si dans un espace 3d on représente un objet déformable par un ensemble de particules ayant chacune 3 degrés de liberté, le déplacements des objets rigides est couramment réduit à celui d'une particule en ayant 6 (3 pour la position de l'objet, et 3 pour sa rotation). Dans ce cas de figure, une

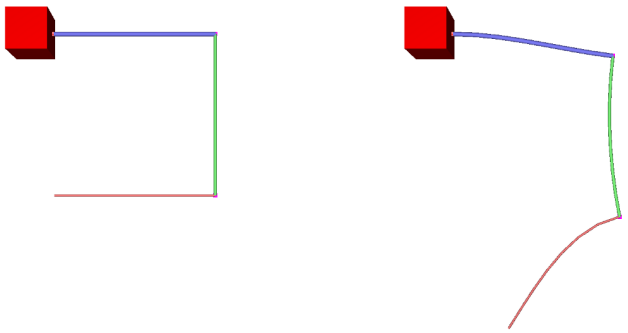


FIGURE 3.17 – Plusieurs modèles de poutres déformables sont fixées entre elles par des contraintes de dimension 6.

contrainte de dimension 6 par exemple pourra être utilisée pour fixer un objet rigide à un autre, aussi bien au niveau de sa position que de sa rotation.

Pour illustrer les contraintes de point fixe avec rotation, nous avons utilisé un modèle de poutre déformable 1D constitué d'un ensemble de nœuds où chacun enregistre également les informations de rotation locale. Nous pouvons alors utiliser la position et la rotation de chaque particule pour créer une contrainte de dimension 6. Ce type de contrainte est alors très intéressant pour contraindre ce modèle de poutre à un objet rigide ou à lui-même, permettant ainsi de réaliser facilement une arborescence de cables (exemple simplifié à la figure 3.17).

3.5.2 Contact

Les contraintes de contact sont certainement les plus fréquentes dans la majorité des simulateurs physiques mais également parmis les plus délicates à traiter. Le cas particulier de la résolution de contact a fait l'objet de nombreuses publications et est toujours un domaine de recherche très actif. Ces contraintes sont créées dès lors qu'une collision entre deux objets est détectée, afin d'empêcher leur interpénétration. Elles se démarquent des autres types de contraintes par le fait qu'elles sont générées dynamiquement par un algorithme durant la simulation, plutôt que d'être définies statiquement lors de l'écriture de la scène à simuler. Nous créons donc une contrainte de contact pour chaque point que nous donne l'algorithme de détection de collision, et les supprimons à la fin du pas de temps.

Ce sont des contraintes unilatérales, puisqu'elles se limitent à repousser les deux objets jusqu'au point où ils ne sont plus en interpénétration, et ne cherchent pas à les réunir ensemble lorsqu'ils se séparent (contrairement aux contraintes bilatérales, qui elles fonctionnent dans les deux sens). Elles agissent le plus souvent dans la direction du contact : ainsi si une particule entre en collision avec un triangle, on choisira comme direction de la contrainte la normale de ce triangle.

Le même type de contrainte permet de simuler le contact entre des modèles hétérogènes. Nous simulons à la figure 3.18 le comportement d'une chaîne dont les maillons utilisent des formalismes différents pour leur simulation.

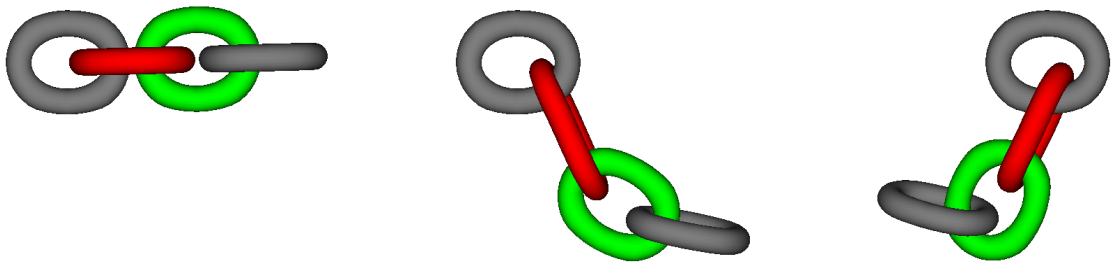


FIGURE 3.18 – Simulation d'une chaîne articulée. Le premier maillon est fixe et non déformable. Les autres maillons sont simulés respectivement par la méthode des éléments finis, des masses-ressorts, et un simulateur d'objets rigides.

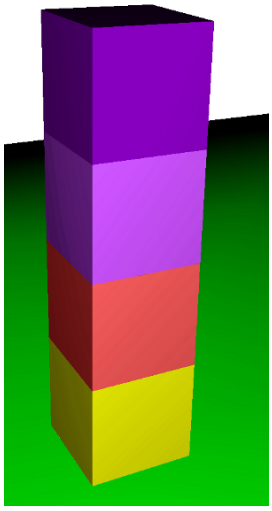


FIGURE 3.19 – Empilement d'objets rigides dont la collision est résolue à l'aide de contraintes de contacts.

Notre méthode de résolution de contraintes permet de gérer des cas de collision complexes, tels qu'un empilement d'objets rigides (voir figure 3.19). Ce cas montre une grande interdépendance entre les collisions, dans le sens où la résolution du contact au niveau du sol a une influence sur le contact au dessus du cube, et ainsi de suite. L'algorithme du Gauss-Seidel nécessitera un plus grand nombre d'itérations, au fur et à mesure que l'on ajoutera des objets au dessus de la pile. Cependant l'empilement est stable, et on peut utiliser la cohérence temporelle pour accélérer grandement la convergence de la résolution.

Détection de collision par distance

Faisons une petite parenthèse dans la présentation des contraintes pour parler rapidement de la détection de collision que nous utilisons. Via le framework *SOFA*, nous employons la détection de collision par proximité, qui consiste à répondre à la collision avant même que les objets entrent en contact. Le principe est de créer une contrainte de contact lorsque la distance entre les deux primitives de collision est inférieure à une certaine **distance d'alarme**.

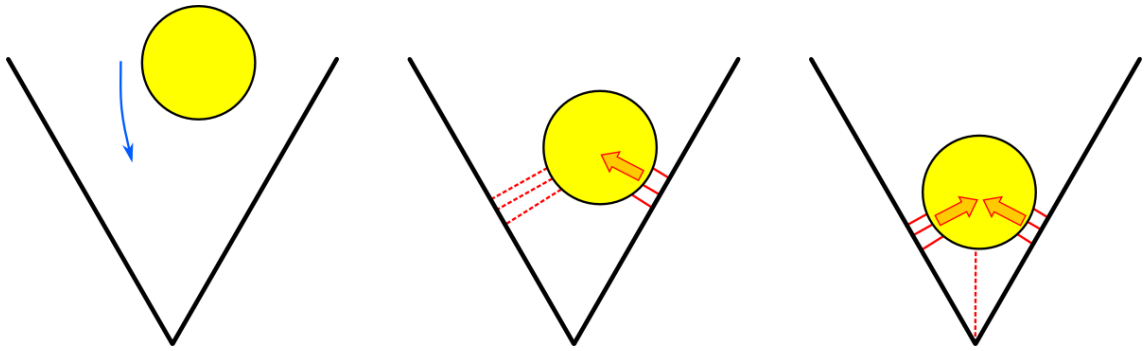


FIGURE 3.20 – Une sphère rigide tombant dans un 'V' formé par deux plans, cas bien géré par la détection de collision par distance qui crée des contraintes avant que le contact soit effectif. Les contraintes sont représentées par des traits rouge, celles n'étant pas actives après résolution sont en pointillés.

Cependant, cette contrainte ne sera active, c'est à dire qu'elle cherchera à séparer les objets, uniquement si la distance devient inférieure à la **distance de contact** définie pour la scène.

Cette méthode présente pour nous certains avantages sur les détections continues ou discrètes après contact. Tout d'abord, elle est simple et rapide, convenant bien aux simulations temps réel. Mais surtout, elle réagit bien aux cas où de nouvelles collisions apparaissent durant la résolution des contraintes, ce qui est fréquent pour les objets déformables ou lorsque plusieurs objets sont empilés.

Prenons l'exemple d'une sphère tombant dans un 'V' formé par deux plans (voir figure 3.20). La sphère entre tout d'abord en contact avec le plan de droite, mais la résolution de la collision a tendance à la pousser à travers le plan de gauche. Avec une détection de collision après contact, il faudrait une seconde réponse à la collision pour prendre en compte ce second plan. Il faudrait ainsi plusieurs itérations de l'ensemble détection et réponse à la collision pour gérer complètement ce cas.

Une détection de collision par distance va créer tout de suite 2 contraintes de contact, une de chaque côté. Celle sur le plan de droite sera active dès le début de la résolution, alors que celle de gauche, dont la violation est comprise entre la distance d'alarme et celle de contact, peut être activée durant la résolution si l'objet se déplace vers la gauche sous l'effet des contraintes. La méthode par proximité (ou distance) est donc bien adaptée à la résolution de type "time-stepping" que nous utilisons, où toutes les contraintes sont résolues en même temps, quelque soit leur ordre d'arrivée dans le pas de temps. Une résolution de type "event-driven" qui utilise l'information de l'instant de collision utilisera de préférence une méthode de collision continue.

Cependant, la détection de collision par distance présente le même désavantage que les autres méthodes discrètes, dans le sens où une collision peut échapper à l'algorithme, ici lorsque les objets en mouvement se déplacent plus rapidement durant un pas de temps que la distance d'alarme. L'autre inconvénient est la conséquence directe du concept de distance de contact, puisque les objets ne seront pas réellement en contact l'un avec l'autre : ils lévitent légérement au dessus du sol et sont toujours séparés par cette distance de contact. Avec le mécanisme de multi-représentation des objets, on peut facilement contourner ce problème, en

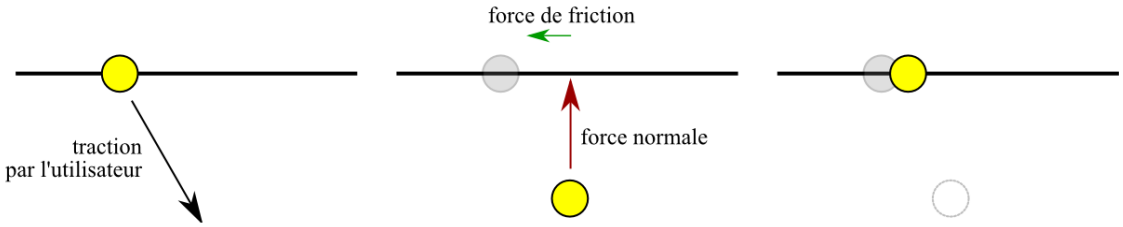


FIGURE 3.21 – Cas simple du frottement : un objet glissant le long d'une ligne, dont le mouvement est freiné par le frottement proportionnellement à la force qui est appliquée dans le sens normal à la ligne.

créeant un modèle de collision qui soit légèrement plus contracté que le modèle visuel (d'une distance équivalente à la moitié de celle de contact).

3.5.3 Frottement

Les contraintes de frottement sont toujours créées en relation avec une autre contrainte, le plus souvent de contact. Elles ajoutent un comportement dissipatif s'opposant au mouvement relatif entre les deux objets.

Nous modélisons le frottement en utilisant la loi de Coulomb, dont le cône est évalué à chaque itération de la résolution des contraintes. Ainsi, nous nous servons du cône exact plutôt que d'une approximation en pyramide avec un nombre de faces fini.

La figure 3.21 présente le cas le plus simple du frottement. Un objet est autorisé à glisser le long d'une ligne et nous calculons une force de frottement s'opposant au déplacement selon la loi de Coulomb, en utilisant l'amplitude de la force de la contrainte qui ramène l'objet sur la ligne.

Contact avec frottement

Si un coefficient de frottement est défini pour une scène donnée, nous attachons une contrainte de frottement à chaque contrainte de contact créée suite à la détection de collision. Elle sera chargée de résister au déplacement dans les deux directions tangentes à la surface de contact. Pour connaître la force de frottement à appliquer, on a besoin de récupérer la force de réponse de la contrainte de contact (la force normale), c'est pourquoi on construit une contrainte de dimension 3, qui sera chargée de résoudre en une seule fois le contact frottant.

Chaque contrainte pouvant avoir ses propres données, il est possible de choisir un coefficient de frottement pour chaque paire d'objets entrant en collision, de telle sorte à modéliser des propriétés physiques particulières à chacun. La figure 3.22 montre l'influence du coefficient de frottement sur le déplacement libre d'objets rigides sur un plan incliné.

3.5.4 Point glissant

Afin de tester notre framework de contraintes, nous avons créé une contrainte de point glissant, qui a pour but de maintenir un objet sur une courbe (voir définition complète dans

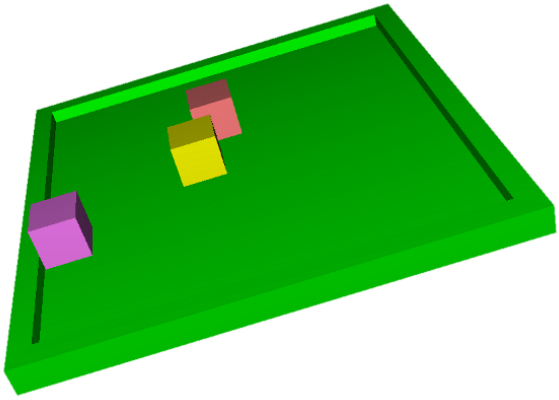


FIGURE 3.22 – Influence du coefficient de frottement. Trois cubes rigides glissent sur un plan incliné avec un coefficient de frottement différent (de haut en bas, respectivement : 0.35, 0.3 et 0.25).

la thèse de Lenoir [Len04]). C'est une contrainte bilatérale, définie entre une courbe décrite comme un ensemble de nœuds connectés par des segments rectilignes, et un ensemble de points appartenant à l'objet devant être contraint.

Sur un segment

Durant la phase de mouvement libre, l'objet que l'on veut faire glisser sur le segment peut se déplacer dans toutes les directions de l'espace, aussi bien dans la direction tangentielle à la ligne porteuse que dans les directions normales. La contrainte de point glissant devra ramener l'objet sur le segment à chaque pas de temps lors de la résolution des contraintes. Nous n'allons donc que contraindre les directions normales à cette ligne, afin que l'objet puisse glisser le long de celle-ci.

A chaque pas de temps lors de la mise à jour des contraintes, nous calculons le point sur le segment qui est le plus proche du point de contrainte, à l'aide d'une projection orthogonale. Nous interrogeons également la direction tangentielle à la courbe en ce point. Une contrainte bilatérale est ensuite créée selon deux directions orthogonales quelconques (voir figure 3.23), la violation de la contrainte étant la distance entre l'objet et le segment suivant chacune de ces directions.

La direction tangentielle n'est dans un premier temps pas contrainte, l'objet peut alors se déplacer le long du segment durant le mouvement libre, il n'est que projetté sur le point de la courbe le plus proche. Nous avons également créé une spécialisation de cette contrainte de point glissant, en ajoutant du frottement dans la direction tangentielle au segment, dont l'amplitude dépend de la force tirant l'objet hors du segment. La contrainte de point frottant a été illustrée à la figure 3.21.

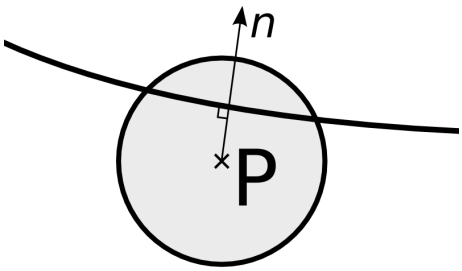


FIGURE 3.23 – La contrainte de point glissant définie une direction vers le point de la courbe le plus proche, la violation est calculée comme la distance entre ce point et l'objet.

Le long d'une courbe

Tout l'intérêt des contraintes de point glissant se révèle lorsque l'on généralise leur action à une courbe et non plus seulement un segment de droite. En effet, les applications sont plus nombreuses, dès lors que l'on veut faire glisser un objet le long d'un fil déformable 1d : pour un collier de perles, ou inversement faire passer un fil au travers d'un modèle déformable pour réaliser par exemple la simulation d'un lacet de chaussure. Ce type de contrainte nous rapproche donc un peu plus de l'objectif de ce travail qui est la simulation des interactions du geste de suture, où en particulier le fil de suture doit pouvoir glisser à l'intérieur d'objets déformables.

Après avoir calculé la projection orthognale du point de contrainte sur la ligne porteuse, nous cherchons la position relative de ce point par rapport aux extrémités du segment. Nous pouvons ainsi savoir lorsque l'objet glisse en dehors des bornes de ce segment. Dans le cas des contraintes de point glissant précédentes, nous avions alors le choix entre créer une contrainte unilatérale à cet endroit afin d'empêcher l'objet de sortir du segment, ou bien de déconnecter l'objet de la courbe en détruisant alors cette contrainte.

Pour les contraintes de point glissant sur une courbe, on détecte lorsque la projection d'un point de contrainte sort des bornes d'un segment, pour rattacher alors la contrainte au segment voisin. Nous gardons en mémoire le segment sur lequel se projetait la contrainte au pas de temps précédent, afin de gérer le cas particulier où la projection du point ne se trouve dans les bornes d'aucun segment. On privilégie alors toujours le rattachement de la contrainte à l'autre segment, afin que le déplacement d'un objet le long de la courbe ne soit pas bloqué par certaines configurations. Nous réprésentons à la figure 3.24 plusieurs étapes clées de la recherche de segment lors du déplacement d'une contrainte de point glissant le long d'une courbe déformable.

Les contraintes de point glissant permettent donc de simuler des objets rigides ou déformables glissant, avec frottement, le long d'un câble, lui même rigide ou déformable. L'exemple d'un collier de perle est montré à la figure 3.25.

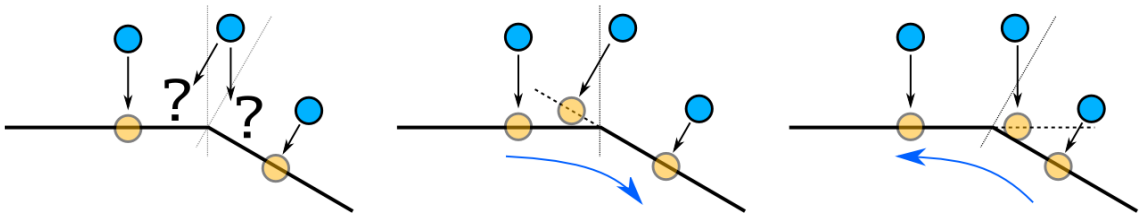


FIGURE 3.24 – Déplacement d'une contrainte glissante le long d'une courbe. (a) On utilise une projection orthogonale pour déterminer le segment sur lequel est fixé un point de contrainte. Lorsque le point projeté dépasse les bornes du segment, on teste le segment voisin. Il arrive qu'un point ne se projette sur aucun segment (la projection est au delà des limites). (b) et (c) Nous gardons en mémoire le segment sur lequel se projetait le point au pas de temps précédent. Ainsi, lorsque ce cas particulier se présente, on choisit de fixer le point de contrainte sur le segment voisin, afin de permettre le glissement le long de la courbe.

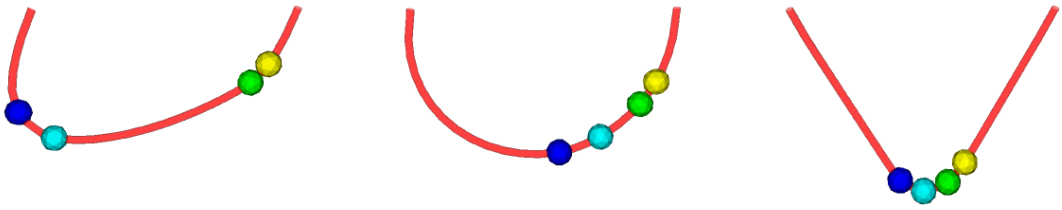


FIGURE 3.25 – Plusieurs objets rigides contraints à glisser le long d'un fil déformable.

3.5.5 Simulations utilisant des contraintes hétérogènes

Le but de notre méthode est de pouvoir faire interagir dans des simulations interactives des objets hétérogènes mais également d'utiliser simultanément des contraintes de types différents afin de représenter des interactions complexes.

Nous présentons dans un premier temps 2 scènes simples. La figure 3.26.a montre l'utilisation de contraintes bilatérales, plus précisement des contraintes de points fixes en 3d, en plus de contraintes de contact. Le cube du haut est fixé, les autres étant attachés l'un à l'autre, tous les objets étant rigides. La détection de collision peut créer des contraintes de contact, comme ici entre les deux cubes du bas.

# de contraintes	15
Mouvement libre	0.6 ms
Détection de collision	0.3 ms
Définition des contraintes	0.08 ms
Résolution	0.02 ms
Correction du mouvement	0.05 ms

TABLE 3.1 – Mesures pour la scène de la figure 3.26.a

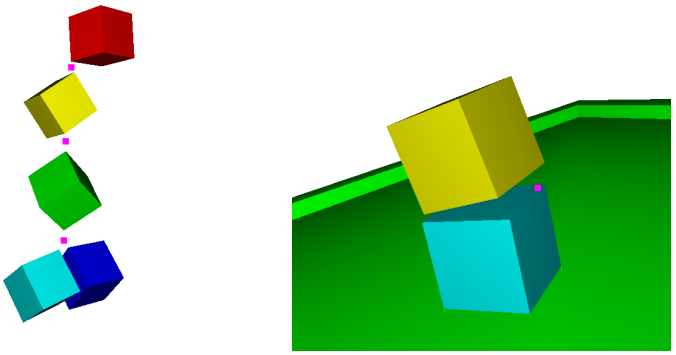


FIGURE 3.26 – Exemples de scènes utilisant simultanément plusieurs types de contraintes. (a) Cubes rigides contraints avec des contraintes bilatérales de points fixes. (b) Collision avec frottement et contrainte de point fixe.

La figure 3.26.b illustre des contraintes de contact avec du frottement. Les deux cubes rigides sont connectés entre eux par une contrainte de point fixe 3d, et tombent librement sur un plan incliné. En variant le coefficient de frottement, nous pouvons avoir adhérence ou glissement. L'utilisateur peut également intéragir avec les objets, et ainsi les empiler comme montré ici.

# de contraintes	27
Mouvement libre	0.2 ms
Détection de collision	0.4 ms
Définition des contraintes	0.09 ms
Résolution	0.9 ms
Correction du mouvement	0.04 ms

TABLE 3.2 – Mesures pour la scène de la figure 3.26.b

Ces 2 scènes sont entièrement compatibles avec le retour d'effort, la résolution mécanique ayant une fréquence de 1kHz.

Enfin, nous présentons une simulation plus complexe, utilisant tous les types de contraintes que nous avons présenté, ainsi que plusieurs types d'objets. La figure 3.27.b présente la configuration de la scène. Nous utilisons des objets plus détaillés, un rigide et un déformable, qui glissent le long d'un câble déformable représenté à l'aide d'un modèle de poutre 1d. Ce câble est fixé à ses extrémités par des contraintes bilatérales de point fixe, et la détection de collision génère automatiquement des contraintes de contact avec frottement entre les deux objets en mouvement.

La simulation présente un taux de rafraichissement interactif, d'approximativement 20 fps. Il faut noter que les calculs ajoutés par notre méthode, à savoir la définition des contraintes, leur résolution et la correction du mouvement sont relativement rapides. En effet, le goulot

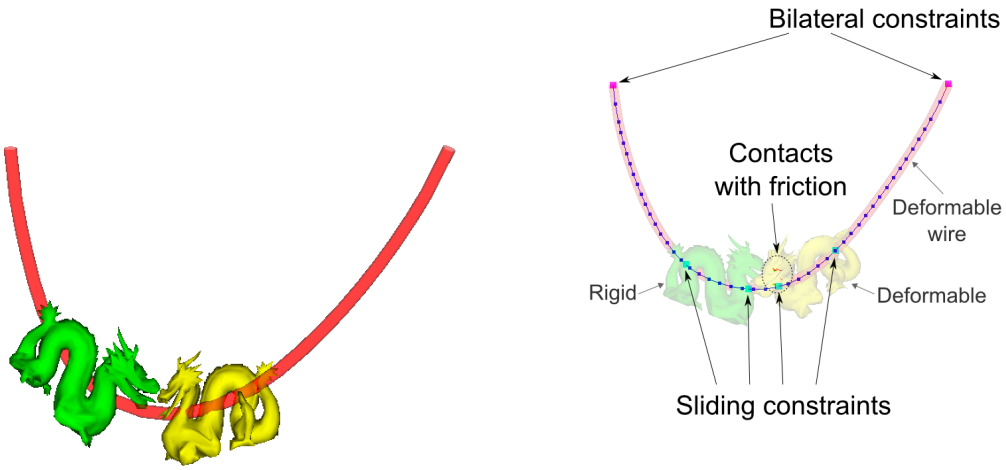


FIGURE 3.27 – Tous les types de contraintes sont utilisés pour un scène complexe comprenant objets rigides et déformables.

d'étranglement de cette scène est la détection de proximité qui est offerte par le framework que nous utilisons.

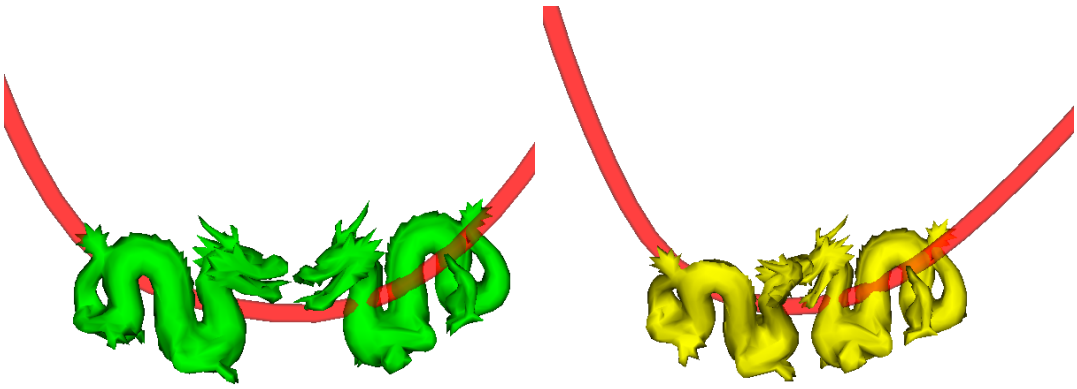


FIGURE 3.28 – La même scène que précédemment, avec (a) 2 objets rigides ; (b) 2 modèles déformables (différence visible principalement sur le contact au niveau des têtes des dragons).

# de contraintes	60
Mouvement libre	26 ms
Détection de collision	41 ms
Définition des contraintes	3 ms
Résolution	0.5 ms
Correction du mouvement	2 ms

TABLE 3.3 – Mesures pour la scène de la figure 3.27

3.6 Conclusion

Nous avons présenté dans ce chapitre une méthode générale pour résoudre des types variés de contraintes dans des simulations interactives. Nous avons montré comment modifier un algorithme de Gauss-Seidel existant afin de séparer son rôle de couplage mécanique du traitement spécifique à chaque contrainte, qui peut alors être externalisé. En reconnaissant les propriétés communes à toutes les contraintes, la méthode présentée reste générique car elle n'a pas besoin d'être modifiée pour chaque nouveau type de contrainte. L'intégration implicite que nous utilisons nous garantit une certaine stabilité et nous permet, avec certaines optimisations du Gauss-Seidel, de réaliser des simulations interactives même pour des interactions complexes. Nous avons enfin montré que de nouvelles contraintes peuvent rapidement être créées et ajoutées à nos simulations. Cette méthode de contraintes généralisées a fait l'objet d'une publication à VRIPHYS'08 : *Unified Processing of Constraints for Interactive Simulation* [GDG08].

Nous allons nous baser sur cette méthode de gestion des contraintes pour travailler à la conception de simulations médicales. Les deux chapitres suivants vont ainsi s'intéresser aux contraintes nécessaires pour représenter les interactions complexes des gestes d'insertion d'aiguilles et de suture.

Chapitre 4

Un modèle d'interaction pour l'insertion d'aiguille

L'insertion profonde d'aiguilles dans le corps humain est une opération délicate, qui demande souvent une grande précision alors que le praticien ne dispose pas d'un retour visuel de la trajectoire. La simulation de ces gestes doit permettre à l'entraînement des chirurgiens ainsi qu'à leur assistance dans un cadre de préparation à l'opération.

Nous voulons montrer que les contraintes de complémentarité que nous avons présentées précedemment sont tout à fait adaptées à une simulation temps réel et précise de l'insertion d'aiguilles. En effet, alors que les travaux précédents (voir section 2.4.1) se sont principalement basés sur des approches en force, nous sommes convaincus qu'une méthode par contraintes pourra reproduire tous les comportements étudiés dans la littérature. Pendant la validation de notre modèle d'interaction, Chentanez *et al.* [CAR+09] ont proposé eux aussi une méthode utilisant des contraintes pour l'insertion d'aiguille, renforçant ainsi notre conviction.

Nous montrons ainsi dans cette partie que l'utilisation de contraintes permet de créer un modèle d'interaction très complet, qui peut reproduire à lui seul tous les types d'interactions présentées dans les travaux antérieurs, là où les modèles présentés dans ceux-ci ne répondaient à chaque fois qu'à certaines problématiques. Nous obtenons une simulation réaliste basée pour les interactions sur les lois physiques et utilisant des modèles déformables complexes qui ont été précédemment validés, tout en restant temps réel pour permettre l'interaction de l'utilisateur dans un contexte de simulation médicale avec retour haptique. De plus, la méthode des contraintes généralisées nous permet de rester assez indépendants des modèles et des formalismes choisis pour simuler les objets en interaction. Nous proposons également certaines fonctionnalités inédites, tel que la gestion du retrait partiel de l'aiguille pour son repositionnement, ainsi que l'insertion de multiples aiguilles simultanément, qui peuvent interagir entre elles directement ou au travers du tissu dans lequel elles sont plongées. Nous avons présenté ce travail à MICCAI'09 sous le nom : *Interactive Simulation of Flexible Needle Insertions Based on Constraint Models* [DGM+09].

Nous séparerons la simulation en plusieurs aspects. Tout d'abord nous aborderons la problématique du contact entre la membrane des tissus et la pointe de l'aiguille, avant et au moment de la pénétration de celle-ci. Nous présenterons alors les contraintes que nous utili-

sons pour gérer les interactions au niveau du bout de l'aiguille, mais également tout le long du corps de celle-ci. La discrétisation des contraintes sera présentée d'un point de vue un peu plus proche de l'implantation. Enfin, nous validerons le modèle complet en reproduisant certaines simulations dont des mesures expérimentales existent dans la littérature.

Modélisation des déformations de l'aiguille et des tissus

Nous désirons mettre au point un modèle d'interaction générique, qui soit le moins dépendant possible des modèles déformables utilisés dans la simulation. En effet, comme nous allons le voir succinctement ici, les travaux récents sur l'insertion d'aiguille ont fait appel à un grand nombre de modèles différents, chacun choisissant alors les formalismes les mieux adaptés à sa méthode d'interaction.

Les premiers travaux sur la simulation d'insertion d'aiguilles ont utilisé des modèles d'aiguilles rigides [DS03a, APT+03a]. Souvent plus épaisses, elles permettent d'arriver facilement à la cible souhaitée mais elles ne sont pas recommandées pour les actes médicaux car étant plus invasives que les aiguilles fines et souples. Ces modèles d'aiguilles rigides ne sont donc valables que pour des distances d'insertion courtes et n'ont été qu'une première approche avant une modélisation plus réaliste. En effet, les mêmes auteurs ont montré que suite à la déformation des tissus, l'aiguille pouvait occasionnellement ne pas atteindre la cible prévue.

Dehghan *et al.* [DGS06] ont étudié les principaux modèles utilisés pour simuler la déformation de l'aiguille. On peut retenir principalement le modèle à élément fini triangulaire utilisé par DiMaio [DS05a] et qui fut étendu à la 3D par Goksel [GSD+05], ainsi que les éléments de poutre linéaires de Glozman et Shoham [GS04]. Les auteurs proposent également un modèle discret composé de masses reliées par des ressorts angulaires, qui bien que très simple présente de très bon résultats lors du test de la poutre encastrée.

La majorité des travaux sur la simulation temps réel d'insertion d'aiguille ont utilisé des modèles éléments finis pour simuler la déformation des tissus. DiMaio et Salcudean [DS03a] emploient des modèles déformables linéaires pour la simulation d'insertion d'aiguilles en 2D. Alterovitz *et al.* [APT+03b] ont également utilisé des modèles de matériaux en élasticité linéaire pour un simulateur de curiethérapie de la prostate en 2D. Goksel *et al.* [GSD+05] simulent cette même opération avec un maillage en 3D et un modèle linéaire pour la prostate. Le modèle élements finis linéaire considère des linéarités géométriques et mécaniques correspondant à des petites déformations et petits déplacements dans le tissu. Cependant, lors de l'insertion d'une aiguille dans un corps mou, on observe couramment des rotations et de grands déplacements, et un modèle géométrique non-linéaire est donc plus adapté.

Nienhuys et van der Stappen [NvdS04] ont proposé l'utilisation d'un modèle hyperalastique néohookéen pour la simulation d'insertion d'aiguille dans un environement plan. Leur hypothèse d'une géométrie linéaire et d'un matériau entièrement compressible (coefficient de Poisson nul) simplifie les calculs mais ne correspond pas aux tissus déformables rencontrés lors d'opérations médicales. En étudiant l'influence de la vitesse d'insertion de l'aiguille sur la déformation des tissus, Crouch *et al.* [CSWO05] montrent que les modèles à éléments finis traditionnellement utilisés pour ces simulations ne peuvent pas bien rendre compte de la relaxation des tissus lorsque l'aiguille ne se déplace plus. Pour simuler de manière réaliste cette phase de relaxation, un modèle viscoélastique serait plus approprié.

Dehghan et Salcudean [DS06] ont étudié la différence entre les modèles MEF linéaires et non-linéaires pour la simulation en 2D. Ils se sont intéressés aux effets des non-linéarités, des conditions limites ainsi qu'aux forces intervenant sur le corps de l'aiguille. Un matériau néohookéen est utilisé pour modéliser à la fois les non-linéarités géométriques et mécaniques. Ils montrent que dans les cas où le tissu est largement contraint à plusieurs de ses côtés, comme c'est souvent le cas dans les expérimentations sur fantômes, il n'est soumis qu'à des petits déplacements et petites rotations. Ces conditions limites ont pour conséquence une très faible différence entre les modèles linéaires et non-linéaires. Dans le cas où le tissu peut tourner sur lui-même, ce qui est observé en particulier pour la curiethérapie de la prostate, un modèle géométrique non-linéaire sera nécessaire. Il faut retenir de leur étude que les effets non-linéaires de la géométrie sont plus important que ceux de la mécanique, et que les conditions limites sont décisives sur la géométrie du problème et doivent être prises en compte pour une simulation réaliste.

Marchal *et al.* [MPT06] proposent un modèle déformable discret, sans maillage et basé sur un principe de mémoire de forme pour la simulation de la curiethérapie de la prostate. Leur modèle permet d'assembler plusieurs composants aux propriétés physiques différentes, et ils simulent ainsi la prostate dans son environnement, en collision avec la vessie et le rectum. Le modèle déformable préserve le volume des organes, et a été validé [MPT05] par une comparaison avec les données du Truth Cube. On peut également noter l'utilisation par Vidal *et al.* [VJHG08] d'un modèle masse-ressort pour un simulateur haptique de radiologie interventionnelle.

4.1 Contraintes à la pointe de l'aiguille

Nous proposons de modéliser l'interaction entre l'aiguille et les tissus en utilisant des contraintes de complémentarité basées sur la méthode présentée au chapitre précédent. Nous avons créé trois contraintes qui gèrent respectivement la pénétration de l'aiguille à travers la membrane de l'organe, la découpe des tissus par la pointe de l'aiguille, et le frottement tout le long du corps de l'aiguille. Pour décrire les différentes contraintes, nous reprendrons la typologie établie par Simone [SO02].

Nous pensons que les approches en forces des travaux précédents présentent plusieurs limitations qui rendent leur utilisation inadaptée dans le cadre de simulations médicales interactives. En effet, les profils de force utilisés pour la réponse à l'insertion d'aiguille sont le plus souvent extraits de mesures expérimentales, même si ensuite des auteurs ont proposé des justifications aux forces mesurées. Ainsi, une paramétrisation pour un de ces modèles d'interaction n'est valide que pour une combinaison prédéterminée de vitesse d'insertion, matériaux des tissus rencontrés, et forme de l'aiguille. Enfin, dans ces méthodes, le déplacement latéral de l'aiguille engendre rarement de déformations du tissu.

Par l'utilisation de contraintes, nous voulons passer outre ces limitations, pour que les forces d'interactions soient calculées selon des lois physiques suivant les paramètres prédéterminés (matériaux, diamètre de l'aiguille) ou observés durant la simulation (compression des tissus, trajectoire et vitesse d'insertion). Les contraintes permettront un couplage complet entre le corps de l'aiguille et les tissus qu'elle traversera, afin que la déformation de l'un se répercute sur l'autre. De plus, nous désirons atteindre une simulation interactive, où

l'insertion de l'aiguille est contrôlée par l'utilisateur, qui n'est aucunement restreint dans les manipulations qui lui sont possibles.

Le modèle d'interaction que nous proposons ne nécessite pas de remailler le tissu, et est le plus indépendant possible du modèle déformable choisi pour le tissu. Nous gérons à la fois les aiguilles rigides et flexibles, et prenons en compte la déflexion causée par les pointes biseautées. Nous utilisons un modèle de frottement complexe, et le calcul des forces d'interaction se base sur des paramètres physiques que l'on trouve communément dans la littérature. Nous permettont à l'utilisateur d'insérer plusieurs aiguilles dans le même tissu, et à ces aiguilles d'interagir entre elles. Les aguilles peuvent être retirées entièrement ou partiellement puis être insérées à nouveau suivant un autre chemin. Enfin, nous voulons atteindre des temps de calcul permettant la simulation temps réel et le retour haptique (sans vouloir dire une actualisation de l'ensemble de la scène à 1000 Hz).

Bien que nous désirions séparer la partie mécanique de notre modèle, nous faisons l'hypothèse que le modèle volumique du tissu peut être ramené à un maillage tétrahédrique pour réaliser nos calculs sur des éléments simples. Cela n'exclu pas l'utilisation de modèles déformables héxahédriques ou à mémoire de forme, car le framework *SOFA* sur lequel nous nous basons offre la possiblité de lier plusieurs représentations à un même modèle. De la même façon, le modèle mécanique utilisé pour l'aiguille permet la simulation d'aiguilles déformables aussi bien que d'aiguilles rigides, mais nous considérerons pour nos contraintes uniquement une série de segments rigides qui forment la fibre moyenne de l'aiguille. Nous appelerons axe de l'aiguille la direction formée par le premier segment à la pointe de l'aiguille.

La première contrainte que nous avons créée pour la simulation de l'insertion d'aiguille prend en compte le contact entre la pointe de l'aiguille et la membrane du tissu jusqu'à l'instant de la pénétration. La collision entre le corps de l'aiguille et les objets environnants sont toujours gérés avec une détection de collision par proximité et une réponse avec des contraintes de contact avec frottement (voir section 3.5.2). Nous ne reviendrons pas sur cet aspect de la collision dans ce chapitre. Pour la pointe de l'aiguille, nous utilisons une détection de collision continue, et gérons cette collision avec une contrainte de complémentarité que nous spécialisons pour prendre en compte l'aspect particulier de la pénétration de l'aiguille.

Une fois que l'aiguille a pénétré les tissus, une autre contrainte est placée à la pointe de l'aiguille, afin de représenter la résistance des organes face à l'insertion de l'aiguille. La force de réponse calculée par cette contrainte correspond à la force de découpe, et dépendra de la forme de la pointe. Nous présenterons aussi l'influence de cette forme sur la déflexion de l'aiguille.

4.1.1 Détection de contact

Nous avons présenté à la section 3.5.2 notre méthode générale pour gérer le contact entre objets déformables. Elle est basée sur une détection du contact par proximité, et une contrainte unilatérale est créée afin d'assurer la non-pénétration. Pour la problématique de l'insertion d'aiguille, la détection par proximité ne convient pas car elle fournit un point de contact situé à l'extérieur de l'organe (voir figure 4.1.a). Nous utilisons donc ponctuellement une détection de collision continue pour le contact entre la pointe de l'aiguille et la membrane entourant les tissus. En considérant une interpolation linéaire du mouvement durant un pas de temps, celle-ci nous permet de calculer l'instant de la collision et donc le point d'impact exact.

Nous montrons sur la figure 4.1 que seule la détection continue fournit le point de contact exact lorsque l'aiguille n'est pas parfaitement perpendiculaire à la surface de l'objet. Nous obtenons par cette méthode une plus grande précision, dont la différence avec les autres méthodes n'est pas négligeable et a été observée facilement lors de tests simples d'insertion d'aiguille.

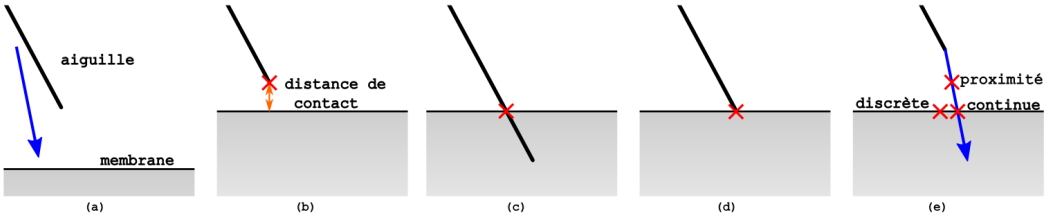


FIGURE 4.1 – Comparaison des techniques de détection de collision pour le contact entre l'aiguille et la membrane de l'objet. (a) Une aiguille s'approche de la membrane de l'organe. (b) La détection par proximité fournit un point de contact à l'extérieur de l'objet. (c) La détection de collision discrète calcule l'intersection entre l'aiguille et la membrane sans prendre en compte les déplacements. (d) La détection continue donne l'instant exact de la collision. (e) Comparaison des points de contact fournis par les différentes méthodes.

Nous ne nous intéressons pour cette section qu'à la collision entre la pointe de l'aiguille et les tissus, la collision avec le reste de l'aiguille étant gérée comme pour les autres objets de la scène. Pour la collision, la membrane de l'organe est représentée par une surface triangulée. Dans les cas les plus simples, cette surface est formée d'un sous-ensemble des triangles du maillage volumique tétrahédrique ou hexahédrique de l'objet. Nous désirons le plus possible éviter de lier notre méthode à un modèle mécanique particulier, et nous permettons alors que la surface triangulée utilisée pour la détection de collision soit séparée de la représentation du modèle volumique. Cela nous permet d'avoir des niveaux de détails différents pour la collision et la mécanique. Nous pouvons ainsi rendre compte de la géométrie complexe des organes pour la collision tout en maintenant un modèle déformable simple, la seule obligation étant que la surface doit entièrement être comprise dans le volume de l'objet. Cette séparation nous a également permis de modéliser des inclusions multiples dans un tissu en gardant un seul modèle volumique. Ces trois cas sont représentés à la figure 4.2.

Pour détecter un éventuel contact entre la pointe de l'aiguille et la surface triangulée de la membrane, nous faisons à chaque pas de temps un test de collision entre chaque triangle et la pointe. Un test des boites englobantes est réalisé au préalable pour filtrer la majorité des triangles et ne considérer qu'une petite partie de la surface. Le calcul de la collision continue est effectué entre la pointe $\mathbf{p0}$ de l'aiguille soumise à un déplacement $d1$ durant la phase de mouvement libre et un triangle $(\mathbf{p1}, \mathbf{p2}, \mathbf{p3})$ dont les déplacements des points sont $(\mathbf{d1}, \mathbf{d2}, \mathbf{d3})$. Nous cherchons l'instant $t \in [0; 1]$ de la collision, le point de contact p s'obtenant immédiatement par $\mathbf{p} = \mathbf{p0} + t.\mathbf{d0}$. Pour cela, on sait qu'à l'instant de la collision, la pointe de l'aiguille se trouve dans le plan du triangle. Cela revient à chercher t tel que le produit scalaire entre la normale du triangle à ce moment précis et un vecteur formé par la pointe de l'aiguille et un sommet du triangle est nul. Dans un premier temps nous calculons la normale du triangle au temps t notée $\mathbf{n}(t)$ via le produit vectoriel des côtés

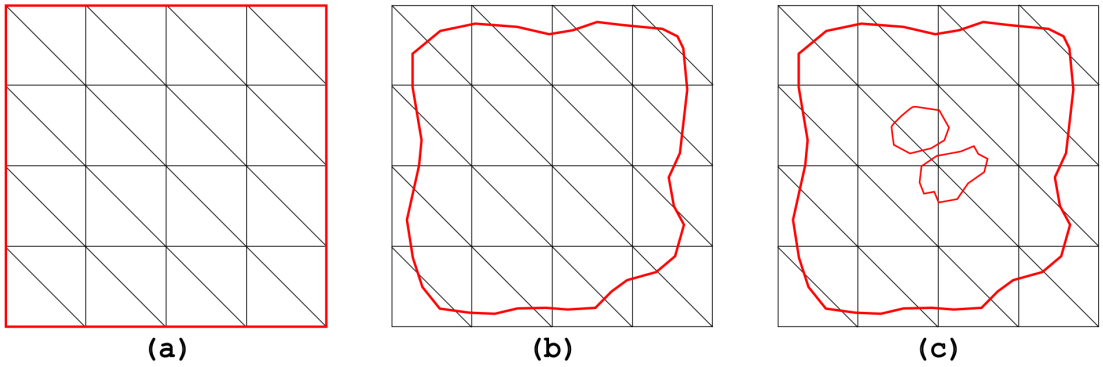


FIGURE 4.2 – Séparation du modèle déformable volumique (en gris) et de la surface triangulée utilisée pour la collision (en rouge). Vue en coupe. (a) La surface de collision peut être calculée à partir de la représentation volumique. (b) Utilisation d'une surface triangulée plus détaillée. (c) Inclusion d'autres surfaces à l'intérieur de l'objet qui vont également pouvoir interagir avec l'aiguille.

du triangle : $\mathbf{n}(t) = [(\mathbf{p2} - \mathbf{p1}) + t(\mathbf{d2} - \mathbf{d1})] \times [(\mathbf{p3} - \mathbf{p1}) + t(\mathbf{d3} - \mathbf{d1})]$, nous obtenons une équation quadratique sous la forme d'une matrice. On réalise alors le produit scalaire $\mathbf{n}(t) \cdot [(\mathbf{p1} - \mathbf{p0}) + t(\mathbf{d1} - \mathbf{d0})]$, dont on pose le résultat comme étant nul, ce qui nous donne une équation cubique à résoudre. On utilise la méthode de Tartaglia-Cardan pour déterminer les racines de l'équation, puis on sélectionne la plus petite qui est comprise entre 0 et 1. Il ne reste plus qu'à vérifier que le point de contact est bien compris dans les limites du triangle. La détection de collision continue peut être sujette à des erreurs numériques, et nous introduisons un petit ϵ dans les calculs pour ne pas passer à côté d'un point de contact dans certains cas de figure (collision sur les bords d'un triangle, ou vitesse relative importante).

4.1.2 Contact avec frottement

Une fois que le contact entre la pointe de l'aiguille et la membrane est établi, nous créons une contrainte de complémentarité au point d'impact pour gérer leur interaction. Nous utilisons une contrainte unilatérale du même type que celle employée plus généralement pour répondre au contact entre objets et que nous avons présenté à la section 3.5.2. La direction de la contrainte est la normale du triangle à l'instant précis du contact, et la violation de la contrainte est calculée comme la distance parcourue par la pointe de l'aiguille entre le moment de l'impact et la fin de la phase de mouvement libre. Il faut noter que la contrainte n'est pas gardée d'un pas de temps à l'autre, mais est recréée à chaque fois qu'une collision est détectée car elle liée avec un triangle de la surface en particulier. Ainsi, si l'aiguille s'éloigne de la surface sans avoir pénétré, aucune contrainte ne limite son déplacement.

Comme nous l'avons présenté à la section 2.1.2, plusieurs travaux ont modélisé la force de réaction du tissu à la poussée de l'aiguille par un ressort non-linéaire ou une fonction exponentielle par rapport à la distance parcourue par l'aiguille. Nous décidons de suivre une approche différente, dans le sens où nous considérons que cette force n'est autre que la réponse calculée par la contrainte qui empêche l'aiguille de pénétrer dans le tissu. La forme de la force mesurée à la base de l'aiguille avant la pénétration n'est donc pas un paramètre de notre

modèle d'interaction, car elle est uniquement la conséquence du choix du modèle déformable de l'organe et de ses paramètres physiques, choix que nous laissons à l'utilisateur. L'utilisation d'un modèle déformable avec un matériau linéaire donnera ainsi une réponse bien différente d'un modèle non linéaire ou encore non compressible. Les paramètres physiques utilisés (tel que le module de Young) pour modéliser les déformations du tissus et de l'aiguille auront également un rôle à jouer dans le comportement observé.

Les aiguilles employées dans les actes chirurgicaux ont en général une pointe permettant une découpe efficace des tissus, ce qui a également pour conséquence qu'elles ne glissent pas ou très peu à la surface de la peau ou des organes. Cependant cela peut arriver à la surface des inclusions plus dures (typiquement les os mais aussi d'autres tissus), et notre modèle peut gérer ces cas en permettant à l'aiguille de se déplacer latéralement, mais en appliquant une force de frottement selon le modèle de Coulomb. Nous modélisons ce frottement dans la même contrainte de contact, selon la méthode présentée à la section 3.5.3. Nous fournissons donc à la contrainte deux directions tangentes au plan du triangle au moment du contact ainsi que la violation correspondant au déplacement latéral effectué par la pointe jusqu'à la fin du mouvement libre. Nous obtenons donc une contrainte selon trois directions, qui empêche l'interpénétration de l'aiguille dans le tissu et utilise la force de réaction normale calculée pour mettre à jour une éventuelle force de frottement. Les forces de réaction de la contrainte sont ensuite appliquées directement à la pointe de l'aiguille, et sont réparties sur les sommets du triangle selon les coordonnées barycentriques du point de contact, en suivant la méthode décrite à la section 3.3.1.

4.1.3 Contrainte de pénétration

Puisque nous voulons simuler l'acte d'insertion d'aiguille dans un organe, nous devons à un moment donné permettre à la pointe de l'aiguille de pénétrer à travers la membrane. Ceci est la principale différence de cette contrainte par rapport à la réponse au contact que nous utilisons généralement. Ainsi, durant la résolution de la contrainte de contact, nous observons la valeur de la force de réaction projetée suivant l'axe de l'aiguille. Lorsque celle-ci dépasse un certain seuil, la contrainte change d'état et la force de réaction est plafonnée afin de laisser l'aiguille pénétrer. On voit ici l'intérêt d'employer une contrainte de complémentarité pour modéliser cette interaction, car nous pouvons rendre compte de trois états différents avec un seul ensemble d'inégalités, comme le montre la figure 4.3. On note Q la pointe de l'aiguille, P le point de contact ou de pénétration sur la surface de la membrane et $\vec{n}$ est la normale de la surface au point P. On peut observer la forme caractéristique en escalier de la loi de cette contrainte, due au fait que la force de réaction de la contrainte n'est pas autorisée à dépasser le seuil de pénétration f_p.

La valeur de la force de pénétration f_p peut être facilement mesurée expérimentalement, c'est le pic observé dans la force d'insertion avant une brusque baisse qui correspond alors à la pénétration. Cette valeur dépend autant de la rigidité de la membrane que de la forme de la pointe de l'aiguille, et est paramétrée pour chaque tissu que l'aiguille doit traverser. La contrainte de pénétration peut être appliquée à plusieurs reprises durant la simulation de l'insertion d'aiguille pour chaque couche de tissu que celle-ci rencontre, avec plusieurs valeurs de f_p pour modéliser les différents comportements mécaniques de ces tissus. Si l'aiguille rencontre un os ou un autre objet très dur, on fixe f_p à une valeur très élevée de telle façon

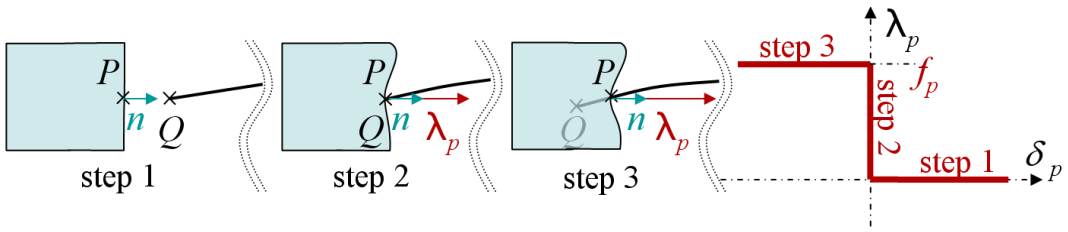


FIGURE 4.3 – **Les étapes de la pénétration** *Durant l'étape 1, Q s'approche de la surface mais l'écart avec P est positif ($\delta_p \geq 0$) et la force d'interaction doit être nulle ($\lambda_p = 0$). Durant l'étape 2, Q touche la surface du tissu sans la percer. L'écart entre P et Q est null ($\delta_p = 0$) et la force d'interaction est nécessairement positive dans la direction de la normale à la surface ($\lambda_p \geq 0$). La valeur de cette force est strictement inférieure à un seuil de force de pénétration $\lambda_p \leq f_p$. Lors de l'étape 3, la pointe de l'aiguille entre dans le tissu, l'écart est négatif ($\delta_p \leq 0$) et la force de contrainte est égale au seuil ($\lambda_p = f_p$).*

qu'elle ne soit jamais atteinte par λ_p. L'aiguille ne peut alors pas percer ces régions, et glisse le long de leur surface.

Avant la pénétration, le mouvement latéral de l'aiguille est contraint par le frottement dans les directions tangentes à la surface de la membrane. Lorsque la force de pénétration est atteinte, l'aiguille peut pénétrer dans l'organe. Le déplacement dans les directions latérales n'est plus possible car le corps de l'aiguille doit obligatoirement passer par le point de pénétration. Cela revient à contraindre le mouvement de l'aiguille à être strictement dans la direction de son axe, et donc à contraindre le mouvement dans les directions normales au corps de l'aiguille à être nul. Nous pourrions créer deux contraintes bilatérales selon ces directions normales, mais elles ne doivent être actives que lorsque l'aiguille pénètre, ce que seule la contrainte de pénétration peut savoir. Nous ne voulons pas non plus attendre le pas de temps suivant la pénétration pour contraindre correctement le déplacement latéral de l'aiguille, auquel cas l'aiguille pénètrerait un cours instant selon la normale de la surface.

Nous ajoutons donc la gestion de ce comportement à la contrainte de pénétration, en changeant de repère durant la résolution si l'aiguille est amenée à pénétrer la membrane (voir figure 4.4). En pratique, nous calculons la violation de la contrainte dans l'espace global des mouvements, et nous lui fournissons deux repères, à savoir celui formé par la normale et deux directions tangentes à la surface, ainsi qu'un autre formé par la direction de l'aiguille et deux directions normales à son corps. En pratique, on résoud tout d'abord le contact frottant, les directions du frottement étant données par le premier repère, et on calcule à l'aide du second repère la projection de la force de réponse sur l'axe de l'aiguille. Si cette force est supérieure à la force de pénétration, l'aiguille franchit la membrane ; on utilise alors le second repère pour limiter les déplacements latéraux.

Une fois que l'aiguille a pénétré le tissu, la contrainte de pénétration n'a plus de raison d'être et sera donc supprimée au pas de temps suivant. On garde cependant en mémoire l'emplacement du point de pénétration, sous la forme de coordonnées barycentriques des sommets du triangle de la surface avec lequel l'aiguille est entrée en contact. Ces données seront utilisées pour une contrainte d'interaction interne maintenant l'aiguille sur son chemin et gérant le frottement, que l'on présentera en détails à la section 4.2.

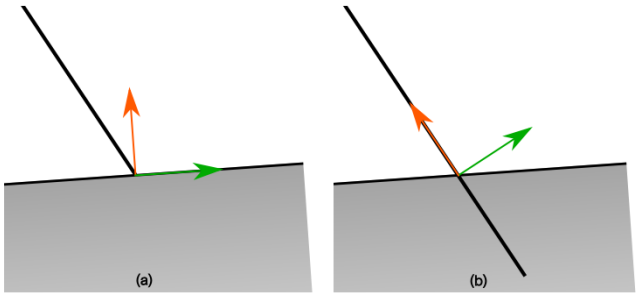


FIGURE 4.4 – Changement de repère lors de la pénétration de l'aiguille. (a) La contrainte calcule un frottement selon les directions tangentes à la surface de l'organe. La force de pénétration est calculée cependant selon l'axe de l'aiguille. (b) Une fois que l'aiguille est à l'intérieur du tissu, la contrainte empêche tout déplacement selon les directions normales au corps de l'aiguille.

L'aiguille ayant pénétré la membrane, nous devons maintenant tenir compte de la force de découpe intervenant au niveau de la pointe de l'aiguille. Cette force correspond à la résistance des tissus constitutifs de l'organe à l'avancée de l'aiguille, que sa pointe doit couper progressivement pour pouvoir avancer. Une nouvelle contrainte de complémentarité sera donc placée à la pointe de l'aiguille afin de simuler cette interaction.

Le choix de la forme de la pointe de l'aiguille, symétrique ou biseautée, a une influence directe sur une éventuelle déflexion de l'aiguille au cours de son insertion dans les tissus. Nous prendrons également en compte ce comportement via une contrainte agissant sur les déplacements dans les directions normales à l'axe de l'aiguille.

4.1.4 Contrainte de découpe

Lors de l'insertion d'une aiguille, les mesures expérimentales [SO02, OSO04] ont montré l'existence d'une force au niveau de la pointe de l'aiguille. Celle-ci peut être considérée comme constante pour une vitesse d'insertion stable, et n'est observée que lors de l'avancée de l'aiguille dans les tissus. C'est la force nécessaire exercée par la pointe de l'aiguille pour qu'elle découpe les tissus et puisse progresser à l'intérieur de l'organe. Si la force appliquée par l'utilisateur n'atteint pas cette valeur, l'aiguille ne fait que pousser les tissus sans les découper, ils peuvent alors se déformer mais l'aiguille n'est pas amenée à avancer. Lorsque l'aiguille s'arrête ou est retirée, aucune force n'est mesurée au niveau de la pointe.

Nous proposons de modéliser ce comportement par l'utilisation d'une contrainte de complémentarité car nous pouvons décrire la force de découpe f_c via un ensemble d'inégalités. A chaque pas de temps, après le mouvement libre, nous calculons la distance de pénétration de la pointe de l'aiguille, ramenée à la configuration au repos non déformée du tissu. La violation de la contrainte correspond à l'écart entre cette mesure et la distance de pénétration maximale enregistrée au cours de la simulation. Celle-ci est mise à jour après résolution des contraintes à la fin du pas de temps de simulation. Cette contrainte reprend la forme de la loi utilisée pour la contrainte de pénétration, mais en utilisant la valeur de la force de découpe, comme illustré par la figure 4.5.

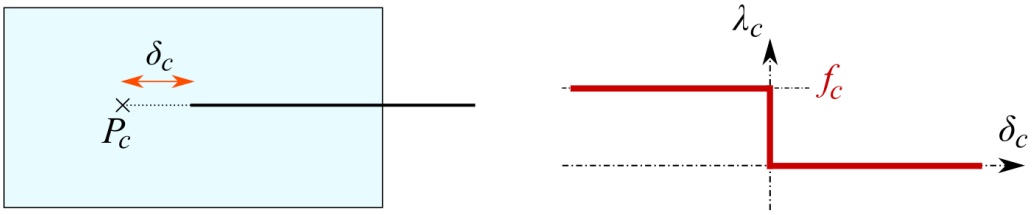


FIGURE 4.5 – Contrainte de découpe : P_c est le point le plus avancé atteint par l'aiguille lors de son insertion, δ_c étant l'écart avec la position actuelle de la pointe de l'aiguille. La loi de la contrainte impose une force égale à f_c si l'aiguille avance dans les tissus ou nulle si elle recule. Tant que la force imposée à l'aiguille n'atteint pas f_c, celle-ci reste immobile.

Cette valeur de la force de découpe peut également être extraite des courbes de mesures expérimentales lors de l'insertion d'une aiguille. En effet, une fois que l'aiguille a pénétré le tissu, on observe une diminution subite de la force mesurée à la base de l'aiguille, mais qui n'est pas nulle. A ce moment, la distance de pénétration à l'intérieur des tissus est très petite, et on peut considérer que la force de frottement qui dépend de l'aire de l'aiguille en contact avec les tissus n'est pas significative. La force de découpe est alors la seule force mesurée qui s'oppose au mouvement de l'aiguille, et on lit sa valeur directement sur la courbe. Il est également possible de réaliser l'insertion d'une aiguille, son retrait partiel puis sa réinsertion suivant le même chemin : lors de la seconde insertion, les tissus auront déjà été découpés et la force de découpe sera nulle ; sa valeur sera alors obtenue en faisant la différence entre les valeurs des deux insertions. La figure 4.6 illustre la lecture des paramètres de notre modèle sur des mesures expérimentales.

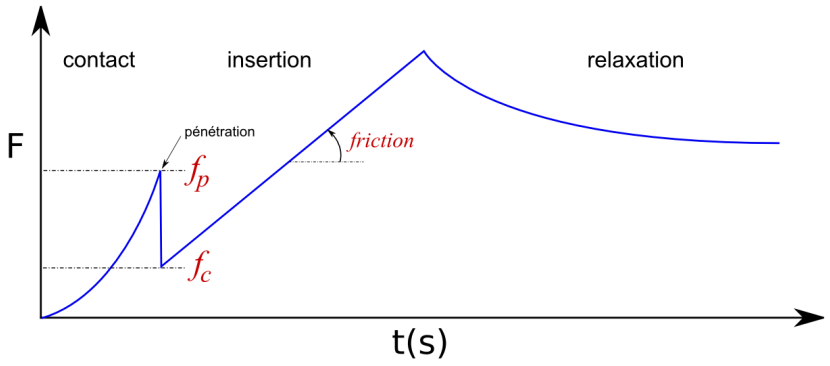


FIGURE 4.6 – Lecture des paramètres de notre modèle à partir de la force mesurée à la base de l'aiguille lors d'une expérience d'insertion.

Contrairement à de nombreux travaux antérieurs sur la simulation d'insertion d'aiguille, nous n'imposons pas une insertion continue à vitesse constante. L'aiguille, manipulée par l'utilisateur, peut être partiellement retirée puis réintroduite. Nous prenons en compte ce cas de figure avec le modèle de contrainte présenté, car aucune force de découpe n'est appliquée tant que la pointe de l'aiguille n'est pas au point le plus profond atteint lors d'une insertion précédente.

4.1.5 Incidence de la forme de l'aiguille

Nous l'avons vu à la section 2.4.2, une aiguille déformable peut se courber au fur et à mesure de son insertion si sa pointe n'est pas de forme symétrique. Nous simulons l'influence de la forme de l'aiguille par une autre contrainte au niveau de sa pointe, et qui agit sur les déplacements dans les directions normales à son axe.

Lorsque la pointe de l'aiguille est symétrique, nous voulons que l'aiguille avance toujours dans la direction de son axe. Ainsi, une aiguille droite aura une trajectoire rectiligne du moment que la déformation des tissus environnants reste homogène. Contraindre la pointe de l'aiguille à avancer dans la direction de son premier segment revient en fait à contraindre ce segment à passer par l'ancienne position de la pointe. Cela revient à créer une contrainte de point glissant (présentée à la section 3.5.4) entre l'axe de l'aiguille et un point dont la position est sans cesse mise à jour. Au début de chaque pas de temps nous calculons les coordonnées barycentriques de la pointe de l'aiguille par rapport aux sommets du tétraèdre qui l'englobe. Après le mouvement libre, on s'en sert alors pour trouver la nouvelle position de ce point dans la configuration déformée. Une contrainte bilatérale est alors créée pour amener le premier segment de l'aiguille sur cette position. La figure 4.7 illustre ces différentes étapes.

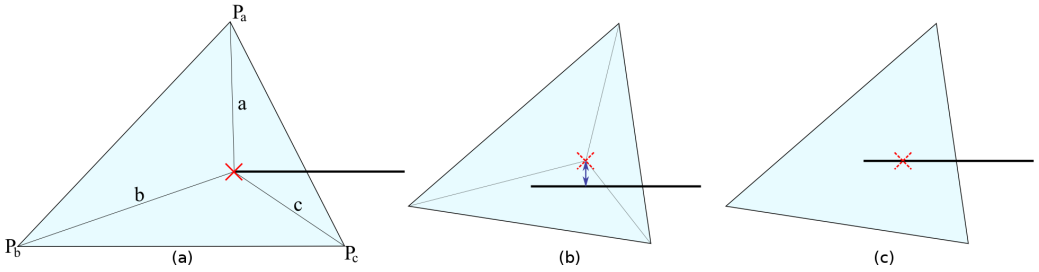


FIGURE 4.7 – Contrainte de la pointe : (a) configurations de l'aiguille et de l'élément dans lequel se trouve la pointe avant le mouvement libre. (b) Après le mouvement libre, l'ancienne position de la pointe est ramenée dans la nouvelle configuration de l'élément. (c) L'aiguille est contrainte à passer par ce point.

La forme biseautée de certaines aiguilles entraîne une pression asymétrique des tissus sur la pointe et une déflexion de l'aiguille lors de son insertion. Nous proposons de modéliser ce comportement en procédant à une légère rotation du repère formé par les direction de la contrainte du bout de l'aiguille. Ainsi, plutôt que d'obliger l'aiguille à avancer selon une ligne droite, à chaque pas de temps nous introduisons un léger décalage, qui progressivement fait prendre à l'aiguille un chemin courbe. Le reste de l'aiguille est ensuite contraint à rester dans le chemin créé par la pointe et on peut donc observer cette déflexion par la configuration que prend l'aiguille lorsqu'elle est insérée sur une distance assez importante. Lorsque l'aiguille recule, on laisse le repère aligné avec l'axe de l'aiguille car cela induirait des déformations non réalistes des tissus. La figure 4.8 présente les configurations possibles lors de l'insertion puis du retrait d'une aiguille dans un organe.

La rotation de la contrainte n'est pas mise à jour durant sa résolution mais à chaque pas de temps après avoir comparé la distance d'insertion par rapport à celle mesurée précédemment.

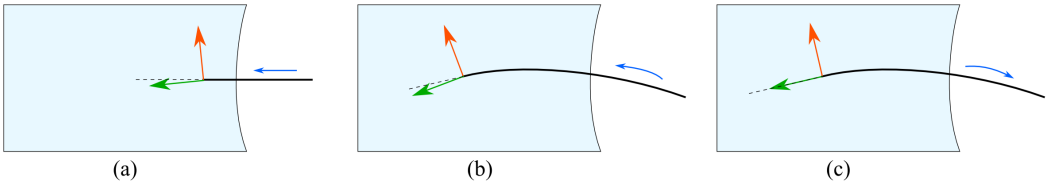


FIGURE 4.8 – Simulation des pointes biseautées par la rotation du repère de la contrainte. (a) Une petite déviation de la direction d'insertion est introduite et mise à jour à chaque pas de temps. (b) On remarque la déflexion de l'aiguille lors de son insertion. (c) Si l'aiguille recule, la contrainte n'est pas tournée.

On choisit donc si on est dans le cas de l'insertion ou du retrait de l'aiguille selon la valeur donnée après l'étape du mouvement libre. Nous pourrions, comme pour la contrainte de pénétration, fournir deux repères à la contrainte qui passerait alors de l'un à l'autre durant la résolution selon les cas. Cependant, l'erreur dans le déplacement ne concerne que le retrait de l'aiguille et n'est pas problématique car celle-ci est toujours maintenue par d'autres contraintes tout le long de son corps. Nous allons maintenant présenter ces contraintes réparties sur tout le chemin créé par l'aiguille.

4.2 Contraintes sur le corps de l'aiguille

Nous venons de présenter deux contraintes s'appliquant à la pointe de l'aiguille, l'une gérant l'interaction avec la membrane avant la pénétration, l'autre s'occupant de la résistance à la découpe et de la déflexion de l'aiguille. Nous créons également des contraintes sur la courbe de l'aiguille, pour tenir compte des interactions entre le corps de l'aiguille et les tissus. Nous retiendrons principalement deux interactions, à savoir le frottement engendré par la pression des tissus alentours et qui s'oppose au mouvement de l'aiguille, ainsi que les forces latérales qui empêchent l'aiguille de quitter le chemin qu'elle a découpé dans les tissus. Au niveau géométrique nous ne considérons que la fibre moyenne de l'aiguille, sous la forme de segments linéaires, pour plus de simplicité. Nous y appliquerons directement les forces calculées par les contraintes, mais tiendrons compte du diamètre de l'aiguille lors du calcul du frottement. Nous appelons ces contraintes les *contraintes d'interaction internes* car elles sont réparties à l'intérieur des tissus, par opposition à la contrainte de la pointe de l'aiguille.

4.2.1 Maintenir l'aiguille dans son chemin

Le rôle premier des contraintes d'interaction internes est de maintenir l'aiguille sur le chemin sur lequel elle évolue. Contrairement aux premiers travaux sur la simulation d'insertion d'aiguille, nous permettons à l'aiguille d'être insérée selon des trajectoires quelconques. De plus, l'aiguille peut se déformer par l'action de l'utilisateur ou via la pression des tissus. Si nous ne contraignons que le point d'entrée et la pointe de l'aiguille, le modèle déformable de l'aiguille aurait vite fait de la ramener dans une configuration non déformée. Nous devons donc agir contre les modèles déformables, en posant des contraintes empêchant tout mouvement dans les directions normales à l'axe de l'aiguille.

Pour ce faire nous générons à intervalles réguliers, au fur et à mesure qu'on insère l'aiguille, un ensemble de contraintes posées sur sa trajectoire. Nous décrivons plus en détails la méthode utilisée pour la discrétisation à la section 4.3. Les contraintes d'interaction internes sont liées à un élément du maillage de l'objet (le plus souvent un tétraèdre), leurs positions à l'intérieur de celui-ci étant repérées par ses coordonnées barycentriques. Une fois qu'elles sont créées, leurs positions ne changent pas jusqu'à ce qu'elles soient supprimées lorsque l'aiguille est retirée.

Chaque contrainte d'interaction interne a pour rôle de maintenir le segment de l'aiguille le plus proche d'elle à passer par sa position, c'est une contrainte de point glissant. Après la phase de mouvement libre, la position de la contrainte est mise à jour en fonction de la nouvelle configuration de l'élément dans lequel elle est localisée. Nous recherchons ensuite le segment de l'aiguille qui est le plus proche de cette position, comme indiqué à la section 3.5.4. Des contraintes bilatérales sont ensuite créées dans les directions normales au segment concerné pour l'amener à la position de la contrainte. Nous représentons sur la figure 4.9 la violation des contraintes qui doit être mise à jour à chaque pas de temps.

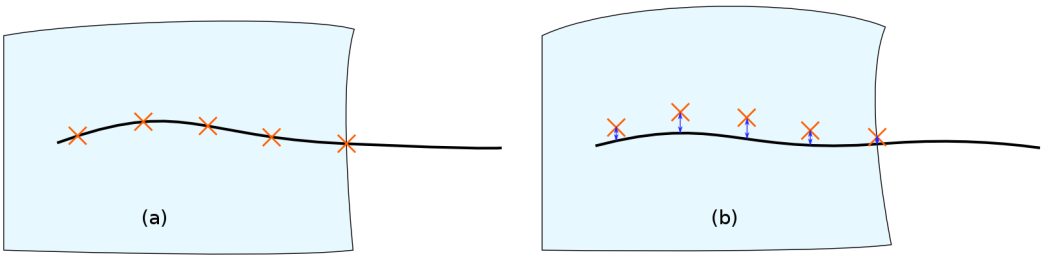


FIGURE 4.9 – Contraintes sur le chemin de l'aiguille. (a) Avant le mouvement libre, les contraintes sont sur la courbe créée par l'aiguille. (b) Après le mouvement libre et suite aux déformations des tissus, la position des contraintes a changé. Des contraintes bilatérales vont rassembler les contraintes et les segments de l'aiguille correspondants.

Durant la résolution des contraintes, le Gauss-Seidel s'efforce de chercher l'équilibre entre les forces appliquées par les contraintes et les forces de réaction des modèles déformables de l'aiguille et des tissus. Ces contraintes d'interaction internes créent donc une relation bidirectionnelle entre les tissus et l'aiguille. Ainsi, si l'utilisateur impose un déplacement latéral à la base de l'aiguille, les tissus vont se déformer suite à la pression de l'aiguille. De la même façon, si pour une raison les tissus se déforment l'aiguille va être en partie entraînée tout en offrant une certaine résistance selon sa rigidité. On peut observer alors que l'insertion d'une aiguille assez rigide dans un tissu déformable a tendance à le rigidifier localement, parce que la raideur de l'aiguille qui est maintenant couplée aux tissus s'ajoute à la raideur de ceux-ci.

4.2.2 Frottement dans un matériau homogène

Les contraintes d'interaction internes gèrent également le frottement qui s'applique au corps de l'aiguille et qui freine son mouvement. Lorsque l'aiguille pénètre les tissus, elle les repousse localement dans toutes les directions normales, sur une distance correspondant à son rayon. Les tissus résistent à ce deplacement, et appliquent en réaction un pincement sur toute la surface du corps de l'aiguille. Cette pression crée une force de frottement lorsque l'aiguille

est insérée ou retirée, dont la valeur est proportionnelle à la distance de pénétration (voir figure 4.10).

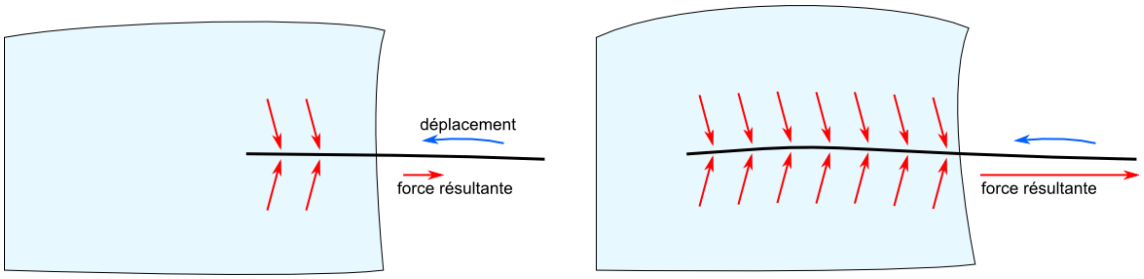


FIGURE 4.10 – Influence de la distance de pénétration sur le frottement ressenti par l'utilisateur lors de l'insertion d'une aiguille. La force de frottement, résultante du pincement exercé par le tissu tout le long du corps de l'aiguille, est proportionnelle à la surface de l'aiguille sur laquelle peut s'appliquer ce pincement.

Nous répartissons ce phénomène de frottement sur l'ensemble des contraintes d'interaction internes posées dans les tissus. Chaque contrainte gère donc en plus le frottement dans la direction tangente au segment de l'aiguille correspondant. Il peut prendre deux états : frottement statique lorsqu'il n'y a pas de mouvement relatif $\delta_f = 0$ entre les deux objets qui sont alors en adhérence, ou frottement dynamique lorsque l'aiguille glisse à l'intérieur des tissus $\delta_f \neq 0$. Pour passer en frottement dynamique, la force d'interaction au niveau de la contrainte doit être supérieure à la valeur seuil $\mu.p$, avec p la pression exercée par les tissus et μ le coefficient de frottement. Dans un premier temps nous estimons la pression à partir de la raideur des tissus, et nous considérons que sa valeur est homogène quelque soit la position dans l'organe et reste constante durant la simulation. La résistance du frottement est donnée par le graphe de la figure 4.11. Nous utilisons un modèle de frottement de Coulomb auquel nous avons ajouté un coefficient d'amortissement α pour la frottement dynamique, d'une façon similaire au modèle de frottement de Karnopp [Kar85].

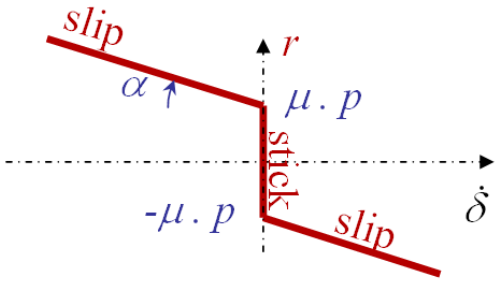


FIGURE 4.11 – Graphe de la loi de frottement utilisée pour contraindre le mouvement de l'aiguille dans les tissus.

Nous devons alors intégrer r sur la longueur de l'aiguille qui a été insérée. Le calcul de la force de frottement est réparti sur les contraintes d'interaction internes distribuées le long de l'aiguille. À chaque contrainte est attribué un segment de la courbe de l'aiguille, jusqu'à

la contrainte suivante dans la direction de la pointe. Elle calcule alors la force qui lui sera appliquée : $\lambda = l\pi d.r$ avec l la longueur du segment et d le diamètre de l'aiguille. La longueur l est mesurée dans la configuration au repos et non déformée des tissus, et reste donc constante entre deux contraintes d'interaction internes pour la durée de la simulation. Dans le cas d'aiguilles flexibles, la longueur de la courbe de l'aiguille entre deux contraintes peut changer si celle ci se déforme. Il est envisageable de mettre à jour cette longueur pour prendre en compte ce cas, mais en pratique la différence est négligeable si on utilise pour discrétiser le chemin de l'aiguille un nombre suffisant de contraintes.

Lorsque l'aiguille est insérée dans les tissus, les contraintes d'interaction internes sont créées au fur et à mesure de sa progression, et placées juste après la pointe de l'aiguille. Si on intègre naïvement le frottement sur la longueur du segment appartenant à chaque contrainte, l'insertion d'une nouvelle contrainte provoque un brusque changement dans la valeur du frottement pour la seconde contrainte après la pointe (voir figure 4.12.a). On observe alors des saccades dans le mouvement de l'aiguille lors de son insertion, à cause du modèle déformable des tissus réagissant à une baisse locale du frottement. Nous gérons donc différement l'intégration du frottement sur le segment entre la pointe de l'aiguille et la contrainte suivante, afin que le frottement appliqué à une nouvelle contrainte soit nul lors de sa création. En pratique on attribue une force de frottement à la pointe de l'aiguille qui correspondrait à la distance minimale entre deux contraintes (voir la discrétisation des contraintes à la section 4.3). Ensuite, la seconde contrainte calcule la force de frottement pour la distance entre elle et la pointe de l'aiguille, puis soustrait la valeur calculée pour la pointe de l'aiguille. L'instant immédiatement après la pénétration de l'aiguille est gérée indépendamment jusqu'à ce que la distance entre le point d'insertion et la pointe de l'aiguille soit significative. Nous évitons donc au maximum les sursauts du tissu lors de l'insertion ou la suppression d'une contrainte. La figure 4.12.b illustre cette méthode.

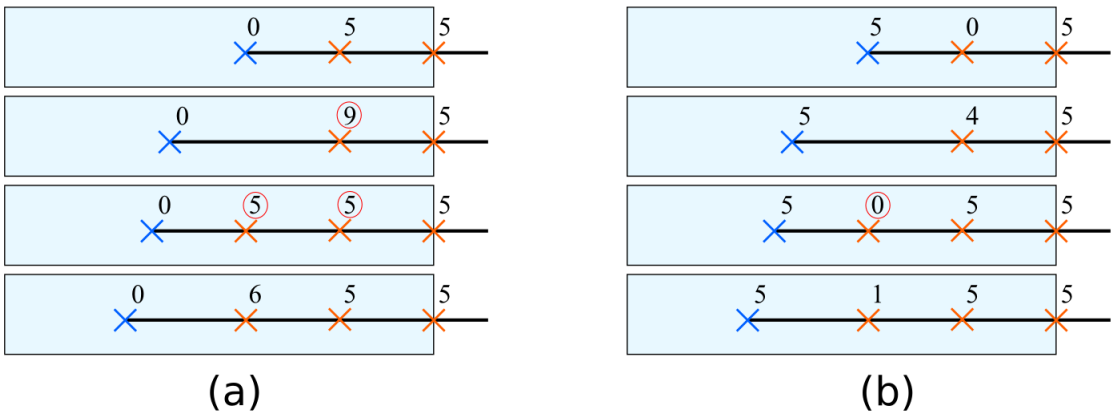


FIGURE 4.12 – Répartition du frottement sur les contraintes. (a) Si chaque contrainte calcule la longueur du segment qui lui est attribué, l'ajout d'une contrainte entraîne un changement brusque dans certaines valeurs. (b) La répartition sur le segment au niveau de la pointe de l'aiguille est modifiée pour que l'ajout et la suppression de contraintes se fasse sans modification locale du frottement.

4.2.3 Prise en compte de la contrainte dans le matériau

Dans la section précédente, nous avons considéré que le pincement exercé par les tissus lors de l'insertion d'une aiguille est constant et homogène. Cela permet de garder un modèle d'interaction avec un nombre restreint de paramètres et de le séparer totalement du modèle mécanique utilisé pour simuler la déformation des tissus. La valeur de p est alors un paramètre du modèle fourni par l'utilisateur, qui doit être estimé par rapport aux mesures expérimentales afin que la simulation se rapproche de l'expérimentation. Dans un second temps nous désirons calculer ce pincement en utilisant les données des modèles déformables utilisés pour les organes. Avec cette méthode, nous espérons simuler par exemple le cas où une aiguille est insérée dans un objet soumis à des pressions extérieures. La contraction de l'objet devrait alors augmenter la force de frottement s'opposant à l'insertion de l'aiguille.

Nous prendrons l'exemple des modèles à éléments finis, pour lesquels il est possible d'évaluer ce pincement via l'expression des contraintes internes au matériau. Le tenseur des contraintes mesure les forces surfaciques, pouvant être assimilées à des pressions, créées par les déformations du matériau, en tout point du volume de l'objet (voir figure 4.13). Nous récupérons donc le tenseur de contrainte de chaque tétraèdre puis le tournons pour l'aligner dans l'axe du segment de l'aiguille qui traverse cet élément, afin de lire la pression dans les directions normales au corps de l'aiguille. La contrainte σ du matériau est obtenu par le calcul : $\sigma = D\epsilon = DBu$ avec u le déplacement des sommets de l'élément depuis la position de repos, B la matrice faisant le lien entre le déplacement et la déformation, et D la matrice des propriétés physiques du matériau. Ces matrices sont calculées avec la méthode des éléments finis de type corotationnel (voir Nesme [NPF05]).

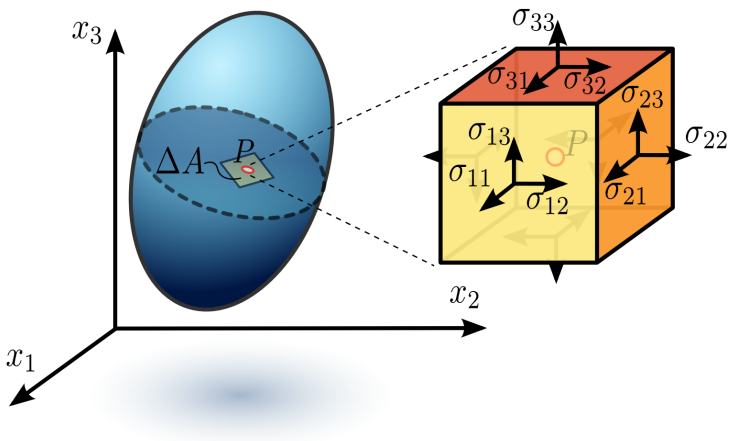


FIGURE 4.13 – Tenseur des contraintes dans un matériau déformable continu.

Nous calculons à chaque pas de temps le tenseur de contrainte pour tous les éléments que traverse l'aiguille. Chaque tenseur est ensuite tourné pour se conformer à la direction du segment de l'aiguille le plus proche. Un repère R fixé sur le segment et dont un axe est confondu avec celui de l'aiguille est utilisé pour transformer le tenseur : $\sigma' = R\sigma R^T$. La force normale pour le frottement utilise alors les données du tenseur dans les directions normales, dont on estime la valeur après intégration sur la distance parcourue par ce segment à l'intérieur

de l'élément :

$$\frac{1}{2}(\sigma_{22} + \sigma_{33})\pi.d \tag{4.1}$$

avec d la longueur du segment dans l'élément, σ_{22} et σ_{33} étant des termes lus dans le tenseur σ. Deux points de contraintes sont souvent séparés de plusieurs segments d'aiguille, eux-mêmes traversant un ou plusieurs éléments du maillage de l'objet, et on additionne alors les valeurs mises à l'échelle en fonction de la longueur des segments en remontant l'aiguille d'une contrainte à la suivante.

Nous obtenons ainsi une estimation de la pression exercée par les tissus sur l'aiguille en fonction de leur déformation. La force de frottement augmente ainsi lorsque l'utilisateur appuye sur l'objet et comprime les éléments entourant le corps de l'aiguille. Cependant, nous ne prenons pas encore en compte les déformations engendrées par l'insertion de l'aiguille elle-même. En effet, lorsqu'elle découpe les tissus au niveau de sa pointe, elle les repousse également dans toutes les directions normales afin de laisser de la place libre pour le volume de l'aiguille. Même si l'aiguille est flexible, on peut considérer que sa section est incompressible, et donc le matériau de l'objet est déplacé d'une distance égale au rayon de l'aiguille. Ce déplacement engendre à son tour déformation et contrainte, d'où provient la partie constante du frottement. Pour calculer cette contribution, nous pourrions déplacer les points du tissu autour de l'aiguille, et mesurer directement la réaction calculée par le modèle déformable. Cependant, pour une mesure correcte il faudrait remailler l'organe pour placer des points directement sur l'aiguille, ce qui est trop coûteux par rapport à la précision supplémentaire que cela nous apporterait. Nous gardons donc une approximation de cette force calculée comme un ressort linéaire tout autour de l'aiguille : $f = \pi kd$ avec d le diamètre de l'aiguille et k la raideur du matériau.

Nous avons mis en place un affichage de la contrainte des éléments de l'objet afin de pouvoir visualiser les déformations des tissus et la pression exercée sur l'aiguille. Cela consiste en une surface triangulée qui intersecte l'objet que l'on veut analyser, et pour laquelle on attribue une couleur à chacun de ses sommets selon l'intensité de la contrainte du matériau à cette position. On récupère tout d'abord le tenseur des contraintes sur les éléments qui nous intéressent, puis on en extrait les valeurs propres $\sigma_1, \sigma_2, \sigma_3$ avec la méthode de Smith [Smi61]. On s'en sert alors pour calculer le critère de Von Mises $\sigma_e = \frac{1}{\sqrt{2}}\sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}$ qui nous permet de représenter la contrainte sous la forme d'un scalaire. Un dégradé continu de couleurs est ensuite utilisé pour faire correspondre à chaque valeur calculée une couleur, avec laquelle on peut alors peindre l'élément du maillage correspondant. Pour faire une coupe de l'objet afin de visualiser les contraintes à l'intérieur d'un maillage volumique, on procède à une intersection entre ce maillage et une surface (plane ou quelconque). Chaque sommet de la surface prend la couleur résultant d'une moyenne pondérée des éléments les plus proches. L'objet pouvant se déformer durant la simulation, le voisinage de chaque sommet de la surface est mis à jour en temps réel. La figure 4.14 illustre cet affichage de la contrainte du matériau lors de l'insertion et le retrait d'une aiguille dans un objet déformable.

Nous avons présenté un travail préliminaire concernant l'impact de la déformation des tissus sur le frottement subie par l'aiguille lors de son insertion. À notre connaissance, aucun travail antérieur ne s'est intéressé à cette problématique. Nous manquons encore de mesures expérimentales pour paramétrer correctement ce modèle d'interaction, ainsi que de témoignages de praticiens sur l'importance que peut avoir ce phénomène dans des cas concrets.

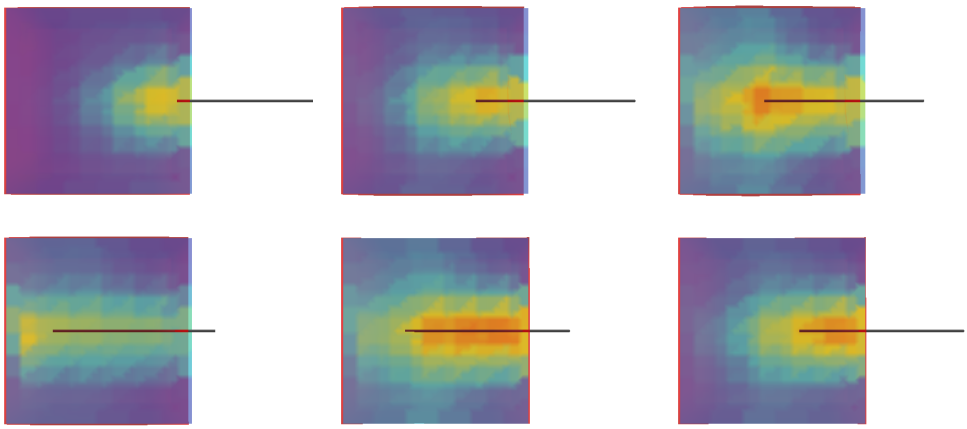


FIGURE 4.14 – Visualisation de la contrainte du matériau lors de l'insertion d'une aiguille. La surface utilisée pour la visualisation réalise une coupe de l'objet selon un plan incluant l'axe de l'aiguille.

Nous pensons que ce travail peut avoir des utilisations concrètes pour d'autres types de simulations, où un fil passe à travers un matériau soumis à des déformations.

4.2.4 Matériau hétérogène

Nous avons présenté deux méthodes pour obtenir la pression p exercée par les tissus sur l'aiguille afin de calculer la force de frottement : soit p est constante et est mise en paramètre de notre modèle d'interaction, soit travailler de concert avec le modèle déformable pour estimer p en fonction du diamètre de l'aiguille et des déformations internes des tissus. Nous proposons une troisième approche, qui est de dire que p varie dans l'espace mais pas dans le temps.

Le frottement peut donc être hétérogène, mais on pose la condition qu'elle reste constante durant la simulation pour chaque point de l'espace et qu'on peut la mesurer. Dans notre simulation, cela revient à réaliser l'intégrale du frottement en utilisant une valeur de p qui est lue localement. On pourrait ainsi plonger l'objet dans une grille de voxel, où une valeur de p est attribué à chaque voxel. Plus simplement, nous utilisons directement le maillage tétrahédrique de l'objet et donnons à chaque élément une valeur.

Les applications sont nombreuses, car il est alors aisé de représenter des inclusions tels que des lobules dans le foie, ou une tumeur dans la prostate, sans avoir besoin de considérer que ce sont des objets séparés. Dans beaucoup de cas, le passage d'un type de tissu à un autre n'implique pas de membrane que l'aiguille doit à nouveau percer, mais uniquement un changement de frottement. On utiliserait alors une modélisation globale sur laquelle seraient calculés les déformations. Ensuite, on assignerait localement différents paramètres p pour séparer les différentes structures anatomiques. Pour aller plus loin, il est tout à fait possible d'utiliser un seul maillage dans lequel on change à la fois les propriétés de l'interaction avec l'aiguille mais aussi les paramètres physiques du modèle déformable pour simuler plusieurs organes dont les surfaces sont liées entre elles. Cela a l'avantage de n'utiliser qu'une seule matrice de compliance pour tout le système et permet de meilleurs temps de calcul. Enfin,

nous pouvons faire varier progressivement les propriétés physiques d'un matériau sans avoir besoin de fixer des zones bien délimitées.

4.3 Discrétisation des contraintes

Bien que nous utilisions un modèle d'organe maillé, et contrairement aux travaux antérieurs sur la simulation de l'insertion d'aiguille, nous n'avons pas besoin de remailler ce modèle ou de repositionner ses sommets sur l'aiguille. Nous sommes conscients que Marchal *et al.* [MPT06] s'en sont également affranchis, mais en utilisant un modèle discret sans maillage. Nous sommes en mesure de créer des contraintes n'importe où dans le volume du tissu, qui mettent en relation un segment de l'aiguille avec un élément de l'objet (dans notre cas, un tétraèdre ou un hexahèdre).

Par l'utilisation d'une intégration implicite ainsi qu'une matrice de compliance précalculée, les forces des contraintes peuvent créer en un seul pas de temps un déplacement sur tous les sommets du maillage. Puisque nous ne sommes pas obligés de créer les contraintes à chaque noeud de l'aiguille, ou encore sur chaque élément du tissu traversé, se pose la question de la création de ces contraintes. Nous ajoutons et supprimons en temps réel les contraintes selon le déplacement de l'aiguille afin d'avoir à tout instant une répartition régulière des contraintes, cela pour que les forces d'interaction soient distribuées correctement à l'intérieur de l'objet. Le but est de maintenir une grande précision dans les calculs tout en permettant une résolution rapide pour la simulation interactive.

4.3.1 Création dynamique de contraintes

Les contraintes placées le long de l'aiguille sont donc créées dynamiquent durant la simulation selon l'avancée de l'aiguille, à deux exceptions près. Rappelons que dans notre méthode, une contrainte est non seulement une loi d'interaction, mais également la position d'un point à partir duquel on calcule la violation. Une contrainte est placée au point d'insertion, dès que l'aiguille a pénétré, remplaçant ainsi la contrainte de contact sur la membrane. Puisque la surface de l'objet et sa représentation volumique peuvent être séparées (comme il a été montré à la section 3.3.2), il peut ne pas y avoir une correspondance directe entre un triangle de la surface et un élément de l'objet. Une recherche est donc effectuée pour trouver le tétraèdre qui inclue le point de pénétration, ou à défaut, celui le plus proche. Cette contrainte est donc la première à être créée dans l'objet, et sera la dernière à être supprimée, uniquement lorsque l'aiguille aura entièrement été retirée. Une contrainte placée à la pointe de l'aiguille est créée en même temps que celle-ci, pour gérer la découpe et la trajectoire prise par l'aiguille. Elle reste active tant que la pointe de l'aiguille est à l'intérieur de l'objet.

Toutes les contraintes d'interaction internes entre le point d'insertion et la pointe de l'aiguille sont ajoutées automatiquement selon un simple critère de distance d'insertion. A chaque pas de temps de la simulation, on calcule la distance d entre la pointe de l'aiguille et la contrainte suivante. Si cette distance atteint un seuil égal à deux fois la distance désirée c entre deux contraintes ($d \geq 2c$), une contrainte est ajoutée sur la courbe de l'aiguille, de telle façon à ce que la distance entre cette nouvelle contrainte et la suivante soit égale à c. Le choix de la distance c est expliqué à la section 4.3.3. Lors du retrait de l'aiguille, on supprime

les contraintes à commencer par celle la plus proche de la pointe, lorsque la distance qui les sépare est inférieure à $c - \epsilon$, avec ϵ une petite distance pour mettre en place un hystérésis et éviter de supprimer une contrainte nouvellement créée. La figure 4.15 illustre l'ajout et la suppression de contraintes.

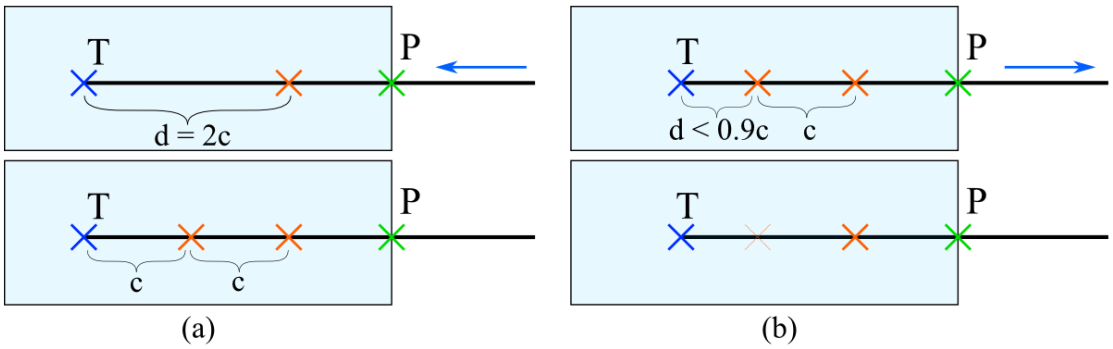


FIGURE 4.15 – Placement dynamique des contraintes d'interaction internes. (a) Ajout d'une nouvelle contrainte lorsque la distance entre la pointe de l'aiguille et la contrainte suivante atteint $2c$. (b) Suppression d'une contrainte si sa distance à la pointe est inférieure à c, avec un certain hystérésis.

Chaque contrainte d'interaction interne met en relation un point placé sur un segment de l'aiguille avec un point plongé dans un élément de l'objet. Bien que la position du point sur l'aiguille soit mise à jour continuellement car l'aiguille glisse dans la contrainte comme nous l'expliquons à la section 4.2.1, la position barycentrique du point de contrainte dans le maillage ne bouge pas au cours du temps. Bien sur, la position absolue de la contrainte changera si l'objet dans lequel est elle définie se déforme ou se déplace. On calcule donc au moment de sa création ses coordonnées barycentriques suivant les sommets du tétraèdre, et on s'en servira tout au long de la simulation pour mettre à jour la position de la contrainte lorsque l'organe se déforme.

Lorsque l'on crée une contrainte dans l'objet, nous avons besoin de connaître le tétraèdre correspondant à cette position et sur lequel on définira la contrainte. Pour éviter de devoir tester tous les tétraèdres du maillage à chaque fois que l'on veut ajouter un point de contrainte, nous utilisons une structure de données enregistrant à la fois les triangles et les tétraèdres traversés par l'aiguille lors de son insertion (voir figure 4.16). Lors de la mise à jour de la contrainte de la pointe de l'aiguille, nous avons précédemment calculé les coordonnées barycentriques de celle-ci par rapport au dernier tétraèdre traversé. On peut ainsi directement savoir si l'aiguille a avancé et traversé une face de ce tétraèdre. Nous mettons alors à jour les structures de données, et refaisons le même test tant que nécessaire au cas où l'aiguille se serait avancée de plusieurs éléments en une fois.

Plusieurs données peuvent être précalculées au moment où on crée une nouvelle contrainte. Ainsi, on mesure la distance avec la contrainte suivante (créée précédemment) dans la configuration non déformée de l'objet, distance qui sera utilisée dans un des calculs possibles du frottement (cas de la pression homogène). Pour intégrer le frottement lorsque la pression change d'un élément à l'autre, on pourra parcourir la liste des triangles traversés pour calcu-

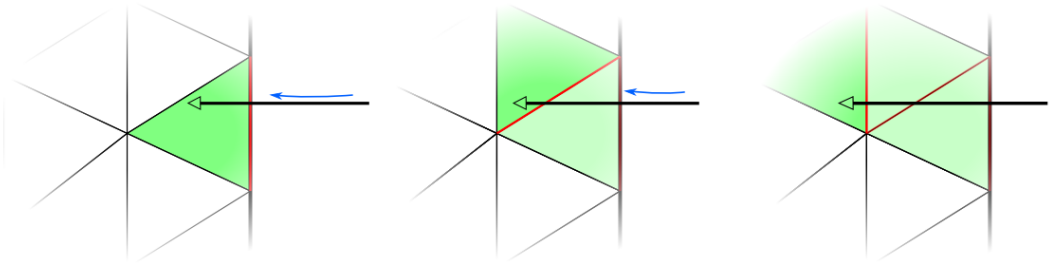


FIGURE 4.16 – Structures de données utilisées lors de l'insertion d'une aiguille. Nous maintenons deux listes, afin de garder en mémoire respectivement les triangles et tétraèdres traversés. Si la pointe de l'aiguille sors d'un tétraèdre, on met à jour en déterminant itérativement la face de sortie du dernier tétraèdre.

ler précisement la longueur d'un segment dans un tétraèdre, puisqu'on peut alors connaître son point d'entrée et son point de sortie s'il y en a un.

Dans notre simulation, nous permettons à l'utilisateur de pouvoir retirer partiellement l'aiguille, pour pouvoir la repositionner et prendre ensuite un autre chemin. Plutôt que de supprimer entièrement les contraintes et les informations sur les éléments traversés, nous pouvons les garder de côté pour une utilisation ultérieure. On peut stocker les différents chemins empruntés par l'aiguille sous la forme d'un arbre où chaque branche est une trajectoire et les feuilles représentent les points les plus avancés atteints par la pointe de l'aiguille. La figure 4.17 illustre les modifications apportées à cet arbre lors d'un repositionnement de l'aiguille. Pour la contrainte de découpe au bout de l'aiguille, nous n'utilisions qu'une mesure de la distance totale de pénétration. Cette structure de données nous permet d'appliquer tout de même la force de découpe si on remarque que l'on emprunte un nouveau chemin. Pour décider si la réinsertion de l'aiguille se fait sur un chemin déjà percé, on calcule la distance entre la pointe de l'aiguille et la courbe qu'elle formait lors de la trajectoire précédente. Si cette distance est supérieure à un certain seuil que l'ont détermine selon les dimensions de la scène et que l'on cherche ou non à reproduire ce comportement, on considère que l'aiguille perce un nouveau chemin. Nous pourrions utiliser ces informations pour appliquer une force pour que l'aiguille ait tendance à suivre un chemin déjà formé plutôt que de pouvoir s'en écarter facilement. Nous avons cependant besoin de mesures expérimentales pour pouvoir quantifier exactement ce phénomène et savoir s'il est réellement utile de l'intégrer à une simulation médicale d'insertion d'aiguille.

4.3.2 Passage d'un objet à l'autre

Nous avons étudié dans les sections précédentes les contraintes qui sont créées lors de l'insertion d'une aiguille dans un tissu déformable, depuis le point d'insertion jusqu'à la pointe de l'aiguille. Nous permettons également à l'aiguille de traverser entièrement l'objet, cas que nous devons gérer particulièrement car il entraîne ajout et suppression de contraintes selon la configuration locale de la scène. Nous allons séparer le cas simple où l'objet traversé par l'aiguille est isolé, et le cas où il est lié par sa membrane à d'autres tissus.

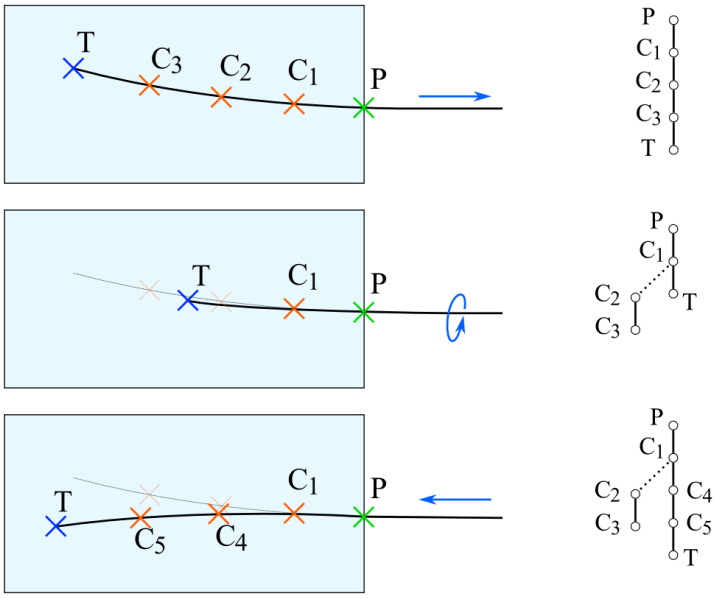


FIGURE 4.17 – Stockage de tous les chemins empruntés par l'aiguille. Lorsque l'aiguille est retirée, les contraintes ne sont pas supprimées mais seulement désactivées, et placées sur une branche séparée d'un arbre représentant les différentes trajectoires.

Lorsque l'on détecte que l'aiguille intersecte un triangle de la surface de collision, ou bien un triangle dans le volume de l'objet mais qu'aucun tétraèdre ne se trouve de l'autre côté, cela signifie que la pointe de l'aiguille sort de l'organe. Nous désactivons alors la contrainte de la pointe de l'aiguille car elle n'est plus en interaction avec les tissus. Nous créeons aussi une contrainte au point de sortie, afin d'empêcher tout mouvement latéral au niveau de la surface. La force de frottement qui était calculée pour la pointe de l'aiguille est alors appliquée à ce point. On réactive ensuite la détection de collision entre la pointe de l'aiguille et ce même objet, dans le cas où il serait non convexe et que l'aiguille pourrait à nouveau pénétrer ailleurs. L'aiguille peut également rencontrer un autre objet, et l'on recommencera la même démarche avec celui-ci, en gardant bien sûr les contraintes déjà existantes.

Dans des scènes plus complexes, les tissus ne sont pas séparés par du vide, mais sont directement en contact les uns les autres. Il peut donc arriver qu'il y ait plusieurs pénétrations, dès que la pointe de l'aiguille entre en contact avec une membrane, avant que l'aiguille ne sorte vraiment des tissus. Dans le cas où une surface est partagée par deux organes, on détecte les éventuelles collisions avant de regarder si l'aiguille est sortie de l'objet dans lequel elle se trouve. On crée alors une contrainte de contact avec cette membrane, et l'aiguille doit pénétrer le second objet avant de pouvoir quitter le premier. Puisqu'une contrainte est toujours créée en réponse à la pénétration de l'aiguille, nous n'ajoutons pas de contrainte au niveau du point de sortie. En effet, nous aurions alors deux contraintes (la contrainte de pénétration et une contrainte de sortie) au même endroit dans le système physique. Avec les infimes imprécisions du calcul du frottement et de la contrainte de chemin, elles seraient légèrement antagonistes et ne pourraient pas être résolues. L'algorithme du Gauss-Seidel ne convergerait pas, et la simulation ralentirait artificiellement parce qu'il ne s'arrêterait qu'après un nombre fixe d'itérations, qu'on choisit en général très grand pour obtenir une bonne précision quand

c'est justifié. Dans une scène de ce type, nous créeons donc une contrainte d'entrée au niveau de chaque membrane traversée, des contraintes d'interaction internes sur le chemin emprunté par l'aiguille, et une seule contrainte de sortie lorsque la pointe de l'aiguille est à l'air libre.

4.3.3 Influence sur les performances

Dans un premier temps nous ajoutions une contrainte d'interaction interne à chaque triangle traversé par l'aiguille. Il arrivait très frequemment que la distance entre deux contraintes consécutives soit très inférieure à la longueur d'un segment de l'aiguille. Le problème posé pouvait rapidement devenir surcontraint dès que le nombre de contraintes se rapprochait du nombre de dégrés de liberté du système. Suite à cette observation nous avons mis en place la méthode de discrétisation régulière des contraintes que nous avons présentée précédemment. La distance désirée entre deux contraintes est un paramètre de notre modèle d'interaction, et se pose alors la question de trouver la valeur idéale pour une scène donnée, ou plus simplement de savoir quelles sont les valeurs qui ne peuvent pas convenir.

Si la distance c voulue entre deux contraintes est petite, nous allons en créer une grande quantité. Dans un premier temps, les degrés de libertés du tissu autour de l'aiguille permettent une déformation complexe de telle façon que toutes les contraintes soient satisfaites. Un échantillonage important des contraintes assure alors une grande précision de la simulation, puisqu'il y aura moins d'approximations dans les calculs du frottement, par exemple. Cependant, si on crée plus de points de contraintes qu'il n'y a de segments de l'aiguille et que les degrés de liberté dans le tissu ne sont pas assez nombreux, le problème devient surcontraint. La résolution des contraintes sera de moins en moins efficace car l'algorithme du Gauss-Seidel ne convergera plus, et l'erreur lorsqu'il s'arrêtera ira en grandissant. On obtient alors le résultat inverse de ce qu'on désirait au départ, à savoir que trop de contraintes nuisent à la précision de la simulation. Les performances sont affectées dans tous les cas par l'augmentation du nombre de contraintes (voir figure 4.18).

On voudra donc réduire le nombre de contraintes utilisé pour obtenir une simulation temps réel. Il faudra cependant éviter une discrétisation trop grossière des contraintes, pour que le frottement soit assez bien réparti le long de l'aiguille. En effet, le frottement n'est appliqué sur les tissus qu'au niveau des contraintes, et si leur nombre est faible la force portée par chacune aura une intensité plus importante. On pourrait alors observer des déformations non réalistes des tissus, ceux-ci étant fortement solicités localement autour des contraintes, et peu déformés dans l'espace entre deux contraintes. Pour garantir une bonne répartition du frottement permettant une déformation continue des tissus, tout en gardant un temps de calcul raisonnable, nous avons observé que la longueur des segments de l'aiguille était souvent une très bonne valeur pour choisir la distance c entre les contraintes.

4.4 Paramétrisation et validation du modèle d'interaction

Nous présenterons dans cette section plusieurs exemples utilisant notre méthode d'interaction basée sur des contraintes de complémentarité. N'ayant pas réalisé nous-mêmes de mesures

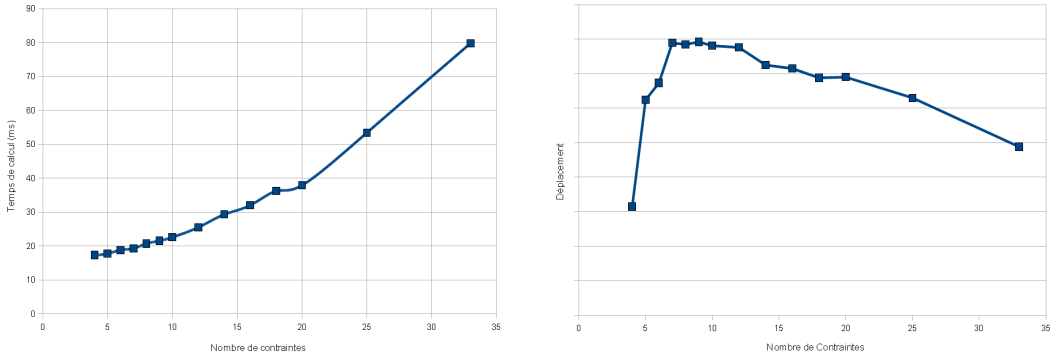


FIGURE 4.18 – Influence du nombre de contraintes sur la vitesse de convergence et la précision de la résolution des contraintes. (à gauche) Temps de calcul de la simulation selon le nombre de contraintes placées le long de l'aiguille, pour une distance d'insertion constante. (à droite) Déplacement d'un point de l'objet proche du corps de l'aiguille. Il se déplace en moyenne de 5mm, et selon le nombre de contraintes on observe une différence pouvant aller jusqu'à 2,6%.

expérimentales, nous nous servirons des résultats publiés par d'autres auteurs pour paramétrer notre modèle. Plusieurs scénarios seront également proposés pour valider notre démarche par rapport aux autres modèles existants ainsi qu'aux mesures auquelles nous avons accès.

Pour l'ensemble de ces exemples, nous utilisons un modèle déformable pour l'aiguille basé sur la théorie des poutres et qui peut se déformer en subissant de grands déplacements. Nous présenterons plus en détails ce modèle mécanique à la section 5.1.3. Pour l'adapter aux besoins de nos simulations, nous pouvons agir sur E le module de Young, ν le coefficient de Poisson et A l'aire de la section de l'aiguille (le modèle permet de simuler des aiguilles creuses). La résolution de ce modèle est optimisée avec l'utilisation d'une matrice tri-diagonale, et le calcul de la compliance sur une aiguille discrétisée finement (50 éléments) est rapide (quelques millisecondes).

Les tissus humains font souvent preuve d'un comportement visco-élastique anisotropique qui demande des modèles éléments finis très complexes lorsqu'une grande précision est requise. Dans nos tests, nous utiliserons des formes géométriques basiques ainsi qu'un modèle de tissu plus simple. Nous choisissons un modèle de type corotationnel (voir par exemple Nesme [NPF05]) qui permet de prendre en compte les non-linéarités géométriques des déformations. Nous utilisons la loi constitutive de Hooke : les déformations sont considérées comme étant isotropes et élastiques, et nous fixons le coefficient de Poisson à la quasi incompressibilité ($\nu = 0.49$). Un comportement visqueux est obtenu avec le modèle de Rayleigh qui fourni une atténuation proportionnelle aux forces élastiques et inertielles. Ce modèle déformable est utilisé ici uniquement dans une approche de validation de notre modèle d'interaction, une simulation médicale nécessiterait un modèle plus complexe. En se servant des travaux de Saupin [SDCG08] sur l'estimation de la matrice de compliance pour un modèle de type corotationnel, nous obtenons un temps de calcul très faible, qui nous permet d'obtenir une simulation temps réel de l'insertion d'aiguille.

4.4.1 Test d'insertion

Dans un premier temps, nous évaluons notre modèle d'interaction dans un cas d'insertion simple. Une aiguille rigide (dans notre cas, sa raideur est très élevée) est insérée dans un matériau déformable dont un côté est fixé. Nous reproduisons l'expérience de Dehghan *et al.* [DWZA+07] où une aiguille est insérée dans un fantome, partiellement retirée, puis réinserée sur la même distance (voir figure 4.19).

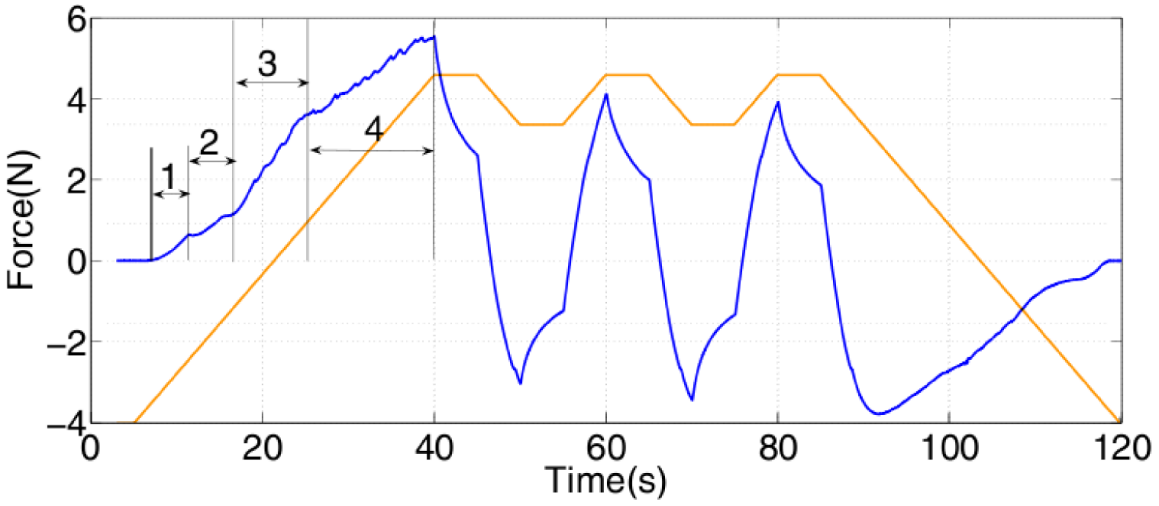


FIGURE 4.19 – Mesure de la force à la base de l'aiguille lors de l'insertion dans un matériau déformable. On représente la force mesurée (en bleu) en réponse au déplacement imposé à l'aiguille (en orange). L'aiguille rencontre lors de son insertion deux types de tissus différents, durant les phases numérotées de 1 à 4. [DWZA+07]

Nous estimons la force appliquée à la base de l'aiguille comme étant la somme des forces de réactions des contraintes dans la direction axiale de l'aiguille. Nous présentons nos mesures à la figure 4.20. Notre simulation donne donc des résultats proches de leurs mesures. La seule phase de l'expérimentation où les résultats s'écartent de l'expérimentation, est lors de la pénétration. Cela s'explique par le fait que l'expérience commence avec l'aiguille déjà très légèrement pénétrée dans la paroi.

4.4.2 Évitement d'un obstacle

L'utilisation d'une aiguille flexible permet de contrôler le chemin imprunté dans les tissus, en appliquant à l'aiguille des déplacements latéraux qui vont la déformer et tourner légèrement la direction de sa pointe. Quelques travaux (comme [DS05b]) ont cherché à mettre en place le planning de l'insertion d'aiguilles symétriques flexibles dans le but d'atteindre un point difficile d'accès en évitant plusieurs obstacles. Nous prenons une autre démarche en proposant une simulation interactive permettant l'entraînement des novices. A terme, cette simulation pourrait aussi permettre l'entraînement pré-opératoire sur des données patients. La figure 4.21 montre une simulation schématique de l'insertion d'une aiguille flexible dans un matériau

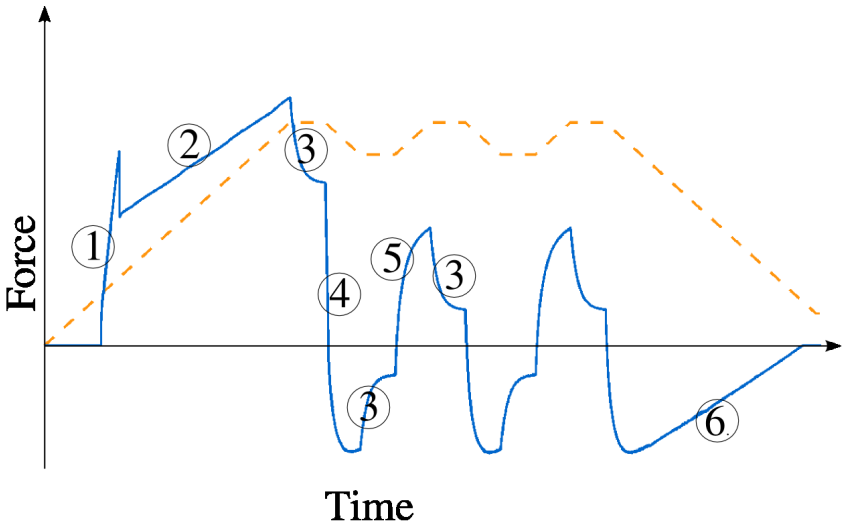


FIGURE 4.20 – Forces d'interaction lors de l'insertion d'une aiguille. Nous imposons un déplacement à la base de l'aiguille (pointillés en orange) et on mesure la somme des forces des contraintes dans la direction de l'aiguille (en bleu). Nous mettons en évidence plusieurs phases de la simulation : (1) la pointe de l'aiguille est en contact avec la membrane qui se déforme alors suite à la pression exercée ; (2) après la pénétration dans le tissu, la force de frottement augmente proportionnellement à la distance d'insertion ; (3) l'aiguille est immobile, il y a relaxation des tissus ; (4) retrait partiel de l'aiguille, la force de frottement s'oppose au déplacement ; (5) réinsertion de l'aiguille sur la même distance, la force de découpe ne s'applique plus car on reste sur un chemin déjà percé ; (6) retrait complet de l'aiguille, le frottement diminue progressivement.

déformable. La tâche de l'utilisateur (non expérimenté) est d'atteindre avec la pointe de l'aiguille une cible tout en évitant un obstacle placé directement entre le point d'insertion et la cible. L'aiguille est contrôlée par un phantom (une interface à 6 degrés de liberté) et la simulation tourne à plus de 50hz sur toute la durée de l'expérience.

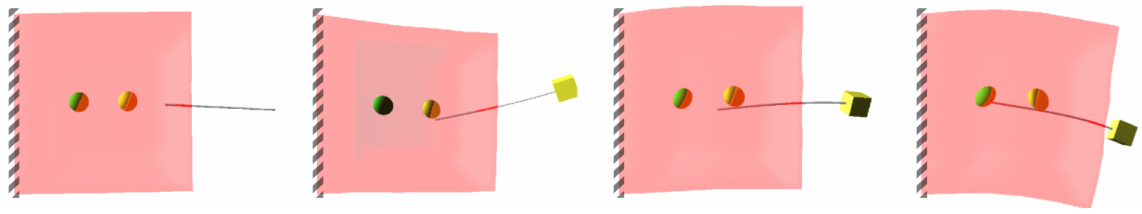


FIGURE 4.21 – Naviguation d'une aiguille flexible symétrique dans un matériau déformable. L'utilisateur contrôle le mouvement de la base de l'aiguille en position et en rotation, et doit atteindre une cible (en vert) en évitant de pénétrer dans un obstacle (en jaune).

4.4.3 Déflexion de l'aiguille

Comme nous l'avons expliqué à la section 2.4.2, une aiguille flexible ayant une pointe biseautée subit une déflexion en pénétrant dans les tissus. Notre modèle permet de reproduire ce phénomène en simulation. Nous voulons reproduire les mesures de Webster *et al.* [WKC+06], qui consistent à insérer dans un fantôme quasi-rigide une aiguille biseautée très flexible, qui se déformera et créera une trajectoire courbe. Dans un premier temps, nous utilisons leur insertion simple (figure 4.22.a) pour paramétrer notre modèle : nous agissons sur la direction de la contrainte de découpe jusqu'à reproduire la trajectoire qu'ils observent. En gardant les mêmes paramètres, nous reproduisons alors l'insertion avec rotation (figure 4.22.b). Nous montrons à la figure 4.23 que nous obtenons sensiblement la même courbe.



FIGURE 4.22 – Déflexions observées lors de l'insertion d'un aiguille biseautée flexible dans un matériau rigide. (à gauche) Une insertion simple durant laquelle l'aiguille est insérée sans la tourner de 23.5cm dans le fantôme. (à droite) Insertion avec rotation : l'aiguille est insérée tout d'abord de 8.3cm, puis elle est tournée de 180 °, et enfin insérée de 16.7cm supplémentaires. [WKC+06]

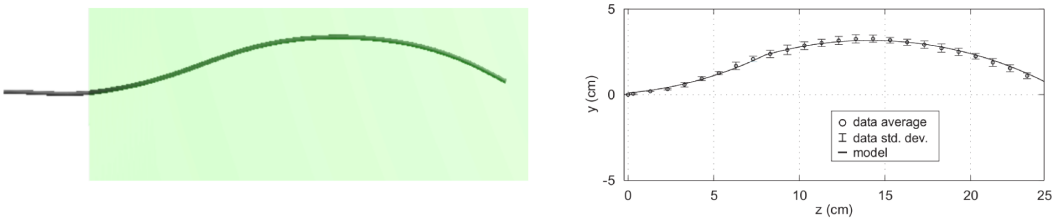


FIGURE 4.23 – Validation de notre modèle de déflexion. (à gauche) Reproduction de l'insertion avec rotation de Webster *et al.* avec notre modèle d'interaction. (à droite) Modèle non-holonomique de Webster et mesures expérimentales.

Une application des aiguilles flexibles biseautées est la possibilité d'éviter des obstacles dans les tissus en déformant moins ceux-ci que dans le cas d'une aiguille rigide. Des algorithmes de planning ([ABG08] par exemple) permettent alors à des robots de reproduire les trajectoires optimales qui ont été calculées. Nous montrons à la figure 4.24 que notre modèle peut également être utilisé dans cet optique, bien que nous n'ayons pas étudié les algorithmes

de calcul de chemin. Nous avons donc fourni nous-mêmes les déplacements à imposer à l'aiguille pour qu'elle évite l'obstacle et atteigne la cible.

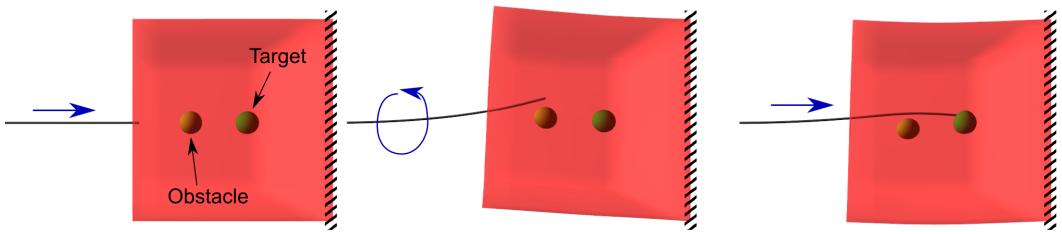


FIGURE 4.24 – Évitement d'un obstacle par l'utilisation d'une aiguille flexible biseautée. L'aiguille est insérée environ sur la moitié de la distance avant d'être tournée de 180 °puis insérée le restant de la distance, et atteint donc la cible sans toucher l'obstacle.

4.4.4 Insertion de plusieurs aiguilles simultanément

Certaines opérations, telles que la curiethérapie avec la méthode pratiquée dans certains hopitaux français, nécessitent l'utilisation simultanée de plusieurs aiguilles ; ce que notre méthode permet de réaliser. Nous utilisons le modèle d'interaction pour toutes les aiguilles, mais dupliquons les structures de données afin de pouvoir faire varier les paramètres d'interaction pour chacune. Les contraintes sont cependant résolues au niveau global, et l'algorithme du Gauss-Seidel permet alors de les faire interagir entre elles, par l'intermédiaire du ou des tissus dans lesquels elles sont insérées.

Les tissus in-vivo sont non-homogènes et les instruments médicaux fins peuvent être déviés par des régions plus dures inclues dans les tissus déformables. Nous simulons ce phénomène en utilisant un maillage volumique 3D composé de deux régions dont la raideur diffère. Lors de son insertion, une aiguille entre en collision avec la surface de la région dure ; comme la force au niveau de la pointe est inférieure à la force de pénétration requise pour pénétrer, l'aiguille glisse le long de la membrane. En gardant les mêmes paramètres pour la simulation mais en augmentant la raideur de l'aiguille, celle-ci réussira à pénétrer la seconde région, comme l'illustre la figure 4.25. Il est également possible de rigidifier la région déformable en insérant plusieurs autres aiguilles. Cela permet une insertion précise de l'aiguille déformable sans avoir à augmenter sa rigidité, et cette technique est communément utilisée pour la curiethérapie. Il faut noter que nous ne détectons pas les collisions entre les aiguilles ; on pourrait l'activer mais nous ne connaissons pas d'exemple concret où cela serait utile. Nous en tenons compte cependant pour la simulation de la suture (voir chapitre suivant).

4.5 Conclusion

Nous avons montré dans ce chapitre l'intérêt d'utiliser des contraintes de complémentarité pour la simulation de l'insertion d'aiguilles flexibles dans des matériaux déformables. Nous reproduisons par des contraintes les différents types de forces d'interactions qui sont courants

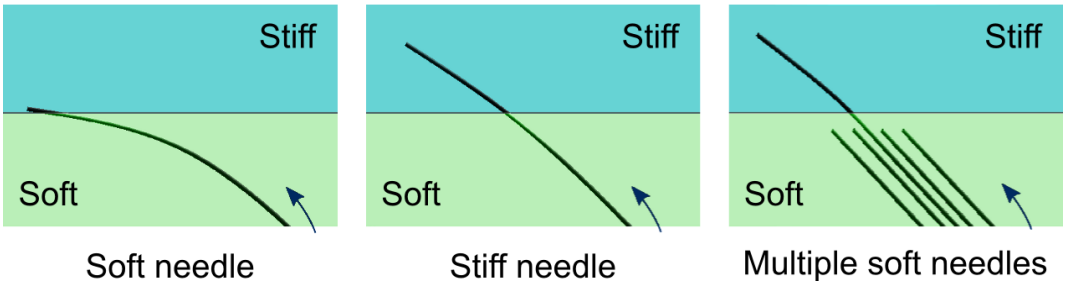


FIGURE 4.25 – Déviation de l'aiguille lors de la pénétration de couches non-homogènes. (à gauche) L'aiguille ne peut pas pénétrer la zone plus rigide et glisse le long de sa surface. (au centre) Première solution : augmenter la rigidité de l'aiguille, qui pourra alors pénétrer. (à droite) Seconde solution : des aiguilles supplémentaires sont insérées dans le tissu déformable, et augmentent localement la rigidité de celui-ci. L'aiguille flexible pénètre alors, car elle ne peut pas se déformer au niveau de la membrane.

dans les travaux actuels basés pour la plupart sur des approches en force. Notre méthode ne nécessite pas de remaillage de l'objet dans lequel est inseré l'instrument, est partiellement indépendante du choix des modèles déformables et a un surcoût assez faible pour permettre une simulation en temps réel. Par ailleurs, notre modèle utilise un nombre réduit de paramètres physiques qui peuvent être extraits des mesures existantes, et permet de simuler des scénarios complexes.

Nous avons validé notre modèle avec des mesures expérimentales de travaux antérieurs, et nous avons proposé quelques idées nouvelles. Certains de nos travaux sont encore en phase d'expérimentation, et nous aurions besoin de faire des tests plus approfondis et si possible les valider par des expériences sur des fantômes. Ainsi, nous proposons de calculer la force de frottement en tenant compte de la pression des tissus qui résulte de leur déformation, et nous donnons des pistes pour contraindre une aiguille à reprendre une trajectoire qui a été précedemment percée dans les tissus lors d'une autre insertion.

Par la suite nous aimerions améliorer certains aspects de notre modèle d'interaction, à savoir le modèle de frottement pour mieux prendre en compte la vitesse d'insertion, et la gestion de la déflexion des aiguilles biseautées afin de se baser sur des paramètres physiques directement mesurables. Nous voulons utiliser et valider notre modèle pour des scènes plus complexes et plus proches des procédures cliniques, et pour cela également employer des modèles déformables plus réalistes, et basés sur des données reconstruites de patients réels. Enfin, nous aimerions travailler à l' utilisation de la simulation comme un outil de planning pour certains protocoles thérapeutiques qui se basent sur l'insertion d'instruments fins et flexibles, tels que la stimulation cérébrale profonde ou l'anasthésie loco-régionale.

Chapitre 5

Vers la simulation du protocole de suture

La suture est une tâche complexe en chirurgie et micro-chirurgie, dont la simulation a fait l'objet de nombreuses recherches récentes, car présentant un intérêt certain pour l'entraînement des praticiens.

On peut séparer principalement deux aspects intervenant dans la simulation de la suture. D'un côté, il faut un modèle déformable 1D pour représenter le fil chirurgical, avec des besoins de temps réel, de réalisme et potentiellement la possibilité d'y faire des noeuds. D'un autre côté, pour s'intéresser à la manipulation réalisée par le chirurgien, il faut créer un modèle d'interaction pour que l'aiguille pénètre dans les tissus, que le fil de suture glisse dans ceux-ci, que les deux parois de la plaie entrent en contact, et que la tension du fil associée au frottement maintienne fermée la plaie lorsque plusieurs points de suture sont réalisés.

Il faut noter que les travaux de recherches se sont portés jusqu'à présent principalement sur le fil de suture, avec des approches purement géométriques et d'autres plus physiques. La recherche sur les interactions qui doivent être prises en compte lors du geste de suture, en particulier le passage du fil dans les tissus, en sont encore à leur début. Les modèles d'interactions proposés ne sont pas à nos yeux suffisamment élaborés pour le degré de réalisme que nous désirons atteindre dans nos simulations. Les résultats que nous avons obtenu avec le modèle d'interaction basé sur les contraintes de complémentarité pour la simulation d'insertion d'aiguille flexible nous laissent à penser que nous pouvons utiliser à nouveau une méthode basée sur les contraintes pour simuler l'interaction du geste de suture.

Nous montrons dans cette partie que l'utilisation d'un modèle déformable 1D réaliste combiné à un modèle d'interaction basé sur les contraintes permet de réaliser une première simulation prometteuse du geste de suture. Nous proposons de représenter le comportement du fil de suture par un modèle basé sur la théorie des poutres, utilisé jusqu'à présent en simulation médicale pour la simulation de catheter. Les contraintes de complémentarité existantes sont adaptées pour prendre en compte les spécificités du fil de suture, et de son interaction avec des tissus mous dans le cadre de la simulation de la suture. Nous avons réalisé pour SCA'09 un poster présentant la méthode et le simulateur de suture obtenu : *Suturing Simulation Based on Complementarity Constraints* [GDC+09]

5.1 Le modèle de fil de suture

On décrit généralement le comportement mécanique d'un fil de suture à l'aide d'un modèle déformable 1D. De tels modèles ont été étudiés dans la communauté graphique depuis plus de 20 ans [TPBF87]. Nous renvoyons à la thèse de Theetten [The07] pour un état de l'art général mais néanmoins très complet de la simulation d'objets déformables 1D.

Cependant, le fil de suture présente des caractéristiques qui lui sont propres : réalisé dans des matériaux synthétiques très résistants, il est peu élastique en élongation, mais son très fin diamètre lui confère néanmoins une flexion importante. Celle-ci permet en particulier de réaliser des noeuds dans le fil de suture, aspect nécessaire pour le geste de suture. Ainsi des modèles ont été créés ou adaptés pour les besoins spécifiques de la simulation de suture. Nous allons dans la suite de cette section présenter plusieurs approches pour la modélisation du fil de suture, aborder la problématique de la réalisation de noeuds, et enfin présenter le modèle que nous choisissons pour notre simulation.

5.1.1 Les modèles déformables 1D

On peut regrouper les modèles déformables 1D utilisés pour le fil chirurgical en quatre principales techniques : les chaînes articulées, les modèles masses-ressort, les modèles de Cosserat, les modèles à base de poutres, et enfin les splines.

Chaînes articulées

Brown *et al.* [BMcLS01] ont proposé de modéliser le fil de suture comme un objet articulé. Les auteurs considèrent que le fil est un cylindre qui se fléchit de façon lisse, avec étirement minimal. Puisque le fil n'est pas élastique, ils décident de ne pas utiliser de ressorts ni de forces pour modéliser son comportement. Les caractéristiques principales du fil sont imposées par un ensemble de contraintes issues de la physique, et qui sont mises à jour à chaque pas de temps. Ces travaux ont été repris dans [BLM04], [MPD05] et [ZGH+07].

Le fil est donc composé de dizaines de segments courts et rigides, tous de la même longueur, connectés entre eux à leurs sommets par des joints sphériques, permettant une rotation selon deux axes. Ces sphères peuvent également servir au moment de la détection de collision, et sont créées de telle façon à se chevaucher légèrement. Au moment du rendu, un arc parabolique est créé entre deux liens successifs afin d'améliorer l'apparence visuelle de la courbure du fil. En suprimant les degrés de liberté sur les angles entre les premiers segments, il est possible de modéliser une aiguille rigide à l'extrémité du fil de suture.

L'utilisateur interagissant avec le fil crée une contrainte entre l'outil virtuel et un ou plusieurs noeuds du fil. D'autres contraintes sont créées lors de la collision, ou pour les tâches de suture à l'emplacement où le fil pénètrent le tissu. Le comportement du fil est alors déterminé selon une approche *suivre le meneur* (voir figure 5.1).

Un noeud contraint N_i est déplacé pendant un pas de temps d'une position x_i^{old} vers x_i^{new}. La nouvelle position de son voisin N_{i+1} est le point x_{i+1}^{new}, placé à une distance fixe d sur la droite formée par x_i^{new} et x_{i+1}^{old}. Le déplacement est alors propagé le long du fil jusqu'à la

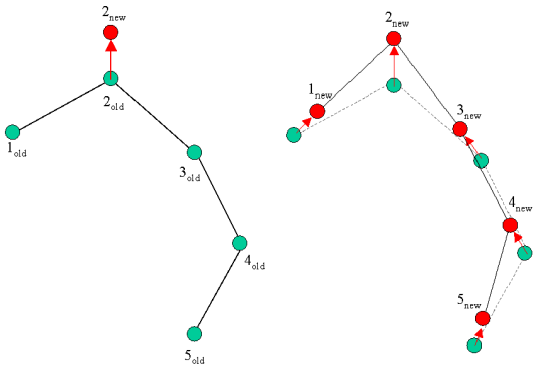


FIGURE 5.1 – (a) Le noeud contraint 2 est déplacé. (b) Les noeuds 1 et 3 suivent, puis 4, puis 5.

prochaine contrainte ou l'extrémité du fil. Le même calcul est entrepris pour le noeud N_{i-1} et ses voisins successifs.

Lorsque deux noeuds sont contraints, par exemple si l'utilisateur se saisit du fil avec deux instruments, alors les mouvements des noeuds intermédiaires calculés selon chaque contrainte peuvent être antagonistes. Dans ce cas, les positions de ces points résultent des moyennes des positions calculés respectivement selon l'une et l'autre des contraintes. Cette moyenne peut avoir pour conséquence de modifier légèrement la longueur de certains segments du fil. Cependant, ces variations de longueur ont tendance à se corriger d'elles mêmes sur plusieurs pas de temps, le fil pouvant glisser dans la majorité des contraintes. La différence entre la longueur supposée d'une section du fil $n.d$ et sa longueur calculée est la *tension* du fil. Elle n'est pas proportionnelle à la rigidité du fil, mais uniquement à la distance suivant laquelle il a été étiré. La tension peut être ensuite appliquée aux points contraints, de la même manière qu'une force, comme étant la réaction du fil sur les objets qui le contraignent.

Ce modèle de comportement du fil est linéaire avec le nombre de noeuds et de contraintes, et nécessite peu de calculs. Malgré sa simplicité, les travaux l'utilisant ont montré qu'il donnait des mouvements visuellement réalistes et physiquement plausibles au fil de suture. Il est souvent employé dans les simulateurs chirurgicaux car il permet de gagner du temps de calcul pour des tâches plus complexes.

Approches masses-ressorts

L'utilisation de particules massiques liées par des ressorts est très courante pour la simulation d'objets déformables 1D. Les premières méthodes ont été proposées pour la simulation de serpents par Miller *et al.* [Mil88], pour la modélisation physique par Thingvold *et al.* [TC90], ainsi que pour l'animation de cheveux par Rosenblum *et al.* [RCT91].

On retrouve l'utilisation d'approches masses-ressorts pour les applications de suture à partir des travaux de Phillips *et al.* [PLK02], qui se sont intéressés à la modélisation de noeuds dans le fil chirurgical. Leur modèle de corde est construit comme une spline de ressorts longitudinaux connectés entre eux par des points de contrôle. En connaissant la longueur

initial d'un segment et son étirement, la force agissant sur un point de contrôle est calculée avec la loi de Hooke : $\mathbf{f} = -k(d - d_0)$.

LeDuc *et al.* [LPD03] utilisent également un assemblage de particules liées par des ressorts pour leur modèle déformable de fil de suture. Ils montrent plusieurs modèles de complexité variable (voir figure 5.2) jusqu'à un modèle composé de ressorts longitudinaux, de ressorts de torsion, d'amortisseurs de torsion et d'un amortissement visceux. Pourtant, ils choisissent un modèle plus simple et moins réaliste composé uniquement de ressorts, sans amortissement. Le calcul de celui-ci est en effet plus rapide, et en utilisant une intégration quasi-statique ils ajoutent un amortissement visceux sur toute la scène sans ajouter à la complexité de la simulation.

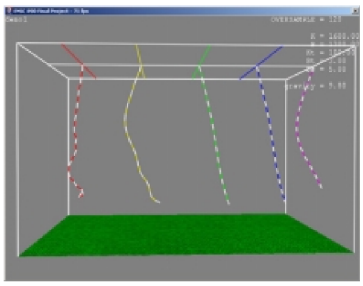


FIGURE 5.2 – Divers modèles de sutures par masses ressorts, par [LPD03]

Pour leur simulateur de microchirurgie, Wang *et al.* ([WBD+05] ; [WBRB05] ; [WSBB08]) ont développé un modèle de fil qui en reproduit les propriétés principales que sont l'élongation, la compression, la courbure et la torsion. La structure topologique du fil de suture est représentée par une liste de particules à quatre degrés de liberté connectées par des liens (voir figure 5.3). Une masse et un moment d'inertie polaire sont assignés à chaque particule et des ressorts longitudinaux et angulaires attachés à chaque lien gèrent le comportement de l'ensemble.

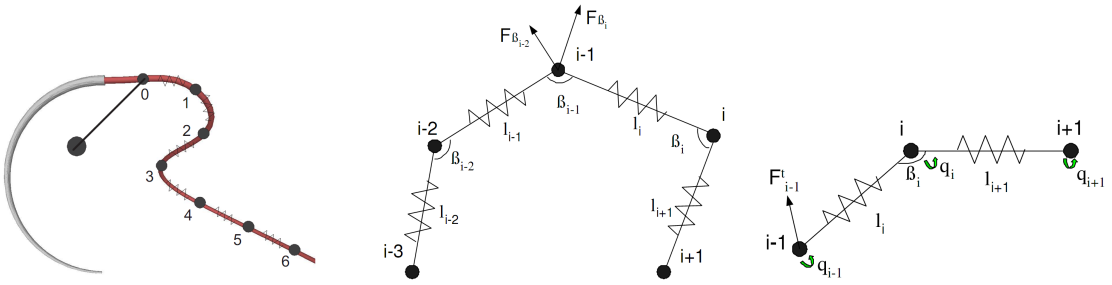


FIGURE 5.3 – La structure du modèle de suture de [WSBB08], ses ressorts de déformation, en élongation, flexion et torsion.

Modèle de Cosserat

La théorie des tiges élastiques des Cosserat [CC09] a été bien étudiée dans le domaine de l'élasticité non-linéaire, et a fait l'objet d'une discussion très complète dans le livre de Antman [Ant95]. Ce type de modèles est composé par une suite d'éléments 1D non-extensibles mais qui gèrent la flexion et la torsion.

Pai [Pai02] a introduit le modèle de Cosserat pour la simulation d'un fil de chirurgie sous le nom de "Strand". La configuration d'une tige est décrite à chaque point de la courbe par une courbure spaciale $\mathbf{r}(s)$ et un repère "directeur" (voir figure 5.4), qu'il assemble sous la forme d'un repère $\mathbf{E}(s) = [\mathbf{e}_1\, \mathbf{e}_2\, \mathbf{e}_3\, \mathbf{r}]\,(s)$. La dérivée de ce repère selon l'abscisse curviligne, $E' = dE/ds$, est alors intégrée en coordonnées relatives dans une équation différentielle cinématique :

$$E' = \begin{bmatrix} [u] & [v] \\ 0 & 0 \end{bmatrix} \tag{5.1}$$

Le vecteur u est appelé le vecteur de Darboux, et pour une tige de Cosserat, u et v sont respectivement les efforts en rotation et en translation. La contrainte $\xi = (m^T, n^T)^T$ est exprimée de la même manière en fonction des forces et moments appliqués par unité de longueur η dans une autre équation différentielle :

$$\xi' = \eta - \begin{bmatrix} [u] & [v] \\ 0 & [u] \end{bmatrix} \xi \tag{5.2}$$

Un des avantages des tiges de Cosserat est qu'elles utilisent les vecteurs directeurs pour calculer la flexion et la torsion, et non les positions absolues. De ce fait, on se place dans le cadre des grands déplacement, et les effets nonlinéaires dus aux déformations sont pleinement pris en compte même si le comportement du matériau est linéaire. Pour les applications graphiques, Pai considère donc des lois constitutives simples :

$$m = K(u - \hat{u}) \tag{5.3}$$

$$n = L(v - \hat{v}) \tag{5.4}$$

avec $\hat{u}$ et $\hat{v}$ les efforts au repos.

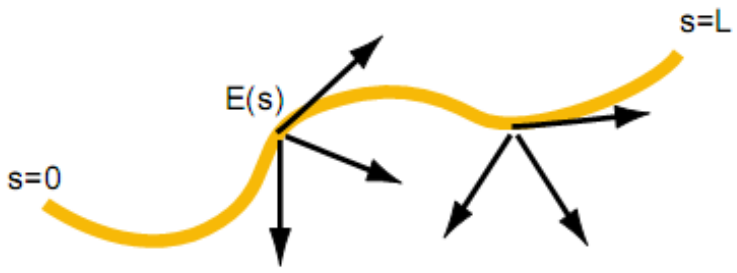


FIGURE 5.4 – Modèle de "Strand" formé de trois segments. [Pai02]

Les équations précédentes forment un ensemble d'équations différentielles ordinaires à résoudre pour déterminer la configuration d'un élément. Pai fait remarquer qu'une formulation en élasticité 3D aurait résulté en un système d'équations différentielles partielles, plus

complexes à résoudre. Ce problème de conditions limites est bien connu dans les simulations d'objets rigides. Pour la résolution, on considère la position et la rotation à une extrémité, les forces et les moments à l'autre extrémité. Le calcul se fait alors en deux étapes : on parcourt une première fois la courbe en partant de la fin pour mettre à jour les déformations et les contraintes, puis on remonte la courbe pour calculer les positions et les orientations de chaque point.

Cette modélisation est physiquement correcte et donne des résultats précis, mais il ne considère pas la dynamique, et l'application de forces nécessite des techniques de "shooting" complexes qui ne convergent pas toujours, comme l'ont montré Grégoire *et al.* [GS06].

En applicant le formalisme des équations de Lagrange aux tiges de Cosserat, Bertails *et al.* [BAC+06] obtiennent l'ensemble des déformations par la résolution d'un système linéaire. Leur modèle de *super-hélices* est utilisé pour la simulation de cheveux et donne des résultats visuellement convaincants, mais la nécessité d'un grande nombre de segments pour la gestion de l'autocollision ([vdHNGT03]) est coûteuse. La complexité quadratique du modèle initial n'est plus un problème, puisque l'utilisation de l'algorithme de Featherstone [Fea87, Fea99] couplé à l'exploitation de la cinématique récursive des *super-hélices* a permis dans des travaux récents [Ber09] d'atteindre une complexité linéaire.

Le modèle de fil de Spillmann *et al.* [ST07] permet une complexité linéaire en temps, mais en utilisant une intégration explicite, qui le rend difficile d'utilisation pour des interactions avec d'autres modèles déformables. En se basant sur les travaux de Grégoire *et al.* [GS06], ils discrétisent les tiges et évaluent les énergies en utilisant la méthode des éléments finis ; le comportement dynamique devenant alors indépendant de la résolution choisie. Leur modèle, appellé *CoRdE*, attribue à chaque point un champ de position ainsi qu'un champ de rotation représenté sous la forme d'un quaternion (voir figure 5.5). Ces deux champs sont couplés, le vecteur tangent du champ de position devant être confondu avec le vecteur tangent du repère correspondant au quaternion. Bien que le résultat soit convaincant, il y a une perte de précision et de stabilité due aux contraintes supplémentaires qui violent l'équilibre des forces.

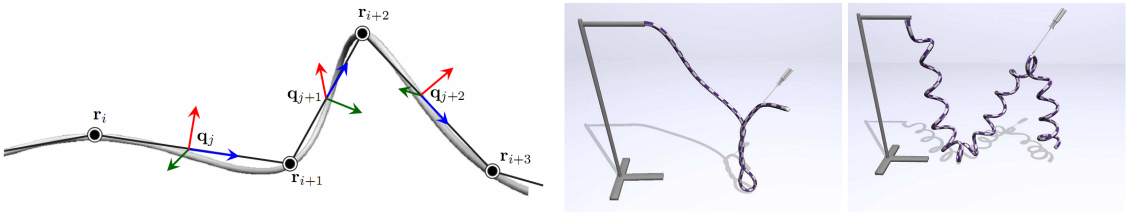


FIGURE 5.5 – Modèle de CoRdE, représenté par un champ de position et des quaternions. [ST07]

Splines

Dans les applications de conception assistée par ordinateur, les splines sont probablement l'outil le plus classique pour la modélisation d'objets 1D. Pour l'animation physique, Terzopoulos *et al.* [TPBF87] ont introduit les splines dynamiques et donnent une formulation énergétique

d'une courbe, basée sur ses déformations géométriques. L'énergie de déformation est dans le cas 1D composée de trois termes, à savoir l'élongation s calculée d'après le tenseur métrique de l'objet, un terme dépendant de la flexion locale κ ainsi qu'un terme de torsion τ dont le calcul n'est pas précisé. L'énergie de déformation est ainsi :

$$\epsilon(r) = \int_0^L \alpha(s - s_0)^2 + \beta(\kappa - \kappa_0)^2 + \gamma(\tau - \tau_0)^2 du \tag{5.5}$$

avec α, β et γ les raideurs respectives à l'élongation, la flexion et la torsion du matériau. La résolution se fait ensuite par une formulation Lagrangienne de l'équation de Newton :

$$\frac{\delta}{\delta t}\left(\mu \frac{\delta \mathbf{r}}{\delta t}\right) = \mathbf{f}(\mathbf{r}, t) \tag{5.6}$$

où $\mathbf{r}$ est la position de la particule, μ la densité massique de l'objet et $f(\mathbf{r}, t)$ est la contribution des forces extérieures au point d'abscisse $\mathbf{r}$ et au temps t. Pour calculer l'énergie de déformation, les auteurs découpent l'espace paramétrique en une grille régulière et définissent les équations du mouvement en chaque noeud de cette grille. Ce modèle est donc une approche discrète car les équations dynamiques sont résolues en des points discrets, mais tout en se basant sur une formulation continue.

Rémion *et al.* [RNG99] [RNG00] reprend le formalisme de Lagrange pour la simulation de splines 1D, sans les discrétiser spatialement. Ce formalisme impose que tout objet soit défini par un ensemble de coordonnées généralisées qui décrivent entièrement la configuration du modèle. Les auteurs font remarquer que les splines sont définies par leurs points de contrôle, et proposent donc d'utiliser ces points de contrôle comme les coordonnées généralisées du modèle. En utilisant l'équation de la spline pour définir le système d'équations dynamiques, le modèle créé est continu, et ses propriétés physiques sont alors définies commes des fonctions paramétriques par une formulation intégrale sur l'abscisse paramétrique. Il est ainsi possible de définir une énergie continue de déformation comme l'ont fait Nocent et Rémion [NR01] en utilisant le tenseur de Green-Lagrange pour mesurer les déformations non-linéaires. Ce modèle est mécaniquement rigoureux et permet de prendre en compte les grands déplacements et grandes déformations.

Pour atteindre une simulation très rapide, Lenoir *et al.* [LMGC02] [LMGC04] se sont concentrés sur une énergie de flexion en petits déplacements. Ils modélisent l'énergie de façon discrète à l'aide de ressorts placés non pas sur les points de contrôle de la spline, mais sur des points de la courbe échantillonnés suivant l'abscisse paramétrique. Trois types de ressorts sont utilisés. Les ressorts d'élongation et de compression (voir figure 5.6.a) sont placés sur chaque couple de points directement voisins, et permettent d'obtenir une énergie discrète d'élongation. Ce même type de ressorts est placé entre les points distants de 2 comme montré sur la figure 5.6.b. Si la configuration locale n'est pas étirée, ceux-ci produisent une énergie de flexion plutôt que d'élongation. Enfin, des ressorts angulaires sont placés sur chaque triplet consécutifs de points de support (comme indiqué sur la figure 5.6.c). La torsion n'a pas été considérée dans le calcul de l'énergie de déformation. Ce modèle a été utilisé pour simuler un fil de suture, et permet des manipulations et interactions complexes tout en gardant un temps de calcul très petit pour une simulation en temps réel.

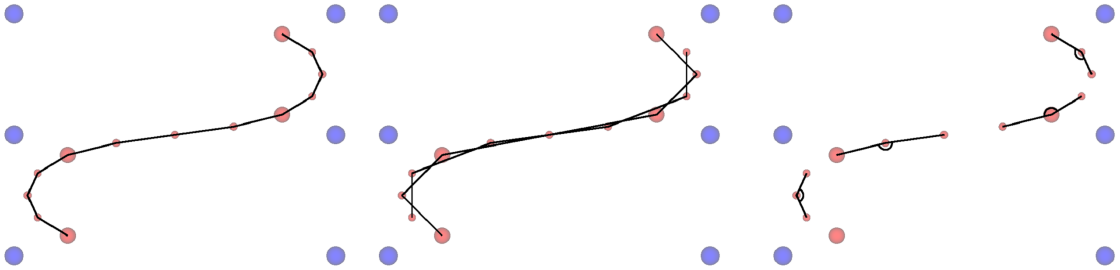


FIGURE 5.6 – Placement des ressorts dans la méthode de Lenoir [LMGC02]. Ressorts en élongation (a), en flexion (b) et les ressorts angulaires (c) dont seulement un sur trois est affiché par soucis de lisibilité.

Theetten *et al.* [TGAB08] [The07] ont proposé un modèle de splines méchaniquement rigoureuses et géométriquement exactes, dans le cadre des petites déformations et grands déplacements. Ils définissent la configuration de l'objet 1D à partir d'un champ de déplacement mais également par un champ de rotation qui donne le roulis de la section (voir figure 5.7.a). Les auteurs appliquent directement des forces et des moments sur le repère local de la fibre moyenne. Ceux-ci sont de trois types, à savoir la force normale correspondant à l'élongation, le moment de flexion due à la courbure de la fibre moyenne et enfin le moment de torsion. L'expression des dérivées de l'énergie de déformation est alors formulée de manière géométriquement exacte. La plasticité des matériaux est également simulée par un décalement de l'élasticité linéaire lorsque la déformation devient plastique. Une optimisation rigoureuse et un basculement automatique entre un schéma de résolution dynamique et un statique [TGDM07] permettent une simulation temps réel.

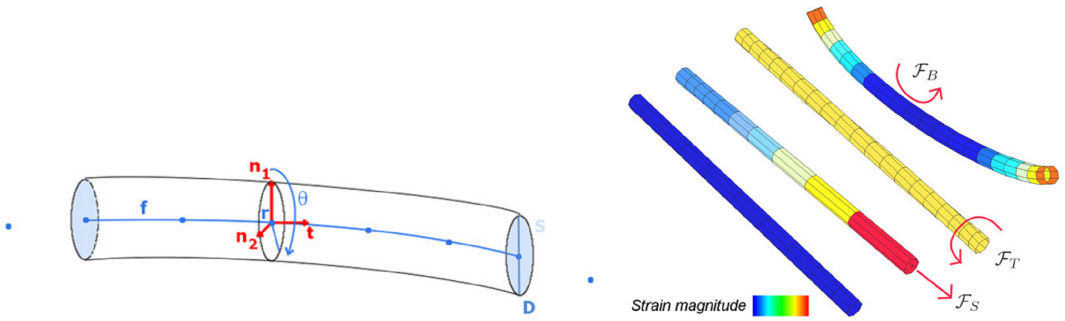


FIGURE 5.7 – Modèle de poutre de Theetten. (a) paramètres géométriques. (b) termes énergétiques : élongation, torsion et flexion.

5.1.2 Réalisation de noeuds

La réussite d'une suture chirurgicale repose en partie sur la compétence du praticien de faire des noeuds solides. Un noeud qui casse ou qui glisse à cause d'une mauvaise réalisation peut avoir des conséquences graves jusqu'à l'ouverture de la plaie. Récemment, un fil de suture ne nécessitant pas de noeuds a été mis au point, dans lequel des crochets ont été créés dans

un fil monofilament. Malgrés des propriétés intéressantes [LRK+03], ce type de fil est encore peu utilisé. La réalisation de noeuds a été étudiée du point de vue de la simulation, que ce soit pour des applications médicales et l'entraînement des chirurgiens, ou plus généralement dans la communauté graphique.

La simulation de noeuds nécessite un grand nombre de degrés de liberté dans une petite région de l'espace afin de répondre au besoin de précision mécanique. Cependant une grande partie de l'objet 1D n'est pas concerné par le noeud et reste peu déformée, et ne demande qu'un petit nombre de degrés de liberté pour le calcul de sa déformation. C'est pourquoi la multi-résolution adaptative parait adaptée dans le cadre de la simulation de noeuds dans une corde ou un fil chirurgical. De plus, les interactions qui apparaissent dans un noeud sont particulières, il s'agit d'autocollisions d'un modèle 1D très déformable, et un segment du fil est souvent compressé par deux autres segments lorsque le noeud est serré. Ainsi, ces collisions font souvent l'état d'un traitement et d'optimisations qui leur sont propres.

Multi-résolution

Pour améliorer l'efficacité et la précision des simulations d'objets déformables, les méthodes adaptatives réduisent le nombre de degrés de liberté là où la précision mécanique n'est pas absolument nécessaire. Le raffinement ou la réduction est gouvernée par un critère d'erreur qui mesure l'adéquation d'une résolution particulière. Cela implique que soit plusieurs résolutions sont précalculées, soit que le maillage est subdivisé quand c'est nécessaire. Pour les objets 3D, la première approche est la plus courante car le remaillage est une tâche complexe. Un framework de simulation adaptative utilisant des modèles éléments finis a été proposé par Debunne *et al.* [DDCB01]. La subdivision de modèles 2D a été bien etudiée [ZSD+99] et des simulations d'objets déformables l'ont intégré, tel que Green *et al.* [GTS02] pour des coques minces. Les objets 1D ont une topologie très simple qui peut être pleinement exploitée pour des méthodes adaptatives.

Pour une simulation de création de noeuds Phillips *et al.* [PLK02] proposent un premier algorithme simple de subdivision adaptative basé sur la distance entre deux points (voir figure 5.8). Lorsque la distance entre deux points de contrôle adjacents est supérieure à β la subdivision intervient, et si la distance entre les extrémités d'un triplet de points est inférieure à α le point central est retiré. Une grande partie des oscillations sont évitées par l'utilisation d'un hystéresis ($\alpha < \beta$). Les nouveaux points sont insérés au centre de masse du ressort et récupèrent la moité de la masse de chaque voisin. Cette répartition des masses est erronée et conduit à des oscillations et des discontinuités énergétiques, qu'ils atténuent en ajoutant de l'amortissement.

Dans sa thèse, Lenoir [Len04] présente une technique de multi-résolution pour son modèle de spline dynamique. La résolution locale peut s'adapter automatiquement en insérant ou en supprimant un seul point de contrôle à la fois. Deux cas peuvent nécessiter l'insertion d'un nouveau point. Un critère d'insertion est la coubure locale de la spline, calculée avec la méthode de Grisoni et Marchal [GM03]. Un point est placé au milieu du segment lorsque les mesures de la courbure sur deux segments consécutifs sont supérieurs à un seuil. L'autre critère est mécanique, dans le sens où un point de contrôle est automatiquement inséré à l'endroit où l'utilisateur interagit avec la spline.

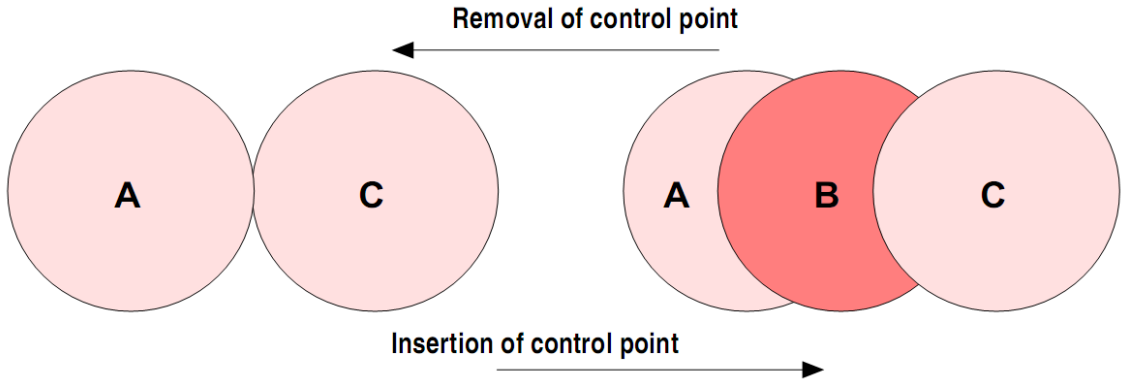


FIGURE 5.8 – Insertion et suppression de points dans la méthode de Phillips [PLK02].

L'algorithme d'insertion de point laisse la spline inchangée au niveau de la forme, mais va engendrer des modifications au niveau mécanique qui demandent de recalculer les matrices de masses et les énergies de déformation. Grâce à la propriété de localité des splines, ce calcul a une complexité en $O(n)$ avec n le nombre de points de contrôle. La suppression d'un point est réalisée lorsque trois points de contrôle sont proches de l'alignement. En pratique, la suppression induit souvent des modifications visibles sur le comportement du modèle si elle n'est pas réalisée à l'équilibre. L'ajout de la multi-résolution dans leur modèle de spline a permis de simuler le serrage d'un noeud dans un fil, sans avoir besoin d'un modèle de départ très détaillé (voir figure 5.9). Une conséquence intéressante de l'utilisation de spline est qu'il est possible de la découper mécaniquement en insérant quatre fois le même point à l'endroit de la découpe.

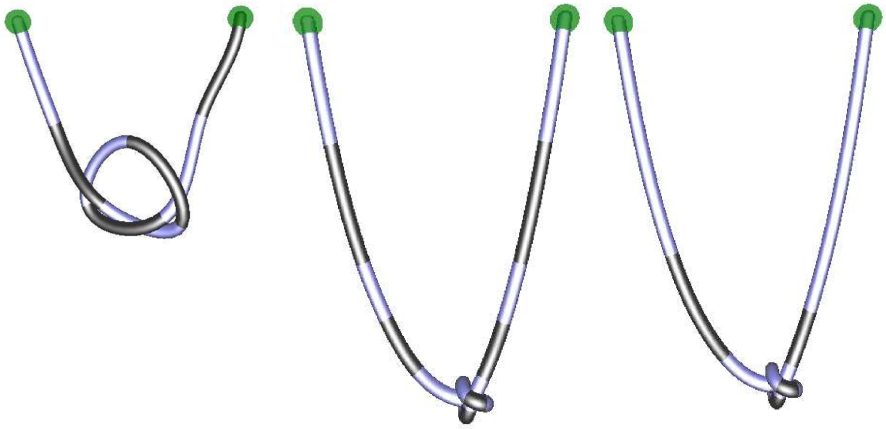


FIGURE 5.9 – Serrage d'un noeud avec le modèle de Lenoir [LMGC02]. (a) sans multirésolution, (b) avec insertion de points, (c) suppression automatique de points.

Spillmann et Teschner ont proposé une méthode adaptative [ST08] pour leur modèle CoRdE de tige de Cosserat en vue d'une simulation de noeuds. Comme pour le modèle de Lenoir *et al.*, ils raffinent la courbe selon un critère géométrique de la courbure locale, qui est également utilisé pour la suppression de points. En plus de considérer les interactions

de l'utilisateur sur le fil, ils s'intéressent à la configuration du contact et subdivisent ainsi un élément si ses deux sous-éléments sont en contact. Ils sont fusionnés à nouveau si aucun contact n'intervient durant un certain temps.

L'insertion du nouveau point au milieu du segment pouvant entraîner une variation de la courbure ou de l'élongation, un des points voisins de celui insérés est déplacé. Le but de cette opération est de trouver les positions optimales pour que l'énergie de déformation sur les deux éléments considérés soit minimale (voir figure 5.10). Ces nouvelles positions sont la solution d'un problème de minimisation contraint non-linéaire, qui est résolu via la méthode du gradient conjugué. Lors de la suppression d'un point, un des points voisins est déplacé de la même façon à la position qui minimise l'énergie.

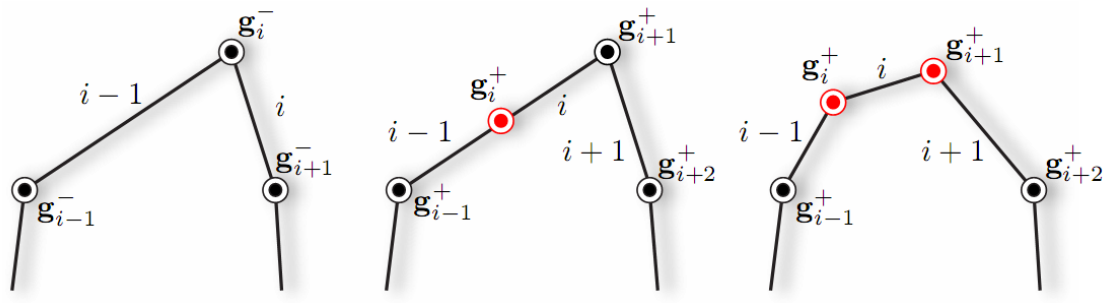


FIGURE 5.10 – Recalcul de la position des points lors d'une insertion dans la méthode de Spillmann [ST08].

Détection et réponse à la collision

Deux approches sont couramment utilisées pour détecter les auto-collisions du fil : on peut chercher les intersections entre les segments reliant les points de contrôle ou alors pour certains modèles qui s'y prêtent mieux, faire une détection de collisions entre des sphères placées le long de la courbe.

Ainsi, Phillips *et al.* [PLK02] placent une sphère de collision sur chaque point de contrôle du modèle de fil. De nouveaux points de contrôle et donc des sphères sont automatiquement créés lorsque le fil se déforme, et il n'y a jamais de vide entre deux sphères consécutives car le critère de création correspond à deux fois le rayon d'une sphère. Le voisinage direct et le second voisinage sont ignorés lors de la détection de collision. La réponse à la collision est basée sur le modèle de pénalité de Mirtich *et al.* [MC95]. Il ignore les forces de frottement et considère une collision élastique, en garantissant que les nouvelles vitesses des points de contrôle le feront se séparer (voir figure 5.11). Ce modèle permet le serrage de noeuds, mais n'est ni valide physiquement, ni temps réel pour une simulation interactive.

Le goulet d'étranglement du modèle précédent est l'étape de la détection de collision. En partant de cette observation, Brown *et al.* [BLM04] proposent de plonger leur modèle de fil non adaptatif dans un arbre de volumes englobants. Les objets 1D s'y prêtent bien, car la proximité topologique des liens est invariante puisque la séquence des liens est fixe, et qu'une proximité topologique implique automatiquement une proximité géométrique. Chaque

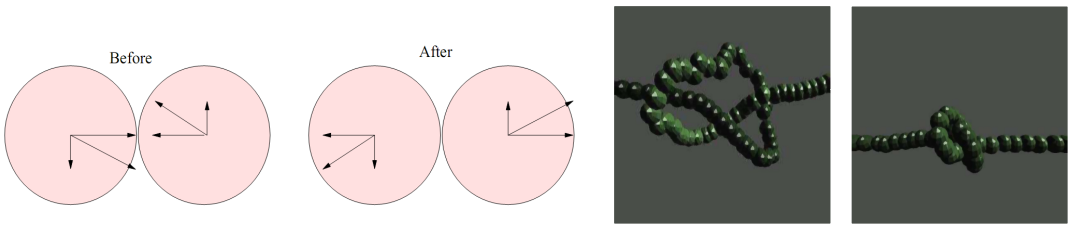


FIGURE 5.11 – Collision de points de contrôle, et serrage d'un noeud ([PLK02]).

segment du fil est donc placé dans une sphère englobante dont la position et le rayon sont recalculés à chaque pas de temps. Un parcours de l'arbre permet de trouver les sphères qui se chevauchent, auquel cas un test d'intersection des segments est réalisé en tenant compte du rayon de la section du fil.

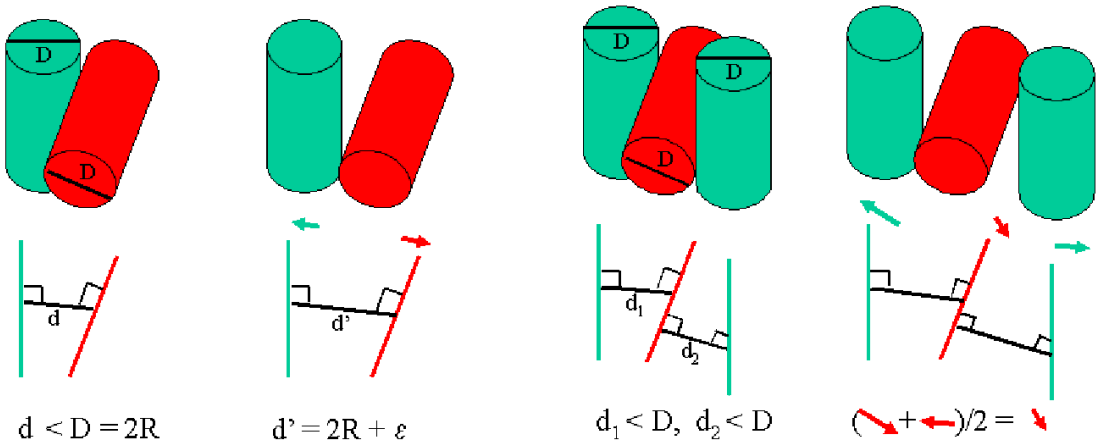


FIGURE 5.12 – Déplacements appliqués aux segments du fil en collision ([BLM04]).

Lorsque deux segments sont en contact, ils sont déplacés d'une même distance permettant de les séparer entièrement, en ajoutant une petite marge ϵ. Si un segment est pressé simultanément par deux objets, le déplacement qui lui est appliqué est la moyenne des deux déplacements calculés indépendamment (voir figure 5.12). De plus, une contrainte de contact est créée qui assure que les segments concernés ne bougeront pas durant le pas de temps suivant. Cette astuce permet de créer un simili modèle de frottement, puisque cette contrainte ajoute une courte sensation d'adhérence. Les segments en collision ayant été repoussés, ils peuvent alors librement glisser l'un sur l'autre au pas de temps qui suivra, et une nouvelle collision sera détectée si besoin. Le comportement de frottement peut être modifié en réglant la distance additionnelle de séparation ϵ. Une certaine gamme de valeurs de ϵ donne des résultats visuellement réalistes, mais l'inconvénient principal de ce modèle d'autocollision est que le frottement dépend directement du temps de calcul. Une optimisation permet de détecter que des noeuds sont créés, de regrouper leurs points de contrôle en des clusters et de déplacer rigidement chaque noeud. Globalement, ce modèle de fil et d'autocollision est très impressionant visuellement (voir figure 5.13) et promet un temps de calcul très petit, mais

est basé plus sur la géométrie que la physique. Ce modèle a été repris par Wang *et al.* pour un simulateur de chirurgie laparoscopique [WBD+05].



FIGURE 5.13 – Création d'un noeud simple avec le modèle de Brown [BLM04].

Lenoir *et al.* [LMGC02] [Len04] reprennent une approche de détection de collision basée sur des sphères. Cependant ils placent les sphères non pas sur les points de contrôle de leur spline, mais les distribuent régulièrement le long de la fibre moyenne du fil. Cette distribution est mise à jour à chaque pas de temps afin de prendre en compte les élongations de l'objet. Puisque le placement des sphères est indépendant de la mécanique, leur rayon peut correspondre au rayon de la section du fil. La détection de collision ne tient pas compte des sphères dont la distance topologique est inférieure à un seuil. La réponse à la collision est une simple méthode basée sur les pénalités, la force appliquée étant modélisée par un ressort placé entre les centres des sphères et dont la longueur au repos est deux fois le rayon des sphères. Cette méthode est simple à mettre en place car elle somme simplement les contributions des forces dues à toutes les collisions. Cependant, elle ne permet pas de gérer les collisions de manière robuste, deux objets pouvant passer au travers l'un de l'autre si l'utilisateur tire plus fort que la force de contact. Ce modèle a permis de simuler en temps réel des noeuds dans un fil de suture mais les noeuds formés sont facilement cassables.

Spillmann *et al.* [ST08] détectent les éventuelles collisions qui apparaissent après une évolution du fil non contraint durant un pas de temps. Une hiérarchie de boites englobantes alignées sur les axes est utilisée pour accélerer la détection des auto-collisions intervenant entre les différents segments. Des contraintes de non-pénétration sont créées et un processus iteratif permet de trouver les déplacements mimimaux qui les satisfont. A chaque itération est ajouté au déplacement de contact une approximation du déplacement dû au frottement. Lorsque les distances d'interpénétration ont été réduites à un certain seuil, les déplacements de contacts permettent de calculer des forces de contact qui sont alors appliquées sur les points de contrôle afin qu'ils soient accélérés vers les positions voulues. Cette méthode permet de simuler des collisions inélastiques, ce qui correspond tout à fait au contact entre deux fils. Les collisions étant détectées à des pas de temps discrets, il peut arriver que certaines collisions soient ratées à cause d'une vitesse trop élevée, cela étant rare en pratique si les pas de temps sont suffisamment petits.

5.1.3 Modèle unifié d'aiguille et de fil

Pour mener à bien notre simulation du geste de suture, nous avons besoin d'un modèle déformable 1D pour le fil de suture. Nous nous imposons certaines contraintes : il doit être temps réel et stable, basé sur les principes de la physique en prenant en compte l'élongation, la flexion et la torsion, et enfin doit pouvoir subir de grands déplacements pour modéliser les configurations complexes de la suture et potentiellement des noeuds. Bien que la plupart des travaux antérieurs aient considéré l'aiguille comme étant rigide, nous désirons pouvoir également en modéliser les déformations. Si possible, à la fois l'aiguille de suture et le fil doivent pouvoir être modélisés avec le même modèle mécanique, en ne changeant que ses paramètres localement.

Le modèle choisi a été proposé dans la communauté graphique par Cotin *et al.* [CDL+05] pour la simulation de catheter en radiologie interventionnelle. A partir de ces travaux, Lenoir *et al.* [LCDN06] ont créé un modèle combiné pour la simulation d'un catheter dans un guide métallique et Dequidt *et al.* [DMD+08] ont simulé la pose de coils lors d'embolisation d'anévrismes. Une de nos contributions est l'utilisation de ce modèle pour simuler le comportement du fil de suture ainsi que de l'aiguille flexible. Les simulations d'insertion d'aiguille que nous avons présentées au chapitre 4 ont également utilisé ce modèle.

Bien connu dans la communauté mécanique et basé sur la théorie des poutres en 3D telle que présentée par Przemieniecki [Prz85], ce modèle est continu et résolu avec la méthode des éléments finis. Un modèle de poutre décrit les déformation d'une courbe représentant la fibre moyenne des efforts de déformation. Le fil est discrétisé par des éléments de poutre reliés en série, chaque élément ayant 6 degrés de liberté : 3 en position et 3 en rotation. On calcule pour chaque élément une matrice de rigidité élémentaire $\overline{K}_e$ de dimension (12,12), qui relie les positions spatiales et angulaires des deux extrémités aux forces et moments qui leur sont appliqués :

$$\overline{K}_e = \frac{E}{l}\begin{bmatrix}
A & & & & & & & & & & & \\
0 & \frac{12I_z}{l^2(1+\Phi_y)} & & & & & & & & & & \\
0 & 0 & \frac{12I_y}{l^2(1+\Phi_z)} & & & & & & & & & \\
0 & 0 & 0 & \frac{GJ}{E} & & & & & & & & \\
0 & 0 & \frac{-6I_y}{l(1+\Phi_z)} & 0 & \frac{(4+\Phi_z)I_y}{1+\Phi_z} & & & & & & & \\
-A & 0 & 0 & 0 & 0 & 0 & A & & & & & \\
0 & \frac{-12I_z}{l^2(1+\Phi_y)} & 0 & 0 & 0 & \frac{-6I_z}{l(1+\Phi_y)} & 0 & \frac{12I_z}{l^2(1+\Phi_y)} & & & & \\
0 & 0 & \frac{-12I_y}{l^2(1+\Phi_z)} & 0 & \frac{6I_y}{l(1+\Phi_z)} & 0 & 0 & 0 & \frac{12I_y}{l^2(1+\Phi_z)} & & & \\
0 & 0 & 0 & \frac{-GJ}{l} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{E} & & \\
0 & 0 & \frac{6I_y}{l(1+\Phi_z)} & 0 & \frac{(2-\Phi_z)I_y}{1+\Phi_z} & 0 & 0 & 0 & \frac{6I_y}{l(1+\Phi_z)} & 0 & \frac{(4+\Phi_z)I_y}{l(1+\Phi_z)} & \\
0 & \frac{6I_z}{l(1+\Phi_y)} & 0 & 0 & 0 & \frac{(2-\Phi_y)I_z}{1+\Phi_y} & 0 & \frac{-6I_z}{l(1+\Phi_y)} & 0 & 0 & 0 & \frac{(4+\Phi_y)I_z}{1+\Phi_y}
\end{bmatrix} \tag{5.7}$$

avec $G = \frac{E}{2(1+\nu)}$ où E est le module de Young et ν le coefficient de Poisson ; A est l'aire de la section de la poutre et l sa longueur ; I_y et I_z sont les moments d'inertie de la section ; $\Phi_y = \frac{12EI_z}{GA_{sy}l^2}$ et $\Phi_z = \frac{12EI_y}{GAszl^2}$ représentent les paramètres de déformation en cisaillement avec A_{sy} et A_{sz} les aires de cisaillement dans les directions y et z.

Les matrices de rigidité $\overline{K}_e$ sont calculés dans des coordonnées locales définies dans le repère associé au premier noeud de la poutre. Pour passer des déplacements $\overline{\mathbf{u}}$ locaux à l'élément aux déplacements $\mathbf{u}$ dans les coordonnées globales, on utilise la matrice Λ définie par :

$$\overline{\mathbf{u}} - \overline{\mathbf{u}}_0 = \Lambda \mathbf{u} \tag{5.8}$$

avec $\overline{\mathbf{u}}_0$ la configuration initiale de la poutre. Λ est une matrice de coefficients obtenue à partir des cosinus des angles entre les systèmes de coordonnées locales et globales. La matrice de rigidité K_e dans l'espace global est alors construite : $K_e = \Lambda^T \overline{K}_e \Lambda$. Lorsque la poutre se déforme, Λ est modifiée et doit être recalculée, alors que $\overline{K}_e$ reste constante pour des petites déformations de la poutre dans le repère local. Pour prendre en compte de grands déplacements, il faut que la discrétisation soit suffisante pour que la déformation de chaque élément reste petite.

Si le premier noeud du modèle est contraint, l'équation de la première poutre peut être utilisée pour calculer le repère local du second noeud, et il est ainsi possible itérativement de mettre à jour Λ pour chaque élément. Dans le cas contraire, la mise à jour utilise la solution obtenue au pas de temps précédent, plutôt que la configuration initiale comme c'est généralement le cas avec la méthode du corotationnel. Pour permettre au modèle en l'absence de forces de revenir à l'état initial plutôt que seulement à l'état précédent, une force élastique $\overline{\mathbf{f}}_e$ est ajoutée sur chaque élément :

$$\overline{\mathbf{f}}_e = -\gamma \overline{K}_e (\overline{\mathbf{u}} - \overline{\mathbf{u}}_0) \tag{5.9}$$

avec $0 < \gamma \leq 1$.

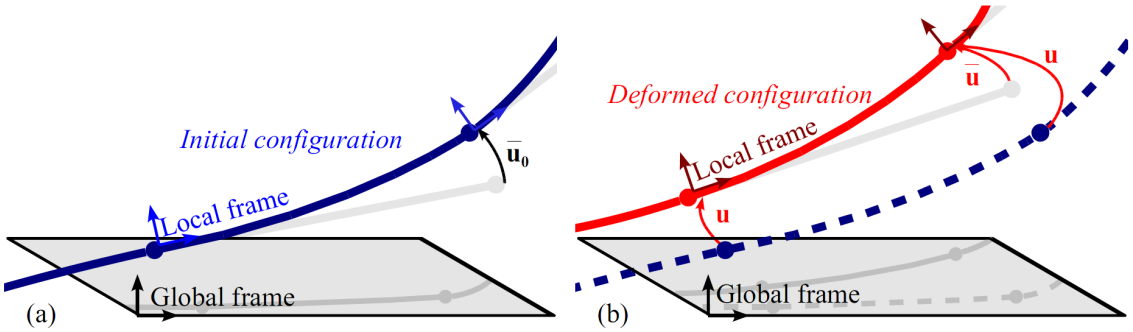


FIGURE 5.14 – Configurations initiale (en bleu) et déformée (en rouge). (a) Le déplacement initial du second noeud dans le repère local du premier noeud est $\overline{\mathbf{u}}_0$. (b) Sa déformation est notée $\overline{\mathbf{u}}$.

Les éléments de poutre étant connectés en série, la matrice de rigidité K globale est une matrice tridiagonale avec une largeur de bande de 12, puisque chaque K_e est une matrice de dimension(12,12). En positionnant la masse sur les noeuds, les matrices de masse et d'amortissement sont également diagonales et le système linéaire est résolu alors avec l'algorithme proposé par Kumar *et al.* [KP93]. La solution est obtenue avec une complexité en $O(n)$ plutôt que $O(n^3)$, permettant ainsi une simulation en temps réel même pour des modèles de plusieurs centaines d'éléments.

Par rapport aux modèles communemment utilisés pour la simulation de fil de suture, celui-ci gère la torsion autour de son axe central. Celle-ci peut avoir un effet non négligeable dans les gestes de suture, car la manipulation continue de l'aiguille (entre autres le fait de la passer d'un instrument à l'autre) engendre souvent un grand nombre de rotations du fil. La figure 5.15 montre des expérimentations pour mettre en évidence le couplage flexion-torsion du modèle. Le modèle prend également en compte la résistance au cisaillement, aspect qui en pratique sera peu utile pour la simulation du fil de suture.



FIGURE 5.15 – Mise en évidence du couplage flexion-torsion. L'extrémité de droite est fixe, et l'on applique une torsion à l'extrémité gauche du fil.

Ce modèle a été validé expérimentalement par Dequidt *et al.* [DMD+08] dans le cadre d'un déployement de coil pour une procédure d'embolisation d'anévrisme. Les coils qu'ils ont simulés sont des bobines hélicoïdales d'un fil de platine très fin et possédant une forme au repos qui est documentée. Un coil a été plongé dans un environnement sans contact et uniquement sous l'influence de la gravité, puis sa configuration obtenue à l'aide d'une technique d'angiographie 3D par rayons X. La même scène a été simulée à l'aide du modèle de poutres, l'erreur relative mesurée étant très faible, prouvant ainsi la précision du modèle déformable.

En utilisant une paramétrisation adéquate des éléments de poutre, il est possible tout aussi bien de simuler des objets très déformables tels que le fil de suture que des objets quasi-rigides comme les aiguilles. Nous simulons alors l'assemblage aiguille et fil comme un objet 1D formé d'une seule série de poutres, mais en paramétrant différemment les éléments qui correspondent à chacun. Ainsi, la résolution se fait directement sur l'ensemble, et nous n'avons pas à gérer le passage depuis l'aiguille sur le fil différemment que d'un élément à l'autre.

Ce modèle présente l'avantage d'être paramétré avec les mesures physiques standards des matériaux. Nous n'avons donc pas besoin d'identifer les paramètres à partir d'expériences, car les propriétés des fils chirurgicaux et des aiguilles de suture sont connues et souvent communiquées par les fabricants. Ainsi, les aiguilles de suture qui sont essentiellement rigides sont modélisées avec un module de Young $E = 200GPa$, un coefficient de Poisson $\nu = 0.3$, une masse volumique $\rho = 2800kg/m^3$ et un diamètre de $d = 1mm$. Pour notre simulation, nous considérons un fil de suture fabriqué en nylon, avec les propriétés suivantes : module de Young $E = 2.8GPa$, coefficient de Poisson $\nu = 0.4$, masse volumique $\rho = 1160kg/m^3$ et un diamètre $d = 0.25mm$. Contrairement à beaucoup de travaux antérieurs sur la simulation de suture, le fil est donc extensible. Mais en pratique, avec les paramètres décrits ci-dessus, on observe que le modèle se déforme plus facilement en flexion et en torsion qu'en élongation.

Nous modélisons visuellement le fil de suture avec la technique des cylindres généralisés de Binford [Bin87] qui permet de générer une surface lisse à partir d'une courbe et d'une coupe transversale (voir figure 5.16). La déformation de la fibre moyenne du fil est alors appliquée à ces cylindres. Il est possible d'utiliser une discrétisation des cylindres différente de celle du fil en utilisant une interpolation de la coubure à partir des repères locaux des éléments de poutre. On obtient ainsi une représentation visuelle plus raffinée que le modèle mécanique, en particulier dans les zones courbes.

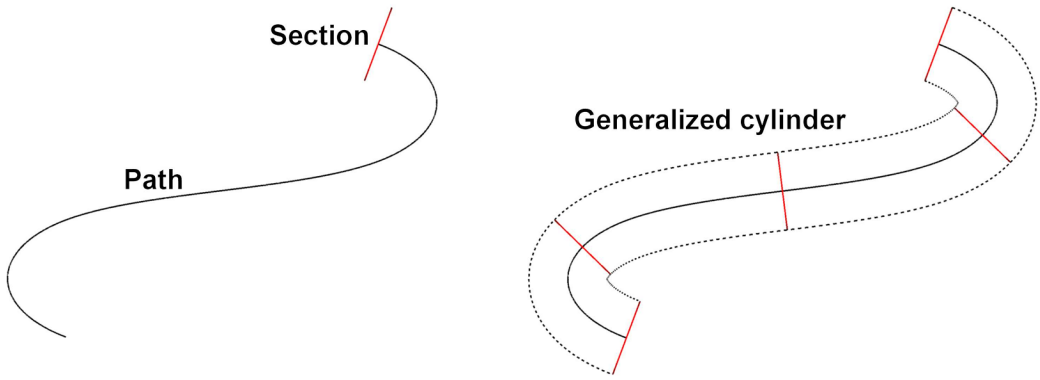


FIGURE 5.16 – Construction d'un cylindre généralisé autour du modèle de fil ([LCDN06]).

Dans la suite de ce travail, nous utilisons ce modèle déformable pour représenter à la fois aiguille et fil de suture, en choisissant une discrétisation régulière assez fine pour prendre en compte les déformations fortes pouvant intervenir au niveau d'un point de suture. Dans un premier temps nous n'utilisons pas d'adaptation dynamique de la discrétisation du fil de suture, ce qui nous empêche de représenter efficacement les courbures extrèmes nécessaires pour réaliser des noeuds.

5.2 Interactions lors du geste de suture

En partant de l'observation que les interactions intervenant lors du passage d'un fil de suture dans un tissu sont très similaires à l'insertion d'une aiguille dans ce même tissu, nous voulons généraliser notre méthode pour réaliser la simulation d'un geste de suture. Nous réutilisons les contraintes présentées précédemment pour l'insertion d'aiguille, en les modifiant lorsque c'est nécessaire pour prendre en compte les spécificités de cette opération. De plus, le fil de suture étant par nature très souple il peut entrer en contact avec tous les objets en présence, y compris lui-même, là où les interactions se concentrent pour les aiguilles principalement à leurs extrémités.

5.2.1 Les modèles d'interaction existants

La plupart des simulateurs de suture existants s'intéressent au geste de suture tel qu'il doit être appris par les praticiens, et leur permet de mettre en pratique leur compétence afin

de s'améliorer. Un de leurs aspects les plus importants est donc le modèle d'interaction choisi pour simuler l'interaction du fil avec le tissu qui doit être suturé.

L'aspect planification a été abordé dans les travaux de Nageotte *et al.* [NZdMD05], qui ont réalisé un simulateur calculant le chemin optimal de l'aiguille pour une tâche de suture en chirurgie laparoscopique. En effet, comme l'a montré Cao *et al.* [CMP96] la suture y est très complexe, à la fois à cause des contraintes cinématiques imposées par les trocards et parce que les praticiens n'ont pas de retour visuel direct une fois que l'aiguille a pénétré les tissus. Ainsi, par une modélisation géométrique de la tâche de suture, ils calculent un chemin possible qui peut alors être présenté au chirurgien via une méthode de réalité augmentée, ou être utilisé par un robot (voir figure 5.17).

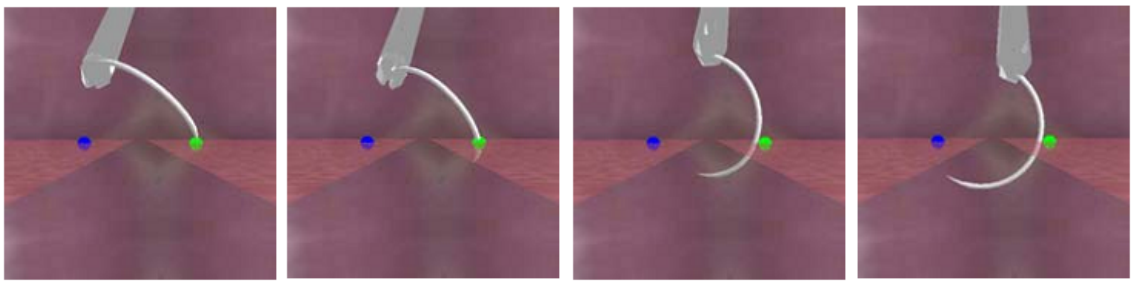


FIGURE 5.17 – Chemin de l'aiguille calculé avec la méthode de Nageotte *et al.* [NZdMD05].

Lorsque l'aiguille chirurgicale entre en contact avec le tissu, deux approches différentes sont actuellement utilisées. La méthode la plus simple est de faire passer le fil par le point du maillage le plus proche. Pour que la simulation soit visuellement plausible, il faut alors une discrétisation assez fine pour qu'il y ait pour chaque point de contact possible un noeud du maillage assez proche. Cette technique a été utilisée dans les travaux pionners de LeDuc *et al.* [LPD03] (voir figure 5.18) pour la suture d'objets surfaciques, et reprise pour la suture de vaisseaux sanguins par Zhang *et al.* [ZGH+07].

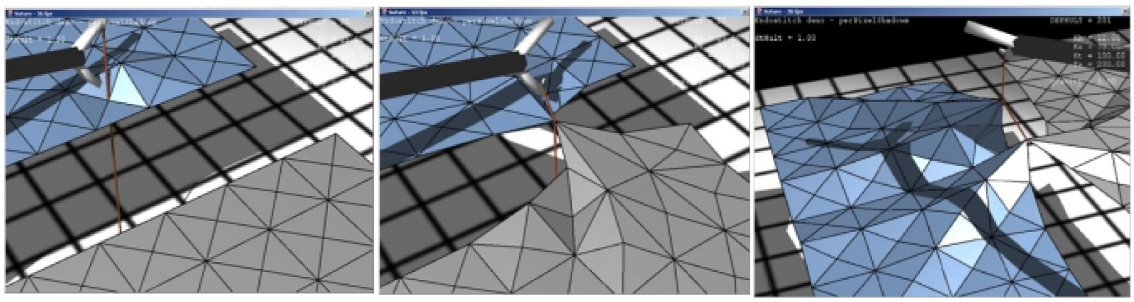


FIGURE 5.18 – Le fil de suture passe par les sommets du maillage dans la méthode de LeDuc *et al.* [LPD03].

Pour une plus grande précision lors de la pénétration de l'aiguille, Marshall *et al.* [MPD05] ont choisi de remailler localement la géométrie du tissu afin de créer un point à l'endroit exact du contact (voir figure 5.19). Ils utilisent pour cela l'algorithme de subdivision de Loop

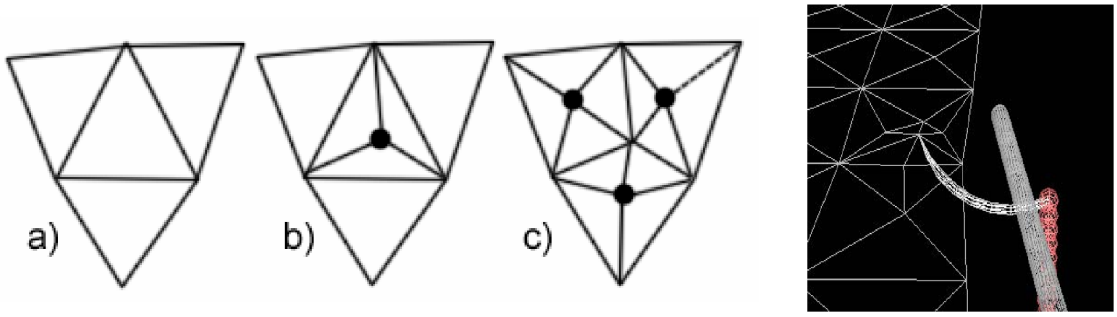


FIGURE 5.19 – Technique de remaillage pour le point d'entrée du fil de suture (Marshall *et al.* [MPD05]).

[Loo87], raffinent deux fois les triangles puis déplacent légèrement les nouveaux points pour que les triangles autour du point d'entrée ne soient pas trop petits. Shi *et al.* [SP08] remaillent également les polygones avec lesquels l'aiguille entre en contact, sans donner de précision sur la méthode utilisée. L'un des inconvénients des techniques de remaillage est que les objets peuvent rapidement devenir complexes si plusieurs points de suture sont réalisés.

Les premiers modèles d'interactions entre le fil et les tissus dans les simulateurs de suture se sont basés sur la nature discrète du fil. Fil et tissu sont alors tous les deux discrétisés en un ensemble de particules ou de "noeuds". Ainsi, LeDuc *et al.* [LPD03] traitent un des noeuds du tissu comme étant le trou, et le connectent à un des noeuds du fil de suture. Lorsque le fil est tiré, le noeud du tissu se déplace avec lui, entraînant ensuite le reste de l'objet. Lorsque les forces appliquées à ces noeuds par les équations des modèles masses-ressorts sont supérieures à un certain seuil correspondant à la force de frottement, le fil glisse dans l'objet (voir figure 5.20). Pour simuler le glissement, le noeud N est détaché du noeud S_0 de la suture puis attaché à nouveau au noeud suivant S_1. Si le fil de suture continue d'être tiré, ce noeud de l'objet pourra être à nouveau détaché au pas de temps suivant, donnant l'impression que le fil glisse au travers d'un trou dans le tissu. Le mouvement du fil est donc saccadé tout comme le point à la surface du tissu qui oscille avec la traction et le relâchement successifs du fil. Pour cacher ces phénomènes, la longueur des segments de la courbe doit être réduite, induisant alors une plus grande discrétisation du fil et un temps de calcul plus long. Le cas où deux objets glissent en même temps le long du fil de suture doit être géré particulièrement pour que leur ordre sur le fil ne change pas.

Zhang *et al.* [ZGH+07] utilisent la même technique en ajoutant une simulation du perçage de l'aiguille. Ainsi, le noeud à la surface de l'objet résiste d'abord à la pression exercée par l'aiguille, déformant localement le tissu. Lorsque la tension dépasse un seuil, ce noeud est attaché au fil et celui-ci peut alors glisser à travers le trou.

Marshall *et al.* [MPD06] sont les seuls à notre connaissance qui modélisent l'ensemble des forces s'appliquant à l'aiguille et au fil lors d'une simulation de suture. Nous ne revenons pas sur les forces agissant uniquement sur l'aiguille, qui ont été étudiées à la section 2.1.2. Bien qu'ils utilisent un modèle de fil entièrement géométrique, ils proposent un calcul de sa tension. La tension du fil entre deux contraintes correspond à son élongation par rapport à la longueur initiale de ce segment, et n'est pas proportionnelle à la déformation. Cette tension peut alors être appliquée à l'objet auquel est attachée la contrainte, tel que l'aiguille ou le tissu (voir

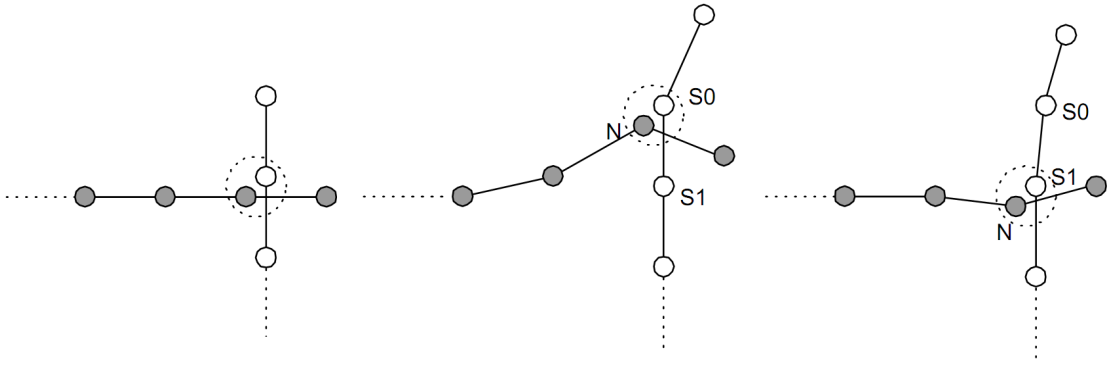


FIGURE 5.20 – Le fil de suture glissant dans un trou du tissu (LeDuc *et al.* [LPD03]). (a) Un noeud du fil (en blanc) est connecté à un des noeuds du tissu (en gris). (b) Celui-ci se déplace lorsque l'utilisateur tire sur le fil. (c) Quand la force de réponse du modèle atteint un certain seuil de "frottement", les deux noeuds sont détachés, et une nouvelle connexion est établie avec le noeud suivant du fil.

figure 5.21). Le fil peut glisser au travers du tissu lorsque la force tangentielle au fil dépasse le frottement. Une approximation de cette force est calculée à partir de la tension du fil des deux côtés du noeud appartenant au tissu. Le mouvement du fil à l'intérieur de l'objet est contraint par des ressorts latéraux qui ont tendance à le ramener sur la courbe formée par le tracé de l'aiguille.

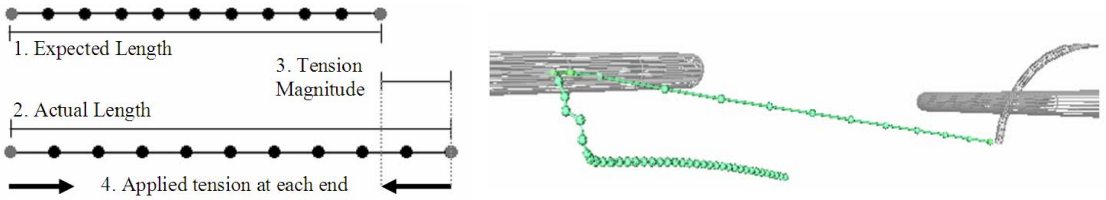


FIGURE 5.21 – Calcul de la tension du fil de suture et exemple durant la simulation (Marshall *et al.* [MPD06]).

Lenoir *et al.* [LMGC04] proposent d'utiliser des contraintes de point glissant pour modéliser une tâche de suture (voir figure 5.22). Ces contraintes sont fixes sur l'objet, permettent au fil de glisser le long du point contraint, et imposent au fil une direction définie par l'orientation de l'aiguille à ce point. Le modèle de frottement de Coulomb est employé pour contrôler le glissement du fil. La déformation de l'organe due à la traction du fil n'est pas directement prise en compte par les contraintes. En effet, les points de l'objet sont considérés comme étant statiques durant un pas de temps donné. En utilisant le principe d'action/réaction, les forces normales et de frottement calculées pour les contraintes sont inversées et appliquées à l'organe. Pour garantir la stabilité du système, la résolution des contraintes est basée sur la méthode des multiplicateurs de Lagrange et une intégration implicite est utilisée.

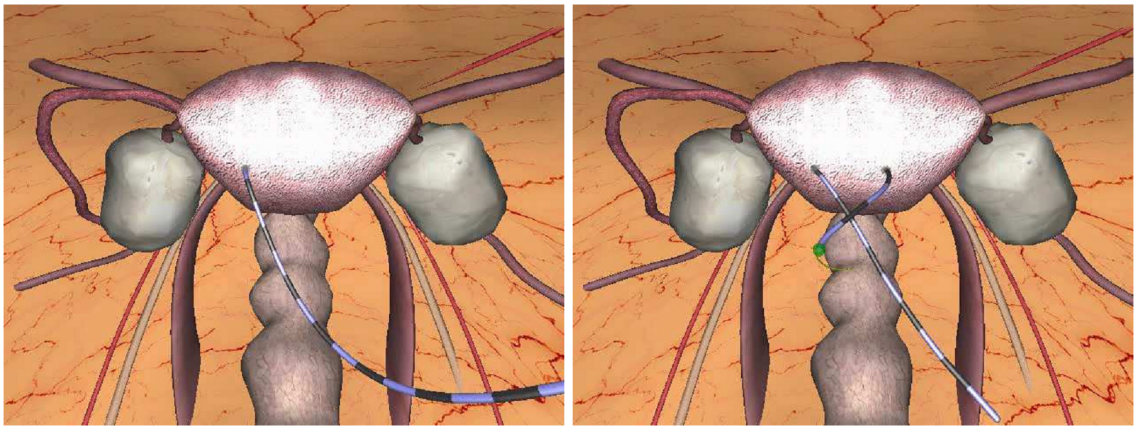


FIGURE 5.22 – Suture d'un organe dans la cavité abdominale par Lenoir *et al.* [LMGC04]).

5.2.2 Collision et autocollision du fil

Nous proposons de gérer les interactions de l'opération de suture avec un modèle basé sur les contraintes de complémentarité. La simulation du fil de suture amène des problématiques nouvelles à prendre en compte. En effet, sous l'effet seul de la gravité, le fil se pose sur les objets environnants et glisse sur eux lorsqu'il est tiré pour réaliser un point de suture. Cependant, on ne peut pas utiliser la détection de collision habituelle, puisque nous devons autoriser le fil à pénétrer les organes là où l'aiguille de suture a percé la membrane. Nous devons donc segmenter le fil de suture de telle façon à gérer séparemment les parties qui sont à l'intérieur des tissus et celles à l'extérieur pouvant entrer en contact avec les objets, y compris le fil lui-même.

La création des contraintes suit la méthode présentée au chapitre précédent (4.3), aussi bien au niveau de la pénétration de l'aiguille que son avancée dans les tissus. Nous différencions trois types de contraintes, à savoir le point d'entrée dans un organe, les contraintes qui sont placées dans le volume de l'objet, et enfin le dernier point de contrainte au niveau de la sortie de l'aiguille. Chaque contrainte fait le lien entre un point fixe dans le volume de l'objet et un segment de la courbe formée par l'association de l'aiguille et du fil de suture. Le fil pouvant glisser dans ces contraintes, nous recherchons à chaque pas de temps le segment correspondant à chacune. Une fois que cette mise à jour est effectuée, nous repèrons les segments qui correspondent aux contraintes d'entrée et de sortie. En parcourant alors la courbe depuis la fin du fil jusqu'à la pointe de l'aiguille, on peut repérer les parties qui sont plongées dans le volume de l'objet comme étant une liste de segments entre une contrainte d'entrée et une contrainte de sortie. Nous prenons également en compte le cas particulier où l'extrémité du fil est également à l'intérieur de l'organe.

Nous obtenons un vecteur de booléens qui donne pour chaque segment de la courbe la valeur *vrai* si celui-ci est dans l'objet, *faux* s'il est à l'air libre. L'état de chaque segment du fil est employé pour désactiver les tests concernant les parties de la courbe qui sont à l'intérieur de l'organe. Nous recherchons également les éventuelles autocollisions du fil de suture, en omettant de tester les voisins directs $(n + 1)$. Les noeuds dans le fil sont donc détectés et des contraintes de contact avec frottement sont créées. La détection de collision est effectuée avant la mise à jour des contraintes, puisqu'on crée des contraintes pour résoudre le contact

entre les objets. Les informations utilisées pour activer ou désactiver la collision de certains segments du fil de suture sont donc issues de l'état des contraintes au pas de temps précédent.

Filtrage de la collision

Afin d'être exhaustifs, nous devons aborder également une technique que nous utilisons en supplément à la détection de collision par distance minimale locale, à savoir le filtrage de points de contacts. En effet, la détection de collision réalise en dernier lieu une recherche de contacts potentiels en testant un grand nombre de paires de primitives. Lorsque la zone de contact est grande, on peut alors obtenir un nombre important de points de contacts, dont certains sont redondants. Par une analyse géométrique assez simple, il est possible de savoir quels contacts ne présentent pas une distance minimale, et donc de les rejeter, gardant ainsi les contacts les plus significatifs. La figure 5.23 montre un exemple sommaire de filtrage de points de contacts lors de la collision de deux objets solides.

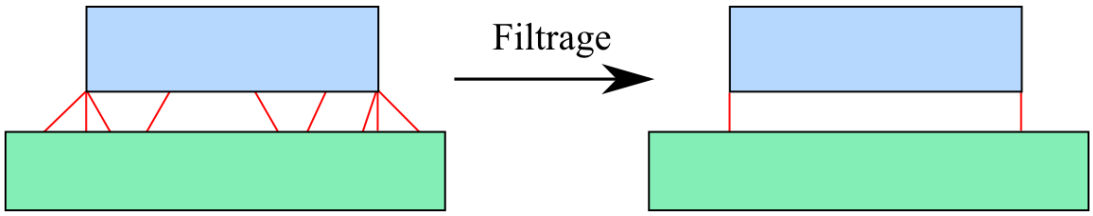


FIGURE 5.23 – Filtrage des points de contacts afin d'éliminer ceux qui sont redondants.

Le filtrage consiste à définir pour chaque primitive de collision (point, segment et triangle) une zone de l'espace dans laquelle les points de contacts seront conservés. En dehors de cette zone on peut les ignorer car on sait que les contacts potentiels sont forcement plus proches d'une autre primitive de l'objet. Nous allons ici détailler le filtrage dans le cas d'un objet 1D tel que le fil de suture.

Considérons un ensemble de points reliés par des segments rectilignes. Pour chaque point, nous construisons un cône de filtrage dont les bords sont perpendiculaires aux segments directement voisins de celui-ci. Ainsi un point de contact n'est gardé que s'il se trouve dans ce cône (zone b de la figure 5.24), à l'extérieur de la courbure formée par la courbe à cet endroit. Le noeud à l'extrémité de la courbe n'ayant qu'un seul segment voisin, on garde tout le demi-espace dans la direction opposée à ce segment (zone a de la même figure). Pour les segments de la courbe, on gardera un point de contact si sa projection orthogonale sur la ligne portant le segment est comprise entre les deux extrémités de celui-ci (zones c et d).

Dans ce cas simple, on peut voir que tout l'espace est couvert par l'union des cônes de filtrage, et on peut comprendre assez intuitivement la raison de celui-ci. Ainsi, prenons l'exemple d'une particule s'approchant de cet objet 1D (voir figure 5.25). Dans le premier cas, la particule se situe dans le cône de filtrage du segment, mais dans aucun de ceux des noeuds à ses extrémités. L'algorithme ne gardera donc que le point de contact détecté entre la particule et ce segment. On peut voir en effet que les distances entre la particule et les deux sommets du segment sont plus importantes que celle avec le segment. Dans le second cas la

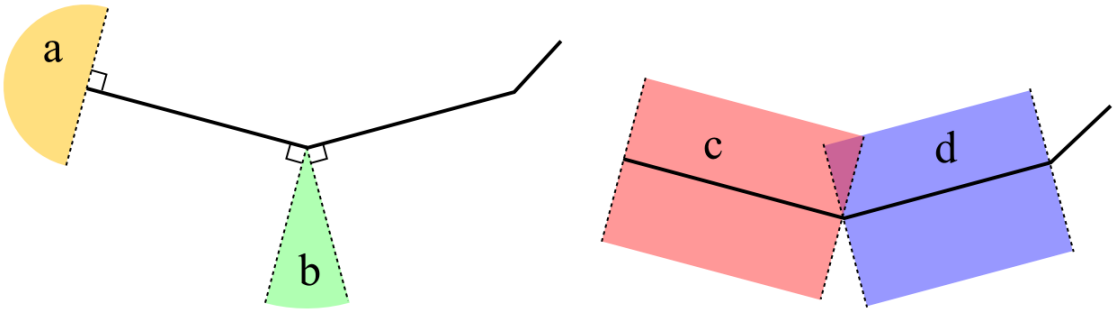


FIGURE 5.24 – Cônes de filtrage définis pour les points et segments d'un objet 1D.

particule se situe dans le cône de filtrage de chaque segment, mais pas dans celui du noeud reliant ceux-ci. On retiendra donc cette fois-ci deux points de contacts.

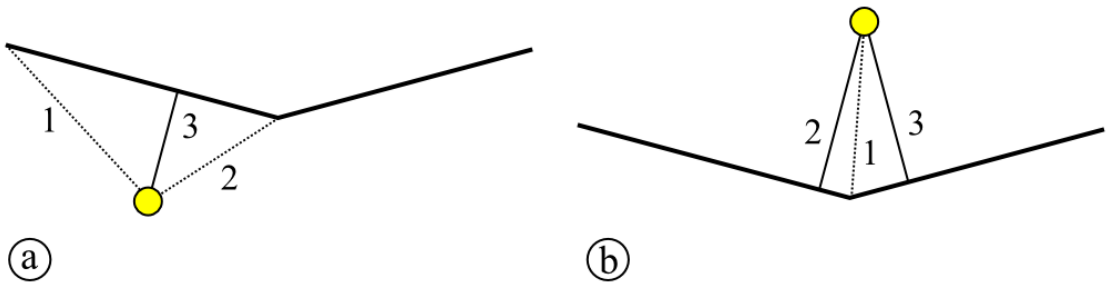


FIGURE 5.25 – Exemples de contacts filtrés dans le cas d'une particule approchant un objet 1D. En trait plein les contacts sont conservés, alors que ceux en pointillés sont filtrés car ne présentant pas la distance minimale ou sont redondants.

Pour l'aiguille et le fil de suture également, nous utilisons le filtrage de collision tel que nous l'avons présenté. Nous devons cependant désactiver la collision pour les segments qui se trouvent dans le volume des objets, ou pour la pointe de l'aiguille. Cependant, il est nécessaire de garder ces segments activés pour l'étape de filtrage, qui retire tous les contacts entre les noeuds du fil et la membrane lorsque celui est tiré à travers elle, tout en gardant correctement les contacts lorsque le fil glisse sur la surface de l'objet. Il en va de même pour la pointe de l'aiguille, puisque nous utilisons une détection de collision continue pour le point à l'extrémité, mais nous conservons le filtrage pour ne pas reporter simplement le contact sur le second noeud de la courbe. En pratique, nous ne changeons donc rien pour l'étape de filtrage, mais empêchons les contacts faisant intervenir un segment ou un noeud du fil marqué comme étant inactif (voir figure 5.26).

5.2.3 Contacts

Pour la simulation du geste de suture, nous devons traiter obligatoirement un nombre accru de contacts (voir figure 5.27). En effet, en plus de l'autocollision du fil devant permettre la réalisation de noeuds, les deux parties d'organe devant être suturées doivent être mises

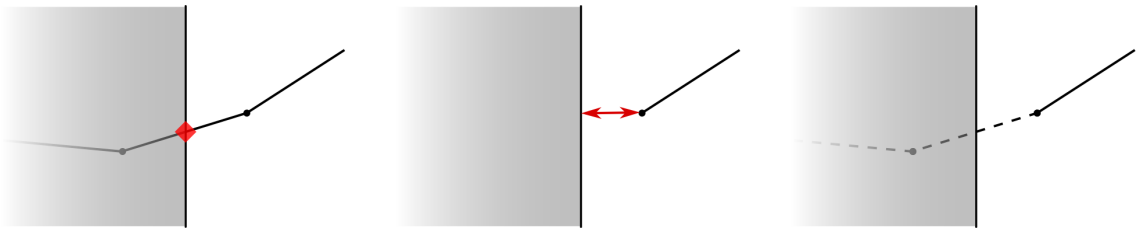


FIGURE 5.26 – Gestion des contacts lorsque des segments sont dans le volume de l'objet. (à gauche) Si l'on ne fait rien, des contacts seront détectés entre le fil et la surface de l'objet. (au milieu) Si on supprime simplement les 2 premiers segments, un contact sera détecté avec une extrémité du dernier segment. (à droite) On désactive donc les 2 premiers segments lors de la détection de collision, mais en gardant tous les segments pour l'étape de filtrage.

en contact. La zone de contact ainsi créée est surfacique et peut présenter des aspérités, augmentant de ce fait le nombre de points de contact potentiels. De même, le fil et l'aiguille ainsi que les autres instruments utilisés durant le geste chirurgical doivent pouvoir interagir entre eux et avec les organes. Dans tous les cas, les contacts doivent être modélisés avec précision pour rendre compte de la nature réelle de l'interaction, tout en restant compatible avec la simulation temps-réel. Nous employons la détection de collision par proximité pour tous les contacts éventuels de la scène, et utilisons le filtrage des points de contact afin de supprimer les informations redondantes. Chaque contact ainsi détecté et qui a été gardé après filtrage va donner lieu à la création d'une contrainte unilatérale de contact avec frottement (voir section 3.5.3).

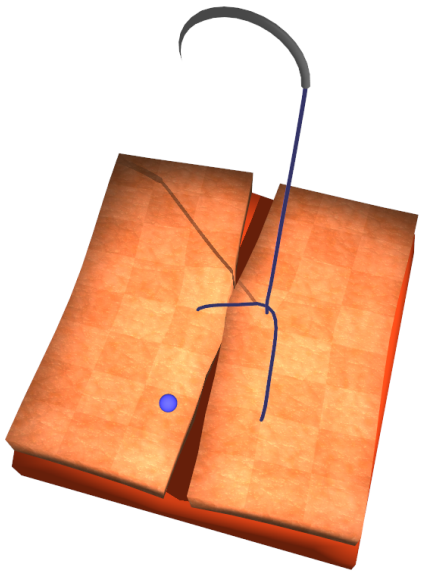


FIGURE 5.27 – Exemples de contacts intervenant durant la suture : contact entre les deux objets déformables à suturer, entre le fil de suture et ces objets, et également autocolision du fil.

5.2.4 Pénétration

L'utilisation d'une aiguille courbe dans le geste de suture rend un peu plus délicat son insertion par le praticien, la direction d'insertion devant être dans l'axe de la pointe de l'aiguille. De plus, la manipulation de l'aiguille par des pinces de laparoscopie complexifie encore le geste. Il est donc important de modéliser avec précision la pénétration de l'aiguille, qui ne doit pas être possible lorsque l'orientation de l'aiguille est mauvaise, ou que la force appliquée n'est pas adaptée.

Dans le geste chirurgical de suture, il y a plusieurs pénétrations successives de l'aiguille dès lors qu'il s'agit de réaliser plusieurs points de suture. Lorsque l'on s'intéressait à l'insertion d'aiguille, nous avions posé l'hypothèse que celle-ci n'entre pas à deux reprises dans le même objet. En effet, il fallait d'abord retirer entièrement l'aiguille avant que l'on active à nouveau la détection de collision entre cet objet et la pointe de l'aiguille. Nous avons donc modifié légèrement le modèle d'interaction dans le cas de la suture afin que l'aiguille puisse sortir d'un objet et y être réintroduite plus loin. En pratique, nous testons les collisions éventuelles entre la pointe de l'aiguille et tous les objets déformables qui ont été marqués comme pouvant être suturés. Lorsque la pointe de l'aiguille est dans le volume d'un objet, nous désactivons temporairement la détection de collision, jusqu'à ce que l'on observe lors de la phase de création de contrainte que la pointe est sorti de l'organe (voir section 4.3).

Lorsqu'un point de contact est détecté entre la pointe de l'aiguille et une membrane, nous utilisons la contrainte de pénétration déjà présentée au chapitre précédent (4.1.3). L'ajout du fil de suture après l'aiguille ne change en rien le traitement de l'insertion de celle-ci, puisqu'au niveau du modèle d'interaction nous ne différencions pas pour le moment les segments de la courbe faisant partie de l'aiguille et ceux composant le fil de suture. En effet, c'est uniquement au niveau du modèle mécanique que nous les avons séparés. Pour certaines interventions, le fil peut être noué à l'aiguille et on peut observer que le passage de celui-ci au travers d'une membrane offre une certaine résistance. Nous ne proposons pas dans l'état de simuler ce phénomène, qui nécessiterait de créer une seconde contrainte de pénétration au niveau de l'attache du fil. Une seconde détection de collision serait par contre inutile pour savoir lorsque l'on passe, au niveau de la membrane de l'organe, de l'aiguille au fil. On peut en effet se servir de l'abscisse curviligne de la première contrainte de pénétration qui reste au niveau de la membrane pour calculer la distance de ce point par rapport à la surface de l'organe.

Il faut noter que comme pour l'insertion d'aiguille simple, nous sommes capables de simuler plusieurs couches de tissus, pour rendre compte par exemple de l'hétérogénéité de la peau composée du derme et de l'épiderme. Lorsque deux couches de tissus sont soudées entre elles, nous voulons éviter de créer à l'interface des contraintes redondantes, et ne créons qu'une contrainte de pénétration sur la surface faisant l'interface. Nous distinguons cependant le cas où une membrane interne sépare deux couches dans un organe, de celui-ci où deux surfaces externes sont en collision. Dans ce dernier cas, qui apparait lorsque nous rapprochons les deux parois d'une plaie durant la suture, nous créons bien une contrainte de sortie sur l'une, et une contrainte de pénétration sur l'autre.

5.2.5 Frottement

Lors de leur déplacement à l'intérieur des tissus, l'aiguille ainsi que le fil de suture sont ralentis par le frottement exercé par ceux-ci. Nous utilisons le même procédé que lors de l'insertion d'aiguille pour générer dynamiquement des contraintes le long du chemin créé par la pointe de l'aiguille. Les contraintes de chemin modélisent le frottement dans la direction tangentielle au mouvement au travers de la loi de Coulomb. Chaque contrainte peut ainsi avoir deux états : adhérence ou glissement.

En intégrant dans notre approche le frottement, nous pouvons simuler des scénarios réalistes. En effet, il est bien connu des chirurgiens qu'un fil de suture tressé présente un frottement bien plus important qu'un monofilament, ce qui se traduit dans notre modèle par un coefficient de frottement μ différent. Les praticiens choisissent le type de fil en fonction des tissus concernés, ainsi que de l'amplitude de la force qui va s'opposer à la suture et potentiellement réouvrir la plaie. Nous ne pourrions donc pas imaginer de simulation de suture sans prendre en compte le frottement, puisque c'est le frottement entre le fil de suture et les tissus que celui-ci traverse qui va maintenir ensemble les deux parois de la plaie jusqu'à cicatrisation. La figure 5.28 illustre dans une simulation le rôle du frottement dans une opération de suture.

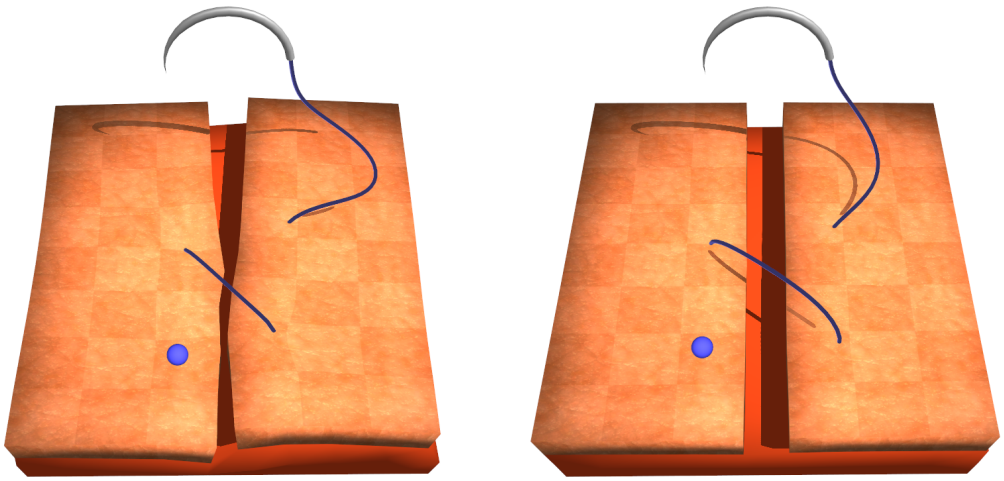


FIGURE 5.28 – Mise en évidence de l'importance du frottement pour la suture. (à gauche) Le frottement statique maintient le point de suture fermé même lorsqu'on ne tire plus sur le fil. (à droite) Avec un coefficient de frottement plus faible, on observe que le fil de suture glisse lentement dans les tissus, jusqu'à équilibrer les forces qui s'opposent à la fermeture de la plaie.

Le coefficient de frottement entre le fil de suture et les tissus est un paramètre de notre modèle d'interaction, qui peut être défini différement pour chaque organe. Un test simple permet d'appliquer un autre coefficient de frottement pour les contraintes qui sont encore le long de l'aiguille, mais ce dernier a une importance moindre car il n'agit que lors du passage de l'aiguille de suture dans un tissu.

Lors de la simulation d'insertion d'aiguille, nous avions toujours une aiguille assez rectiligne. Lors du traitement des contraintes de chemin, nous cherchions le segment de l'aiguille sur lequel se projetait orthogonalement le point de contrainte. Il faut noter cependant que dans la simulation de suture nous sommes en présence à la fois d'une aiguille souvent courbe, et d'un fil qui à l'intérieur des tissus forme des arcs de cercle. Nous devons donc modifier le traitement de ces contraintes, pour ne pas projeter arbitrairement le point de contrainte sur un segment de la courbe qui en est trop éloigné. Pour trouver à quel segment de l'ensemble aiguille et fil de suture va être mis en relation une contrainte de chemin, nous partons du segment qui avait été trouvé au pas de temps précédent. Nous testons alors les segments voisins dans les deux directions, puis les voisins des voisins, et ainsi de suite. Nous privilégions ainsi le segment de la courbe qui est le plus proche de l'ancien segment, plutôt que de sélectionner celui qui est le plus proche du point de contrainte (voir figure 5.29).

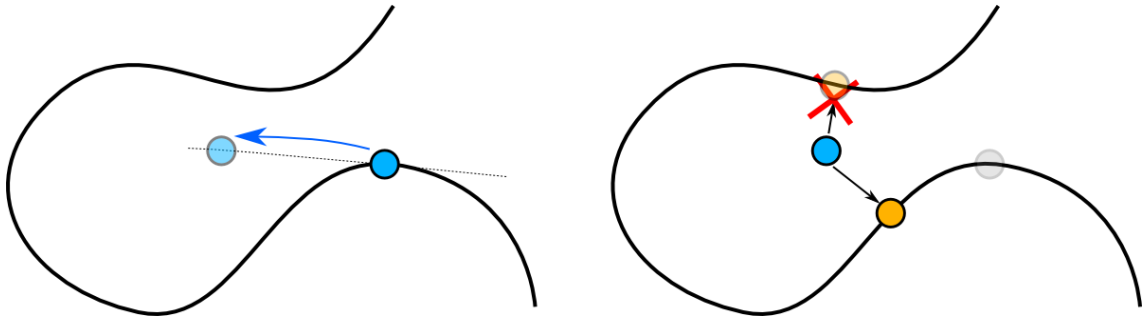


FIGURE 5.29 – Recherche de segment dans une courbe. Dans certaines configurations, le point de contrainte peut se projeter orthogonalement sur plusieurs segments du fil. Nous sélectionnons celui dont la distance le long de la courbe, par rapport au segment choisi au pas de temps précédent, est la plus courte.

L'approche que nous utilisons discrétise le fil en une série de segment rectilignes. Les contraintes de point glissant peuvent donc dans certains cas présenter un comportement où le fil glisse par à-coups. Des tests ont été réalisés pour réaliser une interpolation de la tangente au fil à la position des contraintes, tests concluants pour un fil statique mais peu efficaces dans la simulation de la suture. Un nouveau modèle déformable 1D est actuellement en développement pour permettre entre autres d'assurer une certaine continuité.

5.2.6 Préhension de l'aiguille par l'utilisateur

Pour nos premiers tests, nous avons fait subir à l'aiguille de suture un mouvement prédéfini, en définissant à intervalle régulier position et rotation désirées. Nous visons à ce que l'utilisateur puisse réaliser lui même le geste de suture, en interagissant avec l'aiguille et le fil. Pour cela, il sera au contrôle de pinces de laparoscopie virtuelles (via des périphériques haptiques adaptés) et devra pouvoir manipuler l'aiguille en l'attrapant avec celles-ci.

L'idéal du point de vue du réalisme de la simulation serait de gérer la préhension de l'aiguille par les pinces avec le seul modèle de contact présenté dans ce travail. Des collisions seraient détectées entre les mâchoires des pinces et le fil de suture ou l'aiguille, des contraintes

de contact avec frottement étant alors créées. Des preuves de concept ont déjà été réalisées avec le framework *SOFA*, pour la manipulation par des pinces de laparoscopie d'objets déformables. Pour que la manipulation soit efficace, il faudrait que l'objet attrapé puisse se déformer légèrement. En effet, bien qu'il soit possible de saisir des objets parfaitement rigides, on arrive rapidement à une situation sur-contrainte, engendrant une convergence plus lente de l'algorithme de résolution et ainsi une prise moins stable.

Pour transposer cette méthode au fil de suture, nous rencontrons deux problèmes. Tout d'abord, le fil a été jusqu'à présent représenté du point de vue de la collision comme un simple objet 1D. Cela peut engendrer des difficultés pour l'algorithme de détection de collision lors de l'étape de filtrage des contacts redondants. Il est envisageable alors d'entourer le fil de suture d'une enveloppe cylindrique pour avoir un modèle de collision 3D, comme nous le faisons pour la visualisation. Cependant, cela surchargerait la détection de collision pour une application qui n'est pas encore très utile en pratique.

De plus, il est nécessaire de créer au minimum quatre contraintes de contact avec frottement pour gérer correctement la préhension par des pinces. Le fil étant très rigide, ces contraintes sont fortement interdépendantes, ce qui allourdit la résolution des contraintes. Nous avons donc à la fois des difficultés à attraper le fil de suture avec les pinces de laparoscopie, à le maintenir dans les mâchoires de celles-ci, et l'interaction obtenue est peu stable car très sensible aux plus petites approximations dans la résolution des contraintes (la simulation de suture faisant intervenir un grand nombre de contraintes, il arrive fréquement d'arrêter le Gauss-Seidel avant convergence).

Nous proposons donc mettre en place une contrainte 6D afin de gérer l'interaction entre les mâchoires des pinces de laparoscopie et l'aiguille ou le fil de suture. Cette contrainte fixera à la fois la position et la rotation des deux modèles entre eux. Nous réduirons ainsi par l'utilisation de cette contrainte la complexité de la simulation, tout en garantissant une préhension parfaite du fil. Cependant nous devons alors gérer d'un côté la préhension du fil de suture, et de l'autre côté la possibilité d'attraper tous les autres objets présents dans la simulation. De plus nous considérons en première approche qu'il n'y a pas de frottement entre les mâchoires de la pince et l'objet attrapé. En effet, bien que le frottement pour la contrainte de contact soit bien maîtrisée, des études supplémentaires sont requises avant d'envisager prendre en compte le frottement du point de vue de la rotation autour de son axe du fil maintenu par la pince. La figure 5.30 présente l'utilisation de cette contrainte dans le cadre d'un geste de suture au sein d'un simulateur laparoscopique.

L'utilisateur manipulant les pinces doit pouvoir interagir selon ses besoins avec le fil et l'aiguille ; nous allons donc placer dynamiquement la contrainte suite à ses actions. Lorsque l'angle entre les mâchoires d'une pince de laparoscopie devient faible, nous regardons la configuration respective de cette pince et des aiguilles et fils de suture présents dans la scène afin de décider si une contrainte sera créée ou non. Nous voulons pour cela vérifier qu'un segment du fil se trouve bien entre les mâchoires de la pince. Lors de la création de la scène, nous prenons soin de définir un segment représentant la zone d'effet des mâchoires lorsque celles-ci se referment. On recherche alors le segment du fil qui correspond le mieux, par le calcul d'une distance segment-segment qui nous donne également les deux points (l'un sur le fil, l'autre entre les mâchoires) étant les plus proches. Afin que la contrainte soit créée, il faut que le point obtenu pour la pince soit bien compris entre les bornes que l'on a défini pour les mâchoires. Nous allons alors contraindre les positions des deux points obtenus telle une contrainte de

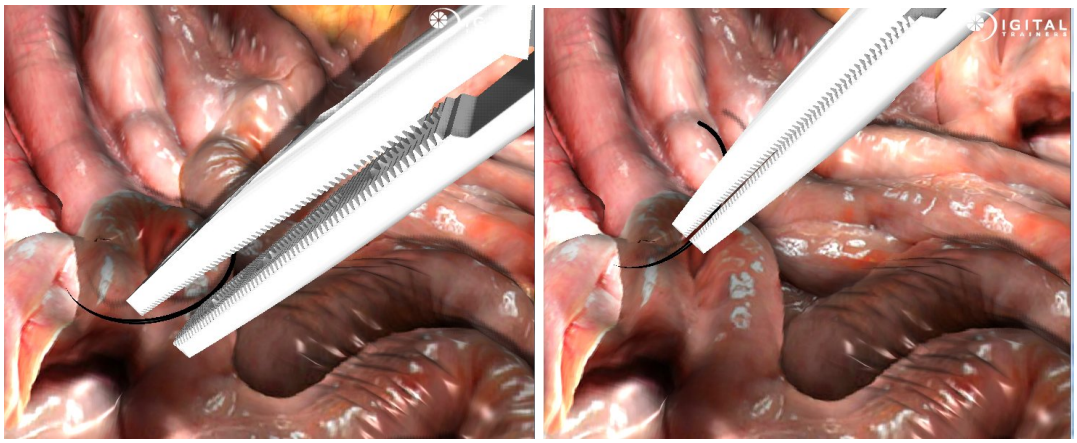


FIGURE 5.30 – Utilisation d'une contrainte 6D pour manipuler aiguille et fil de suture à l'aide de pinces laparoscopiques.

point fixe. Afin de faciliter la préhension, nous autorisons une légère distance entre le fil de suture et l'axe centrale de la pince ; la contrainte ramène l'un sur l'autre lorsque les mâchoires se ferment.

Les mâchoires de la pince, lorsqu'elles sont fermées, forment un plan sur lequel le fil doit être contraint. Nous devons donc contraindre le fil à se placer dans ce plan. Nous enregistrons tout d'abord la rotation initiale relative entre le fil et la pince au moment de la préhension et ajoutons à cette rotation initiale une seconde rotation correspondant à l'angle entre l'axe du fil et ce plan. Afin d'éviter des comportements irréalistes, nous ne permettons pas la préhension du fil si celui-ci est perpendiculaire aux machoires de la pince.

La contrainte ainsi créée fixe la position et la rotation d'un point sur le fil à un point entre les mâchoires de la pince. Pour la résoudre nous faisons appel à une contrainte bilatérales à 6 dimensions, mais appliquons des tolérances différentes pour la rotation et la translation. En effet, nous calculons une erreur de résolution des contraintes équivalente à une distance, ce qui n'a pas de sens pour la contrainte fixant la rotation. Afin d'avoir un résultat probant, il a fallu appliquer pour cette contrainte une tolérance 100 fois plus faible que pour la contrainte en position.

5.3 Simulation d'un geste simple de suture

Nous voulons illustrer notre modèle d'interaction dans le cadre de la simulation du geste chirurgical de suture en mettant en place une scène assez simple reproduisant l'apprentissage de la réalisation de points de suture.

Nous disposons un tissu déformable composé de deux couches, dont celle supérieure un peu plus rigide (illustrant par exemple la peau) est séparée en deux moitiés. L'objectif est de réaliser une suture joignant les deux parties de la peau. Cette simulation réunit tous les types de contraintes que nous avons présentés, et illustre la flexibilité et l'efficacité de notre approche. En particulier, nous montrons que plusieurs points de suture virtuels peuvent être

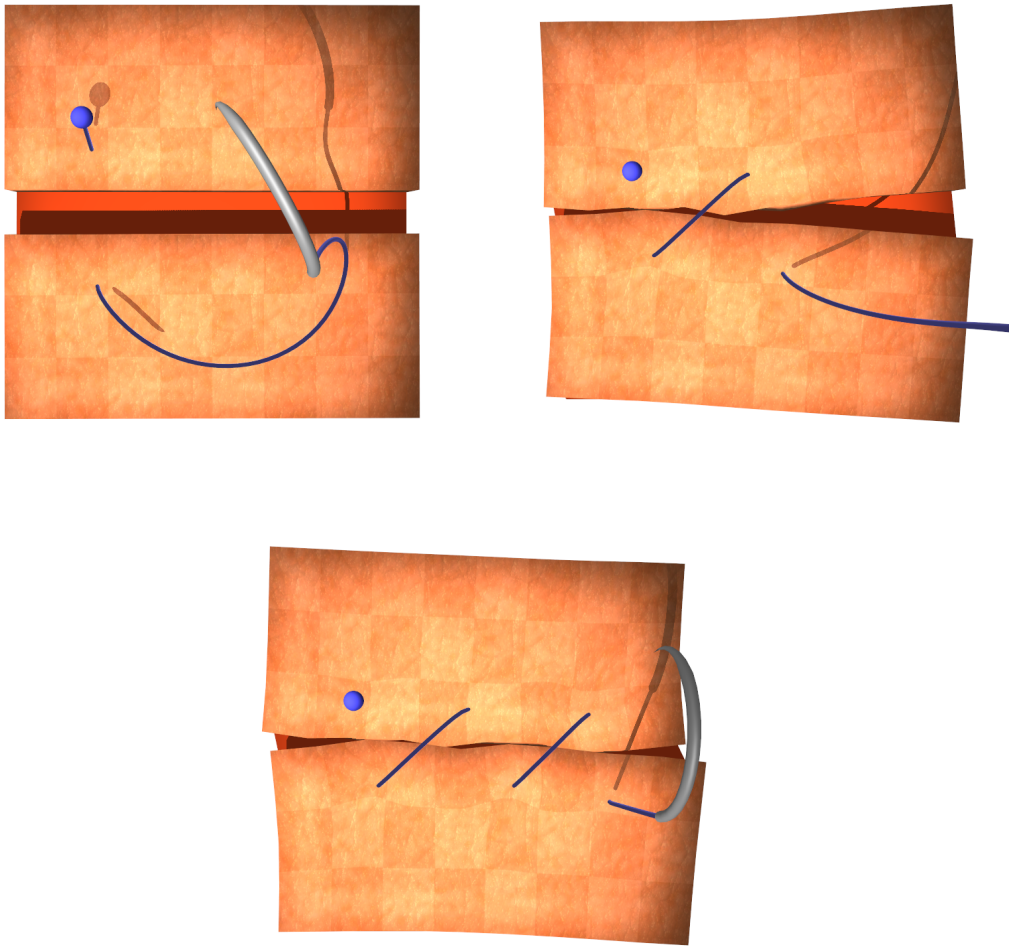


FIGURE 5.31 – Simulation d'un geste basique de suture : trois points de suture sont réalisés et serrés pour fermer une plaie.

simulés, et que l'on peut tirer sur le fil de suture afin de serrer les points de suture, tout en prenant en compte les contacts entre les deux parties de la plaie. Nous gérons également les autocollisions du fil ainsi que les contacts qui interviennent entre celui-ci et la surface des tissus déformables. Enfin, à travers le calcul du frottement statique le long du fil de suture plongé dans les tissus, nous simulons le fait que la suture tient lorsque le fil est relâché.

Pour cette simulation nous utilisons un modèle déformable à éléments finis héxahédraux avec co-rotationnel, afin de gérer les non-linéarités géométriques de la déformation. L'emploi d'éléments héxahédraux est justifiée ici car ils sont en général plus stables lorsque nous avons d'importantes déformations locales, comme c'est le cas durant un geste de suture.

Dans un soucis de simplicité pour mettre en place la simulation, l'aiguille n'est pas ici dirigée par l'utilisateur. C'est un mouvement prédéfini qui est appliqué à la base de l'aiguille de suture, et qui permet ainsi de reproduire exactement le même geste d'une simulation à l'autre,

nous permettant alors d'observer l'impact des paramètres de notre modèle d'interaction sur le résultat de la simulation.

	Contacts	Frottement	Suture
Mouvement libre Fil	1.58	1.57	2.02
Mouvement libre Tissus	8.30	8.25	8.28
Collision	1.86	2.46	2.97
Calcul de la compliance Fil	14.77	15.01	22.43
Calcul de la compliance Tissus	0.48	0.48	0.96
Résolution du Gauss-Seidel	1.19	1.37	2.14
Application des forces des contraintes	2.18	1.83	2.30
TOTAL	30.35	30.96	41.10
(FPS)	32.95	32.30	24.33

TABLE 5.1 – Mesure de temps de calcul moyens en ms enregistrés sur plusieurs exemples, pour chaque étape clé de notre approche. Chaque colonne corresponds à une simulation différente : Contacts (représentée à la figure 5.27), Frottement (figure 5.28.b) et Suture (figure 5.31).

Le tableau 5.3 présente les mesures de performances de notre méthode, obtenue en chronométrant les étapes clés de l'algorithme, sur des simulations de plus en plus complexes. Les simulations ont été réalisées sur un Core2Duo 2,66GHz avec 2Go de RAM. L'aiguille et le fil sont représentés par un seul modèle composé de 100 éléments de poutres et le modèle déformables est constitué de 300 héxahédrons.

On peut remarquer sur les mesures que globalement, le calcul de la compliance et la résolution des contraintes augmentent avec le nombre de contraintes présentes dans la scène. Le goulot d'étranglement actuel est le calcul de la compliance du fil, que nous comptons améliorer en résolvant les contraintes sans construire la compliance. Nous obtenons toujours une simulation interactive en réalisant jusqu'à trois points de suture, au-delà de quoi la manipulation commence à saccader.

5.4 Conclusion

Nous avons montré dans ce chapitre qu'il est possible d'utiliser nos contraintes de complémentarité pour simuler une variété d'interactions ayant lieu durant une tâche de suture. Pour assurer des comportements réalistes durant la simulation, nous proposons également d'utiliser un unique modèle pour représenter à la fois l'aiguille et le fil de suture. Ce modèle déformable basé sur la théorie des poutres utilisé pour cet assemblage permet en effet de représenter aussi bien la quasi-rigidité de l'aiguille que la grande flexibilité du fil.

Nous simulons actuellement toutes les étapes d'un geste de suture mis à part la réalisation de noeuds dans le fil de suture. Cette limitation n'est pas causée par le modèle d'interaction mais par l'implantation actuelle du modèle de fil chirurgical. Celui-ci utilise des éléments de longueur constante, et représenter efficacement un noeud nécessiterait alors un très grand nombre d'éléments. Nous prévoyons d'introduire une adaptation dynamique de la discrétisa-

tion du fil basé sur la coubure locale, afin de pouvoir simuler la réalisation de noeuds ainsi que d'autres interactions sans demander un nombre d'éléments trop important. Enfin, une autre conséquence de la discrétisation du fil est visible lors de l'autocollision du fil. En effet, lorsque le fil glisse sur lui-même on peut observer des saccades dans le mouvement, car tous les segments de la courbe sont rectilignes. Le fil saute donc d'un segment à l'autre car ne pouvant pas se déformer localement au niveau d'un seul segment. On peut bien évidemmement réduire ces saccades en augmentant la discrétisation, ce qui alourdit alors le calcul des déformations du fil, mais aussi celui de sa compliance. Pour supprimer totatement ce phénomène il faudrait que la contrainte de contact utilisée pour l'autocollision du fil utilise une réprésentation implicite de celui-ci, comme l'ont fait Dequidt *et al.* [DLC07] pour les contacts entre un catheter et les vaisseaux sanguins.

Nous avons présenté une simulation du geste basique de suture sur une plaque d'entraînement, avec réalisation de trois points de suture. Des optimisations sont nécessaires, au niveau du calcul de la compliance du fil ainsi que de la résolution des contraintes, si l'on veut s'intéresser à des simulations plus complètes de la suture, avec plusieurs points de suture ou réalisation de noeuds.

Ce travail sur la suture a fait l'objet d'un poster à SCA'09 : *Suturing Simulation Based on Complementarity Constraints* [GDC+09], et nous allons soumettre un article de journal pour pouvoir présenter la méthode plus en détails. J'ai réalisé un transfert de technologies pour intégrer le module de suture à un simulateur de laparoscopie auprès de l'entreprise Digital-Trainers. Je continuerai là bas à faire évoluer mon modèle d'interaction basé sur les contraintes de complémentarité, afin de permettre à court terme de réaliser des sutures n'importe où dans une cavité abdominale, d'offrir un retour d'effort à l'utilisateur et d'optimiser la création des contraintes en faisant des simplifications quand cela est possible pour réaliser le geste de la suture en entier, plusieurs points de suture et des noeuds. En collaboration avec l'équipe de recherche Shaman[8], j'espère à plus long terme pouvoir faire évoluer la résolution du Gauss-Seidel et le calcul de la compliance.

8. http://www.inria.fr/recherche/equipes/shaman.fr.html

Chapitre 6

Conclusion

Nous avons présenté dans ce travail une méthode générique pour traiter efficacement des contraintes de types différents dans des simulations temps réel mettant en œuvre aussi bien des objets rigides que des modèles déformables. On a montré que l'on peut unifier le traitement des contraintes, en séparant leur traitement spécifique dans un objet logiciel extérieur au solveur des contraintes, qui peut alors se concentrer sur leurs paramètres communs afin de rester générique.

La linéarisation d'une fonction de "mapping" permet de poser le problème contraint dans l'espace des contraintes, indépendamment du formalisme choisi par les modèles simulés, nous permettant ainsi de faire interagir des modèles déformables (aussi bien 1d que 3d) avec des objets rigides. L'utilisation conjointe d'une matrice de compliance mise à jour à chaque pas de temps avec une intégration implicite offre une grande précision. L'algorithme du Gauss-Seidel que nous avons choisi pour coupler mécaniquement les contraintes au travers de leur compliance converge en pratique dès lors que les contraintes sont définies avec soin (pas de contraintes antagonistes, et système non surcontraint). Bien que cet algorithme présente une vitesse de convergence assez lente, nous avons montré qu'il est possible de gagner quelques itérations dans la résolution en appliquant quelques techniques d'accélération simple à mettre en œuvre. Les optimisations apportées récemment au calcul de la compliance des modèles déformables nous a permis de réaliser des simulations interactives. L'approche des contraintes généralisées a fait l'objet d'une publication à VRIPHYS'08 : *Unified Processing of Constraints for Interactive Simulation* [GDG08]. Son implantation a été réalisée dans le framework libre *SOFA* [9], qui nous permet d'utiliser certaines fonctionnalités déjà existantes (modèles déformables, détection de collision, intégration numérique, fonctions de "mapping").

La méthode présentée a été utilisée pour le prototypage rapide de nouvelles contraintes, qui ont ensuite été appliquées à la simulation d'insertion d'aiguilles flexibles. Plusieurs types de contraintes ont ainsi été créés en plus des contraintes de base (unilatérales de contact, bilatérales, et frottement), telles que des contraintes glissantes sur une structure 1d, de pénétration de l'aiguille au niveau de la membrane de l'organe, de découpe, de chemin, etc. Ces contraintes ont été utilisées dans la modélisation du geste de l'insertion d'aiguille, rigide ou flexible, biseautée ou non. Nous nous sommes basés sur les mesures et les lois expérimentales

9. http://www.sofa-framework.org

qui sont reconnues dans ce domaine de simulation afin de paramétrer nos contraintes, puis les valider en reproduisant les résultats des expériences des travaux antérieurs. L'originalité de notre approche est de proposer un modèle d'interaction basé sur des contraintes de complémentarité là où les travaux passés étaient principalement focalisés sur des approches en force. Nous pensons que notre modèle d'interaction est très complet, reproduisant tous les phénomènes observés, et offre de nombreux avantages tels que la simulation temps réel, la déformation des organes et de l'aiguille tout en étant assez générique pour être utilisable dans un grand nombre de simulations. Nous gérons ainsi l'hétérogénéité des organes, l'insertion de multiples aiguilles simultanément, le retrait partiel de l'aiguille pour son repositionnement, la traversée de plusieurs membranes ou régions présentant des paramètres physiques variés. Nous avons enfin réalisé des tests pour prendre en compte la déformation interne des matériaux qui peuvent compresser plus ou moins l'aiguille et faire ainsi varier le frottement mesuré tout le long de son corps. Ce modèle d'interaction a été présenté et publié à MICCAI'09 : *Interactive Simulation of Flexible Needle Insertions Based on Constraint Models* [DGM+09].

Cette méthode d'interaction a permis une simulation interactive du geste de suture, grâce aux travaux entrepris sur l'insertion d'aiguille. Les choix qui ont été fait très tôt dans la méthode de gestion des contraintes nous ont permis de réaliser des simulations ne nécessitant pas de remaillage tout en gardant une grande précision dans le positionnement des contraintes. Nous avons utilisé pour la déformation du fil de suture ainsi que pour l'aiguille un modèle déformable 1D précédemment validé mais qui n'avait été utilisé que pour la simulation de catheter. Nous avons pu réaliser des simulations réalistes et interactives comprenant plusieurs points de suture, où tous les objets en présence interagissent entre eux. Les résultats obtenus sont encourageants et valident la méthode de définition et de résolution des contraintes que nous avons proposée. Les travaux sur la simulation de suture ont été présentés sous la forme d'un poster à SCA'09 : *Suturing Simulation Based on Complementarity Constraints* [GDC+09].

L'application des contraintes de complémentarité à la simulation interactive de suture fait actuellement l'objet d'un transfert de technologie auprès de l'entreprise *Digital-Trainers*[10] spécialisée dans la réalisation de simulateurs médicaux. Ceux-ci sont réalisés en collaboration étroite avec l'*IRCAD*[11] qui apporte son expertise médicale, et se basent pour la simulation physique sur le framework *SOFA*, le même qui a été utilisé pour développer les méthodes proposées dans ce travail.

Je continue chez Digital-Trainers mon travail de thèse, en incorporant le geste de suture à un simulateur de laparoscopie, proposant à fin d'entraînement les différentes tâches élémentaires spécifiques à ces opérations. L'interaction entre les pinces à retour d'effort et l'aiguille est perfectionnée, et nous étudions la réalisation de nœuds dans le fil de suture.

Ayant affaire à un simulateur commercial, les critères de temps de calcul et de vraisemblance deviennent plus importants que l'exactitude des modèles utilisés. Il devient donc nécessaire de parfaire les optimisations proposées, et de s'assurer de la stabilité de la méthode quelque que soit les cas d'utilisations.

Ce transfert de technologie et cette application concrète sont la meilleure façon possible de terminer mon travail de thèse, lui donnant tout son sens.

10. http://www.digital-trainers.com
11. http://www.ircad.fr/

De nombreuses pistes peuvent maintenant être explorées à partir de la méthode présentée. Nous désirons à court terme pouvoir manipuler les objets de la scène via des interfaces à retour d'effort, et intégrer à notre simulation de suture la possibilité de réaliser des nœuds dans le fil chirurgical. La prise en compte de la pression latérale le long de l'aiguille ou du fil est une problématique que nous avons soulevée lors de la simulation d'insertion d'aiguille, mais que nous complèterons peut être pour un autre simulateur. Nous envisageons à plus long terme d'optimiser la simulation de suture en analysant dynamiquement la scène pour interchanger plusieurs descriptions mécaniques selon les besoins en réalisme et le temps de calcul disponible. Enfin, nous voulons continuer à améliorer l'algorithme du Gauss-Seidel que nous utilisons pour résoudre les contraintes, en intégrant au fur et à mesure des optimisations portant sur plusieurs points.

Dans un simulateur médical, l'interaction avec l'aiguille se fera de préférence à l'aide de périphériques adaptés, proposant aux utilisateurs une expérience proche de la vraie opération. Dans notre cas, nous pouvons par exemple imaginer d'utiliser des pinces de laparoscopie afin de manipuler l'aiguille. Il est ensuite intéressant de proposer un retour haptique à l'utilisateur. Certains travaux récents ([MPD06],[ZGH+07]) ont ainsi présenté des méthodes pour ajouter un tel retour haptique dans des simulateurs de suture.

La sensation haptique est temporellement beaucoup plus sensible que la vue. En effet, on considère que là où une trentaine d'images par secondes suffisent pour donner l'illusion, il faudra mettre à jour le retour d'effort à raison de 1000 Hz environ. Pour atteindre cette vitesse, on peut soit optimiser l'ensemble de la scène et choisir des modèles simplifiés ou séparer une boucle de simulation spécialement pour l'haptique.

En première approche, on peut faire appel à un *god-object* : l'interface haptique contrôle un objet virtuel qui n'interagit pas avec la scène. Un ressort 6d est alors fixé entre cet objet virtuel et le modèle de l'aiguille ; les forces mesurées sur le ressort étant ensuite répercutées à l'utilisateur. L'aiguille est simulée dans la boucle de simulation de la scène, mais le *god-object* et le ressort sont mis à jour dans une seconde boucle tournant à 1000 Hz. Avec cette méthode, seules les interactions intervenant durant la boucle principale de simulation sont prises en compte.

Une méthode plus réaliste, proposée par Saupin *et al.* [SDC08], consiste à résoudre correctement les contraintes, dont leur résolution est répartie dans deux boucles de simulation tournant à des fréquences différents (environ 50Hz pour la simulation globale et 1000Hz pour l'haptique).

On pourra également essayer de modéliser de manière plus réaliste la préhension de l'aiguille par les pinces de laparoscopie. L'utilisateur contrôle les pinces, dont les machoires peuvent se refermer sur un objet. Une collision est alors détectée et des contraintes de contact avec frottement gèrent l'interaction, permettant ainsi à l'utilisateur de pouvoir manipuler via les pinces absolument tous les objets de la scène. Puisque nous utilisons des contraintes de contact plutôt qu'une contrainte de point fixe, les objets attrapés par les pinces peuvent glisser en dehors des pinces s'ils sont tirés assez fort. On reproduit alors un comportement courant observé en opération.

Dans les simulations de suture que nous avons présentées à la section 5.3, nous n'avons pas étudié la faisabilité de nouer le fil une fois que les points de sutures sont réalisés. En effet, nous n'avions alors pas de manière simple de manipuler l'aiguille et le fil nous permettant de réaliser ce geste. Dès lors qu'il est possible d'utiliser un dispositif physique tel que des pinces de laparoscopie pour interagir avec l'aiguille, on peut s'intéresser à nouveau à la réalisation

de nœuds.

Pour faire un nœud, la détection de collision par proximité que nous utilisons convient parfaitement, le contact avec frottement pouvant être géré par des contraintes. Cependant, le modèle non adaptatif que nous utilisons actuellement pour représenter le fil de suture nécessiterait une discrétisation très fine tout le long du fil pour pouvoir réaliser un nœud. Cela serait particulièrement inefficace au niveau des performances puisque le temps de calcul de la compliance du fil (qui est jusqu'à présent la phase la plus longue de notre simulation) augmente en même temps que la discrétisation.

Il est alors intéressant d'utiliser un modèle de fil adaptatif : de nouveaux éléments de poutre sont insérés le long du fil selon un paramètre de courbure locale. La masse est répartie sur les nouveaux nœuds selon leur abscisse curviligne, les mêmes paramètres physiques étant appliqués aux éléments insérés. Lorsque pendant plusieurs pas de temps deux éléments consécutifs sont alignés (à une petite erreur ϵ près), ils sont fusionnés et remplacés par un seul élément. La méthode est efficace pour le raffinement, il subsiste encore de petites oscillations introduites lorsque deux éléments de poutre sont fusionnés. Pour éviter de devoir ajouter de l'amortissement, il faut améliorer le critère de supression afin d'être sûr que les vitesses relatives des éléments sont très faibles. (la suppression d'un élément lorsque la poutre oscille amplifie le mouvement).

Ce modèle, qui a été proposé pour la simulation d'un endoscope, doit encore être testé dans la simulation du geste de suture. Si sa stabilité est prouvée dans cette application, il sera alors mis en œuvre pour la réalisation de nœuds dans le fil de suture.

Nous avons réalisé des tests préliminaires (voir section 4.2.3) pour mettre en évidence la contrainte interne aux éléments des objets déformables lorsque ceux-ci sont soumis à des forces extérieures. Durant ce travail, nous n'avons pas pu aller plus loin et réellement prendre en compte cette pression dans le calcul du frottement.

Il serait envisageable de continuer à étudier ce point, en réalisant des tests de telle manière à mesurer expérimentalement l'influence de cette pression sur le frottement mesurable au niveau du corps d'une aiguille ou d'un fil chirurgical. Une fois ces données obtenues, il devrait être possible de finaliser le modèle d'interaction, puis de le valider.

Cette contribution à la force de frottement n'est surement pas indispensable pour une simulation réaliste de l'insertion d'aiguille, dans le sens où la pression exercée par les autres organes sur le modèle déformable qui nous intéresse peut être considérée comme étant stable tout au long de la simulation. Cependant, on pourrait imaginer des cas où un fil de suture traversant un modèle déformable peut voir son déplacement totalement bloqué par la seule force exercée par un autre objet appuyant sur celui-ci. Si une application directe de ce modèle d'interaction venait à être trouvée, il serait intéressant de continuer à l'étudier.

Si le temps de calcul sur notre scène d'exemple de la suture permet une simulation interactive, il s'avère nécessaire de réaliser des optimisations avant de mettre en place une simulation complète du geste de la suture avec réalisation de plusieurs points de suture et d'un nœud dans le fil. En effet, on résoud à chaque pas de temps l'ensemble des contraintes ; or le temps de calcul de leur résolution croit suivant le nombre de contraintes dans le système. De plus, dans le cas des contraintes utilisées pour la simulation de la suture, on se rend compte qu'elles changent très peu d'état dès lors qu'elles ne sont pas situées sur le dernier point de suture, celui sur lequel interagit principalement l'utilisateur. On peut donc imaginer, comme dans les simulateurs de suture commerciaux existants, de simplifier les points de suture qui n'évoluent plus.

Tout d'abord, on définit un point de suture du point de vue des contraintes comme étant l'ensemble compris entre un point d'entrée sur un objet et un point de sortie sur l'objet suivant. On basera ensuite les optimisations sur l'observation que les contraintes de chemin qui se situent entre ces deux points ne changeront d'état qu'après qu'une de ces deux contraintes (entrée ou sortie) ne change d'état elle-même. Les points de suture seront traités comme des entités à part entière, et dès lors que la position relative entre le fil de suture et les points de contraintes reste fixe sur plusieurs pas de temps, on pourra considérer ce point de suture comme étant immobile. On pourra alors décider de ne pas simuler physiquement cette partie de la scène, afin de gagner un temps de calcul précieux. En effet, en réduisant le nombre de contraintes on diminue le temps de calcul de leur résolution, mais aussi du calcul de la compliance. Un point de suture inactif peut repasser à l'état simulable si la force exercée par le fil à l'un des points extrèmes dépasse la force de frottement à cette position.

Un travail supplémentaire est requis afin de trouver la méthode idéale pour simplifier localement les modèles déformables, en appliquant des régions rigides autour des points de suture qui n'évoluent plus. De la même façon, les nœuds réalisés sur le fil de suture doivent pouvoir être rigidifiés. Cette optimisation devrait être plus aisée à mettre en place puisque ne faisant intervenir qu'un seul modèle en interaction avec lui-même. En détectant la tension du fil dans le nœud ou à ses deux extrémités, on considère qu'il est serré. On peut alors remplacer les éléments de poutre le constituant par un seul élément reliant les deux points extrèmes, et ne garder la configuration du nœud qu'aux niveaux visuel et de collision. Les nœuds n'étant pas dénoués par les praticiens durant les opérations, il est inutile de vérifier si l'utilisateur cherche à les défaire.

Comme nous avons pu le montrer durant ce travail, l'algorithme du Gauss-Seidel peut bénéficier de plusieurs techniques pour en améliorer la vitesse de convergence.

La méthode de départ à chaud est actuellement très naive dans son implantation, et devrait être largement repensée pour être généralisable à tous les types de contraintes. L'utilisation de structures de données plus adaptées permettrait de lier une contrainte aux primitives auxquelles elle s'applique. A partir de là, il serait envisageable de stocker les informations de cette contrainte afin de les récupérer au pas de temps suivant, même si la contrainte est supprimée puis recrée à partir de zéro. Cela est par exemple le cas de toutes les contraintes de contact, qui sont recréées entièrement après chaque étape de détection de collision.

La technique de sur-relaxation que nous avons également présentée doit être paramétrée à la main afin de trouver le coefficient qui offre la meilleure vitesse de convergence, sans faire apparaître de vibrations ou même de divergence de l'erreur. D'une simulation à l'autre, celui-ci peut avoir des influences très différentes. L'idéal serait de trouver une méthode adaptative où ce paramètre (et potentiellement d'autres) serait modifié en fonction de la vitesse de convergence mesurée durant la résolution des contraintes. Cela permettrait par la même occasion de détecter les cas où l'algorithme converge trop lentement et qui font exploser le temps de calcul ; et d'y remédier en arrêtant l'algorithme ou en essayant ponctuellement d'autres méthodes d'accélération.

Une autre méthode devant être testée consiste à regrouper les contraintes. D'un côté, le regroupement selon le type des contraintes permettrait de paralléliser le calcul du Gauss-Seidel sur plusieurs processeurs ou sur les processeurs massivement parallèles des cartes graphiques. Des essais concluants ont ainsi été obtenus par [CA09] pour le calcul du Gauss-Seidel sur GPU, mais seulement pour des contraintes de contact.

En triant les contraintes il devrait également être possible de séparer les paires de

contraintes qui n'ont aucun couplage mécanique entre elles. La résolution des contraintes pourrait alors se faire par bloc de contraintes ; on obtiendrait ainsi plusieurs LCP de petite taille plutôt qu'un seul de grande taille, sensiblement plus rapides à calculer.

Une problématique similaire est que l'on observe souvent durant la résolution certaines contraintes qui très rapidement ne changent plus d'état et dont la force de réponse reste stable. Il est alors inutile de devoir repasser sur celles-ci à chaque itération, et d'interroger leur loi de comportement. On pourrait donc réduire progressivement la taille du problème à résoudre et se concentrer uniquement sur les contraintes qui continuent à évoluer. Certains cas particuliers tels que le changement d'état d'une contrainte de pénétration d'une aiguille peuvent cependant avoir une grande influence sur les autres contraintes, et pourraient nécéssiter de considérer à nouveau celles-ci dans la résolution.

Pour les contraintes s'appliquant le long d'un modèle 1d comme le fil de suture ou l'aiguille chirurgicale, il a été montré que le tri des contraintes selon l'abscisse paramétrique de leur point d'application améliore sensiblement leur résolution. On peut alors profiter de cette structure pour propager de proche en proche les déplacements demandés par les contraintes. Une fois arrivé au bout de la structure, la technique conseille de repartir dans le sens inverse, plutôt que de recommencer au début comme nous le faisons dans le cas général qui ne présente pas de structure particulière.

Nous avons montré que dans certaines scènes complexes telles que les exemples de suture, le calcul de la compliance représentait souvent une part très importante d'un pas de la simulation, bien plus que la simple résolution des contraintes. Des travaux récents [CDL+05] ont montré qu'il était possible de mettre en œuvre un Gauss-Seidel sans avoir à construire la matrice de compliance.

Si l'on considère le calcul de la force de réponse d'une contrainte en particulier, nous avons besoin de connaître à chaque itération sa nouvelle violation calculée jusqu'à présent en sommant la contribution de toutes les autres contraintes, ainsi que la compliance à cette position. Celle-ci correspond à la diagonale de la matrice de compliance et s'obtient instantanément ; nous résolvons alors cette contrainte selon sa loi de comportement. La différence par rapport à la méthode où l'on construit la compliance est que l'on délègue alors aux objets de la scène le calcul de l'influence de cette force de réponse sur les autres contraintes.

Plutôt que d'itérer chaque contrainte pour calculer la violation δ_a, on demande alors directement à l'objet auquel est attaché cette contrainte de nous donner la nouvelle position de celle-ci. On peut alors optimiser le calcul selon le type du modèle (rigide, déformable 3d ou 1d), et on réduit par la même occasion le coût de chaque itération du Gauss-Seidel. Le gain le plus important concerne ainsi les objets rigides, pour lesquels ils suffit de réaliser l'addition des forces des contraintes qui y sont appliquées. Cette méthode a été mise en œuvre sur un algorithme de Gauss-Seidel spécialisé pour la résolution des contraintes de contact. Elle doit pouvoir être transposée à notre technique générique de résolution des contraintes.

L'entraînement aux gestes opératoires est un sujet qui nous intéresse beaucoup, et c'est pourquoi nous voulons continuer à perfectionner la simulation de suture, dans tous les aspects que nous venons de présenter. À plus long terme, nous aimerions également intervenir au niveau per-opératoire. Nous pensons tout particulièrement à l'opération de la greffe de cornée, qui fait intervenir une suture continue d'environ 30 points. La répartition de ces points, et la tension du fil de suture à leur niveau, influent sur les déformations de la cornée ; la géométrie que prend alors celle-ci détermine la bonne vue du patient ou non.

Bibliographie

[ABG08] R. Alterovitz, M. Branicky et K. Goldberg. « Motion Planning Under Uncertainty for Image-Guided Medical Needle Steering ». *International Journal of Robotics Research*, 27(11-12) :1361–1374, 2008.

[ACR95] U. M. Ascher, H. Chin et S. Reich. « Stabilization of DAEs and invariant manifolds ». *Numerische Mathematik*, 67(2) :131–149, 1995.

[AGHmD03] O. E. Arash, O. Génevaux, A. Habibi et J. michel Dischler. « Simulating Fluid-Solid Interaction ». Dans *in Graphics Interface*, pages 31–38, 2003.

[AGO05] R. Alterovitz, K. Goldberg et A. Okamura. « Planning for Steerable Bevel-tip Needle Insertion Through 2D Soft Tissue with Obstacles ». Dans *Proceedings of ICRA 2005*, pages 1652–1657, 2005.

[AMS00] F. S. Azar, D. N. Metaxas et M. D. Schnall. « A Finite Element Model of the Breast for Predicting Mechanical Deformations during Biopsy Procedures ». Dans *MMBIA '00 : Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, page 38, Washington, DC, USA, 2000. IEEE Computer Society.

[Ant95] S. Antman. *Nonlinear Problems of Elasticity*. Springer, 1995.

[AOW+08] B. Adams, M. Ovsjanikov, M. Wand, H.-P. Seidel et L. J. Guibas. « Meshless modeling of deformable shapes and their motion ». Dans *SCA '08 : Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 77–86, 2008.

[APA07] N. Abolhassani, R. Patel et F. Ayazi. « Effects of Different Insertion Methods on Reducing Needle Deflection ». Dans *Proceedings of IEEE EMBS 2007*, pages 491–494, 2007.

[APM07] N. Abolhassani, R. Patel et M. Moallem. « Needle Insertion into soft Tissue : A survey ». *Medical Engineering and Physics*, 29 :413–431, 2007.

[APS98] M. Anitescu, F. A. Potra et D. E. Stewart. « Time-Stepping for Three-Dimensional Rigid Body Dynamics ». *Computer Methods in Applied Mechanics and Engineering*, 177(3–4) :183–197, 1998.

[APT+03a] R. Alterovitz, J. Pouliot, R. Taschereau, I.-C. J. Hsu et K. Goldberg. « Needle Insertion and Radioactive Seed Implantation in Human Tissues : Simulation and Sensitivity Analysis ». Dans *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pages 1793–1799, 2003.

[APT+03b] R. Alterovitz, J. Pouliot, R. Taschereau, I.-C. J. Hsu et K. Goldberg. « Simulating needle insertion and radioactive seed implantation for prostate brachytherapy ». Dans *Proceedings of Medicine Meets Virtual Reality 11*, pages 19–25, 2003.

[Asc97] U. M. Ascher. « Stabilization of invariants of discretized differential systems ». *Numerical Algorithms*, 14(1–3) :1–24, 1997.

[BAC+06] F. Bertails, B. Audoly, M.-P. Cani, B. Querleux, F. Leroy et J.-L. Lévêque. « Super-helices for predicting the dynamics of natural hair ». Dans *SIGGRAPH '06*, pages 1180–1187. ACM, 2006.

[Bar89] D. Baraff. « Analytical methods for dynamic simulation of non-penetrating rigid bodies ». *SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, 23(3) :223–232, 1989.

[Bar91] D. Baraff. « Coping with friction for non-penetrating rigid body simulation ». Dans *SIGGRAPH '91 : Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 31–41, 1991.

[Bar94] D. Baraff. « Fast contact force computation for nonpenetrating rigid bodies ». Dans *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 23–34, 1994.

[Bar96] D. Baraff. « Linear-time dynamics using Lagrange multipliers ». Dans *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 137–146, New York, NY, USA, 1996. ACM.

[Bau72] J. Baumgarte. « Stabilization of constraints and integrals of motion in dynamical systems ». *Computer Methods in Applied Mechanics and Engineering*, 1 :1–16, 1972.

[BB88] R. Barzel et A. H. Barr. « A modeling system based on dynamic constraints ». Dans *SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 179–188, 1988.

[BBB07] C. Batty, F. Bertails et R. Bridson. « A fast variational framework for accurate solid-fluid coupling ». *ACM Trans. Graph.*, 26(3) :100, 2007.

[BC98] A. Belegundu et T. Chandrupatla. *Optimization concepts and applications in engineering*. Prentice Hall, 1998.

[BC00] D. Bourguignon et M. Cani. « Controlling anisotropy in mass-spring systems ». Dans *Computer animation and simulation 2000 : proceedings of the Eurographics Workshop*, pages 113–123, 2000.

[Ben07] J. Bender. « Impulse-based dynamic simulation in linear time ». *Comput. Animat. Virtual Worlds*, 18(4-5) :225–233, 2007.

[Ber09] F. Bertails. « Linear Time Super-Helices ». *Comput. Graph. Forum*, 28(2) :417–426, 2009.

[BHG92] D. Breen, D. House et P. Getto. « A physically-based particle model of woven cloth ». *The Visual Computer*, 8(5) :264–277, 1992.

[BHT00] P. Brett, A. Harrison et T. Thomas. « Schemes for the identification of tissue types and boundaries at the tool point for surgical needles ». Dans *IEEE Transactions on Information Technology in Biomedicine*, volume 4, pages 30–36, 2000.

[BHW94] D. E. Breen, D. H. House et M. J. Wozny. « Predicting the drape of woven cloth using interacting particles ». Dans *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 365–372, 1994.

[Bin87] T. Binford. « Generalized cylinder representation ». *Encyclopedia of Artificial Intelligence*, pages 321–323, 1987.

[BLM04] J. Brown, J.-C. Latombe et K. Montgomery. « Real-time knot-tying simulation ». *Vis. Comput.*, 20(2) :165–179, 2004.

[BMcLS01] J. Brown, K. Montgomery, J. claude Latombe et M. Stephanides. « A Microsurgery Simulation System ». Dans *Medical Image Computing and Computer-Assisted Interventions (MICCAI)*, pages 137–144, 2001.

[BN95] M. Bro-Nielsen. « Modelling elasticity in solids using active cubes : application to simulated operations ». Dans *Computer Vision, Virtual Reality and Robotics in Medicine*, pages 533–541, 1995.

[BnC96] M. Bro-nielsen et S. Cotin. « Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation ». *Computer Graphics Forum*, 15(3) :57–66, 1996.

[But03] J. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2003.

[BW97] D. Baraff et A. Witkin. « Partitioned Dynamics ». Rapport Technique CMU-RI-TR-97-33, *Robotics Institute, Carnegie Mellon University*, 1997.

[BW98] D. Baraff et A. Witkin. « Large steps in cloth simulation ». Dans *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, 1998.

[BWK03] D. Baraff, A. Witkin et M. Kass. « Untangling cloth ». *ACM Transactions on Graphics*, 22(3) :862–870, 2003.

[CA09] H. Courtecuisse et J. Allard. « Parallel Dense Gauss-Seidel Algorithm on Many-Core Processors ». Dans *Proceedings of the 2009 11th IEEE International Conference on High Performance Computing and Communications-Volume 00*, pages 139–147. IEEE Computer Society, 2009.

[CAR+09] N. Chentanez, R. Alterovitz, D. Ritchie, L. Cho, K. Hauser, K. Goldberg, R. J. Shewchuk et F. J. O'Brien. « Interactive Simulation of Surgical Needle Insertion and Steering ». Dans *The Proceedings of ACM SIGGRAPH 2009*, 2009.

[CC09] E. Cosserat et F. Cosserat. *Théorie des Corps déformables*. Jacques Gabay, 1909.

[CDA96] S. Cotin, H. Delingette et N. Ayache. « Real Time Volumetric Deformable Models for Surgery Simulation ». Dans *VBC '96 : Proceedings of the 4th International Conference on Visualization in Biomedical Computing*, pages 535–540, 1996.

[CDA99] S. Cotin, H. Delingette et N. Ayache. « Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation ». *IEEE Transactions on Visualization and Computer Graphics*, 5(1) :62–73, 1999.

[CDL+05] S. Cotin, C. Duriez, J. Lenoir, P. Neumann et S. Dawson. « New approaches to catheter navigation for interventional radiology simulation ». Dans *Proceedings of MICCAI'05*, pages 534–542, 2005.

[CFL67] R. Courant, K. Friedrichs et H. Lewy. « On the partial difference equations of mathematical physics ». *IBM J. Res. Dev.*, 11(2) :215–234, 1967.

[CGC+02] S. Capell, S. Green, B. Curless, T. Duchamp et Z. Popović. « Interactive skeleton-driven dynamic deformations ». *ACM Trans. Graph.*, 21(3) :586–593, 2002.

[CGFO06] N. Chentanez, T. G. Goktekin, B. E. Feldman et J. F. O'Brien. « Simultaneous coupling of fluids and deformable bodies ». Dans *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 83–89, 2006.

[Chi95] H. S. Chin. « Stabilization Methods for Simulations of Constrained Multibody Dynamics ». Thèse de doctorat, *Department of mathematics, British Columbia*, mai 1995.

[CHP89] J. Chadwick, D. Haumann et R. Parent. « Layered construction for deformable animated characters ». Dans *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, page 252, 1989.

[Cli02] M. B. Cline. « Rigid body simulation with contact and constraint ». Thèse de doctorat, *University of British Columbia*, nov 2002.

[CMP96] C. G. L. Cao, C. L. Mackenzie et S. Payandeh. « Task And Motion Analyses In Endoscopic Surgery ». Dans *ASME IMECE Conference Proceedings : 5th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator systems*, pages 583–590, 1996.

[CMT04] M. Carlson, P. J. Mucha et G. Turk. « Rigid fluid : animating the interplay between rigid bodies and fluid ». *ACM Trans. Graph.*, 23(3) :377–384, 2004.

[Coe99] C. A. C. Coello. « A Survey of Constraint Handling Techniques used with Evolutionary Algorithms ». Rapport Technique, *Laboratorio Nacional de Informática Avanzada*, 1999.

[CP03] M. B. Cline et D. K. Pai. « Post-stabilization for rigid body simulation with contact and constraints ». Dans *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 3744–3751, 2003.

[CSWO05] J. Crouch, C. Schneider, J. Wainer et A. Okamura. « A velocity-dependent model for needle insertion in soft tissue ». Dans *Proceedings of MICCAI'05*, pages 624–632, 2005.

[DAS01] T. Dang, T. M. Annswamy et M. A. Srinivasan. « Development and evaluation of an epidural injection simulator with force feedback for medical training ». Dans *Medicine Meets Virtual Reality 2001*, pages 97–102, 2001.

[DDBC99] G. Debunne, M. Desbrun, A. Barr et M. Cani. « Interactive multiresolution animation of deformable models ». Dans *Eurographics Workshop on Computer Animation and Simulation*, volume 99, pages 133–144, 1999.

[DDCB00] G. Debunne, M. Desbrun, M. Cani et A. Barr. « Adaptive simulation of soft bodies in real-time ». Dans *Computer Animation 2000*, page 15, 2000.

[DDCB01] G. Debunne, M. Desbrun, M.-P. Cani et A. H. Barr. « Dynamic real-time deformations using space & time adaptive sampling ». Dans *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 31–36, New York, NY, USA, 2001. ACM.

[DDK06] C. Duriez, F. Dubois et A. a. Kheddar. « Realistic Haptic Rendering of Interacting Deformable Objects in Virtual Environments ». *IEEE Transactions on Visualization and Computer Graphics*, 12(1) :36–47, 2006.

[DDKA06] C. Duriez, F. Dubois, A. Kheddar et C. Andriot. « Realistic Haptic Rendering of Interacting Deformable Objects in Virtual Environments ». *IEEE Transactions on Visualization and Computer Graphics*, 12(1) :36–47, 2006.

[Del98] H. Delingette. « Toward realistic soft-tissue modeling in medical simulation ». Dans *Proceedings of the IEEE*, pages 512–523, March 1998.

[Des97] M. Desbrun. « Modélisation et animation de matériaux hautement déformables en synthèse d'images ». Thèse de doctorat, *Institut National Polytechnique de Grenoble*, 1997.

[DG96] M. Desbrun et M. Gascuel. « Smoothed particles : A new paradigm for animating highly deformable bodies ». Dans *Computer Animation and Simulation*, volume 96, pages 61–76, 1996.

[DGM+09] C. Duriez, C. Guébert, M. Marchal, S. Cotin et L. Grisoni. « Interactive Simulation of Flexible Needle Insertions Based on Constraint Models ». *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pages 291–299, 2009.

[DGS06] E. Dehghan, O. Goksel et S. E. Salcudean. « A Comparison of Needle Bending Models ». Dans *MICCAI '06 : Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 305–312, 2006.

[DKT95] O. Deussen, L. Kobbelt et P. Tücke. « Using simulated annealing to obtain good nodal approximations of deformable bodies ». Dans *Proceedings of the Sixth Eurographics Workshop on Simulation and Animation*. Citeseer, 1995.

[DLC07] J. Dequidt, J. Lenoir et S. Cotin. « Interactive Contacts Resolution Using Smooth Surface Representation ». Dans N. A. O. J. Maeder, éditeur, *Proceedings of MICCAI*, Lecture Notes in Computer Science, pages 850–857. Springer, September 2007.

[DMD+08] J. Dequidt, M. Marchal, C. Duriez, E. Kerrien et S. Cotin. « Interactive Simulation of Embolization Coils : Modeling and Experimental Validation ». Dans *Proceedings of MICCAI*, pages 695–702, September 2008.

[DRG03] J. Denise, D. Reversat et A. Gagalowicz. « Modeling of the Hysteretic Behaviour of Fabrics ». Dans *Proceedings of Computer Vision / Computer Graphics Collaboration for Model-based Imaging, Rendering, image Analysis and Graphical special Effects*, 2003.

[DS02a] S. P. DiMaio et S. E. Salcudean. « Needle Insertion Modeling and Simulation ». Dans *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2098–2105, 2002.

[DS02b] S. P. DiMaio et S. E. Salcudean. « Needle Insertion Modelling for the Interactive Simulation of Percutaneous Procedures ». Dans *MICCAI '02 : Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 253–260, London, UK, 2002. Springer-Verlag.

[DS02c] S. P. DiMaio et S. E. Salcudean. « Simulated Interactive Needle Insertion ». Dans *HAPTICS '02 : Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, page 344, Washington, DC, USA, 2002. IEEE Computer Society.

[DS03a] S. DiMaio et S. Salcudean. « Needle Insertion Modeling and Simulation ». *IEEE Transactions on Robotic Automation*, 19(5) :864–875, 2003.

[DS03b] S. DiMaio et S. Salcudean. « Needle Steering and Model-Based Trajectory Planning ». Dans *MICCAI '03 : Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 33–40, 2003.

[DS05a] S. DiMaio et S. Salcudean. « Interactive simulation of needle insertion models ». *IEEE Transactions on Biomedical Engineering*, pages 1167–1179, 2005.

[DS05b] S. DiMaio et S. Salcudean. « Needle Steering and Motion Planning in Soft Tissues ». *IEEE Transactions on Biomedical Engineering*, 19(6) :965–974, 2005.

[DS06] E. Dehghan et S. E. Salcudean. « Comparison of Linear and Non-linear Models in 2D Needle Insertion Simulation ». Dans *Proceedings of the Workshop on Computational Biomechanics for Medecine (in conjunction wuth MICCAI 2006)*, pages 117–124, 2006.

[DSB99] M. Desbrun, P. Schröder et A. Barr. « Interactive animation of structured deformable objects ». Dans *Proceedings of the 1999 conference on Graphics interface '99*, pages 1–8, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[DWZA+07] E. Dehghan, X. Wen, R. Zahiri-Azar, M. Marchal et S. Salcudean. « Modeling of Needle-Tissue Interaction Using Ultrasound-Based Motion Estimation ». Dans *MICCAI '06 : Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 709–716, 2007.

[DWZA+08] E. Dehghan, X. Wen, R. Zahiri-Azar, M. Marchal et S. Salcudean. « Needle-tissue Interaction Modeling using Ultrasound-based Motion Estimation : Phantom Study ». *Computer Aided Surgery*, 13(5) :265–280, 2008.

[DXA+08] V. Duidam, J. Xu, R. Alterovitz, S. Sastry et K. Goldberg. « 3D Motion Planning Algorithms for Steerable Needles Using Inverse Kinematics ». Dans *Proceedings of WAFR 2008*, pages 13–18, 2008.

[EWS96] B. Eberhardt, A. Weber et W. Strasser. « A fast, flexible, particle-system model for cloth draping ». *IEEE Computer Graphics and Applications*, pages 52–59, 1996.

[Fau98] F. Faure. « Interactive Solid Animation Using Linearized Displacement Constraints ». Dans *9th Eurographics Workshop on Computer Animation and Simulation, EGCAS, September, 1998*, pages 61–72, 1998.

[Fea87] R. Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, 1987.

[Fea99] R. Featherstone. « A Divide-and-Conquer Articulated-Body Algorithm for Parallel O(log(n)) Calculation of Rigid-Body Dynamics. ». *The International Journal of Robotics Research*, 18(9) :876–892, 1999.

[FM68] A. V. Fiacco et G. P. McCormick. *Nonlinear Programming : Sequential Unconstrained Minimation Techniques*. John Wiley and Sons, Inc, 1968.

[Fun93] Y. C. Fung. *Biomechanics : Mechanical Properties of Living Tissues, 2nd ed.* Springer, 1993.

[GCG94] J.-D. Gascuel et M.-P. Cani-Gascuel. « Displacement constraints for interactive modeling and animation of articulated structures ». *The Visual Computer*, 10(4) :191–204, apr 1994.

[GDC+09] C. Guébert, C. Duriez, S. Cotin, J. Allard et L. Grisoni. « Suturing Simulation Based on Complementarity Constraints ». Dans *Proceedings of SCA 2009 (poster)*, août 2009.

[GDG08] C. Guébert, C. Duriez et L. Grisoni. « Unified Processing of Constraints for Interactive Simulation ». Dans *Proceedings of VRIPHYS'08*, 2008.

[GKW+00] P. Gorman, T. Krummel, R. Webster, M. Smith et D. Hutchens. « Catheter insertion simulation with co-registered direct volume rendering and haptic feedback ». Dans *Medicine Meets Virtual Reality 2000*, pages 106–109, 2000.

[GM97] S. Gibson et B. Mirtich. « A survey of deformable modeling in computer graphics ». *Technical Report TR-97-19, Mitsubishi Electric Research Laboratory*, 1997.

[GM03] L. Grisoni et D. Marchal. « High Performance generalized cylinders visualization ». Dans *SMI '03 : Proceedings of the Shape Modeling International 2003*, page 257, Washington, DC, USA, 2003. IEEE Computer Society.

[GMTT89] J. Gourret, N. Magnenat-Thalmann et D. Thalmann. « The use of finite element theory for simulating object and human body deformations and contacts ». Dans *Proc. Eurographics*, volume 89, 1989.

[Gou94] J. Gourret. « *Modélisation d'Images Fixes et animées* », Chapitre VI - Assemblage et Animation d'Objets Déformables : Simulation par Eléments Finis". Masson, Paris, 1994.

[GPS02] H. Goldstein, C. Poolo et J. Safko. *Classical mechanics (3rd ed.)*. Addison-Wesley, 2002.

[GS04] D. Glozman et M. Shoham. « Flexible Needle Steering and Optimal Trajectory Planning for Percutaneous Therapies ». Dans *Proceedings of MICCAI 2004*, pages 137–144, 2004.

[GS06] M. Grégoire et E. Schömer. « Interactive simulation of one-dimensional flexible parts ». Dans *SPM '06 : Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 95–103, New York, NY, USA, 2006. ACM.

[GSD+05] O. Goksel, S. E. Salcudean, S. P. DiMaio, R. Rohling, et J. Morris. « 3D Needle-Tissue Interaction Simulation for Prostate Brachytherapy ». Dans *MICCAI '05 : Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 827–834, 2005.

[GSLF05] E. Guendelman, A. Selle, F. Losasso et R. Fedkiw. « Coupling water and smoke to thin deformable and rigid shells ». Dans *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 973–981, 2005.

[GTS02] S. Green, G. Turkiyyah et D. Storti. « Subdivision-based multilevel methods for large scale engineering simulation of thin shells ». Dans *SMA '02 : Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 265–272, New York, NY, USA, 2002. ACM.

[GTZZ00] E. Gobbetti, M. Tuveri, G. Zanetti et A. Zorcolo. « Catheter insertion simulation with co-registered direct volume rendering and haptic feedback ». Dans *Medicine Meets Virtual Reality 2000*, pages 96–98, 2000.

[Hab97] A. Habibi. « Modeles physiques supports de la relation mouvement-forme-image ». Thèse de doctorat, *ACROE/CLIPS-Institut National Polytechnique de Grenoble*, 1997.

[HB00] D. House et D. Breen. *Cloth modeling and animation.* AK Peters, Ltd., 2000.

[HBB+04] R. Holbrey, A. Bulpitt, K. Brodlie, M. Walkley et J. Scott. « A model for virtual suturing in vascular surgery ». Dans *Theory and Practice of Computer Graphics, 2004. Proceedings*, pages 50–58, June 2004.

[HBD05] J. Hing, A. Brooks et J. Desai. « Reality-based estimation of needle and soft-tissue interaction for accurate haptic feedback in prostate brachytherapy simulation ». Dans *Proceedings of International Symposium of Robotics Research*, pages 30–36, 2005.

[HBD07] J. Hing, A. Brooks et J. Desai. « A Biplanar fluoroscopic approach for the measurement, modeling and simulation of needle and soft-tissue interaction ». *Medical Image Analysis*, 11 :62–78, 2007.

[Hoo06] W. G. Hoover. *Smooth Particle Applied Mechanics : The State of the Art.* World Scientific Publishing Co., Inc., 2006.

[HS04] M. Hauth et W. Strasser. « Corotational simulation of deformable solids ». Dans *Proceedings of WSCG*, pages 137–145, 2004.

[HSS98] L. Hiemenz, D. Stredney et P. Schmalbrock. « Development of the force-feedback model for an epidural needle insertion simulator ». Dans *Proceedings of Medecine Meets Virtual Reality*, pages 272–277, 1998.

[JV03] J. Jansson et J. S. M. Vergeest. « Combining deformable- and rigid-body mechanics simulation ». Dans *In The Visual Computer*, pages 280–290. SpringerVerlag, 2003.

[KAG+05] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutre et M. Gross. « A unified Lagrangian approach to solid-fluid animation ». Dans *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, pages 125–148, June 2005.

[Kar85] D. Karnopp. « Computer simulation of stick-slip friction in mechanical dynamic systems ». *Journal of Dynamic Systems, Measurement, and Control*, 107 :100–103, 1985.

[KCC+00] Y. Kang, J. Choi, H. Cho, D. Lee et C. Park. « Real-time animation technique for flexible and thin objects ». *WSCG 2000*, pages 322–329, 2000.

[KcM00] U. Kühnapfel, H. Çakmak et H. Maass. « Endoscopic Surgery Training Using Virtual Reality and Deformable Tissue Simulation ». *Computers & Graphics*, 24(5) :671–682, October 2000.

[KEP05] D. M. Kaufman, T. Edmunds et D. K. Pai. « Fast frictional dynamics for rigid bodies ». Dans *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 946–956, 2005.

[KFCO06] B. M. Klingner, B. E. Feldman, N. Chentanez et J. F. O'Brien. « Fluid animation with dynamic meshes ». *ACM Trans. Graph.*, 25(3) :820–825, 2006.

[KKK+01] D.-S. Kwon, K.-U. Kyung, S. M. Kwon, J. B. Ra, H. W. Park, H. S. Kang, J. Zeng et K. Cleary. « Realistic force reflection in a spine biopsy simulator ». Dans *ICRA '01 : Proceedings of the International Conference on Robotics and Automation*, pages 1358–1363, 2001.

[KP93] S. Kumar et A. Petho. « An algorithm for the numerical inversion of a tridiagonal matrix ». *Communications in numerical methods in engineering*, 9(4) :353–359, 1993.

[KWAM01] H. Kataoka, T. Washio, M. Audette et L. Mizuhara. « A model for relations between needle deflection, force and thickness on needle penetration ». Dans *MICCAI '01 : Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 966–974, 2001.

[KWC+02] H. Kataoka, T. Washio, K. Chinzei, K. Mizuhara, C. Simone et A. M. Okamura. « Measurement of the Tip and Friction Force Acting on a Needle during Penetration ». Dans *MICCAI '02 : Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 216–223, London, UK, 2002. Springer-Verlag.

[LCDN06] J. Lenoir, S. Cotin, C. Duriez et P. Neumann. « Interactive physically-based simulation of catheter and guidewire ». *Computer and Graphics*, 30 :416–422, 2006.

[Lem65] C. Lemke. « Bimatrix equilibrium points and mathematical programming ». *Management Science*, 11 :681–689, 1965.

[Len04] J. Lenoir. « Modèle déformable 1D pour la simulation physique temps réel ». Thèse de doctorat, *Université des Sciences et Technologies de Lille*, 2004.

[LF04] J. Lenoir et S. Fonteneau. « Mixing deformable and rigid-body mechanics simulation ». *Computer Graphics International, 2004. Proceedings*, pages 327–334, June 2004.

[LG98] M. C. Lin et S. Gottschalk. « Collision Detection Between Geometric Models : A Survey ». Dans *In Proc. of IMA Conference on Mathematics of Surfaces*, pages 37–56, 1998.

[LMGC02] J. Lenoir, P. Meseure, L. Grisoni et C. Chaillou. « Surgical thread simulation ». Dans *ESAIM : Proceedings*, volume 12, pages 102–107, 2002.

[LMGC04] J. Lenoir, P. Meseure, L. Grisoni et C. Chaillou. « A Suture Model for Surgical Simulation ». Dans *in International Symposium on Medical Simulation*, pages 17–18, 2004.

[Loo87] C. Loop. « Smooth subdivision surfaces based on triangles ». Thèse de doctorat, *Department of Mathematics, The University of Utah*, 1987.

[Lou94] J. Louchet. « Evolution auto-adaptative pour identifier les modèles physiques d'animation particulaire », 1994. Séminaire Animation et Simulation.

[LPC95] J. Louchet, X. Provot et D. Crochemore. « Evolutionary identification of cloth animation models ». Dans *Computer Animation and Simulation*, volume 95, pages 44–54. Citeseer, 1995.

[LPD03] M. LeDuc, S. Payandeh et J. Dill. « Toward Modeling of a Suturing Task ». Dans *In Graphics Interface conference*, 2003.

[LRK+03] J. Leung, G. Ruff, M. King, P. J. Dattilo, E. Nelan et D. Sasselov. « Barbed, Bi-directional Surgical Sutures ». Dans *MEDTEX03, International Conference & Exhibition on Healthcare & Medical Textiles*, 2003.

[LSSF06] F. Losasso, T. Shinar, A. Selle et R. Fedkiw. « Multiple interacting liquids ». Dans *SIGGRAPH '06 : ACM SIGGRAPH 2006 Papers*, pages 812–819, 2006.

[LTW93] Y. Lee, D. Terzopoulos et K. Waters. « Constructing physics-based facial models of individuals ». *In Proceedings of Graphics Interface*, pages 1–8, 1993.

[Mar06] M. Marchal. « Modélisation des tissus mous dans leur environnement pour l'aide aux gestes médico-chirurgicaux ». Thèse de doctorat, *Université Joseph Fourier, Grenoble, France*, 2006.

[Mar08] R. F. Marcia. « On solving sparse symmetric linear systems whose definiteness is unknown ». *Applied Numerical Mathematics*, 58(4) :449–458, 2008.

[Mau99] W. Maurel. « 3D modeling of the human upper limb including the biomechanics of joints, muscles and soft tissues ». Thèse de doctorat, *Laboratoire d'Infographie - Ecole Polytechnique Fédérale de Lausanne, Switzerland*, 1999.

[MbB+04] B. Maurin, L. barbe, B. Bayle, P. Zanne, J. Gangloff, M. D. Mathelin, A. Gangi, L. Soler et A. Forgione. « In Vivo Study of Forces During Needle Insertions ». Dans *Proceedings of the scientific workshop on medical robots, navigation and visualization (MRNV)*, pages 415–422, 2004.

[MC95] B. Mirtich et J. Canny. « Impulse-based simulation of rigid bodies ». Dans *I3D '95 : Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 181–ff., 1995.

[MDH+03] P. Meseure, J. Davanne, L. Hilde, J. Lenoir, L. France, F. Triquet et C. Chaillou. « A physically-based virtual environnement dedicated to surgical simulation ». Dans *IS4TM*, 2003.

[MDM+02] M. Müller, J. Dorsey, L. McMillan, R. Jagnow et B. Cutler. « Stable real-time deformations ». Dans *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 49–54, 2002.

[Mil88] G. Miller. « The motion dynamics of snakes and worms ». *ACM Siggraph Computer Graphics*, 22(4) :169–173, 1988.

[MJ96] J.-J. Moreau et M. Jean. « Numerical treatment of contact and friction : the contact dynamic method ». *Engineering Systems Design and Analysis, vol 4*, pages 201–208, 1996.

[MJK99] S. Miller, J. Jeffrey, C nd Bews et W. Kinsner. « Advances in the virtual reality interstitial brachytherapy system ». Dans *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, pages 349–354, 1999.

[MKN+04] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross et M. Alexa. « Point based animation of elastic, plastic and melting objects ». Dans *SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 141–151, 2004.

[MLM+05] U. Meier, O. Lopez, C. Monserrat, M. Juan et M. Alcaniz. « Real-time deformable models for surgery simulation : a survey ». *Computer methods and programs in biomedicine*, 77(3) :183–197, 2005.

[MPD05] P. Marshall, S. Payandeh et J. Dill. « Suturing for surface meshes ». Dans *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*, pages 31–36, Aug. 2005.

[MPD06] P. Marshall, S. Payandeh et J. Dill. « A Study on Haptic Rendering in a Simulated Surgical Training Environment ». Dans *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2006 14th Symposium on*, pages 241–247, March 2006.

[MPT05] M. Marchal, E. Promayon et J. Troccaz. « Simulating complex organ interactions : Validation of a soft tissue discrete model ». Dans *Proceedings of International Symposium on Visual Computing*, pages 175–182, 2005.

[MPT06] M. Marchal, E. Promayon et J. Troccaz. « Simulating Prostate Surgical Procedures with a Discrete Soft Tissue Model ». Dans *Proceedings of the 3rd Workshop in Virtual Reality Interactions and Physical Simulation*, pages 109–118, 2006.

[MST+04] M. Müller, S. Schirm, M. Teschner, B. Heidelberger et M. Gross. « Interaction of fluids with deformable solids : Research Articles ». *Comput. Animat. Virtual Worlds*, 15(3-4) :159–171, 2004.

[MSZ94] R. M. Murray, S. S. Sastry et L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1994.

[Mur88] K. G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, 1988.

[Mur97] K. Murty. « Linear Complementarity, Linear and Nonlinear Programming. Internet Edition », 1997.

[MW88] M. Moore et J. Wilhelms. « Collision detection and response for computer animation ». Dans *SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 289–298, 1988.

[NCY+00] S. Nath, Z. Chen, N. Yue, S. Trumpore et R. Peschel. « Dosimetric effects of needle divergence in prostate seed implant using 125I and 103Pd radioactive seeds ». *Med-Phys*, 27(5) :1058–1066, 2000.

[Nie03] H. Nienhuys. « Cutting in deformable objects ». Thèse de doctorat, *Utrecht University, Netherlands*, 2003.

[NMK+06] A. Nealen, M. Muller, R. Keiser, E. Boxerman et M. Carlson. « Physically based deformable models in computer graphics ». *Computer Graphics Forum*, 25(4) :809–836, 2006.

[NPF05] M. Nesme, Y. Payan et F. Faure. « Efficient, Physically Plausible Finite Elements ». Dans J. Dingliana et F. Ganovelli, éditeurs, *Eurographics (short papers)*, august 2005.

[NR01] O. Nocent et Y. Remion. « Continuous deformation energy for dynamic material splines subject to finite displacements ». Dans *Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 88–97, New York, NY, USA, 2001. Springer-Verlag New York, Inc.

[NT98] L. P. Nedel et D. Thalmann. « Real time muscle deformations using mass-spring systems ». Dans *Computer Graphics International*, pages 156–165, 1998.

[NvdS04] H. Nienhuys et A. van der Stappen. « A computational technique for interactive needle insertions in 3D nonlinear material ». Dans *IEEE International Conference on Robotics and Automation*, volume 2, pages 2061–2067, 2004.

[NZdMD05] F. Nageotte, P. Zanne, M. de Mathelin et C. Doignon. « A Circular Needle Path Planning Method for Suturing in Laparoscopic Surgery ». Dans *Proceedings of the 2005 IEEE ICRA*, pages 514–519, 2005.

[OA08] A. M. Okamura et R. Alterovitz. « MICCAI 2008 Workshop on Needle Steering : Recent Results and Future Opportunities », 2008.

[OPK+99] R. O'Toole, R. Playter, T. Krummel, W. Blank, N. Cornelius, W. Roberts, W. Bell, et M. Raibert. « Measuring and Developing Suturing Technique with a Virtual Reality Surgical Simulator ». *Journal of the American College of Surgeons*, 189(1) :1322–1327, July 1999.

[OSO04] A. Okamura, C. Simone et M. O'Leary. « Force Modeling for Needle Insertion into Soft Tissue ». *IEEE Transactions on Biomedical Engineering*, 51(10) :1707–1716, 2004.

[OZH00] J. O'Brien, V. Zordan et J. Hodgins. « Combining active and passive simulations for secondary motion ». *Computer Graphics and Applications, IEEE*, 20(4) :86–96, Jul/Aug 2000.

[Pai02] D. K. Pai. « STRANDS : Interactive Simulation of Thin Solids using Cosserat Models ». *Computer Graphics Forum*, 21(3) :347–352, 2002.

[PB88a] J. C. Platt et A. H. Barr. « Constrained differential optimization of neural networks ». Dans *IEEE Neural Information Processing Systems*, pages 612–621, 1988.

[PB88b] J. C. Platt et A. H. Barr. « Constraints methods for flexible models ». *SIGGRAPH Comput. Graph.*, 22(4) :279–288, 1988.

[PBP96] E. Promayon, P. Baconnier et C. Puech. « Physically-Based Deformations Constrained in Displacements and Volume ». *Computer Graphics Forum*, 15(3) :155–164, 1996.

[PCS+06] T. Podder, D. Clark, J. Sherman, D. Fuller, E. Messing, D. Rubens, J. Strang, R. Brasacchio, L. Liao, W. Ng et others. « In vivo motion and force measurement of surgical needle intervention during prostate brachytherapy ». *Medical physics*, 33 :2915, 2006.

[PDA00] G. Picinbono, H. Delingette et N. Ayache. « Real-Time Large Displacement Elasticity for Surgery Simulation : Non-linear Tensor-Mass Model ». Dans *MICCAI '00 : Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 643–652, London, UK, 2000. Springer-Verlag.

[PDA03] G. Picinbono, H. Delingette et N. Ayache. « Non-linear anisotropic elasticity for real-time surgery simulation ». *Graphical Models*, 65(5) :305–321, 2003.

[PKA+05] M. Pauly, R. Keiser, B. Adams, P. Dutré, M. Gross et L. J. Guibas. « Meshless animation of fracturing solids ». Dans *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 957–964, 2005.

[PKZ+05] W. Park, J. S. Kim, Y. Zhou, N. Cowan, A. Okamura et G. Chirikjian. « Diffusion-Based Motion Planning for a Nonholonomic Flexible Needle Model ». Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4611–4616, 2005.

[PLDA02] G. Picinbono, J. Lombardo, H. Delingette et N. Ayache. « Improving realism of a surgery simulator : linear anisotropic elasticity, complex interactions and force extrapolation ». *The Journal of Visualization and Computer Animation*, 13(3) :147–167, 2002.

[PLK02] J. Phillips, A. Ladd et L. E. Kavraki. « Simulated Knot Tying ». Dans *In Proceedings of the IEEE International Conference on Robotics and Automation*, pages 841–846, 2002.

[PPG04] M. Pauly, D. K. Pai et L. J. Guibas. « Quasi-rigid objects in contact ». Dans *SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 109–119, 2004.

[Pro95] X. Provot. « Deformation constraints in a mass-spring model to describe rigid cloth behaviour ». Dans *Graphics interface*, pages 147–147. Citeseer, 1995.

[Prz85] J. S. Przemieniecki. *Theory of Matrix Structural Analysis*. Dover Publications, 1985.

[PS75] C. C. Paige et M. A. Saunders. « Solution of sparse indefinite systems of linear equations ». *SIAM Journal on Numerical Analysis*, 12(4) :617–629, 1975.

[R04] H. R. « Virtual Suturing for Training in Vascular Surgery ». Thèse de doctorat, *School of Computing, The University of Leeds*, 2004.

[RA04] M. Renouf et P. Alart. « Conjugate gradient type algorithms for frictional multi-contact problems : applications to granular materials ». *Computer Methods in Applied Mechanics and Engineering*, 194(18) :2019–2041 vol.194, 2004.

[RCT91] R. E. Rosenblum, W. E. Carlson et E. Tripp. « Simulating the structure and dynamics of human hair : Modelling, rendering and animation ». *The Journal of Visualization and Computer Animation*, 2 :141–148, 1991.

[RK00] D. Ruspini et O. Khatib. « A framework for multi-contact multi-body dynamic simulation and haptic display ». *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, 2 :1322–1327 vol.2, 2000.

[RKC02] S. Redon, A. Kheddar et S. Coquillart. « Gauss' least constraints principle and rigid body simulations ». *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, pages 517–522 vol.1, 2002.

[RMSG+08] A. Robinson-Mosher, T. Shinar, J. Gretarsson, J. Su et R. Fedkiw. « Two-way coupling of fluids to rigid and deformable solids and shells ». Dans *SIGGRAPH '08 : ACM SIGGRAPH 2008 papers*, pages 1–9, 2008.

[RNG99] Y. Remion, J.-M. Nourrit et D. Gillard. « Dynamic Animation Of Spline Like Objects ». Dans *Proceedings of the WSCG'1999 Conference*, pages 426–432, 1999.

[RNG00] Y. Remion, J.-M. Nourrit et D. Gillard. « A dynamic animation engine for generic spline objects ». *Journal of Visualization and Computer Animation*, 11 :17–26, 2000.

[Sau08] G. Saupin. « Vers la simulation interactive réaliste de corps déformables virtuels ». Thèse de doctorat, *Université des Sciences et Technologies de Lille*, 2008.

[SDC08] G. Saupin, C. Duriez et S. Cotin. « Contact Model for Haptic Medical Simulations ». Dans *Proceedings of ISBMS'08*, page 165, 2008.

[SDCG08] G. Saupin, C. Duriez, S. Cotin et L. Grisoni. « Efficient Contact Modeling using Compliance Warping ». Dans *Proceedings of CGI'08*, 2008.

[SK94] K. Shimoga et P. Khosla. « Visual and force feedback to aid neurosurgical probe insertion ». Dans *EMBS '94 : Proceedings of the 16th International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1051–1052, 1994.

[Smi61] O. K. Smith. « Eigenvalues of a symmetric 3x3 matrix ». *Commun. ACM*, 4(4) :168, 1961.

[SO02] C. Simone et A. Okamura. « Modeling of needle insertion forces for robot-assisted percutaneous therapy ». Dans *Proceedings of ICRA 2002*, pages 2085–2091, 2002.

[SP08] F. Shi et S. Payandeh. « On Suturing Simulation with Haptic Feedback ». Dans *EuroHaptics '08 : Proceedings of the 6th international conference on Haptics*, pages 599–608, Berlin, Heidelberg, 2008. Springer-Verlag.

[SSF08] T. Shinar, C. Schroeder et R. Fedkiw. « Two-way Coupling of Rigid and Deformable Bodies ». Dans *2008 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 95–104, juillet 2008.

[SSIF07] E. Sifakis, T. Shinar, G. Irving et R. Fedkiw. « Hybrid simulation of deformable solids ». Dans *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 81–90, 2007.

[ST07] J. Spillmann et M. Teschner. « CoRdE : Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects ». Dans *SCA '07 : Proceedings of the 2007 ACM Symposium on Computer animation*, pages 63–72, 2007.

[ST08] J. Spillmann et M. Teschner. « An Adaptive Contact Model for the Robust Simulation of Knots ». *Computer Graphics Forum*, 24(2) :497–506, 2008.

[TC90] J. A. Thingvold et E. Cohen. « Physical modeling with B-spline surfaces for interactive design and animation ». Dans *SI3D '90 : Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 129–137, New York, NY, USA, 1990. ACM.

[TcH08] L. Tang, Y. chen et X. He. « Compliant Needle Modeling and Steerable Insertion Simulation ». *Computer-Aided Desing and Applications*, 5 :39–46, 2008.

[TF88] D. Terzopoulos et K. Fleischer. « Modeling inelastic deformation : viscolelasticity, plasticity, fracture ». Dans *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, page 278, 1988.

[TGAB08] A. Theetten, L. Grisoni, C. Andriot et B. Barsky. « Geometrically exact dynamic splines ». *Comput. Aided Des.*, 40(1) :35–48, 2008.

[TGDM07] A. Theetten, L. Grisoni, C. Duriez et X. Merlhiot. « Quasi-dynamic splines ». Dans *SPM '07 : Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 409–414, New York, NY, USA, 2007. ACM.

[TGG00] M. Teschner, S. Girod et B. Girod. « Direct computation of nonlinear soft-tissue deformation ». *Proceedings of Vision, Modeling, Visualization VMV'00*, pages 383–390, 2000.

[The07] A. Theetten. « Splines Dynamiques Géométriquement Exactes ». Thèse de doctorat, *Université des Sciences et Technologies de Lille*, 2007.

[TKZ+04] M. Teschner, S. Kimmerle, G. Zachmann, B. Heidelberger, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnetat-Thalmann et W. Strasser. « Collision Detection for Deformable Objects ». Dans *Eurographics State-of-the-Art Report (EG-STAR)*, pages 119–139, 2004.

[TPBF87] D. Terzopoulos, J. Platt, A. Barr et K. Fleischer. « Elastically deformable models ». *SIGGRAPH Comput. Graph.*, 21(4) :205–214, 1987.

[TT94] X. Tu et D. Terzopoulos. « Artificial fishes : Physics, locomotion, perception, behavior ». Dans *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, page 50. ACM, 1994.

[TW90] D. Terzopoulos et K. Waters. « Physically-based facial modeling, analysis, and animation ». *Journal of visualization and Computer Animation*, 1(2) :73–80, 1990.

[TWR88] D. Terzopoulos, A. Witkin et S. Res. « Physically based models with rigid and deformable components ». *IEEE Computer Graphics and Applications*, 8(6) :41–51, 1988.

[VCMT95] P. Volino, M. Courchesne et N. Magnenat Thalmann. « Versatile and efficient techniques for simulating cloth and other deformable objects ». Dans *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, page 144. ACM, 1995.

[vdHNGT03] G. H. M. van der Heijden, S. Neukirch, V. G. A. Goss et J. M. T. Thompson. « Instability and self-contact phenomena in the writhing of clamped rods ». *International Journal of Mechanical Sciences*, 45 :161–196, 2003.

[VG98] A. Van Gelder. « Approximate simulation of elastic membranes by triangulated spring meshes ». *Journal of graphics tools*, 3(2) :42, 1998.

[VJHG08] F. P. Vidal, N. W. John, A. E. Healey et D. A. Gould. « Simulation of ultrasound guided needle puncture using patient specific data with 3D textures and volume haptics ». *Comput. Animat. Virtual Worlds*, 19(2) :111–127, 2008.

[VT00] P. Volino et N. Thalmann. « Implementing Fast Cloth Simulation with Collision Response ». Dans *Proceedings of the International Conference on Computer Graphics*, page 257. IEEE Computer Society, 2000.

[WBD+05] F. Wang, E. Burdet, A. Dhanik, T. Poston et C. Teo. « Dynamic thread for real-time knot-tying ». *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 507–508, March 2005.

[WBRB05] F. Wang, E. Burdet, V. Ronald et H. Bleuler. « Knot-tying with Visual and Force Feedback for VR Laparoscopic Training ». Dans *27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE-EMBS)*, 2005.

[WF04] X. Wang et A. Fenster. « A virtual reality based 3D real-time interactive brachytherapy simulation of needle insertion and seed implantation ». Dans *Proceedings of IEEE International Symposium on Biomedical Imaging : Nano to Macro*, pages 280–283, 2004.

[WFB87] A. Witkin, K. Fleischer et A. Barr. « Energy constraints on parameterized models ». Dans *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 225–232, 1987.

[WGW90] A. Witkin, M. Gleicher et W. Welch. « Interactive dynamics ». *SIGGRAPH Comput. Graph.*, 24(2) :11–21, 1990.

[WH04] W. Wu et P. Heng. « A hybrid condensed finite element model with GPU acceleration for interactive 3D soft tissue cutting ». *Computer Animation and Virtual Worlds*, 15(3-4) :219–227, 2004.

[WKC+06] R. Webster, J. Kim, N. Cowan, G. Chirikjian et A. Okamura. « Nonholonomic Modeling of Needle Steering ». *International Journal of Robotics Research*, 5(6) :509–525, 2006.

[WMO05] R. J. Webster, J. Memisevic et A. M. Okamura. « Design Considerations for Robotic Needle Steering ». Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3599–3605, 2005.

[WSBB08] F. Wang, E. Su, E. Burdet et H. Bleuler. « Development of a microsurgery training system ». Dans *Engineering in Medicine and Biology Society (EMBS 2008)*, pages 1935–1938, Aug. 2008.

[WUSC94] J. Westbrook, D. Uncles, B. Sitzman et L. Carrie. « Comparison of the force required for dural puncture with different spinal needles and subsequent leakage of cerebrospinal fluid ». *Anesthesia Analgesia*, pages 769–772, 1994.

[XWS+98] J. Xuan, Y. Wang, I. A. Sesterhenn, J. W. Moul et S. K. Mun. « 3-D Model Supported Prostrate Biopsy Simulation and Evaluation ». Dans *MICCAI '98 : Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 358–367, 1998.

[YC00] Q. Yu et I.-M. Chen. « A direct violation correction method in numerical simulation of constrained multibody systems ». *International Journal of Computational Mechanics*, 26(1) :52–57, 2000.

[ZGH+07] J. Zhang, L. Gu, P. Huang, J. Dworzak, F. Chen et X. Kong. « Real-time Cutting and Suture Simulation Using Hybrid Elastic Model ». Dans *Engineering in Medicine and Biology Society (EMBS 2007)*, pages 3646–3649, 2007.

[ZKX+98] J. Zeng, C. Kaplan, J. H. Xuan, I. A. Sesterhenn, J. H. Lynch, M. T. Freedman et S. K. Mun. « Optimizing prostate needle biopsy through 3D simulation ». Dans *Proceedings of SPIE*, pages 488–497, 1998.

[ZPS01] Y. Zhang, E. Prakash et E. Sung. « Real-time physically-based facial expression animation using mass-spring system ». Dans *Computer Graphics International 2001*, pages 347–350, 2001.

[ZSD+99] D. Zorin, P. Schröder, T. DeRose, J. Stam, L. Kobbelt et J. Warren. « Subdivision for modeling and animation ». Dans *SIGGRAPH 99 Course Notes*, 1999.

Annexe A

La méthode des contraintes dans SOFA

Nous allons introduire ici l'architecture logicielle qui a été utilisée pour intégrer la méthode des contraintes, présentée dans ce travail, au sein du framework de simulation SOFA. Nous invitons le lecteur à se familiariser tout d'abord avec le fonctionnement général de SOFA au travers de la documentation [12] du projet. Le code de notre travail étant en grande partie publique et facilement disponible [13], nous présenterons les modifications apportées sans reproduire le code source.

Nous avons séparé la mise en oeuvre de la méthode en quatre aspects distincts :

- **le solveur des contraintes**. Il modifie l'algorithme de traitement de la simulation de telle façon à pouvoir incorporer des contraintes hétérogènes, et incorpore l'algorithme du Gauss-Seidel.
- **les objets de *définition* des contraintes**. Suite aux appels du solveur, ils définissent les différentes contraintes de la simulation, calculent les violations et créent les objets de résolution.
- **les objets de *résolution* des contraintes**. Ce sont des objets à courte durée de vie, n'apparaissant pas dans le graphe de scène. Ils sont utilisés dans le Gauss-Seidel pour spécialiser le traitement des contraintes.
- **les *corrections* de contraintes**. Ces objets sont chargés de calculer la matrice de compliance pour les modèles auxquels ils sont associés dans le graphe, puis de leur appliquer les mouvements de correction suite à la résolution des contraintes.

Nous allons, dans les sections qui suivent, présenter plus en détails chacune de ces classes.

Le solveur des contraintes

Sous le nom de **MasterConstraintSolver**, et héritant de **MasterSolverImpl**, le solveur des contraintes est au coeur des modifications que nous avons proposées dans ce travail. Il doit être placé à la racine du graphe de scène pour chaque simulation qui veut utiliser

12. http ://www.sofa-framework.org/documentation
13. https ://gforge.inria.fr/projects/sofa/

des contraintes hétérogènes, et est générique de telle façon à pouvoir gérer tous types de contraintes sans avoir à le modifier. Sa fonction première est de redéfinir la fonction *step*, qui décrit les étapes à suivre pour mener à bien un pas de temps de la simulation.

Son fonctionnement est le suivant : il récupère lors de son initialisation la liste des corrections de contraintes (objets dérivant de **BaseConstraintCorrection**). Ensuite, et pour chaque pas de temps, il réalise plusieurs étapes :

1. Mouvement libre.
2. Collision.
3. Définition des contraintes, qui met en oeuvre plusieurs visiteurs pour appeler respectivement les fonctions suivantes :
   - *buildConstraintMatrix* des contraintes, afin de définir les degrés de libertés contraints.
   - *accumulateConstraint* des mappings, pour remonter les contraintes sur les degrés de libertés des modèles mécaniques.
   - *getConstraintValue* des contraintes, pour obtenir la violation des contraintes.
   - *getConstraintResolution* des contraintes, qui fournit les objets de résolution des contraintes pour chacune d'entre elles.
   - *getCompliance* des corrections des contraintes, afin de calculer la compliance dans l'espace des contraintes.

   Différents traitements sont réalisés entre ces appels, principalement pour mettre à jour les structures de données suivant le nombre de contraintes créées durant ce pas de temps.
4. Résolution des contraintes. L'agorithme du Gauss-Seidel utilise pour ce calcul deux vecteurs, les forces et les violations des contraintes, la matrice de l'opérateur de Delassus, ainsi que la liste des objets de résolution des contraintes. Il permet à ces objets de réaliser des précalculs via l'appel de leur fonction *init*, en leur fournissant les deux vecteurs et cette matrice. Ensuite, à chaque itération et pour chaque contrainte, c'est leur fonction *resolution* qui est utilisée pour obtenir une force de réponse en fonction de la violation calculée à cet instant du traitement. À la fin du Gauss-Seidel, *store* est appelée pour chaque objet de résolution, dans l'objectif de stocker les valeurs finales des forces de réponse pour la méthode du départ à chaud, puis ces objets sont ensuite supprimés.
5. Mouvement correctif. Les forces dues aux contraintes sont transmises aux corrections des contraintes via la fonction *applyContactForce*, lesquelles sont ensuites remises à zéro à la fin du pas de temps par l'appel de *resetContactForce*.

Les objets de définition des contraintes

Ces objets dérivant de **BaseConstraint** sont responsables de la définition des contraintes et sont placés dans le graphe de scène, soit au moment de sa conception pour les contraintes statiques, ou suite à un évènement (exemple des contraintes de contact ajoutées après détection d'une collision). Chacun d'entre eux n'est pas limité à la définition d'une seule contrainte, et ils sont conçus généralement de telle façon à gérer toutes les interactions d'un certain type entre deux modèles de la scène. Par exemple, un objet **UnilateralInteractionConstraint** est créé automatiquement pour répondre à tous les contacts intervenant entre deux objets

en collision, définissant une contrainte par point de contact. L'objet **BilateralInteractionConstraint** peut lui être utilisé pour définir un certain nombre de contraintes de points fixes entre deux modèles.

La définition des contraintes se fait au travers des fonctions suivantes, qui sont appelées par des visiteurs sur demande du solveur des contraintes :

- *buildConstraintMatrix* dans laquelle on remplit la matrice (creuse) des contraintes pour chaque modèle mécanique. À chaque degré de liberté contraint est attribué un identifiant unique, qui sera utilisé par la suite pour renseigner les différentes caractéristiques des contraintes.
- *getConstraintViolation* où le vecteur des violations des contraintes est rempli. L'objet de définition calcule alors la valeur de la violation pour chaque degré de liberté contraint, en remplissant les indices du vecteur correspondant aux identifiants attribués.
- *getConstraintResolution* pour la création des objets de résolution, qui sont placés de la même façon dans un vecteur aux indices correspondants aux contraintes. Des données utiles à la résolution sont souvent fournies à ces objets, via leur constructeur. Les contraintes s'appliquant sur plusieurs degrés de libertés n'utilisent en général qu'un seul objet de résolution, et le vecteur est alors laissé vide pour les indices correspondant aux dimensions supérieures à une.

Les objets de résolution des contraintes

Les objets de résolution, dérivés de **ConstraintResolution**, sont créés par les objets de définition de contraintes puis maintenus par le solveur des contraintes, et sont utilisés exclusivement par l'algorithme du Gauss-Seidel, afin de calculer la force de réponse de la contrainte.

Lors de la création d'un nouveau type de contrainte, on va donc créer une classe d'objet de résolution spécifique au problème, ou faire appel aux objets de résolution génériques que sont :

- **BilateralConstraintResolution** pour les contraintes bilatérales.
- **BilateralConstraintResolution3Dof** pour des contraintes bilatérales de dimension 3, telles que les contraintes de points fixes.
- **UnilateralConstraintResolution** pour les contraintes unilatérales comme les contacts sans friction.
- **UnilateralConstraintResolutionWithFriction** pour les contact avec friction, donc de dimension 3.

Les contraintes agissant sur plusieurs dimensions doivent modifier la variable membre *nbLines* de l'objet de résolution lors de sa création afin que le solveur puisse s'adapter. Il est également possible de modifier la *tolerance* de cet objet afin d'avoir un critère de convergence différent pour certaines contraintes. Dans notre travail, nous utilisons par exemple une tolérance plus basse pour les contraintes agissant sur la rotation des objets rigides.

Les objets de résolution redéfinissent alors les fonctions suivantes, qui sont appelées par l'algorithme du Gauss-Seidel :

- *resolution*. Cette fonction est la raison d'être de ces objets, son rôle étant de calculer la force de réponse de la contrainte, suivant la violation qui est calculée à chaque itération de la résolution.
- *init*. Appelée avant la résolution, elle permet à certains objets de résolution de réaliser des précalculs, tels que le calcul de l'inverse du bloc local de l'opérateur de Delassus. Pour la méthode du départ à chaud, on l'utilise pour renseigner une valeur non nulle de la force de réponse de la contrainte.
- *store*. Après la résolution, cette fonction permet principalement de sauvegarder la valeur finale de la force de la contrainte.

Les corrections de contraintes

Ces objets héritent de **BaseConstraintCorrection** et sont placés dans le graphe de scène au niveau des modèles mécaniques contraints, pour lesquels ils vont calculer la matrice de compliance puis appliquer les forces dues aux contraintes.

Deux fonctions sont utilisées principalement dans notre méthode, à savoir *getCompliance* pour demander le calcul de la compliance pour les degrés de libertés du modèle qui sont contraints, ainsi que *applyContactForce* pour transmettre au modèle les forces des contraintes.

Dans la plupart des cas, les classes existantes sont suffisamment génériques pour être utilisées sans modifications :

- **UncoupledConstraintCorrection** pour les objets rigides et les cas où on ne désire pas de relation mécanique entre deux points du même objet (se rapprochant alors du comportement avec une intégration explicite).
- **PrecomputedConstraintCorrection** utilisée surtout pour les modèles déformables de type FEM. Elle utilise une étape de précalcul, puis une mise à jour locale de la compliance par la technique du compliance warping [SDCG08].
- **LinearSolverConstraintCorrection** spécialisée pour les modèles dont la matrice de raideur est une matrice bande, par exemple le modèle de poutres que nous utilisons pour simuler le fil de suture.