

Contribution à la modélisation de l'articulation du genou : outils géométriques et cinématiques.

THÈSE

présentée et soutenue publiquement le 7 décembre 2007

pour l'obtention du

Doctorat de l'Université de Reims Champagne-Ardenne (URCA)
(Spécialité Informatique)

par

Antoine Jonquet

Composition du jury

<i>Président :</i>	M. Sylvain MICHELIN	Professeur, Université de Marne la Vallée
<i>Rapporteurs :</i>	M. Marc NEVEU	Professeur, Université de Bourgogne
	M. Laurent GRISONI	Maître de conférences HDR, Université de Lille 1
<i>Examineurs :</i>	M. François FAURE	Maître de conférences, Université J. Fourier, Grenoble
	M. Olivier NOCENT	Maître de conférences, URCA (encadrant)
<i>Directeur :</i>	M. Yannick REMION	Professeur, URCA

« *T*HE MOST EXCITING PHRASE TO HEAR IN SCIENCE, *the one*
that heralds new discoveries, is not "Eureka!" but rather, "hmm....
that's funny..." »

Isaac ASIMOV.

Remerciements

En premier lieu, je souhaite adresser de vifs remerciements aux membres de mon jury de thèse. Tout d'abord, Marc NEVEU et Laurent GRISONI pour avoir accepté d'être rapporteurs de cette thèse. Je remercie également Sylvain MICHELIN et François FAURE d'avoir bien voulu participer au jury en tant qu'examinateurs. Je les remercie tous d'avoir consacré du temps à la lecture de ce mémoire, malgré les délais serrés imposés.

Je tiens à remercier Yannick REMION pour m'avoir accueilli au sein du LERI, d'avoir dirigé ce travail et pour avoir toujours su trouver un moment à me consacrer aux moments opportuns. Je souhaite aussi remercier Olivier NOCENT pour m'avoir encadré au cours de ces dernières années, pour sa patience et son expertise mathématique. Ainsi que pour la disponibilité dont il a su faire preuve, malgré la charge croissante de ses responsabilités. Je leur adresse toute ma gratitude pour ces soirées ou week-ends passées sur mes travaux.

Un grand merci aux membres du LERI qui entretiennent cette ambiance amicale, me permettant d'arriver chaque jour au travail avec le sourire (même ces derniers jours!). À commencer par les « chefs » Herman AKDAG, Michel HERBIN, Laurent LUCAS et Yannick REMION pour encourager cette atmosphère conviviale propice à l'échange. Mais aussi à Stéphanie PREVOST, Barbara ROMANIUK, Aassif BENASSAROU, Eric BITTAR, Frédéric BLANCHARD, Jérôme CUTRONA, Eric DESJARDIN, Olivier NOCENT, Jean-Michel NOURRIT, Didier GILLARD, Philippe VAUTROT, Laurent HUSSENET. Ainsi qu'aux petits derniers du labo, Sylvia PIOTIN, Cyril DE RUNZ, Loïc NOLIN et Gilles VALETTE. Un mot particulier aurait été nécessaire pour chacun, mais la place se fait rare. Je tiens cependant à adresser un merci tout particulier à la folle équipée du « ch'wale » et autres sorties pour les goulées d'oxygène (c'est l'omerta - je donne pas de noms, ..., mais vous vous reconnaissez).

Je souhaite aussi remercier la direction, l'équipe enseignante et les techniciens du département Informatique de l'IUT de Reims pour l'accueil et les conditions de travail plus qu'agréables.

Enfin, je tiens à remercier tous ceux qui se sont intéressés, qui m'ont soutenu et encouragé au cours de cette belle aventure. À commencer par ma famille, mes parents, ma sœur et tout les autres... À ceux qui ont essayé de retenir le titre de ma thèse, m'ont offert le couvert ou ont simplement été là, dans le désordre le plus complet, Isa, Emma, Sabine, Julien, Amélie, Cyril, Lise, Jacques, Clém, Emilie, Aude, et tous ceux que j'ai oublié.

Table des matières

Table des matières	v
Table des figures	viii
1 Introduction	1
1.1 Introduction	1
1.2 Contexte scientifique	2
1.3 Organisation du mémoire	5
2 État de l’art	7
2.1 Introduction	8
2.2 L’articulation du genou	8
2.2.1 Les éléments du genou	9
2.2.2 Les mouvements de l’articulation	13
2.3 Représentation de l’articulation	17
2.3.1 Le modèle cinématique	18
2.3.2 Le modèle statique	19
2.3.2.1 Le type I	20
2.3.2.2 Le type II	20
2.3.2.3 Le type III	20
2.3.2.4 Le type IV	21
2.3.3 Le modèle dynamique	21
2.3.4 Représentation des orientations	23
2.4 Extraction et représentation des données	24
2.5 Travaux préliminaires	26
2.5.1 Le modèle cinématique	27
2.5.2 Résultats	31

TABLE DES MATIÈRES

2.6	Conclusion	32
3	Modélisation géométrique	35
3.1	Introduction	35
3.2	Paramétrisation	37
3.2.1	Paramétrisation de surfaces ouvertes	37
3.2.1.1	Paramétrisation de la frontière	38
3.2.1.2	Paramétrisation de l'intérieur	39
3.2.2	Paramétrisation de surfaces fermées	46
3.2.2.1	Paramétrisation sphérique	46
3.3	Interpolation	49
3.3.1	Surface paramétrique à base radiale	50
3.3.2	Partition de l'unité	55
3.3.3	Interpolation sphérique	61
3.4	Conclusion	63
4	Animation dynamique	65
4.1	Introduction	65
4.2	Simulation mécanique de l'articulation du genou	66
4.2.1	Modèles dédiés	66
4.2.2	Méthode des éléments finis	67
4.2.3	Les corps rigides poly-articulés de Newton/Euler	68
4.2.4	Formulation lagrangienne	70
4.2.4.1	Mouvement non-contraint	71
4.2.4.2	Mouvement contraint	71
4.3	Gestion des contraintes unilatérales	74
4.3.1	Travaux précédents	74
4.3.2	Contraintes simultanées	77
4.3.2.1	Une première approche	77
4.3.2.2	L'algorithme correct	79
4.4	Le contact glissant	83
4.4.1	Contrainte de contact glissant	84
4.4.2	Variables libres	86
4.5	Conclusion	88

5 Conclusion générale	91
5.1 Bilan et contributions	91
5.2 Perspectives	93
Publications personnelles	95
Bibliographie	97

Table des figures

2.1	L'articulation du genou gauche	9
2.2	Plans anatomiques principaux	10
2.3	Les surfaces articulaires et les cartilages de l'articulation du genou	11
2.4	Les ménisques de l'articulation du genou	12
2.5	Les attaches de la rotule	13
2.6	Description des mouvements de l'articulation	14
2.7	Les ligaments croisés	15
2.8	Les ligaments latéraux	15
2.9	Le mouvement de roulement/glisement	16
2.10	Séance de capture de mouvements	25
2.11	Représentation simplifiée de l'articulation	27
2.12	Modélisation simplifiée du genou	28
2.13	Interface de modélisation du genou	29
2.14	Le point de contact de la partie externe	30
2.15	Resultats préliminaires	31
3.1	Construction de surface interpolante à paramétrage global	36
3.2	Paramétrisation de la frontière	39
3.3	Paramétrisation avec préservation de forme (Floater)	41
3.4	Configuration des angles autour de l'arête $\mathbf{s}_i - \mathbf{s}_{ij}$	42
3.5	Influence de la pondération sur la paramétrisation	43
3.6	Repère local à un triangle de Levy	45
3.7	Illustration des résultats de la paramétrisation	47
3.8	Exemple de paramétrisation sphérique	49
3.9	Implantation des surfaces paramétriques à base radiale	53
3.10	Illustration des résultats des SPBFs	54
3.11	Temps de calcul des coefficients de la SPBR	55
3.12	Construction du <i>quadtree</i>	57
3.13	Fonctions de distance utilisées pour les partitions de l'unité	58
3.14	Fonctions décroissantes des partitions de l'unité	59
3.15	Implantation des partitions de l'unité	60
3.16	Illustrations des résultats des partitions de l'unité	61

3.17	Temps de calcul des coefficients des partitions de l'unité	62
3.18	Schéma de subdivision et subdivision d'un octaèdre régulier unité	62
3.19	Illustrations des résultats des partitions de la sphère unité	63
3.20	Temps de calcul des coefficients des partitions de la sphère unité	64
4.1	Un exemple simple avec deux contraintes actives simultanément	76
4.2	Illustration des déficiences de l'algorithme de Platt	77
4.3	Illustration des déficiences de l'algorithme de Platt amélioré	79
4.4	Comparaison de l'algorithme de Platt et du nôtre pour une masse $m = 2$.	81
4.5	Comparaison de l'algorithme de Platt et du nôtre pour une masse $m = 3$.	81
4.6	Illustration de notre algorithme de gestion de contraintes	82
4.7	Construction d'un repère local à la surface	84
4.8	Contact de deux surfaces	85

Chapitre 1

Introduction

« THE WHITE RABBIT PUT ON HIS SPECTACLES. “Where shall I begin, please your Majesty?” he asked.
“Begin at the beginning,” the King said, very gravely, “and go on till you come to the end : then stop.” »

Lewis CARROLL, *Alice’s Adventures in Wonderland*. (1865).

Sommaire

1.1	Introduction	1
1.2	Contexte scientifique	2
1.3	Organisation du mémoire	5

1.1 Introduction

Ce travail de thèse fait partie intégrante d’un vaste projet d’animation réaliste d’humain virtuel développé au sein du groupe « Signal, Image et Connaissance » (SIC) du Centre de Recherche en Sciences et Technologies de l’Information et de la Communication (CReSTIC). Ce projet a pour ambition de mettre en œuvre des modèles informatiques efficaces pour l’animation réaliste d’avatars humanoïdes (représentation numérique d’un individu utilisée en simulation et en réalité virtuelle). Pour atteindre cet objectif, ces travaux visent à proposer un modèle d’humain virtuel paramétrable afin qu’il puisse s’adapter aux différentes morphologies. L’animation de ce modèle sera déduite de la résolution des équations dynamiques du mouvement. Les calculs nécessaires à la génération du mouvement pourront être réalisés à l’aide du Moteur d’Animation Dynamique (MADy) basé sur le formalisme lagrangien et conçu par le groupe SIC. Par ailleurs, le modèle devrait disposer de différents niveaux de complexité afin de pouvoir s’adapter aux différents niveaux de réalisme souhaités, pour la simulation ou la réalité virtuelle par exemple. L’avatar pourrait ainsi être utilisé pour simuler une articulation de manière très réaliste ou calculer

l'animation d'un mouvement global de manière rapide et cohérente.

L'idée directrice est de commencer par la modélisation d'une articulation afin de mettre au point une démarche qui puisse être appliquée ensuite au reste du squelette, notre étude se porte ainsi sur le cas particulier d'une articulation complexe : le genou.

Cette complexité est due, d'une part, aux nombreux éléments composant l'articulation : des os (fémur, tibia, rotule et péroné), des ligaments (latéraux, croisés), des ménisques et de nombreux muscles et tendons. D'autre part, cette articulation présente en effet le paradoxe de devoir satisfaire deux contraintes antagonistes : assurer une grande stabilité, pour maintenir la station verticale, ainsi qu'une grande mobilité, lors des déplacements, ce qui est permis d'une part, par une articulation très libre dans ses mouvements, et d'autre part, par des restrictions de mouvements dues à la participation des éléments non osseux de l'articulation.

Le modèle proposé ici se base sur des connaissances d'experts pour modéliser, de manière efficace et réaliste, l'interaction entre le tibia et le fémur tout en prenant en compte les restrictions dues aux ligaments. Il s'agit donc d'une première étape vers une simulation efficace de l'articulation du genou. Une fois ces résultats validés, il sera possible d'ajouter progressivement les structures manquantes, telles que le ménisque ou les muscles, pour aboutir à un modèle d'articulation réaliste et complet.

L'ampleur du travail à réaliser qui se situe à la frontière de la biomécanique et de la modélisation informatique nécessiterait plus d'une thèse. Cependant, j'ai mis en place tous les outils nécessaires à la modélisation de l'articulation du genou dans l'optique d'une simulation cinématique réaliste. Je propose ici de présenter ces outils.

1.2 Contexte scientifique

On a très tôt cherché à construire un modèle d'humain virtuel pour simuler des actions humaines à l'aide d'un ordinateur. Encore aujourd'hui, avec l'amélioration croissante du matériel informatique et des techniques d'imagerie médicale, de nombreuses études sur l'animation de créatures vivantes sont menées. Cependant, nombre de ces travaux ne s'intéressent pas au fonctionnement précis des articulations, pour axer leurs développements sur les mouvements globaux du modèle. Ces réalisations ne sont donc que rarement exploitables pour une simulation biomécanique réaliste des articulations.

Historiquement, les premiers travaux de simulation du corps humain se sont basés sur un modèle simplifié de hiérarchie de repères, pour des contraintes de temps de calcul. Ce modèle repose sur l'exploitation d'une représentation hiérarchique d'un squelette.

L'augmentation de la puissance des ordinateurs a permis ensuite d'utiliser ces modèles en temps réel. Depuis lors, le modèle pour l'animation temps réel a subi quelques améliorations mais est resté sensiblement le même, puisque l'animation produite est globalement et visuellement satisfaisante. Dans le même temps, des travaux se sont développés sur la simulation de certaines parties spécifiques du corps humain, de manière à obtenir une animation la plus exacte possible localement.

Trois principales catégories d'applications justifient l'intérêt pour la simulation du corps humain : la production de films d'animation, la simulation dont le sujet d'étude est l'être humain et le développement d'environnements virtuels comportant la représentation des participants en temps réel. En nous basant sur ces champs d'applications, nous pouvons déterminer deux types de modèles, un *modèle descriptif*, qui se limite à exploiter la cinématique produite par un animateur expérimenté ou un dispositif de capture de mouvement, et un *modèle générateur*, qui produit le mouvement à partir d'informations telles que la répartition des masses, les contraintes ou les forces extérieures.

Dans un premier temps, il est évident que pour des besoins d'animation ou d'environnements virtuels le modèle d'humain virtuel peut se contenter d'utiliser des informations cinématiques afin de se mouvoir de manière plausible. Cependant, que ce soit dans un but immersif ou pour simplifier la tâche des animateurs, il devient vite évident qu'afin d'interagir avec son environnement, le modèle doit disposer de la possibilité d'agir sur les données cinématiques de manière dynamique. Ce modèle repose sur une extension dynamique du modèle à hiérarchie de repères (appelée *ragdoll physics*), purement cinématique à l'origine. Cette méthode fonctionne parfaitement pour simuler des mouvements plausibles du corps humain, mais il n'a pas vocation à fournir des résultats exploitables par les médecins ou les biomécaniciens.

Dans le cas de la simulation, notamment médicale, où l'on cherche à quantifier ou observer un élément particulier du corps humain, les résultats obtenus doivent être les plus exacts possibles, sans véritable contrainte de temps de calcul, même s'il est préférable d'obtenir le résultat le plus rapidement possible. La construction du modèle doit donc être réalisée avec soin et exactitude pour empêcher les erreurs structurelles lors de la simulation.

Il n'existe donc pas de modèle d'humain virtuel complet réaliste, les limitations étant tant techniques que méthodologiques. Seuls existent des modèles globaux, offrant uniquement des versions très simplifiées de l'animation d'un humain : le résultat calculé en temps réel permettant une animation plausible mais non exacte. Ces modèles peuvent être compléxifiés afin d'améliorer le réalisme des résultats. Cependant les articulations et les éléments non fondamentaux, tels que les muscles, sont simplifiés pour obtenir des temps

de calcul acceptables tout en conservant une allure cohérente [HP97, CH99, MTC00]. Par ailleurs, afin d'obtenir une simulation réaliste des mouvements, les modèles sont spécialisés dans l'élément du corps humain sur laquelle porte l'étude comme par exemple l'étude de certaines articulations spécifiques telles que l'épaule [MTH96, BRR⁺02, AMFT07] ou le genou [EZT97, PD01]. Une solution intéressante semblerait être d'assembler des modèles spécifiques, localement exacts, afin d'obtenir un modèle complet réaliste. Je m'intéresserai donc, ici, plus particulièrement aux modèles spécifiques de simulation de l'articulation du genou.

Comme nous l'avons vu précédemment, cette étude prend place dans un projet à long terme visant à la modélisation biomécanique d'un corps humain virtuel pouvant être animé à différents niveaux de complexité. Ce projet a débuté avec les travaux de Gillard [Gil97] qui ont abouti à la réalisation d'un moteur d'animation dynamique permettant d'animer des objets rigides poly-articulés par l'intermédiaire de descripteurs spécifiques. Ce moteur d'animation se base sur la reformulation manuelle des équations dynamiques (équations de Lagrange [Rem00, Noc01]) afin d'obtenir automatiquement un système d'équations différentielles formellement exactes. Ainsi, une fois le modèle mécanique défini, le système d'équations décrivant le mouvement peut être généré spécifiquement. Ces travaux ont été poursuivis par l'intégration de la gestion de modèles déformables dans le moteur [Nou99, Noc01].

Fort de ce moteur d'animation dynamique, la modélisation biomécanique du genou peut alors se décomposer en deux parties :

- une modélisation géométrique permettant d'obtenir des descripteurs des différents tissus de l'articulation intervenant dans le mouvement (os, muscles, ligaments) ;
- une modélisation mécanique décrivant les différents paramètres à animer.

Dans le cadre du projet d'animation biomécanique d'un corps humain virtuel, j'ai donc entrepris la modélisation de l'articulation en créant un modèle abstrait de la structure osseuse et ligamentaire (modèle passif), de manière à pouvoir y insérer le rôle de chaque groupe musculaire (modèle actif), par la suite.

À plus court terme, ce travail s'inscrit au sein du projet régional CPER "SysReeduc" dont le but était la mise en place d'une machine d'aide à la rééducation. Il s'agit d'un projet transverse fédérant plusieurs laboratoires et faisant intervenir de nombreux acteurs tels que des informaticiens, des biomécaniciens et des automaticiens. Mon travail devait permettre de fournir aux automaticiens les courbes d'évolution du modèle à partir des informations fournies par les biomécaniciens. Mais le financement de ce projet étant arrivé avec près de deux ans de retard, les données initiales n'ont pas pu m'être fournies. Je n'ai donc pas pu disposer des données nécessaires à la mise en place pratique d'un modèle d'articulation du genou en temps requis.

Cependant, ce projet m'a permis de rencontrer des intervenants d'autres disciplines que l'informatique graphique, m'offrant ainsi un aperçu intéressant des différences de perception de la problématique de la simulation du genou.

1.3 Organisation du mémoire

L'articulation du genou étant une articulation complexe, la première tâche est donc de se familiariser avec ses différents éléments constitutifs ainsi que la manière dont ils interagissent afin de pouvoir appréhender de manière efficace leur fonctionnement. Le chapitre 2 s'ouvre donc sur une étude de l'articulation du genou.

Après avoir montré que l'articulation du genou ne peut pas être réduite à une simple liaison pivot, nous nous intéresserons aux différentes approches employées pour modéliser cette articulation dans la littérature. Ce chapitre se poursuivra par une étude de la modélisation du genou présentée lors des 11^{èmes} journées du GT Animation et Simulation [Jon04] et tirée pour partie de notre participation à l'Action Spécifique 151 (Humain Virtuel) pour les thèmes 2 « Mouvement et Contrôle du mouvement » [GJ04a] et 3 « Modélisation » [GJ04b]. Je refermerai ce chapitre sur la présentation des résultats obtenus par un travail préliminaire effectué au sein du laboratoire sur la simulation du genou. Ce travail propose de simuler les mouvements du genou en s'appuyant sur les surfaces osseuses, les résultats sont encourageants mais semblent souffrir de trop d'approximations. Lors d'une communication [JGRD03] présentant ces résultats, j'ai exprimé mon souhait de travailler directement sur les surfaces osseuses 3D afin de lever ces approximations.

Le chapitre 3 présente les solutions que j'ai développées pour la modélisation géométrique de l'articulation. Dans le modèle que je me propose de mettre en place, la modélisation géométrique se résume essentiellement aux surfaces osseuses. Pour des raisons mécaniques relatives à l'expression des contraintes, je souhaite disposer d'une surface d'interpolation à paramétrage global à partir d'un maillage triangulaire. En l'absence de solutions appropriées, j'ai proposé une méthode en deux étapes ; la paramétrisation du maillage, puis l'interpolation du maillage paramétré. La première partie du chapitre étudie les différentes techniques de paramétrisation et les solutions que j'ai retenues. La seconde partie présente une méthode originale d'interpolation des données obtenues à l'aide de fonctions à base radiale, ainsi que quelques améliorations. Ces travaux ont été présentés à la communauté lors de communications [JNR05, JNR06].

Dans le chapitre 4, je m'intéresserai à la partie mécanique de la modélisation de l'articulation. Je commencerai par étudier les différents choix offerts pour la simulation fine de

l'articulation en justifiant mon choix pour le formalisme Lagrangien. Puis je présenterai les outils que j'ai réalisés pour la mise en place du modèle mécanique. Je reprendrai les résultats que j'ai présentés à la communauté au cours d'une communication [JNR07a] sur la gestion de contraintes simultanées dans le formalisme Lagrangien. Cette communication a été sélectionnée pour être publiée dans la revue « Journal of Virtual Reality and Broadcasting » [JNR07b], après relecture et corrections mineures. Je présenterai ensuite une solution pour la gestion de contact glissant, nécessaire à une simulation réaliste des contacts osseux.

Ce document se clôturera par un bilan de ce travail, ainsi que sur les perspectives qu'il offre.

Chapitre 2

État de l’art

« “CHESHIRE-PUSS,” she began rather timidly, as she did not at all know whether it would like the name : however, it only grinned a little wider. “Come, it’s pleased so far,” thought Alice, and she went on. “Would you tell me, please, which way I ought to go from here ?”

“That depends a good deal on where you want to get to,” said the Cat.

“I don’t much care where——” said Alice.

“Then it doesn’t matter which way you go,” said the Cat. »

Lewis CARROLL, *Alice’s Adventures in Wonderland*. (1865).

Sommaire

2.1	Introduction	8
2.2	L’articulation du genou	8
2.2.1	Les éléments du genou	9
2.2.2	Les mouvements de l’articulation	13
2.3	Représentation de l’articulation	17
2.3.1	Le modèle cinématique	18
2.3.2	Le modèle statique	19
2.3.3	Le modèle dynamique	21
2.3.4	Représentation des orientations	23
2.4	Extraction et représentation des données	24
2.5	Travaux préliminaires	26
2.5.1	Le modèle cinématique	27
2.5.2	Résultats	31
2.6	Conclusion	32

2.1 Introduction

Ce travail de recherche s'inscrit dans le contexte déjà très riche des travaux antérieurs que je vais mentionner et ordonner. Je pourrai ainsi me positionner et justifier les choix effectués pour cette thèse.

Pour ce faire, je commencerai par réaliser une étude anatomique des composants de l'articulation et de leur interaction. Au travers de ces éléments, nous nous intéresserons au fonctionnement du genou, tout en définissant le vocabulaire qui sera employé tout au long de ce document.

Une fois identifiées les fonctions que nous souhaitons simuler, nous nous intéresserons aux différentes possibilités et aux choix stratégiques déjà étudiés dans de précédents travaux sur la modélisation de l'articulation du genou. Je poursuivrai ensuite cette étude par l'étude préliminaire de la modélisation du genou effectuée par Delphine Deliessche lors de son stage de DEA au sein du laboratoire.

Je concluerai cette section par l'étude des résultats obtenus sur lesquels je m'appuierai pour justifier les orientations scientifiques et techniques que j'ai prises au cours de cette thèse.

2.2 L'articulation du genou

L'articulation du genou est une articulation complexe faisant intervenir de nombreux paramètres dans son action. Afin de l'appréhender de manière claire, nous allons commencer par étudier les différents éléments du genou, puis nous verrons comment ceux-ci interagissent pour produire les différents mouvements de l'articulation.

En effet, l'articulation du genou, outre qu'elle est la plus grosse articulation du corps humain, se compose de nombreux éléments hétérogènes, comme le montre la représentation du genou de la figure 2.1. De plus, cette articulation repose sur les interactions combinées de trois os (le tibia, le fémur et la rotule).

Par ailleurs, il s'agit aussi d'une articulation importante dans le processus de la marche ou de la posture verticale. Elle présente, en effet, le paradoxe de devoir assurer deux fonctions antagonistes : garantir une grande stabilité pour maintenir la station verticale ainsi qu'une grande mobilité lors des déplacements, ce qui est permis d'une part, grâce à une articulation très libre dans ses mouvements et d'autre part, par de nombreuses contraintes. Afin de répondre à ces nombreux impératifs l'articulation du genou se compose de nom-

breux éléments de natures diverses.

Sauf précision, les informations de cette section sont tirées de ces livres d'anatomie [Pla92b],[Che02] et [CG05].

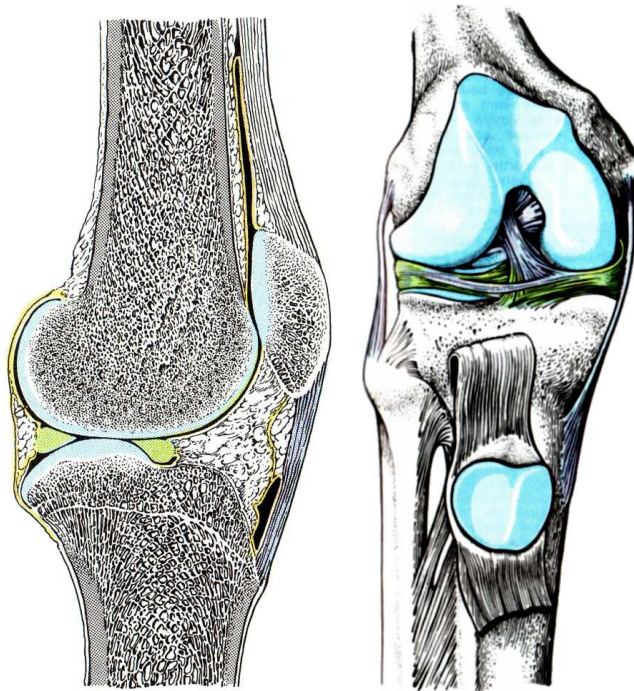


Figure 2.1 – L'articulation du genou gauche, de profil à gauche (plan sagittal) et de face à droite (plan frontal), la rotule rabattue vers le bas.

2.2.1 Les éléments du genou

Avant de commencer, nous devons définir les désignations anatomiques générales que nous utiliserons par la suite pour définir les plans principaux (figure 2.2), ils sont au nombre de trois :

Le plan frontal est perpendiculaire à la surface du sol dans la station debout et parallèle au front, on parle aussi de coupe coronale.

Le plan sagittal est perpendiculaire à la surface du sol dans la station debout et perpendiculaire au plan frontal.

Le plan transversal est perpendiculaire aux plans sagittal et frontal, il est horizontal dans la station debout, il est aussi appelé coupe axiale.

La **ligne médiane** traverse le corps dans la station debout, elle correspond à l'intersection des plans frontal et sagittal.

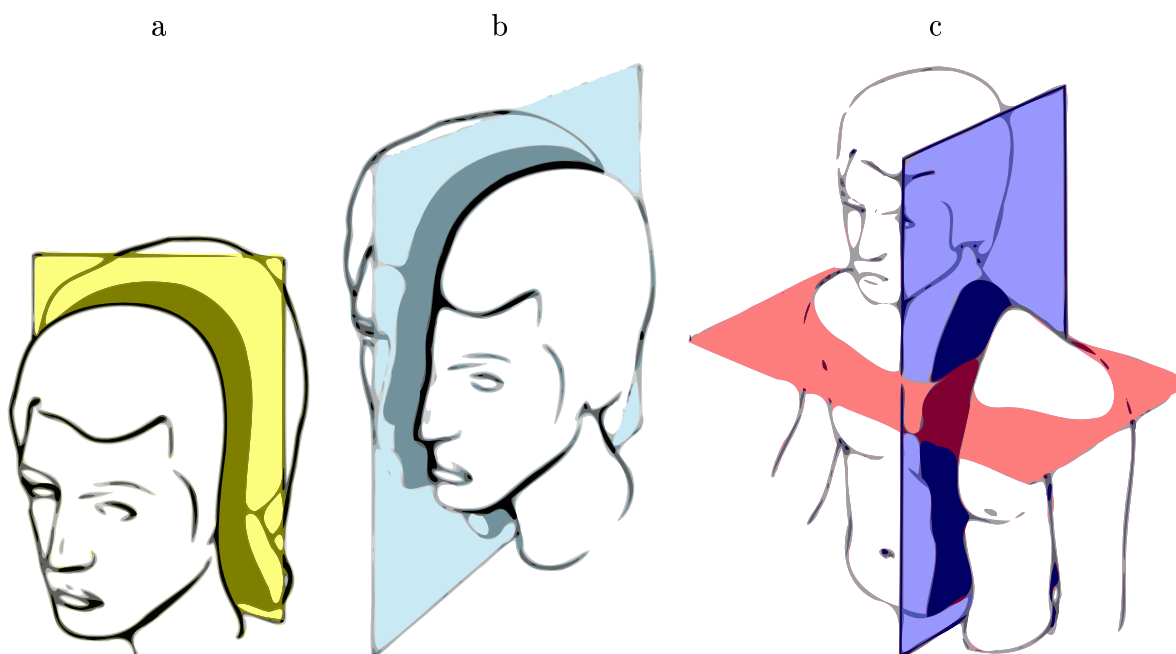


Figure 2.2 – Plans anatomiques principaux - a) frontal, b) sagittal, c) transversal et sagittal.

Comme il est souvent complexe de définir les mouvements dans l'espace, l'étude anatomique pose quelques conventions :

- Un mouvement dans le plan sagittal portant une partie du corps en avant est appelé une *flexion*, en arrière une *extension*.
- Dans le plan frontal, les mouvements portant une région du corps vers la ligne médiane ou loin de celle-ci sont appelés respectivement une *adduction* ou une *abduction*.
- On appelle *rotation interne* ou *externe* le mouvement dans le plan transversal amenant une partie du corps en dedans ou en dehors respectivement.

Nous allons maintenant nous intéresser aux éléments osseux de cette articulation. Le genou met en présence trois os : le tibia, le fémur et la rotule. On distingue deux articulations dans le genou : l'articulation entre le fémur et le tibia (*articulation fémoro-tibiale*) et l'articulation entre le fémur et la rotule (*articulation fémoro-rotulienne*). Nous allons nous intéresser pour l'instant à l'articulation fémoro-tibiale.

Les surfaces articulaires du genou sont les condyles du fémur et les plateaux tibiaux. On distingue un condyle interne et un condyle externe, ainsi qu'un plateau tibial interne et un plateau tibial externe (respectivement vers l'intérieur et l'extérieur du corps), illus-

trés sur la figure 2.3. Les deux condyles du fémur sont reliés sur la face antérieure par la *trochlée fémorale*. Pour le tibia, on peut observer dans la vue supérieure (figure 2.3 e) que les deux plateaux tibiaux sont séparés par la surface *pré-spinale* sur l'avant et la surface *rétro-spinale* sur l'arrière. Ces deux surfaces se rejoignent pour former l'*éminance intercondylienne*.

Les imperfections de ces surfaces sont compensées par un épais revêtement cartilagineux représenté en bleu sur la figure 2.3.

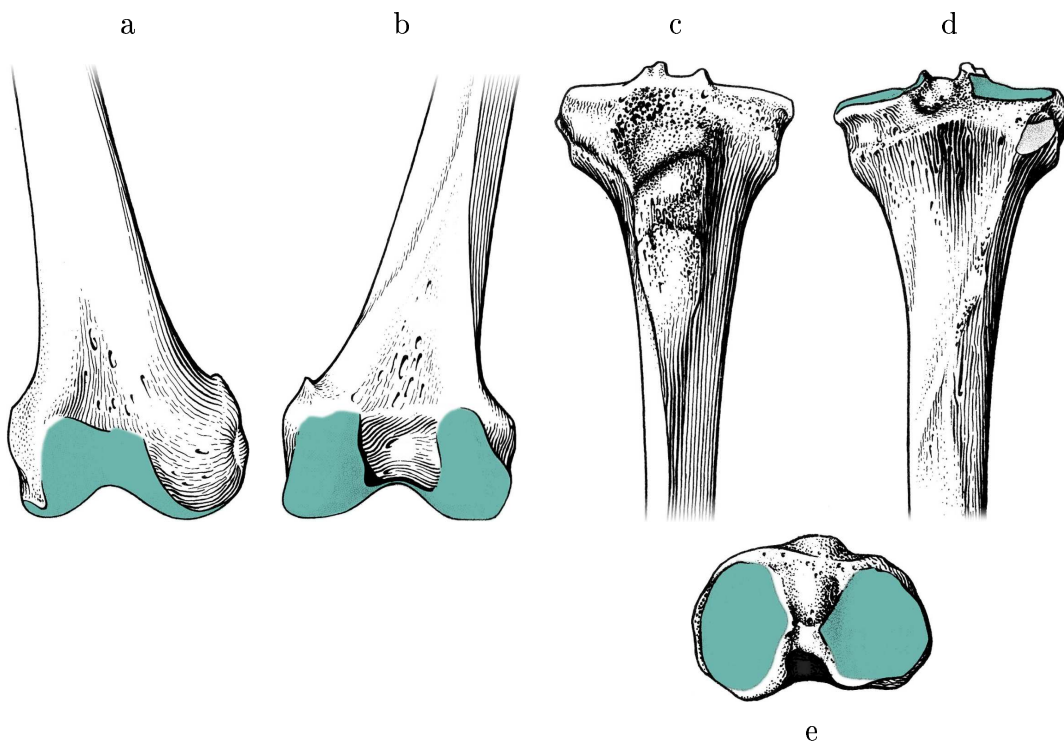


Figure 2.3 – Les surfaces articulaires - a) Fémur droit, vue antérieure, b) vue postérieure. - Tibia droit, c) vue antérieure, d) vue postérieure, e) vue supérieure. - En bleu le cartilage des surfaces osseuses.

Au cours de son action, le genou doit supporter l'ensemble du poids du corps, la charge doit donc être efficacement répartie sur la zone de contact, les ménisques jouent donc le rôle de tampons entre les surfaces osseuses. Les ménisques sont deux lamelles de cartilage fibreux en forme de croissant, et sont fixés sur les aires intercondylaires par leurs deux extrémités, comme l'illustre, en vert, la figure 2.4.

Puisque les ménisques sont attachés seulement par leurs extrémités, ils se déplacent lors des mouvements du genou. Ils avancent en extension et reculent en flexion, poussés par les condyles. Lors des rotations, l'un des ménisques avance, tandis que l'autre recule dans le sens de la rotation. Les ménisques ont une fonction triple :

- Ils améliorent la répartition des pressions en augmentant la surface de contact.

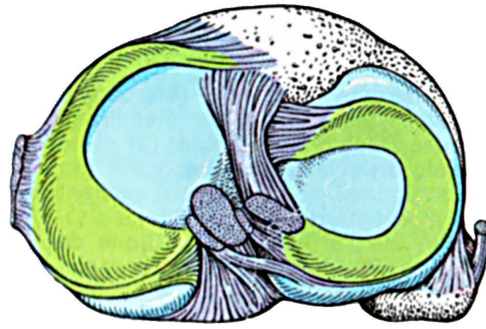


Figure 2.4 – Les ménisques de l'articulation du genou (en vert).

- Ils améliorent la stabilité en augmentant la concavité des plateaux tibiaux.
- Ils améliorent la répartition du liquide synovial par leurs mouvements.

Le liquide synovial est un liquide qui tapisse certaines surfaces de l'articulation et contribue à améliorer le glissement entre les surfaces osseuses.

La rotule est un petit os, situé en avant de la base du fémur et enchâssé dans le tendon du quadriceps. La surface articulaire intervenant dans l'*articulation fémoro-rotulienne* se trouve sur sa face antérieure. On y distingue deux gorges séparées par une crête saillante venant s'insérer dans la *trochlée fémorale*. La rotule est rattachée au genou par un ensemble de ligaments (illustré par la figure 2.5) :

- aux condyles fémoraux par les *ailerons rotuliens*
- aux ménisques par les ligaments *ménisco-rotuliens*
- entre le tibia et le quadriceps par les *tendons du quadriceps* et le *ligament rotulien*.

Son rôle principal est de protéger le tendon du quadriceps en faisant office de poulie sur la partie antérieure de l'articulation. Ainsi le tendon ne frotte pas sur l'articulation lors de la flexion. Ceci entraîne d'importantes contraintes de pression entre la rotule et la *trochlée* et d'étirement entre le *tendon du quadriceps* et le *ligament rotulien*.

Comme nous venons de le voir, les os du genou ne sont pas emboîtés et sa stabilité doit donc être assurée par les ligaments afin d'offrir une rotation principale assortie de mouvements secondaires.

En avant, ils sont de deux sortes : les ligaments *ménisco-rotuliens* et les *ailerons rotuliens* qui relient la rotule respectivement aux ménisques et aux condyles. les tendons du quadriceps se croisent sur la rotule puis forment le ligament rotulien.

En arrière, un *plan ligamentaire* très puissant suit la forme des condyles et empêche l'hyperextension du genou, assurant la stabilité postérieure lors de la station debout.

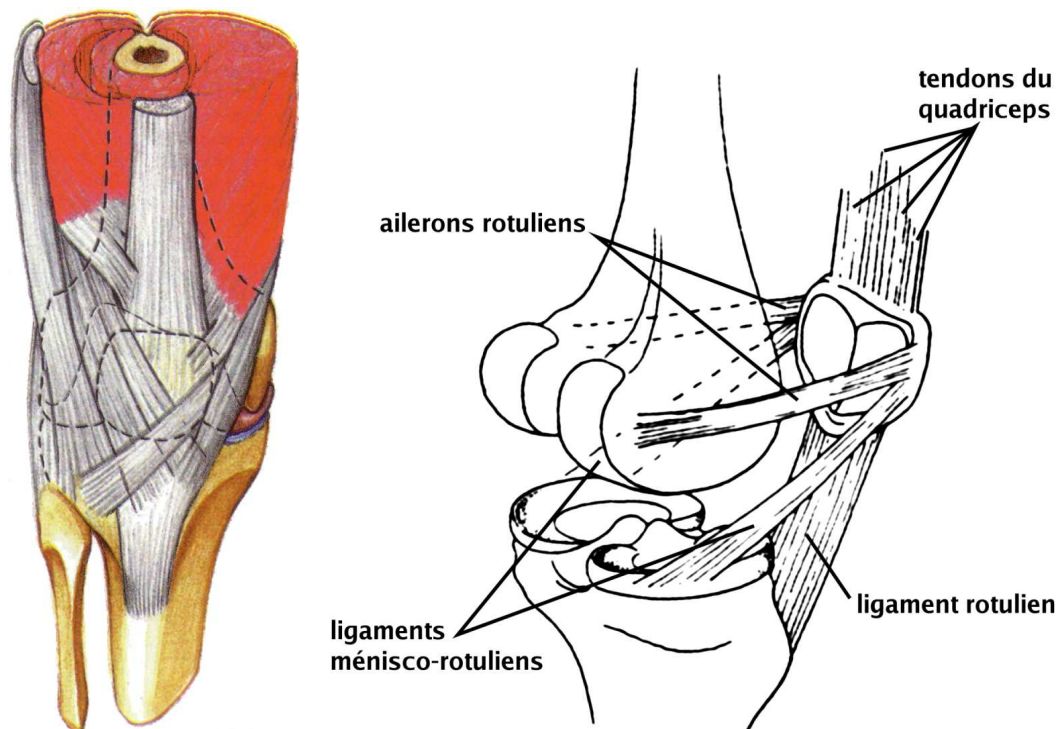


Figure 2.5 – les attaches de la rotule.

Sur les côtés, la présence des ligaments latéraux assure la stabilité latérale du genou. On distingue le ligament latéral interne à l'intérieur, et le ligament latéral externe à l'extérieur de l'articulation.

Au centre de l'articulation se trouvent les deux ligaments croisés. Ils se composent du *ligament croisé antéro-externe*, s'étendant de la surface *pré-spinale* du plateau tibial à la face interne du condyle externe du fémur, et du *ligament croisé postéro-interne*, s'attachant sur la surface *rétro-spinale* du plateau tibial à la face externe du condyle interne du fémur. Leur rôle principal est d'empêcher les mouvements de tiroirs *antéro-postérieurs* et *latéraux*, illustrés par la figure 2.6.

2.2.2 Les mouvements de l'articulation

De part sa grande complexité structurelle, l'articulation du genou propose de nombreux mouvements illustrés par la figure 2.6. Une rotation principale (la *flexion/extension*) ainsi que deux rotations secondaires : l'une *interne/externe* (ou *varus/valgus*) autour de l'axe du tibia, l'autre *adduction/abduction* dans le plan frontal. Elle présente également, genou fléchi, des mouvements mineurs de *latéralité droite et gauche* ainsi que des mouvements de tiroirs *antéro-postérieurs*.

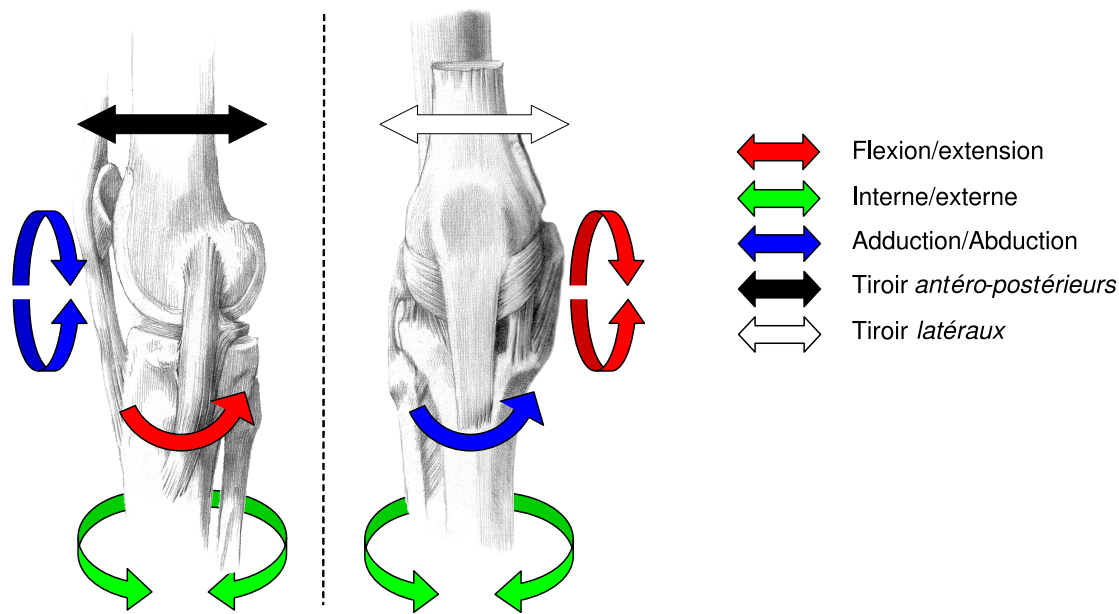


Figure 2.6 – Description des mouvements de l'articulation.

Comme nous l'avons vu précédemment, les ligaments contraignent fortement les mouvements de l'articulation. De par leur tension, les ligaments croisés empêchent les mouvements de tiroir *antéro-postérieurs* et *latéraux*. Cependant, il est intéressant de noter que la torsion des ligaments augmente lors de la rotation interne et diminue avec la rotation externe. Ainsi, les ligaments croisés se tendent ou se détendent lors des *rotations internes* ou *externes*.

Les ligaments latéraux contraignent la rotation dans le plan frontal en interdisant les baillements latéraux (ou mouvements de compression/distraktion). De plus, de par leurs implantations, ils contraignent les mouvements de tiroir *antéro-postérieurs* et *latéraux*. Par ailleurs, leurs implantations impliquent, comme l'illustre la figure 2.8, que les ligaments latéraux soient détendus en *rotation interne* et tendus en *rotation externe*. Ainsi, en action combinée avec les ligaments croisés, ils contraignent les *rotations interne/externe*. Pour finir, on peut noter qu'ils se détendent aussi lors de la flexion.

La rotule et ses ligaments conjointement au plan ligamentaire postérieur empêchent les mouvements de tiroir *antéro-postérieurs*.

Donc en extension, la majorité des ligaments sont tendus et le genou est normalement stabilisé par la tension ligamentaire. Tandis qu'en flexion, les ligaments sont presque tous détendus et le genou peut donc effectuer des mouvements secondaires.

Par ailleurs, au cours du mouvement de *flexion/extension* se produisent de légères

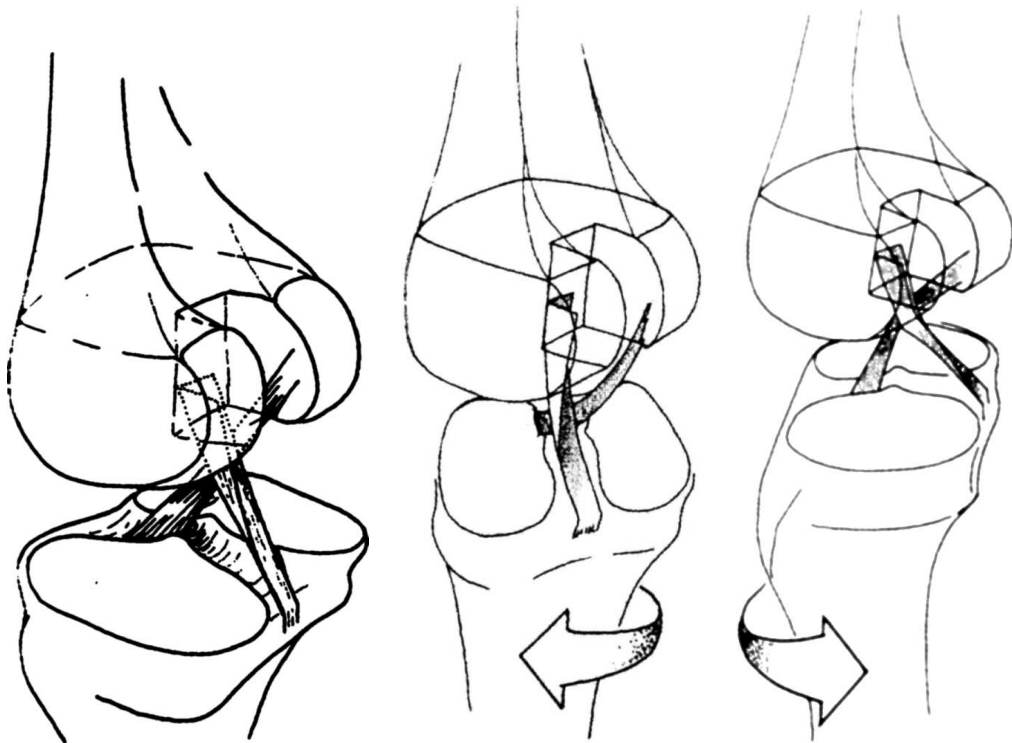


Figure 2.7 – Les ligaments croisés.

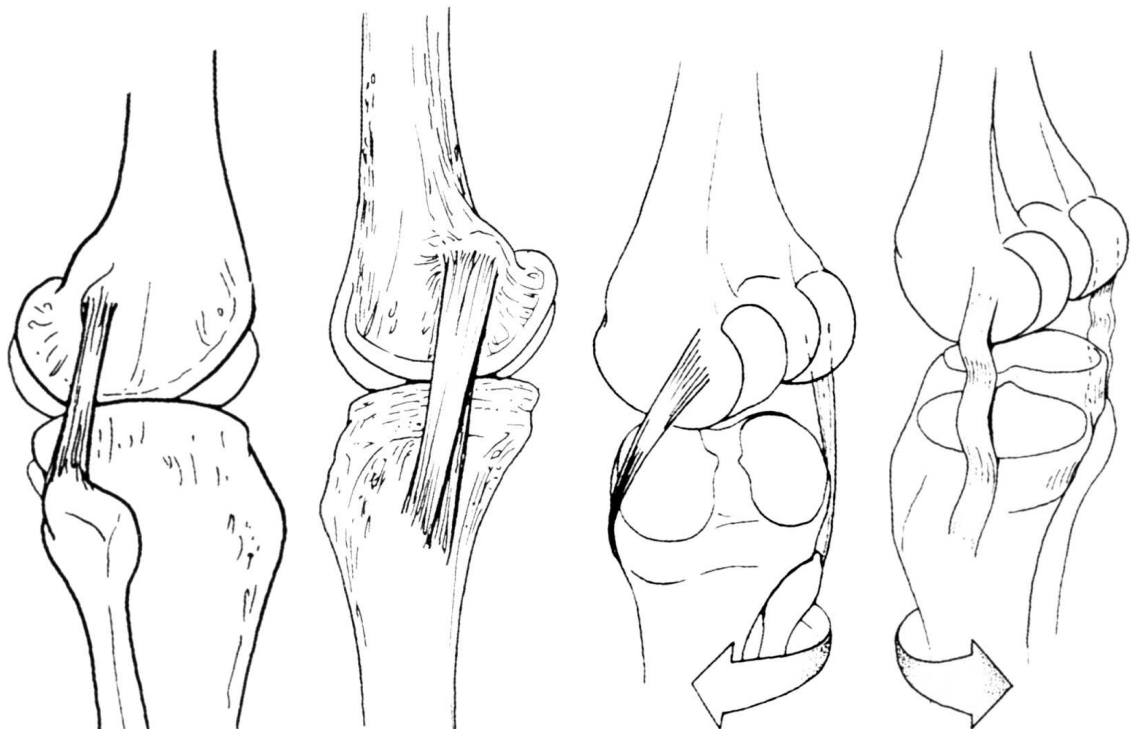


Figure 2.8 – Les ligaments latéraux.

rotations automatiques entre le fémur et le tibia.

Ces mouvements accordés par l'articulation sont intrinsèquement dépendants de la surface osseuse intervenant dans l'articulation.

On distingue deux types de flexion, la flexion active d'une amplitude inférieure à environ 130° et la flexion passive pouvant atteindre à peu près 150° .

Lors de la flexion active, les muscles fléchisseurs se contractent et forment une masse musculaire à l'arrière de l'articulation et limitent la flexion. La flexion passive est plus ample puisque les muscles fléchisseurs sont détendus et se laissent alors écraser l'un sur l'autre.

Dans tous les cas, le mouvement des condyles fémoraux sur les plateaux tibiaux est constitué d'une combinaison de *roulement/glisement*, illustré par la figure 2.9. La première figure illustre la position initiale d'un condyle fémoral sur un plateau tibial, avec une représentation des points de contact en noir. Dans les figures suivantes, les points de contact initiaux sont conservés en gris et les trajectoires des points de contact sont représentées par des flèches rouges. Lors du mouvement de *roulement* (b), le point de contact du condyle fémoral parcourt la même distance que le point de contact du plateau tibial, tandis que dans le cas du *glissement* (c), seul le point de contact du condyle fémoral se déplace. Le mouvement de *roulement/glisement* (d) est une combinaison des deux dans laquelle le point de contact du condyle fémoral parcourt une distance supérieure au point de contact du plateau tibial. Au cours de la marche normale, les condyles commencent par rouler sur à peu près 15° puis glissent et roulent, jusqu'à environ 100° de flexion, où ils finissent par glisser sans rouler. Nous rappelons que ces informations proviennent du livre [CG05]

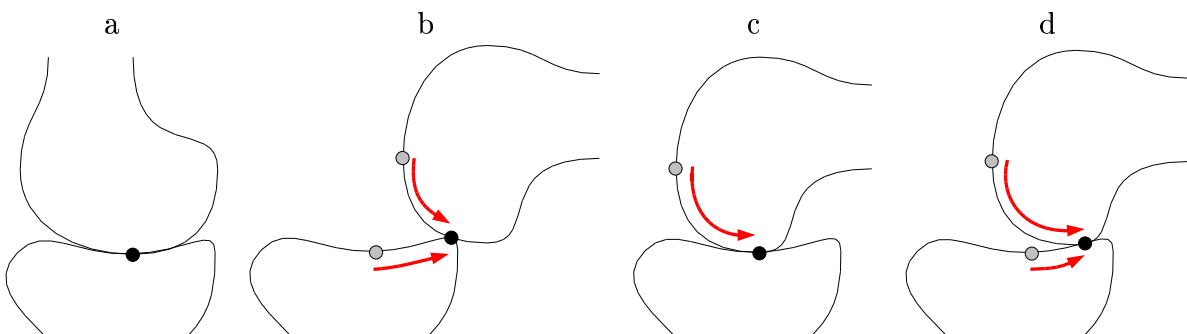


Figure 2.9 – Illustration du mouvement de roulement/glisement d'un condyle fémoral sur un plateau tibial - a) position initiale, b) roulement, c) glissement, d) roulement/glisement

Dans le plan sagittal (figure 2.1), on constate que les condyles fémoraux présentent

un rayon de courbure qui diminue vers l'arrière, c'est à dire que le condyle est plus plat à l'avant (plus grande surface pour la posture verticale) et plus arrondi à l'arrière (meilleur développement du mouvement de flexion). Cependant, les deux condyles n'ont pas la même courbure, le rayon de courbure du condyle interne est supérieur à celui du condyle externe.

La surface du plateau tibial interne est concave, tandis que le plateau tibial externe présente une forme en selle dont la gouttière va d'avant en arrière. Ceci explique en partie les rotations automatiques s'effectuant au cours du mouvement de *flexion/extension*. L'autre raison étant que le condyle interne est plus maintenu, puisque le ligament latéral interne est moins élastique que le ligament latéral externe.

2.3 Représentation de l'articulation

Comme Bull et Amis l'expliquent [BA98], il existe de nombreuses manières de représenter l'articulation du genou. Cependant, certaines d'entre elles se révèlent inefficaces pour une modélisation réaliste du genou. Si les techniques plaçant l'articulation du genou dans un espace 2D étaient justifiées lorsque la puissance des ordinateurs n'était pas suffisante pour résoudre efficacement des systèmes 3D, elles sont aujourd'hui inadaptées pour une interprétation correcte du mouvement de la marche.

Bien que la simulation 3D de l'articulation du genou soit l'axe de recherche principal actuellement, l'utilisation de la 2D pour la simulation reste toutefois utile, pour effectuer des simulations simplifiées, permettant ainsi une validation méthodologique.

Dans le cas de la simulation d'une articulation, on peut répertorier trois types de modèle : cinématique, statique et dynamique :

Le modèle cinématique permet d'exprimer l'espace des évolutions possibles en position, vitesse et accélération des éléments constitutifs de l'articulation. Ce modèle ne prend pas en compte les données relatives à la masse et aux efforts extérieurs.

Le modèle statique relie les conformations stables décrites par le modèle cinématique (position et orientation), aux efforts qui les maintiennent (efforts dont la somme des contributions est nulle). On utilise des modèles quasi-statiques pour simuler les effets du mouvement.

Le modèle dynamique , quant à lui, relie les évolutions des conformations (vitesse et accélération) aux efforts et à l'inertie.

Je vais détailler ces modèles en étudiant les différentes approches employées pour modéliser l'articulation du genou dans des travaux précédents. Pour finir, je discuterai

rapidement des différentes représentations des orientations disponibles pour qualifier les rotations de l'articulation.

2.3.1 Le modèle cinématique

Comme nous l'avons vu précédemment, le modèle cinématique s'intéresse uniquement à la position, la vitesse et l'accélération des éléments de l'articulation.

Au vu de la complexité des mouvements secondaires, il serait réducteur de ne considérer qu'une seule rotation (la flexion/extention). Dès lors, il faut caractériser les mouvements qui nous intéressent.

Le mouvement de l'articulation du genou peut être caractérisé par trois rotations et trois translations. Les axes de ces trois rotations sont généralement considérés comme dissociés.

Le premier impératif est de localiser l'axe de flexion/extension. Shino [Shi61] avance que cet axe se déplace d'à peu près 1 cm pour 130 ° de rotation. Frankel et al. [FBB71] approfondissent ces déclarations en proposant que le centre de rotation se déplace sur le plan de la rotation sous la forme d'une spirale. Ce qui semble être confirmé par la littérature au vue de l'évolution de la courbure des condyles.

De leur côté, Crowninshield et al. [CPJ76] avèrent les déclarations de McLeod et al. [MC74] selon lesquelles, le centre de rotation se trouverait dans le condyle fémoral externe, proche de la surface du plateau tibial lorsque l'articulation est en pleine extension et se déplacerait petit à petit vers l'arrière lors de la flexion. Smidt [Smi73] utilise une translation dans la région du condyle fémoral externe pour simuler ce déplacement tandis que Lewis et Lew [LL77, LL78] utilisent déjà un modèle à 6 degrés de liberté définissant un axe unique optimal.

Crowninshield et al. [CPJ76] décrivent dans leur article le mouvement de rotation *interne/externe* comme deux rotations d'axe antéro-postérieur se situant respectivement dans les condyles interne et externe.

En opposition, Seering et al. [SPNS80] proposent un axe unique de rotation situé dans le fémur et passant par un point situé à moins de 0,5 cm et à mi-chemin des deux points de la surface des condyles fémoraux les plus distants possibles. Le centre de la rotation *interne/externe* est situé par Mosely [Mos53] au centre du condyle interne, il quantifie cette rotation d'à peu près 30 ° pour une flexion de 130 °. Wang et al. [WWW73] et Crowninshield et al. [CPJ76] corroborent cette assertion en localisant le centre de cette rotation

sur la partie interne du genou mais Wang et al. spécifient la rotation comme une fonction de l'angle de flexion/extension.

En opposition, Seering et al. [SPNS80] utilisent un axe perpendiculaire au plateau tibial, passant par le point médian des centres des plateaux tibiaux interne et externe. Wang et Walker [WW74] quant à eux, définissent l'axe de rotation comme perpendiculaire à la crête de l'*éminance intercondylienne*.

Selon Goodfellow et O'Connor [GO78], les mouvements de tiroirs antéro-postérieurs doivent apparaître dans un axe antéro-postérieur défini par la crête de l'éminence intercondylienne, et parallèle au plateau tibial dans le plan sagittal. Edwards et al. [ELL70] et Crowninshield et al. [CPJ76] expriment ces mouvements dans un axe reposant sur le plan défini par les contacts osseux, selon l'insertion fémorale des ligaments croisés et passant par le centre du plateau tibial.

Les mouvements de tiroir latéraux, selon Morrison [Mor70], sont définis sur un axe de translation passant par les points de contact les plus distants, passant au milieu et perpendiculairement à la crête de l'*éminance intercondylienne*.

Enfin, pour les mouvements de compression/distraction, Edwards et al. [ELL70] et Crowninshield et al. [CPJ76] emploient l'axe longitudinal du tibia passant par le centre du plateau tibial, alors que les résultats de Yoshioka et al. [YSSC89] suggèrent que la moyenne des axes de son jeu d'étude diffère de 2° de l'axe du tibia. Edwards et al. [ELL70] étudient la cinématique de l'articulation du genou pour déterminer la longueur des ligaments latéraux et croisés. Ce système comporte 3 DdL : l'angle de flexion dans le plan sagittal, la rotation tibiale dans le plan transversal et la translation de compression/distraction.

Au vu de ces nombreuses études, on peut constater que la construction du modèle cinématique de l'articulation du genou reste un problème ouvert et qu'il n'existe pas encore de solution réellement convaincante.

2.3.2 Le modèle statique

Le modèle cinématique dépend des informations et des connaissances que l'on a sur le fonctionnement de l'articulation que l'on souhaite étudier. Pour se dégager de cette contrainte, certains chercheurs ont souhaité modéliser l'articulation en se basant sur les informations géométriques des éléments de l'articulation pour les faire interagir en y appliquant des forces internes et externes. Dans le but d'obtenir une description cohérente de l'articulation, ils ont commencé par modéliser l'articulation en recherchant la position d'équilibre entre les forces internes/externes (modèle statique) puis ont simulé le mouve-

ment en introduisant un déséquilibre (modèle semi-statique). Hefzy and Grood [HG88] ont classifié ces modèles en quatre principaux groupes :

Le type I regroupe ceux qui sont utilisés pour déterminer la répartition des forces dans les muscles et ligaments lors d'efforts exercés par l'articulation.

Le type II représente les systèmes dans lesquels les forces de chaque ligament sont représentées comme des fonctions dépendantes de l'articulation.

Le type III regroupe les systèmes principalement utilisés pour déterminer les forces de frottement/glissement aux points de contact.

Le type IV se compose des systèmes les plus complexes incluant à la fois les contraintes ligamentaires et géométriques.

2.3.2.1 Le type I

Aucune implémentation graphique d'un modèle de *type I* n'a été faite, cependant il est possible de la concevoir à partir de Hefzy [HG88].

2.3.2.2 Le type II

Les premiers modèles de *type II* ont été mis en place par Crowninshield et al. [CPJ76]. Crowninshield et al. [CPJ76] ont poursuivi les travaux de Edwards et al. [ELL70] en présentant le premier modèle analytique de genou prédisant les réponses de l'articulation soumise à des forces externes ou à des déplacements. Ce modèle a été obtenu en ajoutant 2 DdL : mouvement de *varus/valgus* et mouvement *antéro-postérieur* ainsi qu'en complexifiant la représentation des ligaments. La tension de l'articulation est calculée par la méthode inverse, aussi utilisée par Wismans [WVJ⁺80]. Les prédictions obtenues ont été comparées avec des données expérimentales, les résultats sont considérés comme acceptables pour les mouvements de *varus/valgus* et de rotation latérale, mais pas pour les mouvements *antéro-postérieurs*.

Dans un premier article, Grood et Hefzy [GH82] évaluent la contribution des ligaments aux efforts du genou en utilisant une matrice d'analyse structurelle. L'avantage de cette méthode étant que l'effort appliqué à l'articulation est calculé directement à partir des équations du modèle. Dans leur second article, Hefzy et Grood [HG83] ont étendu leur travail en incorporant une description des ligaments plus réaliste.

2.3.2.3 Le type III

Selon Hefzy et Grood [HG88] le modèle de *type III* a initialement été développé pour évaluer les variations d'effort du genou selon l'ostéotomie du tibia. Plusieurs modèles [Min81, DB78, MP77] ont été mis en place, mais tous sont dans une optique calcul formel

reposant sur des courbes et des fonctions.

2.3.2.4 Le type IV

Il n'existe à ce jour que trois modèles de *type IV*, ceux de Wismans et al. [WVJ⁺80], Andriacchi et al. [AMHG83] et Blankevoort et al. [BHD84]. Wismans et al. [WVJ⁺80] proposent un modèle 3D quasi-statique du genou. Le modèle comprend les structures ligamentaires et capsulaires ainsi que la représentation géométrique 3D de l'articulation, permettant ainsi d'exprimer la position du fémur par rapport au tibia en fonction de l'angle de flexion. La tension et l'étirement des ligaments ainsi que la position des deux points de contact, la direction et l'intensité des forces aux points de contact sont aussi désignées comme des fonctions de la flexion.

Andriacchi et al. [AMHG83] présentent un modèle 3D utilisant un modèle de corps rigide pour le fémur et le tibia, et un modèle d'éléments finis pour les ligaments, ménisques et cartilage. Les rotations secondaires ont été prédites, puis quantitativement comparées aux valeurs expérimentales présentées par Markolf et al. [MMA76].

Ils ont découvert qu'imposer des contraintes au mouvement déplace l'axe de rotation suffisamment pour altérer la réponse du genou. Ils en ont déduit qu'il était très important de pouvoir simuler un axe de rotation mobile dans le modèle du genou et concluent que les applications utilisant un axe de rotation fixe imposent un effort non naturel à l'articulation.

2.3.3 Le modèle dynamique

Si le but de l'étude est la construction d'un système permettant de simuler de manière efficace le comportement d'une articulation en fonction d'une grande variété de contraintes, alors les modèles statiques ou quasi-statiques, considérant des restrictions artificielles sur l'articulation sont inadéquats.

Il faut donc étendre l'étude faite sur ces modèles pour pouvoir gérer dynamiquement les contraintes que l'on souhaite y appliquer. On distingue deux types de modèles : 2D [TE93, ARH93, WHB84, MEA83] et 3D [KAC88, MEA83].

Tumer et Engin [TE93] ont développé un modèle 2D du genou, incorporant à la fois l'articulation *patello-fémorale* et *tibio-fémorale*. Le modèle permet la simulation de la réponse à la contraction de certains muscles de l'articulation et à l'application de forces externes. Le même jeu de muscles que Wongchaisuwat et al. [WHB84] a été modélisé.

Abdel-Rahmen et Hefzy [ARH93] ont construit leur modèle comme deux corps rigides connectés par cinq ligaments. Le fémur est considéré comme fixe tandis que le tibia est libre dans le plan sagittal. L'articulation est considérée comme sans frottement avec un seul point de contact. Les forces externes appliquées sont le poids de la jambe, s'appliquant au centre de gravité du tibia comme défini par Moeinzadeh et al. [MEA83], ainsi qu'une force de charge s'appliquant elle aussi au centre de gravité du tibia dans un axe perpendiculaire à l'axe du tibia et dirigé vers l'arrière. Ils obtiennent ainsi trois équations de mouvement, deux équations d'accélération selon l'axe x et y et une équation d'accélération angulaire. Trois autres équations sont ajoutées pour résoudre le point de contact, soit six équations pour six inconnues. Ils ont ainsi pu démontrer, en accord avec la littérature, que pour une flexion de 20 à 90°, les efforts inverses sont essentiellement fournis par le ligament croisé postérieur et le ligament collatéral interne.

Wongchaisuwat et al. [WHB84] s'intéressent au mouvement de roulement/glisement qui s'opère entre deux os. Pour se faire, ils modélisent le mouvement dans la coupe sagittale, et considèrent le tibia comme un pendule ne pouvant perdre le contact avec le fémur. Ils introduisent ensuite des efforts pour simuler les ligaments et les efforts fournis par les muscles, formulent les équations des mouvements résultants dans un système Newton-Euler puis résolvent le système. Les résultats obtenus sont difficilement comparables à la littérature à cause de différences trop importantes entre les modèles et l'expérience. Mais Wongchaisuwat et al. présentent leurs prédictions de réponse des ligaments comme très correctes.

Moeinzadeh et al. [MEA83] ont développé un modèle en 2D du genou constitué de deux corps rigides attachés ensemble par la représentation des ligaments croisés et collatéraux. Le fémur est considéré comme fixe tandis que le tibia est mobile dans un plan. Le but de l'étude était de démontrer les effets de l'application d'une force externe sur le tibia plutôt que d'effectuer une simulation paramétrée. Leur modèle a permis de constater qu'en extension, les ligaments collatéraux et croisé antérieur s'allongent tandis que le croisé postérieur se contracte lors de l'application d'une charge sur le tibia.

L'essentiel des travaux que je viens de citer s'intéressent à la modélisation du genou sous un aspect continu. En effet, il n'était pas possible à l'époque de gérer de gros volumes de données. Un effort particulier a donc été fait pour obtenir une approximation géométrique des surfaces osseuses et de leurs interactions. Cette approche est intéressante puisqu'elle permet d'obtenir des informations importantes sur la caractérisation et les relations entre les différents éléments de l'articulation, cependant dans le souci d'obtenir une simulation personnalisable, il semble nécessaire à l'heure actuelle d'utiliser des informations réelles.

D'autre part, nombre de ces travaux tentent d'obtenir une simulation de l'action du genou en contraignant directement les rotations sur leur axe. Or comme le souligne Markolf et al. [MMA76], il est nécessaire de simuler des axes de rotation mobiles afin de ne pas imposer d'efforts non naturels à l'articulation.

C'est pourquoi nous souhaitons introduire un modèle disposant d'une représentation 3D de ces surfaces osseuses, reposant sur des données réelles. L'interaction entre ces surfaces devra être libre et précise tandis que leur mouvement relatif sera contraint de manière complexe.

2.3.4 Représentation des orientations

Afin d'orienter les éléments dans l'espace, nous avons besoin de définir un système de coordonnées, cinq sont couramment utilisés : les angles projetés, les angles d'Euler, les angles de Cardan et Bryant, les quaternions et les angles duaux.

Le système des angles projetés est l'approche adoptée par Sutherland et al. [SOBW88] dans l'étude qu'ils ont effectuée sur la marche. Cette technique consiste à projeter les segments représentant les angles locaux sur les plans XY, YZ et ZX d'un repère global fixe afin d'obtenir les angles dans un ensemble de référence. Cette méthode est simple et facile à mettre en place mais la cinématique 3D obtenue est instable et approximative. Elle se révèle même erronée dans le cas d'une rotation supérieure à 5° .

L'utilisation des angles d'Euler et de Cardan-Bryant consiste à paramétrer le mouvement de rotation en fonction des axes d'une base orthogonale $\{i, j, k\}$. Les angles d'Euler sont définis par la rotation autour de la séquence d'axes kjk [LBIB48, DOTG91, RK91] alors que les axes de Cardan-Bryant sont définis par la séquence de rotation ijk [EGW88, MEA83]. Le résultat est évidemment dépendant de la séquence mais les variations restent faibles dans le cas de rotations de petites amplitudes.

Le principal défaut, appelé « gimbal lock », de ces méthodes est la singularité qui fait que de petites variations dans la matrice de rotation provoquent de grandes variations dans la rotation finale lorsque la rotation autour de j approche $\pi/2$, ainsi qu'un blocage complet des rotations lorsque l'angle de rotation autour de j est égal à $\pi/2$. Les autres cas se partagent en deux groupes, connus sous le nom de « Codman's Paradox » [MI91].

Afin de se soustraire à ces limitations, il est possible d'utiliser les quaternions unitaires. Ceux-ci permettent de caractériser des rotations de l'espace autour d'un axe quelconque [Sho85, Han06]. Ils sont définis comme une extension de l'ensemble des nombres complexes : l'espace des quaternions étant composé d'une direction réelle et de trois directions

imaginaires indépendantes. L'ajout d'un paramètre et d'une contrainte supplémentaire permet de résoudre le problème de « gimbal lock » et offre une solution efficace pour l'interpolation de deux rotations quelconques, ce qui nous intéresse particulièrement dans le cadre de l'animation.

Les angles duaux [YK01, YK02] partent du même principe que les quaternions, et ajoutent une composante supplémentaire à chaque rotation. Ainsi, une translation est liée à chaque rotation afin de pouvoir caractériser efficacement des rotations de l'espace autour d'un axe quelconque, tout en gardant un formalisme naturel.

Nous avons choisi d'utiliser pour notre simulation les quaternions, puisqu'il s'agit de la solution la plus couramment utilisée dans la communauté de l'informatique graphique pour éviter les problèmes de « gimbal lock » et obtenir une caractérisation efficace des rotations de l'espace. Les angles duaux auraient pu être une autre solution viable mais leur utilisation reste malgré tout marginale et anecdotique.

2.4 Extraction et représentation des données

Comme nous venons de le voir, les solutions pour la modélisation sont nombreuses. Néanmoins, quelle que soit la solution méthodologique envisagée il est nécessaire de tenir compte des données qui seront utilisées par le modèle pour l'animation. Afin d'obtenir le résultat le plus exact possible, ces informations doivent être extraites de données réelles.

Il est intéressant de noter que l'évolution technologique en terme d'acquisition (IRM, capture de mouvement) nous offre de nouvelles perspectives pour l'élaboration et la validation de modèles anatomiques. Ainsi, la simulation biomécanique est passée d'une approche empirique « continue » reposant sur des courbes et surfaces mathématiques représentant les données réelles à une approche « discrète » basée directement sur les données réelles comme la méthode des éléments finis. Notre travail souhaite faire la jonction entre ces modèles théoriques ne reposant pas sur des données réelles et ces modèles discrets ne permettant pas d'extraire l'expression du mouvement global qu'ils simulent.

Une première étape consiste à obtenir des données géométriques et topologiques du système que l'on souhaite modéliser. L'extraction de ces données à partir de mesures réelles telles que des IRM ou des radios, repose sur des algorithmes de traitement d'information volumique. De nombreuses solutions existent, il s'agit là cependant d'un domaine de recherche toujours ouvert qui dépasse le cadre de cette thèse.

Une fois le modèle géométrique obtenu, il reste à obtenir les informations mécaniques

des éléments constituant le modèle. Les informations de mouvement telles que la vitesse, l'amplitude et la nature des transformations effectuées peuvent être obtenues par capture de mouvement ou *motion capture* [Koc02, CWZ⁺02]. Dans le cadre du projet SysReeduc, nous travaillons conjointement avec le LACM (Laboratoire d'Analyse des Contraintes Mécaniques) qui dispose d'un système d'acquisition optique VICON. Ce système se compose de plusieurs caméras proche infra-rouge, disposées de manière à ce que les marqueurs disposés sur le sujet d'étude soient toujours visibles par plusieurs d'entre elles. Il est ensuite possible de reconstruire le parcours 3D des marqueurs à partir de ces différentes vues. La figure 2.10 illustre une séance de capture de mouvement au LACM. On distingue très nettement les petites sphères réfléchissantes qui servent de marqueurs et disposées stratégiquement sur les sujets.



Figure 2.10 – Une séance de capture de mouvements au LACM (<http://taiar.univ-reims.fr/>).

Cette méthode, désormais couramment utilisée dans l'industrie du cinéma ou des jeux vidéo, pose toutefois quelques problèmes dans le cadre de la simulation précise d'une articulation où le positionnement précis des marqueurs est requis. En effet, ces marqueurs sont généralement disposés à même la peau et sont donc sujets aux mouvements de la chair sur les os et les muscles, appelés glissements de peau. Ces mouvements sont difficilement évaluables et viennent perturber le calcul des positions des marqueurs. Bien qu'un soin tout particulier soit fourni lors du positionnement initial des marqueurs afin d'en minimiser l'amplitude, ces mouvements sont malgré tout inévitables.

Les informations physiologiques des éléments peuvent être obtenues à partir de la littérature et des connaissances d'experts. Ainsi, recoupées avec les informations géométriques nous pouvons disposer d'informations personnalisables.

Certains efforts intervenant sur le système au cours de l'exercice peuvent être mesurés

par des capteurs spécifiques, tel que des tapis mesurant la pression ou des systèmes à retour de force permettant de mesurer l'effort fourni [CWZ⁺02, KPJ02]. Cependant, la mesure des efforts pour la modélisation en dynamique inverse est une tâche ardue et loin d'être efficace puisqu'elle doit reposer sur une connaissance appropriée des interactions entre la zone de capture et la zone d'application de l'effort dans la simulation. C'est à dire, par exemple, les interactions entre la plante du pied pour un tapis mesurant la pression et le tibia pour une simulation du genou.

Il est à noter que pour la validation d'un modèle, dans le cadre d'une étude fine, il est nécessaire de disposer de l'ensemble des informations personnalisées pour chaque sujet d'étude afin de rester cohérent sur les informations que l'on compare et valide. Dans notre cas, à cause du retard pris par le projet SysReeduc nous ne disposons malheureusement d'aucune donnée expérimentale.

Cependant, à terme, nous envisageons de pouvoir effectuer des simulations sur des données réelles, extraites d'IRM ou de radios. Dans un premier temps, notre étude n'étant que cinématique, la modélisation géométrique s'intéresse essentiellement aux surfaces osseuses. Celles-ci pourront être obtenues à l'aide d'algorithmes d'extraction d'informations géométriques à partir de données volumiques et stockées sous forme de maillages triangulaires. Nous travaillons donc sur ce type de données pour la modélisation géométrique. Par la suite, afin de valider notre modèle, les courbes cinématiques obtenues par simulation pourront être comparées à celles obtenues par capture de mouvement.

Les informations physiologiques n'interviendront que dans une optique de simulation dynamique. Il sera alors nécessaire de les corrélérer aux données volumiques des IRM ou radios afin de parvenir à un modèle personnalisé. Mais il s'agit là d'un travail complexe, dépassant le cadre de cette thèse.

2.5 Travaux préliminaires

Ce travail de thèse prend la suite d'une étude réalisée dans le cadre du stage de DEA de Delphine Deliessche. Ce travail était un premier pas vers la réalisation d'un modèle biomécanique de l'articulation du genou.

Dans cette optique, après l'étude du fonctionnement de l'articulation et grâce à la participation d'un expert kinésithérapeute, un modèle simplifié du genou a été mis en place, permettant de simuler de manière efficace l'interaction entre le tibia et le fémur tout en prenant en compte les restrictions dues aux ligaments.

Ce modèle repose sur l'hypothèse que les deux mouvements interne/externe et adduction/abduction du genou sont inhérents à la structure de l'articulation, et qu'ils peuvent être exprimés en fonction du mouvement de flexion/extension.

En accord avec l'expert et par leur faible amplitude, les mouvements de tiroir sont négligés dans cette étude pour se focaliser uniquement sur les rotations. Le modèle articulaire recherché n'a donc qu'une coordonnée généralisée θ_0 , caractérisant l'angle de flexion/extension et deux angles secondaires $\theta_1 = f_1(\theta_0)$ et $\theta_2 = f_2(\theta_0)$ associés aux deux rotations corrélées à la flexion/extension.

Ce modèle a ensuite été introduit au sein de MADy, le moteur d'animation dynamique développé au sein du laboratoire, afin de le tester et d'en valider les résultats. Je propose d'expliquer le modèle mis en place, d'en étudier les résultats et d'en tirer les perspectives développées dans cette thèse.

2.5.1 Le modèle cinématique

Le modèle cinématique consiste à caractériser et à représenter les différentes structures anatomiques d'intérêt et à décrire leurs interactions et domaines d'évolution.

Le rôle principal de la rotule est de servir de poulie entre le ligament rotulien et les tendons du quadriceps. Cette étude préliminaire ne portant que sur la cinématique de l'articulation, celle-ci a été réduite à la seule interaction du tibia et du fémur. Dans le cadre de la simulation, une articulation se résume à un ensemble de liaisons mécaniques représentant les interactions entre les os.

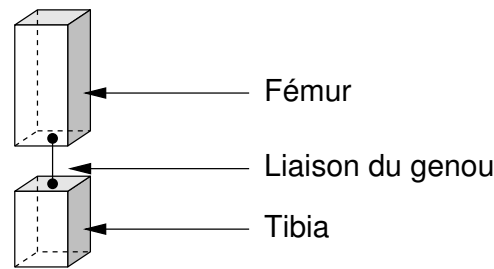


Figure 2.11 – Représentation simplifiée de l'articulation.

Ici, l'articulation du genou, et donc l'interaction entre le tibia et le fémur, est modélisée en se focalisant sur les contacts articulaires au nombre de deux : la partie interne (condyle fémoral interne - plateau tibial interne) et la partie externe (condyle fémoral externe - plateau tibial externe).

D'après la littérature et des connaissances d'experts, le rayon de courbure du condyle fémoral interne reste à peu près constant au cours de la marche, soit une amplitude de flexion d'à peu près 70° . Il vient donc parfaitement s'enchasser dans la forme concave du plateau tibial interne. L'action combinée de ces deux surfaces interdit donc les translations dans la partie interne. Elle peut être modélisée par une liaison mécanique de type rotule dont le centre de rotation est le centre de la sphère inscrite au condyle fémoral interne.

La partie externe se compose du plateau tibial externe et du condyle fémoral externe. Avec l'accord de l'expert, le plateau tibial a été modélisé sous la forme d'un plan tangent. L'approche employée pour représenter le condyle fémoral externe, a été de le modéliser sous la forme d'une courbe spline de Catmull-Rom représentant le chemin des points de contact entre le condyle fémoral externe et le plateau tibial externe sur la surface du condyle, comme illustré par la figure 2.12.

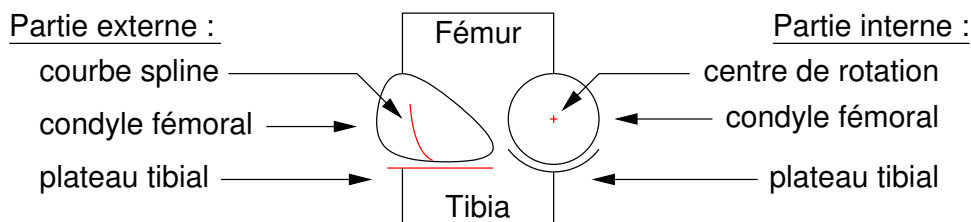


Figure 2.12 – Modélisation simplifiée des contacts osseux de l'articulation du genou.

Afin d'acquérir ces données, l'étude s'appuie sur des données réelles extraites de trois coupes d'images IRM T1 d'un genou gauche (coupe coronale, sagittale et axiale), extraites de l'atlas DAVID (Online Atlas of Human Anatomy for Clinical Imaging Diagnosis). Une application a été développée, illustrée par la figure 2.13, permettant la visualisation des trois plans de coupe, d'une part en projection parallèle et d'autre part en projection perspective, qui après recalage des trois plans, montre une approximation 3D de l'articulation.

En se basant sur ces trois projections, on peut identifier les différents éléments intervenant dans l'articulation. Le centre de rotation de la partie interne coïncide avec le centre de la sphère inscrite dans le condyle fémoral interne. Celle-ci étant obtenue à partir de trois points définis sur la surface du condyle fémoral interne dans les coupes axiale et coronale. La coupe coronale permet la localisation des plateaux tibiaux et on peut donc disposer le plan tangent représentant le plateau tibial externe de manière interactive.

Pour finir, la courbe spline définissant le chemin des points de contact sur la surface du condyle fémoral externe au cours de la flexion/extension est obtenue en plaçant les points de contrôle de manière interactive dans la coupe sagittale pour suivre au mieux la structure osseuse. Cette courbe spline est ensuite déformée de manière à tenir compte

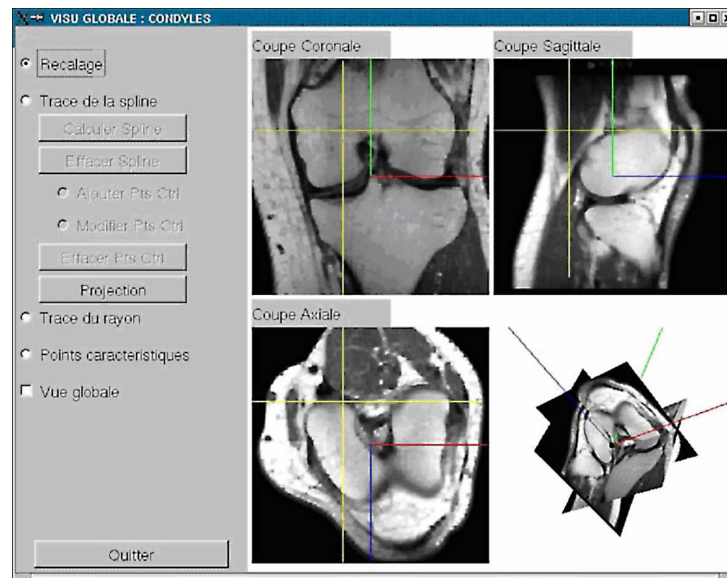


Figure 2.13 – L’interface de l’application de modélisation du genou.

de la forme arrondie du condyle fémoral externe dans le plan transversal en projetant ses points de contrôle sur une sphère définie par son centre et son rayon obtenus interactivement dans les coupes axiale et coronale.

On souhaitait obtenir au final un degré de liberté complexe représentant le mouvement de flexion/extension. La partie interne modélisée sous la forme d’une liaison mécanique de type rotule interdit les translations et n’autorise que trois rotations, soit trois degrés de liberté. L’ajout de contraintes va réduire la dimension de l’espace d’évolution des degrés de liberté. Deux contraintes doivent donc être introduites pour obtenir le mouvement principal. Au final, on obtient un degré de liberté unique en observant l’évolution des angles secondaires par rapport à l’angle principal (flexion/extension) au cours des simulations. Il est possible ensuite, après interpolation d’obtenir les rotations secondaires à partir de la flexion/extension, soit un seul degré de liberté.

La première contrainte consiste à imposer un contact permanent entre le plateau tibial et le condyle fémoral afin de simuler l’action des ligaments. Cette contrainte consiste à imposer qu’un point de la courbe de Catmull-Rom, représentant le chemin des points de contact entre le condyle fémoral externe et le plateau tibial externe au cours de la flexion sur le condyle, soit en contact avec le plateau tibial, comme illustré par la figure 2.14.

Le plateau tibial est représenté par un plan $P(\mathbf{o}, \mathbf{n})$ défini à partir d’un point $\mathbf{o}$ et d’une normale $\mathbf{n}$. On cherche à identifier le point $\mathbf{p}(u)$ de la courbe spline, qui devrait être en contact tangent avec le plan du plateau tibial. Le paramètre u recherché doit assurer

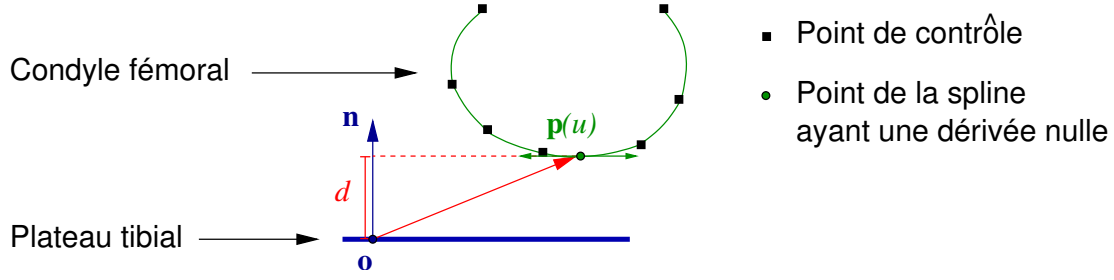


Figure 2.14 – Le point de contact entre le condyle fémoral externe et le plateau tibial externe.

le contact comme suit :

$$\mathbf{n} \cdot (R^{-1}(\theta_0, \theta_1, \theta_2)\mathbf{p}(u) - \mathbf{o}) = 0 \quad (2.1)$$

Avec $R(\theta_0, \theta_1, \theta_2)$ la matrice de transformation du repère du fémur au tibia.

Par ailleurs, la courbe de Catmull-Rom étant supposée continûment dérivable, la contrainte de contact tangent peut être formulée par une relation d'orthogonalité entre la tangente $\mathbf{p}'(u)$ à la courbe et la normale au plan.

$$\mathbf{n} \cdot R^{-1}(\theta_0, \theta_1, \theta_2)\mathbf{p}'(u) = 0 \quad (2.2)$$

On peut constater qu'une inconnue supplémentaire a été introduite pour caractériser le point de contact.

La seconde contrainte repose sur une simplification anatomique, en considérant que le ligament *croisé antéro-externe* reste tendu au cours de la marche et que la rotule vient maintenir le fémur en contact avec le tibia, la longueur du ligament *croisé antéro-externe* peut être considérée comme constante durant cet exercice. On peut alors définir une contrainte de distance constante entre un point $\mathbf{f}$ lié à la face interne du condyle externe du fémur et un point $\mathbf{t}$ lié à la surface *pré-spinale* du plateau tibial. Ces deux points sont placés interactivement dans l'application de modélisation. Cette contrainte s'exprime donc par l'équation 2.3.

$$\|R^{-1}(\theta_0, \theta_1, \theta_2)\mathbf{f} - \mathbf{t}\|^2 = c^2 \quad (2.3)$$

Nous disposons donc au final de trois degrés de liberté et d'une inconnue supplémentaire permettant d'identifier le point de contact. Dans le même temps, nous avons défini trois contraintes (équations 2.1, 2.2 et 2.3). Nous obtenons, finalement, l'expression d'un degré de liberté complexe, exprimant le mouvement de flexion/extension du genou.

Une fois la description cinématique de l'articulation terminée, nous pouvons expéri-

menter les mouvements obtenus par la mise en application du modèle.

2.5.2 Résultats

Ce modèle permet d'obtenir pour un angle θ_0 de flexion/extension donné les angles des rotations interne/externe θ_1 et abduction/adduction θ_2 correspondants. En faisant varier l'angle de flexion de 0° à 80° , on peut obtenir des courbes simulées définies sur le même intervalle que les courbes de référence fournies par l'expert. Nous pouvons donc comparer ces deux jeux de courbes (figure 2.15) afin de juger de la pertinence du système. On peut constater que la méthodologie semble correcte puisque les allures des courbes simulées et de référence sont similaires.

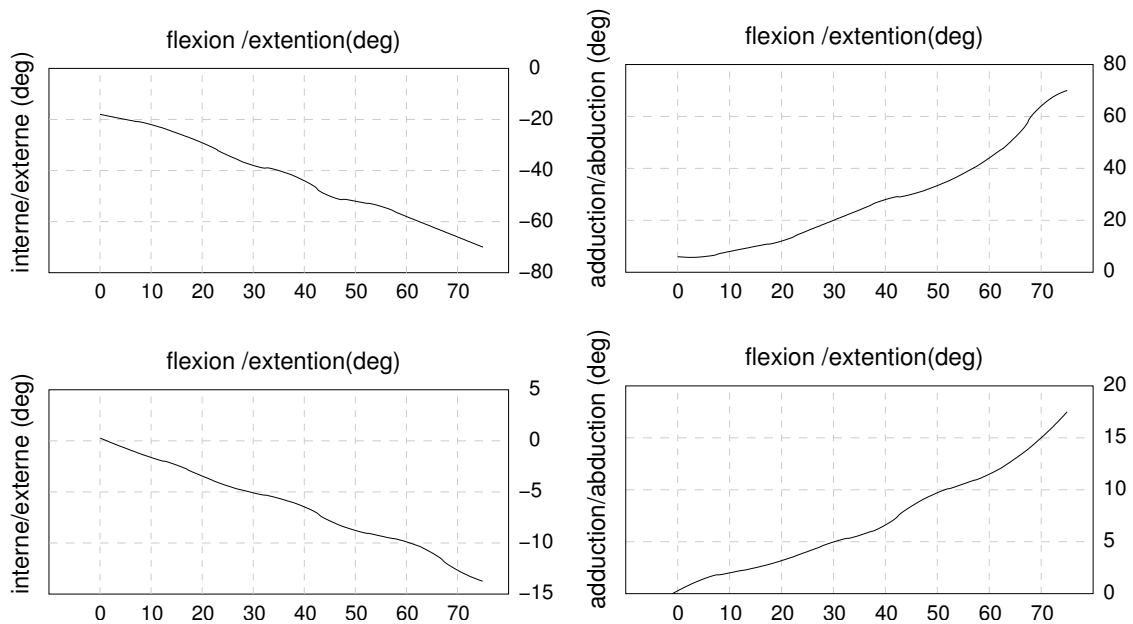


Figure 2.15 – Variation des angles de rotations secondaires en fonction de l'angle de rotation de la flexion/extension. La rotation interne/externe à gauche et adduction/abduction à droite. Les résultats obtenus par le modèle sur la ligne du haut et les valeurs de référence sur la ligne du bas.

Cependant, si on observe les échelles d'angle des rotations internes, on constate, que par rapport aux valeurs de référence, les valeurs simulées sont amplifiées. Cette amplification des courbes de rotation est probablement due à une approximation du système lors de la modélisation. Celle-ci peut résulter d'une erreur lors du positionnement manuel des éléments du modèle ou d'une simplification méthodologique arbitraire.

Nous avons identifié quelques sources probables d'erreurs. Pour commencer, la modélisation du contact de la partie externe nous semble assez peu fiable, car elle repose sur

le profil 2D du condyle fémoral déformé par projection sur une sphère afin d'obtenir une estimation 3D du chemin des points de contact sur le condyle. Par ailleurs, la simplification de la partie interne du genou sous la forme d'une liaison rotule, interdisant ainsi les mouvements de tiroir dans cette partie de l'articulation, pourrait imposer des contraintes arbitraires au système.

De manière générale, il nous semble intéressant de tenter de limiter l'intervention humaine au cours du processus de modélisation géométrique en automatisant la modélisation et le positionnement des éléments de l'articulation. C'est pourquoi j'ai proposé d'utiliser une représentation 3D des surfaces osseuses.

2.6 Conclusion

Nous avons vu dans ce chapitre que l'articulation du genou est une articulation complexe faisant intervenir de nombreux éléments dans son action et que de cette complexité structurelle résulte une complexité comportementale se traduisant par l'apparition de mouvements secondaires en marge de la flexion/extention : les rotations interne/externe et adduction/abduction ainsi que les mouvements de tiroir.

Nous avons vu qu'il n'existe pas de modèle cinématique de l'articulation du genou qui soit pertinent. Or puisque les modèles statiques et dynamiques reposent sur celui-ci, nous ne pouvons pas encore construire de modèle statique ou dynamique pertinent du genou. Les travaux antérieurs sur la simulation réaliste de l'articulation du genou reposent généralement sur une approximation géométrique des structures de l'articulation. Or, afin d'obtenir un modèle de genou personnalisé et efficace, il est nécessaire de s'appuyer sur les données (géométriques dans un premier temps) propres à chaque individu à l'aide de techniques telles que l'IRM. Par ailleurs, en accord avec les conclusions du tour d'horizon et afin de ne pas introduire d'efforts non-naturels dans le modèle, nous ne devons pas nous contenter de liaisons mécaniques classiques.

L'articulation du genou ne peut donc pas être réduite à une simple liaison pivot et les premiers travaux du laboratoire ont donc consisté à définir une liaison à un degré de liberté complexe, s'exprimant comme la combinaison de trois rotations. Cette approche, reposant sur l'exploitation des degrés de liberté réels, ainsi que l'intégration naturelle des contraintes nous a encouragés à employer le formalisme Lagrangien.

Une autre approche, largement employée pour la simulation, aurait pû être la méthode des éléments finis. Celle-ci pouvant reposer sur l'exploitation directe des données brutes. Cependant, comme nous le verrons plus en détail dans le chapitre 4, le formalisme

Lagrangien est plus approprié. En effet, il permet l'intégration naturelle de contraintes complexes au modèle, permettant ainsi de contraindre indirectement le positionnement et l'orientation des structures de l'articulation. De plus, la caractérisation éventuelle du mouvement global simulé est immédiate, contrairement à la méthode des éléments finis.

Les résultats obtenus sont engageants car les courbes obtenues par simulation ont la même allure que les courbes de référence. Cependant, il existe un facteur d'échelle entre ces courbes et celles produites par la simulation qui peut provenir de plusieurs approximations telles que l'utilisation de données 2D pour la création d'un modèle 3D ou la modélisation et le positionnement manuel des différents éléments de l'articulation. De plus, comme nous l'avons vu, si nous ne souhaitons pas introduire d'efforts arbitraires au sein du système, nous devons introduire un pivot mobile dans notre modèle d'articulation.

Afin de pallier à ces différents problèmes, nous avons envisagé de passer par une modélisation 3D du système en nous basant sur un maillage 3D des surfaces osseuses. Cette approche devrait nous permettre d'automatiser une partie du processus de modélisation et ainsi de limiter les risques d'approximation lors de la modélisation ou le positionnement manuel des éléments de l'articulation. Mais surtout, le modèle reposera sur de réelles données 3D, et évitera ainsi les approximations structurelles.

Cependant, l'introduction d'une modélisation 3D des surfaces osseuses au sein de notre moteur d'animation dynamique nécessite que nous adoptions une représentation surfacique continue des données 3D, il s'agit du problème que nous abordons dans le chapitre 3.

De plus, nous avons souhaité profiter de l'opportunité de passer à une représentation 3D pour complexifier la structure de notre modèle en supprimant la liaison mécanique de type rotule pour la remplacer par des structures surfaciques liées par des contraintes. La disparition de cette liaison interne rend nécessaire l'intégration de contraintes pouvant gérer les contacts glissants entre les surfaces afin de simuler les mouvements de roulement/glisement au sein de l'articulation. Pour traduire ces contacts dans le moteur d'animation, nous utiliserons le formalisme des *contraintes à variables libres* initié par des échanges avec le LIFL qui ont finalement abouti dans [LGM⁺04] et développé parallèlement dans un rapport technique interne par Remion [Rem03].

Dans un premier temps, la liaison entre le tibia et le fémur peut être modélisée à l'aide de contraintes de contact glissant permanent. Mais, dans le souci de proposer un modèle articulaire suffisamment générique pour intégrer des pathologies liées au vieillissement ou aux blessures, il est important de considérer le décrochage éventuel du contact osseux dû au jeu articulaire. Cette prise en compte se traduit par la mise en place d'une gestion rigoureuse de l'activation et la désactivation de contraintes unilatérales multiples. Le cha-

pitre 4 traite de manière générale de la gestion mécanique du modèle et de l'intégration de ces outils en particulier.

Chapitre 3

Modélisation géométrique

« “WHAT ’s the good of Mercator’s North Poles and Equators,
Tropics, Zones, and Meridian Line?”
So the Bellman would cry : and the crew would reply
“They are merely conventional signs!”

“Other maps are such shapes, with their islands and capes!
But we’ve our brave Captain to thank :
(So the crew would protest) that he’s bought us the best-
A perfect and asolute blank!” »

Lewis CARROLL, *The Hunting of the Snark*. (1876).

Sommaire

3.1	Introduction	35
3.2	Paramétrisation	37
3.2.1	Paramétrisation de surfaces ouvertes	37
3.2.2	Paramétrisation de surfaces fermées	46
3.3	Interpolation	49
3.3.1	Surface paramétrique à base radiale	50
3.3.2	Partition de l’unité	55
3.3.3	Interpolation sphérique	61
3.4	Conclusion	63

3.1 Introduction

Je tiens à rappeler que, dans le cadre de ce travail de recherche, notre objectif est de parvenir à effectuer le suivi continu des points de contact sur les surfaces cartilagineuses

du fémur et du tibia lors du mouvement de flexion extension. A ce titre, les surfaces que nous sommes amenés à utiliser doivent satisfaire un certain nombre de critères de nature géométrique tels que :

1. une description explicite des surfaces s'appuyant sur un paramétrage global afin de garantir l'identification unique du point contact.
2. une continuité de type C^2 afin de pouvoir effectuer le suivi continu du point de contact en position, vitesse et accélération.

Par ailleurs, nous avons vu dans le chapitre précédent que les surfaces osseuses manipulées au cours de la simulation doivent être initialisées à partir de données géométriques réelles pour garantir une simulation précise des interactions entre les éléments de l'articulation du genou. Ces informations, qui sont issues la plupart du temps de périphériques d'acquisition tels que des scanners 3D ou IRM, sont représentées en général sous forme d'un maillage triangulaire. Dans ce travail, je propose une méthode permettant de générer une surface paramétrique C^2 interpolante à paramétrage global à partir d'un maillage triangulaire quelconque. Cette méthode se décompose en deux étapes illustrées par la figure 3.1.

La première étape consiste à construire une paramétrisation globale du maillage triangulaire. Je rappelle qu'il est important que la paramétrisation soit globale afin de permettre un suivi continu du contact osseux durant le mouvement de flexion extension de l'articulation du genou. La deuxième étape génère une surface paramétrique continûment différentiable interpolant le maillage triangulaire au moyen de surfaces paramétriques à base radiale. Dans la mesure où la surface est décrite explicitement, le point de contact

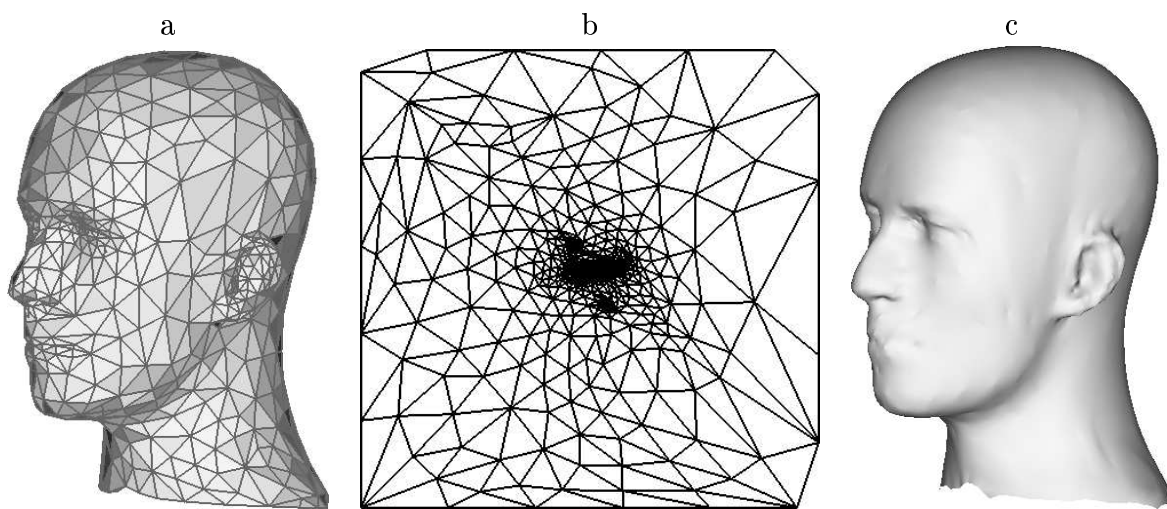


Figure 3.1 – Construction de surface interpolante à paramétrage global - a) le maillage initial, b) le maillage initial plaqué sur l'espace paramétrique, c) la surface interpolante.

est naturellement identifié par un paramètre 2D.

De manière similaire, j'ai organisé cette section en deux parties, la première traite de la paramétrisation de surfaces triangulaires avec une discussion sur les méthodes existantes et les décisions que j'ai prises. La seconde partie présente la construction d'une surface paramétrique interpolant le maillage triangulaire au moyen de surfaces à base radiale. Rappelons que ces résultats ont fait l'objet de deux communications [JNR05, JNR06].

3.2 Paramétrisation

3.2.1 Paramétrisation de surfaces ouvertes

La paramétrisation d'une surface définie à partir d'un maillage triangulaire consiste à associer chaque sommet de la surface 3D à un couple de coordonnées désignant un point de l'espace paramétrique afin d'obtenir une relation bijective entre l'espace paramétrique et la surface 3D. La paramétrisation des points intérieurs d'un triangle est obtenue par interpolation linéaire des paramètres aux sommets. Bien que certaines surfaces puissent facilement être paramétrées par projection orthogonale sur un plan, la plupart des surfaces peuvent faire apparaître des chevauchements une fois projetées de la sorte dans l'espace paramétrique. Il n'est donc pas trivial de définir automatiquement une paramétrisation pour une surface quelconque.

Les premiers travaux de paramétrisation sont attribués à l'astronome grec Claudius Ptolemaeus (90-168), qui imagine dans ces travaux « Géographie » les premières projections coniques pour passer de la sphère au plan. Cependant cette projection induit des déformations d'angles et d'aires. Plusieurs solutions ont été apportées pour pallier à ces déformations mais il a fallu attendre 1569 pour que Gerardus Mercator invente une projection où les trajets à cap constant sont représentés par des droites. Cette représentation est encore utilisée bien qu'elle induise des déformations grandissantes à mesure que l'on s'éloigne de l'équateur.

En informatique graphique, cette technique a été introduite pour la première fois par Catmull [Cat74]. Depuis, différentes approches ont été décrites afin d'améliorer ces résultats comme l'utilisation de transformations géométriques simples [Pea85], de méthodes de dépliage [BVIG91] ou encore d'optimisation globale [KL96]. Il s'agit en effet d'un domaine de recherche très actif notamment en raison de l'usage croissant de maillages triangulaires issus des scanners 3D et à la complexification des modèles polygonaux manipulés en simulation et en visualisation. Au delà du plaquage de texture, de nombreuses applications de la paramétrisation ont vu le jour telles que l'approximation de surfaces et le remaillage

de modèles polygonaux par exemple.

La paramétrisation d'une surface introduit généralement des distorsions à la fois au niveau des angles et des aires entre la surface 3D d'origine et sa version paramétrique. Une paramétrisation efficace tend à minimiser ces distorsions. La minimisation de ces deux distorsions peut être contradictoire et on est amené en général à faire un choix entre les deux ou se contenter d'un compromis. De nombreuses solutions existent dans la littérature, dont [FH05, HLS07] proposent un très bon tour d'horizon. Il est à noter qu'il existe des méthodes de paramétrisation à base de dépliage ou de découpage que nous n'aborderons pas ici car elles ne permettent pas d'obtenir un paramétrage global de la surface sans discontinuité sur le domaine paramétrique.

Dans ce travail, les surfaces d'intérêt se composent des têtes des os, je me suis donc restreint, dans un premier temps, à la paramétrisation de surfaces semblables (au niveau topologique) à un disque.

Toutes les méthodes de paramétrisation de maillages ouverts présentées par la suite se conforment au même schéma : on commence par générer une paramétrisation des sommets qui composent la frontière du maillage triangulaire pour ensuite calculer une paramétrisation des sommets intérieurs.

3.2.1.1 Paramétrisation de la frontière

La paramétrisation de la frontière repose sur deux choix : la forme de la frontière dans l'espace paramétrique et la répartition des points sur cette frontière. Généralement, la contrainte principale pour la forme de la frontière de l'espace paramétrique est la convexité. C'est pourquoi on trouve régulièrement des formes au paramétrage simple : rectangle, cercle ou triangle. Par ailleurs, la paramétrisation rectangulaire est idéale pour le plaquage de texture.

Ces solutions peuvent cependant générer d'importantes distorsions aux abords de la frontière de l'espace paramétrique. Une solution intéressante [KV03, LKL02] permettant de pallier ce défaut consiste à compléter la frontière du maillage triangulaire à l'aide de sommets "virtuels" intermédiaires au niveau des zones de distorsion afin de lisser ces déformations.

Une fois la forme du domaine paramétrique choisie, il reste à définir la fonction de projection de la frontière du maillage sur la frontière du domaine paramétrique. La plupart des articles ne considérant que des surfaces représentées par des maillages triangulaires, les sommets appartenant à la frontière du maillage sont positionnés sur la frontière du

domaine paramétrique. La répartition de ces sommets se fait généralement de façon uniforme ou conformément à l'abscisse curviligne afin de tenir compte de la longueur de arêtes du maillage 3D (figure 3.2).

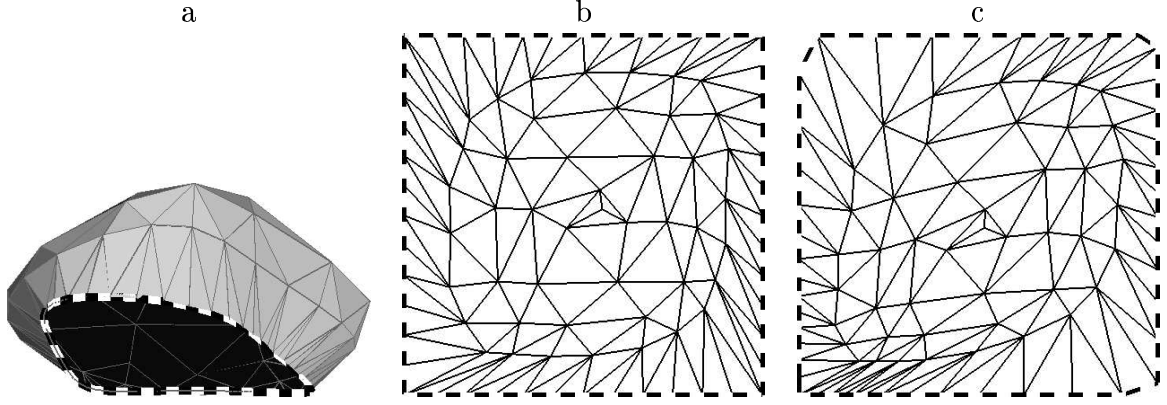


Figure 3.2 – Paramétrisation de la frontière - a) maillage triangulaire, b) paramétrisation uniforme, c) paramétrisation en abscisse curviligne.

3.2.1.2 Paramétrisation de l'intérieur

Dans l'optique de définir une paramétrisation d'une surface triangulaire, la théorie des plongements de graphes planaires fournit un formalisme approprié à l'étude de ce type de problèmes, et la notion d'application barycentrique décrite par Tutte dans [Tut63] se prête particulièrement bien à cette approche.

En effet, il est possible, sous certaines conditions, de définir un graphe planaire à partir d'un maillage 3D. Il reste ensuite à organiser ce graphe afin qu'il s'adapte à l'espace paramétrique souhaité. Tutte [Tut63] introduit la notion d'application barycentrique pour le plaquage uniforme d'un graphe planaire sur un espace 2D.

Floater [Flo97, Flo03b] reprend ces concepts en étendant la notion d'isobarycentre à celle de combinaison convexe, dont les coefficients permettent de minimiser les déformations. Afin de détailler cette dernière méthode, nous rappelons ci-dessous quelques éléments de la théorie des graphes :

Un graphe $G = G(S, A)$ est constitué d'un ensemble fini de sommets $S = \{s_1, \dots, s_n\}$ et d'un ensemble d'arêtes $A = \{\{i, j\} : 1 \leq i, j \leq n, i \neq j\}$. Le sommet s_i est voisin du sommet s_j si $\{i, j\} \in A$. Par convention, les voisins de s_i sont notés s_{ij} pour $j = 1, \dots, d_i$. Le degré d_i (ou valence) du sommet s_i correspond au nombre total de ses voisins. Un graphe $G = G(S, A)$ est dit planaire si ses sommets se projettent en des points distincts

du plan de sorte que ses arêtes ne soient pas sécantes.

Un graphe planaire partitionne le plan en régions connexes appelées faces. Dans le cadre de surfaces triangulaires, il s'agit d'un ensemble de triangles $T = \{(i, j, k) : 1 \leq i, j, k \leq n, i \neq j \neq k\}$. La paramétrisation d'un maillage consiste donc à définir un graphe à partir de ce maillage puis à le transformer en graphe planaire dont la frontière épouserait la frontière du domaine paramétrique. La paramétrisation étant bijective, la transformation en graphe planaire doit être inversible. Le graphe planaire ne doit donc pas admettre de chevauchements.

Pour ce faire, Tutte exprime les coordonnées paramétriques $\mathbf{w}_i = (u_i, v_i)$ d'un sommet interne $\mathbf{s}_i$ du graphe comme le barycentre des coordonnées paramétriques de ses voisins $\mathbf{w}_{ij} = (u_{ij}, v_{ij})$ (équation 3.1). De cette manière, il en déduit un système linéaire dont la résolution assure le non recouvrement des faces du graphe.

$$u_i = \sum_{j=1}^{d_i} \alpha_{ij} u_{ij} \quad , \quad v_i = \sum_{j=1}^{d_i} \alpha_{ij} v_{ij} \quad avec \quad \sum_{j=1}^{d_i} \alpha_{ij} = 1 \quad (3.1)$$

Le choix des coefficients α_{ij} de la combinaison convexe agit sur la déformation appliquée au maillage lors de sa transformation en graphe planaire. Il est possible d'attribuer une importance identique à chacun des voisins, il s'agit de la paramétrisation barycentrique illustrée par l'équation 3.2. On peut aussi choisir des pondérations qui vont permettre de préserver la forme du maillage ou la proportion des aires de chaque face.

$$\alpha_{ij} = \frac{1}{d_i} \quad \forall j = 1, \dots, d_i \quad (3.2)$$

Floater [Flo97] propose une méthode permettant de préserver la forme des faces du maillage 3D dans le graphe planaire, qu'il appelle *shape-preserving*. Cette méthode se décompose en deux étapes : premièrement, elle consiste à projeter le sommet d'intérêt et ses voisins dans un plan, puis à déterminer la pondération adaptée pour chaque voisin.

On commence donc par déterminer un paramétrage local en exprimant les sommets voisins à l'aide des coordonnées polaires (figure 3.3).

Le sommet s_i se projette en un point $\mathbf{p}_i = (0, 0)$. Soit $\langle \mathbf{s}_j, \mathbf{s}_i, \mathbf{s}_{j+1} \rangle$ l'angle entre les vecteurs $\overrightarrow{\mathbf{s}_i \mathbf{s}_j}$ et $\overrightarrow{\mathbf{s}_i \mathbf{s}_{j+1}}$ et $\theta_i = \sum_{j=1}^{d_i} \langle \mathbf{s}_j, \mathbf{s}_i, \mathbf{s}_{j+1} \rangle$ (le parcours des voisins de $\mathbf{s}_i$ étant cyclique, l'indice suivant d_i est égal à 1). Les voisins $\mathbf{s}_{ij}$ du sommet $\mathbf{s}_i$ se projettent en des points

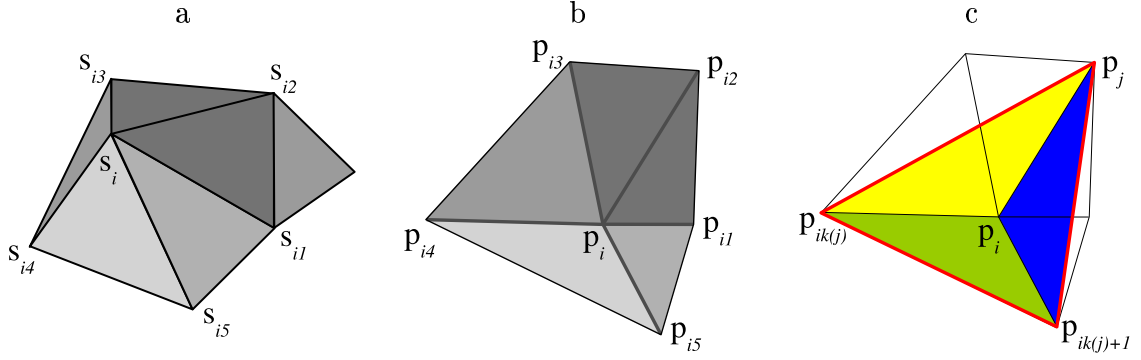


Figure 3.3 – Illustration du calcul des coefficients pour la méthode de préservation de forme.

$\mathbf{p}_{ij}$ définis par l'équation 3.3.

$$\mathbf{p}_{ij} = \left(\|\overrightarrow{s_i s_{j+1}}\|, \frac{1}{\theta_i} \sum_{k=1}^j \langle \mathbf{s}_k, \mathbf{s}_i, \mathbf{s}_{k+1} \rangle \right) \quad (3.3)$$

Dans cet espace paramétrique, on définit $p_{ik(j)}$ et $p_{ik(j)+1}$ les deux points à l'extrémité de l'arête concourante avec la droite définie par $\mathbf{p}_i = (0,0)$ et $\mathbf{p}_{ij}$. On obtient donc un triangle $(\mathbf{p}_{ij}, \mathbf{p}_{ik(j)}, \mathbf{p}_{ik(j)+1})$ contenant le point $\mathbf{p}_i$. On peut donc définir pour chaque triangle ainsi obtenu une quantification normalisée de l'importance de son aire (équation 3.4) :

$$\begin{aligned} \lambda_{ij} &= \frac{\text{aire}(\mathbf{p}_i, \mathbf{p}_{ik(j)}, \mathbf{p}_{ik(j)+1})}{\text{aire}(\mathbf{p}_{ij}, \mathbf{p}_{ik(j)}, \mathbf{p}_{ik(j)+1})}, \\ \lambda_{ik(j)} &= \frac{\text{aire}(\mathbf{p}_j, \mathbf{p}_i, \mathbf{p}_{ik(j)+1})}{\text{aire}(\mathbf{p}_{ij}, \mathbf{p}_{ik(j)}, \mathbf{p}_{ik(j)+1})}, \\ \lambda_{ik(j)+1} &= \frac{\text{aire}(\mathbf{p}_j, \mathbf{p}_{ik(j)+1}, \mathbf{p}_i)}{\text{aire}(\mathbf{p}_{ij}, \mathbf{p}_{ik(j)}, \mathbf{p}_{ik(j)+1})} \end{aligned} \quad (3.4)$$

En posant $\chi_{ij}^1(l) = \lambda_{ij}$ si $j = l$, $\chi_{ij}^2(l) = \lambda_{ik(j)}$ si $k(j) = l$ et $\chi_{ij}^3(l) = \lambda_{ik(j)+1}$ si $k(j) + 1 = l$, on obtient finalement pour chaque voisin du sommet $\mathbf{s}_i$:

$$\alpha_{ij} = \sum_{\substack{k \in \{1, \dots, d_i\} \\ k \neq j}} \frac{\chi_{ij}^1(k) + \chi_{ij}^2(k) + \chi_{ij}^3(k)}{d_i} \quad (3.5)$$

D'après les équations 3.4 et 3.5, on a $\sum_{j=1}^{d_i} \alpha_{ij} = 1$. On peut donc exprimer les coordonnées paramétriques de $\mathbf{s}_i$ comme une combinaison convexe des coordonnées paramétriques de ces voisins $\mathbf{s}_{ij}$.

Enfin, Floater propose une autre pondération [Flo03a] reposant sur la propriété de la

valeur moyenne pour les fonctions harmoniques. Les coefficients de pondération sont alors définis par l'équation 3.6 :

$$\alpha_{ij} = \frac{\tan(\langle \mathbf{s}_{ij-1}, \mathbf{s}_i, \mathbf{s}_{ij} \rangle / 2) + \tan(\langle \mathbf{s}_{ij}, \mathbf{s}_i, \mathbf{s}_{ij+1} \rangle / 2)}{\|\mathbf{p}_j - \mathbf{p}_i\|} \quad (3.6)$$

avec les angles $\langle \mathbf{s}_{ij-1}, \mathbf{s}_i, \mathbf{s}_{ij} \rangle$ et $\langle \mathbf{s}_{ij}, \mathbf{s}_i, \mathbf{s}_{ij+1} \rangle$ comme décrit dans la figure 3.4.

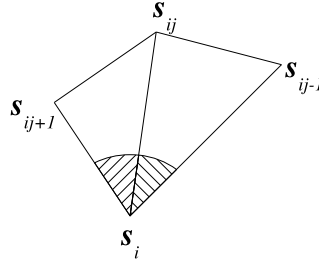


Figure 3.4 – Configuration des angles autour de l'arête $\{\mathbf{s}_i, \mathbf{s}_{ij}\}$.

Ces méthodes fonctionnent bien en règle générale tant que le maillage triangulaire est bien proportionné (les angles de l'ensemble des triangles sont inférieurs à $\pi/2$) même si des contre-exemples peuvent être trouvés [DCDS97, Flo98]. En définitive, cette méthode permet de calculer un couple de coordonnées (u_i, v_i) distinctes pour chaque sommet $\mathbf{s}_i$ du maillage 3D puisqu'elle interdit tout recouvrement en limitant le domaine de définition à un espace paramétrique défini par la frontière du graphe planaire. De par le choix des pondérations α_{ij} , cette technique offre un certain contrôle sur la manière dont le maillage 3D est plaqué sur l'espace paramétrique (figure 3.5).

Sur une idée similaire à celle de Levy [LM00], Desbrun et al. [DMA02, MLBD02] proposent de complexifier le système des combinaisons convexes 3.1 en composant les coefficients de pondération à partir de plusieurs pondérations réduisant les déformations sur les angles et les distances. Ils réexpriment donc le système 3.1 par le système matriciel suivant 3.7.

$$\begin{aligned} \mathbf{M}\mathbf{U} &= \mathbf{C} \\ \begin{bmatrix} \mu\mathbf{M}^{(1)} & + & (1-\mu)\mathbf{M}^{(2)} \\ \mathbf{0} & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{w}^{interne} \\ \mathbf{w}^{frontière} \end{bmatrix} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{c}^{frontière} \end{bmatrix} \end{aligned} \quad (3.7)$$

Les vecteurs $\mathbf{w}^{interne}$ et $\mathbf{w}^{frontière}$ correspondant respectivement aux coordonnées paramétriques des sommets internes et des sommets de la frontière du maillage. Le vecteur $\mathbf{c}^{frontière}$ contient les coordonnées paramétriques des sommets de la frontière du maillage avec lesquelles on initialise le système. Et $\mathbf{M}^{(1)}$ et $\mathbf{M}^{(2)}$ sont les matrices des coefficients de pondération définies par les équations 3.8 et 3.9.

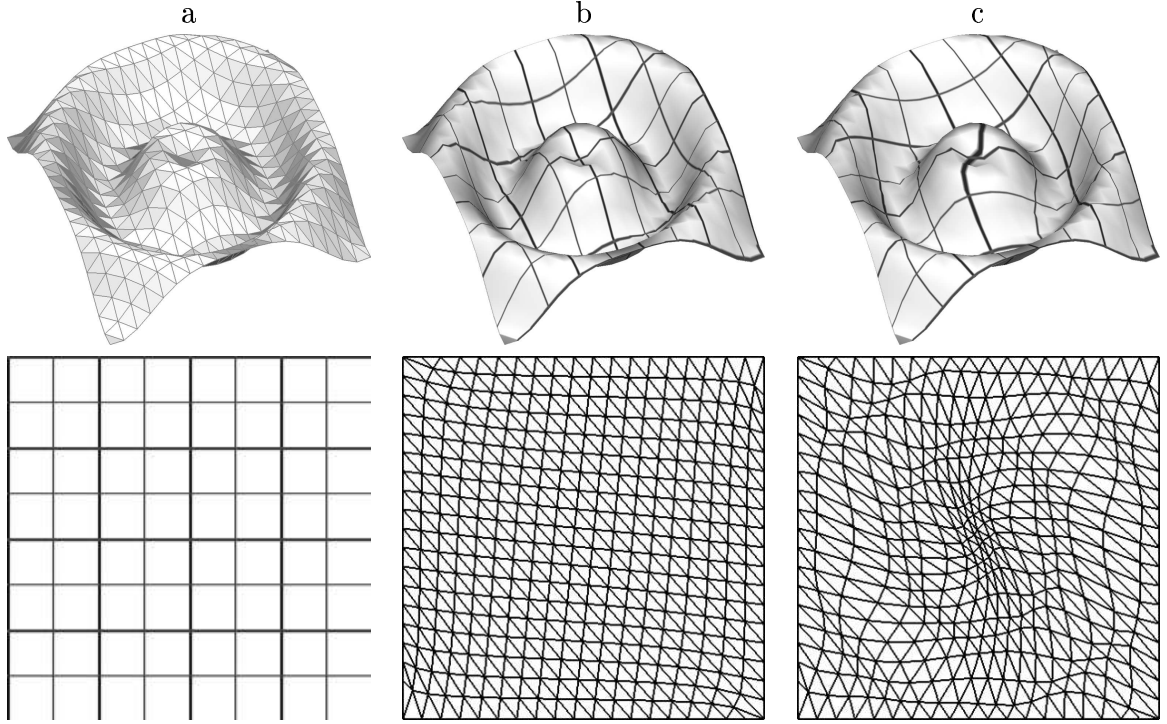


Figure 3.5 – Influence de la pondération sur la paramétrisation. (a) maillage initial et quadrillage régulier de l'espace paramétrique (b) paramétrisation barycentrique (c) paramétrisation avec préservation de forme.

$$M_{ij}^{(1)} = \begin{cases} \cot(\langle \mathbf{s}_i, \mathbf{s}_{ij-1}, \mathbf{s}_{ij} \rangle) + \cot(\langle \mathbf{s}_i, \mathbf{s}_{ij+1}, \mathbf{s}_{ij} \rangle) & \text{si } j \in \{1, \dots, d_i\} \\ \sum_{k=1}^{d_i} M_{ik}^{(1)} & \text{si } j = i \\ 0 & \text{sinon} \end{cases} \quad (3.8)$$

$$M_{ij}^{(2)} = \begin{cases} \frac{\cot(\langle \mathbf{s}_{ij-1}, \mathbf{s}_i, \mathbf{s}_{ij} \rangle) + \cot(\langle \mathbf{s}_{ij}, \mathbf{s}_i, \mathbf{s}_{ij+1} \rangle)}{\|\overline{\mathbf{s}_i \mathbf{s}_j}\|^2} & \text{si } j \in \{1, \dots, d_i\} \\ \sum_{k=1}^{d_i} M_{ik}^{(2)} & \text{si } j = i \\ 0 & \text{sinon} \end{cases} \quad (3.9)$$

Ils disposent ainsi d'un critère de quantification de déformation en angle et en distance pour chaque arête. Ils peuvent ainsi mélanger ces deux solutions optimales à l'aide d'un coefficient réel μ d'une manière similaire à Levy [LM00].

De plus, Desbrun et al. [DMA02] proposent une gestion élégante de contraintes de positionnement de certains points du maillage en introduisant des multiplicateurs de La-

grange dans le système 3.7. Ils obtiennent donc le système suivant :

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}^T \\ \mathbf{J} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{\Lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{D} \end{bmatrix} \quad (3.10)$$

Avec $\mathbf{J}_{ij} = 1$ si la $j^{\text{ème}}$ contrainte concerne le $i^{\text{ème}}$ sommet du graphe et 0 sinon. $\mathbf{\Lambda}$ correspond au vecteur des multiplicateurs de Lagrange et $\mathbf{D}$ aux valeurs souhaitées pour les sommets contraints.

Une dernière remarque sur la méthode des combinaisons convexes repose sur la résolution des systèmes exprimés par les équations 3.1 ou 3.10. Celles-ci reviennent à inverser une matrice carrée de dimension égale au nombre de sommets du maillage, ce qui peut rapidement devenir prohibitif en termes de calcul et de stockage. Cependant, puisque la matrice à inverser est remplie avec les coefficients de pondération, et que ceux-ci ne sont exprimés que pour les arêtes du maillage, on peut remarquer que la matrice est globalement creuse. Dans le but d'optimiser l'utilisation de la mémoire et les ressources de calcul, il est conseillé d'utiliser une bibliothèque profitant des avantages de ce type de matrice. J'ai utilisé, pour ces travaux, la bibliothèque **SparseLib++** [DLN⁺94] qui permet de stocker et d'inverser des matrices creuses de manière optimisée.

Les méthodes reposant sur les combinaisons convexes fonctionnent bien, mais il est intéressant de remarquer que la plupart des articles de paramétrisation travaillent sur l'expression d'une énergie à minimiser. En fait, la résolution de systèmes linéaires (combinaison convexe) peut être perçue comme la forme forte d'un problème de minimisation. La forme forte consiste à dériver la quantité à minimiser pour résoudre une équation où cette dérivée est identiquement nulle. L'approche basée sur la minimisation offre une plus grande richesse d'expression en termes de contraintes à imposer à la paramétrisation. Elle est aussi plus naturelle car elle peut être comparée à une approche physique d'un problème d'élasticité. C'est ce que démontre par exemple Guskov [Gus04] pour les coefficients du *shape-preserving* tandis que Greiner et al. [GH97] démontrent que tout choix d'un coefficient de pondération symétrique tel que $\alpha_{ij} = \alpha_{ji}$ dans le système 3.1 revient à minimiser l'énergie d'un ressort.

Levy [LM00] propose une méthode permettant de contrôler la nature des distorsions que l'on cherche à réduire : préservation des distances ou des angles. Dans ce but, il intègre à une méthode purement topologique, comme la paramétrisation barycentrique de Tutte, la prise en compte de deux termes quantifiant les déformations sur les angles ν (équation 3.12) et les distances μ (équation 3.13).

Si T désigne l'ensemble de tous les triangles du maillage, Levy définit pour tout triangle $t = (\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k) \in T$ un repère local $R_t = \{\mathbf{s}_i, \mathbf{a}, \mathbf{b}, \mathbf{n}\}$, comme décrit par la figure 3.6,

afin de calculer les informations nécessaires pour quantifier les déformations.

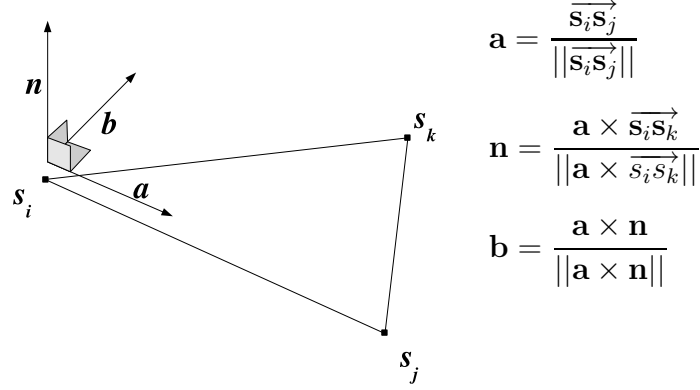


Figure 3.6 – Construction du repère local d'un triangle par de Levy [LM00].

À l'aide de ce repère, il évalue les dérivées partielles en u et v de la fonction de paramétrisation $\mathbf{p}_t(u, v)$ en s'appuyant sur les équations 3.11.

$$\begin{aligned} \partial_u \mathbf{p}_t &= d^{-1} \begin{bmatrix} u_i((\mathbf{s}_j - \mathbf{s}_k) \cdot \mathbf{b}) + u_j((\mathbf{s}_k - \mathbf{s}_i) \cdot \mathbf{b}) + u_k((\mathbf{s}_i - \mathbf{s}_j) \cdot \mathbf{b}) \\ u_i((\mathbf{s}_j - \mathbf{s}_k) \cdot \mathbf{a}) + u_j((\mathbf{s}_k - \mathbf{s}_i) \cdot \mathbf{a}) + u_k((\mathbf{s}_i - \mathbf{s}_j) \cdot \mathbf{a}) \end{bmatrix} \\ \partial_v \mathbf{p}_t &= d^{-1} \begin{bmatrix} v_i((\mathbf{s}_j - \mathbf{s}_k) \cdot \mathbf{a}) + v_j((\mathbf{s}_k - \mathbf{s}_i) \cdot \mathbf{a}) + v_k((\mathbf{s}_i - \mathbf{s}_j) \cdot \mathbf{a}) \\ v_i((\mathbf{s}_j - \mathbf{s}_k) \cdot \mathbf{b}) + v_j((\mathbf{s}_k - \mathbf{s}_i) \cdot \mathbf{b}) + v_k((\mathbf{s}_i - \mathbf{s}_j) \cdot \mathbf{b}) \end{bmatrix} \end{aligned} \quad (3.11)$$

avec $d = ((\mathbf{s}_j - \mathbf{s}_i) \cdot \mathbf{a}) \cdot ((\mathbf{s}_k - \mathbf{s}_i) \cdot \mathbf{b}) - ((\mathbf{s}_k - \mathbf{s}_i) \cdot \mathbf{a}) \cdot ((\mathbf{s}_j - \mathbf{s}_i) \cdot \mathbf{b})$

Il peut maintenant mesurer les déformations en angles ζ_t (équation 3.12) et en distances ξ_t (équation 3.13) pour chaque triangle t .

$$\zeta_t = \mathcal{A}_t^2 \cdot (\partial_u \mathbf{p}_t \cdot \partial_v \mathbf{p}_t)^2 \quad (3.12)$$

$$\xi_t = \mathcal{A}_t^2 \cdot ((\partial_u \mathbf{p}_t^2 - 1)^2 + (\partial_v \mathbf{p}_t^2 - 1)^2) \quad (3.13)$$

avec $\mathcal{A}_t^2 = \|\overrightarrow{\mathbf{s}_i \mathbf{s}_j} \times \overrightarrow{\mathbf{s}_i \mathbf{s}_k}\|^2 / 4$, l'aire au carré du triangle t .

Pour finir, Levy ré-exprime les équations de Tutte (équations 3.1 et 3.2) sous la forme des moindres carrés (équation 3.14) auxquelles il ajoute les deux termes quantifiant les déformations des angles ζ et des distances ξ . Le premier terme de l'équation 3.14 représente donc la paramétrisation ne tenant compte que des informations topologiques, tandis que les deux termes suivants servent à introduire la gestion des déformations en angles pour

le second et en distance pour le troisième.

$$\varepsilon = \sum_{i=1}^n \left(\left(1 + \sum_{j=1}^{d_i} \frac{u_{ij}}{d_i} \right)^2 + \left(1 + \sum_{j=1}^{d_i} \frac{v_{ij}}{d_i} \right)^2 \right) + \sum_{t \in T} \Phi_t \cdot \zeta(t) + \sum_{t \in T} (1 - \Phi_t) \cdot \xi(t) \quad (3.14)$$

Le paramètre Φ_t permet d'exprimer la prise en compte des déformations comme une combinaison linéaire de ζ_t et de ξ_t , permettant ainsi une grande liberté dans la sélection du type de déformation prise en compte. On pourrait imaginer, par exemple, utiliser un “ curseur ” pour choisir de privilégier la préservation des angles ou des distances dans la combinaison linéaire. De plus, le coefficient Φ_t étant défini pour chaque triangle t , Levy propose dans la suite de son article une méthode de recalage local interactif de la paramétrisation.

Par ailleurs, les méthodes à base de minimisation d'énergie permettent la mise en place de processus itératifs de minimisation de critères particuliers, tel qu'un coefficient de déformation de contraction du maillage [SSGH01, YBS04] $r(u, v) = \sqrt{\Gamma(u, v)^2 + \gamma(u, v)^2}$ avec $\Gamma(u, v)^2$ et $\gamma(u, v)^2$ les valeurs propres du tenseur métrique de la paramétrisation. La figure 3.7 illustre les résultats obtenus par la méthode décrite dans [YBS04]. De plus, cette approche, économe en temps de calcul et itérative, permet la participation d'un utilisateur pour la mise en place de procédures interactives de correction de plaquage de texture [LM00].

3.2.2 Paramétrisation de surfaces fermées

3.2.2.1 Paramétrisation sphérique

Comme nous l'avons vu, les méthodes décrites précédemment sont limitées à la paramétrisation de surfaces topologiquement équivalentes à un disque. Pour notre application, cette restriction n'est pas réellement contraignante puisque les zones de contact des surfaces osseuses se situent sur la tête des os et nous pouvons donc nous contenter de découper cette partie du maillage et ainsi remplir la contrainte topologique. Cependant, dans un souci de généricité et devant le nombre important de modèles concernés nous avons souhaité nous intéresser aux maillages fermés et donc à la paramétrisation sphérique.

Naturellement, la paramétrisation sphérique de surfaces topologiquement équivalentes à une sphère a été largement étudiée ces dernières années. Et on distingue plusieurs types d'approche. La paramétrisation de surfaces fermées ne peut pas reposer sur l'exploitation directe des méthodes planaires puisque par définition une surface fermée ne dispose pas d'une frontière. Il n'est donc pas possible de réaliser le calcul des paramètres en deux étapes, d'abord la frontière puis le maillage intérieur. Cependant, la première idée

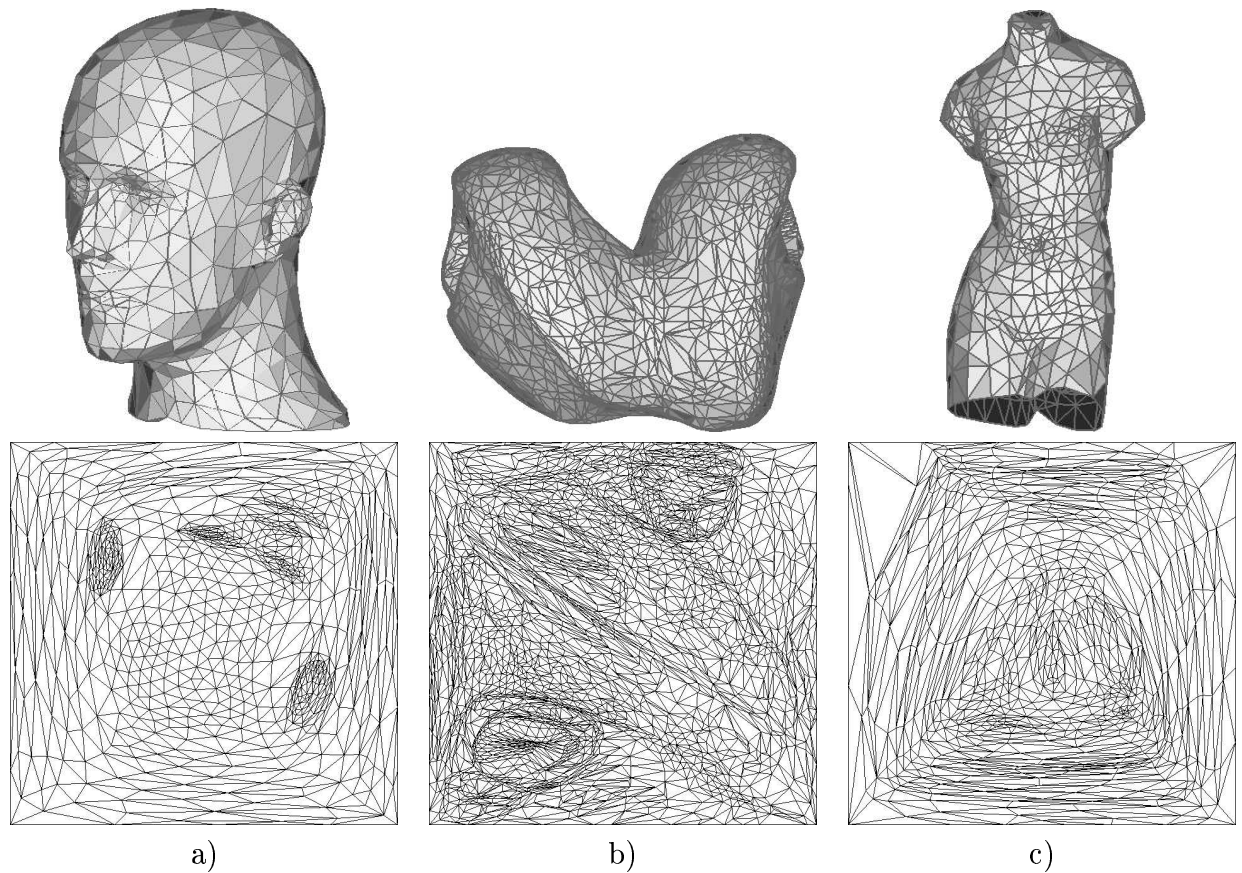


Figure 3.7 – Illustrations des résultats de la paramétrisation - a) 689 sommets, b) 711 sommets, c) 2000 sommets.

a consisté à faire apparaître une frontière sur le maillage afin de pouvoir réexploiter les travaux antérieurs sur la paramétrisation planaire. Haker et al. [HAT⁺00] proposent une paramétrisation planaire de la sphère en utilisant un des triangles comme frontière. Puis, ils utilisent une projection stéréographique pour revenir à la sphère unité.

Isenburg et al. [IGG01] commencent par découper le maillage en deux, puis ils projettent la frontière ainsi créée sur un cercle unité, afin de paramétrer les deux demi-maillages sur les hémisphères de la sphère unité. Ces méthodes présentent l'avantage d'être faciles à implanter et de reposer sur un formalisme bien établi.

L'autre solution consiste à développer des méthodes exploitant les caractéristiques de la sphère. Là encore, une première voie consiste à étendre les résultats obtenus sur les maillages ouverts. La généralisation de la méthode des coordonnées barycentriques à la sphère a été démontrée par Gotsman et al. [GGS03]. Cependant, ils ne proposent pas de méthode de résolution du système quadratique qu'ils obtiennent. Une méthode de résolution est présentée dans [SYGS05], elle repose sur une estimation initiale de la

paramétrisation obtenue, puis sur la résolution du système par des processus itératifs. La solution initiale est obtenue comme dans [IGG01] par la découpe et la paramétrisation des deux demi-maillages.

Une seconde approche repose sur l'utilisation de maillages en multi-résolution. Cette technique repose sur la construction d'un maillage en multi-résolution par simplification du maillage jusqu'à obtenir un maillage simple, dont on peut aisément définir la paramétrisation comme un tétraèdre par exemple. Les premiers travaux [ST98] ne s'intéressent qu'à la topologie du maillage lors de la simplification, alors qu'il est possible d'introduire des étapes de relaxation permettant de minimiser la déformation de la paramétrisation [PH03].

Enfin, une dernière approche s'intéresse plus généralement aux maillages de 2-variété et consiste à découper le maillage en plusieurs patches topologiquement équivalents à des disques, puis à calculer la paramétrisation de chacun de ces patches, on utilise généralement des patches triangulaires. Deux problèmes se posent alors, disposer d'une découpe appropriée et obtenir une paramétrisation continue sur les bordures des patches.

Les premiers travaux traitent du problème de continuité en projetant l'ensemble déplié des patches sur un domaine paramétrique simple (généralement le plan). Eck et al. [EDD⁺95] proposent d'utiliser une triangulation de Delaunay duale au diagramme de Voronoi de points d'intérêt. Schröder et al. proposent deux autres méthodes; dans un premier article [LSS⁺98] ils construisent les patches triangulaires par simplification du maillage initial et dans un second article [PSS01], ils utilisent des informations fournies par l'utilisateur afin de déterminer les zones d'intérêt. Plus récemment, Khodakovsky et al. [KLS03] et Gu et al. [GY03] ont proposé deux méthodes de paramétrisation globale offrant un lissage efficace des déformations locales. Cette solution impose généralement un découpage de la paramétrisation afin de parvenir au final à une paramétrisation plane.

La méthode que j'ai appliquée propose de découper le maillage selon son équateur afin d'obtenir deux maillages ouverts. je peux ensuite paramétriser ces deux maillages sur deux disques unités à l'aide des méthodes décrites précédemment. Les coordonnées paramétriques de ces deux maillages sont ensuite projetées sur les deux hémisphères de la sphère unité par projection sphérique afin d'obtenir pour le maillage initial une paramétrisation sphérique. La continuité de la paramétrisation le long de l'équateur est assurée par l'utilisation des mêmes coordonnées paramétriques pour les points de l'équateur des deux demi-maillages. J'utilise, pour finir, un processus itératif décrit dans [YBS04], afin de lisser l'équateur d'une part et de minimiser la déformation d'autre part. Un exemple de paramétrisation sphérique obtenue est illustrée par la figure 3.8.

Cependant, les méthodes reposant sur la décomposition en multi-résolution du maillage

par simplification, comme [ST98] ou [PH03], peuvent être des méthodes intéressantes puisqu'elles semblent permettre d'obtenir une répartition régulière et homogène des maillages sur l'espace paramétrique.

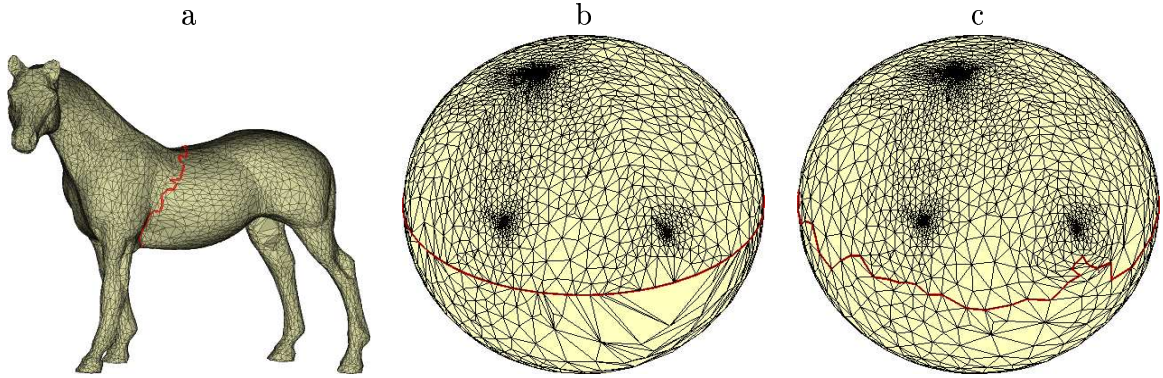


Figure 3.8 – Exemple de paramétrisation sphérique - a) le maillage 3D, b) la paramétrisation des deux hémisphères, c) après minimisation de la déformation [YBS04].

Afin de pouvoir interpoler un maillage triangulaire à partir de ses coordonnées paramétriques, et de disposer ainsi d'une surface continue à paramétrage global, j'ai mis en place des procédures permettant d'obtenir la paramétrisation d'un maillage triangulaire topologiquement équivalent à un disque ou à une sphère.

Dans l'optique de disposer d'une méthode de paramétrisation unifiée, permettant de pouvoir tester différentes pondérations pour un faible coût de développement, j'ai utilisé les méthodes à base de combinaisons convexes. Les systèmes caractéristiques de ces méthodes ont été résolus grâce à une bibliothèque de gestion des matrices creuses appelée **SparseLib++** [DLN⁺94] qui induit des temps de calculs raisonnables.

3.3 Interpolation

Nous disposons maintenant d'un paramètre $\mathbf{w}_i$ pour chaque sommet $\mathbf{s}_i = (u_i, v_i)$ du maillage triangulaire initial avec $i \in \{1, \dots, n\}$. L'étape suivante consiste à utiliser ces paramètres pour construire une surface paramétrique continue interpolant les n données discrètes initiales. Je tiens à rappeler que ces données peuvent être réparties de manière hétérogène dans l'espace des données, ainsi que dans le domaine paramétrique, et que nous souhaitons pouvoir disposer d'une continuité C^2 sur la surface résultante.

Par ailleurs, afin de s'affranchir de la gestion de frontière sur les patches au cours de la simulation, j'ai rejeté les méthodes reposant sur un paramétrage local à base de patches.

Nous cherchons donc à obtenir une surface $\mathbf{s} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ telle que :

- $\mathbf{s}(\mathbf{w}_i) = \mathbf{s}_i, \quad \forall i \in \{1, \dots, n\}$.
- $\mathbf{s}(\mathbf{w})$ s'applique à des densités hétérogènes de données.
- $\mathbf{s}(\mathbf{w})$ soit dérivable au moins deux fois.

3.3.1 Surface paramétrique à base radiale

La contrainte de non restriction sur la répartition ou la topologie du maillage initial à un maillage régulier nous interdit l'utilisation de techniques d'interpolation locale à base de patches, comme les surfaces paramétriques locales. De plus, afin d'exprimer la contrainte de contact, il nous faut disposer d'une expression continûment différentiable de la surface interpolante. Or, pour assurer cette continuité, les techniques à base de patches doivent imposer des contraintes sur les bordures, se traduisant par des contraintes d'alignement sur le maillage. Par ailleurs, des aberrations géométriques sont susceptibles d'apparaître à cause de la répartition non régulière des sommets du maillage triangulaire. J'ai donc orienté mon travail sur des méthodes d'interpolation globale.

Les fonctions à base radiale (Radial Basis Functions ou RBFs) sont largement utilisées pour construire des surfaces implicites à partir d'un ensemble discret de sommets [CBC⁺01]. Ces méthodes consistent à définir un champ de potentiel continu signé $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ dans l'espace 3D qui s'annule en chaque sommet ($f(\mathbf{s}_i) = 0 \quad \forall i = 1, \dots, n$). Afin d'obtenir une surface autre que la solution triviale $f(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \mathbb{R}^3$, il est nécessaire de compléter les données initiales avec des points supplémentaires se trouvant à l'intérieur ($f(\mathbf{x}) < 0$) et à l'extérieur ($f(\mathbf{x}) > 0$) de la surface. Cette méthode accroît significativement la taille du système linéaire à résoudre pour obtenir les coefficients de la surface. De plus, la description implicite de la surface rend impossible le suivi continu du point de contact dans le temps, qui constitue une condition nécessaire à l'expression des contraintes au sein du formalisme Lagrangien.

Cependant, les RBFs disposent de propriétés intéressantes comme l'indépendance vis-à-vis de la répartition géométrique et topologique des données initiales. Je propose donc une nouvelle utilisation des RBFs pour définir une surface paramétrique qui interpole un maillage triangulaire 3D. À notre connaissance, cette démarche est véritablement originale au sein de la communauté scientifique de la modélisation géométrique, les seuls travaux qui pourraient s'en rapprocher figurent dans le domaine du traitement d'images [NN00, RM95].

Dans cette méthode, la surface reconstruite n'est pas à considérer comme un champ scalaire défini sur l'espace 3D mais comme une bijection entre un sous-ensemble du plan (espace paramétrique) et celle-ci. Notre surface paramétrique à base radiale (SPBR) $\mathbf{s}(\mathbf{w})$

est donc définie comme suit :

$$\mathbf{s} : \begin{cases} \mathbb{R}^2 \rightarrow \mathbb{R}^3 \\ \mathbf{w} \rightarrow \mathbf{s}(\mathbf{w}) = \mathbf{p}(\mathbf{w}) + \sum_{i=1}^n \lambda_i (\phi_i \circ d)(\mathbf{w}_i, \mathbf{w}) \end{cases} \quad (3.15)$$

Dans l'équation 3.15 on utilise le polynôme $\mathbf{p} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ d'approximation de la solution. Celui-ci peut généralement être négligé.

Au sein de cette écriture paramétrique, le terme λ_i représente le coefficient 3D de pondération du sommet $\mathbf{s}_i$ du maillage triangulaire initial. ϕ_i correspond à la $i^{\text{ème}}$ fonction radiale de la base, avec pour paramètre la distance calculée à l'aide de la fonction d entre le point d'évaluation $\mathbf{w}$ et $\mathbf{w}_i$, le vecteur des coordonnées paramétriques du sommet $\mathbf{s}_i$.

Pour construire une SPBR, on dispose donc d'un jeu de points $\{\mathbf{s}_1, \dots, \mathbf{s}_n\}$, pour lesquels on dispose de coordonnées paramétriques $\{\mathbf{w}_1, \dots, \mathbf{w}_n\}$. On décompose p selon la base canonique $\mathcal{B} = \{\mathbf{w}^\alpha\}_{\alpha \in \mathbb{N}^2}$. Ici le degré maximal du polynôme est choisi lors de la construction de la SPBR et on obtient donc un nombre fini t de monômes organisés dans l'ensemble $\{\mathbf{p}_1, \dots, \mathbf{p}_t\}$. Il reste à obtenir les n coefficients 3D de pondération des points de contrôle λ_i et les t coefficients $\mathbf{c}_i$ des monômes de la base du polynôme d'approximation. Pour cela on dispose de deux jeux de contraintes :

- $\mathbf{s}(\mathbf{w}_i) = \mathbf{s}_i, \quad \forall i \in \{1, \dots, n\}.$
- $\sum_{i=1}^n \lambda_i \cdot \mathbf{p}_j(\mathbf{w}_i) = 0 \quad \forall j \in \{1, \dots, t\}.$

La première contrainte est définie par l'interpolation, la seconde sert à compléter le système à l'aide de contraintes d'orthogonalité, simples à formuler. À partir de ces contraintes, on peut définir le système suivant :

$$\begin{bmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\Lambda} \\ \mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix} \quad (3.16)$$

Avec pour inconnues $\boldsymbol{\Lambda}$ et $\mathbf{C}$, les matrices composées respectivement des vecteurs λ_i et $\mathbf{c}_k$ pour $i \in \{1, \dots, n\}$ et $k \in \{1, \dots, t\}$. La matrice $\mathbf{S}$ se compose des n points 3D $\mathbf{s}_i$. Les coefficients des matrices $\mathbf{A}$ et $\mathbf{P}$ sont initialisés comme suit : $\mathbf{A}_{ij} = \phi_i \circ d(\mathbf{w}_i, \mathbf{w}_j)$ et $\mathbf{P}_{ik} = \mathbf{p}_k(\mathbf{w}_i)$ pour $i, j \in \{1, \dots, n\}$ et $k \in \{1, \dots, t\}$

Les coefficients $\mathbf{A}_{ij}$ dépendent des valeurs des fonctions radiales $\phi_i : \mathbb{R}^+ \rightarrow \mathbb{R}^+$. La classe de fonctions radiales la plus répandue est la classe des fonctions étudiées par [Duc77], d'ordre o et en dimension d :

$$\phi_i(r) = \begin{cases} r^{2o-d} \ln r & \text{si } d \text{ est pair} \\ r^{2o-d} & \text{si } d \text{ est impair} \end{cases} \quad (3.17)$$

En 2D et à l'ordre 2, on obtient la fonction spline plaque mince (*thin-plate*) : $\phi_i(r) =$

$r^2 \ln r$, tandis qu'en 3D pour un ordre 2 ou 3, on obtient deux fonctions radiales efficaces pour l'interpolation offrant un bon compromis résultat/performance : la fonction biharmonique $\phi_i(r) = r$ et la fonction triharmonique $\phi_i(r) = r^3$.

Par ailleurs, les fonctions radiales $\phi_i(r)$ conditionnent l'ordre de dérivation de la fonction $\mathbf{s}$, c'est pourquoi de nombreux travaux ([CBC⁺01, Reu03]) proposent une fonction multiquadrique (équation 3.18) comme fonction radiale du fait de ses propriétés de dérivabilité. Dans l'équation 3.18, le terme c_i est le minimum des carrés des distances entre le sommet $\mathbf{s}_i$ et ses voisins $\mathbf{s}_{ij}$.

$$\phi_i(r) = \sqrt{r^2 + c_i} \quad \text{où } c_i = \min(d(\mathbf{w}_i, \mathbf{w}_{ij})^2) \quad (3.18)$$

Il est à remarquer que ces fonctions radiales ne sont pas à support compact, c'est à dire que leur valeur augmente à mesure que r augmente. La matrice $\mathbf{A}$ ne peut pas être creuse. Afin d'obtenir des matrices creuses, pour améliorer les performances, il est intéressant d'obtenir des fonctions radiales décroissantes, qui valent 0 à partir d'une certaine distance du centre, il s'agit des fonctions radiales à support compact. On redéfinit la fonction radiale par l'équation 3.19 avec $\varphi_i(r)$ la fonction radiale à support compact unitaire.

$$\phi_i(r) = \begin{cases} \varphi_i(r) & \text{si } 0 \leq r \leq 1 \\ 0 & \text{si } r > 1 \end{cases} \quad (3.19)$$

Dans [Wen95] puis [Wen05], Wendland propose une étude approfondie de ces fonctions à support compact $\varphi_i(r)$, pour une dimension et une continuité souhaitée. Gelas [Gel06] répertorie ces résultats et propose des exemples de fonctions pour les trois premières dimensions avec différentes continuités.

Pour les travaux de cette thèse, j'ai généralement utilisé la fonction multiquadrique (équation 3.18) pour ces qualités d'interpolation et propriétés de dérivabilité. En effet, nous souhaitons disposer d'au moins une C^2 continuité pour nos surfaces à base radiale. Je n'ai pas pris le temps de mettre en place les fonctions à support compact puisque le choix de la taille du support de ces fonctions a une forte incidence sur le résultat final et que ce choix ne m'a pas semblé trivial. j'ai choisi, pour augmenter la densité des maillages traités, d'employer une méthode hiérarchique : la partition de l'unité (sous-section 3.3.2).

Maintenant que nous pouvons calculer les coefficients $\mathbf{A}_{ij}$ de la matrice $\mathbf{A}$, il ne reste plus qu'à résoudre le système 3.16 pour obtenir les valeurs des matrices $\mathbf{A}$ et $\mathbf{C}$. Une fois ces coefficients obtenus, nous pouvons les injecter dans l'équation 3.15 afin d'obtenir pour tout point $\mathbf{w}$ du domaine paramétrique un point $\mathbf{s}(\mathbf{w})$ de la surface 3D.

Pour finir, je souhaite disposer pour le suivi continu du point de contact sur la surface des dérivées premières de la surface. Dans le cadre général, pour un point de l'espace paramétrique $\mathbf{w}$ composé de d coordonnées $u^{(k)}$ avec $k \in \{1, \dots, d\}$, on obtient par dérivation de l'équation 3.15 le résultat suivant :

$$\partial_{u^{(k)}} \mathbf{s}(\mathbf{w}) = \partial_{u^{(k)}} \mathbf{p}(\mathbf{w}) + \sum_{i=1}^n \lambda_i (\phi'_i \circ d)(\mathbf{w}_i, \mathbf{w}) \partial_{u^{(k)}} d(\mathbf{w}_i, \mathbf{w}) \quad \text{avec } k \in \{1, \dots, d\} \quad (3.20)$$

L'expression de la dérivée dépend donc de la dérivée partielle du polynome $\partial_{u^{(k)}} \mathbf{p}(\mathbf{w})$ et de la dérivée ϕ'_i de la fonction radiale, ainsi que de la dérivée partielle de la fonction distance $\partial_{u^{(k)}} d(\mathbf{w}_i, \mathbf{w})$. En tenant compte de ces besoins, je propose l'implantation illustrée par la figure 3.9, qui me semble être un bon compromis entre modularité et efficacité.

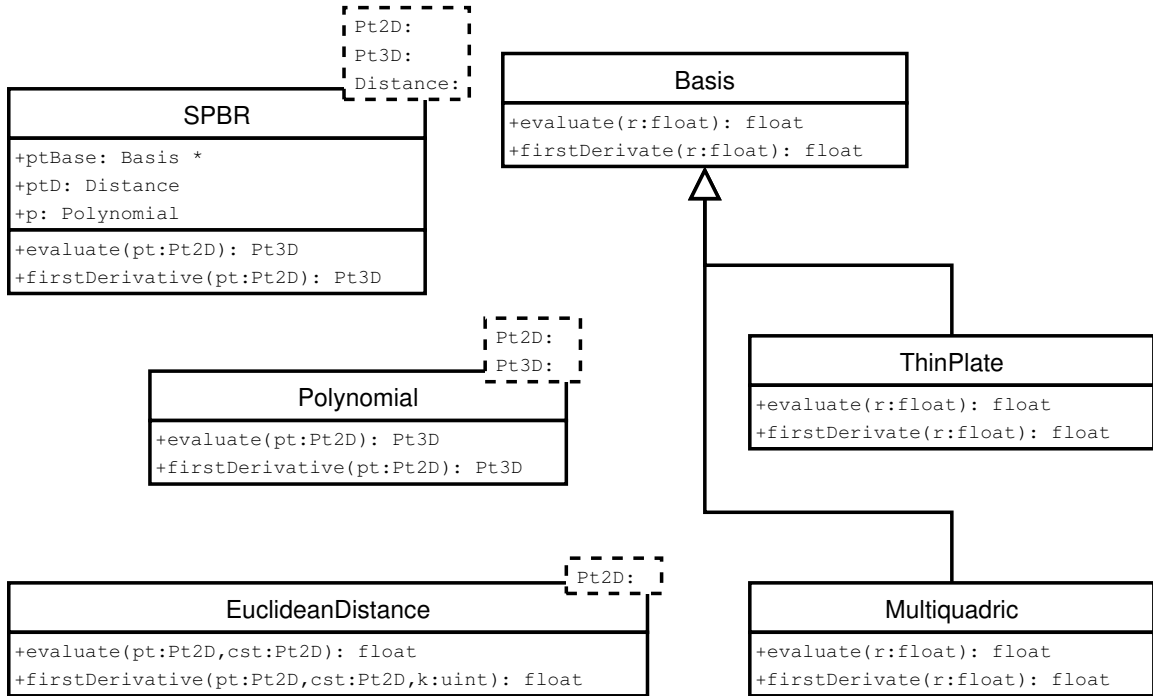


Figure 3.9 – Diagramme UML de l'implantation des surfaces paramétriques à base radiale.

Une surface paramétrique à base radiale repose sur un polynôme, un calcul de distance et un type de fonction radiale pour la base.

La classe **Polynomial** peut être construite de manière générique, afin de définir l'ordre du polynôme à sa création. Ainsi, en l'intégrant à la classe **SPBR**, il sera possible de définir l'ordre souhaité pour le polynôme, lors de l'exécution du programme, à la construction de la surface.

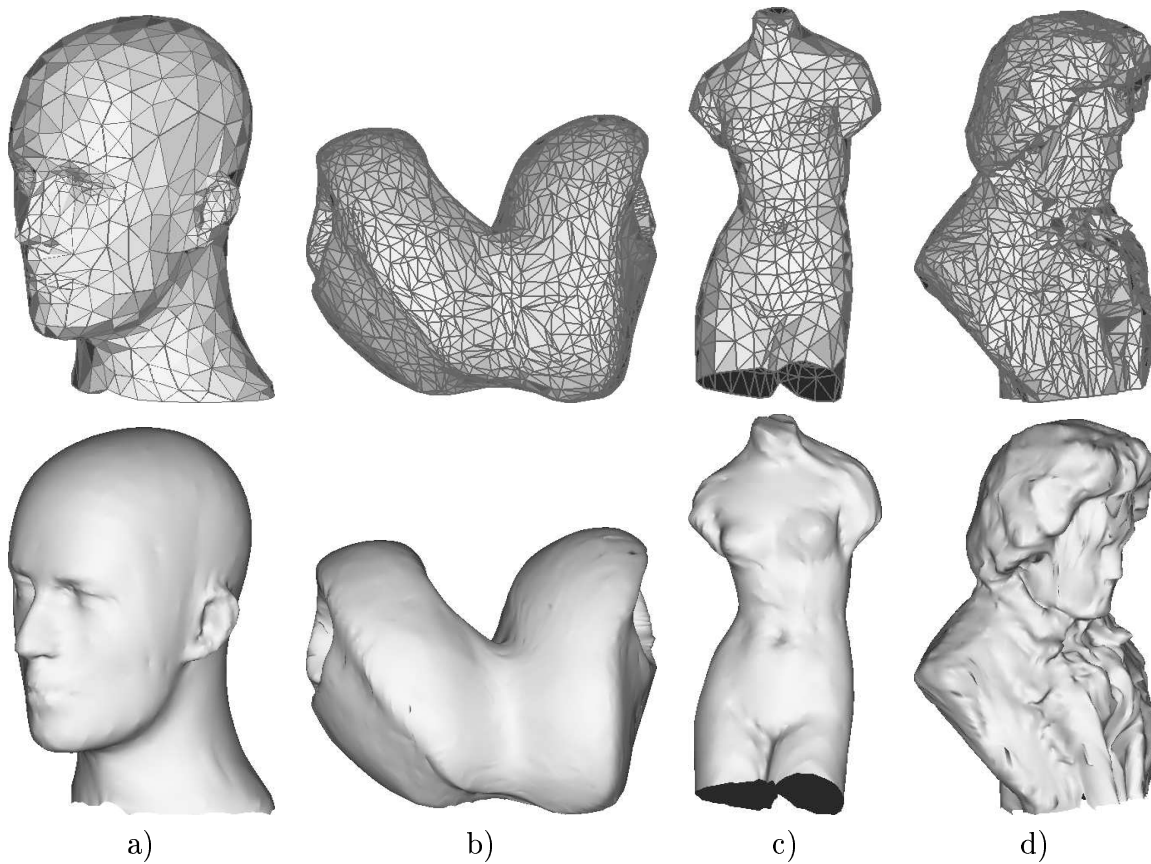


Figure 3.10 – Illustration des résultats de l'interpolation par surface paramétrique à base radiale - a) 689 sommets, b) 711 sommets, c) 2000 sommets, d) 2173 sommets.

D'autre part, je souhaite conserver la liberté de pouvoir sélectionner le type de distance employée (euclidienne ou sphérique par exemple). Mais ce choix reste dépendant du type de paramétrisation employé, critère vraisemblablement connu à l'implantation du programme. C'est pourquoi nous avons intégré le calcul de distance à la classe **SPBR** sous la forme d'un *template*. Ainsi, le programme reste efficace, puisque la spécialisation est effectuée lors de la précompilation, tout en conservant la modularité même si la sélection doit se faire lors du développement.

Pour finir, cette implantation ayant été réalisée dans un but d'expérimentation, je souhaite pouvoir sélectionner de manière interactive le type des fonctions radiales pour la base. J'ai donc dû mettre en place une interface de classe pour gérer les types des fonctions radiales, la classe abstraite **Basis**, qui est ensuite implantée par des classes dérivées **ThinPlate** ou **Multiquadric** par exemple. Il suffit donc ensuite de changer l'objet pointé par le pointeur du type de fonction de base de la classe **SPBR** au cours de l'exécution. On dispose alors d'un mécanisme de sélection interactif au prix de l'alourdissement des appels de fonction de ces classes, dû aux fonctions virtuelles.

Maillage	Nombre de sommets	Nombre de faces	Temps moyen	Écart type
a)	689	1 355	4 645	116
b)	711	1 396	4 807	123
c)	2 000	3 961	106 470	2 693
d)	2 173	4 290	130 221	3 254

Tableau 3.11 – Temps de calcul (en msec) des coefficients des SPBRs de la figure 3.10 pour 20 calculs. Configuration matérielle utilisée : Athlon XP 2200+, 512 Mo de RAM.

Je tiens à faire remarquer que dans un souci d'efficacité, il serait plus intéressant d'effectuer la sélection du type des fonctions radiales de la base lors du développement, en intégrant le type des fonctions radiales aux *templates* de la classe **SPBR**. On pourrait alors tirer partie des avantages des fonctions *inline* et abandonner les fonctions virtuelles. L'impact sur l'ergonomie serait assez faible puisqu'on connaît généralement à l'avance la fonction radiale que l'on souhaite utiliser, puis on en change rarement. Cependant, on peut noter qu'il est à priori possible d'utiliser autant de types de fonctions radiales que de points de contrôle. Et dans ce cas la seule solution reste un tableau de pointeurs de **Basis**, derrière lesquels se trouvent les différents types de fonctions radiales.

La figure 3.10 illustre les résultats que l'on obtient par interpolation des sommets paramétrés à partir de surfaces paramétriques à base radiale. Cependant, comme l'illustre le tableau 3.11, présentant les temps de calcul des SPBRs pour ces illustrations, afin de conserver des temps de calculs raisonnables, il est préférable de travailler sur des maillages de quelques centaines de sommets. Plusieurs alternatives existent pour pallier ces limitations. Je présente dans la sous-section suivante l'une de ces méthodes 3.3.2 qui m'a semblé la plus adaptée.

3.3.2 Partition de l'unité

Je propose dans ce chapitre une amélioration de la méthode d'interpolation continue, adaptée à des maillages plus denses, basée sur la technique d'évaluation rapide des SPBR à partir de la méthode des partitions de l'unité introduite par Wendland [Wen02a, Wen02b].

La méthode des partitions de l'unité repose sur le principe de décomposition d'un problème complexe en plusieurs sous-problèmes plus simples à résoudre, puis à fusionner les sous-résultats obtenus afin d'obtenir le résultat global. Ceci se traduit ici par la décomposition du domaine paramétrique en plusieurs sous-domaines sur lesquels nous construisons des surfaces continues interpolantes. Nous disposons ainsi de plusieurs surfaces locales que nous fusionnons pour obtenir la surface interpolante globale.

Je commencerai donc par présenter la méthode des partitions de l'unité. La première étape consiste à subdiviser le domaine paramétrique $\mathcal{P}$ en m sous-domaines $\mathcal{P}_i$ se chevauchant, de telle sorte que $\mathcal{P} \subseteq \bigcup_{i=1}^m \mathcal{P}_i$. Nous pouvons ensuite construire une partition de l'unité sur cet ensemble de sous-domaines paramétriques comme définie par l'équation 3.21.

$$\mathbf{s}(\mathbf{w}) = \frac{\sum_{i=1}^m \mathbf{f}_i(\mathbf{w}) w_i(\mathbf{w})}{\sum_{i=1}^m w_i(\mathbf{w})} \quad (3.21)$$

Nous obtenons donc une somme normalisée pondérée des fonctions d'interpolation locales $\mathbf{f}_i(\mathbf{w})$, dans notre cas, des SPBR. La pondération s'effectue à l'aide des fonctions à support compact $w_i(\mathbf{w})$ sur lesquelles je reviendrai plus tard. Dans un premier temps, intéressons-nous à la subdivision du domaine paramétrique.

Je rappelle que nous nous intéressons pour l'instant à un domaine paramétrique 2D, correspondant au carré unité. La subdivision la plus naturelle et la plus simple à mettre en place est la subdivision reposant sur une grille régulière. Cependant, cette solution peut poser problème dans le cas d'une répartition très hétérogène, puisqu'alors certains sous-domaines risquent de ne pas disposer de suffisamment d'échantillons pour réaliser une interpolation efficace.

Nous devons donc utiliser une subdivision adaptative en fonction des données à interpoler. Nous avons utilisé un schéma adaptatif de gestion de données couramment utilisé en informatique : le *quadtree* [Mea82].

Il s'agit de subdiviser récursivement chaque domaine d'intérêt en quatre sous-domaines lorsque celui-ci remplit un critère particulier. Dans notre cas, chaque domaine doit être subdivisé s'il contient plus de n_{max} points. La figure 3.12 illustre sur la ligne du haut les quatre premières étapes de la construction d'un *quadtree*.

Pour la partition de l'unité, nous avons besoin que les sous-domaines se chevauchent, nous introduisons alors un coefficient c quantifiant la valeur du chevauchement, nous permettant d'augmenter la dimension des sous-domaines. L'algorithme de construction reste donc le même, comme l'illustre la ligne du bas de la figure 3.12.

Notons que, pour pallier le problème précédent de sous-échantillonnage d'un sous-domaine, nous augmentons dans une phase finale la valeur locale du coefficient c_i pour tout sous-ensemble $\mathcal{P}_i$ n'ayant pas au moins n_{min} points, jusqu'à ce qu'il satisfasse la condition. Nous pouvons ainsi contrôler que tous les sous-ensembles disposent de suffisamment de points pour que l'interpolation locale soit pertinente.

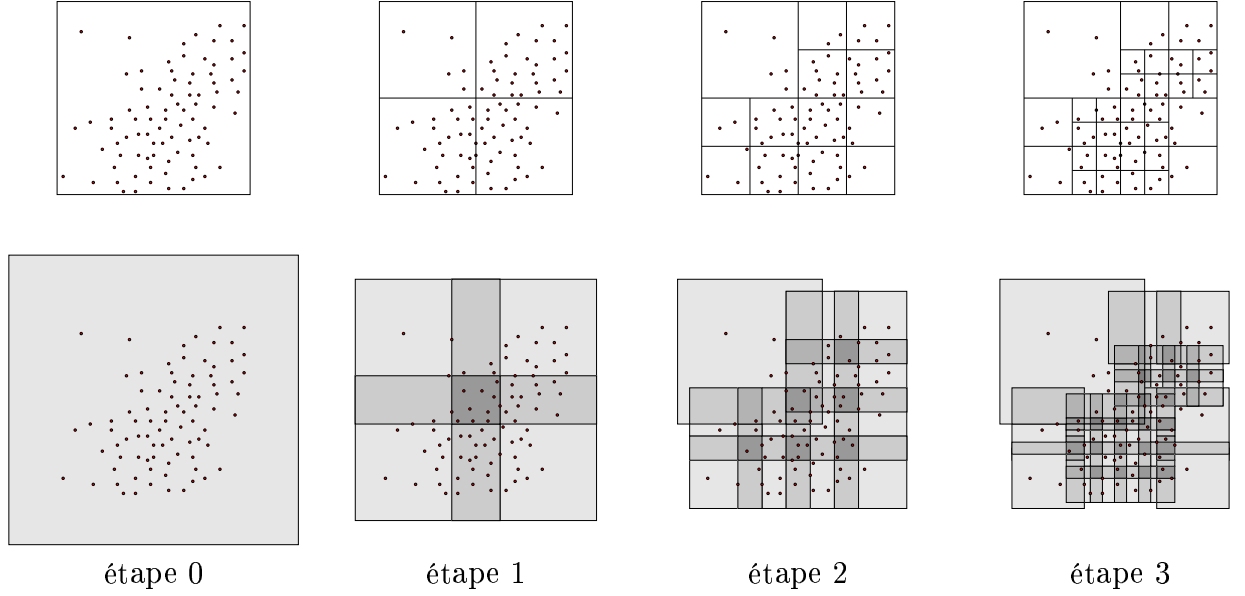


Figure 3.12 – Les quatre premières étapes de la construction du *quadtree* - traditionnel sur la rangée supérieure et avec chevauchement sur la rangée inférieure.

Tous ceci nous permet, d'une part, de réduire le temps de construction de la surface puisque nous construisons plusieurs petites surfaces interpolantes plutôt qu'une seule grande et d'autre part, de réduire le temps d'évaluation de la surface globale puisque nous pouvons profiter de l'arborescence du *quadtree* pour n'évaluer que les SPBR $\mathbf{f}_i(\mathbf{w})$ locales qui participent à l'évaluation finale $\mathbf{s}(\mathbf{w})$. L'évaluation finale est réalisée à l'aide de l'équation 3.21 avec $\{\mathbf{f}_i(\mathbf{w})\}$ pour $i \in \{0, \dots, m\}$ l'ensemble des surfaces continues d'interpolation. Les fonctions de pondération $w_i(\mathbf{w})$ sont définies par l'équation 3.22.

$$w_i(\mathbf{w}) = \begin{cases} \phi(d_i(\mathbf{w})) & \text{si } \mathbf{w} \in \mathcal{P}_i \\ 0 & \text{sinon} \end{cases} \quad (3.22)$$

C'est cette fonction qui assure la continuité de la surface globale $\mathbf{s}(\mathbf{w})$ lors des chevauchements de surfaces locales $\mathbf{f}_i(\mathbf{w})$. Pour cela, nous utilisons une fonction de distance $d_i(\mathbf{w})$ pour le $i^{\text{ème}}$ sous-domaine paramétrique et une fonction décroissante ϕ . Le calcul de la distance est défini comme valant 0 au centre du sous-domaine et 1 sur sa frontière et au-delà, ainsi la fonction décroissante ϕ n'a besoin d'être définie décroissante que sur l'intervalle $[0, 1]$.

L'utilisation classique du *quadtree* aboutit à des sous-domaines de forme carrée et il devient donc difficile d'utiliser la distance euclidienne pour parvenir au résultat souhaité. Deux solutions sont alors envisageables :

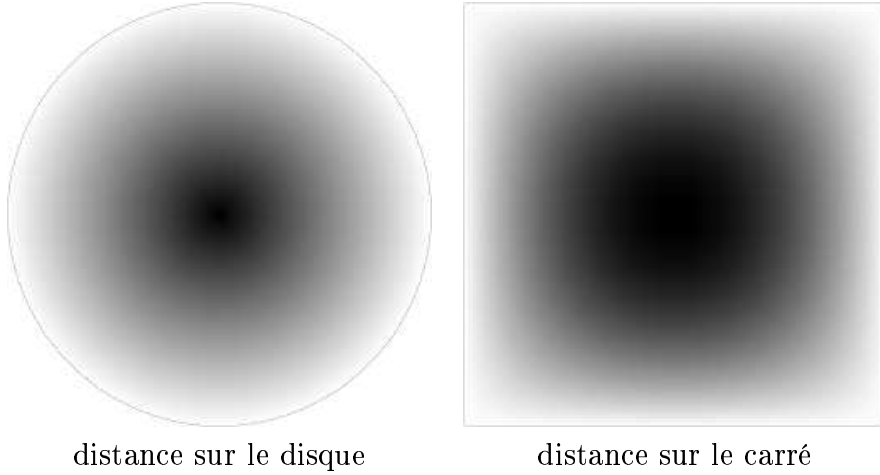


Figure 3.13 – Fonctions de distance utilisées pour les partitions de l’unité, avec la convention 0 :noir→ 1 :blanc.

- La première consiste à construire des sous-domaines circulaires circonscrits aux carrés initiaux, il devient alors possible d’utiliser la distance euclidienne pour définir $d_i(\mathbf{w})$ comme illustré par l’équation 3.23 avec $\mathbf{w}_{c_i}$ le centre du sous-domaine et r_i le rayon du sous domaine.
- L’autre méthode consiste à définir un calcul de distance dans un carré.

Dans ce but je rappelle que puisque la subdivision s’effectue sur le carré unité aligné sur les axes, chacun des sous-domaines $\mathcal{P}_i$ peut être défini par deux points : le coin le plus petit $\mathbf{w}_i^{min} = (u_i^{min}, v_i^{min})$ et le coin le plus grand $\mathbf{w}_i^{max} = (u_i^{max}, v_i^{max})$. On peut dès lors exprimer la distance grâce à l’équation 3.24. La figure 3.13 présente le résultat de ces deux fonctions de distance sur leurs domaines respectifs.

$$d_i^{sphère}(\mathbf{w}) = \frac{\|\mathbf{w} - \mathbf{w}_{c_i}\|}{r_i} \quad (3.23)$$

$$d_i^{carrée}(\mathbf{w}) = 1 - \frac{16(u - u_i^{min})(u_i^{max} - u)(v - v_i^{min})(v_i^{max} - v)}{(u_i^{max} - u_i^{min})^4} \quad (3.24)$$

D’après l’équation 3.22, nous n’évaluons jamais de point n’appartenant pas au sous-domaine avec les fonctions $\phi(x)$ et d_i . Les résultats de d_i sont donc compris entre 0 et 1. Les fonctions $\phi(x)$ doivent donc être décroissantes sur l’intervalle $[0, 1]$. La figure 3.14 propose quelques exemples, de fonctions $\phi(x)$.

Notons que le calcul de la dérivée partielle de la surface globale $\partial_{u^{(k)}} \mathbf{s}(\mathbf{w})$, pour la $k^{\text{ème}}$ coordonnée $u^{(k)}$ de l’espace paramétrique, nécessite la même dérivée partielle pour la fonction de pondération $\partial_{u^{(k)}} w_i(\mathbf{w})$: celles-ci dépendent de la dérivée de la fonction

décroissante ϕ , il faut donc prendre soin de sélectionner une fonction décroissante avec un ordre de dérivabilité suffisant.

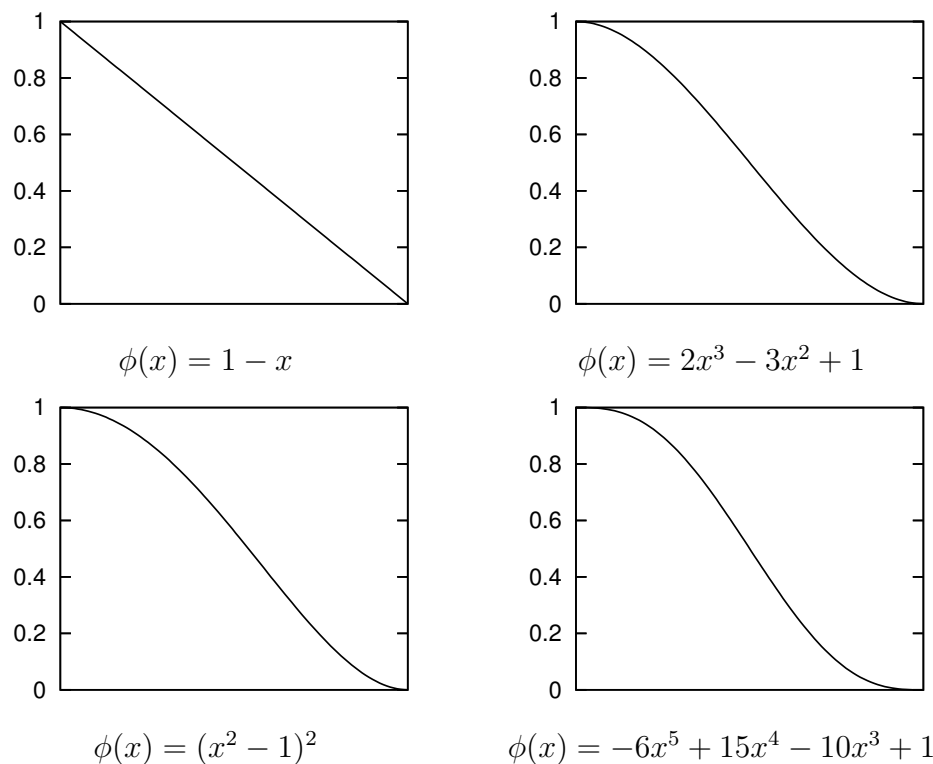


Figure 3.14 – Exemples de fonctions décroissantes utilisées dans les partitions de l’unité.

J’ai implanté ces travaux, structurés selon l’architecture illustrée par la figure 3.15. Je souhaite toujours conserver une certaine modularité structurelle afin de pouvoir interchanger les composants à moindre coût. L’élément qui a le plus gros impact sur la structure des partitions de l’unité est le modèle de subdivision qui est employé. En effet, il est choisi en fonction du domaine paramétrique et c’est lui qui impose les calculs de distance.

J’ai donc utilisé des patrons de classe (*template*) pour les choix des distances, puisque l’on connaît le type de distance souhaitée lors de l’implantation des méthodes du modèle de subdivision. Et j’ai employé le polymorphisme pour les choix dépendant du domaine paramétrique comme les modèles de subdivision. Je peux ainsi en changer en fonction du maillage chargé lors de l’exécution. L’utilisation des fonctions virtuelles impose un surcoût à l’appel des fonctions.

Au final, nous disposons d’une classe pour les partitions de l’unité (**PartitionOfUnity**) dépendante d’un noyau de subdivision (**SubdivisionKernel**). C’est lui qui détermine

le schéma de subdivision du domaine paramétrique, et qui évalue et fusionne les sous-domaines. Nous avons implanté le *quadtree* avec interpolation locale à base de surfaces paramétriques à base radiale (**QuadTree_SPBR**).

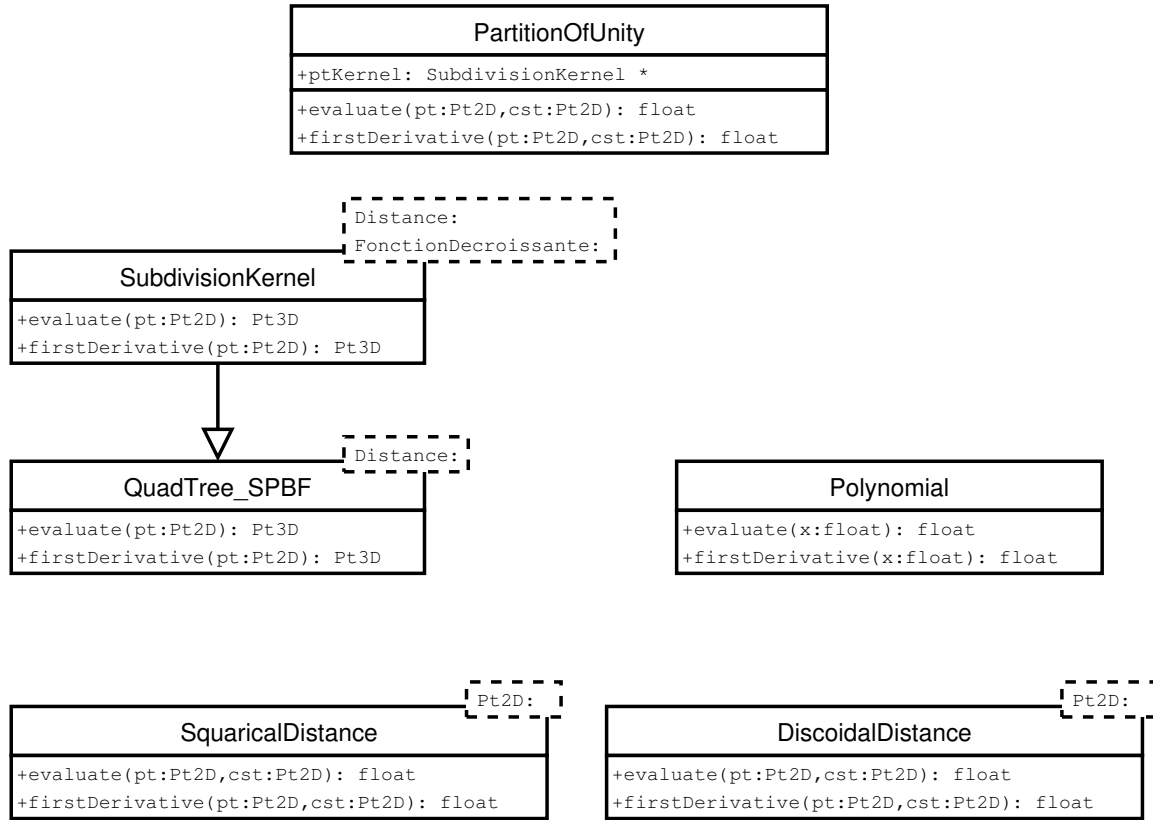


Figure 3.15 – Diagramme UML de l’implantation des partitions de l’unité.

Les noyaux de subdivision doivent disposer d’un polynôme (**Polynomial**) construit selon la fonction décroissante $\phi(x)$ souhaitée et d’un calcul de distance que l’on peut spécialiser dans notre cas en distance sur le disque (**DiscoidalDistance**) ou le carré (**SquaricalDistance**).

Notons qu’il serait intéressant de définir la fonction d’interpolation locale utilisée par le noyau de subdivision par un *template* afin de pouvoir tester facilement d’autres méthodes d’interpolation. Mais cette solution est difficile à mettre en place à cause des données nécessaires à l’interpolation. En effet, les SPBR supposent un accès à l’ensemble des points du sous-domaine, information qui pour cause du chevauchement est redondante. Pour limiter la taille du stockage, j’ai stocké l’ensemble des points dans la classe **QuadTree_SPBR** à la racine de l’arbre, puis chaque feuille de l’arbre accède aux données à l’aide de pointeurs. Cette architecture étant très spécifique aux SPBR, il devient très difficile de l’appliquer à tout type de fonction d’interpolation. En définitive, j’ai dû

sacrifier à la modularité pour limiter l'utilisation de la mémoire.

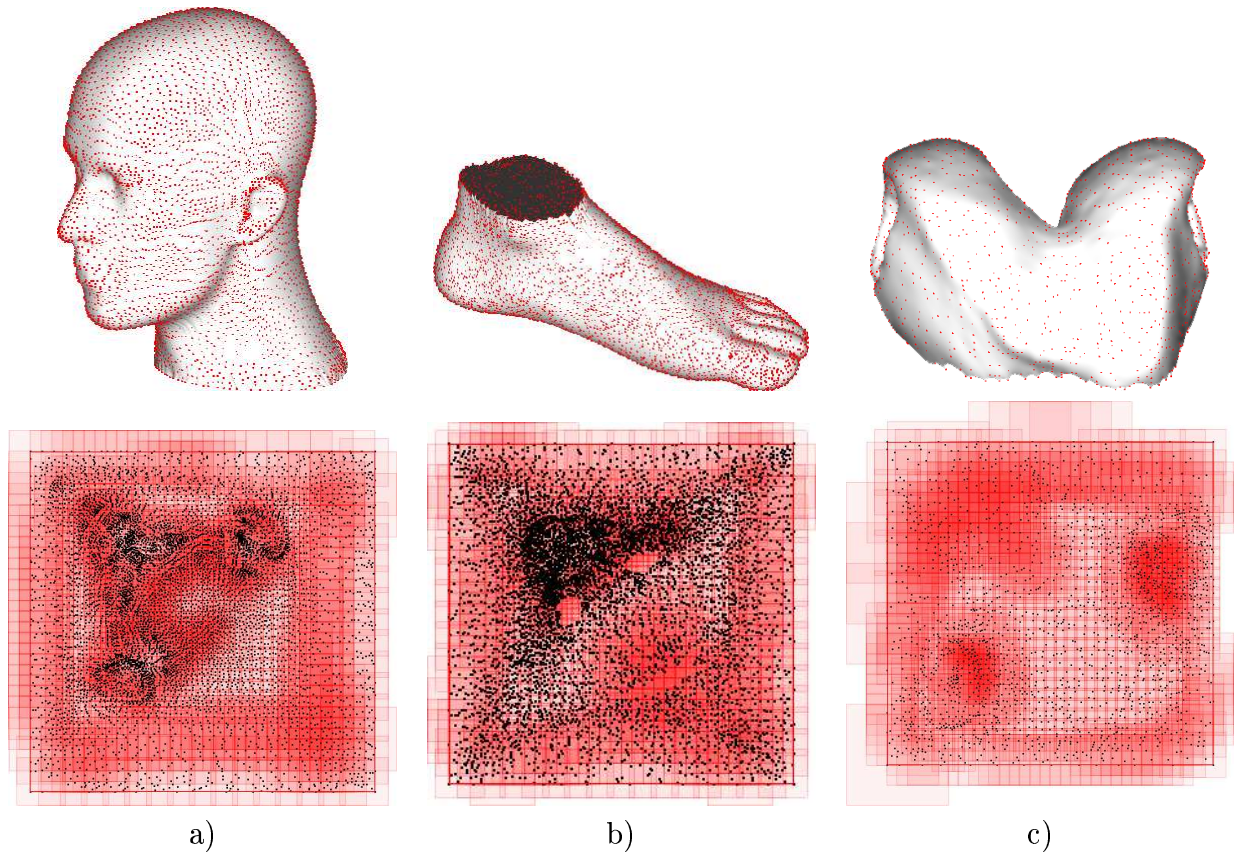


Figure 3.16 – Illustration des résultats de l'interpolation par partition de l'unité. En haut, la surface paramétrique à base radiale, avec les sommets initiaux en rouge. En bas, les sommets initiaux dans l'espace paramétrique, avec les sous-domaines définis par le *quadtree* en rouge.

La figure 3.16 illustre les résultats que j'ai obtenus sur des maillages ouverts de genre topologique équivalent à un disque et le tableau 3.11 présente les temps de calcul. En comparant les résultats des tableaux 3.11 et 3.17, on peut observer que le calcul des coefficients des partitions de l'unité est plus efficace que pour la surface paramétrique à base radiale, bien que les maillages soient composés de plus de sommets.

3.3.3 Interpolation sphérique

La méthode des partitions de l'unité que je viens de décrire dans la section précédente repose sur la subdivision de l'espace paramétrique d'un maillage ouvert, ici le carré unité. Cette contrainte sur la topologie du maillage ne pose pas de problème dans notre cas, puisque nous pouvons nous contenter des surfaces de la tête des os. Cependant, une

Maillage	Nombre de sommets	Temps moyen	Écart type
a)	7 129	39 512	46
b)	5 853	29 280	58
c)	2 173	34 938	111

Tableau 3.17 – Temps de calcul (en msec) des coefficients des partitions de l'unité à base de SPBRs de la figure 3.16 pour 20 calculs. Configuration matérielle utilisée : Athlon XP 3200+, 1 Go de RAM.

majorité des maillages disponibles sont fermés et de genre topologique 0 (semblable à la sphère unité). Je me suis donc intéressé à l'adaptation de la technique des partitions de l'unité au domaine paramétrique de la sphère unité.

Dans un premier temps, j'ai tenté d'adapter notre méthode à moindre coût en conservant la subdivision par *quadtrees*. Pour calculer la distance entre deux sommets, j'ai tenté de les projeter sur la sphère unité, à partir de leur coordonnées paramétriques, puis de calculer la distance géodésique les séparant. Cette méthode introduit des déformations dans les interpolations locales, notamment au niveau des pôles, dues à l'utilisation de la distance géodésique sur une paramétrisation planaire. Nous avons donc mis en place un nouveau schéma de subdivision travaillant sur le domaine de la sphère unité.

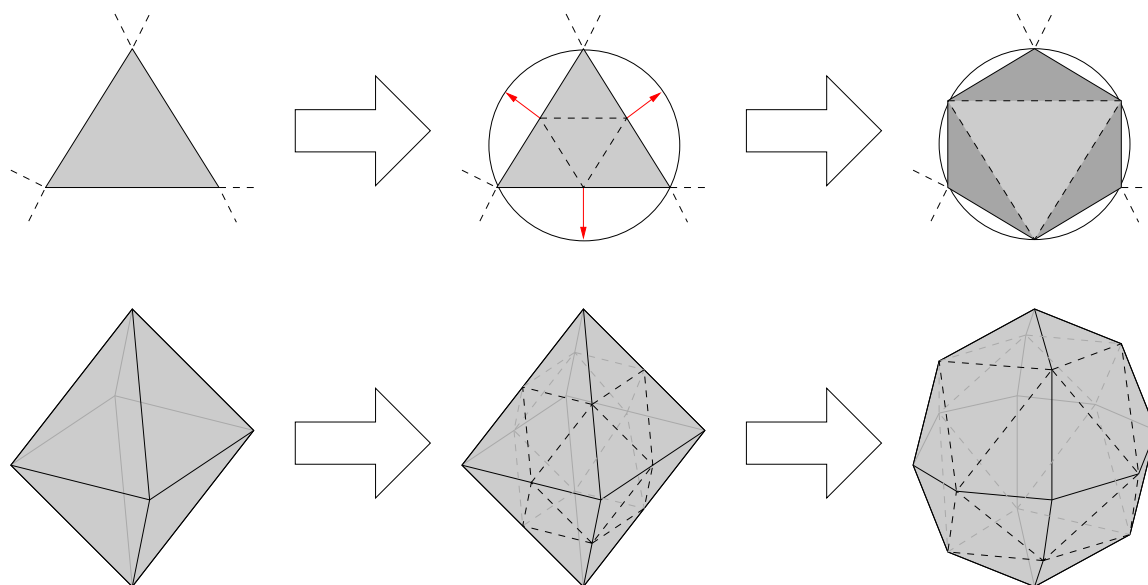


Figure 3.18 – Schéma de subdivision et subdivision d'un octaèdre régulier.

Celui-ci repose sur la subdivision récursive des facettes triangulaires d'un octaèdre régulier inscrit dans la sphère unité. On subdivise les arêtes de chaque triangle équilatéral en leur milieu afin d'obtenir quatre nouveaux triangles équilatéraux. Ces triangles sont

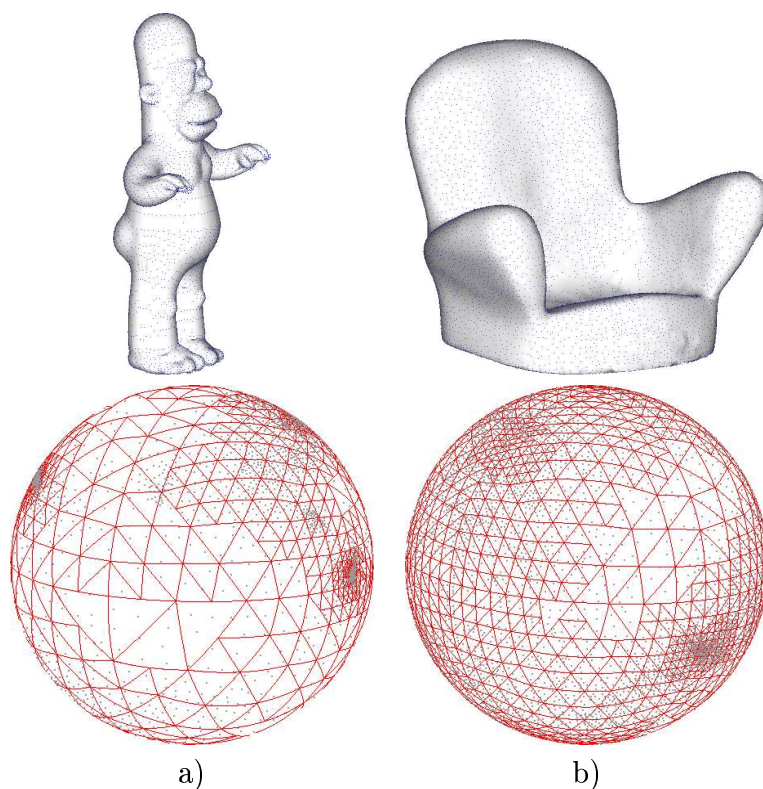


Figure 3.19 – Illustration des résultats de l'interpolation par partition de la sphère unité.

ensuite projetés sur la sphère unité afin d'obtenir des triangles sphériques, nous pouvons alors obtenir pour chacun de ces triangles une section circulaire de la sphère correspondant à son cercle circonscrit. En faisant varier la dimension de cette section grâce au coefficient de chevauchement c , on obtient les sous-domaines souhaités, la figure 3.18 illustre ce schéma de subdivision et la construction de sous-domaines sur la sphère unité.

On obtient donc au final une surface paramétrique interpolante reposant sur la sphère unité. Les déformations introduites dans les interpolations locales ont disparu puisque celles-ci s'effectuent pour chaque sous-domaine directement sur la sphère unité. La paramétrisation sphérique n'intervenant plus dans le processus d'interpolation. La figure 3.19 illustre les résultats obtenus et le tableau 3.20 présente les temps de calcul pour ces illustrations. En observant les tableaux 3.17 et 3.20, on constate que cette méthode offre une efficacité similaire à la méthode reposant sur le carré unité.

3.4 Conclusion

La problématique était d'obtenir une description des surfaces osseuses à partir de données réelles. Ces surfaces doivent être exploitables dans un contexte lagrangien pour le

Maillage	Nombre de sommets	Temps moyen	Écart type
a)	4930	62 054	32
b)	9551	106 198	10

Tableau 3.20 – Temps de calcul (en msec) des coefficients des partition de la sphère unité à base de SPBRs de la figure 3.19 pour 20 calculs. Configuration matérielle utilisée : Athlon XP 3200+, 1 Go de RAM.

suivi continu de contact. Afin d'identifier et de suivre le point de contact sur la surface, nous devons disposer d'une paramétrisation de celle-ci.

Comme nous l'avons vu dans la section 3.2, les surfaces osseuses peuvent être représentées à l'aide de surfaces triangulaires topologiquement équivalentes à des disques ou à des sphères. Nous avons donc présenté plusieurs méthodes pour paramétrer le maillage initial. L'étape suivante consiste à interpoler les données initiales en s'appuyant sur la paramétrisation obtenue. Afin de permettre le suivi continu du contact, cette surface interpolante doit pouvoir disposer d'une continuité C^2 . De plus, il est à noter que la répartition spatiale des données peut-être hétérogène. Nous avons réussi à concilier ces deux contraintes fortes en proposant une utilisation originale des fonctions à base radiale pour la construction de surfaces paramétriques : les surfaces paramétriques à base radiale (section 3.3). Nous améliorons, ensuite ces résultats à l'aide des partitions de l'unité afin d'améliorer les temps de calculs. Pour finir, nous étendons cette méthode à la sphère unité.

Nous avons donc rempli nos objectifs en proposant une méthode d'obtention de surface interpolante à paramétrage global pouvant disposer d'une continuité C^2 et reposant sur une paramétrisation planaire ou sphérique. Ces travaux ont, par ailleurs, fait l'objet de deux communications [JNR05, JNR06].

Cependant il est important de noter que cette surface est interpolante et que celle-ci passe donc par chacun des points du maillage initial. La surface obtenue est donc très sensible au bruit pouvant se trouver dans les données initiales. Le maillage triangulaire doit donc être traité avec soin afin d'offrir une surface la moins perturbée possible. Par ailleurs, la déformation du maillage au cours de la paramétrisation influe fortement sur la variation de l'amplitude des dérivées. Bien que cela ne semble pas a priori être contraignant dans le cas du suivi continu de contact, un soin tout particulier doit être pris pour minimiser cette variation de l'amplitude des dérivées lors de la paramétrisation.

Chapitre 4

Animation dynamique

« “WELL, in *our* country,” said Alice, still panting a little, “you’d generally get to somewhere else – if you ran very fast for a long time, as we’ve been doing.”

“A slow sort of country!” said the Queen. “Now *here*, you see, it takes all the running *you* can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!” »

Lewis CARROLL, *Through the Looking-Glass*. (1872).

Sommaire

4.1	Introduction	65
4.2	Simulation mécanique de l’articulation du genou	66
4.2.1	Modèles dédiés	66
4.2.2	Méthode des éléments finis	67
4.2.3	Les corps rigides poly-articulés de Newton/Euler	68
4.2.4	Formulation lagrangienne	70
4.3	Gestion des contraintes unilatérales	74
4.3.1	Travaux précédents	74
4.3.2	Contraintes simultanées	77
4.4	Le contact glissant	83
4.4.1	Contrainte de contact glissant	84
4.4.2	Variables libres	86
4.5	Conclusion	88

4.1 Introduction

Nous disposons à présent des outils nécessaires à la modélisation géométrique de l’articulation. En effet, à la suite du chapitre précédent, nous pouvons définir une surface

d'interpolation continue à paramétrage global sur le maillage triangulaire obtenu à partir de données réelles. Cette surface continue à paramétrage global nous permet de caractériser de manière unique un point de la surface osseuse et de suivre son évolution au cours de la simulation.

Il reste donc maintenant à mettre en place les outils mécaniques nécessaires à la modélisation et à la simulation de l'articulation du genou. Pour cela, nous allons commencer par un rapide tour d'horizon des possibilités qui nous sont offertes pour l'animation de l'articulation. Et nous justifierons notre choix d'utiliser le formalisme Lagrangien.

Nous aborderons ensuite la gestion de contraintes dans le formalisme Lagrangien en nous penchant plus particulièrement sur la gestion de l'activation/désactivation des contraintes unilatérales multiples. Nous montrerons les lacunes des méthodes existantes et proposerons une solution efficace à ce problème.

Pour finir, nous traiterons le problème du contact glissant nécessaire à la simulation réaliste de l'articulation. Dans un premier temps, nous présenterons la méthode qui repose sur l'utilisation de nouveaux degrés de liberté virtuels à évolution libre, appelés « variables libres ». Et dans un second temps, nous définirons ces variables libres et la manière de les introduire dans le système mécanique.

4.2 Simulation mécanique de l'articulation du genou

Dans la littérature informatique et biomécanique, deux types d'approches sont couramment employées pour la simulation d'articulations :

- La première, la méthode des éléments finis, s'est développée avec la puissance de calcul des ordinateurs.
- La seconde repose sur la méthode des solides rigides poly-articulés exploitant les lois fondamentales du mouvement de Newton/Euler. Elle est la plus ancienne et elle est encore largement utilisée car elle permet des calculs d'évolution très rapides.

Enfin on peut noter qu'une troisième approche, plus générique mais encore plutôt marginale, repose sur le formalisme de Lagrange.

4.2.1 Modèles dédiés

Ces dernières décennies, avec la puissance croissante des ordinateurs, de nombreux modèles d'animation dédiés à certains aspects du corps humain ont été développés. La marche notamment a fait l'objet de recherches particulières comme [BC89, Wie94, BC96,

MMKA04, BDM06]. Ils modélisent les articulations à l'aide d'un formalisme mécanique, Lagrange pour Bruderlin et al. [BC96] et Newton-Euler pour les autres, pour en extraire des courbes paramétriques du mouvement pour chaque articulation. On peut alors obtenir des animations cohérentes et efficaces de la marche. Ces modèles sont malheureusement peu personnalisables puisque les courbes obtenues reposent sur des données anthropométriques génériques. De plus, l'articulation du genou est représentée par une liaison mécanique de type pivot, or nous avons vu dans le chapitre 2 que cette solution n'est pas suffisante dans le cadre d'une simulation réaliste de l'articulation du genou.

Il est donc nécessaire de revenir à des méthodes de simulation plus générales afin d'obtenir des courbes de mouvement exactes, avant de nous intéresser à des simulateurs dédiés. On peut cependant noter l'intérêt d'obtenir une extraction réaliste des courbes d'animation d'une articulation, notamment du genou, pour la réexploitation en animation traditionnelle ou cinématique inverse.

4.2.2 Méthode des éléments finis

La méthode des éléments finis est une méthode de calcul numérique de l'évolution d'un modèle géométrique. Elle repose sur un échantillonnage 1D, 2D ou 3D du modèle géométrique, les paramètres relatifs aux matériaux sont ensuite affectés à chacun des sous-domaines du maillage. On déduit ensuite, pour chaque étape de l'animation, le déplacement des nœuds (sommets) d'après les déformations des mailles adjacentes, grâce aux lois de la mécanique des milieux continus.

La qualité du résultat est donc intrinsèquement liée à la qualité du maillage initial. Or le calcul s'effectuant en chaque sommet du maillage voit sa complexité augmenter proportionnellement à la qualité du maillage. Cette méthode présente donc d'une part le désavantage d'être coûteuse en temps de calcul et d'autre part de voir ce temps de calcul directement dépendant du résultat souhaité. Malgré tout, cette méthode a l'avantage de reposer sur un mécanisme générique bien défini.

Dernièrement, certains travaux ont été effectués pour accélérer les temps de calcul, mais ceux-ci reposent généralement sur une linéarisation et un précalcul des efforts dus aux matériaux [Cot97, Pic01], contraignant ainsi le système à de petits mouvements.

En se basant sur la méthode des éléments finis, McCarthy et al. [MHB⁺03] ont modélisé un genou virtuel à 6 DdL et en ont animé les différents éléments de manière réaliste. Ils ont ainsi pu comparer leurs résultats à la littérature, trouvant des résultats concordants, prouvant ainsi, s'il en était besoin, la robustesse de la méthode.

Le calcul de l'évolution s'effectuant pour chacun des sous-domaines, on obtient donc au final un formalisme naturel pour l'évolution de modèle déformable. Liesenfeld et al. [LS96] mixent donc l'utilisation de solides poly-articulés pour l'animation du squelette et d'éléments finis pour animer les corps déformables de l'articulation comme les tendons. Cependant, leur modèle repose sur des liaisons mécaniques classiques, qui ne suffisent pas à modéliser la complexité de l'articulation.

Plus récemment, Halloran et al. [HPR05] ont utilisé la méthode des éléments finis pour simuler l'utilisation d'une prothèse après remplacement total du genou. Ils reproduisent les articulations tibio-fémorale et patello-fémorale avec les tendons du quadriceps et des ligaments associés. Ce travail permet d'obtenir une étude de la répartition des efforts sur les zones de contact pour un modèle déformable de l'articulation.

Cette méthode est donc largement répandue et s'adapte parfaitement aux jeux de données réelles [MWH⁺02] qu'il est désormais possible d'obtenir. Cependant, cette méthode est coûteuse en temps de calcul et en mémoire si on ne souhaite pas se restreindre à de petits déplacements. De plus, ce formalisme est peu adapté à l'animation de corps rigides et il est donc très difficile d'en extraire le mouvement global généré. Ce formalisme bien que riche pour la simulation nous semble peu adapté et nous avons donc orienté nos travaux vers l'animation de corps rigides poly-articulés.

En effet, nous cherchons à obtenir un DdL formel complexe du mouvement de flexion/extension du genou. Or la méthode des éléments finis produit trop d'informations puisqu'elle simule la déformation de tous les éléments du modèle. Il devient donc très difficile d'en extraire le DdL de flexion/extension que nous cherchons.

4.2.3 Les corps rigides poly-articulés de Newton/Euler

La méthode des corps rigides poly-articulés dans le formalisme de Newton/Euler permet de simuler l'action d'un squelette décrit à l'aide d'une hiérarchie de repères. La position et l'orientation de chaque repère étant exprimées dans celui de son père. À l'aide des lois fondamentales du mouvement de Newton/Euler, on peut alors obtenir pour le $i^{\text{ème}}$ solide du modèle hiérarchique les accélérations comme le montre l'équation 4.1 et 4.2.

$$m_i \ddot{\mathbf{g}}_i = \sum_1^{n_i} \mathbf{f}_{ij} \quad (4.1)$$

$$\mathbf{R}_i \mathbf{I}_i \mathbf{R}_i^T \dot{\boldsymbol{\omega}}_i = \sum_1^{n_i} \overrightarrow{\mathbf{gp}}_j \times \mathbf{f}_{ij} \quad (4.2)$$

Avec $\mathbf{g}_i$ et $\ddot{\mathbf{g}}_i$ respectivement la position et l'accélération du centre de gravité du $i^{\text{ème}}$ solide, $\dot{\omega}_i$ son accélération de rotation. De plus m_i est la masse du $i^{\text{ème}}$ solide et $\mathbf{f}_{ij}$ pour $j \in \{1, \dots, n_i\}$ sont les forces externes qui s'y appliquent aux points $\mathbf{p}_j$. Pour finir $\mathbf{R}_i$ et $\mathbf{I}_i$ sont deux matrices : la première exprime la transformation pour passer du repère local au repère global et la seconde représente la matrice d'inertie exprimée dans le repère local.

Les valeurs m_i et $\mathbf{I}_i$ doivent être constantes durant la simulation, il ne reste donc qu'à initialiser au début de l'animation puis à mettre à jour au cours de la simulation les valeurs restantes. La position du centre de gravité $\mathbf{g}_i$ et la matrice de transformation $\mathbf{R}_i$ du $i^{\text{ème}}$ solide sont mises à jour par un parcours récursif de l'arborescence du modèle.

Dans le formalisme de Newton/Euler, la gestion des efforts de maintien des liaisons n'est pas naturelle. La mise à jour de ces efforts nécessite un premier parcours de l'arborescence, afin d'évaluer la déviation et les efforts nécessaires pour y répondre. On peut alors faire évoluer les positions et orientations des solides par un second parcours de l'arbre de repères.

Une fois les accélérations obtenues, il ne reste plus qu'à employer une méthode d'intégration numérique telle que la méthode explicite d'Euler pour obtenir les position et orientation du repère à l'étape suivante. Cette approche permet donc de créer et résoudre les équations du mouvement, de façon itérative, dans des délais efficaces, proportionnels au nombre d'objets du système. Cependant, les boucles cinématiques et les contraintes ne sont pas naturellement prises en compte par ce formalisme. Il est donc nécessaire d'ajouter des étapes spécifiques au processus pour les intégrer au moteur d'animation [SZ90].

Wilhelms [WB85] et Armstrong-Green [AG85] génèrent des équations du mouvement sur ce principe en travaillant sur un arbre de solides articulés à l'aide de 3 DdL en rotation pour chaque lien entre un père et un fils, seule la racine de l'arbre possède 6 DdL (3 en rotation et 3 en translation). Ils lient ensuite l'accélération du centre de masse d'un fils $\mathbf{a}_F$ par rapport aux accélérations de son père par une fonction. L'algorithme fonctionne alors en deux passes :

- un parcours récursif de l'arborescence des fils vers leur père permettant l'initialisation des paramètres dépendant des efforts extérieurs, des éléments inertiels et des couples de réaction entre fils et père.
- un parcours récursif de l'arborescence du père vers ses fils permettant le calcul des accélérations.

Ko [Ko95] utilise ce formalisme pour simuler la marche humaine. Il développe les équations spécifiquement pour le mouvement des jambes avec 97 degrés de liberté. Il intègre également un module de gestion des boucles qui permet de prendre en compte la phase de mouvement où les deux pieds touchent le sol.

Fang et al. [FP03] introduisent une étape de gestion des contraintes par pénalité pour animer un humain virtuel. Cette méthode offre un outil efficace permettant de générer des mouvements physiquement cohérents à partir d'étapes clés définies par l'animateur. Cependant, comme dans la plupart des méthodes précédentes, le genou n'y est représenté que par une liaison de type pivot afin de simplifier la modélisation.

En effet, l'intégration d'autres DdL impose de définir des contraintes afin de restreindre le champ d'activité. Or, d'une part, comme nous l'avons vu dans la section 2, l'expression de ces contraintes n'est pas encore clairement définie, et d'autre part, la gestion des contraintes n'est pas traitée efficacement dans ce formalisme. En effet, le formalisme newtonien ne permet d'exprimer des contraintes que sur les six degrés de liberté de chaque repère. De plus, à l'inverse des éléments finis, cette méthode ne permet pas la simulation de modèles déformables. Il ne serait donc pas possible d'intégrer directement des éléments déformables tels que les tendons ou les muscles au modèle. Une solution élégante est d'employer un formalisme plus général tel que celui de Lagrange.

4.2.4 Formulation lagrangienne

La mécanique lagrangienne, aussi qualifiée de mécanique analytique, peut être vue comme une extension des équations de Newton-Euler à des systèmes particuliers plus variés tels que les corps déformables. Dans ce formalisme, la configuration d'un système mécanique est décrite par un ensemble de paramètres réels : le vecteur d'état $\mathbf{x}$ des *coordonnées généralisées* qui identifie les degrés de liberté réels (DdL), éventuellement complexes, du système mécanique simulé. Cette approche permet de générer une plus grande variété d'animations et ce, de manière efficace en utilisant un paramétrage adapté. Avec ce formalisme, les DdL ne sont plus limités aux rotations et translations. Par exemple, un paramètre $u \in [0, 1]$ définissant la position d'un point sur une courbe paramétrée 3D peut tout à fait être considéré comme une coordonnée généralisée.

Associé aux fonctions de positionnement Φ et de vitesse $\hat{\Phi}$, un vecteur d'état permet de calculer la position 3D de toutes les particules $\omega \in \Omega$ constitutives du système dans une configuration donnée. Si ρ désigne la densité de masse au sein du système, l'énergie cinétique peut être définie de la manière suivante :

$$K = \frac{1}{2} \int_{\Omega} \rho(\omega) \hat{\Phi}(\mathbf{x}, \dot{\mathbf{x}}, \omega)^2 d\omega \quad (4.3)$$

On désigne $\boldsymbol{\eta}$ le vecteur des configurations accessibles et $\dot{\boldsymbol{\eta}}$ le taux de variation de

celui-ci. L'équation 4.3 peut être redéfinie par :

$$K = \frac{1}{2} \dot{\mathbf{x}}^T \mathbf{M}(\mathbf{x}) \dot{\mathbf{x}} \quad (4.4)$$

avec $\mathbf{M} = \int_{\Omega} \rho(\omega) \frac{\partial \Phi}{\partial \boldsymbol{\eta}}(\mathbf{x}, \omega)^T \frac{\partial \Phi}{\partial \boldsymbol{\eta}}(\mathbf{x}, \omega) d\omega$

Dans l'équation 4.4, les éléments structurels du système sont représentés par la *matrice des masses généralisées* $\mathbf{M}$. Le mouvement est représenté par le vecteur d'état $\mathbf{x}$ et sa dérivée temporelle $\dot{\mathbf{x}}$. Or, comme $\boldsymbol{\eta}$ et $\dot{\boldsymbol{\eta}}$ désignent respectivement le vecteur des configurations accessibles et le taux de variation de celui-ci, l'énergie cinétique peut être vue comme une fonction de $\boldsymbol{\eta}$ et $\dot{\boldsymbol{\eta}}$.

$$\kappa(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = \frac{1}{2} \dot{\boldsymbol{\eta}}^T \mathbf{M}(\boldsymbol{\eta}) \dot{\boldsymbol{\eta}} \quad (4.5)$$

4.2.4.1 Mouvement non-contraint

L'évolution d'un système libre gouvernée seulement par un ensemble de forces externes repose sur l'utilisation des équations du mouvement de Lagrange (équation 4.6).

$$\frac{d}{dt} \frac{\partial \kappa}{\partial \dot{\eta}_i}(\mathbf{x}, \dot{\mathbf{x}}) - \frac{\partial \kappa}{\partial \eta_i}(\mathbf{x}, \dot{\mathbf{x}}) = \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}), \quad i \in \{1, \dots, n\} \quad (4.6)$$

Le vecteur $\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}})$ correspond à la somme des puissances virtuelles des forces externes dans les mouvements virtuels d'entraînement par chaque DdL, pour la configuration définie par le vecteur d'état $\mathbf{x}$ et sa dérivée temporelle $\dot{\mathbf{x}}$.

Après développement des équations 4.6, on obtient le système d'équations différentielles ordinaires exprimé sous la forme matricielle ci-après :

$$\mathbf{M} \ddot{\mathbf{x}} = \mathbf{f} \quad (4.7)$$

Afin d'alléger les notations, nous noterons $\mathbf{M}$ la *matrice des masses généralisées* $\mathbf{M}(\mathbf{x})$ et $\mathbf{f}$ le vecteur des sommes de puissances virtuelles des forces externes dans les mouvements virtuels d'entraînement par chaque DdL ($\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}})$). Le vecteur $\ddot{\mathbf{x}}$ est la dérivée seconde du vecteur d'état ($\mathbf{x}$) par rapport au temps.

4.2.4.2 Mouvement contraint

Il existe deux types de contraintes bilatérales, les contraintes holonomes, qui portent sur les coordonnées généralisées ($\mathbf{x}$), et les contraintes cinématiques, qui portent linéairement sur les vitesses généralisées ($\dot{\mathbf{x}}$). Dans le cadre de cette thèse, je ne m'intéresse qu'aux contraintes holonomes. Par convention, une contrainte holonome est toujours définie comme une équation 4.8 où $\mathcal{E}$ est l'ensemble des indices de toutes les contraintes

bilatérales.

$$g_k(\mathbf{x}) = 0 \quad \forall k \in \mathcal{E} \quad (4.8)$$

Les contraintes holonomes restreignent le champ des configurations à l'intersection des $\text{card}(\mathcal{E})$ hyper-surfaces définies par l'équation 4.8. Il existe de nombreuses méthodes de gestion des contraintes, regroupées dans un état de l'art de Baraff [Bar93]. Mais il en existe trois principales pour intégrer les contraintes dans l'équation 4.7

La méthode par projection consiste à modifier le vecteur d'état $\mathbf{x}$ et sa dérivée première $\dot{\mathbf{x}}$ de manière à satisfaire la contrainte. Cette modification peut être effectuée de manière itérative comme pour la méthode de Newton-Raphson [VF02]. Bien que cette approche soit très simple et qu'elle semble assurer la réalisation instantanée des contraintes, cette méthode n'est pas assez robuste : en effet, elle ne peut pas garantir une convergence du processus en cas d'activation simultanée de contraintes.

La méthode par pénalité intègre au système de nouvelles forces externes agissant comme des ressorts virtuels dans le but de minimiser le carré de l'équation de la contrainte, considéré comme une fonction d'énergie positive. Le principal avantage de cette méthode est qu'elle est compatible avec n'importe quel moteur d'animation dynamique puisqu'elle ne repose que sur des forces externes. Malheureusement, cette méthode ne permet qu'une réalisation approximative des contraintes, autorisant l'interpénétration de deux objets par exemple. Dans le but de réduire cette interpénétration, la raideur du ressort virtuel doit être augmentée significativement rendant le système numérique instable.

La méthode de Lagrange consiste à calculer les quantités exactes des efforts, représentés par les *multiplicateurs de Lagrange*, nécessaires à l'accomplissement des contraintes. Cette méthode assure que chaque contrainte du système est satisfaite de manière exacte. Comme l'utilisation des multiplicateurs de Lagrange introduit de nouvelles inconnues au système, l'équation 4.7 doit être complétée avec un nouveau jeu d'équations augmentant ainsi la dimension du système initial à résoudre. Cependant nous considérons cette méthode comme la plus apte à gérer efficacement les contraintes géométriques.

Pour toutes les raisons citées ci-dessus, nous avons choisi la méthode de Lagrange pour gérer nos contraintes géométriques. Conformément au principe des travaux virtuels, chaque contrainte g_k ajoute une nouvelle force de maintien dont les puissances virtuelles forment un vecteur perpendiculaire au plan tangent à l'hypersurface $g_k(\mathbf{x}) = 0$. Le multiplicateur de Lagrange λ_k correspond au coefficient de proportionnalité entre le vecteur des puissances virtuelles de l'effort de maintien et la normale à l'hyperplan. Avec ces nouvelles forces, l'équation 4.7 est modifiée comme suit :

$$\mathbf{M} \ddot{\mathbf{x}} = \mathbf{f} + \sum_{k \in \mathcal{E}} \lambda_k \frac{\partial g_k}{\partial \mathbf{x}} \quad (4.9)$$

Nous ajoutons ensuite de nouvelles équations à notre système en calculant la dérivée seconde de l'équation 4.8, afin d'obtenir l'équation 4.10.

$$\sum_{i=1}^n \frac{\partial g_k}{\partial x_i} \ddot{x}_i = - \sum_{i,j=1}^n \frac{\partial^2 g_k}{\partial x_i \partial x_j} \dot{x}_i \dot{x}_j \quad \forall k \in \mathcal{E} \quad (4.10)$$

Dans le but de corriger la dérive numérique due aux erreurs d'arrondi, Baumgarte propose dans son article [Bau72] un schéma de stabilisation des contraintes illustré par l'équation 4.11. Le paramètre τ^{-1} peut être vu comme la vitesse de réalisation de la contrainte.

$$\sum_{i=1}^n \frac{\partial g_k}{\partial x_i} \ddot{x}_i = - \sum_{i,j=1}^n \frac{\partial^2 g_k}{\partial x_i \partial x_j} \dot{x}_i \dot{x}_j - \gamma_1 \sum_{i=1}^n \frac{\partial g_k}{\partial x_i} \dot{x}_i - \gamma_2 g_k \quad (4.11)$$

Baumgarte propose $\frac{2}{\tau}$ et $\frac{2}{\tau_2}$ comme valeurs respectives de γ_1 et γ_2 . Mais le choix de ces valeurs est contesté par Remion [Rem00] qui démontre que les coefficients γ_1 et γ_2 dépendent de la méthode d'intégration numérique utilisée pour la résolution des équations du mouvement.

En mélangeant les équations 4.7 et 4.11, on obtient un système linéaire où la dérivée seconde du vecteur d'état $\mathbf{x}$ et le vecteur contenant les multiplicateurs de Lagrange Λ sont les inconnues.

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}^T \\ -\mathbf{J} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \Lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ -\mathbf{d} \end{bmatrix} \quad (4.12)$$

$\mathbf{J}$ est la matrice jacobienne de l'ensemble des contraintes géométriques et $\mathbf{d}$ correspond au terme de droite de l'équation 4.11.

Wieland [Wie94] et Troy [Tro98] utilisent ce formalisme pour simuler le processus de la marche, mais encore une fois la liaison du genou est représentée par une liaison pivot. Il n'existe pas de simulation du genou reposant sur un autre modèle utilisant le formalisme lagrangien probablement parce que la définition des degrés de liberté et des contraintes n'est pas triviale, ce qui nous conforte dans l'intérêt de notre démarche.

Cependant, les contacts osseux n'étant pas forcément continus, la gestion des contacts osseux à l'aide de contraintes suppose de pouvoir traiter efficacement les activations et désactivations des contraintes, notamment lorsque celles-ci se produisent simultanément.

4.3 Gestion des contraintes unilatérales

Le problème d'activation et de désactivation des contraintes est un problème auquel on est confronté dès que l'on utilise des contraintes unilatérales. Par convention, une contrainte unilatérale est toujours définie comme une équation 4.13 où $\mathcal{F}$ est l'ensemble des indices de toutes les contraintes unilatérales.

$$g_k(\mathbf{x}) \geq 0 \quad \forall k \in \mathcal{F} \quad (4.13)$$

Pour un vecteur d'état donné $\mathbf{x}$, nous rappelons les définitions suivantes :

- la contrainte est dite *violée* par $\mathbf{x}$ quand $g_k(\mathbf{x}) < 0$. C'est à dire que le vecteur d'état $\mathbf{x}$ correspond à une configuration non autorisée.
- la contrainte est dite *satisfaite* par $\mathbf{x}$ quand $g_k(\mathbf{x}) \geq 0$.
- la contrainte est dite *active* quand $g_k(\mathbf{x}) = 0$. Dans ce cas, le vecteur d'état $\mathbf{x}$ appartient à la frontière du demi-espace défini par la contrainte unilatérale g_k .

La gestion des contraintes unilatérales est plus difficile que la gestion des contraintes bilatérales. Puisque chacune ne doit être gérée que si elle est violée ou active. En fait, l'algorithme est un peu plus compliqué comme nous l'expliquons dans la sous-section suivante.

C'est pourquoi nous définissons deux sous-domaines de $\mathcal{F}$: $\mathcal{F}^+$ est l'ensemble des indices de toutes les contraintes unilatérales gérées et $\mathcal{F}^-$ est l'ensemble des indices des contraintes unilatérales ignorées. Dans ces conditions, on a $\mathcal{F} = \mathcal{F}^- \cup \mathcal{F}^+$. Enfin la matrice jacobienne des contraintes $\mathbf{J}$ de l'équation 4.12 est construite à partir de l'ensemble des contraintes g_k où $k \in \mathcal{E} \cup \mathcal{F}^+$.

4.3.1 Travaux précédents

Je vais expliquer ici une partie des travaux développés par Remion dans son mémoire d'habilitation à diriger des recherches [Rem00]. Ces travaux n'ayant pas encore été implantés, j'ai pu les étudier et les implanter au sein de MADy pour en valider le fonctionnement.

Dans la communauté de l'informatique graphique, la principale méthode publiée sur la gestion des contraintes unilatérales à partir de multiplicateurs de Lagrange, connue sous le nom de “*Generalized Dynamic Constraints*”, est proposé par Platt dans [Pla92a]. Dans cet article, il décrit comment utiliser les multiplicateurs de Lagrange pour assembler et simuler les collisions entre des modèles numériques. Cette méthode est une extension des travaux de Barzel et Barr [BB88] qui spécifient comment les contraintes doivent être satisfaites. De plus, Platt propose une méthode pour mettre à jour $\mathcal{F}^+$ (l'ensemble des contraintes unilatérales gérées) au cours de l'animation. Cet algorithme peut-être rapproché des méthodes classiques relatives aux *ensembles actifs* [Bjö96, NW00].

Nous ne nous intéressons pas au processus de détection de la collision, qui est un problème à part entière. Nous sommes conscients que cette problématique difficile peut être résolue de nombreuses manières et nous encourageons le lecteur à se référer à l'état de l'art de Teschner et al. [TKZ⁺05]. Lors de la phase de détection de collisions, nous supposons que le moteur dynamique peut revenir en arrière dans le temps jusqu'au moment précis où l'activation de la contrainte est détectée. Cette assertion peut nécessiter une augmentation significative du coût de calcul, restreignant ainsi cette méthode à la production d'animations en temps différé selon la complexité de la scène simulée. Malgré cela, il demeure possible que plusieurs contraintes soient actives simultanément.

Au début de l'animation, l'algorithme de Platt considère que l'ensemble $\mathcal{F}^+$ est rempli avec l'ensemble des contraintes actives.

ALGO. 1 – Algorithme de Platt

```

1 Résoudre l'équation 4.12 pour obtenir  $\ddot{\mathbf{x}}$  et  $\Lambda$ 
2 Mettre à jour  $\mathbf{x}$  et  $\dot{\mathbf{x}}$  (intégration numérique)
3 pour chaque  $k \in \mathcal{F}$  faire
4   si  $k \in \mathcal{F}^+$  alors
5     si  $\lambda_k \leq 0$  alors
6        $k$  est déplacé dans  $\mathcal{F}^-$ 
7   sinon
8     si  $g_k(\mathbf{x}) < 0$  alors
9        $k$  est déplacé dans  $\mathcal{F}^+$ 

```

Pour chaque étape de l'animation, en concordance avec l'algorithme de Platt 1, nous résolvons l'équation 4.12 pour mettre à jour le vecteur d'état dans le but d'obtenir les nouvelles positions et vitesses à la fin de l'étape. Nous pouvons alors vérifier l'état des contraintes unilatérales. Si une contrainte g_k est active, elle reste gérée jusqu'à ce que son multiplicateur de Lagrange soit négatif ou nul (c'est à dire que le multiplicateur de Lagrange définit une force qui empêche la désactivation de la contrainte). En fonction des nouvelles valeurs du vecteur d'état $\mathbf{x}$, si la contrainte g_k précédemment inactive est maintenant violée ($g_k(\mathbf{x}) < 0$) alors l'indice de la contrainte doit être ajouté à $\mathcal{F}^+$ pour empêcher le système d'entrer dans cette configuration.

Bien que cet algorithme semble offrir une solution appropriée à la gestion des contraintes unilatérales, quelques problèmes persistent. Nous avons mis en place une scène simple, décrite par la figure 4.1 pour illustrer les lacunes de la méthode de Platt. Une particule de masse m est contrainte à glisser sur un plan 2D. Elle part d'un coin en angle aigu modélisé à l'aide de deux contraintes unilatérales linéaires $g_1(\mathbf{x}) \geq 0$ et $g_2(\mathbf{x}) \geq 0$ où $g_1(\mathbf{x}) = x - y$ and $g_2(\mathbf{x}) = y$. Pour finir, cette particule n'est affectée que par une seule force externe

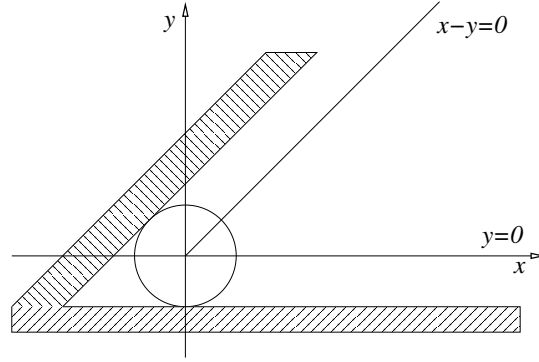


Figure 4.1 – Un exemple simple avec deux contraintes actives simultanément.

$\mathbf{f} = (2, -1)$. Dans cet exemple, le vecteur d'état $\mathbf{x}$ est composé des coordonnées 2D (x, y) de la particule. Selon l'équation 4.7, la matrice des masses généralisées pour ce système est définie par :

$$\mathbf{M} = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \quad (4.14)$$

Comme les contraintes géométriques $g_1(\mathbf{x})$ et $g_2(\mathbf{x})$ sont linéaires, leurs dérivées première et seconde ne produisent aucun des termes définis dans l'équation 4.11 :

$$\mathbf{d} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.15)$$

En tenant compte des valeurs initiales du vecteur d'état $\mathbf{x} = (0, 0)$, les deux contraintes $g_1(\mathbf{x})$ et $g_2(\mathbf{x})$ sont actives. Leur indice est donc ajouté dans $\mathcal{F}^+$ et la matrice jacobienne des contraintes $\mathbf{J}$ est définie par :

$$\mathbf{J} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \quad (4.16)$$

à partir des équations 4.12, 4.14, 4.15, et 4.16, on obtient un système linéaire dont les inconnues sont les dérivées secondes du vecteur d'état $\mathbf{x}$ et les deux multiplicateurs de Lagrange λ_1 et λ_2 associés à g_1 et g_2 :

$$\begin{bmatrix} m & 0 & -1 & 0 \\ 0 & m & 1 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 0 \\ 0 \end{bmatrix} \quad (4.17)$$

Les solutions sont $\ddot{\mathbf{x}} = (0, 0)$ et $\Lambda = (-2, -1)$. La première erreur, qui se présente, est donc la retenue abusive de la particule dans le coin. En effet, celle-ci ne bouge pas au cours de cette étape car $\ddot{x}$ et $\ddot{y}$ sont nuls. De plus, comme λ_1 et λ_2 sont tous les deux négatifs,

les contraintes associées sont déplacées dans $\mathcal{F}^-$. Ce qui signifie qu'à la prochaine étape, le système sera libre de toute contrainte. Comme la force reste constante, la prochaine valeur de $\ddot{\mathbf{x}}$ sera égale à $(2m^{-1}, -m^{-1})$. Ce qui amènera la particule à une position non autorisée, sous la ligne $y = 0$. Ces calculs sont illustrés dans la figure 4.2.

L'amplitude de la violation de la contrainte $g_2(\mathbf{x}) = y$ dépend essentiellement du rapport entre la masse m de la particule et l'intensité de la force externe $\mathbf{f}$, ainsi que du pas de temps.

4.3.2 Contraintes simultanées

4.3.2.1 Une première approche

Le problème de la méthode de Platt est qu'elle peut retenir certaines contraintes dans $\mathcal{F}^+$ qui devraient normalement être ignorées. En fait, la condition $g_k(\mathbf{x}) < 0$ utilisée pour remplir $\mathcal{F}^+$ avec les contraintes unilatérales n'est pas bien adaptée et une approche alternative est proposée. Une solution pourrait être de remplacer la condition $g_k(\mathbf{x}) < 0$ par une condition reposant sur la tendance à la violation exprimée par $J_k \ddot{\mathbf{x}} < d_k$. Une contrainte active qui ne remplit pas la tendance à la violation peut être satisfaite mais inactive à la prochaine étape de l'animation et n'a donc pas à être gérée.

Au début de l'animation, la solution de l'équation 4.7 doit être calculée pour obtenir $\ddot{\mathbf{x}}$ et ainsi remplir l'ensemble $\mathcal{F}^+$ avec les contraintes actives qui remplissent la condition de tendance à la violation. Il est évident que par ce procédé nous gérons, dans notre système, moins de contraintes que Platt puisque notre critère est plus strict.

Nous vérifions rapidement que cet algorithme fournit une solution appropriée à notre exemple illustré par la figure 4.1. D'après l'équation 4.7, $\ddot{\mathbf{x}} = (2m^{-1}, -m^{-1})$. Les deux

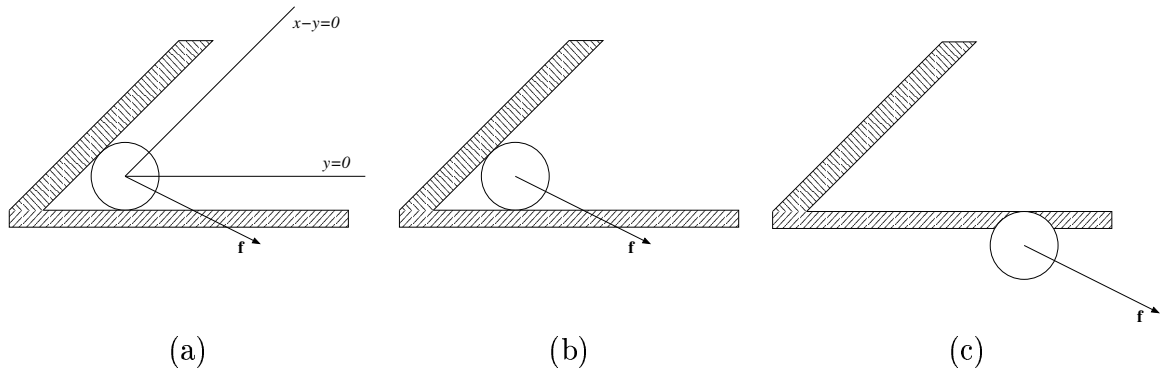


Figure 4.2 – (a) Comme les deux contraintes sont actives, elle sont gérées par l'algorithme de Platt (b) Les multiplicateurs de Lagrange sont négatifs, les contraintes sont donc ignorées (c) La nouvelle accélération non contrainte mène à une position non autorisée

ALGO. 2 – Algorithme de Platt amélioré

```

1 Résoudre l'équation 4.12 pour obtenir  $\ddot{\mathbf{x}}$  et  $\Lambda$ 
2 Mettre à jour  $\mathbf{x}$  et  $\dot{\mathbf{x}}$  (intégration numérique)
3 pour chaque  $k \in \mathcal{F}$  faire
4   si  $k \in \mathcal{F}^+$  alors
5     si  $\lambda_k \leq 0$  alors
6        $k$  est déplacé dans  $\mathcal{F}^-$ 
7   sinon
8     si  $J_k \ddot{\mathbf{x}} < d_k$  alors
9        $k$  est déplacé dans  $\mathcal{F}^+$ 

```

contraintes g_1 et g_2 sont actives puisque $\mathbf{x} = (0, 0)$ mais seule g_2 remplit la condition de tendance à la violation, comme l'illustre l'équation 4.18.

$$\begin{aligned} J_1 \ddot{\mathbf{x}} &= 3m^{-1} \Rightarrow 1 \in \mathcal{F}^- \\ J_2 \ddot{\mathbf{x}} &= -m^{-1} \Rightarrow 2 \in \mathcal{F}^+ \end{aligned} \quad (4.18)$$

Dans cet exemple, l'équation 4.12 devient :

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} \quad (4.19)$$

Les solutions du système linéaire 4.19 sont $\ddot{\mathbf{x}} = (2m^{-1}, 0)$ et $\lambda_2 = 1$. Au final, la particule va glisser le long de l'axe des x sans passer sous la ligne $y = 0$ car la contrainte g_1 qui n'est pas gérée n'introduit pas de réponse artificielle dans le système.

Cette version de l'algorithme semble prendre en compte les multiples contraintes unilatérales de manière appropriée, mais nous pouvons illustrer une erreur de cette méthode avec le même exemple représenté par la figure 4.1 en changeant la force externe $\mathbf{f} = (-1, -2)$.

Au début, comme $\mathbf{x} = (0, 0)$, les contraintes g_1 et g_2 sont actives. À partir de l'équation 4.7, on obtient $\ddot{\mathbf{x}} = (-m^{-1}, -2m^{-1})$, et avec l'équation 4.20, seule la contrainte g_2 est gérée.

$$\begin{aligned} J_1 \ddot{\mathbf{x}} &= m^{-1} \Rightarrow 1 \in \mathcal{F}^- \\ J_2 \ddot{\mathbf{x}} &= -2m^{-1} \Rightarrow 2 \in \mathcal{F}^+ \end{aligned} \quad (4.20)$$

D'après l'équation 4.20, on construit le système linéaire 4.21

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ 0 \end{bmatrix} \quad (4.21)$$

Les solutions sont $\ddot{\mathbf{x}} = (-m^{-1}, 0)$ et $\lambda_2 = 2$. Après la mise à jour de $\mathbf{x}$ et $\dot{\mathbf{x}}$, la particule glisse le long de la ligne définie par la contrainte g_1 pour atteindre une position illégale. Ceci est dû au fait que le multiplicateur de Lagrange λ_2 pousse le système vers un état non autorisé par la contrainte g_1 qui n'est pas gérée dans l'équation 4.12 puisqu'elle ne satisfait pas le critère de tendance à la violation. Ces calculs sont illustrés par la figure 4.3.

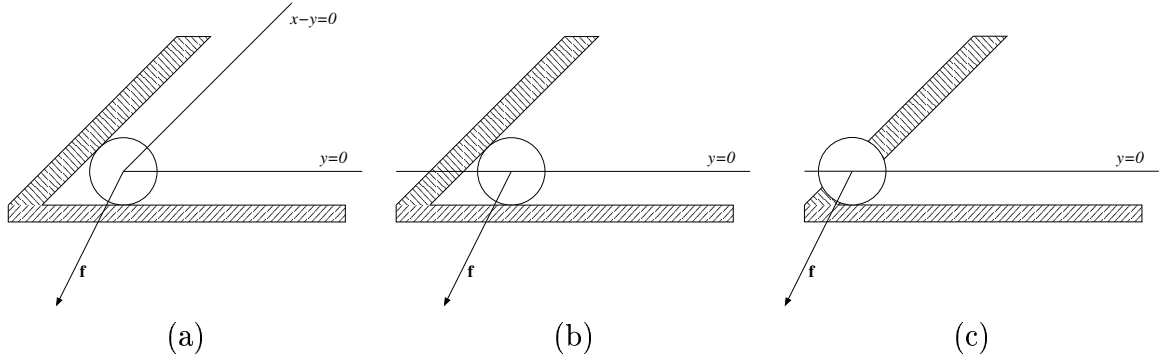


Figure 4.3 – (a) Les deux contraintes sont gérées puisqu'elles sont actives (b) Avec la condition de tendance à la violation, seule la contrainte g_2 reste gérée (c) L'accélération contrainte qui vient d'être calculée amène la particule dans une position illégale

4.3.2.2 L'algorithme correct

L'utilisation de la condition de tendance à la violation $J_k \ddot{\mathbf{x}} < d_k$ améliore la gestion de l'activation des contraintes simultanées, puisque seules les contraintes unilatérales tendant à être violées sont gérées par l'équation 4.12. Mais nous avons vu, avec le second exemple, que ce n'est pas suffisant pour garantir une configuration appropriée. En fait, les contraintes $\mathcal{F}^+$ qui remplissent la condition de tendance à la violation produisent un vecteur Λ de multiplicateurs de Lagrange qui empêchent le système d'entrer dans une configuration illégale compte tenu des contraintes gérées. Cependant, les nouvelles accélérations obtenues $\ddot{\mathbf{x}}$ en tenant compte des efforts de maintien de ces contraintes, peuvent amener le système dans une configuration non autorisée par des contraintes de $\mathcal{F}^-$. La seule solution pour résoudre ce problème est donc d'utiliser ces nouvelles accélérations pour tester si les contraintes unilatérales actives g_k (pour $k \in \mathcal{F}^-$) remplissent la condition de tendance à la violation et doivent donc être déplacées dans $\mathcal{F}^+$. Il faut donc introduire un processus itératif qui calcule les accélérations et teste si une contrainte unilatérale active, précédemment ignorée, doit maintenant être gérée compte tenu de la

tendance à la violation calculée à l'aide des nouvelles accélérations contraintes. Ce processus est répété jusqu'à ce que le système parvienne à une configuration adéquate.

Nous proposons donc une solution simple et efficace pour la gestion de contraintes unilatérales. Au début de chaque étape de l'animation, toutes les contraintes unilatérales actives g_k sont identifiées, et l'ensemble $\mathcal{F}^+$ est vidé. Le processus itératif commence ensuite jusqu'à ce qu'il n'y ait plus de nouvelle insertion dans $\mathcal{F}^+$. Les accélérations contraintes $\ddot{\mathbf{x}}$ sont calculées à partir de l'équation 4.12 et la condition de tendance à la violation $J_k \ddot{\mathbf{x}} < d_k$ évaluée pour chaque contrainte unilatérale active de $\mathcal{F}^-$. Tout indice k d'une contrainte unilatérale g_k qui remplit la condition de tendance à la violation est déplacé dans $\mathcal{F}^+$ puis nous recommençons une nouvelle étape du processus itératif.

ALGO. 3 – L'algorithme correct

```

1 répéter
2   Résoudre l'équation 4.12 pour obtenir  $\ddot{\mathbf{x}}$  and  $\Lambda$ 
3   pour chaque contrainte active  $g_k$  avec  $k \in \mathcal{F}^-$  faire
4     si  $J_k \ddot{\mathbf{x}} < d_k$  alors
5        $k$  est déplacé dans  $\mathcal{F}^+$ 
6 tant que  $\mathcal{F}^+$  est mis à jour;
7 Mettre à jour  $\mathbf{x}$  et  $\dot{\mathbf{x}}$  (intégration numérique)
```

Puisque nous commençons chaque étape avec $\mathcal{F}^+$ vide et que seules les contraintes unilatérales appropriées sont insérées dans $\mathcal{F}^+$ durant le processus itératif, les dernières valeurs calculées de $\ddot{\mathbf{x}}$ et Λ avec l'équation 4.12 amènent à une configuration adéquate du système. Ce processus est ainsi garanti de converger vers une configuration consistante.

Dans une communication récente [RF06], Raghupathi et al. présentent une méthode elle aussi basée sur les multiplicateurs de Lagrange. Dans le souci d'atteindre le temps réel, ils n'autorisent pas le moteur d'animation dynamique à revenir en arrière dans le temps à la première activation de contrainte. Ils doivent gérer les contraintes à la fin de l'étape d'animation, en essayant de déterminer les bonnes accélérations pour assurer l'accomplissement des contraintes. Ils reconnaissent aussi que ce procédé n'est pas assuré de converger dans certains cas.

J'ai implanté ces algorithmes dans MADy afin de pouvoir les comparer. Cette tâche n'a pas été aisée car j'ai dû complètement restructurer le moteur d'animation dynamique déjà existant de manière à ce que l'on puisse passer d'une méthode de gestion des contraintes à l'autre afin de les comparer.

Nous allons maintenant comparer les résultats obtenus avec l'algorithme de Platt et celui de Remion sur l'exemple décrit par la figure 4.1. Les figures 4.4 et 4.5 illustrent une

comparaison des positions et accélérations sur l'axe des y pour une particule de masse $m = 2$ et $m = 3$. Nous rappelons que la contrainte unilatérale g_2 interdit les valeurs négatives en y et que la force constante $\mathbf{f}$ qui s'applique à la particule est égale à $(2, -1)$.

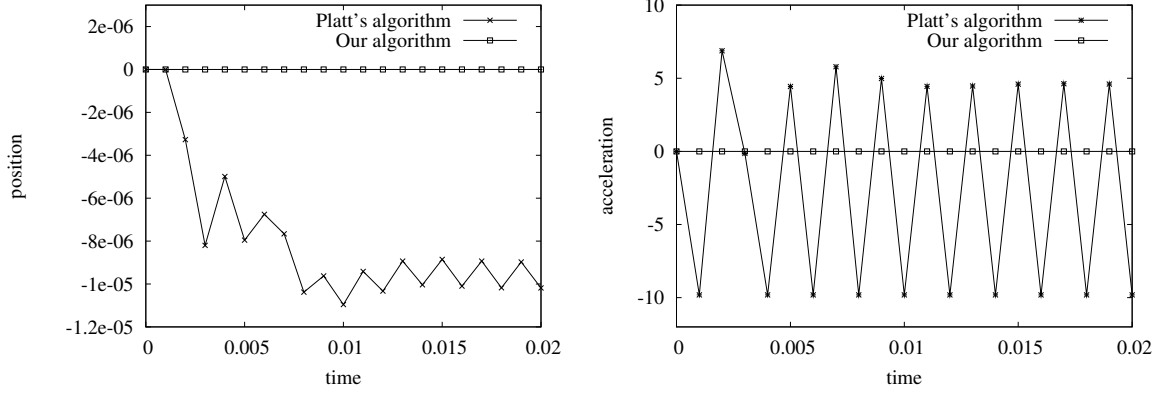


Figure 4.4 – Comparaison de l'algorithme de Platt et du nôtre sur l'exemple illustré par la figure 4.1 pour une masse $m = 2$. Les valeurs numériques correspondent respectivement à la position et à l'accélération le long de l'axe y

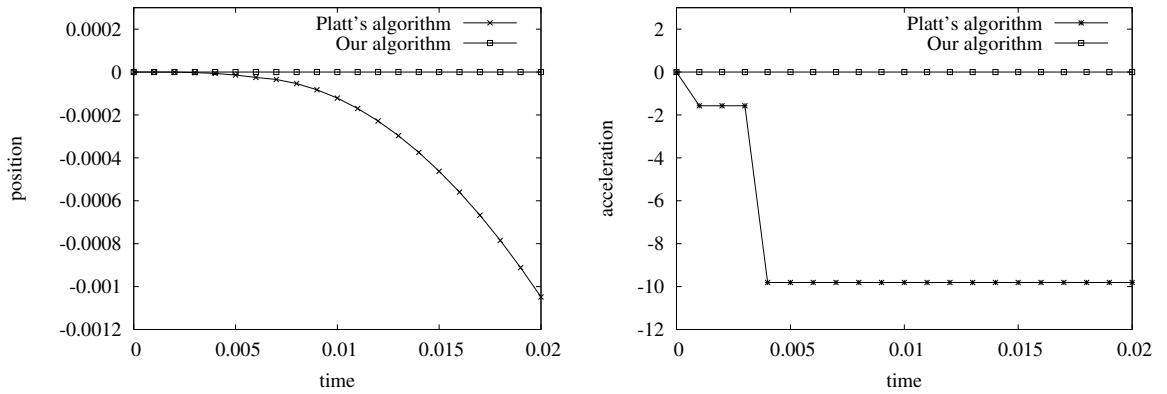


Figure 4.5 – Comparaison de l'algorithme de Platt et du nôtre sur l'exemple illustré par la figure 4.1 pour une masse $m = 3$. Les valeurs numériques correspondent respectivement à la position et à l'accélération le long de l'axe y

Comme le montre la figure 4.4, l'algorithme de Platt retient la particule dans le coin à la première étape de l'animation, et la relâche ensuite à la prochaine étape. Avec comme conséquence, l'évolution de la particule dans une configuration illégale durant les étapes suivantes. Pour la masse $m = 2$, l'erreur de position est inférieure à 10^{-5}m avec une accélération oscillante (colonne de droite). Mais si la masse m est initialisée à 3, comme le montre la figure 4.5, les erreurs sont beaucoup plus importantes et la particule traverse la ligne $y = 0$ modélisée par la contrainte g_2 .

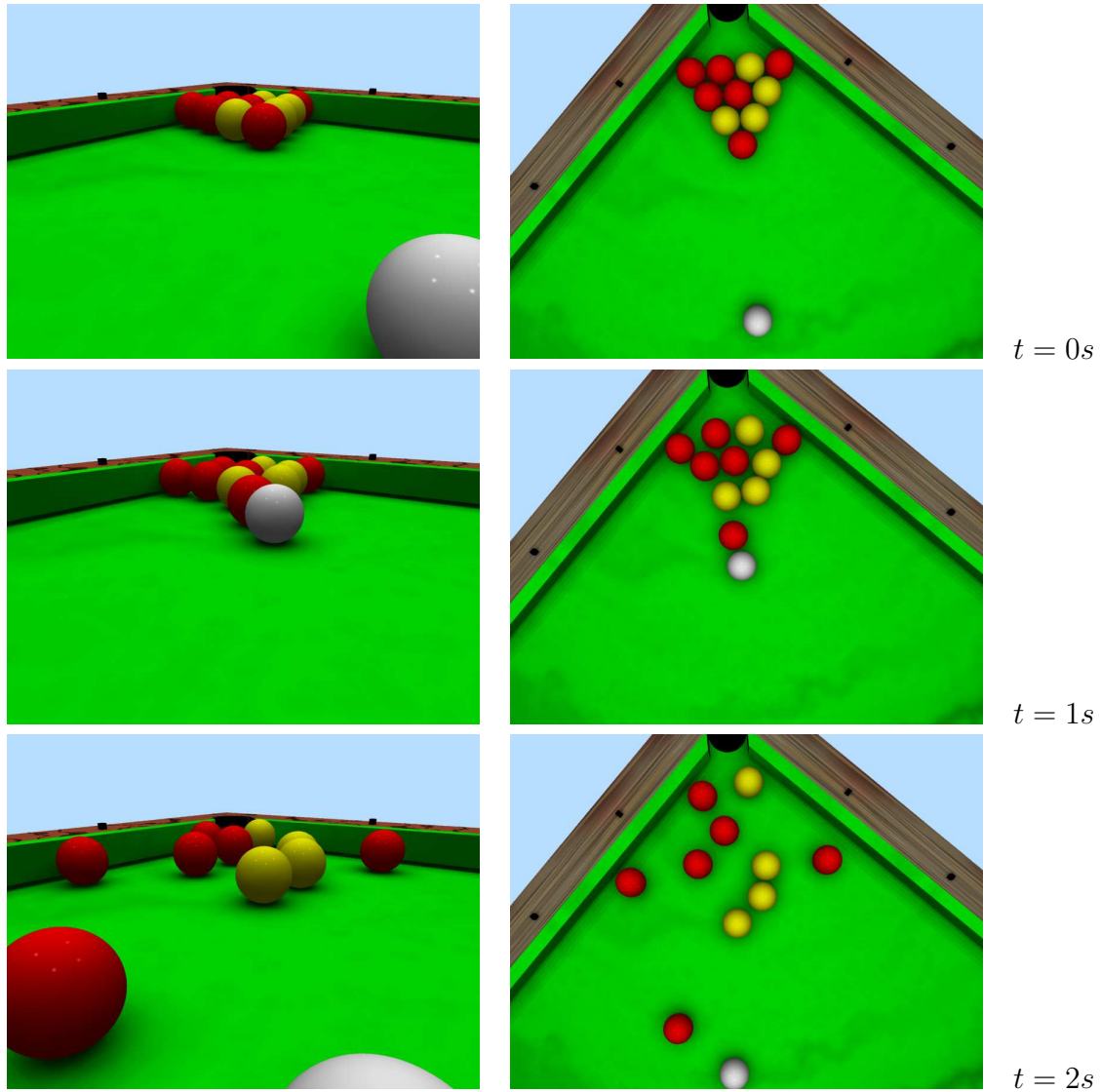


Figure 4.6 – Illustration de notre algorithme de gestion de contraintes dans une scène de billard (11 boules et 77 contraintes unilatérales).

On peut observer sur les illustrations que l'algorithme de Remion maintient la particule sur l'axe des x avec une précision de l'ordre de $10^{-8}m$, dépendante des données de la scène et du pas de temps choisi.

Pour illustrer la gestion de contacts multiples, j'ai modélisé une scène de billard composée de 10 billes initialement immobiles dans un coin et d'une bille mobile glissant vers elles. On définit deux contraintes unilatérales par bille pour modéliser le coin du billard et une contrainte unilatérale pour chaque paire de billes pour empêcher toute interpénétration. Au final, cet exemple est composé de 11 objets mécaniques et de 77 contraintes unilatérales (figure 4.6).

Il n'est pas évident de comparer les temps de calcul des méthodes de Platt et Remion, puisque les contraintes actives simultanées sont mal gérées par l'algorithme de Platt. Si le système entre dans une configuration qui pose problème alors il risque de se trouver dans une configuration illégale pouvant produire des erreurs, soit au court du retour dans le temps soit dans le calcul de l'évolution du système. Par contre, si le système ne rencontre pas de cas problématique, il est évident que dans le pire des cas, la méthode de Remion peut résoudre jusqu'à n systèmes linéaires de dimensions croissantes où n le nombre total de contraintes unilatérales. La complexité de cette solution est donc supérieure à celle de Platt. Cependant, il ne faut pas oublier que cette contribution ne vise pas à améliorer les temps de calcul de la méthode de Platt mais à fournir une solution fiable dans tous les cas à la gestion de l'activation des contraintes multiples, principalement dans un but de temps différé.

Dans le contexte de ce travail, nous disposons maintenant d'un outil fiable pour la gestion fine de l'activation et la désactivation de contraintes unilatérales qui a fait l'objet d'une communication [JNR07b] qui a ensuite été sélectionnée pour une publication dans la revue « Journal of Virtual Reality and Broadcasting » [JNR07a] après relecture et corrections mineures. Il reste maintenant à mettre en place la gestion du contact glissant entre deux surfaces qui nous sera nécessaire au moment de l'activation du contact entre deux surfaces osseuses.

4.4 Le contact glissant

Dans le but d'effectuer une simulation mécanique réaliste de l'articulation du genou, nous devons modéliser le contact entre le fémur et le tibia de manière efficace et robuste. Conformément au principe de la modélisation physique retenu, chaque os est assimilé à un corps rigide dont l'évolution dépend de six degrés de liberté (DdL) représentant sa position et son orientation. En dehors de tout contact, le système mécanique à étudier est donc composé de douze DdL. Le contact entre les deux surfaces osseuses peut se traduire sous deux formes différentes.

La première consiste à réduire manuellement l'ensemble des degrés de liberté du système afin de limiter l'espace des configurations atteignables. Kry et al. [KP03] proposent une méthode visant à exprimer la configuration (position et orientation) d'un des solides par rapport à l'autre. Cette réduction du codage cinématique permet de garantir que le contact sera préservé malgré les erreurs d'arrondi générées lors du calcul numérique. Mais cette approche a un inconvénient majeur : puisque le contact est implicitement intégré dans le codage cinématique du système, il ne peut pas être décroché. Malheureusement, lors d'une flexion extrême du genou, le fémur et le tibia ne sont plus en contact. L'autre

approche consiste à exprimer le contact comme une contrainte dynamique. L'avantage d'une telle approche dans notre contexte de travail est que la contrainte peut aisément être supprimée à tout moment au cours de la simulation. Cette contrainte est intégrée à notre système sous la forme d'une équation implicite dépendant des DdL du système mais aussi de variables supplémentaires permettant d'identifier le point glissant de contact entre les surfaces. Cette méthode repose donc sur l'identification et le suivi des points de contact sur les surfaces. À l'aide des surfaces que nous avons définies dans le chapitre précédent nous disposons d'un formalisme approprié à l'identification des points de contact, il ne reste plus qu'à formaliser la contrainte de contact glissant.

4.4.1 Contrainte de contact glissant

Le formalisme des variables libres repose sur une idée originale des chercheurs du Laboratoire d'Informatique Fondamentale de Lille (LIFL) ainsi que sur des échanges fructueux entre notre laboratoire et le LIFL. Cette technique a été appliquée avec succès par Lenoir dans le cadre particulier de la simulation de sutures [LMGC04, LGM⁺04]. Remion [Rem03] développe, dans un rapport technique interne, un formalisme général permettant d'étendre l'expression des contraintes à des équations intégrant des inconnues surnuméraires appelées variables libres. Au prix d'une réécriture des équations de contraintes, ces variables libres sont gérées au même titre que les degrés de liberté initiaux. Ce schéma de résolution permet donc de formuler des contraintes dynamiques de nature plus complexe comme le contact glissant de type courbe/courbe, courbe/surface et surface/surface. Grâce à l'expression paramétrique de la surface $s : [0, 1]^2 \rightarrow \mathbb{R}^3$, on peut aisément obtenir en tout point de la surface deux vecteurs tangents alignés avec les fibres matérielles issues

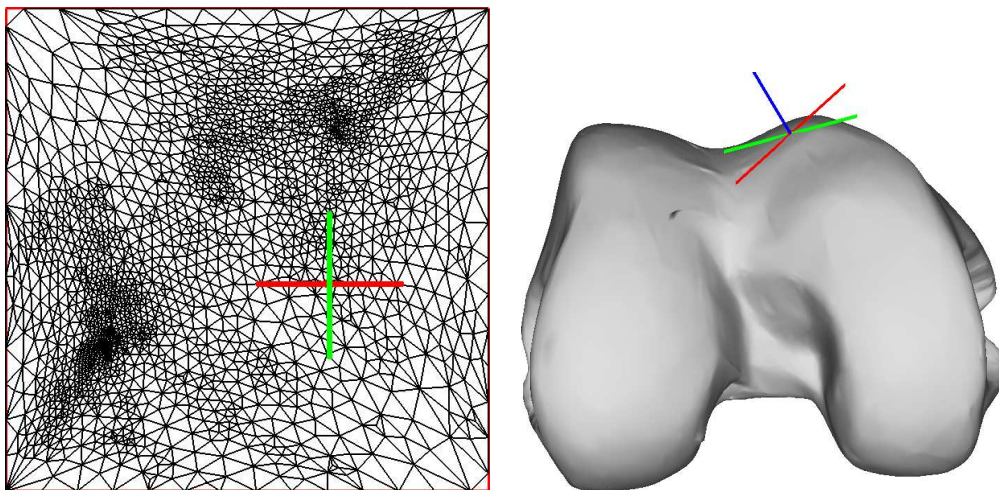


Figure 4.7 – Construction d'un repère local à la surface lors du suivi continu d'un point d'intérêt.

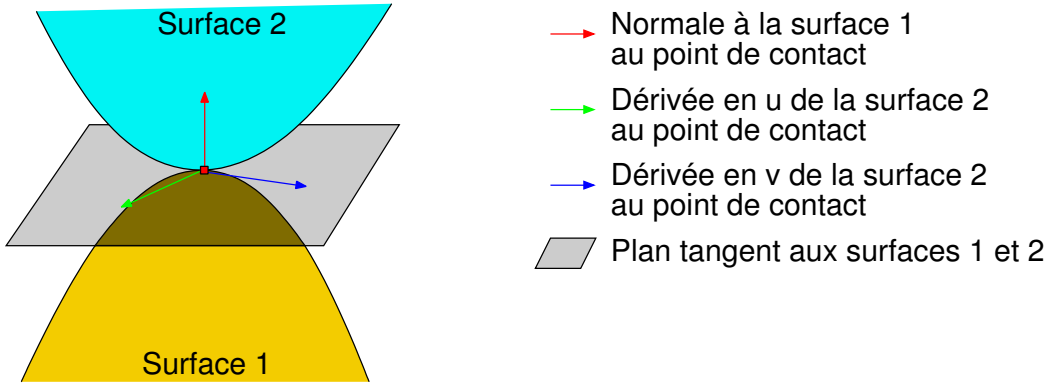


Figure 4.8 – Contact de deux surfaces.

du paramétrage $(\partial_u \mathbf{s}, \partial_v \mathbf{s})$.

La normale est obtenue par un simple produit vectoriel (équation 4.22) qui permet de construire un repère local en ce point (figure 4.7).

$$\mathbf{n}(u, v) = \frac{\partial_u \mathbf{s}(u, v) \times \partial_v \mathbf{s}(u, v)}{\|\partial_u \mathbf{s}(u, v) \times \partial_v \mathbf{s}(u, v)\|} \quad (4.22)$$

On peut alors définir la contrainte de contact glissant entre deux surfaces. Soient $\mathbf{s}_1$ et $\mathbf{s}_2$ deux surfaces paramétriques, le point de contact entre $\mathbf{s}_1$ et $\mathbf{s}_2$ est un point commun aux deux surfaces de sorte que les plans tangents de chaque surface coïncident en ce point (figure 4.8). Si le couple (u_i, v_i) correspond aux coordonnées du point de contact dans le paramétrage de la surface $\mathbf{s}_i$ pour $i = 1$ ou 2 , la contrainte de contact est synthétisée par le système 4.23 :

$$\begin{cases} \mathbf{s}_1(u_1, v_1) = \mathbf{s}_2(u_2, v_2) \\ \partial_u \mathbf{s}_2(u_2, v_2) \cdot \mathbf{n}_1(u_1, v_1) = 0 \\ \partial_v \mathbf{s}_2(u_2, v_2) \cdot \mathbf{n}_1(u_1, v_1) = 0 \end{cases} \quad (4.23)$$

Avec $\mathbf{n}_i(u_i, v_i)$ la normale au point (u_i, v_i) de la surface $\mathbf{s}_i$. On a choisi d'exprimer le contact en utilisant la normale $\mathbf{n}_1$ de la surface $\mathbf{s}_1$ avec les dérivées partielles $\partial_u \mathbf{s}_2$ et $\partial_v \mathbf{s}_2$ de la surface $\mathbf{s}_2$. Il est évidemment possible d'inverser les surfaces $\mathbf{s}_1$ et $\mathbf{s}_2$.

Si l'on considère un système mécanique constitué de deux surfaces $\mathbf{s}_1$ et $\mathbf{s}_2$, nous devons prendre en compte douze DdL (6 pour chaque surface). Avec l'introduction de quatre DdL virtuels (variables libres (u_1, v_1) et (u_2, v_2)), on obtient finalement un système à seize DdL. Mais le système d'équations 4.23 contraint cinq DdL. Ainsi, si on néglige la configuration globale du système (6 DdL correspondant à la position et l'orientation d'une des surfaces), on obtient un système mécanique à cinq DdL, que l'on peut interpréter, à un instant donné, comme deux DdL de glissement dans le plan tangent à la surface,

et trois DdL de rotation autour du point de contact. Le DdL manquant correspond à la translation le long de la normale au plan tangent de contact entre s_1 et s_2 .

À l'aide de cette contrainte, il est donc possible d'imposer un contact glissant entre deux surfaces (figure 4.8). Sur chaque surface, nous identifions le point de contact dont les paramètres sont des variables libres et il ne reste plus qu'à suivre leur évolution dans le système numérique. Nous devons donc maintenant introduire ces variables libres et leur gestion dans le formalisme de Lagrange.

4.4.2 Variables libres

Dans le contexte de l'animation dynamique, l'utilisation des contraintes bilatérales permet de modéliser des restrictions d'évolution du système de manière très générale, comme par exemple « ce point de l'objet doit suivre cette trajectoire », ou encore « ce point de l'objet A doit être attaché à ce point de l'objet B ». Ce qui offre une grande souplesse dans la modélisation de systèmes mécaniques, mais il serait aussi utile, voire nécessaire dans le cas d'un contact glissant, d'exprimer des restrictions d'évolution encore plus complexes, comme « un point quelconque de l'objet A doit être superposé à un point quelconque de l'objet B », ou une contrainte de contact glissant telle que nous l'avons définie ci-dessus.

Ce type de restriction plus générale introduit naturellement des inconnues supplémentaires dans le système, correspondant aux identifiants des points impliqués dans la contrainte. Ces nouvelles inconnues sont naturellement susceptibles d'évoluer au cours du mouvement en fonction du déplacement des points de contact. Nous allons ici étudier la gestion de ces nouvelles inconnues supplémentaires dites “variables libres” dans le système lagrangien.

Considérons un problème mécanique à n DdL quelconques et c contraintes gérées par multiplicateurs de Lagrange. Supposons ce problème posé sous forme classique d'un système linéaire (équation 4.12). Nous rappelons ici ce système :

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}^T \\ -\mathbf{J} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \Lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ -\mathbf{d} \end{bmatrix} \quad (4.24)$$

Avec $\mathbf{M}$ la matrice des masses généralisées, $\mathbf{J}$ la matrice jacobienne de l'ensemble des contraintes géométriques. $\ddot{\mathbf{x}}$ et Λ les inconnues du système, respectivement la dérivée seconde du vecteur d'état et le vecteur des multiplicateurs de Lagrange. Et $\mathbf{f}$ et $\mathbf{d}$ correspondent à la somme des puissances virtuelles des forces externes, pour le premier, et au terme de droite de l'équation 4.11, pour le second.

Considérons maintenant l'ajout d'un jeu de m nouvelles contraintes reposant sur l variables libres. Ces l variables libres sont rassemblées dans un vecteur α susceptible d'évoluer au cours du temps comme les autres DdL du système :

$$\mathbf{g}(\mathbf{x}, \alpha) = \mathbf{0} \quad \mathbf{g} : \mathbb{R}^n \times \mathbb{R}^l \rightarrow \mathbb{R}^m \quad (4.25)$$

Cela revient à ajouter m nouvelles équations et m nouveaux multiplicateurs de Lagrange au système linéaire, ainsi que l nouvelles inconnues. Nous avons ajouté plus d'inconnues que d'équations au système linéaire, il faut donc trouver l équations supplémentaires permettant d'équilibrer le système.

En considérant les variables libres comme des DdL virtuels, ceux-ci doivent être ajoutés au vecteur d'état $\tilde{\mathbf{x}} = (\mathbf{x} \ \alpha)^T$. La fonction d'énergie cinétique $\mathbf{k}$ de ce nouveau vecteur d'état devient alors :

$$\mathbf{k} = \frac{1}{2} \dot{\tilde{\mathbf{x}}}^T \tilde{\mathbf{M}} \dot{\tilde{\mathbf{x}}}, \quad \text{avec } \tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (4.26)$$

Après application des équations de Lagrange à cette fonction d'énergie cinétique, on obtient directement le système suivant :

$$\begin{bmatrix} \tilde{\mathbf{M}} & -\tilde{\mathbf{J}}^T \\ -\tilde{\mathbf{J}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\tilde{\mathbf{x}}} \\ \ddot{\tilde{\Lambda}} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{f}} \\ -\tilde{\mathbf{d}} \end{bmatrix} \quad (4.27)$$

Avec $\tilde{\mathbf{J}}$ la matrice jacobienne de l'ensemble des contraintes (classiques et à variables libres). $\tilde{\Lambda}$ et $\tilde{\mathbf{d}}$ respectivement le vecteur des multiplicateurs de Lagrange et les termes de droite de l'équation 4.11 pour ces contraintes. Et $\tilde{\mathbf{f}}$ correspond à la somme des puissances virtuelles des forces externes, pour le nouveau vecteur d'état. En développant le système 4.27 on obtient :

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} & -\mathbf{J}^T & -\mathbf{J}_{\mathbf{x}}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{J}_{\alpha}^T \\ -\mathbf{J} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{J}_{\mathbf{x}} & -\mathbf{J}_{\alpha} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \ddot{\alpha} \\ \Lambda \\ \Lambda_g \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \\ -\mathbf{d} \\ -\mathbf{d}_g \end{bmatrix} \quad (4.28)$$

Avec $\mathbf{J}$ la matrice jacobienne des contraintes classiques par rapport aux DdL classiques, et $\mathbf{J}_{\mathbf{x}}$ et $\mathbf{J}_{\alpha}$ les matrices jacobienes des contraintes à variables libres respectivement par rapport aux DdL classiques et aux variables libres.

En intégrant ce système dans le moteur d'animation à la place de l'équation 4.12, nous pouvons introduire de nouvelles contraintes reposant sur des nouveaux DdL sur-numéraires α , dont l'évolution reste libre. Nous disposons ainsi d'une nouvelle méthode générique et élégante pour exprimer des contraintes complexes.

Le système 4.28 peut s'écrire sous la forme d'un système d'équations :

$$\begin{cases} \mathbf{M}\ddot{\mathbf{x}} - \mathbf{J}^T\Lambda - \mathbf{J}_x^T\Lambda_g &= \mathbf{f} \\ \mathbf{J}_\alpha^T\Lambda_g &= \mathbf{0} \\ \mathbf{J}\ddot{\mathbf{x}} &= -\mathbf{d} \\ \mathbf{J}_x\ddot{\mathbf{x}} + \mathbf{J}_\alpha\ddot{\alpha} &= -\mathbf{d}_g \end{cases} \quad (4.29)$$

Il est à remarquer que la résolution de ce système peut s'effectuer de manière optimisée par découplage [Rem03].

Nous disposons maintenant de tous les outils permettant l'implantation des variables libres et des contraintes les utilisant. Ces contraintes permettent l'expression de contraintes complexes, tels que les contacts glissants. Lenoir en a testé et validé le fonctionnement dans le cas particulier des courbes splines matérielles [LMGC04, LGM⁺04]. Nous proposons ici, à partir des travaux de Remion [Rem03], d'en généraliser l'expression. En effet, bien que leur intérêt soit évident dans l'optique du genou pour les contacts glissants des surfaces osseuses, ces contraintes peuvent avoir de nombreux champs d'application comme la gestion des contacts entre mailles dans la simulation fine de textiles tricotés, autre travail de recherche mené au sein du CReSTIC/SIC.

4.5 Conclusion

Dans l'optique de créer un modèle efficace de l'articulation du genou, reposant sur les contacts osseux, nous avons vu qu'il était nécessaire d'avoir recours au formalisme Lagrangien. En effet, celui-ci permet une modélisation efficace des structures de type squelettique, tout en laissant la possibilité d'introduire par la suite des structures déformables. De plus, ce formalisme est le plus approprié à la gestion des contraintes de contact qui auront pour but de simuler l'action des ligaments.

Après avoir déterminé le formalisme, nous avons pu nous intéresser à la gestion de contraintes nécessaires à la simulation des contacts osseux. À partir des travaux de Remion [Rem00] j'ai pu implanter une procédure fiable de gestion de l'activation/désactivation des contraintes. En effet, l'algorithme généralement utilisé proposé par Platt [Pla92a] peut-être mis en défaut dans certaines configurations. Cette solution a fait l'objet de deux publications [JNR07a, JNR07b]. Par ailleurs, ce travail m'a demandé de modifier la structure du moteur d'animation dynamique développé au sein du laboratoire, ce qui m'a permis d'acquérir une bonne expérience sur sa structuration et son fonctionnement général.

J'ai ensuite pu proposer une méthode pour la simulation des contacts glissants entre

deux surfaces osseuses ainsi que les outils nécessaires à sa mise en œuvre dans un moteur Lagrangien. En effet, cette méthode repose sur l'utilisation de nouvelles variables et contraintes, mises en place conjointement entre le CReSTIC/SIC et le LIFL. Ces nouvelles données imposent une réécriture du système mécanique. Nous n'avons malheureusement pas eu le temps dans le cadre de cette thèse d'implanter cette technique dans le moteur Lagrangien, cependant, nous disposons maintenant de toutes les données pour la mettre en place.

Chapitre 5

Conclusion générale

« “YOU couldn’t deny that, even if you tried with both hands.”
“I don’t deny things with my *hands*,” Alice objected.
“Nobody said you did,” said the Red Queen. “I said you couldn’t if
you tried.” »

Lewis CARROLL, *Through the Looking-Glass*. (1872).

Sommaire

5.1	Bilan et contributions	91
5.2	Perspectives	93

5.1 Bilan et contributions

Comme nous l’avons vu dans le chapitre 2, la modélisation du genou est une tâche ardue. L’objectif de cette thèse était la mise en place d’un modèle cinématique de l’articulation du genou. Pour des raisons de retard des crédits dans le projet SysReeduc, je n’ai pas pu disposer des données nécessaires à la modélisation du genou. Cependant, j’ai mis en place un jeu d’outils offrant une base saine pour le développement de modèles d’articulations réalistes. Ces modèles pourront permettre par expérimentation virtuelle de qualifier formellement leurs DdL véritables.

Ces informations sont importantes pour la simulation dynamique de l’articulation, mais également dans le cadre de l’animation traditionnelle et de la cinématique inverse. Car avec l’expression formelle des degrés de liberté réels d’une articulation, nous serions en mesure de l’animer de manière plus réaliste qu’avec les liaisons classiques pour un surcoût raisonnable.

Nous nous sommes intéressés à l'articulation du genou car il s'agit d'une articulation complexe. Ce qui impose de mettre en place des outils et des procédures efficaces qui auront de plus grandes chances de pouvoir s'adapter à d'autres articulations. Nous nous appuyons sur des connaissances d'experts afin de modéliser de manière efficace et réaliste l'interaction entre le tibia et le fémur tout en prenant en compte les restrictions dues aux ligaments. La modélisation de l'articulation s'effectue en deux étapes ; la première consiste à fournir une représentation géométrique des structures à simuler et la seconde à décrire les interactions possibles entre ces structures.

La représentation géométrique des structures à simuler repose, pour l'essentiel, sur la construction d'une surface paramétrique continue et interpolante à paramètres globaux pour chacune des surfaces osseuses. Ces surfaces devront être obtenues à partir de données réelles stockées sous la forme d'un maillage triangulaire. J'ai atteint cet objectif en proposant une procédure pour construire ces surfaces en deux étapes ; la paramétrisation, puis l'interpolation.

L'étape de paramétrisation consiste à obtenir une paramétrisation pour le maillage triangulaire initial. Parmi les nombreuses solutions existantes, j'ai présenté quelques méthodes pour la paramétrisation de maillage topologiquement équivalent à un disque ou à une sphère.

L'étape d'interpolation s'appuie sur la paramétrisation obtenue à l'étape précédente, pour construire la surface paramétrique souhaitée. La méthode repose sur l'utilisation de surfaces à base radiale pour interpoler le jeu de données. Cette méthode a l'avantage de pouvoir s'appliquer à un maillage non régulier. Cependant d'importantes ressources sont nécessaires aux calculs des SPBR, limitant leur utilisation à de petits maillages. J'ai donc proposé une extension de cette méthode reposant sur les partitions de l'unité, levant ainsi la limitation sur la densité des maillages. Cette technique s'appuie sur la subdivision du domaine paramétrique. Afin de l'appliquer à différents domaines paramétriques, le carré unité et la sphère unité, j'en ai proposé deux versions.

Ces travaux offrent les outils appropriés à la modélisation géométrique des surfaces osseuses. Nous avons donc pu nous intéresser à la description de leurs interactions. Ces interactions s'exprimant dans le modèle cinématique à l'aide de contraintes, nous nous sommes donc intéressés à la gestion des contraintes. Et afin de disposer d'une méthode fiable, nous avons employé une solution itérative d'activation/désactivation des contraintes, permettant de résoudre les défauts de la méthode de Platt. En effet, la méthode de Platt peut être mise en défaut dans certains cas où des contraintes antagonistes s'activent simultanément. L'algorithme de Remion nous assure une configuration cohérente au cours de l'animation. J'ai donc implanté dans le moteur d'animation dynamique (MADy), développé au sein

du laboratoire, les deux méthodes afin de valider la méthode de Remion. Ce travail de programmation m'a permis, en modifiant la structure du moteur, d'en appréhender le fonctionnement global.

Une fois assuré de disposer d'une gestion fine des contraintes, j'ai mis en place le modèle de contact glissant entre deux surfaces osseuses à l'aide de contraintes dites « à variables libres ». Ces contraintes reposent sur l'utilisation de nouvelles inconnues du système mécanique dont l'évolution au cours de l'animation n'est pas corrélée aux degrés de liberté du système. Ces nouvelles inconnues ont été introduites par le LIFL et le CReSTIC/SIC. J'explique ensuite le fonctionnement et la manière dont ces « variables libres » peuvent être introduites dans le système d'évolution mécanique.

Nous disposons donc au final de tous les outils nécessaires à la simulation cinématique de l'articulation du genou. Et aussitôt que les variables libres auront été intégrées au moteur, nous pourrons valider notre modèle cinématique de l'articulation du genou.

5.2 Perspectives

À très court terme, je compte finir l'intégration des variables libres dans le moteur d'animation dynamique durant les quelques mois suivant la fin de ce travail de thèse. Nous pourrons alors envisager de mettre en place une procédure de validation de ce modèle. Pour que cette procédure soit cohérente, nous devons comparer les courbes cinématiques obtenues par notre modèle à partir des données IRM d'un individu, aux courbes cinématiques obtenues par capture du mouvement du même individu. Évidemment, cette manipulation devra être effectuée un grand nombre de fois et sa mise en place ne sera pas facile. Mais nous pourrons utiliser les contacts que nous avons noués dans le cadre du projet SysReeduc, notamment avec les biomécaniciens du LACM (Laboratoire d'Analyse des Contraintes Mécaniques).

Une fois le modèle validé, nous pourrons l'utiliser pour explorer les courbes cinématiques produites par un jeu de simulation, et ainsi essayer d'en extraire les courbes paramétriques des rotations secondaires par rapport à l'angle de flexion. Sur le même principe, nous pourrons essayer d'identifier d'autres caractéristiques remarquables, comme des corrélations entre les formes osseuses et le mouvement. Pour tous ces travaux, un important travail devra être effectué pour traiter les données, tant en amont de la simulation qu'en aval. Fort heureusement, nous disposons au sein du CReSTIC de collègues spécialistes des domaines relatifs au traitement du signal et des images.

À plus long terme, il sera possible de complexifier le modèle en y ajoutant de nouveaux

éléments (ligaments, rotule, $\dots$), voir même de rendre le modèle actif en y intégrant le fonctionnement des muscles. Par ailleurs, si le modèle est validé, nous pouvons envisager de l'appliquer à d'autres articulations du corps pour obtenir en les regroupant un méta-modèle d'humain virtuel.

D'autre part, nous avons développé les SPBR afin de répondre à nos besoins concernant la modélisation géométrique des surface osseuses. Cette méthode semble également pouvoir s'appliquer à de nombreuses applications telles que le remaillage ou le rendu non-photoréaliste par exemple. Nous n'avons pas eu le temps d'explorer en profondeur ce type de surfaces, et une étude plus poussée semble judicieuse.

Publications personnelles

« “W^{HY},” said the Dodo, “the best way to explain it is to do it.”

Lewis CARROLL, *Alice's Adventures in Wonderland*. (1865).

Soit :

- 1 revue internationale [8].
- 2 communications internationales [6, 7].
- 3 communications nationales sans comité de lecture [1, 2, 5].
- 2 contributions à l'AS Humain Virtuel [3, 4].

- [1] A. JONQUET, D. GILLARD, Y. REMION et D. DELIESSCHE, « Vers une modélisation biomécanique de l'articulation du genou », dans *16èmes journées de l'Association française d'informatique graphique / 1ères rencontres du Chapitre français d'Eurographics*, p. 111–120, Saint-Denis, décembre 2003.
- [2] A. JONQUET, « Tour d'horizon des modèles articulaires, cas d'étude : le genou », dans *11èmes journées du GT Animation et simulation*, Reims, France, juin 2004.
- [3] D. GILLARD et A. JONQUET. partie « Animation Dynamique » du thème 2 « Mouvement et Contrôle du mouvement » du rapport de l'Action Spécifique 151 (Humain Virtuel), 10 pages, 2004.
- [4] D. GILLARD et A. JONQUET. partie « Modèles articulaires » du thème 3 « Modélisation » du rapport de l'Action Spécifique 151 (Humain Virtuel), 11 pages, 2004.
- [5] A. JONQUET, O. NOCENT et Y. REMION, « Utilisation des rbfs pour la construction de surfaces paramétriques interpolantes », dans *18èmes journées de l'Association française d'informatique graphique / 3èmes rencontres du Chapitre français d'Eurographics*, Strasbourg, novembre 2005.
- [6] A. JONQUET, O. NOCENT et Y. REMION, « Smooth parametric surfaces retrieval from triangular meshes using rbfs », dans *Computer Graphics and Applications 2006*, p. 40–46, Novosibirsk Akademgorodok, Russia, juillet 2006.
- [7] A. JONQUET, O. NOCENT et Y. REMION, « The art to keep in touch - the good use of lagrange multipliers », *International Conference on Computer Graphics Theory and Applications*, mars 2007.

- [8] A. JONQUET, O. NOCENT et Y. REMION, « The art to keep in touch - the good use of lagrange multipliers », *Journal of Virtual Reality and Broadcasting - GRAPP 2007 Special Issue*, vol. 4, n° 15, 2007. (version relue et corrigée de [7]).

Bibliographie

« ONCE OR TWICE SHE HAD PEEPED INTO THE BOOK HER SISTER was reading, but it had no pictures or conversations in it, “and what is the use of a book,” thought Alice, “without pictures or conversations?” »

Lewis CARROLL, *Alice's Adventures in Wonderland*. (1865).

- [ARH93] Abdel-Rahman, E. and Hefzy, M. A two-dimensional dynamic anatomical model for the human knee joint. *ASME Transactions, Journal of Biomechanical Engineering*, vol. 115, n° 9 : 357–365, novembre 1993. (Cit   p. 21 et 22)
- [AMHG83] Andriacchi, T., Mikosz, R., Hampton, S., and Galante, J. Model studies of the stiffness characteristics of the human knee joint. *Journal of Biomechanics*, vol. 16 : 23–29, 1983. (Cit   p. 21)
- [AG85] Armstrong, W. and Green, M. The dynamics of articulated rigid bodies for purposes of animation. *Graphics Interface 85 Proceedings*, pages 407–415, 1985. (Cit   p. 69)
- [AMFT07] Atmani, H., Merienne, F., Fofi, D., and Trouilloud, P. Computer-aided surgery system for shoulder prosthesis placement. vol. 12, n° 1 : 63–73, janvier 2007. (Cit   p. 4)
- [Bar93] Baraff, D. Non-penetrating rigid body simulation. dans *State of the Art Reports, Eurographics '93*, September 1993. (Cit   p. 72)
- [BB88] Barzel, R. and Barr, A. H. 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, New York, NY, USA, 1988. ACM Press. (Cit   p. 74)
- [Bau72] Baumgarte, J. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, vol. 1 : 1–16, 1972. (Cit   p. 73)
- [BVIG91] Bennis, C., V  zien, J.-M., Igl  sias, G., and Gagalowicz, A. Piecewise surface flattening for non-distorted texture mapping. dans Sederberg, T. W.,   diteur,

- Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 237–246, 1991. (Cité p. 37)
- [Bjö96] Björck, Å. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, Penn., 1996. 1032 pages. (Cité p. 74)
- [BHD84] Blankevoort, L., Huiskes, R., and Delange, A. *An in-vitro Study of the Passive Kinematic Behaviour of the Human Knee Joint*. R.L.Spilker, décembre 1984. (Cité p. 21)
- [BDM06] Boulic, R., Donikian, S., and Multon, F. Modèles pour les humanoïdes. dans de l'Ecoles de Mines de Paris, P., éditeur, *Le traité de la réalité virtuelle*. 2006. 3ème édition, chapitre 10. (Cité p. 67)
- [BC89] Bruderlin, A. and Calvert, T. Goal-directed, dynamic animation of human walking. *Computer graphics*, pages 233–242, 1989. (Cité p. 67)
- [BC96] Bruderlin, A. and Calvert, T. Knowledge-driven, interactive animation of human running. *Graphics interface'96*, pages 213–221, 1996. (Cité p. 67)
- [BA98] Bull, A. and Amis, A. Knee joint motion : description and measurement. dans *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine*, volume 212, pages 357–372, 1998. (Cité p. 17)
- [BRR⁺02] Büchler, P., Ramaniraka, N., Rakotomanana, L., Iannotti, J., and Farron, A. A finite element model of the shoulder : Application to the comparison of normal and osteoarthritic joints. *Clinical Biomechanics*, vol. 17 : 630–639, 2002. (Cité p. 4)
- [CG05] Calais-Germain, B. *Anatomie pour le mouvement*. DésIris, 2005. (Cité p. 9 et 16)
- [CBC⁺01] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. Reconstruction and representation of 3D objects with radial basis functions. dans *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76, 2001. (Cité p. 50 et 52)
- [Cat74] Catmull, E. E. *A subdivision algorithm for computer display of curved surfaces*. Thèse de doctorat, University of Utah, décembre 1974. (Cité p. 37)
- [CWZ⁺02] Chen, J., Wechsler, H., Zhu, Y., MacMahon, E., and Carmines, D. Knee surgery assistance : Patient model reconstruction, motion simulation, and biomechanical visualization, 2002. soumis dans IEEE Transactions on Bio-medical Engineering. (Cité p. 25 et 26)
- [Che02] Chevallier, J.-M. *Appareil locomoteur 2*. Flammarion Medecine-sciences, 2002. (Cité p. 9)
- [CH99] Chung, S.-k. and Hahn, J. K. Animation of human walking in virtual environments. *EUROGRAPHICS Computer Animation and Simulation '99*, pages 4–15, 1999. (Cité p. 4)

-
- [Cot97] Cotin, S. *Modèles anatomiques déformables en temps-réel*. Thèse de doctorat, Université de Nice-Sofia Antipolis, 1997. (Cité p. 67)
- [CPJ76] Crowninshield, R., Pope, M., and Johnson, R. An analytical model of the knee. *Journal of Biomechanics*, vol. 9 : 397–405, 1976. (Cité p. 18, 19 et 20)
- [DOTG91] Davis, R., Ounpuu, S., Tyburski, D., and Gage, J. A gait analysis data collection and reduction technique. *Human Movement Science*, vol. 10, n° 5 : 575–587, 1991. (Cité p. 23)
- [DB78] Denham, R. and Bishop, R. Mechanics of the knee and problems in reconstructive surgery. *Journal of Bone and Joint Surgery*, vol. 60-B, n° 3 : 345–352, 1978. (Cité p. 20)
- [DMA02] Desbrun, M., Meyer, M., and Alliez, P. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum*, vol. 21 : 209–218, 2002. (Cité p. 42 et 43)
- [DLN⁺94] Dongarra, J., Lumsdaine, A., Niu, X., Pozo, R., and Karin, R. A sparse matrix library in c++ for high performance architectures. dans *Proceedings of the Second Object Oriented Numerics Conference*, pages 214–218, 1994. <http://math.nist.gov/sparselib++/>. (Cité p. 44 et 49)
- [DCDS97] Duchamp, T., Certain, A., DeRose, T., and Stuetzle, W. Hierarchical computation of pl harmonic embeddings. Rapport technique, University of Washington, juillet 1997. (Cité p. 42)
- [Duc77] Duchon, J. Splines minimizing rotation-invariant semi-norms in sobolev spaces. dans Schempp, W. and Zeller, K., éditeurs, *Constructive Theory of Functions of Several Variables*, pages 85–100, Berlin-Heidelberg, 1977. Springer-Verlag. (Cité p. 51)
- [EDD⁺95] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W. Multiresolution analysis of arbitrary meshes. dans *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 173–182, 1995. (Cité p. 48)
- [ELL70] Edwards, R., Lafferty, J., and Lang, K. Ligament strain in the human knee joint. *ASME Transactions, Journal of Basic Engineering*, vol. 92 : 131–136, 1970. (Cité p. 19 et 20)
- [EZT97] Ellis, R., Zion, P., and Tso, C. Interactive visual and force rendering of human-knee dynamics. *International Symposium on Experimental Robotics*, pages 173–182, 1997. (Cité p. 4)
- [EGW88] Ensberg, J., Grimston, S., and Wackwitz, J. Predicting talocalcaneal joint attitudes form talocalcaneal/talocrural joint attitudes. *Journal of Orthopaedic Research*, vol. 6 : 749–757, 1988. (Cité p. 23)
- [FP03] Fang, A. and Pollard, N. Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics*, vol. 22, n° 3 : 417–426, 2003. (Cité p. 70)

- [Flo97] Floater, M. S. Parameterization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, vol. 14, n° 4 : 231–250, 1997. (Cité p. 39 et 40)
- [Flo98] Floater, M. S. Parametric tilings and scattered data approximation. *International Journal of Shape Modeling*, vol. 4 : 165–182, 1998. (Cité p. 42)
- [Flo03a] Floater, M. S. Mean value coordinates. *Computer Aided Geometric Design*, vol. 20, n° 1 : 19–27, 2003. (Cité p. 41)
- [Flo03b] Floater, M. S. One-to-one piecewise linear mappings over triangulations. *Mathematics of Computation*, vol. 72, n° 242 : 685–696, 2003. (Cité p. 39)
- [FH05] Floater, M. S. and Hormann, K. Surface parameterization : a tutorial and survey. dans Dodgson, N. A., Floater, M. S., and Sabin, M. A., éditeurs, *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 157–186. Springer, Berlin, Heidelberg, 2005. (Cité p. 38)
- [FBB71] Frankel, V., Burstein, A., and Brooks, D. Biomechanics of internal derangement of the knee. pathomechanics as determined by analysis of the instant centres of motion. *Journal of Bone and Joint Surgery*, vol. 53-A : 945–962, July 1971. (Cité p. 18)
- [Gel06] Gelas, A. *Compactly supported radial basis functions : multidimensional reconstruction and applications*. Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, 2006. (Cité p. 52)
- [GJ04a] Gillard, D. and Jonquet, A. partie « Animation Dynamique » du thème 2 « Mouvement et Contrôle du mouvement » du rapport de l’Action Spécifique 151 (Humain Virtuel), 10 pages, 2004. (Cité p. 5)
- [GJ04b] Gillard, D. and Jonquet, A. partie « Modèles articulaires » du thème 3 « Modélisation » du rapport de l’Action Spécifique 151 (Humain Virtuel), 11 pages, 2004. (Cité p. 5)
- [Gil97] Gillard, D. *Vers l’animation dynamique d’un corps humain virtuel : utilisation de données biomédicales 3D*. Thèse de doctorat, Université de Reims Champagne-Ardenne, 1997. (Cité p. 4)
- [GO78] Goodfellow, J. and O’Connor, J. The mechanics of the knee and prosthesis design. *Journal of Bone and Joint Surgery*, vol. 60-B : 358–369, 1978. (Cité p. 19)
- [GGS03] Gotsman, C., Gu, X., and Sheffer, A. Fundamentals of spherical parameterization for 3D meshes. dans *SIGGRAPH ’03 : ACM SIGGRAPH 2003 Papers*, pages 358–363, 2003. (Cité p. 47)
- [GH97] Greiner, G. and Hormann, K. Interpolating and approximating scattered 3D-data with hierarchical tensor product B-splines. dans *Surface Fitting and Multiresolution Methods*, Innovations in Applied Mathematics, pages 163–172. Vanderbilt University Press, Nashville, TN, 1997. (Cité p. 44)

-
- [GH82] Grood, E. and Hefzy, M. An analytical technique for modelling knee joint stiffness - part i : Ligamentous forces. *ASME Transactions, Journal of Biomechanical Engineering*, vol. 104 : 330–337, novembre 1982. (Cité p. 20)
- [GY03] Gu, X. and Yau, S.-T. Global conformal surface parameterization. dans *SGP '03 : Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 127–137, 2003. (Cité p. 48)
- [Gus04] Guskov, I. An anisotropic mesh parameterization scheme. *Engineering with Computers*, vol. 20, n° 2 : 129–135, 2004. (Cité p. 44)
- [HAT⁺00] Haker, S., Angenent, S., Tannenbaum, A., Kikinis, R., Sapiro, G., and Halle, M. Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, n° 2 : 181–189, avril 2000. (Cité p. 47)
- [HPR05] Halloran, J. P., Petrella, A. J., and Rullkoetter, P. J. Explicit finite element modeling of total knee replacement mechanics. *Journal of Biomechanics*, vol. 38, n° 2 : 323–331, février 2005. (Cité p. 68)
- [Han06] Hanson, A. J. *Visualizing Quaternions*. Interactive 3D Technology. Morgan Kaufmann, janvier 2006. (Cité p. 23)
- [HG83] Hefzy, M. and Grood, E. An analytical technique for modelling knee joint stiffness - part ii : Geometrical non-linearities. *ASME Transactions, Journal of Biomechanical Engineering*, vol. 105 : 145–153, 1983. (Cité p. 20)
- [HG88] Hefzy, M. and Grood, E. Review of knee models. *Applied Mechanics Review*, vol. 41, n° 1 : 1–13, janvier 1988. (Cité p. 20)
- [HP97] Hodgins, J. K. and Pollard, N. S. Adapting simulated behaviors for new characters. *SIGGRAPH '97 Proceedings*, pages 153–162, 1997. (Cité p. 4)
- [HLS07] Hormann, K., Lévy, B., and Sheffer, A. Mesh parameterization : Theory and practice. dans *SIGGRAPH 2007 Course Notes*, volume 2, pages 1–122, San Diego, CA, août 2007. ACM Press. (Cité p. 38)
- [IGG01] Isenburg, M., Gumhold, S., and Gotsman, C. Connectivity shapes. dans *VIS '01 : Proceedings of the conference on Visualization '01*, pages 135–142, 2001. (Cité p. 47 et 48)
- [Jon04] Jonquet, A. Tour d’horizon des modèles articulaires, cas d’étude : le genou. dans *11èmes journées du GT Animation et simulation*, Reims, France, juin 2004. (Cité p. 5)
- [JGRD03] Jonquet, A., Gillard, D., Remion, Y., and Deliessche, D. Vers une modélisation biomécanique de l’articulation du genou. dans *16èmes journées de l’Association française d’informatique graphique / 1ères rencontres du Chapitre français d’Eurographics*, pages 111–120, Saint-Denis, décembre 2003. (Cité p. 5)

- [JNR05] Jonquet, A., Nocent, O., and Remion, Y. Utilisation des rbfs pour la construction de surfaces paramétriques interpolantes. dans *18èmes journées de l'Association française d'informatique graphique / 3èmes rencontres du Chapitre français d'Eurographics*, Strasbourg, novembre 2005. (Cité p. 5, 37 et 64)
- [JNR06] Jonquet, A., Nocent, O., and Remion, Y. Smooth parametric surfaces retrieval from triangular meshes using rbfs. dans *Computer Graphics and Applications 2006*, pages 40–46, Novosibirsk Akademgorodok, Russia, juillet 2006. (Cité p. 5, 37 et 64)
- [JNR07a] Jonquet, A., Nocent, O., and Remion, Y. The art to keep in touch - the good use of lagrange multipliers. *Journal of Virtual Reality and Broadcasting - GRAPP 2007 Special Issue*, vol. 4, n° 15, 2007. (version relue et corrigée de [JNR07b]). (Cité p. 6, 83 et 88)
- [JNR07b] Jonquet, A., Nocent, O., and Remion, Y. The art to keep in touch - the good use of lagrange multipliers. *International Conference on Computer Graphics Theory and Applications*, mars 2007. (Cité p. 6, 83 et 88)
- [KAC88] Kaufman, K., An, K., and Chao, E. A dynamic mathematical model of the knee joint applied to isokinetic exercise. dans *Computational Methods in Bioengineering*, volume 9, pages 157–167. ASME Biomechanical Engineering Division, R.L. Spilker, 1988. (Cité p. 21)
- [KLS03] Khodakovsky, A., Litke, N., and Schröder, P. Globally smooth parameterizations with low distortion. dans *SIGGRAPH '03 : ACM SIGGRAPH 2003 Papers*, pages 350–357, 2003. (Cité p. 48)
- [Ko95] Ko, H. *Kinematic and dynamic techniques for analyzing, predicting and animating human locomotion*. Thèse de doctorat, University of Pennsylvania, 1995. (Cité p. 69)
- [Koc02] Kocsis, L. More precise measurement method for 3D gait analysis. *Third Conference on Mechanical Engineering Proceedings*, pages 215–234, 2002. (Cité p. 25)
- [KPJ02] Komdeur, P., Pollo, F., and Jackson, R. Dynamic knee motion in anterior cruciate impairment : a report and case study. *Baylor University Medical Center Proceedings 2002*, vol. 15 : 257–259, 2002. (Cité p. 26)
- [KV03] Kós, G. and Várady, T. Parameterizing complex triangular meshes. dans *5th International Conference on Curves and Surfaces*, pages 265–274. Saint-Malo 2002, Nashboro Press, 2003. (Cité p. 38)
- [KL96] Krishnamurthy, V. and Levoy, M. Fitting smooth surfaces to dense polygon meshes. *Computer Graphics*, vol. 30, n° Annual Conference Series : 313–324, 1996. (Cité p. 37)
- [KP03] Kry, P. G. and Pai, D. K. Continuous contact simulation for smooth surfaces. *ACM Trans. Graph.*, vol. 22, n° 1 : 106–129, 2003. (Cité p. 83)

- [LSS⁺98] Lee, A. W. F., Sweldens, W., Schröder, P., Cowsar, L., and Dobkin, D. Maps : multiresolution adaptive parameterization of surfaces. dans *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 95–104, 1998. (Cité p. 48)
- [LKL02] Lee, Y., Kim, H. S., and Lee, S. Mesh parameterization with a virtual boundary. *Computers & Graphics*, vol. 26, n° 5 : 677–686, 2002. (Cité p. 38)
- [LGM⁺04] Lenoir, J., Grisoni, L., Meseure, P., Remion, Y., and Chaillou, C. Smooth constraints for spline variational modeling. dans *International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, pages 58–64, Singapore, juin 2004. (Cité p. 33, 84 et 88)
- [LMGC04] Lenoir, J., Meseure, P., Grisoni, L., and Chaillou, C. A suture model for surgical simulation. dans *2nd International Symposium on Medical Simulation*, pages 105–113, 2004. (Cité p. 84 et 88)
- [LBIB48] Levens, A., Berkeley, C., Inman, V., and Blosser, J. Transverse rotation of the segments of the lower extremity in locomotion. *Journal of Bone and Joint Surgery*, vol. 30-A : 859–872, 1948. (Cité p. 23)
- [LM00] Levy, B. and Mallet, J.-L. Paramétrisation des surfaces triangulées. *Revue Internationale de CFAO et d'Informatique Graphique*, vol. 15, n° 1 : 25–42, juin 2000. (Cité p. 42, 43, 44, 45 et 46)
- [LL77] Lewis, J. and Lew, W. A note on the description of articulating joint motion. *Journal of Biomechanics*, vol. 10 : 675–678, 1977. (Cité p. 18)
- [LL78] Lewis, J. and Lew, W. A method for locating an optimal “fixed” axis of rotation for the human knee joint. *ASME Transactions, Journal of Biomechanical Engineering*, vol. 100 : 187–193, 1978. (Cité p. 18)
- [LS96] Liesenfeld, R. and Stolfi, J. Dynamic animation of elastic bodies. *Anais do IX Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens (SIBGRAPI'96)*, pages 265–272, 1996. (Cité p. 68)
- [MTC00] Magnenat-Thalmann, N. and Cordier, F. Construction of a human topological model from medical data. *IEEE Transactions on Information Technology in Biomedicine*, vol. 4, n° 2 : 137–143, juin 2000. (Cité p. 4)
- [MP77] Maquet, P. and Pelzer, G. Evolution of the maximum stress in osteoarthritis of the knee joint. *Journal of Biomechanics*, vol. 10 : 107–117, 1977. (Cité p. 20)
- [MMA76] Markolf, K., Mensch, J., and Amstutz, H. Stiffness and laxity of the knee—the contributions of the supporting structures. a quantitative in vitro study. *Journal of Bone and Joint Surgery*, vol. 58-A, n° 9 : 583–594, juillet 1976. (Cité p. 21 et 23)
- [MTH96] Maurel, W., Thalmann, D., and Hoffmeyer, P. A biomechanical musculoskeletal model of human upper limb for dynamic simulation. *EUROGRAPHICS Computer Animation and Simulation '96*, pages 121–136, 1996. (Cité p. 4)

- [MHB⁺03] McCarthy, A., Hose, D., Barber, D., Wood, S., Darwent, G., Chan, D., Bickerstaff, D., and Wilkinson, I. A registration-based mr method for calculating in-vivo 3-d knee joint motion : Validating finite element simulations. dans *Proceedings of the International Society for Magnetic Resonance in Medicine*, pages 10–16, mai 2003. (Cité p. 67)
- [MWH⁺02] McCarthy, A., Wilkinson, I., Hose, D., Barber, D., Wood, S., Darwent, G., Chan, D., and Bickerstaff, D. Musculo-skeletal simulation : Finite element meshes derived from magnetic resonance volumes. dans *Proceedings of the International Society for Magnetic Resonance in Medicine*, pages 18–24, mai 2002. (Cité p. 68)
- [MC74] McLeod, W. and Cross, M. Knee joint axis of rotation. *Proceedings of the 27th Annual Conference on Engineering in Medicine and Biology*, 1974. (Cité p. 18)
- [Mea82] Meagher, D. J. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, vol. 2, n° 19 : 129–147, 1982. (Cité p. 56)
- [MMKA04] Ménardais, S., Multon, F., Kulpa, R., and Arnaldi, B. Motion blending for real-time animation while accounting for the environment. dans *Computer Graphics International*, pages 156–159, juin 2004. (Cité p. 67)
- [MLBD02] Meyer, M., Lee, H., Barr, A. H., and Desbrun, M. Generalized barycentric coordinates for irregular polygons. *Journal of Graphics Tools*, vol. 7, n° 1 : 13–22, 2002. (Cité p. 42)
- [Min81] Minns, R. Forces at the knee joint : Anatomical considerations. *Journal of Biomechanics*, vol. 14, n° 9 : 633–643, 1981. (Cité p. 20)
- [MI91] Miyazaki, S. and Ishida, A. New mathematical definition and calculation of axial rotation of anatomical joints. *ASME Transactions, Journal of Biomechanical Engineering*, vol. 113, n° 3 : 270–275, septembre 1991. (Cité p. 23)
- [MEA83] Moeinzadeh, M., Engin, A., and Akkas, N. Two-dimensional dynamic modeling of human knee joint. *Journal of Biomechanics*, vol. 16, n° 4 : 253–264, 1983. (Cité p. 21, 22 et 23)
- [Mor70] Morrison, J. The mechanics of the knee joint in relation to normal walking. *Journal of Biomechanics*, vol. 3 : 51–61, 1970. (Cité p. 19)
- [Mos53] Mosely, H. Disorders of the knee. *Ciba Clinical Symposia*, vol. 5, n° 6, 1953. (Cité p. 18)
- [NW00] Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, New York, 2000. 656 pages. (Cité p. 74)
- [Noc01] Nocent, O. *Animation dynamique de corps déformables continus. Application à la simulation de textiles tricotés*. Thèse de doctorat, Université de Reims Champagne-Ardenne, 2001. (Cité p. 4)

- [NN00] Noh, J.-y. and Neumann, U. Talking faces. dans *IEEE International Conference on Multimedia and Expo (II)*, pages 627–630, 2000. (Cité p. 50)
- [Nou99] Nourrit, J.-M. *Modélisation, animation et visualisation de textiles à base de mailles*. Thèse de doctorat, Université de Reims Champagne-Ardenne, 1999. (Cité p. 4)
- [Pea85] Peachey, D. R. Solid texturing of complex surfaces. *SIGGRAPH Comput. Graph.*, vol. 19, n° 3 : 279–286, 1985. (Cité p. 37)
- [PD01] Piazza, S. and Delp, S. Three-dimensional dynamic simulation of total knee replacement motion during a step-up task. *ASME Transactions, Journal of Biomechanical Engineering*, vol. 123, n° 6 : 599–606, décembre 2001. (Cité p. 4)
- [Pic01] Picinbono, G. *Modèles géométriques et physiques pour la simulation d'interventions chirurgicales*. Thèse de doctorat, Université de Nice-Sofia Antipolis, 2001. (Cité p. 67)
- [Pla92a] Platt, J. C. A generalization of dynamic constraints. *CVGIP : Graphical Models and Image Processing*, vol. 54, n° 6 : 516–525, 1992. (Cité p. 74 et 88)
- [Pla92b] Platzer, W. *Anatomie 1 - Appareil locomoteur*. Flammarion Medecine-sciences, 1992. (Cité p. 9)
- [PH03] Praun, E. and Hoppe, H. Spherical parametrization and remeshing. dans *SIGGRAPH '03 : ACM SIGGRAPH 2003*, pages 340–349, 2003. (Cité p. 48 et 49)
- [PSS01] Praun, E., Sweldens, W., and Schröder, P. Consistent mesh parameterizations. dans *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 179–184, 2001. (Cité p. 48)
- [RF06] Raghupathi, L. and Faure, F. QP-collide : A new approach to collision treatment. dans *Journées du groupe de travail Animation et Simulation (GTAS)*, Annual French Working group on Animation and Simulation, pages 91–101. Institut de Recherche en Informatique de Toulouse, juin 2006. <http://www-evasion.inrialpes.fr/Publications/2006/RF06>. (Cité p. 80)
- [RK91] Ramakrishnan, H. and Kadaba, M. On the estimation of joint kinematics during gait. *Journal of Biomechanics*, vol. 24 : 969–977, 1991. (Cité p. 23)
- [Rem00] Remion, Y. *Animation dynamique : moteur lagrangien généraliste et applications*. Habilitation à Diriger des Recherches, Université de Reims Champagne-Ardenne, 2000. (Cité p. 4, 73, 74 et 88)
- [Rem03] Remion, Y. *Prise en compte de « contraintes à variables libres »*. Rapport technique, CReSTIC - LERI Université Reims Champagne-Ardenne, février 2003. (Cité p. 33, 84 et 88)

- [Reu03] Reuter, P. *Reconstruction and Rendering of Implicit Surfaces from Large Unorganized Point Sets*. Thèse de doctorat, Université Bordeaux 1, décembre 2003. (Cité p. 52)
- [RM95] Ruprecht, D. and Müller, H. Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications*, vol. 15, n° 2 : 37–43, 1995. (Cité p. 50)
- [SYGS05] Saba, S., Yavneh, I., Gotsman, C., and Sheffer, A. Practical spherical embedding of manifold triangle meshes. dans *SMI '05 : Proceedings of the International Conference on Shape Modeling and Applications 2005*, pages 258–267, 2005. (Cité p. 47)
- [SSGH01] Sander, P. V., Snyder, J., Gortler, S. J., and Hoppe, H. Texture mapping progressive meshes. dans *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 409–416, 2001. (Cité p. 46)
- [SZ90] Schröder, P. and Zeltzer, D. The virtual erector set : dynamic simulation with linear recursive constraint propagation. dans *Symposium on Interactive 3D graphics Proceedings*, pages 23–31, Snowbird, Utah, United States, 1990. (Cité p. 69)
- [SPNS80] Seering, W., Piziali, R., Nagel, D., and Schurman, D. The function of the primary ligaments of the knee in varus-valgus and axial rotation. *Journal of Biomechanics*, vol. 13 : 785–794, 1980. (Cité p. 18 et 19)
- [ST98] Shapiro, A. and Tal, A. Polyhedron realization for shape transformation. *The Visual Computer*, vol. 14, n° 8/9 : 429–444, 1998. (Cité p. 48 et 49)
- [Shi61] Shinno, N. Static-dynamic analysis of movement of the knee. *Tokushima Journal of Experimental Medicine*, vol. 8, n° 1 : 101–110, 1961. (Cité p. 18)
- [Sho85] Shoemake, K. Animating rotation with quaternion curves. *ACM SIGGRAPH '85*, vol. 19, n° 3 : 245–254, 1985. (Cité p. 23)
- [Smi73] Smidt, G. Biomechanical analysis of knee flexion and extension. *Journal of Biomechanics*, vol. 6 : 79–92, 1973. (Cité p. 18)
- [SOBW88] Sutherland, D., Olshen, R., Biden, E., and Wyatt, M. *The development of mature walking*. London : Mac Keith Press, 1988. (Cité p. 23)
- [TKZ⁺05] Teschner, M., Kimmerle, S., Zachmann, G., Heidelberger, B., Raghupathi, L., Fuhrmann, A. L., Cani, M.-P., Faure, F., Magnetat-Thalmann, N., and Strasser, W. Collision detection for deformable objects. volume 24 of *Computer Graphics Forum*, pages 61–81, mars 2005. (Cité p. 75)
- [Tro98] Troy, J. J. Real-time dynamic balancing and walking control of a 7-link planar biped. dans *Proceedings of ASME Design Engineering Technical Conferences*, 1998. (Cité p. 73)

-
- [TE93] Tumer, S. and Engin, A. three-body segment dynamic model of the human knee. *ASME Transactions, Journal of Biomechanical Engineering*, vol. 115, n° 4 : 350–356, 1993. (Cité p. 21)
- [Tut63] Tutte, W. T. How to draw a graph. dans *Proceedings of the London Mathematical Society*, volume 13, pages 743–768, 1963. (Cité p. 39)
- [VF02] Vetterling, W. T. and Flannery, B. P. *Numerical Recipes in C++ : The Art of Scientific Computing*. Cambridge University Press, Cambridge (UK) and New York, 2nd édition, 2002. 1032 pages. (Cité p. 72)
- [WW74] Wang, C. and Walker, P. Rotary laxity of the human knee. *Journal of Bone and Joint Surgery*, vol. 65-A : 161, 1974. (Cité p. 19)
- [WWW73] Wang, C., Walker, P., and Wolf, B. The effects of flexion and rotation on the length patterns of the ligaments of the knee. *Journal of Biomechanics*, vol. 6 : 587–596, 1973. (Cité p. 18)
- [Wen95] Wendland, H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, vol. 4, n° 1 : 389–396, décembre 1995. (Cité p. 52)
- [Wen02a] Wendland, H. Fast evaluation of radial basis functions : Methods based on partition of unity. dans Chui, C. K., Schumaker, L. L., and Stöckler, J., éditeurs, *Approximation Theory X : Abstract and Classical Analysis*, pages 473–483. Vanderbilt University Press, 2002. (Cité p. 55)
- [Wen02b] Wendland, H. Surface reconstruction from unorganized points, 2002. (Cité p. 55)
- [Wen05] Wendland, H. *Scattered Data Approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge, janvier 2005. (Cité p. 52)
- [Wie94] Wieland, A. *Modeling, simulation and control of two-legged walking*. Thèse de doctorat, University of California, Los Angeles, 1994. (Cité p. 67 et 73)
- [WB85] Wilhelms, J. and Barski, B. Using dynamic analysis to animate articulated bodies as humans and robots. dans *Graphics Interface'85*, pages 97–104, Montréal, mars 1985. (Cité p. 69)
- [WVJ⁺80] Wismans, J., Veldpaus, F., Janssen, J., Huson, A., and Struben, P. A three-dimensional mathematical model of the knee joint. *Journal of Biomechanics*, vol. 13 : 677–685, 1980. (Cité p. 20 et 21)
- [WHB84] Wongchaisuwat, C., Hemami, H., and Buchner, H. Control of sliding and rolling at natural joints. *ASME Transactions, Journal of Biomechanical Engineering*, vol. 106 : 368–375, novembre 1984. (Cité p. 21 et 22)
- [YK01] Ying, N. and Kim, W. Use of dual euler angles to describe general spatial movements of human joints. dans *2001 Summer Bioengineering Conference Proceedings*, volume 50, pages 273–274, Snowbird, Utah, 2001. (Cité p. 24)

BIBLIOGRAPHIE

- [YK02] Ying, N. and Kim, W. Use of dual euler angles to quantify the three-dimensional joint motion and its application to the ankle joint complex. *Journal of Biomechanics*, vol. 35, n° 12 : 1647–1657, décembre 2002. (Cité p. 24)
- [YSSC89] Yoshioka, Y., Siu, D., Scudamore, R., and Cooke, T. Tibial anatomy and functional axes. *Journal of Orthopaedic Research*, pages 132–137, 1989. (Cité p. 19)
- [YBS04] Yoshizawa, S., Belyaev, A. G., and Seidel, H.-P. A fast and simple stretch-minimizing mesh parameterization. dans *Proceedings of the Shape Modeling International 2004 (SMI'04)*, pages 200–208, 2004. (Cité p. 46, 48 et 49)

Résumé

Ce travail de thèse s'inscrit dans le projet CPER « SysReeduc » visant à la mise en place d'un système d'aide à la rééducation de l'articulation du genou. Au sein de ce projet transdisciplinaire réunissant biomécaniciens, informaticiens et automaticiens, mon travail consistait à fournir, à partir de données géométriques réelles, des courbes d'évolution cinématique de l'articulation. Par ailleurs, ces travaux s'insèrent dans un projet plus vaste d'animation réaliste d'un humain virtuel conduit par le groupe « Signal, Image et Connaissance » (SIC) du « Centre de Recherche en Sciences et Technologies de l'Information et de la Communication » (CReSTIC). Dans cette optique le travail proposé consiste, dans un premier temps, à modéliser une articulation (le genou en ce qui nous concerne) pour valider une méthodologie générale applicable au reste du squelette humain. J'ai ainsi mis en place une série d'outils, tant géométriques que cinématiques, pour la modélisation de l'articulation du genou.

La modélisation géométrique des surfaces osseuses s'effectue à l'aide de surfaces continues à paramétrage global : les surfaces à base radiale. Celles-ci sont construites en deux étapes, à partir des données géométriques du maillage triangulaire issu d'un scanner : une paramétrisation du maillage triangulaire, puis une interpolation à l'aide de fonctions à base radiale. Pour pallier au problème de dimensionnalité, nous proposons ensuite une amélioration de cette méthode reposant sur les partitions de l'unité. Il nous est ainsi possible de définir une surface paramétrique continue à paramétrage global sur un maillage sans trou, ouvert ou fermé.

Disposant d'une description mathématique des surfaces osseuses appropriée à la simulation en mécanique lagrangienne, nous définissons alors un ensemble de contraintes afin d'exprimer le contact glissant entre les surfaces osseuses du fémur et du tibia. Ces nouvelles contraintes nécessitent l'introduction de nouvelles variables, qualifiées de variables libres, et leur intégration pertinente au sein des équations du mouvement. Enfin, nous revenons sur le problème de la gestion des contraintes simultanées en proposant une amélioration d'une méthode classique d'activation et désactivation des contraintes.

Mots-clés: Modélisation géométrique, paramétrisation, animation dynamique, gestion de contraintes.