Université Joseph Fourier Grenoble I

Modélisation d'objets déformables avec un système de particules orientées

Thèse présentée par :

Jean-Christophe Lombardo

pour obtenir le titre de :

Docteur de l'Université Joseph Fourier
spécialité Informatique

Arrêtés ministériels du 5 Juillet 1984 et du 30 Mars 1992

soutenue le 10 Janvier 1996 devant la commission d'examen :

MM.	Philippe	CINQUIN	Président
	Bernard	PÉROCHE	Rapporteur
	Daniel	THALMANN	Rapporteur
	Andrew J.	HANSON	Invité
	Marie-Paule	GASCUEL	
	Claude	PUECH	Directeur

Thèse préparée au sein du laboratoire *iMAGIS/IMAG*
iMAGIS est un projet commun entre le *CNRS*, l'*INRIA*, l'*INPG* et l'*UJF*

Remerciements

Je souhaite remercier les diverses personnes qui ont permis la réalisation des travaux décrits dans ce document. Mes remerciements vont :

- en premier lieu, à Claude Puech, directeur du laboratoire iMAGIS/IMAG, qui a initié ce sujet de recherche et a toujours été présent pour m'apporter son soutien tant du point de vue scientifique qu'humain;
- à Philippe Cinquin, directeur de l'équipe GMCAO du laboratoire TIMC/IMAG qui m'a accueilli dans son équipe pendant la première année de ma thèse et qui a accepté d'être membre du jury;
- à Bernard Péroche et Daniel Thalmann qui ont accepté d'être rapporteur;
- à Andrew J. Hanson qui a accepté de participer a mon jury de thèse;
- à Marie-Paule Gascuel pour les nombreuses discussions que nous avons eu pendant ces trois années de thèses, et pour avoir accpeté d'être membre du jury;
- à Jean-Dominique Gascuel sans qui *Fabule* n'existerait pas, et qui m'a transmis une petite partie de son expertise;
- à Vincent Dessenne de l'équipe GMCAO pour son soutien;
- à Nicolas Tsingos pour son expertise du langage *Tcl*;
- à Mathieu Desbrun pour notre collaboration et son dévouement;
- à toute l'équipe iMAGIS (tant les *doctorants* que les *permanents*) pour l'aide apportée et la chaleureuse ambiance qui règne au laboratoire;
- et enfin à Vincent, Mähnu, Mathieu, Maurice et Claude qui ont relu attentivement ce document.

Je ne peux oublier dans cette page de remerciement Pascale, ma femme, Florian et Océane, mes enfants, qui m'ont apporté l'équilibre nécessaire à la réalisation de ce travail.

Table des matières

Table des figures

Chapitre 1

Introduction

La synthèse d'image par ordinateur se développe autour de deux pôles principaux :

– le rendu, ou synthèse d'images fixes, et

– l'animation, ou synthèse d'images animées.

La modélisation, qui est le troisième axe important de recherche en synthèse d'images, est de plus en plus liée aux deux pôles pré-cités (la modélisation est généralement faite pour l'animation ou pour le rendu). Le rendu a maintenant atteint un niveau de réalisme tel que l'utilisation d'images de synthèse pour le prototypage informatique avant construction est de plus en plus employé, que ce soit par exemple pour le design des voitures ou l'éclairage des théâtres[1]. C'est en étudiant et en simplifiant les modèles utilisés par les physiciens, que le rendu est devenu photoréaliste [Lom92].
L'animation par ordinateur est en train de suivre un chemin similaire.
Initialement, l'animation par ordinateur avait pour objectif de diminuer les coûts de production des studios de dessins animés. Les techniques utilisées n'étaient qu'une automatisation des techniques d'interpolation entre dessins clefs appliquées par les dessinateurs. Ces modèles dits descriptifs permettent et nécessitent un contrôle complet[2] du mouvement, ouvrant ainsi la porte au manque de réalisme. Pour contrecarrer ce dernier point, un nouveau type de modèles est apparu : les modèles dits générateurs qui utilisent les lois de la mécanique pour calculer automatiquement le mouvement.

Le document est structuré comme suit : le chapitre 2 contient une description des systèmes de particules, de leur invention jusqu'aux développements les plus récents. Les concepts fondamentaux, comme par exemple la

1. Par exemple les éclairages de *La bohème* de Puccini, monté au *Metropolitan Opera* de New York ont été entièrement simulés sur ordinateur avant d'être réalisés [DSG91].

2. Le mouvement résultant n'est dû qu'à la volonté de l'animateur et à son savoir faire. Aucune propriété de l'objet n'est prise en compte pour aider au réalisme de l'animation.

définition d'une loi d'interaction et d'une particule orientée, sont présentés dans ce chapitre.
Le chapitre 3 traite le problème des oscillations des particules lié à l'utilisation de systèmes dynamique interagissant. Nous proposons une solution originale en définissant une nouvelle loi d'attraction/répulsion dont les propriétés mathématiques ont été choisie pour limiter les erreurs commises lors de l'intégration des équations de la dynamique.
La modélisation d'objets déformables avec un système de particules orientées est introduite au chapitre 4. Dans la première partie de ce chapitre nous présentons les nouvelles fonctions d'interaction anisotropes qui permettent de maintenir certaines propriétés géométriques des objets modélisés. L'algorithme de simulation ainsi qu'un mécanisme de typage des particules permettant de subdiviser un objet en plusieurs parties sont présentés. La seconde partie de ce chapitre présente une autre application du modèle précédemment défini. Après avoir présenté les méthodes de simulation d'objets déformables avec des surfaces implicites, nous proposons un nouveau type d'objet, les objets modélisés par des surfaces implicites engendrées par des squelettes à mémoire de forme.
L'utilisation de systèmes de particules orientées pour simuler des objets déformables demande de définir chaque particule (position et loi d'interaction). Dès que l'objet n'est plus élémentaire, cette étape devient difficile à réaliser. Le chapitre 5 propose une méthode automatique de reconstruction d'objet à partir de données volumique tri-dimensionnelles sur la géométrie de l'objet. Les choix faits permettrons d'utiliser cet algorithme pour générer automatiquement des objets déformables à partir de données issues de l'imagerie médicale.
Afin de tester la faisabilité des techniques décrites dans ce document, un premier simulateur d'objets bidimensionnels a été écrit. Ce test franchi avec succès, une version 3D des particules orientées a été intégrée à la plateforme d'animation développée par le laboratoire iMAGIS. Ces deux implémentations sont décrites de façon succincte au chapitre 6.
Le chapitre 7 est consacré à l'une des applications possible des particules orientées. Les bases d'une méthode permettant de simuler un muscle, tant par ses déformations (aspect visuel) que par les forces produites (aspect biomécanique), sont données.

Dans la suite de cette introduction, nous allons examiner brièvement les principaux modèles d'objets déformables existants, puis nous décrirons les principes fondamentaux de la dynamique ainsi que les méthodes permettant des les appliquer pour l'animation par modèles physiques.

1.1 Objets déformables

La littérature en informatique graphique propose de nombreux modèles d'objets déformables. Dans cette section, nous allons examiner les principaux modèles.

Les modèles existants peuvent se regrouper en deux grandes familles : ceux qui implémentent les équations physiques de l'élasticité, et ceux où les objets sont construits en assemblant des éléments simples, comme des masses et des ressorts.

1.1.1 Discrétisation des lois de l'élasticité

En 1987, D. Terzopoulos, J. Platt, A. Barr et K. Fleischer [TPBF87] proposent un modèle d'objet déformable élastique qui, pour la première fois, est basé sur les lois physiques de l'élasticité. Dans ce modèle, les coordonnées d'un point $\vec{a}$ de l'objet dans son repère local sont données à l'instant t par $\vec{r}(\vec{a},t)$. Les équations du mouvement des points de l'objet déformable sont données sous forme d'équation de Lagrange :

$$\frac{\partial}{\partial t}\left(\mu\frac{\partial\vec{r}}{\partial t}\right)+\gamma\frac{\partial\vec{r}}{\partial t}+\frac{\delta\epsilon(\vec{r})}{\delta\vec{r}}=\vec{f}(\vec{r},t) \tag{1.1}$$

où $\mu(\vec{a})$ est la densité de l'objet au point $\vec{a}$, $\gamma(\vec{a})$ est le coefficient d'amortissement au point $\vec{a}$, $\vec{f}(\vec{r},t)$ représente les forces externes dont le point d'application est $\vec{r}$ à l'instant t ($\vec{r}$ est la position occupée par le point $\vec{a}$ du corps déformable à l'instant t), et $\epsilon(\vec{r})$ est l'énergie potentielle de déformation de l'objet à l'instant t. Le premier terme modélise l'inertie en fonction de la répartition des masses dans l'objet, le second terme modélise l'absorption d'énergie due à la déformation, et le dernier est l'énergie élastique due à la déformation.
L'énergie potentielle de déformation $\epsilon(\vec{r})$ est définie par :

$$\epsilon(\vec{r})=\int_{\Omega}\|\mathbf{G}-\mathbf{G}_0\|^2 da_x da_y da_z \ . \tag{1.2}$$

Dans cette définition, Ω est l'objet considéré, $\mathbf{G}$ est le tenseur métrique de l'objet dans l'état courant, et $\mathbf{G}_0$ est le tenseur métrique de l'objet non déformé. $\mathbf{G}$ est la matrice 3x3 :

$$\mathbf{G}(\vec{r}(\vec{a}))=\begin{pmatrix} \frac{\partial^2\vec{r}}{\partial a_x^2} & \frac{\partial\vec{r}}{\partial a_x}\cdot\frac{\partial\vec{r}}{\partial a_y} & \frac{\partial\vec{r}}{\partial a_x}\cdot\frac{\partial\vec{r}}{\partial a_z} \\ \frac{\partial\vec{r}}{\partial a_x}\cdot\frac{\partial\vec{r}}{\partial a_y} & \frac{\partial^2\vec{r}}{\partial a_y^2} & \frac{\partial\vec{r}}{\partial a_x}\cdot\frac{\partial\vec{r}}{\partial a_z} \\ \frac{\partial\vec{r}}{\partial a_x}\cdot\frac{\partial\vec{r}}{\partial a_y} & \frac{\partial\vec{r}}{\partial a_x}\cdot\frac{\partial\vec{r}}{\partial a_z} & \frac{\partial^2\vec{r}}{\partial a_z^2} \end{pmatrix} \tag{1.3}$$

Ces lois (Équations 1.1, 1.2 et 1.3) s'appliquent à un matériau continu. La résolution numérique de ces équations repose sur une double discrétisation, spatiale et temporelle. L'objet Ω doit être échantillonné et les lois de l'élasticité doivent être évaluées sur cette représentation discrète de l'objet. Une surface, par exemple, est paramétrée sur un carré de coté 1. Ce carré est ensuite discrétisé par $N \times M$ échantillons a_{ij} de coordonnées $\big(i/(N-1), j/(M-1)\big)$ où $i = [0..N-1]$ et $j = [1..M-1]$. En prenant comme approximation des dérivées partielles les différences finies correspondantes, l'équation de l'élasticité 1.1 est ré-écrite sous une forme discrète adaptée au calcul numérique :

$$\mathbf{M}\frac{\partial^2 \mathsf{r}}{\partial t^2} + \mathbf{C}\frac{\partial \mathsf{r}}{\partial t} + \mathbf{K}(\mathsf{r})\mathsf{r} = \mathsf{f} \ , \tag{1.4}$$

où

- r est le vecteur de taille NM contenant la position courante de tous les échantillons de la surface,
- $\mathbf{M}$ est la matrice diagonale $NM \times NM$ donnant la répartition des masses,
- $\mathbf{C}$ est la matrice $NM \times NM$ d'amortissement,
- $\mathbf{K}(\mathsf{r})$ est la matrice $NM \times NM$ de raideur, et
- f est le vecteur NM des forces extérieures exercées sur le maillage.

La résolution de cette équation se fait en discrétisant le temps. La position $\mathsf{r}_{t+\Delta t}$ des échantillons à l'instant $t+\Delta t$ est calculée en fonction de la position r_t des échantillons à l'instant t :

$$\mathbf{A}_t \mathsf{r}_{t+\Delta t} = \mathsf{g}_t \ , \tag{1.5}$$

où

$$\mathbf{A}_t = \mathbf{K}(\mathsf{r}_t) + \left(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{2\Delta t}\mathbf{C}\right) \ ,$$

et

$$\mathsf{g}_t = \mathsf{f}_t + \left(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{2\Delta t}\mathbf{C}\right)\mathsf{r}_t + \left(\frac{1}{\Delta t}\mathbf{M} - \frac{1}{2\Delta t}\mathbf{C}\right)\mathsf{v}_t \ ,$$

avec

$$\mathsf{v}_t = \frac{\mathsf{r}_t - \mathsf{r}_{t-\Delta t}}{\Delta t} \ .$$

Ce modèle est le premier modèle d'objet déformable reposant sur la caractérisation physique de l'élasticité. L'intégration des lois de la dynamique demande, à chaque pas de temps, de résoudre l'équation différentielle 1.4 et donc d'inverser la matrice $\mathbf{A}_t$. Compte tenu de la taille de cette matrice (n^2 pour un objet discrétisé en n échantillons) cette opération est très coûteuse.

En 1988, D. Terzopoulos et A. Witkin [TW88] proposent de diminuer le temps de calcul en modélisant l'objet en deux parties : un composant de référence indéformable et un composant déformable élastique. Ceci permet, moyennant une approximation supplémentaire sur la loi d'élasticité (l'élasticité ne peut maintenant plus qu'être linéaire), d'avoir une matrice **A**, dans la version de l'équation 1.5 adaptée au composant déformable, indépendante du temps. Le gain est considérable, puisqu'il suffit maintenant d'inverser **A** une fois pour toute et non plus à chaque pas de simulation.

La même année, D. Terzopoulos et K. Fleischer [TF88] proposent une autre amélioration du modèle : en permettant au composant de référence de se déformer lui aussi, il est possible de simuler des comportements inélastiques. Dès que l'énergie absorbée par le composant déformable dépasse un certain seuil, le surplus est transféré au composant de référence. Ce dernier est déformé en conséquence. Suivant le mode de déformation du composant de référence et suivant le seuil à partir duquel les forces lui sont appliquées, les principaux types de comportements de la matière peuvent être simulés : l'élasticité[3], la visco-élasticité[4], la plasticité[5] et même les fractures.
Ce modèle de fracture est cependant très simpliste : même brisé en parties non connexes, l'objet reste unique, alors qu'il devrait y avoir création d'un nouvel objet pour chaque morceau et destruction de l'objet d'origine. Du point de vue de sa représentation, un objet reste modélisé comme un objet connexe (une seule matrice r, une seule matrice $\mathbf{K}$, ...etc, et surtout un seul centre de gravité et un seul repère local) tout le temps de la simulation. Je pense que le comportement des diverses parties d'un objet brisé en morceau ne doit pas correspondre à ce que nous attendons intuitivement. Il est par ailleurs intéressant de constater que dans l'article [TF88] aucun exemple de fracture complète (l'objet est séparé en deux parties non connexes) n'est présenté.

J.-P. Gourret, N. Magnenat Thalmann et D. Thalmann proposent, en 1989, une approche un peu différente [GTT89]. La relation liant les forces externes appliquées à l'objet et les forces internes de réaction aux déformations de l'objet n'est plus exprimée sous forme d'équation de Lagrange mais suivant le principe des travaux virtuels. D'après le principe de la conservation d'énergie, le travail des forces internes de l'objet doit être égal au travail des forces externes qui lui sont appliquées. En raison du mode de calcul de ces travaux, les forces extérieures sont divisés en trois types :

– les forces appliquées globalement à l'objet, c'est le cas par exemple de la pesanteur,

– les forces appliquées à la surface de l'objet, comme par exemple les pressions de surface, et

3. L'objet reprend sa forme initiale instantanément, dès que les forces extérieures cessent de s'appliquer. C'est le comportement du modèle initial.

4. Une partie de l'énergie est absorbée pendant la déformation.

5. Dès que les forces exercées dépassent le seuil de plasticité de l'objet, il se déforme de façon permanente.

– les forces appliquées en des points précis, c'est le cas notamment des forces de collisions qui sont appliquées en certains points de la surface de l'objet.

Le travail de ces forces est respectivement noté W_g, W_s et W_p. La conservation de l'énergie, dans un milieu continu, s'écrit :

$$W_g + W_s + W_p = W_r \ , \tag{1.6}$$

où W_r est le travail des forces internes de réaction de l'objet en réponse aux actions extérieures.

Afin de pouvoir être utilisée, cette formule doit elle aussi être discrétisée. Les auteurs de [GTT89] ont choisi une approche par éléments finis. La conservation d'énergie 1.6 s'applique sur chaque élément e d'un assemblage formant l'objet :

$$W_g^e + W_s^e + W_p^e = W_r^e \ . \tag{1.7}$$

Chaque élément est une brique élémentaire à huit sommets. Les éléments sont connectés par les sommets. Chaque sommet à trois degrés de liberté (pas de rotation) et l'objet est composé de N_{el} briques élémentaires. Le respect de la conservation d'énergie amène au système de $n = 24N_{el}$ équations suivant ($\mathbf{R}$, $\mathbf{R}_i$, $\mathbf{G}$, $\mathbf{S}$, et $\mathbf{P}$ sont des vecteurs de n réels) :

$$\mathbf{R} + \mathbf{R}_i = \mathbf{G} + \mathbf{S} + \mathbf{P} \ , \tag{1.8}$$

où le travail des forces interne $\mathbf{R} + \mathbf{R}_i$ est décomposé en travail des forces initiales $\mathbf{R}_i$ et en travail des forces de déformation $\mathbf{R}$. Les vecteurs $\mathbf{G}$, $\mathbf{S}$ et $\mathbf{P}$ caractérisent respectivement les travaux des forces qui s'appliquent à tout l'objet, à la surface de l'objet et à des nœuds particuliers de l'objet. L'équation 1.8 est ré-écrite sous forme matricielle :

$$\mathbf{KU} = \mathbf{R} \ . \tag{1.9}$$

Cette dernière relation, où $\mathbf{K}$ est une matrice $n \times n$ de raideur, $\mathbf{U}$ est le vecteur (de taille n) des déplacements par rapport à l'état initial, et $\mathbf{R}$ le vecteur (de taille n) des forces appliquées à l'objet, n'est valable que pour des états d'équilibre du système. C'est-à-dire que les animations sont composées d'une suite d'états d'équilibre statiques (chacun nécessitant un calcul itératif), et non de l'évolution dynamique des objets.

L'équation 1.9 caractérise en fait un ressort tridimensionnel[6]. Les auteurs utilisent, pour guider leur animation, un squelette enrobé d'une couche de matière déformable. C'est le squelette qui impose la position des objets.

6. Cette équation est valable dans les états d'équilibre du système, c'est-à-dire les états où les forces externes et internes se compensent. Le vecteur $\mathbf{R}$ représente donc à la fois les forces interne et les forces externes (puisqu'elles sont identiques au signe près). Un objet qui exerce une force kx en réponse à la déformation x est un ressort.

L'équation 1.9 doit donc être utilisée, selon les nœuds, à la fois pour évaluer les forces exercées connaissant les positions et pour calculer les positions connaissant les forces exercées. Les vecteurs $\mathbf{U}$ et $\mathbf{R}$ sont subdivisés en deux parties, l'une (indexée par u pour *unknown*) contient les parties inconnues et l'autre (indexée par k pour *known*) contient les parties connues. L'équation 1.9 s'écrit :

$$\begin{bmatrix} \mathbf{K_{11}} & \mathbf{K_{12}} \\ \mathbf{K_{21}} & \mathbf{K_{22}} \end{bmatrix} \begin{bmatrix} \mathbf{U_u} \\ \mathbf{U_k} \end{bmatrix} = \begin{bmatrix} \mathbf{R_k} \\ \mathbf{R_u} \end{bmatrix} .$$

La résolution se fait en deux temps, les positions inconnues $\mathbf{U_u}$ sont tout d'abord calculées :

$$\mathbf{K_{11}U_u} = \mathbf{R_k} - \mathbf{K_{12}U_k} \ ,$$

puis vient le tour des forces $\mathbf{R_u}$:

$$\mathbf{R_u} = \mathbf{K_{21}U_u} - \mathbf{K_{22}U_k} \ .$$

Grâce à cette formulation, les caractéristiques de tous les objets simulés sont intégrées dans les mêmes matrices $\mathbf{K}$, $\mathbf{U}$ et $\mathbf{R}$.

Ce modèle est conçu pour modéliser des contacts prolongés entre objets. Cet objectif est atteint, avec la restriction que le seul comportement modélisé est l'élasticité linéaire. Par contre la grande richesse de forces externes applicable à l'objet (comme des pressions) est intéressante. Cependant, la conception même du modèle empêche tout tentative de l'appliquer à la simulation d'événements dynamiques tels que les collisions. En effet, seule une suite d'états d'équilibre est simulée, donc seule les déformations quasistatiques sont gérées.

À partir de la même équation de l'élasticité discrète 1.4 que celle utilisée par D. Terzopoulos et al. :

$$\mathbf{M}\frac{\partial^2 \mathsf{r}}{\partial t^2} + \mathbf{C}\frac{\partial \mathsf{r}}{\partial t} + \mathbf{K}(\mathsf{r})\mathsf{r} = \mathsf{f} \ ,$$

A. Pentland et J. Williams [PW89] proposent en 1989 une approche totalement différente. Dans les cas courants, les matrices de répartition de masse $\mathbf{M}$, d'amortissement $\mathbf{C}$ et de raideur $\mathbf{K}$ sont des matrices symétriques, définies positives. Généralement, l'amortissement utilisé est proportionnel à la masse : $\mathbf{C} = s\mathbf{M}$, où s est un scalaire. Dans ce cas, les trois matrices admettent une base commune de diagonalisation. En multipliant l'équation 1.4 par la transposée de la matrice de passage $\mathbf{P}$, une nouvelle relation ne comprenant que des matrices diagonales est obtenue :

$$\tilde{\mathbf{M}}\frac{\partial^2 \tilde{\mathsf{r}}}{\partial t^2} + \tilde{\mathbf{C}}\frac{\partial \tilde{\mathsf{r}}}{\partial t} + \tilde{\mathbf{K}}(\mathsf{r})\tilde{\mathsf{r}} = \tilde{\mathsf{f}} \ , \tag{1.10}$$

avec $x = \mathbf{P}\tilde{x}$, où $x \in \{\mathsf{r}, \mathbf{M}, \mathbf{C}, \mathbf{K}, \mathsf{f}\}$. Le système 1.10 est composé de $3n$ équations différentielles (l'objet est discrétisé en n points et $\tilde{\mathsf{r}}$ est le vecteur de coordonnées généralisé de ces points). Elles sont indépendantes (les matrices $\tilde{\mathbf{M}}$, $\tilde{\mathbf{C}}$ et $\tilde{\mathbf{K}}$ sont diagonales). Ces équations représentent l'évolution des modes vibratoires de l'objet. La i$^{\text{ème}}$ colonne de $\mathbf{P}$ représente le i$^{\text{ème}}$ mode vibratoire. La physique nous apprend que les modes de haute fréquence ne modifient que peu la forme globale d'un objet. A. Pentland et J. Williams proposent donc, dans une première approximation, de ne prendre en compte que les trois premiers modes vibratoires. Les objets ne pourront alors qu'avoir des déformations linéaires ou quadratiques. La suppression des déformations de haute fréquence permet (en plus de n'avoir à résoudre que 9 des $3n$ équations du système 1.10) d'utiliser un pas de temps important pour l'intégration de ces équations. D'autres approximations sont mises en œuvre. Par exemple, ayant remarqué que les déformations dues aux vibrations de basse fréquence (celles qui sont conservées) dépendent plus de la taille de l'objet que de sa forme, les auteurs proposent de calculer au préalable les déformations d'un solide rectangulaire. Ces déformations sont ensuite extrapolées à l'objet en fonction de la dimension de sa boite englobante.

Finalement, le modèle obtenu présente un coût en temps de calcul très faible qui permet une visualisation des objets en temps réel. Le prix à payer est pourtant élevé, puisque les diverses approximations faites mènent à une simulation tout de même restreinte de l'objet.

La démarche des trois modèles précédents est de simplifier suffisamment l'équation de l'élasticité pour pouvoir l'évaluer sur un modèle d'objet numérique. Le temps de calcul nécessaire à la simulation est tel que les auteurs ont petit à petit simplifié le modèle en restreignant la gamme de comportements modélisés[7]. De plus, aucun de ces modèles n'est capable de prendre en compte des vrais changements de topologie.

La section suivante présente une approche toute autre. Il n'est plus question de partir des équations physiques de l'élasticité, mais d'assembler des éléments (masses ponctuelles et ressorts en général) dont le comportement est facilement maîtrisable et simulable à faible coût, pour obtenir un comportement d'ensemble proche de celui des objets réels.

1.1.2 Approche nodale

Le principe fondateur de cette approche est le suivant : l'utilisation d'éléments simples inter-connectés permet de modéliser des comportements complexes. Par exemple, les serpents de G. Miller [Mil88] sont modélisés par un

7. Le premier modèle [TPBF87] est le seul capable, au prix d'un énorme temps de calcul, de simuler une l'élasticité non-linéaire. Le dernier modèle présenté [PW89], bien que basé sur la même équation, simule des comportement assez éloignés de la réalité.

assemblage de briques élémentaires. Chaque brique est composée de huit masses ponctuelles placées au sommet d'un cube. Des ressorts relient ces points. Un ressort est placé sur chaque arête du cube ainsi que sur les diagonales de ses faces. L'assemblage est réalisé en partageant quatre sommets entre deux cubes voisins. À chaque pas de simulation, il suffit, connaissant les positions relatives des masses connectées par un ressort, de calculer les forces produites et d'intégrer les équations de la dynamique.
Une autre approche consiste à doter chaque masse d'une loi qui régit ses interactions avec les autres masses. Cette approche est connue sous le nom de système de particules. Le chapitre 2 lui est consacré.

Certains modèles discrets d'objet déformable auront une description plus détaillée au cours de ce document. Le lecteur peut notamment se référer au chapitre 2.2 qui contient une description du modèle mis au point à l'ACROE (page 25), du modèle de système de particules de G. Miller et A. Pearce (page 27) et du matériau capable de fusion de D. Terzopoulos et al. (page 28). Enfin, le chapitre 7 contient, entre autres, une brève description du réseau masses/ressorts de K. Waters (page 103) et du modèle de personnage de J. Chadwick et al. (page 103).
Un autre modèle d'objets déformable, celui mis au point par M.-P. Gascuel [Gas93], ainsi qu'une variante [DG95] sont présentés dans la partie B du chapitre 4 (page 67).

Comme nous l'avons vu pendant cette brève description des modèles existants, la simulation des objets est basée sur l'application d'au moins une loi de la physique, celle qui relie les mouvements d'un objet (que ce soit un point ou un solide) aux forces qui lui sont appliquées. Cette loi est appelée *principe fondamental de la dynamique*. La section suivante présente ce principe ainsi que les technique à mettre en œuvre pour pouvoir l'utiliser dans l'univers essentiellement discret des ordinateurs.

1.2 Principes fondamentaux de la dynamique

Le travail présenté dans cette thèse concerne l'animation par modèles physiques. Les objets animés sont dotés de certaines caractéristiques et leurs mouvements sont calculés en appliquant les lois appropriées de la mécanique. Ce chapitre décrit les lois physiques utilisées pour ce calcul de mouvements [AF77, Pér95].

En mécanique classique, le mouvement des objets est décrit par un principe fondamental :

$$\frac{d[P_O]}{dt} = [F_{O,ext}] \ , \tag{1.11}$$

où $[P_O]$ est le torseur cinétique de l'objet exprimé au point O fixe dans le repère galiléen considéré et $[F_{O,ext}]$ est le torseur des forces extérieures

appliquées à l'objet, exprimé en O.
Dans les paragraphes suivants, ce principe est étudié dans les deux cas les plus simples : le cas où l'objet est un point matériel et le cas où l'objet est un solide.

1.2.1 Dynamique du point

Dans le cas où l'objet considéré se réduit à un point doté d'une masse m, les forces appliquées le sont en ce point. Il n'y a pas de notion de rotation pour une masse ponctuelle. Le torseur cinétique se réduit au vecteur quantité de mouvement et le torseur des forces se réduit à la somme des forces. Le principe fondamental de la dynamique s'écrit donc :

$$\frac{d\vec{P}(t)}{dt} = \sum \vec{f}_{ext}(t) \ , \tag{1.12}$$

où $\vec{P}(t) = m\vec{v}(t)$ est la quantité de mouvement du point considéré et $\sum \vec{f}_{ext}(\mathrm{t})$ la somme des forces exercées sur ce point.

Les objets considérés ont une masse constante. En notant $\vec{x}(t)$ la position du point à l'instant t, l'équation 1.12 s'écrit :

$$\begin{cases} \sum \vec{f}_{ext}(t) = m\dfrac{dv(t)}{dt} \\ \vec{v}(t) = \dfrac{dx(t)}{dt} \ . \end{cases}$$

Le problème étant de calculer la position à partir des forces, le système est réécrit :

$$\begin{cases} \vec{v}(t) = \dfrac{1}{m}\displaystyle\int_{t_0}^{t} \sum \vec{f}_{ext}(u)du \\ \vec{x}(t) = \displaystyle\int_{t_0}^{t} \vec{v}(u)du \ , \end{cases}$$

t_0 étant la date du début de la simulation. Une animation se fait en échantillonnant le temps et en calculant une image pour chaque échantillon de temps (typiquement pour chaque $1/25^{\text{ème}}$ de seconde). Il nous faut connaître l'état des objets à chaque échantillon de temps Δt. Il est donc plus intéressant de calculer les caractéristiques à l'intant $t + \Delta t$ en fonction de leur valeur à l'instant t :

$$\begin{cases} \vec{v}(t+\Delta t) = \vec{v}(t) + \dfrac{1}{m}\displaystyle\int_{t}^{t+\Delta t} \sum \vec{f}_{ext}(u)du \\ \vec{x}(t+\Delta t) = \vec{x}(t) + \displaystyle\int_{t}^{t+\Delta t} \vec{v}(u)du \ . \end{cases} \tag{1.13}$$

1.2.2 Dynamique du solide

Pour la dynamique classique, un solide est supposé indéformable, et donc par définition, un solide est un système matériel tel que la distance entre deux de ses points, quels qu'ils soient, reste constante au cours du temps.
En exprimant les torseurs au centre de gravité G du solide, le principe fondamental de la dynamique (éq. 1.11) se simplifie :

$$\frac{d\vec{P}}{dt} = \sum \vec{f}_{ext} \ , \tag{1.14}$$

$$\frac{d\vec{L}_G}{dt} = \vec{\mathcal{M}}_{G,ext} \ , \tag{1.15}$$

où $\vec{P} = m\vec{v}_G$ est la quantité de mouvement du centre de gravité du solide, $\vec{L}_G$ le moment cinétique du solide et $\vec{\mathcal{M}}_{G,ext}$ le moment des forces extérieures appliquées au solide, exprimés par rapport à son centre de gravité.
Le centre de gravité du solide se comporte comme un point matériel (éq. 1.12 $\Leftrightarrow$ éq. 1.14). Nous nous intéressons au mouvement du solide autour de son centre de gravité (éq. 1.15). Le moment des forces extérieures $\vec{\mathcal{M}}_{G,ext}$ se calcule à partir des forces extérieures appliquées $\vec{f}_i$ et de leurs points d'application A_i :

$$\vec{\mathcal{M}}_{G,ext} = \sum \vec{f}_i \wedge \vec{GA_i}$$

Le moment d'inertie $\vec{L}_G$ exprimé au centre de gravité est égal au produit de la matrice d'inertie $\mathbf{J}_G$ (calculée par rapport à ce même centre de gravité) par la vitesse angulaire $\vec{\omega}$:

$$\vec{L}_G = \mathbf{J}_G \vec{\omega} \ .$$

Cette formule est à utiliser avec précaution. En effet, la vitesse angulaire $\vec{\omega}$ doit être exprimée dans le même repère centré sur le centre de gravité que la matrice d'inertie $\mathbf{J}_G$.

Remarque : pour une sphère homogène de rayon R et de masse m,

$$\mathbf{J}_G = \frac{2mR^2}{5} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

et l'équation 1.15 se simplifie :

$$\frac{2mR^2}{5}\dot{\vec{\omega}} = \vec{\mathcal{M}}_{G,ext} \ .$$

Les moments d'inertie des primitives homogènes telles que les disques, les cylindres, les cubes, ... etc. se trouvent aisément dans la littérature [AF77, Pér95]. Pour des objets à géométrie plus complexe et/ou non homogène, il faut intégrer la répartition de la masse :

$$\mathbf{J}_G = \begin{bmatrix} \int \rho(y^2+z^2)dV & -\int \rho xydV & -\int \rho xzdV \\ -\int \rho xydV & \int \rho(x^2+z^2)dV & -\int \rho yzdV \\ -\int \rho xzdV & -\int \rho yzdV & \int \rho(x^2+y^2)dV \end{bmatrix}$$

où ρ est la densité de l'élément de volume dV, x, y et z sont les coordonnées de dV dans le repère centré en G dont l'orientation est choisie en fonction des propriétés de symétrie de l'objet.

En considérant la matrice d'inertie indépendante du temps (ce qui est vrai lorsqu'elle est exprimée par rapport à un repère lié au solide, et que le solide ne subit ni déformation ni changement de masse), l'équation 1.15 devient :

$$\frac{d(\mathbf{J}_G\vec{\omega})}{dt} = \vec{\mathcal{M}}_{G,ext} \ ,$$

$$\vec{\omega} \wedge \mathbf{J}_G\vec{\omega} + \mathbf{J}_G\dot{\vec{\omega}} \approx \vec{\mathcal{M}}_{G,ext} \ . \tag{1.16}$$

Formellement, il y a un terme en $\dot{\mathbf{J}}_G$ dans le membre de gauche qui est habituellement négligé. Il est difficile d'aller plus loin tout en restant général. En effet les rotations n'ont pas les propriétés mathématiques des translations. Passer de l'accélération angulaire $\dot{\vec{\omega}}$ à la vitesse de rotation $\vec{\omega}$ puis à l'orientation nécessite de faire des approximations dépendant à la fois de la représentation choisie pour les caractéristiques d'orientation (matrice, vecteur, quaternions unitaires) et du schéma d'intégration.

Nous reviendrons sur ce point après avoir exposé notre choix de représentation pour les rotations.

Représentation de l'orientation d'un solide

De même que la position d'un solide est donnée par une translation par rapport à un repère donné, son orientation est définie par une rotation à partir d'une orientation de référence. Trois possibilités sont couramment employées en animation par ordinateur :

- les matrices de rotation $\mathbf{R}$,
- les vecteurs $\vec{r}$, et
- les quaternions unitaires $\mathtt{q}$.

Il est aisé de passer de l'une de ces représentation à l'autre. Une rotation d'angle φ autour d'un axe de vecteur unitaire $\vec{n}(x, y, z)$ s'écrit :

$$\begin{aligned}
\mathbf{R} &= \begin{pmatrix} x^2(1-\cos\varphi)+\cos\varphi & xy(1-\cos\varphi)-z\sin\varphi & xz(1-\cos\varphi)+ysin\varphi \\ xy(1-\cos\varphi)+z\sin\varphi & y^2(1-\cos\varphi)+\cos\varphi & yz(1-\cos\varphi)-x\sin\varphi \\ xz(1-\cos\varphi)-y\sin\varphi & yz(1-\cos\varphi)+x\sin\varphi & z^2(1-\cos\varphi)+\cos\varphi \end{pmatrix}, \\
\vec{r} &= \varphi\vec{n}\ , \\
\mathtt{q} &= \cos\frac{\varphi}{2} + \sin\frac{\varphi}{2}\vec{n}\ .
\end{aligned}$$

Les quaternions (Annexe B) présentent des propriétés intéressantes :

- la composition de rotation se fait simplement (contrairement aux vecteurs).
- Les coûts en mémoire et en temps de calcul sont réduits (contrairement aux matrices).
- L'application d'une rotation à un vecteur est aisée.

Nous choisissons donc d'utiliser les quaternions pour représenter les orientations.

Intégration des rotations

L'équation 1.16 nous permet d'obtenir la valeur de l'accélération angulaire $\dot{\vec{\omega}}$ (sous forme vectorielle), à l'instant $t + \Delta t$. Pour calculer la vitesse de rotation $\vec{\omega}$ (à laquelle on fait correspondre le quaternion de rotation $\dot{\mathtt{q}}$i) puis l'orientation $\mathtt{q}$ à l'instant $t + \Delta t$ connaissant leurs valeurs à l'instant t, il faut intégrer respectivement l'accélération et la vitesse de rotation pendant l'intervalle de temps Δt.
Lorsqu'une représentation vectorielle est choisie, l'hypothèse est faite que les rotations restent petites et que la somme de deux vecteurs de rotation donne une approximation de la composition des rotations. L'intégrale est alors estimée avec des méthodes classiques.
Les quaternions permettent de calculer exactement la composition de rotation, mais par multiplication et non par addition. Dans ce contexte, la notion même d'intégrale n'est plus très claire. L'utilisation des logarithmes

et des exponentielles de quaternions (Annexe B) nous donne une issue. Nous pouvons écrire [Han93] :

$$\begin{aligned}
\dot{\mathsf{q}}_{t+\Delta t} &= \dot{\mathsf{q}}_t \star e^{\vec{\omega}\,dt} \ , \\
\mathsf{q}_{t+\Delta t} &= \mathsf{q}_t \star (\dot{\mathsf{q}}_{t+\Delta t})^{dt} \\
&= \mathsf{q}_t \star e^{\log(\dot{\mathsf{q}}_{t+\Delta t})\,dt} \ .
\end{aligned}$$

Cette introduction a été l'occasion de préciser les lois de la mécanique utilisées pour l'animation par modèles générateurs et de les présenter sous une forme directement utilisable pour l'informatique graphique.
Nous avons également survolé les méthode classique de modélisation et de simulation d'objets déformables. Le chapitre suivant présente de façon plus approfondie le concept de système de particules à travers son histoire en informatique graphique et ses utilisations actuelles.

Chapitre 2

Systèmes de particules

2.1 Introduction

D'un point de vue général, un ensemble de points munis de lois de mouvement peut être considéré comme un système de particules. Dans ce document, nous nous intéressons essentiellement aux systèmes dynamiques. Les définitions suivantes sont données dans ce contexte.

2.1.1 Définitions

Considérons un ensemble de masses ponctuelles. Ces masses soumises aux actions du monde extérieur se déplacent. Cet ensemble forme le système de particules le plus simple possible. En général, les actions extérieures sont représentées par des forces, et les déplacements sont calculés en appliquant les lois fondamentales de la dynamique.

La première amélioration consiste à introduire des lois d'interaction entre les particules qui sont alors dotées d'un rayon d'influence. Elles n'agissent que sur les particules se trouvant dans leur zone d'influence. Selon le phénomène simulé, ces lois peuvent se réduire à une simple répulsion (le comportement obtenu sera alors proche de celui de grains de sable). Plus classiquement, une loi d'attraction/répulsion est employée. Cette interaction engendre une répulsion à courte portée et une attraction à longue portée. Son rôle est d'assurer la cohésion de l'ensemble de particules tout en le forçant à occuper un volume non nul.
Au cours du temps, un tel système évolue naturellement vers un état d'énergie minimale. Cette configuration de repos du système est obtenue lorsque les forces extérieures et intérieures se compensent.

2.1.2 Système de particules et réseaux masses/ressorts

Dans un système de particules, chaque masse élémentaire est dotée d'une seule loi d'interaction qui régit ses interactions avec toutes les autres particules. Dans le cas contraire où l'interaction est spécifiée pour chaque couple de particules, nous considérons le système comme un réseau masses/ressorts. Cette distinction est faite sur le mode de spécification de l'interaction et non sur son type : un système de particules peut très bien mettre en œuvre un comportement de type ressort (forces de rappel proportionnelles à la distance entre les masses), et un réseau masses/ressorts peu très bien modéliser un comportement tout autre [1]. Cette distinction entre les deux modes de calculs des interactions entre les masses entraîne deux approches algorithmiques différentes qui vont être précisées ci-dessous. Chacune à ses points faibles et ses points forts.

Changement de topologie

Par définition, la topologie n'est pas fixée dans un système de particules. Cela en est l'intérêt fondamental. Chaque particule interagit potentiellement avec toutes les autres. Le système peut se rompre en plusieurs morceaux et/ou se recoller pendant le déroulement normal de l'algorithme.
Dans le cas d'un système masses/ressorts, il faut, pour gérer les fractures, introduire un traitement spécial qui va détruire un ressort donné en fonction de critères donnés (comme la distance entre les deux masses reliées par le ressort). Ce calcul n'est pas très coûteux. Il suffit d'effectuer un test lors de la simulation du ressort et d'être capable de gérer la suppression des ressorts. Par contre, la gestion des recollements est beaucoup plus coûteuse. Il faut, avant de créer un ressort, tester tous les couples de masses pour sélectionner ceux qui sont sujets au recollement.

Coût de la simulation des interactions

Une particule est, a priori, en interaction avec toutes les autres, alors qu'un nœud d'un réseau masses/ressorts n'est en interaction qu'avec les nœuds auxquels il est connecté. Une approche non optimisée conduit donc à un coût $O(n)$ en mémoire et $O(n^2)$ en temps de calcul pour un système de n particules (une interaction par particule et un calcul d'interaction entre chaque couple de particules). Pour un réseau masses/ressorts (n masses, m ressorts, $m \geq n-1$ pour un réseau connexe [2]), les coûts sont $O(m)$ en place mémoire comme en temps de calcul. Pour un réseau complet, $m = n(n-1) = O(n^2)$ et le coût en temps comme en place mémoire devient $O(n^2)$.
En général, les interactions utilisées dans les systèmes de particules ont une

1. Dans ce cas, nous parlons de systèmes masses/ressorts *généralisés*.

2. Le cas $n = m-1$ correspond à un réseau formant une chaîne (maximum deux voisins) de masses, ce qui n'est pas très intéressant. Dans un modèle classique [TPF89, Wat92, Via92], l'objet déformable est composé de mailles élémentaires (8 masses, une sur chaque sommet d'un cube, entièrement connectées par des ressorts) assemblées entres elles. Dans ce cas, $m = 13n$.

portée limitée. Ceci permet de mettre en œuvre des techniques d'optimisation évidentes (comme la subdivision spatiale [Sam90]), et de diminuer ainsi le coût en temps de calcul à $O(n)$ en moyenne.

En conclusion, les systèmes de particules sont moins coûteux en place mémoire, mais plus coûteux en temps de calcul (si aucune optimisation n'est faite) que les réseaux masses/ressorts. Un système de particules optimisé n'a pas une plus grande complexité en temps de calcul en moyenne qu'un réseau masses/ressorts.

Validité physique

Cette économie a cependant une contre-partie. Dans un réseau masses/ressorts, la loi d'interaction I_{AB} est définie pour le couple de masses en interaction et s'applique à chacune d'elles en respectant le principe de l'action et de la réaction. Par contre, il n'y a pas de raison pour que deux particules A et B aient les mêmes lois d'interactions I_A et I_B. Le respect du principe de l'action/réaction nécessite un minimum d'attention. Pratiquement, les interactions I_A et I_B sont considérées comme deux connexions de type ressort : chaque interaction est appliquée aux deux particules. En conséquence, il est difficile de contrôler finement la description de l'objet modélisé. Le fait que les voisinages ne soient pas fixés renforce encore la difficulté.

Un système de particules est donc plutôt plus difficile à manipuler qu'un réseau masses/ressorts.

La frontière entre systèmes de particules et réseau masses/ressorts n'est cependant pas imperméable. Par exemple, suivant les définitions données ci-dessus, le système *CORDIS-ANIMA* [Jim93] est un réseau masses/ressorts. Il peut cependant être utilisé comme un système de particules grâce à des liaisons conditionnelles, liaisons qui ne sont actives que lorsque les masses sont suffisamment proches.

2.2 Historique des systèmes de particules

Depuis leur première utilisation en informatique graphique en 1983 [Ree83], les systèmes de particules ont connu un succès grandissant. Alliant simplicité et puissance, les particules apportent une réponse à de nombreux problèmes, de la modélisation du feu à la simulation d'objets déformables. Créés pour combler les lacunes de la géométrie surfacique classique, ils sont aujourd'hui en compétition avec la géométrie fractale pour la modélisation de paysages forestiers. Alors que les premiers modèles ne mettaient en œuvre qu'une grossière approximation des lois de la dynamique, ils sont aujourd'hui utilisés pour la simulation physique d'objets déformables et de fluides. On trouve des particules dans la modélisation de comportements d'ensembles aussi bien que dans la reconstruction de surfaces à partir de nuages de points. Cette section propose un bref survol de cette évolution et des divers domaines d'application des systèmes de particules. La section 2.2.4 est consacrée à une étude plus détaillée du concept de particule orientée, mis au point par R. Szeliski et D. Tonnesen [ST92], et qui est à la base de notre travail.

2.2.1 Le commencement : les systèmes non interagissants

C'est W. Reeves [Ree83] qui présente, en 1983, le premier système de particules pour l'informatique graphique. Il crée cette méthode de modélisation pour combler une lacune des méthodes surfaciques classiques : la modélisation d'objets dont les contours ne peuvent être définis précisément ("*fuzzy objects*") tels que le feu et les explosions, les nuages, une cascade, de l'herbe, ...etc.

L'idée est de voir un objet non plus comme un ensemble de primitives géométriques définissant sa surface, mais comme un nuage de particules élémentaires définissant son volume. Ce nuage évolue dans le temps. Les particules naissent, changent (de forme, de position, ...) et meurent. La forme de l'objet n'est pas complètement déterminée. La vie des particules est gérée par des processus stochastiques. La puissance de ce principe de modélisation réside entre autres dans la simplicité de l'élément de base (particule sphérique ou ponctuelle) qui permet d'en traiter un grand nombre, et facilite le *motion blur*[3]. D'autre part, l'utilisation de processus stochastiques permet d'obtenir des objets de grande complexité tout en ne spécifiant qu'un minimum de données.

Une scène est, en général, composée de plusieurs systèmes de particules (400 systèmes contenant au total 75 000 particules pour une scène où la caméra est engloutie dans un mur de feu[4]). Les valeurs initiales des paramètres caractérisant chacune des particules sont déterminées aléatoirement dans un domaine caractéristique du système de particules. Vitesse, position mais aussi couleur, durée de vie, forme, taille, . . . etc., sont les attributs de chaque particule. Ils évoluent pendant la vie de la particule. Dans notre présentation de ce modèle, nous ne nous intéressons qu'à la gestion des vitesses et des positions. D'autres aspects tels que l'évolution des couleurs n'apporte rien à la compréhension des mécanismes de base mis en œuvre pour simuler le système de particules. La position d'une particule dans une image est donnée par sa position dans l'image précédente translatée du vecteur vitesse. Pour augmenter le réalisme, une accélération globale simulant la gravité est ajoutée à la vitesse à chaque étape de calcul. Bien que cela n'apparaisse pas

3. Flou créé par le déplacement de l'objet pendant le temps d'ouverture du diaphragme de la caméra. L'existence de ce flou dans une animation (montrant des objets ayant une grande vitesse) accentue grandement l'impression de réalisme.

4. Film *Star Trek II : the wrath of Khan* - Scène *Genesis Demo* - Juin 1982 - Paramount.

explicitement dans [Ree83], ces lois de variation de la vitesse au cours du temps sont en fait une approximation très grossière des lois fondamentales de la dynamique du point.

W. Reeves et R. Blau [RB85] ont adapté ce système pour la modélisation de paysages forestiers. L'idée est qu'en conservant la trace d'une particule durant toute sa vie, il est possible d'obtenir un brin d'herbe, et, qu'avec un système de particules dont les particules sont elles mêmes des systèmes de particules, il est possible de modéliser le détail des embranchements successifs dans un arbre.

Une des difficultés majeures soulevée par cet article est le rendu. En effet, pour l'exemple des explosions ci-dessus, les particules étaient évidemment émettrices de lumière. Il suffisait donc, pour produire une image, de parcourir la liste des particules en ajoutant pour chacune sa contribution au pixel dans lequel elle se projette. Pour des particules non émettrices, le problème devient singulièrement plus complexe. Les particules ne sont visibles que parce qu'elles réfléchissent la lumière issue des sources lumineuses. Même en ne prenant en compte que l'éclairage direct (sans tenir compte des réflexions successives de la lumière) le problème ne peut être résolu par un lancer de rayon classique. Vu le nombre de particules dans une image, le temps de calcul de visibilité est prohibitif. La méthode proposée n'est donc pas exacte. Le lancer de rayon prend en compte une probabilité d'intersection entre le rayon lumineux et la particule, fonction de la place de la particule concernée dans l'objet qu'elle définit et de la place de l'objet dans la scène.

Dans ce système, le calcul de la position de chaque particule à chaque instant repose sur une approximation des lois de la dynamique, approximation grossière certes, mais qui apporte tout de même un certain réalisme aux mouvements calculés. Une grosse lacune limite cependant le champ d'application de cette technique : les particules de ce premier système n'interagissent pas entre elles. De ce fait, aucune collision n'est détectée, et chaque particule évolue indépendamment des autres.

2.2.2 Évolution vers des systèmes dynamiques

G. Miller et A. Pearce ont les premiers utilisé un système de particules pour simuler des fluides [MP89]. Le mouvement est calculé en appliquant une approximation des équations de la dynamique du point, au détail près que les *globules* (nom donné aux particules) n'ont pas d'inertie :

$$\vec{x}_p = \frac{1}{m_p} \int \int \sum_{i=1}^{N} \vec{f}(P_p, P_i) dt dt$$

où $\vec{x_p}$ et m_p sont respectivement la position et la masse de la particule P_p et $\vec{f}(P_i, P_j)$ la force d'attraction/répulsion entre les particule P_i et P_j. En notant $\vec{P}_{ij} = \vec{x}_j - \vec{x}_i$ la position relative des particules P_i et P_j, et $\vec{V}_{ij}$ leur vitesse relative, cette force s'exprime comme suit :

$$\vec{f}(P_i, P_j) = \begin{cases} \vec{P}_{ij}[s_r(\frac{b_1}{\|\vec{P}_{ij}\|^m} - \frac{b_2}{\|\vec{P}_{ij}\|^n}) - s_d\frac{\vec{V}_{ij} \cdot \vec{P}_{ij}}{\|\vec{P}_{ij}\|^2}] & \text{si } i \neq j \ , \\ 0 & \text{sinon .} \end{cases}$$

Dans cette interaction sont regroupés deux termes dont l'influence respective est modulée par les coefficients s_r et s_d. Bien que celà ne soit pas spécifié dans l'article, il est possible de reconnaître dans le premier terme une interaction de type Lennard-Jones et dans le second terme une interaction de frottement.

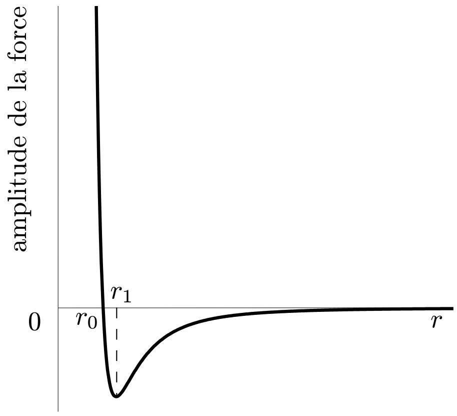
Mis au point pour la simulation des interactions entre les molécules de gaz, l'interaction d'attraction/répulsion de Lennard-Jones [AF77] f_{LJ} dérive du potentiel Φ_{LJ} :

$$\begin{aligned} \Phi_{LJ}(r) &= \frac{B}{r^n} - \frac{A}{r^m} \ , \\ \vec{\mathcal{F}}_{LJ}(\vec{r}) &= -\vec{\mathrm{grad}}(\Phi_{LJ}(r)) \ , \\ \vec{\mathcal{F}}_{LJ}(\vec{r}) &= f_{LJ}(r)\frac{\vec{r}}{\|r\|} \ , \\ f_{LJ}(r) &= \frac{nB}{r^{n+1}} - \frac{mA}{r^{m+1}} \ . \end{aligned}$$

Le comportement de ce type de fonction répond aux exigences de la modélisation d'une force d'attraction/répulsion (Figure 2.1) : lorsque la distance $\|\vec{r}\|$ entre deux particules tombe au dessous d'un certain seuil r_0, une force de répulsion s'exerce sur les deux particules ; lorsque les particules s'éloignent et que la distance devient supérieure à r_0, une force d'attraction prend le relais. Si les particules sont très éloignées, la force d'attraction devient négligeable, et les particules sont libres.

En jouant sur les paramètres de ce modèle (s_r, s_d, b_1 et b_2), les auteurs arrivent à simuler différents comportements allant de la pâte à modeler au sable, en passant par le fluide. En définissant une *température* qui sert à calculer les valeurs des paramètres précédemment cités, ce modèle est capable de simuler un objet passant continûment de l'un à l'autre des différents comportements.

En approfondissant cette idée de changement de comportement en fonction de la température de l'objet, D. Terzopoulos, J. Platt et K. Fleisher proposent dans [TPF89] un modèle d'objet capable de fondre. Ce modèle est hybride entre système de particules et réseau masses/ressorts : lorsque la température est basse, l'objet est solide et les liaisons entre les particules sont modélisées par des ressorts généralisés ; dès que la température dépasse un seuil (température de fusion), l'objet fond et devient fluide. Les masses



$$\vec{\mathcal{F}}_{LJ}(\vec{r}) = f(r)\frac{\vec{r}}{\|\vec{r}\|}$$

FIG. 2.1 - *Force de Lennard Jones*

sont alors dotées d'une loi d'interaction de type système de particules. À l'état solide, les objets sont constitués d'un assemblage de mailles hexaédriques élémentaires. Chacune de ces mailles est formée par huit particules disposées aux sommets d'un cube. Des ressorts couplent toutes les particules d'une même face. Des éléments de conduction thermique, permettant de diffuser la chaleur à l'intérieur de l'objet, sont installés parallèlement à ces ressorts. La propagation de la chaleur est faite en accord avec les lois physiques régissant ce phénomène.
En fonction de la moyenne des températures θ des deux masses qu'il relie, la raideur[5] K_θ du ressort change :

$$K_\theta = \begin{cases} K_0 & \text{si } \theta \leq \theta_s \ , \\ K_0 - \nu(\theta - \theta_s) & \text{si } \theta_s < \theta < \theta_m \ , \\ 0 & \text{si } \theta \geq \theta_m \ . \end{cases}$$

θ_s est appelée seuil thermo-élastique. Passé ce seuil, la raideur de l'objet diminue pour devenir nulle lorsque la température atteint le seuil de fusion θ_m. Une loi d'interaction du type Lennard-Jones prend alors le relais.
Les mouvements sont calculés en intégrant les équations de la dynamique du point avec un schéma d'Euler (Annexe A).

Bien que ce soit essentiellement un réseau masses/ressorts, nous présentons ici le système *CORDIS-ANIMA* [LJF+91, Jim93], mis au point à l'ACROE (Grenoble). Ce système mérite le nom de simulateur physique. En effet, les objets simulés peuvent être appréhendés non seulement de façon

5. Nous rappelons qu'un ressort de raideur K et de longueur à vide l_0 exerce la force $f = -K(l - l_0)$ lorsque que sa longueur est l.

visuelle, en regardant leur évolution, mais aussi de façon auditive (les vibrations des objets peuvent être transmises à un haut-parleur) ou encore kinesthésique grâce au *TGR* (*Transducteur Gestuel Rétro-actif*) mis au point par ce même laboratoire. Reposant sur un calculateur parallèle qui permet un fonctionnement en temps réel, ce modeleur-simulateur est aussi bien utilisé pour la simulation sonore de cordes vibrantes que pour l'animation par modèles physiques. L'objectif avoué est d'arriver à définir une physique algorithmique.
Un objet est composé de masses élémentaires reliées par des *liaisons* (ressorts généralisés). Différents types de liaisons (ou *ressorts*) sont implémentés. Certains sont classiques, comme par exemple ceux modélisant une élasticité (linéaire ou non) ou une attraction/répulsion [6]. D'autres le sont moins, comme ceux modélisant un comportement plastique ou un frottement solide. Ces comportements complexes sont obtenus en modélisant le ressort par un automate d'état finis. En fonction de son histoire, il se trouve dans tel ou tel état et suit telle ou telle loi.

2.2.3 Systèmes de modélisation de mouvements

D. Haumann [WH91, HWA+92] s'est intéressé au contrôle du mouvement des particules. L'exemple choisi est celui des feuilles mortes virevoltant dans le vent. Les particules utilisées sont dynamiques et soumises à différentes actions extérieures. Le vent est simulé par la composition de quatre champs de vitesses bidimensionnels élémentaires. Ces champs sont :

- *uniform* : uniforme. Ce champ applique une translation aux particules. Il est composé de vecteurs colinéaires.
- *sink* : radial convergent. Les particules soumises à ce champ se concentrent vers un point, centre de ce champ.
- *source* : radial divergent. Les particules soumises à ce champ fuient son centre.
- *vortex* : circulaire. Il fait tourner les particules autour de son centre.

Des comportements complexes peuvent être obtenus en additionnant ces champs. Par exemple, placer un champ *source* et un champ *vortex* horizontaux et de même centre dans une région où s'exerce un champ *uniform* vertical de bas en haut simule le tourbillon créé par une tornade. Afin de tester si le positionnement des champs dans l'espace produit le résultat souhaité, une simulation est effectuée en introduisant dans la scène un ensemble de particules classiques dotées d'un frottement visqueux. Leur mouvement permet de réévaluer la définition des champs. L'emploi de ces particules

6. Le modèle d'attraction/répulsion est une approximation linéaire par morceaux des forces de Lennard-Jones

simples ne permet pas de calculer les mouvements de rotations des feuilles. C'est en couplant les particules par des ressorts que toute la généralité du mouvement est obtenue. Une feuille est un petit réseau masses/ressorts.

Une telle approche multi-échelles pour la définition de scénarios d'animation est intéressante. Dans un contexte de simulation dynamique, la spécification de scénario est un réel problème. Ceci est dû au manque de contrôle de l'utilisateur. Définir les grandes lignes de la séquence en utilisant un modèle simplifié des objets mis en œuvre permet de gagner du temps. Nous utiliserons ce principe lors de la mise au point de nos propres simulations.

2.2.4 Systèmes de modélisation de formes : particules orientées

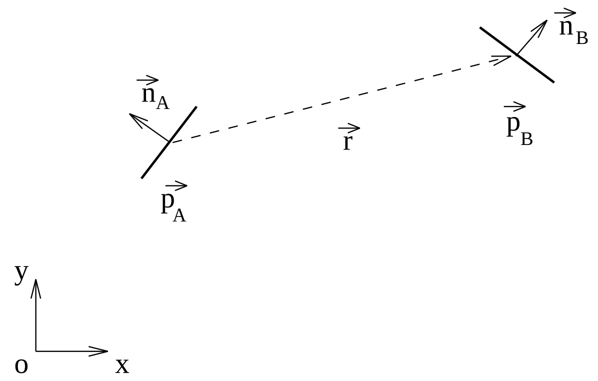
Les systèmes de particules, en tant qu'outils de modélisation de formes, ont franchi une grande étape avec l'introduction des particules orientées [ST92]. Pour R. Szeliski et D. Tonnesen, l'élément de base du système de modélisation n'est plus isotrope. La particule peut enfin servir à autre chose qu'à modéliser des amas ou des fluides. Cette approche est décrite dans la section suivante.

2.3 Système de particules orientées

R. Szeliski et D. Tonnesen [ST92] présentent le concept de particule orientée ainsi qu'une de ses applications : un outil de modelage surfacique d'objets tridimensionnels (3D). Le système de particules est utilisé pour créer des formes 3D. On peut rapprocher ceci de la sculpture virtuelle. L'utilisateur dispose d'outils avec lesquels il agit sur la forme qu'il crée. Un algorithme de génération de particules permet l'extension automatique de la surface. Un mécanisme de gel/dégel des particules (dégel des particules proches de l'outil, gel des autres) permet d'élaborer des formes complexes. Les auteurs présentent aussi une fonctionnalité de reconstruction de surface, à partir d'un nuage de points, par interpolation. Les surfaces reconstruites sont modélisées l'ensemble des particules orientées, c'est-à-dire un ensemble de points et de normales.

En plus des classiques masse et position, chaque particule est caractérisée par une orientation (Figure 2.2). Chaque particule peut être vue comme un élément de surface, défini par une position dans l'espace et une normale[7]. L'élément de base n'est plus ponctuel (du point de vue de ses interactions avec le monde, sa forme géométrique n'a pas d'importance), les forces d'interaction ne sont plus isotropes. Le système est régi par les lois de

7. Dans le reste de ce document, nous parlerons de *plan tangent* à la particule pour désigner le plan passant par le point occupé par la particule et perpendiculaire à sa normale.



$$\vec{r} = \vec{p_B} - \vec{p_A}$$

FIG. 2.2 - *Particules orientées - Définitions*

la dynamique du solide :

$$\begin{aligned}\sum F &= m\gamma \ , \\ \sum \dot{\omega} &= J^{-1}\tau \ .\end{aligned}$$

La loi d'interaction entre deux particules est une somme pondérée de quatre types d'interaction : l'interaction d'attraction/répulsion, et les interactions coplanaire, cosphérique et co-normale. Chacune de ces lois, composée d'une force et d'un moment de torsion, agit sur les particules pour les rapprocher d'une configuration donnée : la force d'attraction/répulsion fixe la distance moyenne entre deux particules, l'interaction coplanaire tend à positionner les particules dans le même plan, l'interaction co-normale fait tourner les particules de sorte à ce que leurs normales soient coplanaires, et enfin l'interaction cosphérique positionne les particules sur une sphère non spécifié. Les forces et moments utilisées dérivent de potentiels.

2.3.1 Force d'attraction/répulsion

Les forces d'attraction/répulsion sont des forces isotropes : elle ne dépendent que de la distance entre les particules. Le moment de torsion produit est nul. Une interaction d'attraction/répulsion se réduit donc à une force du type $\vec{\mathcal{F}}(\vec{r}) = f(\|\vec{r}\|)\frac{\vec{r}}{\|\vec{r}\|}$, $\vec{r}$ étant le vecteur de translation permettant de passer de la particule A à la particule B (Figure 2.2). Par abus de notation, nous confondons la distance entre deux particules r et la translation qui permet de passer de l'une à l'autre $\|\vec{r}\|$, ainsi que la force exercée $\vec{\mathcal{F}}(\vec{r})$ et la mesure algébrique de cette force $f(r)$.
Comme dans la plupart des systèmes de particules, D. Tonnesen et R.

Szeliski utilisent une force du type force d'interaction inter-moléculaire de Lennard-Jones (voir page 28).

2.3.2 Forces de forme

La forme des objets dépend de trois types d'interactions. Toutes sont construites sur le même modèle. Le potentiel dont elles dérivent peut être décomposé en deux facteurs : l'un est fonction des positions et orientations relatives des particules, l'autre ne dépend que de la distance entre elles et ne sert qu'à en limiter la portée. Le premier facteur est une fonction positive basée sur des propriétés géométriques des positions et orientations relatives souhaitées. L'expression du potentiel émis par la particule A en B est donnée par :

– Force coplanaire :

$$\Phi_P = (\vec{n_A} \cdot \vec{r})^2 \psi(\|\vec{r}\|) \ .$$

La particule B est dans le plan tangent à A si et seulement si $\vec{n_a}$ et $\vec{r}$ sont orthogonaux. Le moment de torsion $\vec{\mathcal{T}_p}$ et la force $\vec{\mathcal{F}_p}$ s'obtiennent en dérivant ce potentiel :

$$\begin{aligned} \vec{\mathcal{T}_p} &= -2(\vec{n}_A \cdot \vec{r})(\vec{r} \wedge \vec{n}_A) \ , \\ \vec{\mathcal{F}_p} &= -2(\vec{n}_A \cdot \vec{r})\vec{r} \ . \end{aligned} \tag{2.1}$$

– Force co-normale :

$$\Phi_N = \|\vec{n_A} - \vec{n_B}\|^2 \psi(\|\vec{r}\|) \ .$$

Si les normales de A et B sont égales, alors elles sont dans le même plan.

– Force cosphérique :

$$\Phi_C = ((\vec{n_a} + \vec{n_b}) \cdot \vec{r})^2 \psi(\|\vec{r}\|) \ .$$

Les particules A et B sont sur une même sphère si la somme de leur normales est orthogonale à $\vec{r}$.

Les propriétés des surfaces définies à l'aide de ce système sont spécifiées en modulant les coefficients de chacune de ces interactions. Par exemple la raideur d'un plan est accentuée en augmentant la part de la force coplanaire, la torsion avec la force co-normale.

Le concept de particule orientées présenté par R. Szeliski et D. Tonnesen nous semble très intéressant. Dans cette thèse, nous proposons de le généraliser, puis grâce à la définition de nouvelles interactions, nous proposons de l'utiliser pour modéliser des objets dynamiques (contrairement aux objets produits par le modeleur qui sont statiques) déformables.

L'utilisation d'un système de particules (orientées ou non) dynamique demande, entre autres, de gérer les oscillations des particules. Le chapitre suivant est consacré à ce thème.

Chapitre 3

Maintenir la cohésion du système de particules

Après avoir rappelé la technique classiquement utilisée pour maintenir la cohésion d'un ensemble de particules, nous allons étudier le problème des oscillations lié à cette technique. Une solution originale sera ensuite présentée et discutée.

3.1 Force de Lennard-Jones

Afin de maintenir la cohésion d'un ensemble de particules, une interaction d'attraction/répulsion est introduite. Cette interaction à portée limitée engendre une force de répulsion lorsque les particules sont trop proches, et une force d'attraction lorsqu'elles sont trop éloignées. La quasi-totalité des systèmes de particules utilisent une interaction d'attraction/répulsion dérivant du modèle de potentiel intermoléculaire de Lennard-Jones [AF77].

3.1.1 Description

Ce modèle a été originellement conçu pour décrire les interactions entre les molécules d'un gaz. La force exercée $\vec{F}_{LJ}$ dérive du potentiel E_{LJ} qui s'exprime comme suit :

$$\begin{aligned} E_{LJ}(r) &= \frac{\beta}{r^n} - \frac{\alpha}{r^m} , \\ \vec{F}_{LJ}(r) &= -\vec{\mathrm{grad}}_r(E_{LJ}) \end{aligned} \tag{3.1}$$

où r est la distance entre les deux particules interagissantes [1], et n, m, α et β sont des constantes du modèle. Une donnée intéressante est la distance de repos r_0 (Figure 3.1), distance à laquelle aucune force n'est exercée.

1. Deux particules sont en interaction lorsque l'une est dans le rayon d'action des interactions de l'autre.

Cette force est isotrope et ne dépend que de la distance entre les particules en interaction. L'emploi d'une telle force d'attraction/répulsion présente deux inconvénients majeurs : le jeu de paramètres n'est pas adapté à l'informatique graphique et le système oscille.
L'utilisateur non spécialiste peut difficilement prévoir l'influence de telle ou telle modification de la valeur d'un des paramètres sur le comportement de l'objet modélisé. Il existe dans la littérature [Jim93] une ré-écriture de ce potentiel faisant apparaître explicitement un facteur d'échelle E_0 et la distance de repos r_0 entre les particules :

$$\begin{aligned} E_{LJ}(r) &= -E_0 \left(2(\frac{r_0}{r})^n - (\frac{r_0}{r})^{2n} \right) , \\ \vec{\mathcal{F}}_{LJ}(r) &= \frac{2nE_0}{r_0} \left((\frac{r_0}{r})^{2n+1} - (\frac{r_0}{r})^{n+1} \right) \frac{\vec{r}}{|\vec{r}|} . \end{aligned} \quad (3.2)$$

Cette expression contient cependant encore le paramètre non intuitif n.
Un autre fait complique l'utilisation de ces paramètres : ils sont liés. Une action sur l'un d'entre eux modifie globalement la courbe et nécessite une intervention sur les autres paramètres pour obtenir l'interaction souhaitée.
L'autre inconvénient de la force d'attraction/répulsion de Lennard-Jones est plus grave, il nous a conduit à abandonner ce modèle de force d'attraction/répulsion. Les systèmes de particules l'utilisant oscillent.

3.1.2 Oscillations

Comme toute force dérivant d'un potentiel, les forces de Lennard-Jones sont conservatives. La force exercée ne dépendant que de la position relative des particules à l'instant considéré, il n'y a pas de dissipation d'énergie. Des particules qui ne sont pas initialement dans un état stable, ne l'atteignent jamais. Le phénomène d'oscillation est donc inhérent au modèle. Le problème est dû à la grande amplitude des oscillations produites, ceci étant encore aggravé par l'utilisation de techniques discrètes d'intégration des lois de la dynamique.
La technique couramment employée pour lutter contre ces grandes oscillations consiste à introduire un frottement fluide lié au milieu dans lequel les particules évoluent. Mais ce frottement doit être important pour produire l'effet souhaité. Tous les mouvements se trouvent ainsi uniformément amortis et notamment ceux qui sont dus aux actions extérieures. Cela oblige à employer des forces extérieures de plus grande amplitude pour conserver le mouvement souhaité. Comme l'intégration des équations de la dynamique est approchée de façon discrète, la manipulation de forces de grande amplitude est délicate et conduit à utiliser un pas d'intégration très petit. Ceci n'est pas acceptable dans le contexte d'animation interactive qui est le nôtre. Une autre approche consiste à introduire une interaction de frottement entre les particules. Le frottement dépend de la densité de particules

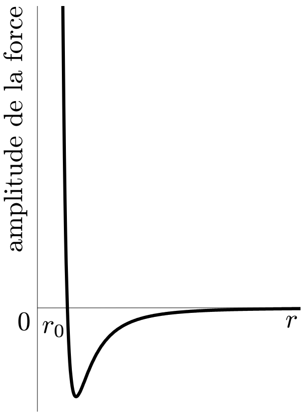


FIG. 3.1 - *Force inter-moléculaire de Lennard-Jones*

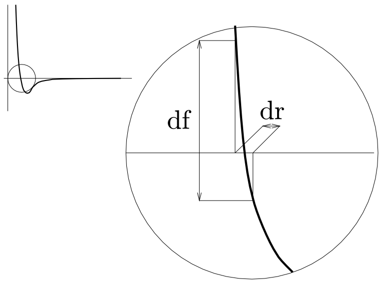


FIG. 3.2 - *Influence d'un petit déplacement autour de la position d'équilibre sur l'amplitude de la force de rappel exercée*

[MP89, DG94]. Cette approche locale donne de meilleurs résultats que l'approche globale. Toutefois, aucune de ces méthodes ne lutte contre la cause des oscillations : toute deux tentent de les masquer.

Les particules oscillent autour de leur position de repos. Considérons le cas où seulement deux particules sont en interaction et où elles sont proches de leur état d'équilibre. Si à un instant donné elles sont trop éloignées l'une de l'autre, une force d'attraction s'applique pendant le Δt suivant. A moins d'intégrer les équations de la dynamique avec un pas de temps très petit, il est probable qu'à l'instant suivant, les particules soient trop proches l'une de l'autre. Un force de répulsion s'exerce donc pendant le pas de temps suivant. Et ainsi de suite.
Les oscillations viennent du fait que les forces appliquées lorsque les particules sont proches de leur état d'équilibre sont trop grandes. L'amplitude des ces forces de rappel est liée à la pente de la tangente à la courbe autour de la distance de repos (Figure 3.2).

3.2 Redéfinition de la force d'attraction/répulsion

Notre approche est différente : puisque l'amplitude des oscillations est liée à la fonction d'attraction/répulsion, nous en écrivons une nouvelle. Cette nouvelle loi n'est pas issue de la mécanique et seules ses propriétés mathématiques nous intéressent. Nous ne perdons cependant pas de crédibilité par rapport à la physique : l'utilisation d'une loi microscopique modélisant les interactions entre les molécules d'un gaz pour simuler le phénomène macroscopique que sont les interactions entre particules sortait déjà du cadre de validité physique.

Il est possible d'exprimer mathématiquement les propriétés intuitives qu'une fonction $\vec{\mathcal{F}}(\vec{r})$ doit vérifier pour modéliser une force d'attraction/répulsion :

- $\vec{\mathcal{F}}(\vec{r})$ est isotrope et elle s'exerce suivant la direction de la droite passant par les deux particules. Son module et son sens ne dépendent que de la distance entre les deux particules :

$$\vec{\mathcal{F}}(\vec{r}) = f(r)\frac{\vec{r}}{r}$$

avec $r = \|\vec{r}\|$.
- $\vec{\mathcal{F}}(\vec{r})$ assure la non pénétration des particules :

$$\lim_{r \to 0^+} f(r) = +\infty \ .$$

- $\vec{\mathcal{F}}(\vec{r})$ n'a qu'un point d'équilibre :

$$\exists! r_0, f(r_0) = 0 \ .$$

- $\vec{\mathcal{F}}(\vec{r})$ est répulsive quand les particules sont trop proches :

$$\forall r < r_0, f(r) > 0 \ .$$

- La force de répulsion augmente quand les particules se rapprochent :

$$\forall r, 0 < r < r_0, f'(r) < 0$$

- $\vec{\mathcal{F}}(\vec{r})$ est attractive quand les particules sont trop éloignées :

$$\forall r > r_0, f(r) < 0 \ .$$

- Au fur et à mesure que les particules s'éloignent, la force de rappel croît, atteint un maximum, puis décroît pour devenir négligeable dès que la distance est trop grande :

$$\begin{aligned} \exists r_1 > r_0, f'(r_1) = 0 \ , \\ \forall r_0 < r < r_1, f'(r) < 0 \ , \\ \forall r > r_1, f'(r) > 0 \ , \\ \lim_{r \to +\infty} f(r) = 0 \ . \end{aligned}$$

L'objectif est de diminuer l'amplitude des forces de rappel lorsque les particules sont proches de l'équilibre et cela sans modifier les forces exercées lorsque les particules en sont éloignées. Nous choisissons une expression qui non seulement vérifie les conditions nécessaires pour modéliser une force d'attraction, mais en plus vérifie une nouvelle contrainte :

$$f'(r_0) = 0 \ .$$

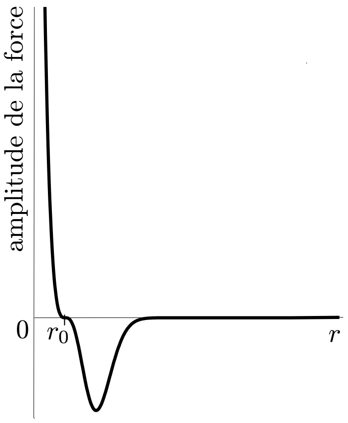


FIG. 3.3 - *Forces de Cohésion*

Une pente nulle à la position d'équilibre garantit en effet des forces de rappel d'amplitude de plus en plus faible lorsque les particules se rapprochent de leur position de repos.

> **Remarque :** Il nous faut insister sur un point important : nous avons choisi d'introduire *une tangente* nulle à la position d'équilibre et non *un palier* nul autour de la position d'équilibre. Cette dernière solution revient, d'un point de vue mécanique, à introduire un jeu. C'est effectivement une solution pour diminuer les vibrations, mais elle ne garantit pas l'unicité de la position d'équilibre [2].

Nous avons retenu l'expression suivante :

$$\begin{array}{rcl} F_{Ch}(r) & = & E_0 \frac{(r_0-r)^3}{r} e^{-\alpha(r_0-r)^2} \ , \\ \vec{\mathcal{F}}_{Ch}(r) & = & F_{Ch}(r)\frac{\vec{r}}{r} \ . \end{array} \tag{3.3}$$

Nous appellerons *force de cohésion* cette nouvelle force d'attraction/répulsion (Figure 3.3). Afin de contrôler directement le zéro de la fonction et sa dérivée en ce point, nous ne l'avons pas écrite en dérivant un potentiel. L'expression mathématique donne directement l'intensité de la force.
Cette force est elle aussi conservative. En l'absence de frottement dissipant l'énergie, les oscillations persistent. Leur amplitude est toutefois grandement diminuée. L'emploi de faibles forces de frottement est alors suffisant pour éliminer (au moins visuellement) les oscillations. Dans nos simulation, nous

2. La force étant nulle sur un intervalle $[x_0-\epsilon, x_0+\epsilon]$, les particules peuvent se déplacer à vitesse constante sur cet intervalle.

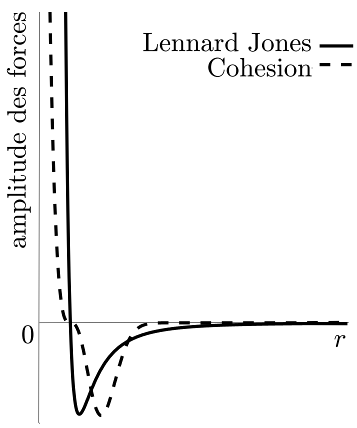


FIG. 3.4 - *Forces de Lennard-Jones et de Cohésion*

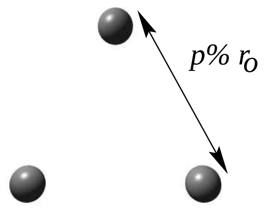


FIG. 3.5 - *Système test : 3 particules équidistantes*

utilisons à la fois un faible coéfficient de frottement lié au milieu et une interaction de frottement entre les particules pour dissiper l'énergie du système. Le jeu de paramètres est plus intuitif que celui du modèle de Lennard-Jones. En effet, r_0 donne la distance de repos, et E_0 permet de jouer sur l'amplitude de la force et donc sur la cohésion de l'objet [3]. Le paramètre α est plus délicat à manipuler. Il permet de diminuer le rayon d'influence de la force, mais E_0 doit être utilisé pour restaurer la force d'attraction maximale. Expérimentalement, nous avons constaté que l'influence d'une modification de r_0 sur l'amplitude de la force d'attraction maximale est moindre que pour une modification similaire avec la force de Lennard-Jones.

3.3 Résultats

Une étude expérimentale a montré que l'utilisation de la force de cohésion non seulement réduit énormément les oscillations, mais en plus offre une modélisation beaucoup plus robuste [4] qu'avec une force du type Lennard-Jones.

Pour l'étude comparative des oscillations, deux systèmes de particules sont initialisés dans les mêmes conditions de frottement, chacun avec un

3. Plus E_0 est grand, plus il faudra exercer une force extérieure grande pour faire changer la topologie de l'objet. Plus E_0 est petit, plus l'objet se brisera facilement en morceaux.

4. Dans des cas extrêmes (particules loin de leur position d'équilibre), les oscillations, au lieu de s'atténuer en amenant le système au repos, s'amplifient et provoquent des changements de topologie non souhaités, véritables explosions de l'objet. Il est normal que la topologie de l'objet change lorsque les forces extérieures sont plus grandes que les forces internes. Il est par contre anormal que l'objet explose à cause d'un pas d'intégration trop grand, ce qui se produit ici.

type de force d'attraction/répulsion (dont les paramètres ont été choisis pour assurer un comportement similaire, voir la figure 3.4). Les particules sont initialement placées à proximité de leur position de repos. Pendant la simulation, les oscillations sont mesurées. Appelons $\vec{\delta x_i}$ le déplacement de la particule i pendant un pas de temps. Nous avons choisi la somme de l'amplitude des déplacements des particules autour du barycentre du système :

$$\sqrt{\sum_i \|\vec{\delta x_i}\|^2 - \|\sum_i \vec{\delta x_i}\|^2}$$

comme mesure de l'amplitude des oscillations. Les tracés effectués avec divers nombres de particules ont montré que les systèmes utilisant la force de cohésion convergent toujours plus vite que ceux utilisant la force de Lennard-Jones.

Pour nos tests, deux systèmes, l'un utilisant une interaction de Lennard-Jones et l'autre une interaction de cohésion, sont composés de 3 particules placées sur un triangle équilatéral dont les côtés ont pour longueur un certain pourcentage p de la distance de repos théorique (Figure 3.5). Deux types d'expérience ont été menées avec ces systèmes. Les simulations ont été calculées :

– pour un dt fixé, en faisant varier p, et

– pour un p fixé, en faisant varier dt.

Dans le premier cas, nous avons mesuré l'amplitude des oscillations : c'est un test de robustesse. Dans le second cas, nous avons mesuré le temps nécessaire pour que l'amplitude des oscillations devienne inférieur à un seuil fixé : c'est un test de rapidité de convergence.

Test de robustesse :

Les simulations ont été calculées pour p allant de 100% à 40% (Figure 3.6) en utilisant un pas de temps $dt = 0.02s$ pendant $1s$. Avec une longueur inférieure à 40%, les deux systèmes divergent. Le système utilisant la force de Lennard-Jones diverge dès que la distance entre les particules tombe en dessous de 65%, alors que le système utilisant la force de cohésion converge encore avec une distance initiale de 40% de la distance de repos. La force de cohésion, dans ce cas extrême, ne produit pas plus d'oscillations que la force de Lennard-Jones dans un bon cas ($p = 97\%$).
Pendant une animation, sauf en cas de collision provoquant un déplacement rapide d'une ou plusieurs particules, les changements de forme sont lents. Ceci signifie que, si à l'instant t le système est au repos, à l'instant $t + dt$ il sera dans un état proche de son état d'équilibre. Les courbes du haut montrent que l'amplitude des oscillations reste faible.

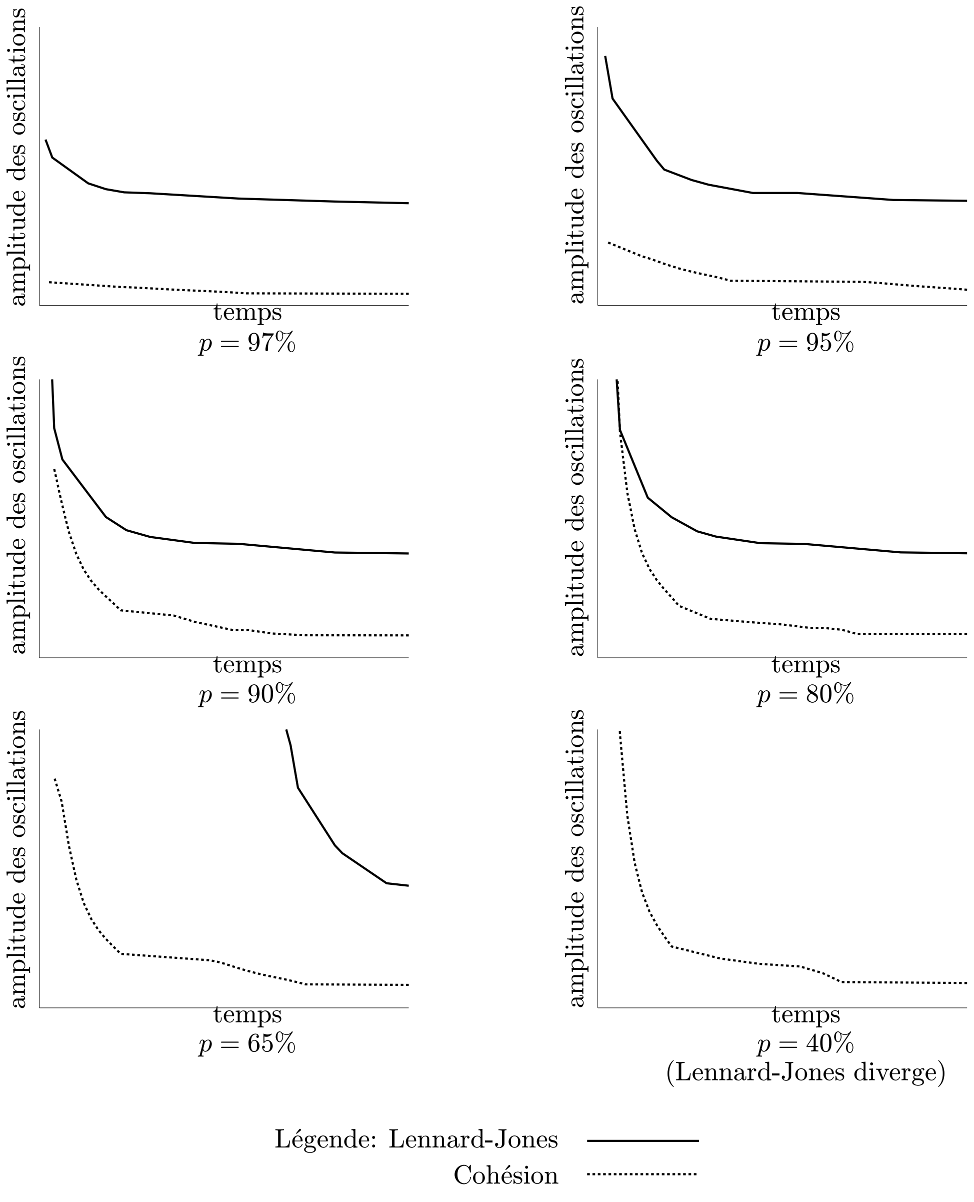
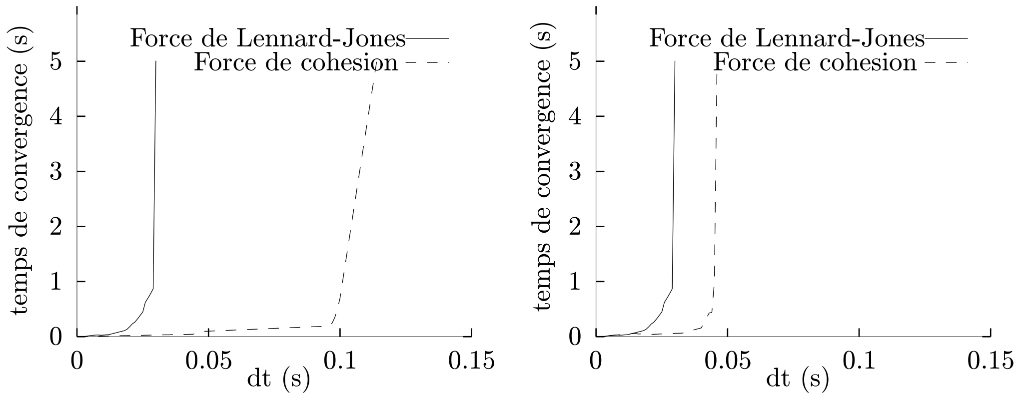


FIG. 3.6 - *Comparaison de l'amortissement des oscillations*



$p = 90\%$ $p = 70\%$

FIG. 3.7 - *Comparaison de la vitesse d'amortissement*

L'autre enseignement à tirer de cette expérience est le suivant : dans le système utilisant la force de Lennard-Jones, les oscillations résiduelles (après une seconde de simulation) sont beaucoup plus importantes que dans le système utilisant la force de cohésion. Ces oscillations entraînent une gêne visuelle et un semblant de non déterminisme dans le comportement du système. Selon qu'un événement se produit lorsque les particules sont les plus proches les unes des autres ou lorsqu'elles sont éloignées, le comportement ne sera pas le même. Ceci est incontrôlable.

Test de rapidité de convergence :

De façon similaire, nous avons effectué un test de rapidité de convergence. Comme précédemment, deux systèmes de trois particules sont initialisés. Cette fois, les simulations successives sont effectuées pour la même position initiale en utilisant un pas d'intégration allant de $1/1000s$ au dt tel que les deux systèmes divergent. Le temps à partir duquel les oscillations sont d'une amplitude inférieure à un seuil déterminé (nous avons choisi $2,5\%r_0$, qui est l'amplitude résiduelle du système de Lennard-Jones dans l'expérience précédente) est tracé en fonction du pas de temps (Figure 3.7). L'expérience a été faite pour un système initialisé proche de sa position d'équilibre ($p = 90\%$) et pour un système perturbé ($p = 70\%$).

Cette expérience montre que le système utilisant la force de cohésion converge toujours plus vite et pour des dt plus importants que le système utilisant les forces de Lennard-Jones. La différence est très importante dans le cas d'un système peu perturbé. Ceci fournit un indice pour déterminer le pas de temps à utiliser dans nos animations. En effet, tant que les particules sont proches de leur état d'équilibre, nous pouvons intégrer la dynamique avec un pas de temps grand, et donc réduire les calculs. Par contre, lorsque le système est perturbé, il nous faut diminuer le dt. La détection du passage de l'état *proche de l'équilibre* à l'état *perturbé* peut se faire en étudiant l'amplitude des forces appliquées [5]. Cette constatation nous permet de facilement implémenter un moteur dynamique à pas de temps variable.

3.4 Du point de vue théorique

L'intuition a fortement guidé l'écriture de la force de cohésion. Cette section comporte une étude un peu plus théorique. En se référant aux méthodes d'intégration des équations de la dynamique du point décrites dans l'annexe A (page 111), nous examinons les erreurs faites lors de l'intégration numérique des forces de Lennard-Jones et de cohésion.

L'erreur commise lors du calcul de la position en utilisant le schéma d'intégration d'Euler modifié est (éq. A.4, page 113) :

$$\vec{\epsilon}_x = O(\frac{d\vec{f}_{ext}}{dt}\Delta t^3 + \vec{f}_{ext}\Delta t^2) \ .$$

Remarquons que la force exercée f_{ext} est donnée en fonction de la position relative x des objets et non du temps.

$$\begin{aligned}\vec{\epsilon}_x &= O(\frac{d\vec{f}_{ext}}{dx}\frac{dx}{dt}\Delta t^3 + \vec{f}_{ext}\Delta t^2) \\ &= O(\vec{f'}_{ext}\vec{v}\Delta t^3 + \vec{f}_{ext}\Delta t^2) \ . \end{aligned} \tag{3.4}$$

Les deux forces d'attraction/répulsion sont nulles en r_0. Par contre, la dérivée de la force de Lennard-Jones a une valeur élevée en r_0, d'où une erreur importante.

Dans l'expression 3.4, la fonction modélisant les forces appliquées $\vec{f}_{ext}$ et sa dérivée sont prises en un point inconnu de l'intervalle Δt considéré. Pour pouvoir comparer les deux forces d'attraction/répulsion, nous devons étudier leurs dérivées premières et secondes (la variation de la dérivée première dépend de la dérivée seconde).

La force de cohésion F_{Ch} s'exprime comme suit (éq. 3.3, page 39) :

$$F_{Ch}(r) = E_0\frac{(r_0-r)^3}{r}e^{-\alpha(r_0-r)^2} \ .$$

Ses dérivées première et seconde par rapport à r sont :

$$\begin{aligned}\frac{dF_{Ch}}{dr} &= E_0\frac{(r_0-r)^2}{r}e^{-\alpha(r_0-r)^2}\left(-3+2\alpha(r_0-r)^2-\frac{r_0-r}{r}\right) \ , \\ \frac{d^2F_{Ch}}{dr^2} &= E_0\frac{r_0-r}{r}e^{-\alpha(r_0-r)^2}\left(6-14\alpha(r_0-r)^2+6\frac{(r_0-r)^2}{r}\right. \\ &\qquad \left. +4\alpha^2(r_0-r)^4-4\,\alpha\frac{(r_0-r)^3}{r}+2\frac{(r_0-r)^2}{r^2}\right) \ .\end{aligned}$$

5. Dans un système perturbé, les forces appliquées sont grandes.

La valeur algébrique de la force de Lennard-Jones F_{LJ} exprimée comme suit (équation 3.2, page 36) :

$$F_{LJ}(r) = 2\frac{nE_0}{r_0}\left((\frac{r_0}{r})^{2n+1} - (\frac{r_0}{r})^{n+1}\right)$$

admet les dérivées première et seconde suivantes :

$$\begin{aligned}
\frac{dF_{LJ}}{dr} &= 2\frac{nE_0}{r_0^2}\left((n+1)(\frac{r0}{r})^{n+2} - (2n+1)(\frac{r_0}{r})^{2n+2}\right) \;, \\
\frac{d^2F_{LJ}}{dr^2} &= -2\frac{nE_0}{r_0^3}\left((n+1)(n+2)(\frac{r_0}{r})^{n+3} - (2n+1)(2n+2)(\frac{r_0}{r})^{2n+3}\right) \;.
\end{aligned}$$

Au point $r = r_0$, nous avons :

$$\begin{aligned}
\frac{dF_{Ch}(r_0)}{dr} &= 0 \;, \\
\frac{d^2F_{Ch}}{dr^2} &= 0 \;, \\
\frac{dF_{LJ}}{dr} &= 2\frac{nE_0}{r_0^2}\left((n+1) - (2n+1)\right) \\
&= -2\frac{n^2E_0}{r_0^2} \;, \\
\frac{d^2F_{LJ}}{dr^2} &= -2\frac{nE_0}{r_0^3}\left((n+1)(n+2) - (2n+1)(2n+2)\right) \\
&= 6\frac{n^2(n+1)E_0}{r_0^3} \;.
\end{aligned}$$

Le fait que la dérivée seconde de la force de cohésion soit nulle en r_0 implique que la dérivée première varie peu autour de r_0, et donc que sa valeur reste faible. L'erreur commise pendant l'intégration de la force de cohésion est donc petite lorsque les particules sont proches de leur état d'équilibre.

La figure 3.8 montre les dérivées des forces de Lennard-Jones et de cohésion sur un intervalle autour de r_0, pour des valeurs des paramètres des deux fonctions les rendant similaires (Figure 3.4). Nous observons que la dérivée de la force de cohésion, nulle en r_0, reste négligeable sur tout l'intervalle d'étude, ce qui n'est pas le cas de la force de Lennard-Jones. Ceci confirme bien que l'erreur commise en intégrant la force de cohésion est très inférieure à celle commise lors de l'intégration de la force de Lennard-Jones.

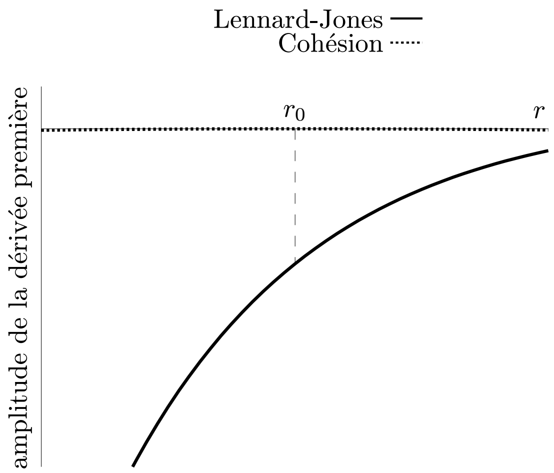


FIG. 3.8 - *Dérivée première des forces d'attraction/répulsion autour de l'équilibre*

Chapitre 4

Objets déformables et particules orientées

Nous souhaitons modéliser des objets qui, sous l'action d'un champ de forces extérieures, se déforment. Lorsque le champ de forces disparait, l'objet doit reprendre sa forme initiale si les forces externes sont restées petites par rapport aux forces internes. Dans le cas contraire, l'objet a subi des déformations irrémédiables (changements de forme, voire de topologie).

Ce chapitre contient le cœur de mon travail. Dans une première partie, sont présentées les nouvelles interactions anisotropes et l'algorithme de simulation qui nous permettent de modéliser de tels objets déformables. Ce modèle permet de simuler des surfaces et des objets déformables. Le modèle de surface repose sur la conservation de propriétées mathématiques locales (courbures). Les interactions sont dotées de zones d'influence (et de repos pour l'interaction de cohésion) ellipsoïdales, permettant ainsi d'obtenir un échantillonnage de l'objet [1] dont la densité dépend de la direction.
La deuxième partie présente une variante de ce nouveau modèle d'objets déformables, reposant sur l'utilisation conjointe d'un système de particules orientées et d'un modèle d'objets déformables par surfaces implicites. Cette partie présente tout d'abord le modèle de simulation d'objets déformables par surfaces implicites présenté par M.-P. Gascuel au SIGGRAPH'93 puis l'une de ses variantes, avant de décrire notre nouveau modèle d'objets déformables implicites à squelette souple.

Partie A : Surfaces et objets discrets

Cette partie présente une méthode de modélisation de surfaces et d'objets déformables reposant sur l'utilisation d'un système de particules orientées.

1. l'ensemble de particules modélisant un objet forme un échantillonnage de cet objet

Les objets sont représentés par un échantillonnage discret (l'ensemble des particules).

4.1 Conserver la forme de l'objet

Comme nous l'avons vu au chapitre 2, les systèmes de particules classiques permettent de modéliser des substances plus ou moins fluides. Ces corps déformables présentent l'intérêt de ne pas avoir de topologie fixe. Par contre, ils n'ont pas de forme propre. Lorsqu'ils sont au repos, leur géométrie est complètement déterminée par les objets avec lesquels ils sont en contact. Le concept de particule orientée va nous permettre d'imposer des contraintes géométriques, donc de modéliser de véritables objets et non plus des substances. Ceci est fait tout en conservant les propriétés des systèmes de particules non-orientées.

Nous proposons d'utiliser un système similaire au modeleur interactif de R. Szeliski et D. Tonnesen (Section 2.2.4, page 31) afin de simuler des objets *à mémoire de forme*. Il nous faut attirer l'attention du lecteur sur un point : le modeleur est dynamique, c'est-à-dire que le processus de modélisation d'objets repose sur des déformations induites par l'application de champs de forces. Mais l'objet modélisé n'est pas dynamique, il est statique. La position des différentes particules composant l'objet n'est pas le résultat d'un équilibre dynamique (état d'énergie minimale du système), mais de la volonté de l'utilisateur.
C'est la différence principale entre un modeleur et un simulateur. Dans le modeleur, l'utilisateur, par le biais d'un mécanisme de gel et dégel des particules, contrôle le moteur dynamique. Lorsque la position de certaines particules est jugée satisfaisante, elles sont soustraites à l'action des lois de la dynamique, qu'elles soient dans une position d'équilibre ou non. Elles sont comme clouées dans l'espace, leur position est fixe et elles peuvent servir de points d'appui pour les déformations d'autres parties de la surface. Dans un simulateur, l'utilisateur n'a pas de contrôle sur le moteur dynamique. Les particules ne restent dans une position que si c'est un état d'énergie minimale. La géométrie de l'objet non déformé est définie par la position des particules lorsqu'elles sont *toutes* au repos.
La forme de l'objet est définie par les interactions locales entre les particules. Le jeu d'interactions proposé par R. Szeliski et D. Tonnesen n'est pas suffisant pour la simulation dynamique d'objets déformables. En effet, il faut que le lieu des positions relatives des particules lorsqu'elles sont au repos par rapport à l'interaction considérée soit une configuration géométrique stable. C'est bien le cas pour l'interaction coplanaire, mais ce ne l'est pas pour l'interaction cosphérique[2]. En effet, une particule orientée définit un plan

2. Se référer à la section 2.2.4 (page 33) pour la définition des interactions selon R. Szeliski et D. Tonnesen.

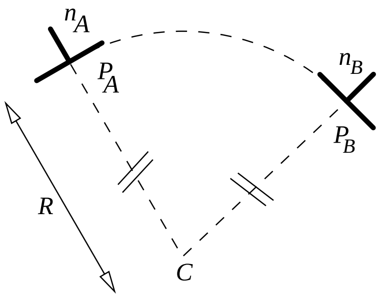


FIG. 4.1 - *Deux particules cosphériques*

unique (un point et une normale) alors qu'elle définit une infinité de sphères (toutes tangentes au même point : la particule). L'interaction cosphérique contraint les particules à se positionner sur une sphère quelconque. Considérons, pour simplifier, un exemple 2D où deux particules sont initialement sur un cercle[3]. Si un événement (une collision par exemple) cause un déplacement relatif de ces deux particules, l'interaction cosphérique va produire des forces et des moments dont l'effet sera de rendre les particules à nouveau co-cycliques. Après l'événement extérieur, les particules se trouvent à proximité d'un cercle et vont rapidement s'y positionner. Ce cercle n'a aucune raison d'être le même que le cercle initial. L'objet n'a donc aucune chance de retrouver sa forme. La contrainte imposée par l'interaction cosphérique n'est donc pas assez forte. Afin de modéliser des objets qui retrouvent leur forme initiale après déformation, nous devons définir une interaction imposant une contrainte plus forte aux particules.

4.1.1 Interaction cosphérique

Nous souhaitons ré-écrire l'interaction cosphérique de façon à spécifier le rayon de la sphère sur laquelle doivent se situer les particules. Une particule et un rayon définissent bien une sphère unique.
Une méthode simple pour obtenir les forces et moments adéquats est de les dériver d'un potentiel adéquat. Un tel potentiel doit être minimal dans la configuration de repos. Toute fonction de distance entre un état arbitraire du système (l'état courant) et l'état de repos désiré est candidate. Nous décrivons maintenant notre choix.

Une particule orientée A (de position P_A et de normale n_A) et un rayon R définissent un cercle unique. Le centre de ce cercle est le point $\vec{P_A} - R\vec{n_A}$ (Figure 4.1). Deux cercles sont identiques si et seulement si ils ont même rayon et même centre. Donc deux particules A et B sont sur le même cercle

3. Les particules étant définies entre autres par une position et un vecteur normal, deux particules ne sont pas trivialement sur un cercle.

de rayon R si et seulement si :

$$\vec{P}_A - R\vec{n}_A = \vec{P}_B - R\vec{n}_B$$

ou, de façon équivalente :

$$\|\vec{r} + R(\vec{n}_A - \vec{n}_B)\| = 0 \ ,$$

où $\vec{r} = \vec{P}_B - \vec{P}_A$. Afin d'éviter le calcul d'une racine carrée à chaque pas de temps, nous avons choisi :

$$\Phi_C = \|\vec{r} + R(\vec{n}_A - \vec{n}_B)\|^2 \tag{4.1}$$

comme potentiel. En dérivant le potentiel décrit par l'équation 4.1, nous obtenons donc la nouvelle loi d'interaction cosphérique composée du moment $\vec{\mathcal{T}}_C$ et de la force $\vec{\mathcal{F}}_C$:

$$\left\{ \begin{array}{lcl} \vec{\mathcal{T}}_C & = & 2R[\vec{n}_A \wedge (\vec{r} - R\vec{n}_B)] \ \Psi(\vec{r}) \\ \vec{\mathcal{F}}_C & = & 2[\vec{r} + R(\vec{n}_A - \vec{n}_B)] \ \Psi(\vec{r}) \ . \end{array} \right. \tag{4.2}$$

La fonction Ψ dépend en particulier de la distance entre les particules. Elle est utilisée pour limiter la portée de l'interaction.

4.1.2 Interaction de forme

La nouvelle interaction cosphérique définie précédemment permet bien de modéliser des objets à mémoire de forme. Par contre les objets modélisés doivent être composés de morceaux de sphères ou de plans. Ceci est très insuffisant.
Aller plus loin nécessite un rappel préalable de certaines propriétés mathématiques des surfaces [LFA75].

Définition de la courbure

Soit M un point d'une surface Σ de classe C^1 (Figure 4.2). Soit P le plan passant par M, contenant la normale en M à Σ, et dans la direction φ autour de M. Soit Γ la courbe définie par l'intersection de P et de Σ. On appelle *courbure normale* de la surface Σ au point M, dans la direction φ, l'inverse du rayon du cercle osculateur C de la courbe Γ. Le cercle osculateur est la limite du cercle tangent à Γ en M passant par un point M' de Γ, lorsque M' tend vers M.
Dans ce document, afin d'alléger le texte, nous nous permettons d'appeler courbure ce qui en toute rigueur est la courbure normale.

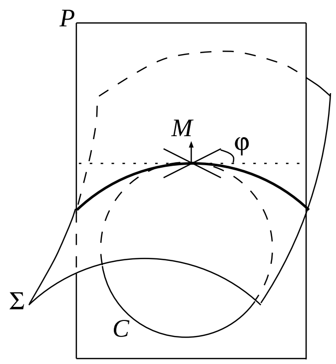


FIG. 4.2 - *Définition de la courbure*

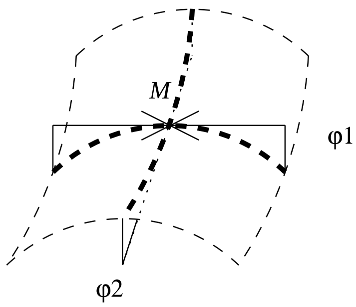


FIG. 4.3 - *Courbures principales*

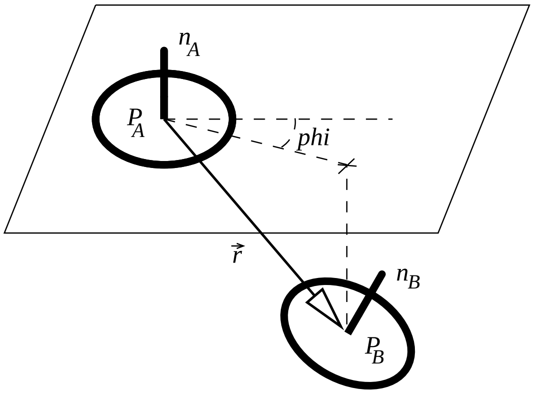


FIG. 4.4 - *Calcul de la direction de la particule en interaction*

Équation aux courbures principales

La courbure en M est donc une fonction de la direction autour de M. On appelle *ombilic* un point tel que la courbure soit constante dans toutes les directions[4].

- **Si M n'est pas un ombilic**, alors il existe deux *directions principales* φ_1 et φ_2 où la courbure atteint ses extremums ρ_1 et ρ_2, appelés *courbures principales* (Figure 4.3). Les directions principales sont orthogonales :

  $$\varphi_2 = (\varphi_1 + \frac{\pi}{2}) \,\mathrm{modulo}\, \pi \ .$$

  L'équation aux courbures principales donne la courbure ρ dans une direction φ quelconque :

  $$\rho = \rho_1 \cos^2(\varphi_1 + \varphi) + \rho_2 \cos^2(\varphi_2 + \varphi) \qquad (4.3)$$

- **Si M est un ombilic**, alors la courbure ρ_0 est constante dans toutes les directions. L'équation 4.3 reste valable :

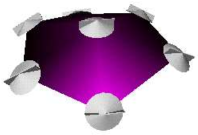
  $$\rho = \rho_0 = \rho_0(\cos^2(\varphi_0 + \varphi) + sin^2(\varphi_0 + \varphi))$$

  où φ_0 est quelconque.

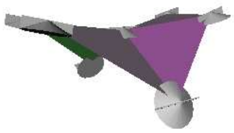
Définition de l'interaction

Pour le calcul de l'interaction de forme, chaque particule est dotée d'une direction principale φ_1 et de deux courbures principales ρ_1 et ρ_2 associées (ρ_1 est associée à φ_1 et ρ_2 à $\varphi_1 + \frac{\pi}{2}$). L'équation aux courbures principales nous donne une courbure (et donc un rayon) dans toutes les directions autour

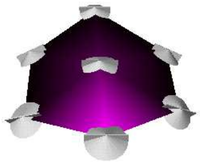
a) ombilic



b) point elliptique



c) point hyperbolique



d) point parabolique

e) surface localement plane

FIG. 4.5 - *Différentes configurations locales des surfaces*

d'un point donné. La direction φ de la particule en interaction est calculée en projetant sa position dans le plan perpendiculaire à la normale (Figure 4.4).

L'interaction de forme est définie en calculant le rayon de courbure dans la direction de la particule considérée (éq. 4.3) et en l'utilisant dans l'interaction cosphérique définie précédemment. Tout type de surface[5] peut être défini grâce à cette interaction.
Suivant la valeur donnée aux paramètres ρ_1 et ρ_2 (φ_1 ne change pas le type de point de la surface, seulement la direction des courbures), une particule peut modéliser (la particule considérée est au centre des figures) :

- un ombilic, si $\rho_1 = \rho_2$ (Figure 4.5a),
- un point elliptique, si $\rho_1\rho_2 > 0$ (Figure 4.5b),
- un point hyperbolique, si $\rho_1\rho_2 < 0$ (Figure 4.5c),
- un point parabolique, si $\rho_1\rho_2 = 0$ (une des courbures principales est nulle, Figure 4.5d), ou
- un plan, si $\rho_1 = \rho_2 = 0$ (Figure 4.5e).

Pour des raisons évidentes de coût (le calcul de l'interaction de forme demande de calculer φ, ce qui est inutile dans le cas de surfaces localement planes ou sphériques), nous conservons les interactions coplanaire et cosphérique bien que l'interaction de forme permette de modéliser les mêmes comportements[6].

L'interaction de forme, telle qu'elle vient d'être définie, permet de simuler n'importe quelle surface. L'utilisateur fixe les deux courbures principales, et l'interaction produit les forces et les moments de torsion pour positionner les particules sur la surface caractérisée par ces courbures.

4.1.3 Autres interactions définies

Afin d'augmenter les possibilités de notre modélisation, nous définissons quelques nouvelles interactions.

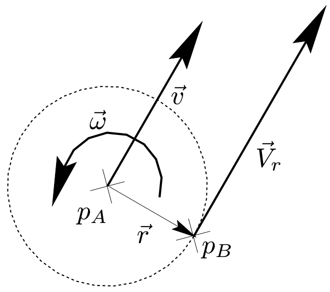
Interaction co-linéaire

Afin de modéliser des objets filaires, nous introduisons une autre interaction, nommée *co-linéaire*. Cette interaction positionne les particules sur la ligne définie par la position et la normale. Deux particules A et B

4. Par exemple, tout point d'une sphère est un ombilic.

5. Les zones où la surface n'est pas C^1 sont modélisées par des particules n'ayant pas d'interaction de forme.

6. Avec $\rho_1 = \rho 2 = 0$ la particule modélise une portion de plan, et avec $\rho_1 = \rho 2 \neq 0$ la particule modélise une portion de sphère de rayon $1/\rho_1$.



$$\vec{V}_r = \vec{v} + \vec{\omega} \wedge \vec{r}$$

FIG. 4.6 - *Vitesse relative de deux particules*

sont alignées si et seulement si $\vec{n}_A \wedge \vec{r} = \vec{0}$ et $\vec{n}_B \wedge \vec{r} = \vec{0}$. Le potentiel $\Phi_L = \|\vec{n}_A \wedge \vec{r}\|^2 + \|\vec{n}_B \wedge \vec{r}\|^2$ est nul lorsque les particules sont alignées. En le dérivant nous obtenons les interactions :

$$\begin{aligned} \vec{\mathcal{T}}_L &= -2(\vec{r} \cdot \vec{n}_A)(\vec{r} \wedge \vec{n}_A) \ , \\ \vec{\mathcal{F}}_L &= 2(\vec{r} \wedge \vec{n}_A) \wedge \vec{n}_A + 2(\vec{r} \wedge \vec{n}_B) \wedge \vec{n}_B \ . \end{aligned}$$

Interaction de frottement

Toutes les interactions utilisées ne dépendent que des positions relatives des particules. Elles sont donc toutes conservatives. Afin de ne pas avoir de système oscillant à l'infini nous introduisons des frottements. Comme nous l'avons dit au chapitre 3, nous utilisons une interaction de frottement fluide entre les particules. L'interaction de frottement tend à empêcher tout mouvement relatif des particules en interaction. Pour des particules non-orientées, M. Desbrun et M.-P. Gascuel [DG94] ont proposé la force de frottement :

$$\vec{\mathcal{F}}_f = k\|\vec{v}\|\vec{v}\ \Psi(\vec{r})$$

où $\vec{r}$ et $\vec{v}$ sont respectivement la position et la vitesse relative des deux particules en interaction, et Ψ une fonction qui permet de limiter la portée de l'interaction.

Il nous faut étendre ce frottement entre deux points matériels (particules non-orientées) aux interactions entre deux solides (particules orientées). En considérant que le solide défini par la particule A est une sphère de rayon $\|\vec{r}\|$ (Figure 4.6), la vitesse relative $\vec{V}_r$ de A par rapport à B est donnée par :

$$\vec{V}_r = \vec{v} + \vec{\omega} \wedge \vec{r}$$

où $\vec{\omega}$ est la vitesse relative de rotation des deux particules.

Nous définissons l'interaction de frottement composée du moment $\vec{\mathcal{T}}_f$ et de

FIG. 4.7 - *Ellipsoïde d'influence*

la force $\vec{\mathcal{F}}_f$ comme suit :

$$\begin{aligned}\vec{\mathcal{T}}_f &= -k\|\vec{\omega}\|\vec{\omega}\ \Psi(\vec{r})\ , \\ \vec{\mathcal{F}}_f &= -k\|\vec{v}+\vec{\omega}\wedge\vec{r}\|(\vec{v}+\vec{\omega}\wedge\vec{r})\ \Psi(\vec{r})\end{aligned}$$

où $\vec{r}$, $\vec{v}$, $\vec{\omega}$ sont respectivement les position, vitesse linéaire et vitesse angulaire relatives des particules en interaction et k un coefficient de frottement.

4.1.4 Limiter la portée des interactions

Toutes les interactions définies sont le produit d'une partie géométrique, fonction des positions relatives des particules, et d'une fonction scalaire $\Psi(\vec{r})$, dépendant entre autres de la distance entre les particules. Nous utilisons cette dernière, non seulement pour limiter la portée des interactions mais aussi pour limiter la dépendance à l'échelle de l'interaction. L'*échelle* d'un objet est donné par r_0 la distance de repos pour l'interaction d'attraction/répulsion. Comme nous le verrons plus tard (Section 5.1.2), l'intérêt de la modélisation avec des particules est de faire cohabiter, au sein d'un même objet, des particules de rayons différents (ceci permet de sur-échantillonner les zones intéressantes et de sous-échantillonner les zones de peu d'intérêt). Ceci n'est réalisable que si, lorsque r_0 varie, le comportement de l'interaction ne varie pas ou presque. Les fonctions $\Psi(\vec{r})$ peuvent jouer ce rôle.

De plus, l'utilisation de particules orientées nous permet de définir une fonction Ψ anisotrope, et par là même de définir une zone d'influence ellipsoïdale pour les interactions.
Chaque interaction est dotée d'un ellipsoïde d'influence caractérisé par trois rayons x_{max}, y_{max} et z_{max} suivant chacun des axes du repère de la particule. $\Psi(\vec{r})$ est définie comme étant le produit d'un terme de limitation de la portée $\alpha(\vec{r})$ et d'un terme d'indépendance à l'échelle spécifique à chaque interaction.
La fonction α doit être nulle lorsque la particule considérée est en dehors de l'ellipsoïde d'influence de l'interaction (Figure 4.7). En écrivant $\vec{r}=(x,y,z)$,

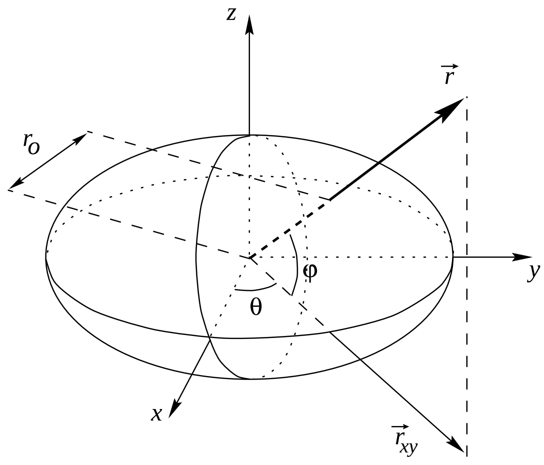


FIG. 4.8 - *Calcul du rayon de repos directionnel*

nous avons choisi :

$$\alpha(\vec{r}) = \begin{cases} 0 & \text{si } d(\vec{r}) > 1 \ , \\ 1 - d(\vec{r})^2 & \text{sinon} \ , \end{cases}$$

où

$$d(\vec{r}) = (\frac{x}{x_{max}})^2 + (\frac{y}{y_{max}})^2 + (\frac{z}{z_{max}})^2 \ .$$

Le choix de $1-d(\vec{r})^2$ plutôt que $1-d(\vec{r})$ est justifié par le fait que la première expression reste plus proche de 1 et donc perturbe moins l'amplitude de l'interaction.

Interaction de cohésion

Afin de tirer pleinement partie des possibilités offertes par le rayon d'influence ellipsoïdal, il nous faut aussi définir un rayon de repos ellipsoïdal, de manière à permettre une variation de la densité d'échantillonnage en fonction de la direction. En fait, il suffit de remplacer la valeur de la distance de repos r_0 de l'expression de la force de cohésion par une fonction $r_0(\vec{r})$.
La nouvelle interaction de cohésion est dotée de trois rayons de repos r_{0x}, r_{0y} et r_{0z} suivant les axes du référentiel de la particule. Il nous faut calculer, pour une position relative $\vec{r}$ de la particule en interaction, le rayon de repos r_0 (Figure 4.8). De façon triviale (r_0 étant la distance du centre à un point de l'ellipse), nous avons :

$$r_0 = \sqrt{(r_{0x} \cos\theta \cos\varphi)^2 + (r_{0y} \sin\theta \cos\varphi)^2 + (r_{0z} \sin\varphi)^2} \ .$$

D'un autre coté, nous pouvons exprimer les sinus et cosinus des angles θ et φ en fonction des coordonnées x, y et z de $\vec{r}$:

$$\cos\theta = \frac{x}{\|\vec{r}_{xy}\|} \ , \qquad \sin\theta = \frac{y}{\|\vec{r}_{xy}\|} \ ,$$

$$\cos\varphi = \frac{\|\vec{r}_{xy}\|}{\|\vec{r}\,\|} \ , \qquad \sin\varphi = \frac{z}{\|\vec{r}\,\|} \ .$$

Ceci mène donc à :

$$r_0(\vec{r}\,) = \sqrt{\frac{r_{0x}^2 x^2 + r_{0y}^2 y^2 + r_{0z}^2 z^2}{x^2 + y^2 + z^2}} \ . \tag{4.4}$$

Interaction coplanaire

Nous utilisons l'interaction coplanaire proposée par R. Szeliski et D. Tonnesen (éq. 2.1, page 33). Elle est proportionnelle à $\|\vec{r}\,\|^2$, nous prenons donc :

$$\Psi_p(r) = \frac{\alpha(r)}{r^2} \ .$$

Interaction cosphérique

L'interaction cosphérique est proportionnelle à $R\|\vec{r}\,\|^2$, donc :

$$\Psi_c(r) = \frac{\alpha(r)}{Rr^2} \ .$$

Interaction de forme

L'interaction de forme est l'interaction cosphérique avec le rayon donné par l'équation aux courbures principales. Dans l'expression de Ψ nous prenons le rayon moyen (l'inverse de la courbure moyenne) comme facteur de mise à l'échelle :

$$\Psi_{\mathrm{f}}(r) = \frac{\alpha(r)(\rho_1 + \rho_2)}{2r^2} \ .$$

Interaction co-linéaire et de interaction de frottement

Ces deux interactions sont proportionnelles à $\|\vec{r}\|^2$, comme l'interaction coplanaire :

$$\Psi_l(r) = \Psi_f(r) = \Psi_P(r) = \frac{\alpha(r)}{r^2} \ .$$

4.2 Modélisation de surfaces à mémoire de forme

Dans la section précédente, nous avons définis un ensemble de lois d'interactions anisotropes. Cette section présente l'utilisation de ces interactions pour simuler des objets déformables. Tout d'abord, nous généralisons le concept de particule orientée afin de pouvoir utiliser notre interaction de forme et de pouvoir faire de la modélisation volumique en plus de la modélisation surfacique. Afin de modéliser conjointement plusieurs objets, et afin de modéliser des objets plus complexes, nous définissons un mécanisme de typage des particules, et un ensemble de règles qui influent sur le mode de calcul des interactions. Nous présentons finalement l'algorithme de calcul de la simulation, puis quelques exemples de simulation d'objets déformables.

4.2.1 Définition généralisée des particules orientées

Selon R. Szeliski et D. Tonnesen, une particule orientée est un *élément de surface* (Section 2.2.4), c'est-à-dire un couple (position, normale).
Afin de pouvoir aussi modéliser des éléments de volume, nous avons choisi de représenter une particule orientée par un référentiel complet : une position et 3 axes. Ce choix entraîne un surcoût : dans les deux cas, la position est donnée par une translation par rapport à une position de référence et l'orientation par une rotation par rapport à une orientation de référence. Pour calculer l'orientation courante d'une particule selon R. Szeliski et D. Tonnesen, l'application de la formule de Rodriguez (voir annexe B, équation B.1, page 117) se réduit à 8 multiplications et 4 additions, alors que pour le calcul de l'orientation d'un référentiel complet (en utilisant les quaternions) il faut effectuer 16 multiplications et 12 additions. Cependant, modéliser une particule comme un référentiel complet permet d'obtenir une plus grande richesse du modèle (la "normale" utilisée dans la description des interactions est un axe quelconque exprimé dans le repère local de la particule) et permet aussi l'utilisation de l'interaction de forme (le calcul de cette interaction demande de calculer la direction de la particule en interaction dans le plan tangent de la particule de référence).

Les interactions précédemment définies nous permettent de modéliser et de simuler des objets quelconques à mémoire de forme. Chaque particule est dotée d'une loi d'interaction qui est une somme pondérée des interactions en jeu.

4.2.2 Typage des particules

Pour modéliser des comportements plus complexes (en particulier "non homogènes"), nous introduisons un mécanisme de typage des particules. Deux classes de type sont définies : le type *charnière* et les types *standards.* À priori, il y a un seul type charnière et autant de types standards que nécessaire. Le principe sous-jacent est de subdiviser un objet en parties connectées par des charnières. Un type standard est attribué à chaque partie de l'objet. Nous définisons trois modes d'interactions :

- **interaction complète** : la loi d'interaction utilisée est la somme pondéré des interactions d'attraction/répulsion et des interactions géométriques modélisant la forme de l'objet.
- **Interaction charnière** : l'interaction se limite à une attraction/répulsion.
- **Répulsion simple** : l'interaction est réduite à une répulsion.

Les règles suivantes sont appliquées lors du calcul de l'interaction entre les particules :

- *une charnière a une interaction charnière avec chaque particule standard.*
- *Une particule standard a une interaction complète avec toutes les autres particules du* même *type standard* et *avec les charnières.*
- *Les particules qui n'ont ni une interaction complète ni une interaction charnière ont une répulsion simple.*

4.2.3 Définition des objets

Comme dans la plupart des systèmes de particules utilisés pour simuler des corps déformables, et afin de simuler plusieurs objets dans la même scène, nous introduisons la notion d'objet. Intuitivement, nous souhaitons que deux objets restent au cours de la simulation deux entités distinctes et se repoussent. Ceci est réalisé simplement en introduisant un identificateur d'appartenance à un objet sur chaque particule. Lors du calcul de l'interaction entre deux particules, la règle suivante est appliquée :

- *si deux particules appartiennent au même objet, l'interaction est calculée en utilisant les règles décrites ci-dessus, sinon, une répulsion simple est appliquée.*

L'utilisation d'une telle règle permet la détection et le traitement des collisions entre objets modélisés avec des particules *sans coût supplémentaire.*

Notre simulateur 2D supporte aussi des collisions "hétérogènes" avec des objets définis par une fonction booléenne intérieur/extérieur (c'est par exemple le cas du sol). Comme les particules sont des éléments discrets de la surface modélisée, la collision peut être détectée en testant chaque particule avec cette fonction. Si une particule est à l'intérieur de l'objet, c'est qu'il y a collision. Une méthode de pénalité est implémentée pour traiter ces collisions.

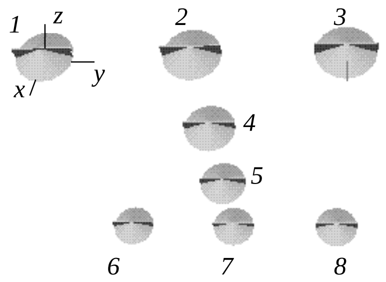


FIG. 4.9 - *Définition de l'objet*

4.2.4 Algorithme de simulation

L'algorithme de simulation du système de particules est un algorithme standard :

Tant que dure la simulation
  Pour chaque particule p_i de la scène
    Pour chaque particule p_j ($j \neq i$) de la scène
      (Force,Couple) = appliquer la loi d'interaction de p_i à p_j
      $\text{(Force,Couple)}_i^{interne}$ += (Force,Couple)
      $\text{(Force,Couple)}_j^{interne}$ -= (Force,Couple)
  Pour chaque particule p_i de la scène
    (Force,Couple)_i = $\text{(Force,Couple)}_i^{interne}$ + actions extérieures
    Intégrer les équations de la dynamique pour calculer
      l'accélération, la vitesse, puis la position.

La seule différence avec un système de particules non-orientées réside dans le fait que les interactions produisent, en plus des forces, des moments de torsion. Donc, l'accélération, la vitesse et la position ont une composante en translation et et une composante en rotation.

4.3 Exemples

Torsion

Cet exemple illustre, entre autres, le mécanisme de typage à l'intérieur d'un objet. Les particules 1 et 3, 4 et 5, et 6 et 7 sont respectivement des types standards A, B et C, alors que les particules 2 et 7 sont des charnières. Les particules de types A et C ont pour loi d'interaction une somme pondérée de l'interaction de cohésion, de l'interaction de frottement et d'une interaction colinéaire d'axe[7] y. La loi d'interaction des particules du type B est composée par la somme pondérée de l'interaction de cohésion,

7. C'est-à-dire que le vecteur normal $\vec{n}$ utilisé pour la définiton de l'interaction est pris égal au vecteur unitaire de l'axe y du repère local de la particule concernée.

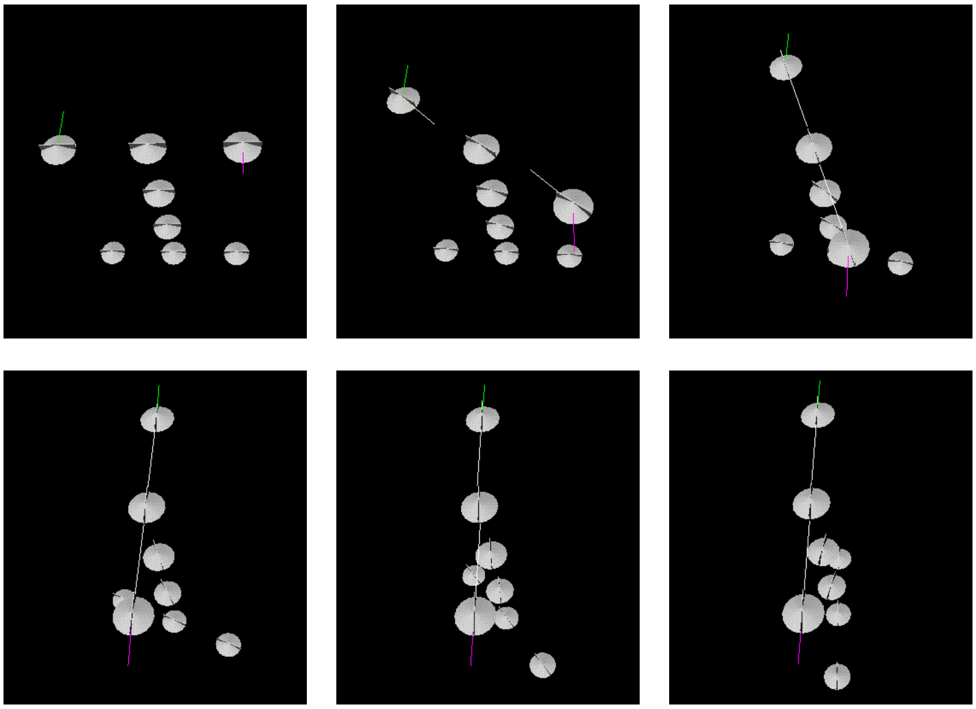


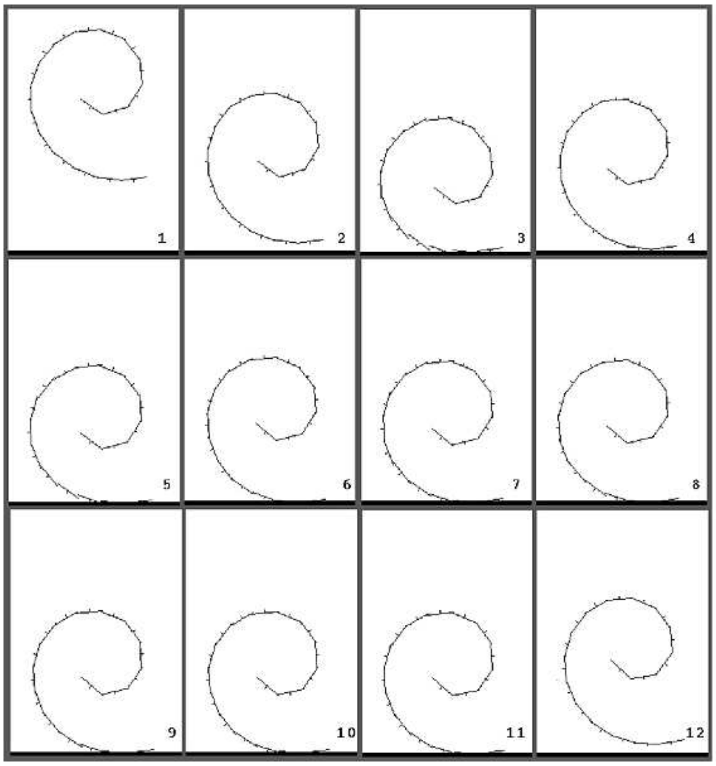
FIG. 4.10 - *Torsion*

de l'interaction de frottement, et d'une interaction coplanaire d'axe x. Les particules charnières ne sont dotées que d'une interaction de cohésion.

La simulation est calculée en appliquant des forces extérieures opposées aux particules 1 et 3. Ces forces, représentées par des traits gris sur la figure 4.10, ont pour effet de faire tourner la barre supérieure de l'objet. Les forces et moments d'interaction (en blanc) agissent pour d'une par maintenir la cohésion de l'objet (les particules sur lesquelles les forces extérieures sont appliquées ne sont pas arrachées à l'objet), et pour lui faire progressivement reprendre sa forme initiale. La dernière image montre le système (objet + forces extérieures) au repos. Les forces produites par l'interaction de cohésion compensent les forces extérieures. Les autres interactions ne produisent plus ni forces ni moments. L'interaction de frottement est nulle puisque les particules sont au repos et ont donc une vitesse nulle. Les interactions colinéaire et coplanaire ne produisent, plus non plus, ni force ni moment car les particules ont atteint une position où les contraintes géométriques imposées par ces interactions sont respectées :

- les particules de type A, ainsi que la charnière 2 qui est dans la zone d'influence de 1 et 3, sont alignées suivant leur axe y.

- Les particules de type B, ainsi que les deux charnières sont dans le plan normal à leur axe x.



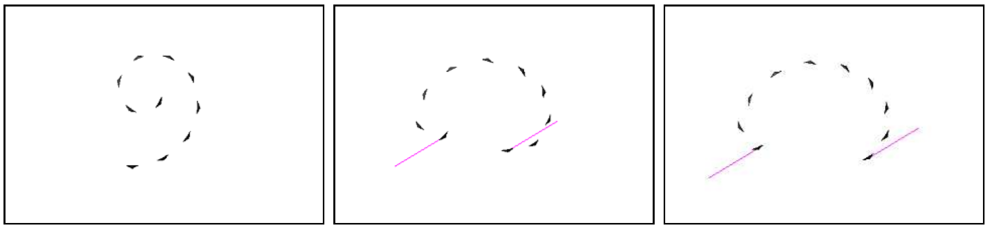
Les particules sont représentées par le plan tangent (un segment en 2D) et la normale

FIG. 4.11 - *Simulation 2D : une spirale tombant sur le sol*

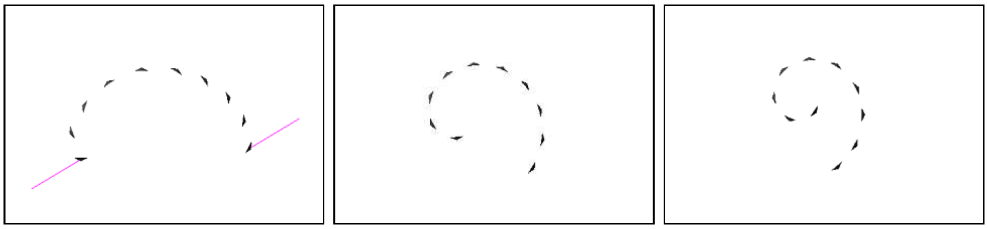
– les particules de type C, ainsi que la charnière 7, sont alignées suivant leur axe y.

Une spirale tombe sur le sol

La figure 4.11 montre un exemple 2D de simulation d'un objet modélisée avec des particules orientées. La loi d'interaction de chaque particule est une somme pondérée des interactions de cohésion, de forme et coplanaire. Chaque particule a une courbure locale spécifique. Plus la particule est proche du centre de la spirale, plus la courbure de son interaction de forme est grande. Les coefficients des forces d'interaction ont été choisis pour modéliser une spirale assez "raide" (les forces d'interaction sont de grande amplitude, donc les déformations sont faibles). L'objet est exposé à un champ de gravité. La collision avec le sol et le contact prolongé qui en découle sont traités avec une méthode de pénalité.
Pendant la simulation, l'objet, initialement dans un état de repos (Figure 4.11 #1), tombe sur le sol (#2). Les forces extérieures appliquées lors de la collision déforment la spirale (#3). À cause de sa raideur, elle rebondit



a) Application des forces extérieures



b) Reprise de la forme initiale

Les traits gris représentent les forces extérieures

FIG. 4.12 - *Déformation de la spirale*

jusqu'à trouver un nouvel état d'équilibre (#4-11) où la position relative des particules crée des forces internes qui compensent les forces externes dues au champ de gravité et au contact avec le sol. En #12, le champ de gravité à été supprimé : l'objet s'est détendu pour reprendre sa forme initiale.

Déformations de la spirale

La spirale précédemment définie (en version 3D cette fois) est maintenant soumise à deux forces extérieures opposées, appliquées à ses extrémités (Figure 4.12a). L'effet de ces forces est de dérouler la spirale. Lorsque le système atteint une position d'équilibre (les forces extérieures sont compensées par les forces d'interaction), la spirale est relâchée (les forces extérieures sont détruites). Sous l'action des forces d'interaction, la spirale reprend progressivement sa forme (Figure 4.12b).

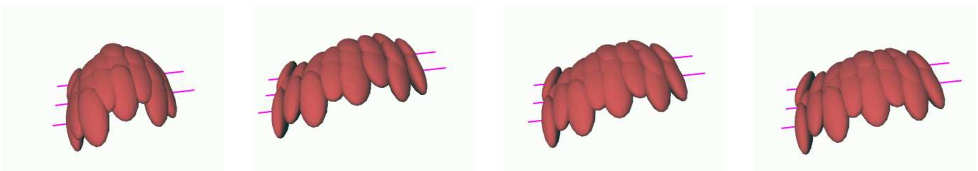
Une portion de cylindre

Un morceau de cylindre est défini par vingt particules (Figure 4.13). Chaque particule est dotée d'une loi d'interaction composée d'une force de cohésion, d'une interaction de frottement et d'une interaction de forme. Cette dernière a une courbure principale nulle (suivant l'axe du cylindre) et l'autre positive.

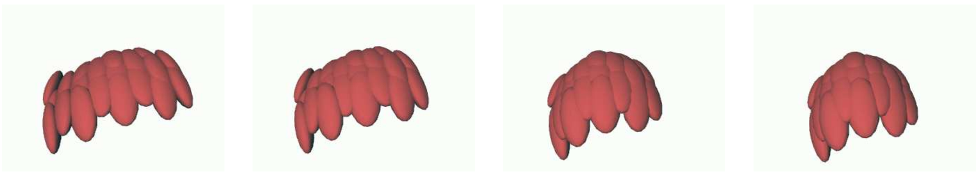
Initialement dans une position d'équilibre, le cylindre est soumis à des forces extérieures opposées s'appliquant sur ses extrémitées (Figure 4.14a). L'effet de ces forces est d'aplatir le cylindre. Lorsque l'état de repos déformé

a) Vue de face b) Vue de profil c) Vue de dessus

FIG. 4.13 - *Définition du morceau de cylindre*



a) Déformation



b) Reprise de la forme initiale

Les forces extérieures sont représentées par des traits gris.
Pour une meilleur compréhension, les zones de repos de l'interaction de cohésion sont montrée plutôt que les particules elles-même.

FIG. 4.14 - *Simulation du cylindre*

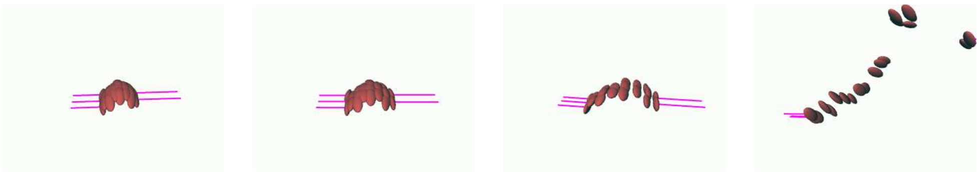


FIG. 4.15 - *Éclatement du cylindre*

est atteint, les forces extérieures sont relâchées (Figure 4.14b). Le cylindre reprend alors sa forme.

La figure 4.15 montre une séquence similaire à celle de la figure 4.14a). Cette fois l'amplitude des forces extérieures a été augmentée, tandis que l'amplitude de la force de cohésion a été diminuée. Le cylindre commence donc à se deformer, puis il se brise en petits morceaux. Dans cette simulation, toutes les particules ont un force de cohésion de même amplitude (d'où la rupture simultanées en plusieurs points), il est possible de simuler des points de rupture en ne diminuant que l'amplitude des forces de cohésion des particules modélisant ces points.

4.4 Conclusion

L'intérêt de la modélisation que nous avons proposée dans cette première partie du chapitre est double. Tout d'abord les objets simulés se déforment tant que l'action extérieure qui provoque la déformation est raisonnable (après ils se brisent ou accusent des déformations irréversibles). Puis lorsque l'action extérieure cesse, ils reprennent peu à peu leur forme au cours du temps, comportement complexe peu aisé à obtenir avec d'autres méthodes. Le fait de ne placer des particules qu'à la surface des objets permet une modélisation plus compacte que pour un système de particules classique.

La visualisation est cependant malaisée. Voir un ensemble de points ne permet plus, dès que la scène devient complexe, de la comprendre. Une solution serait de générer et de maintenir une polygonalisation de l'objet en utilisant par exemple un graphe de Delaunay. Du fait des changements de voisinage (voire de topologie), ceci est assez coûteux.
Une autre solution, très répandue, réside dans l'utilisation des surfaces implicites. Comme nous allons le voir dans la partie suivante, l'utilisation des surfaces implicites apporte non seulement une visualisation de qualité, mais permet aussi de définir un nouveau modèle hybride d'objets hautement déformables.

Partie B :

Particules orientées et surfaces implicites

Les surfaces implicites ont été utilisées par de nombreux auteurs pour modéliser des objets de forme lisse. En 1993, M.-P. Gascuel a présenté une méthode pour modéliser des objets déformables avec des surfaces implicites [Gas93]. Des travaux récents ont utilisé un système de particules non-orientées pour animer les squelettes d'un objet implicite. Cela a permis de modéliser des matériaux hautement déformables (glaise, pâte à modeler, pâte semi-liquide), capables de changements de topologie [DG95]. Pour cela, les problèmes de la répartition des forces de réaction entre les particules et du ré-échantillonnage adaptatif de la surface implicite après un changement de topologie ont été résolus.
Nous commençons cette partie par deux sections (les sections 4.5 et 4.6) qui rappellent certains points importants des travaux évoqués ci-dessus. Puis, à la section 4.7, nous proposons d'étendre ces méthodes en utilisant un système de particules orientées pour supporter les squelettes. Les particules ne sont plus à la surface de l'objet mais à l'intérieur. Les squelettes utilisés pour définir les surfaces implicites sont positionnés dans le repère local de chaque particule. La surface de l'objet est une isosurface des potentiels engendrés par les squelettes. Ceci permet de modéliser un nouveau type d'objets : des objets à mémoire de forme définis par des surfaces implicites. De tels objets combinent les propriétés des deux modèles : mémoire de forme et topologie variable, traitement précis des contacts et visualisation de haute qualité.

4.5 Objets déformables avec des surfaces implicites

Cette section décrit le modèle d'objet déformable présenté par M.-P. Gascuel au SIGGRAPH'93 [Gas93]. Un objet est composé d'une partie rigide modélisée par un ensemble de squelettes dans un référentiel local entourée d'une peau élastique modélisée par une surface implicite. La simulation de l'objet se fait en animant la partie rigide en appliquant les lois de la dynamique du solide puis en calculant la déformation de la peau élastique.

4.5.1 Géométrie de l'objet

Chaque squelette S_i émet un potentiel f_i fonction de la distance à S_i. En chaque point de l'espace, le potentiel $f = \sum f_i$ définit l'objet :

- un point X est à l'intérieur de l'objet si $f(X) > 1$,
- à l'extérieur si $f(X) < 1$, et

– sur la surface de l'objet si $f(X) = 1$.

Les potentiels f_i sont à portée limitée, ce qui permet de ne modifier que localement l'objet lors de la modification d'un squelette. Deux données e et R influent sur la façon dont f modélise l'objet. L'épaisseur d'enrobage e, telle que $f_i(e) = 1$, donne l'épaisseur de la couche de peau déformable autour du squelette. Le rayon d'action R du potentiel influe sur la façon dont deux squelettes mélangent leurs potentiels : R est tel que $\forall x > R,\ f_i(x) = 0$ (Figure 4.16).
Cette technique permet de créer des objets de façon assez intuitive (les squelettes donnent une idée de la forme de l'objet) et permet de développer des outils accessibles aux non spécialistes [TG95].

4.5.2 Déformations

Afin d'obtenir un modèle déformable, il faut définir une correspondance entre force et déformation. Pour de simplifier les calculs, le choix est fait de ne considérer que les déformations radiales[8] (selon l'axe reliant le point du squelette le plus proche du point de contact à ce dernier). Une déformation élastique est du type :

$$d\vec{\mathcal{F}_r} = k(Y)d\vec{Y} \tag{4.5}$$

où $d\vec{\mathcal{F}_r}$ est la force de rappel exercée lors du déplacement infinitésimal $d\vec{Y}$ du point Y à la surface de l'objet. $k(Y)$ est la raideur de l'objet en ce point.

Par définition du gradient :

$$df(\vec{Y}) = \vec{\text{grad}}_{\vec{Y}} f(\vec{Y}) \cdot d\vec{Y}$$

et donc, d'après l'équation 4.5 :

$$df(\vec{Y}) = \vec{\text{grad}} f(\vec{Y}) \cdot \frac{d\vec{\mathcal{F}_r}}{k(Y)} \ . \tag{4.6}$$

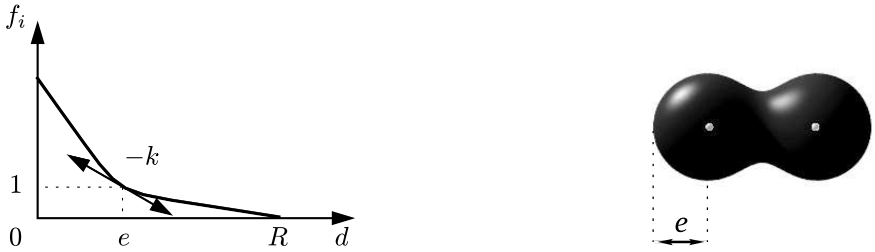
Comme nous l'avons vu dans le paragraphe précédent, la géométrie de l'objet est donnée par l'isosurface de la fonction potentielle f. La façon dont f varie autour de l'isovaleur considérée n'influe pas sur la forme de l'objet. Une astuce de ce modèle consiste à utiliser les variations de f pour coder les caractéristiques physiques (raideur) de l'objet déformable (Figure 4.16). Si l'on pose :

$$\vec{\text{grad}} f(\vec{Y}) = -k(Y)\vec{n}$$

où $\vec{n}$ est la normale à la surface en Y, l'équation 4.6 devient :

$$df(\vec{Y}) = -k(Y)\vec{n} \cdot \frac{d\vec{\mathcal{F}_r}}{k(Y)} \ .$$

8. Les forces tangentielles à l'objet comme par exemple les frottements n'entraînent donc pas de déformations avec ce modèle.



a) Potentiel modélisant une élasticité linéaire

b) Objet modélisé avec deux squelettes ponctuels

FIG. 4.16 - *Définition d'objets déformables implicites*

La variation du potentiel lors la déformation radiale de l'objet faisant passer le point P_0 au point P_1 est :

$$\int_{P_0}^{P_1} df(Y)dY = -\int_{P_0}^{P_1} \vec{n} \cdot d\vec{\mathcal{F}}_r = -\vec{n} \cdot \vec{\mathcal{F}}_r \ ,$$

puisque par définition $\vec{n}$ reste constant pendant une déformation radiale. En nommant $g(P)$ le champ de déformation, la relation :

$$g(P) = -\vec{n} \cdot \vec{\mathcal{F}}_r \tag{4.7}$$

donne immédiatement la correspondance souhaitée entre force et déformation.

4.5.3 Gestion des collisions

Détecter et traiter les collisions entre deux objets implicites nécessite de connaître un échantillonnage d'au moins l'un des deux objets. En effet, en testant les échantillons avec la fonction implicite de l'autre objet, nous pouvons savoir s'il y a interpénétration et donc collision.
L'article [Gas93] propose une méthode originale et peu coûteuse pour maintenir un échantillonnage sur un objet implicite déformable. Les échantillons, ou *graines*, sont initialement calculés sur l'objet non-déformé par une classique méthode de *marching cube* [WMW86, LC87]. Chaque graine (ainsi que la normale à la surface en ce point) est mémorisée dans le repère local de l'objet, donc invariant par rotation et translation de l'objet. En cas de déformation de l'objet, il faut replacer les graines à la surface de l'objet. Les déformations étant radiales et la topologie fixée, il suffit de déplacer chaque graine de la zone déformée suivant la normale et de retrouver la surface de l'objet pour obtenir un échantillonnage valide.

La déformation dans la zone de contact est modélisée par l'adjonction d'un potentiel. Si deux objets O_1 et O_2 sont en collision, la surface de contact est créée en ajoutant respectivement les potentiels de déformation g_{21} et g_{12}

aux potentiels f_1 et f_2 modélisant les objets. En tout point P de la surface de contact, ces potentiels doivent vérifier (f_i^d est la fonction potentielle modélisant l'objet O_i déformé) :

$$\begin{aligned} f_1^d(P) = f_1(P) + g_{21}(P) &= 1 \ , \\ f_2^d(P) = f_2(P) + g_{12}(P) &= 1 \ . \end{aligned}$$

Ce qui donne :

$$\begin{aligned} g_{21}(P) &= 1 - f_1(P) \ , \\ g_{12}(P) &= 1 - f_2(P) \ . \end{aligned}$$

Sur la surface de contact, on a bien $f_1^d(P) = f_2^d(P) = 1$ et cette surface se raccorde bien au reste de l'objet puisque, par définition des g_{ij}, pour tout point P' tel que $f_1(P') = 1$ on a $g_{21}(P') = 0$, et pour tout point P'' tel que $f_1(P'') = 1$, on a $g_{12}(P'') = 0$.

Pour un point P sur la surface de contact, l'équation 4.7 nous permet d'écrire :

$$\begin{aligned} \vec{\mathcal{F}}_{r1} &= -g_{21}(P)\vec{n}_1(P) \ , \\ \vec{\mathcal{F}}_{r2} &= -g_{12}(P)\vec{n}_2(P) \ , \end{aligned}$$

P étant sur la surface de contact, $\vec{n}_1(P) = -\vec{n}_2(P)$ et $g_{21}(P) = -g_{12}(P)$, donc $\vec{\mathcal{F}}_{r1} = -\vec{\mathcal{F}}_{r2}$. Le principe d'action/réaction est bien vérifié.
Les forces de réactions ainsi que les forces de frottement au point de contact sont modulées par l'aire δS de l'élément de surface modélisée par la graine. De ces forces sont déduits une force et un moment de torsion qui sont appliqués au barycentre de l'objet.

4.6 Matériaux hautement déformables

M. Desbrun [Des94] et M.-P. Gascuel ont proposé de modifier le modèle précédemment décrit afin de modéliser non plus des objets élastiques mais des sortes de pâtes plus ou moins fluides, capables de changement de topologie. Ceci est réalisé en utilisant un système de particules pour supporter les squelettes des objets implicites. Deux problèmes principaux doivent être résolus pour permettre cette modélisation : le maintien d'un échantillonnage valide lors des changements de topologie et la répartition aux particules des forces appliquées sur la "peau" implicite de l'objet. Le contrôle de la variation de volume lors des changements de topologie et une gestion de ces derniers sont également traités [DG95].

4.6.1 Échantillonnage adaptatif

Chaque squelette est attaché à une particule. Les squelettes sont échantillonnés individuellement et les graines sont mémorisées dans un repère local lié à la particule. Si aucune précaution n'est prise, le nombre de graines

a) toutes les graines b) les graines valides

FIG. 4.17 - *Invalidation des graines redondantes*

sera très important et certaines d'entre-elles sur-échantillonneront certaines zones de l'objet. Afin d'éviter cette redondance, une politique d'invalidation des graines redondantes est mise en œuvre. Lors de la phase de migration, dès qu'une graine pénètre dans une zone de l'espace où le potentiel émit par le squelette dont elle est issue n'est pas prépondérant, elle est invalidée. En effet, dans ce cas, des graines issues d'autres squelettes (les squelettes dont le potentiel est prépondérant qu point considéré) vont échantillonner la même zone. Un gain important est ainsi réalisé. Dans la figure 4.17, seulement 20 graines sont conservées alors qu'il y en a 40 au total, ce qui implique un gain de 50% sur le temps de traitement des collisions.

4.6.2 Répartition des forces

Dans le modèle initial, les forces exercées à la surface de l'objet étaient transmises sous forme de résultante dynamique au barycentre de l'objet.
Dans le cas de squelettes supportés par un système de particules, il faut répartir les forces sur les particules. Les particules étant des masses ponctuelles, aucune rotation n'est possible.
Les points d'application des forces de contact sont les graines. Le mécanisme d'invalidation des ces dernières garantit que le squelette le plus proche d'un point échantillonné par une graine est celui qui l'a lancée. Cette remarque conduit à une politique de répartition des forces particulièrement simple et peu onéreuse : les forces appliquées sur chaque graine émise par un squelette sont sommées et lui sont appliquées. Le système de particules prend en charge la propagation de la déformation dans la structure interne de l'objet.

Des considérations sur les variations de volume liées au déplacement des particules modélisant l'objet ainsi que des solutions à ce problème sont présentées dans [DG94, DG95].

4.7 Objets hautement déformables

Tout en enrichissant la gamme de comportements modélisés, le modèle de matériau hautement déformable [DG94] impose certaines simplifications du modèle initial d'objet déformable [Gas93] :

– les squelettes sont portées par des masses ponctuelles, il n'y a donc pas de notion de rotation des squelettes. De ce fait, l'utilisation de squelettes non ponctuels n'est plus pertinente (plus exactement, il faut utiliser des squelettes à symétrie sphérique).

– Pour la même raison, il n'est pas envisageable d'utiliser des fonctions potentiel anisotropes.

– Les objets modélisés n'ont pas de forme propre, d'où l'appellation de matériau plutôt que d'objet.

L'utilisation d'un système de particules orientées pour supporter les squelettes [9] apporte une solution à ces problèmes.

4.7.1 Modification du modèle

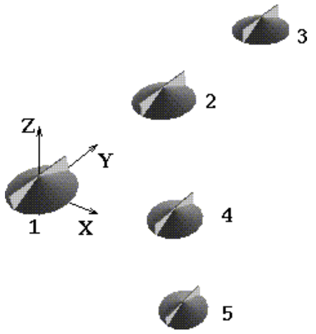
Comme les particules orientées sont des référentiels, les utiliser comme support des squelettes de la surface implicite permet plusieurs améliorations. Des fonctions potentiels anisotropes [WW89] peuvent être utilisées à la place des fonctions de distances utilisées dans [Gas93, DG94]. Chaque squelette peut être un objet complexe défini par une collection de primitives placées dans le référentiel de la particule. Et enfin, les interactions décrites permettent de modéliser des objets ayant une forme propre, capables de se déformer aussi bien au niveau de l'enrobage, comme les objets décrits dans [Gas93], qu'au niveau de la structure interne, et capable de changements de topologie.

Les principes mis en œuvre sont les mêmes que pour le matériau hautement déformable décrit dans [DG94, DG95], au détail prêt que l'application des forces de contact (collisions et frottements) sur les graines se traduit par l'application d'une force et d'un moment résultants sur la particule orientée.

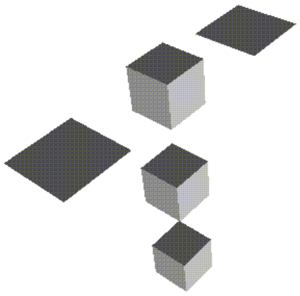
4.7.2 Exemples

La figure 4.19 montre une séquence d'images extraites de la simulation de la chute d'un objet modélisé avec des particules orientées sur un obstacle implicite rigide. Le “T” est composé de cinq particules orientées (Figure 4.18a). Le squelette associé à chaque particule est composé d'une seule primitive, un cube ou un carré (Figure 4.18b). La loi d'interaction de chaque particule

9. Les squelettes sont définis dans les repères locaux que sont les particules orientées.



a) Position des particules orientées



b) Squelettes correspondants

FIG. 4.18 - *Définition de l'objet*

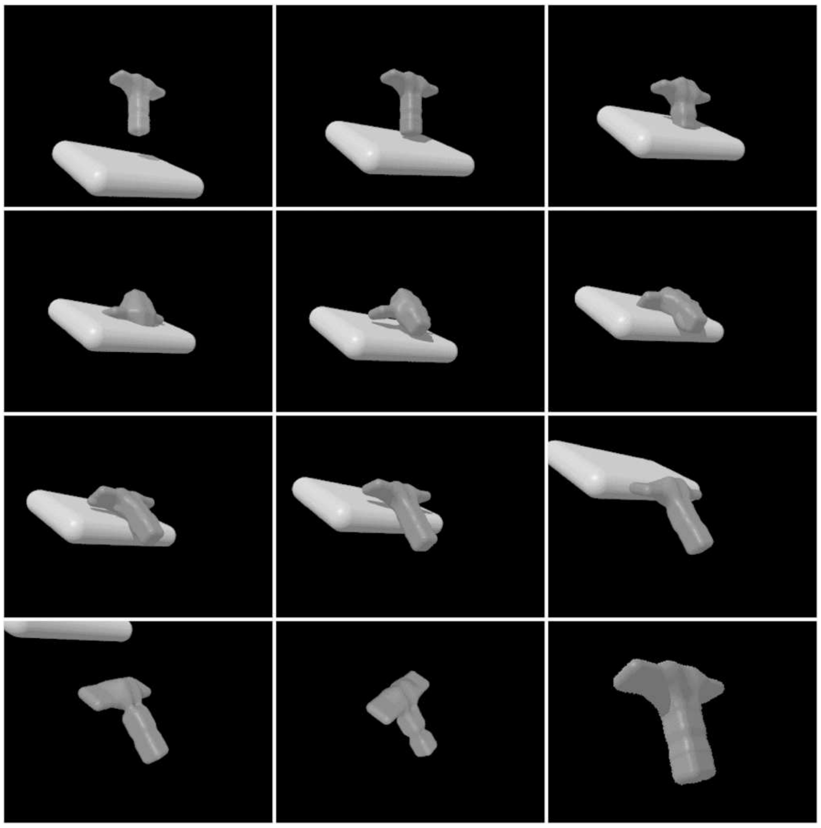


FIG. 4.19 - *Simulation 3D: chute du* T

FIG. 4.20 - *Définition des objets*

est la somme des interactions de cohésion, de frottement et d'une interaction coplanaire d'axe X. Les particules 1 et 3 disposent en plus d'une interaction colinéaire d'axe Y, alors que les particules 4 et 5 ont une interaction colinéaire d'axe Z. Les coefficients des différentes interactions ont été choisis pour modéliser un objet mou (les déformations sont grandes). Lors de la collision, l'objet s'écrase, rebondit, puis glisse le long de l'obstacle. Ce glissement a pour effet de faire tourner la barre horizontale du “T” autour de son axe. La dernière image de cette séquence a été prise plus tard, lorsque l'objet a retrouvé sa forme initiale.

La figure 4.21 montre des images extraites de la bande vidéo montrée à *Graphics Interface '95* [LP95b].
Le *I* est modélisé par quatre particules, dotées chacune d'une interaction de cohésion, d'une interaction de frottement et d'une interaction colinéaire d'axe Z. Le G est, lui, modélisé par dix particules, dotées chacune d'une interaction de cohésion, d'une interaction de frottement, d'un interaction coplanaire d'axe Z, et d'une interaction cosphérique. Le rayon de la sphère est fixé pour chaque particule : plus la particule est située à l'intérieur de la spirale formée par le G, plus le rayon est petit. Chaque particule supporte un squelette carré ou cubique (voir la figure 4.20 où les particules sont représentées sur l'image de gauche, alors que les squelettes sont montrés sur l'image de droite.).

Le sol est rigide et a une position fixée dans l'espace, alors que la balle est un objet mobile déformable du type [Gas93].
Le *GI* est initialement dans une position de repos, posé sur le sol. La balle est lancée avec une vitesse initiale horizontale. Tous les objets sont soumis à la gravité. La balle écrase le I sur le G. Le I est alors projeté en arrière, alors que le G et la balle tombent sur le sol. Le fait que la balle heurte le sommet du I sur l'avant de sa face droite induit un mouvement de rotation. La rotation se propage progressivement à toutes les particules.

La figure 4.22 montre une séquence réalisée lors de la mise au point de la séquence précédente. Ces images sont directement issues de copies d'écrans. Le rendu est celui proposé en interactif par *Fabule* (voir le cha-

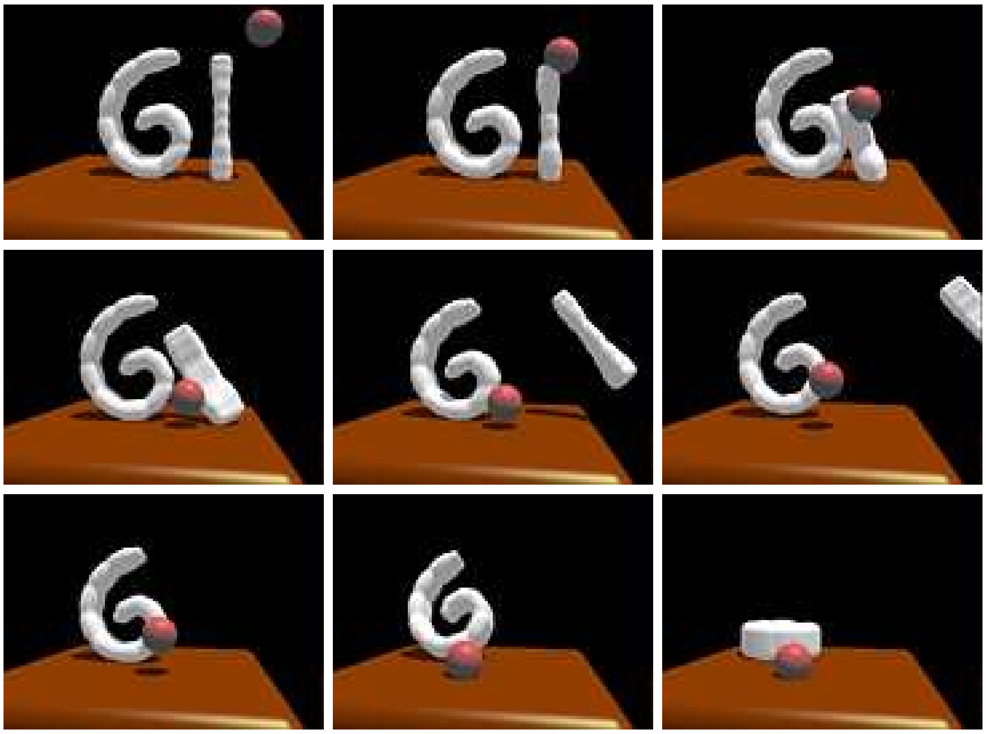


FIG. 4.21 - *Vidéo présentée à* Graphics Interface '95

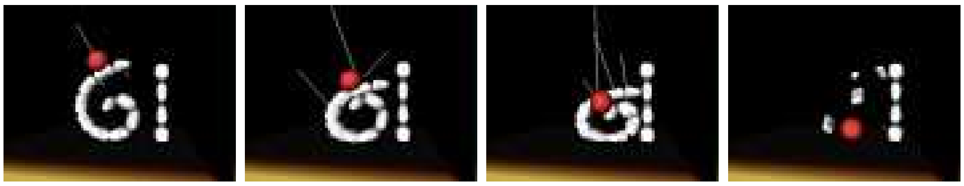


FIG. 4.22 - *Explosion du* G

pitre 6, page 91). Ici, les paramètres physiques (notamment la masse et la vitesse) de la balle sont tels que le G explose sous le choc. Les forces extérieures appliquées par la balle sont supérieures au forces d'interaction (montrées par des vecteurs gris sur la figure).

Partie C :
Conclusion

Nous avons présenté dans ce chapitre deux modèles d'objets déformables reposant sur l'utilisation d'un système de particules orientées.

L'un, utilisant uniquement des particules orientées, est plutôt défini pour la modélisation surfacique d'objets, bien que ce soit possible de l'utiliser pour faire une modélisation volumique (c'est-à-dire en plaçant aussi des particules à l'intérieur de l'objet). En effet, l'interaction de forme permet de maintenir les courbures locales d'une surface. Chaque particule échantillonne donc un élément de surface, en conservant non seulement la position et la tangente de la surface en ce point, mais aussi des propriétés plus complexes telles que les courbures normales. Augmenter l'information en chaque point permet de diminuer le nombre d'échantillons nécessaire pour reproduire la surface. L'utilisation de rayons d'influence et de rayons de repos dépendant de la direction est un concept nouveau qui permet d'adapter vraiment l'échantillonnage en fonction de la configuration locale de la surface, et donc de diminuer encore le nombre d'échantillons nécessaires pour obtenir un modèle de qualité donnée. Le comportement des objets obtenus est cependant complexe, puisqu'il sont capables de déformations autour d'une forme de référence et de fracture.
L'inconvénient de ce modèle est celui de tous les modèles discrets, les collisions avec les autres objets ne peuvent être détectés qu'aux points de discrétisation. L'utilisation de systèmes de particules pose aussi le problème de la visualisation. Il est difficile d'appréhender la géométrie d'un objet tridimensionnel lorsqu'il est visualisé comme un ensemble de points répartis dans l'espace, et le maintien d'une polygonisation est coûteux du fait de la possibilité de changement de voisinage des particules.

Contrairement au premier modèle qui est entièrement nouveau, le second est une amélioration de modèles existants. Ce modèle est hybride. La surface des objets est continue, et elle gère les interactions de l'objet avec le monde extérieur. L'utilisation de particules orientées comme support des squelettes, permet de modéliser des objets implicites à squelette souple. Ces nouveaux objets sont capables de déformations superficielles élastiques (l'application de forces de faible amplitude à la surface de l'objet n'engendre que des déformations de l'enrobage implicite), de déformations plus importantes (des forces de plus grande amplitude vont déformer le squelette de l'objet) et de fractures. Cette diversité de comportement, la capacité de générer des surfaces de contact exactes, la possibilité d'avoir une visualisation de très haute qualité (par lancer de rayons par exemple) rendent ce nouveau modèle très intéressant. La gestion des surfaces implicites exige cependant des coûts en temps de calcul assez importants.

La définition manuelle des objets (placer les particules, définir les interactions) est fastidieuse. Afin d'automatiser ce processus, nous avons défini une méthode de reconstruction d'objets d'après des données géométriques tridimensionnelles. Cette méthode est exposée dans le chapitre suivant.

Chapitre 5

Reconstruction d'objets à partir de données 3D

Nous proposons une méthode pour créer un modèle discret d'objet déformable à partir de sa définition géométrique implicite. Nos tests sont effectués avec des données fictives de type surface implicite. Un de nos objectifs étant de pouvoir à terme traiter les images 3D de type médical, nous faisons le choix de rester aussi généraux que possible. Nous prenons comme hypothèse de départ :

> *l'espace est le lieu d'un champ de potentiel scalaire dont la valeur en chaque point X est donnée par $\mathcal{F}(X)$, et l'objet est défini par une isovaleur C de ce potentiel.*

Aucune autre supposition n'est faite sur la fonction qui modélise le potentiel. Les fonctions potentiel utilisées pour tester les méthodes développées dans les paragraphes suivants sont des fonctions monotones de la distance à un ensemble de squelettes.

Nous présentons ici les deux parties constituant la méthode qui, bien que conceptuellement dissociées, sont simultanées : le positionnement des particules sur la surface de l'objet et la définition des interactions entre les particules pour obtenir un modèle déformable indépendant des données de départ.

5.1 Échantillonner l'objet

Afin de conserver le maximum de généralité, l'approche choisie est la suivante : un ensemble de particules initiales est placé dans une région de l'espace proche de l'objet (Figure 5.1, une seule particule initiale est placée au centre de l'objet). Le potentiel est utilisé pour en dériver un champ de forces et un champ de moments de torsion dont l'effet est de placer ces particules tangentiellement à la surface de l'objet. Des algorithmes de génération de nouvelles particules sont mis en œuvre pour couvrir l'intégralité

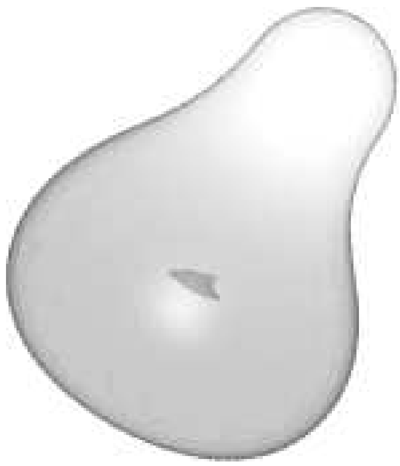


FIG. 5.1 - *Initialisation de l'échantillonnage*

de l'objet.

5.1.1 Champ de placement

L'objet est caractérisé par une fonction $\mathcal{F}(X)$ définie en tout point X de l'espace et une isovaleur C de cette fonction. La surface de l'objet est le lieu des points tels que $\mathcal{F}(X) = C$, où C est une constante. À partir de ce potentiel, une énergie de placement E est définie :

$$E(X) = (\mathcal{F}(X) - C)^2 \ . \tag{5.1}$$

L'énergie est minimale à la surface de l'objet. Un champ de forces $\vec{F}$ et un champ de moments de torsion $\vec{\mathcal{M}}$ sont dérivés de E :

$$\vec{F}(X) = -\vec{\mathrm{grad}}(E(X)) \ , \tag{5.2}$$

$$\vec{\mathcal{M}}(X) = \frac{(\vec{n} \cdot \vec{\mathrm{grad}}(E(X)))(\vec{n} \wedge \vec{\mathrm{grad}}(E(X)))}{\|\vec{\mathrm{grad}}(E(X))\|^2} \tag{5.3}$$

où $\vec{n}$ est le vecteur normal de la particule positionnée en X.

L'effet conjugué de ces deux champs est de positionner les particules sur la surface de l'objet, leurs normales étant normales à la surface. Toute particule initialement placée dans une portion de l'espace où les champs définis ci-dessus s'exercent se placera d'elle-même sur l'objet pendant la simulation dynamique. Chaque particule fournit un point de la surface de l'objet ainsi que la normale à l'objet en ce point. Ceci n'est cependant pas suffisant pour échantillonner un objet. Il faut que ces points soient régulièrement espacés. Cet espacement doit pouvoir être contrôlé.

5.1.2 Répartir les points d'échantillonnage

L'utilisation d'un système de particules permet très facilement d'obtenir une répartition "uniforme" [1] de points d'échantillonnage. Il suffit d'introduire une loi d'interaction de type répulsion entre les particules [Tur92, WH94]. Les particules s'éloigneront les unes des autres jusqu'à ce que les répulsions des diverses particules se compensent.

Nous avons choisi d'utiliser une loi d'interaction de type attraction/répulsion. Notre objectif final est d'obtenir un modèle de l'objet indépendant de sa représentation initiale. En effet, les données de type médical sont très volumineuses (typiquement un cube de 252^3 pixels, la couleur de chaque pixel étant codée sur un ou deux octets). Leur manipulation demande donc du temps et de la mémoire. Le modèle de l'objet reconstruit doit pouvoir être utilisé sans les données de départ. Il faut donc que les particules à la surface de l'objet soient dans une position de repos même lorsque le champ de placement est supprimé. Une des premières conditions est que le système soit stable par rapport à la force d'attraction/répulsion, ce qui est impossible avec une pure répulsion.

De plus, le rayon de repos de l'interaction d'attraction/répulsion fournit un moyen direct pour contrôler la densité de l'échantillonnage. Un grand rayon de repos a pour effet d'éloigner les particules. L'échantillonnage obtenu est donc peu dense. À l'opposé, un petit rayon de repos produit un échantillonnage dense et donc beaucoup d'échantillons. Il n'est pas très intéressant d'échantillonner un objet uniformément. En effet, avec un maillage uniforme il faut soit mailler très fin pour avoir les détails de l'objet et dans ce cas la masse de données est trop importante, soit mailler plus grossièrement et les détails sont perdus. En règle générale, il est souhaitable d'avoir plus d'échantillons dans les zones accidentées que dans les zones planes. Pour obtenir cet effet, il suffit d'indexer le rayon de repos des particules sur la courbure de la zone de surface qu'elles échantillonnent, avec un rayon plus faible (et donc plus de particules) lorsque la courbure est plus forte.
L'utilisation de la zone d'influence ellipsoïdale et de la nouvelle force de cohésion avec rayon de repos dépendant de la direction nous permet de franchir encore une étape : proposer un échantillonnage adapté suivant les directions. Après avoir montré comment nous calculons les courbures directionnelles en chaque point d'échantillonnage, nous exposerons notre méthode pour adapter l'échantillonnage suivant les directions.

5.1.3 Génération automatique de nouvelles particules

Les techniques exposées précédemment permettent d'obtenir un échantillonnage de l'objet adapté à la courbure locale, mais il demande à l'utilisateur de créer lui-même les particules. Afin d'obtenir un processus automa-

1. Les échantillons sont répartis régulièrement.

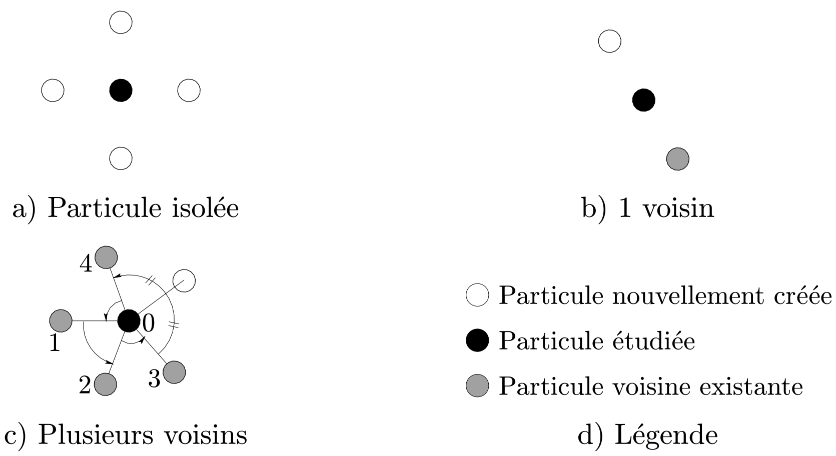


FIG. 5.2 - *Politique de création des nouvelles particules*

tique de reconstruction de l'objet, nous développons certaines heuristiques de création de particules.
Afin de disposer d'un critère d'arrêt simple pour l'algorithme de génération, nous faisons le choix de ne jamais détruire de particules. Ainsi, l'algorithme est stoppé lorsqu'il est impossible de créer de nouvelles particules.

Où créer des particules?

L'objectif étant de créer un échantillonnage régulier[2], il faut ajouter des particules là où elles manquent. Une solution rapide serait de créer directement des particules à la surface de l'objet. Cependant, nous ne connaissons pas la position de la surface. Il nous faudrait tester la valeur du potentiel en tout point de l'espace et créer une particule là où le potentiel est proche de C et où il n'y a pas de particules.

Par contre, les particules existantes sont soumises à l'action du champ de placement. Les particules stables se trouvent dans une zone où ce champ est nul, donc à la surface de l'objet[3]. Les nouvelles particules doivent donc être créées au voisinage de celles déjà existantes. Pour éviter que la création de nouvelles particules n'entraîne l'explosion du système, il faut que le nouveau système soit proche de l'équilibre.

Pour générer automatiquement des nouvelles particules, il faut détecter, parmi les particules existantes stables, celles qui n'ont pas un voisinage *complet*: sur un plan, l'état stable d'un système de particules homogènes (toutes ont le même rayon de repos) forme un pavage hexagonal. Chaque particule

2. Toute la surface doit être échantillonnée, sans qu'il y ai d'amas de particules, ni de trous dans l'échantillonnage.

3. Sauf dans le cas où il y a eu une explosion du système et que des particules ont été rejetées en dehors de la zone d'application du potentiel définissant l'objet.

a six voisines formant six triangles équilatéraux. Nous appelons voisinage complet, une telle configuration.

Chaque particule P est traitée successivement. La première étape de l'algorithme de génération de particules est le calcul du voisinage. Toute particule P_i dans la zone d'influence de P est considérée comme étant voisine de P.
La projection P'_i de chaque voisine dans le plan tangent à la particule P est calculée. Ramené à cette configuration plane, l'objectif est de créer des particules pour se rapprocher de la configuration idéale tout en restant attentif à ne pas en créer trop.
Toutes les particules sont créées dans le plan tangent à la particule étudiée, à une distance égale à sa distance de repos. Trois cas sont distingués suivant le nombre de voisins (Figure 5.2) :

- les particules isolées,
- les particules n'ayant qu'un voisin, et
- les autres.

Dans le premier cas, quatre particules sont créées[4] équidistantes deux à deux. Dans le second cas, une seule particule symétrique de la voisine par rapport à la particule étudiée est créée. Dans le cas général, les particules doivent être créées dans les *trous*[5] du voisinage. Ces trous sont détectés en évaluant l'angle entre deux particules voisines. Ceci se fait en plusieurs étapes. Tout d'abord, l'angle entre chaque P'_i et un axe arbitraire (dans notre cas, l'axe Ox de la particule P) sont calculés. Après avoir trié ces angles, en ordre croissant par exemple, il suffit d'évaluer la différence entre deux angles consécutifs pour obtenir l'écart angulaire entre deux particules voisines projetées dans le plan tangent à P. Cet écart angulaire est ensuite testé par rapport à un seuil : si l'écart est plus grand que le seuil, alors il y a un *trou* et une particule placée sur la bissectrice à une distance r_0 est candidate à la création. Le seuil choisit est $\frac{2\pi}{5}$. Sur l'exemple de la figure 5.2c, le classement des particules donne 4, 1, 2, 3. Les écarts entre 4 et 1, 1 et 2, 2 et 3 sont jugés insuffisants pour la création d'une particule. Par contre, l'écart entre 3 et 4 montre l'existence d'un *trou*, particule est donc créée sur la bissectrice entre 3 et 4.

4. Seulement quatre particules sont créées et non six comme on aurait pu s'y attendre, car dans des zones de forte variation de courbure (et donc de rayon d'action des particules), le motif de base d'un ensemble stable de particules n'est plus forcement le motif où une particule a six voisines placées au sommet d'un hexagone régulier dont elle est le centre. Ce choix de quatre particules est arbitraire, l'idéal serait de calculer le nombre de particules en fonction des courbures locales.

5. C'est-à-dire les zones où il n'y a pas déjà des voisins.

Remarque : Le calcul des angles est coûteux. Dans notre contexte (la reconstruction est faite une fois pour toute), nous ne nous sommes pas intéressés à l'optimisation de ce processus. Ce principe de détection des trous peut être appliqué en utilisant, par exemple les tangentes des angles, ce qui est moins coûteux.

Avant la création effective d'une particule, un dernier test est effectué afin d'éviter de surcharger une zone de l'espace. Il ne faut pas créer une particule trop proche de particules déjà existantes. D'une part, deux particules trop proches risquent d'augmenter l'énergie du système de façon incontrôlée sous l'action des forces de cohésion, et d'autre part, il faut veiller à ne pas créer trop de particules. Ce test est simple : seules sont créées les particules qui n'ont pas de voisine à moins de $\frac{r_0}{2}$.

Nous savons maintenant où placer les nouvelles particules. Il reste à déterminer quand les créer.

Quand créer des particules ?

Afin d'automatiser le processus de création de particules, il faut décider de l'instant de cette création, c'est-à-dire comment sélectionner les particules à étudier. Comme indiqué précédemment, cette création ne doit avoir lieu qu'autour des particules qui sont à la fois proches de la surface mais aussi, et ce pour éviter que le système n'oscille trop, relativement stables. Une donnée physique permet de quantifier ceci : l'énergie mécanique. En effet l'énergie mécanique d'un objet est la somme de son énergie cinétique et de son énergie potentielle. L'énergie cinétique est un marqueur évident de la stabilité d'une particule à un instant donné alors que l'énergie potentielle permet de prédire l'évolution de son état. Les seules forces externes appliquées au système sont les forces de placement. Ces forces et moments sont nuls à la surface de l'objet. L'énergie potentielle du système est donc bien minimale quand les particules sont à la surface de l'objet. Les forces et moments de placement dérivant d'une énergie de placement (Section 5.1.1, équation 5.1), l'énergie potentielle en un point n'est autre que cette énergie de placement en ce même point. Cette énergie est nulle à la surface de l'objet.

$$
\begin{aligned}
E_c &= \frac{1}{2} m V^2 + \vec{\omega} \cdot J\vec{\omega} \ , \\
E_p &= E(X) = (\mathcal{F}(X) - C)^2 \ , \\
E_m &= E_c + E_p \ .
\end{aligned}
$$

Pour créer le modèle de la surface, à intervalles de temps réguliers, l'énergie mécanique de chaque particule est calculée. Si elle est inférieure à un certain seuil (nous utilisons 10^{-3} Joules) alors la particule est candidate à l'extension de son voisinage.

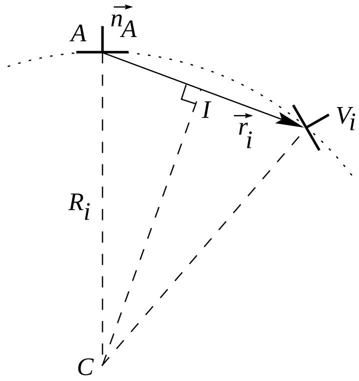


FIG. 5.3 - *Calcul du rayon de la sphère tangente à A et passant par V_i*

5.2 Définir les interactions entre les particules

Afin d'obtenir un modèle déformable de l'objet reconstruit, il nous faut définir les interactions entre les particules. Pour ce faire, il nous faut calculer, d'après la configuration du système, les directions et courbures principales de chaque particule. Puis nous en déduirons une distance de repos.

Les particules ont trois types d'interaction avec leur voisines :

- une interaction de cohésion qui permet de maintenir une distance de repos entre elles. Cette distance de repos est propre à chaque particule et dépend de la courbure locale de l'objet à la position de la particule.
- Une interaction de frottement. Elle modélise un frottement fluide entre deux solides. Cette interaction s'oppose à tout mouvement relatif (translation et rotation) entre les particules. Elle se traduit par une force proportionnelle au carré de la vitesse relative et un moment proportionnel au carré de la vitesse de rotation relative.
- Une interaction de forme. Celle-ci a pour rôle de maintenir la courbure locale des particules.

Plusieurs de ces interactions font intervenir la notion de *courbure locale*. Le paragraphe suivant précise ce terme et indique la méthode de calcul employée.

5.2.1 Courbure locale

Dans ce paragraphe, nous exposons comment calculer les caractéristiques d'une particule A de normale $\vec{n}_A$ ayant n voisines V_i. La translation entre les points respectivement occupés par A et par V_i est notée $\vec{r}_i$.

Il n'y a qu'une sphère tangente à la particule A et passant par le point occupé par la particule V_i. Soient C le centre de cette sphère, R_i son rayon

et I le milieu du segment $[AV_i]$. C est tel que $\vec{OC} = \vec{OA} - R_i\vec{n_A}$. C est l'intersection de la droite $(A, \vec{n_A})$ avec le plan perpendiculaire à $\vec{r_i}$ passant par I (Figure 5.3).
Le rayon de la sphère s'obtient par :

$$R_i = -\frac{\vec{r_i} \cdot \vec{r_i}}{2\vec{n_A} \cdot \vec{r_i}} \ . \tag{5.4}$$

Démonstration :

$$\begin{aligned}
\vec{IC} \cdot \vec{r_i} = 0 \ &\Leftrightarrow \ (\vec{OC} - \vec{OI}) \cdot \vec{r_i} = 0 \\
&\Leftrightarrow \ (\vec{OA} - R_i\vec{n_A} - \frac{\vec{OA} + \vec{OV_i}}{2}) \cdot \vec{r_i} = 0 \\
&\Leftrightarrow \ (\frac{-\vec{r_i}}{2} - R_i\vec{n_A}) \cdot \vec{r_i} = 0 \\
&\Leftrightarrow \ (\frac{\vec{r_i}}{2} \cdot \vec{r_i}) + (R_i\vec{n_A} \cdot \vec{r_i}) = 0 \\
&\Leftrightarrow \ R_i = -\frac{\vec{r_i} \cdot \vec{r_i}}{2\vec{n_A} \cdot \vec{r_i}} \ \blacksquare
\end{aligned}$$

Pour chaque voisin V_i, nous pouvons donc calculer une orientation φ_i dans le plan perpendiculaire à $\vec{n}_A$ (voir la section 4.1.2, page 52) et une courbure $\rho_i = 1/R_i$ (d'après l'équation 5.4).

Les caractéristiques φ_A, ρ_A^{Max} et ρ_A^{min} de A sont tout d'abord estimées. Par définition des courbures principales :

$$\begin{aligned}
\rho_A^{Max} &= \max_{i=1..n} \rho_i \ , \\
\rho_A^{min} &= \min_{i=1..n} \rho_i \ , \\
\varphi_A &= \varphi_i \text{ avec } \ i \text{ tel que } \ \rho_A^{Max} = \rho_i \ .
\end{aligned}$$

Nous définissons l'erreur :

$$\mathcal{E} = \sum_{i=1}^{n} (\rho_A^{Max} \cos^2(\varphi_A - \varphi_i) + \rho_A^{min} \sin^2(\varphi_A - \varphi_i) - \rho_i)^2$$

qui est la somme des carrés des différences entre les courbures calculées pour chaque particule V_i avec l'équation 5.4, et celles données par l'équation aux courbures principales en utilisant la valeur courante des caractéristiques de la particule A.

Pour raffiner l'estimation, nous effectuons une descente de gradient itérative. À chaque pas, les nouvelles valeurs des caractéristiques de A (φ_A,

ρ_A^{Max} et ρ_A^{min}) sont déduites des valeurs précédentes par :

$$\begin{aligned}
\varphi'_A &= \varphi_A - \gamma \frac{\partial \mathcal{E}}{\partial \varphi_A} \,, \\
\rho_A^{Max\prime} &= \rho_A^{Max} - \gamma \frac{\partial \mathcal{E}}{\partial \rho_A^{Max}} \,, \\
\rho_A^{min\prime} &= \rho_A^{min} - \gamma \frac{\partial \mathcal{E}}{\partial \rho_A^{min}} \,.
\end{aligned}$$

Le processus itératif est stoppé dès que l'erreur tombe en dessous d'un seuil (10^{-3} dans nos exemples). Après une étude expérimentale de la vitesse de convergence, nous avons choisi $\gamma = 0.1$. Avec un γ plus grand, le nombre de cas où l'erreur ne converge pas augmente sensiblement. Avec un γ plus petit, la convergence est plus lente sans que la différence entre le nombre des cas où l'erreur diverge soit sensible. Dans les cas où l'erreur diverge, la surface échantillonnée par la particule A est considérée comme localement plane (courbures nulles).

5.2.2 Adapter l'échantillonnage

Intuitivement, un bon échantillonnage doit être dense dans les régions accidentées et clairsemé dans les régions planes. Comme le suggère G. Turk dans [Tur92], il est possible d'aller plus loin en postulant que l'échantillonnage doit être dense dans les directions à forte courbure et épars dans les directions à faible courbure. Notre modèle permet de prendre ceci en compte de façon très simple.
Maintenant que chaque point d'échantillonnage (c'est-à-dire chaque particule) est doté de deux courbures principales, nous allons utiliser ces dernières pour adapter la densité de l'échantillonnage suivant les directions. Automatiser ceci demande de répondre à deux questions :

- quels sont les paramètres significatifs pour l'utilisateur ?
- Comment projeter l'espace des courbures sur l'espace des rayons de repos ?

Le paramètre le plus intuitif est sans doute la distance souhaitée entre deux échantillons. Il fixe l'échelle de la reconstruction. L'utilisateur doit aussi pouvoir régler le niveau de détail de la reconstruction : ce niveau est fixé par la distance minimale de repos entre les particules (les détails plus petits ne seront pas reconstruits, cette donnée permet d'éliminer le bruit des données de départ). Finalement, l'utilisateur doit pouvoir fixer la distance maximale entre les particules, afin de ne pas créer de trous (en cas de simulation conjointe du modèle reconstruit et d'autres objets, les collisions entre le modèle et les objets plus petits peuvent ne pas être détectées, ce qui a le désagréable effet de permettre aux objets de se passer au travers les uns des

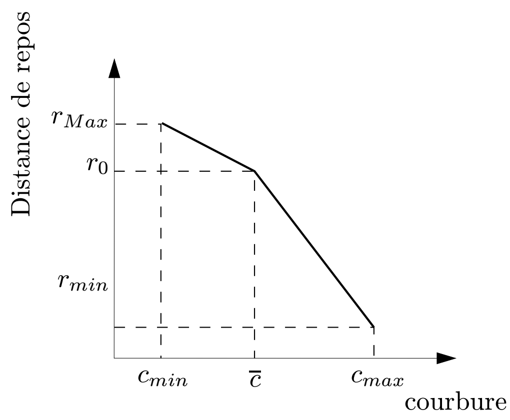


FIG. 5.4 - *Distance de repos en fonction de la courbure locale*

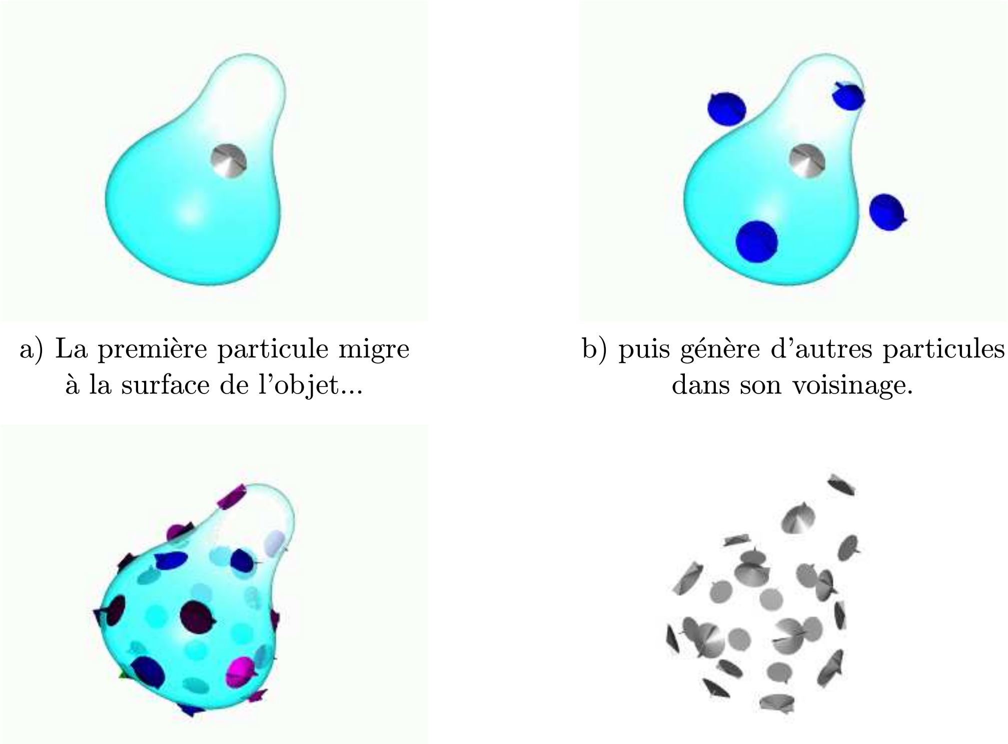
autres...).
Afin de contrôler la qualité de l'échantillonnage, nous fournissons donc trois paramètres-utilisateur : un rayon maximal r_{Max}, un rayon moyen r_0 et un rayon minimal r_{min}. Une zone plane sera échantillonnée de façon à ce que les échantillons soient distants de r_{Max}. La distance minimale entre les échantillons (dans les zones de forte courbure) est r_{min}. r_0 est la distance *idéale* entre les particules, c'est-à-dire la distance si l'objet est une sphère (la courbure est constante).

Afin d'adapter l'échantillonnage suivant les directions, pour chaque particule, deux rayons de repos (un suivant chaque direction principale) sont calculés, puis injectés dans l'interaction de cohésion directionnelle. De ces rayons de repos sont déduits des rayons d'action maximaux, utilisés pour les autres interactions. Nous utilisons $r_{Max} = \gamma r_0$, et dans nos exemples $\gamma = 2,5$.

La fonction de transfert entre l'espace des rayons de repos et l'espace des courbures utilisée est linéaire par morceaux (Figure 5.4). Tout d'abord les valeurs minimale c_{min}, moyenne $\overline{c}$ et maximale c_{Max} des valeurs absolues des courbures locales sont évaluées.

$$\begin{aligned} c_{min} &= \min_{p\in\mathcal{P}} |\rho_p^{min}| \ , \\ \overline{c} &= \frac{1}{n}\sum_{p\in\mathcal{P}} \frac{|\rho_p^{min}| + |\rho_p^{Max}|}{2} \ , \\ c_{Max} &= \max_{p\in\mathcal{P}} |\rho_p^{Max}| \ , \end{aligned}$$

où ρ_p^{Max} et ρ_p^{min} sont respectivement les courbures maximale et minimale de la particule p, et $\mathcal{P}$ est l'ensemble des particules définies. Ensuite les rayons de repos suivant les directions principales ($\vec{u}$) sont tous deux calculés en



a) La première particule migre à la surface de l'objet...

b) puis génère d'autres particules dans son voisinage.

c) Reconstruction en cours

d) objet reconstruit

FIG. 5.5 - *Reconstruction d'une poire implicite*

fonction de la courbure principale concernée ($\rho_{\vec{u}}$) en appliquant :

$$r_{\vec{u}} = \begin{cases} r_{Max} - \dfrac{|\rho_{\vec{u}}| - c_{min}}{\overline{c} - c_{min}}(r_{Max} - r_0) & \text{si} \quad |\rho_{\vec{u}}| < \overline{c} \\ r_{min} - \dfrac{|\rho_{\vec{u}}| - c_{Max}}{\overline{c} - c_{Max}}(r_{min} - r_0) & \text{sinon} \; . \end{cases} \tag{5.5}$$

5.3 Exemples

L'objet à reconstruire est une poire modélisée avec des surfaces implicites (Figure 5.1). Une particule initiale est placée au centre de l'objet. Sous l'action des champs de placement, elle se place à la surface de l'objet (Figure 5.5a). Une fois que son énergie mécanique est faible (la particule est à la surface de l'objet et sa vitesse est faible), elle est sélectionnée comme base pour générer d'autres particules autour (Figure 5.5b). Le processus se poursuit (Figure 5.5c), jusqu'à ce que lobjet soit entièrement recouvert de particules. À ce moment, la reconstruction s'arrête, et le système de particules obtenu est une approximation discrète déformable de l'objet de départ (Figure 5.5d).

5.4 Conclusion

Une telle technique de segmentation de données présente plusieurs avantages :

- le modèle obtenu est une représentation compacte de l'objet. Il est inutile de conserver les données de départ.
- L'échantillonnage de l'objet est adapté aux caractéristiques locales de l'objet (courbures).
- Le modèle obtenu est dynamique.
- Gérer les interactions de l'objet reconstruit avec d'autres objets est simple. L'objet obtenu est modélisé par un ensemble de particules orientées, il ne diffère en rien des objets crées manuellement.

Perspectives

Afin de valider notre méthode, il nous faut l'appliquer à des données réelles (imagerie médicale 3D). Un travail reste à faire pour confronter les caractéristiques de l'objet déformable obtenu à l'objet réel de départ (élasticité). Le réglage du comportement physique de l'objet modélisé se fait principalement en jouant sur les coefficients de pondération de chaque interaction primitive (cohésion, forme et frottement) composant la loi d'interaction de chaque particule. Par exemple augmenter l'importance de l'interaction de forme (et donc l'amplitude des forces et moments produits) permet de modéliser un objet plus dur.

Chapitre 6

Implémentation

Les méthodes développées dans ce manuscrit ont donné lieu à deux implémentations. Un premier simulateur 2D a tout d'abord été écrit afin de tester la validité des idées fondatrices. Une version 3D a ensuite été intégrée à la plate-forme d'animation *Fabule* [Gas94] développée au sein du groupe Animation du laboratoire iMAGIS. Un très bref aperçu de ces réalisations est donné dans ce chapitre.

Les sections suivantes présentent tout d'abord les simplifications du modèle pour son implémentation 2D, puis la plate-forme *Fabule* et enfin l'implémentation 3D des systèmes de particules orientées.

6.1 Modélisation 2D de particules orientées

La modélisation dans un plan entraîne une simplification d'importance au modèle : il n'y a qu'un seul axe de rotation possible, l'axe perpendiculaire au plan.

Dans ce cas, la position d'une particule est définie par une translation dans le plan (deux réels) et un angle de rotation (un réel). Les caractéristiques physiques sont la masse, l'inertie, les vitesses linéaire et angulaire. Pour que cette définition ne soit plus celle d'un solide élémentaire mais celle d'une particule orientée, il faut rajouter la loi d'interaction. La loi d'interaction d'une particule étant une somme pondérée des interactions décrites dans ce manuscrit, chaque particule est caractérisée par la valeur des coefficients à appliquer à chaque fonction d'interaction. Les paramètres spécifiques des interactions (comme le rayon de l'interaction co-sphérique) sont aussi mémorisés au niveau de la particule.

La simulation des particules se fait suivant un mode pseudo-parallèle : l'état de chaque particule (position, vitesses) à l'instant $t + \Delta t$ est calculé en fonction de l'état des particules à l'instant t et des forces extérieures appliquées pendant l'intervalle de temps $[t, t + \Delta t]$. Ce nouvel état est mémorisé dans des variables temporaires. Ce n'est qu'après avoir calculé les interactions

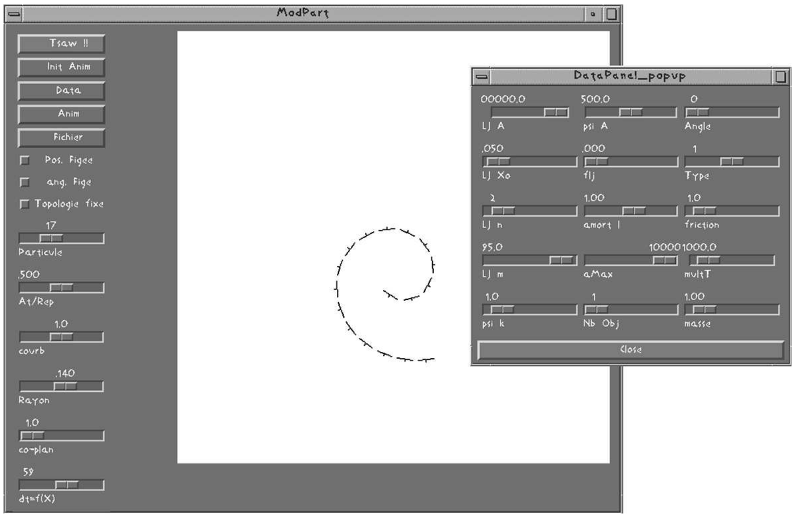


FIG. 6.1 - *Interface du simulateur 2D*

de chaque particule que le nouvel état est mis à jour. L'intégration de la dynamique se fait en utilisant un schéma d'Euler modifié (voir Annexe A).

La programmation a été faite en *C++*. L'interface graphique repose sur la librairie *OSF/Motif* et le langage de description d'interface *User Interface Langage (UIL)*. L'ensemble[1] représente environ 4500 lignes de *C++*. La figure 6.1 montre une vue d'ensemble de l'interface du logiciel. La quasi totalité des paramètres (caractéristiques des particules, des interactions, du milieu ambiant) sont accessibles par l'interface graphique. Un mécanisme de sélection permet d'affecter une ou plusieurs particules lors de la modification d'un paramètre. La création d'un objet se fait de manière interactive, par un dialogue avec l'interface. Des possibilités de sauvegarde et de restauration de la définition des objets, mais aussi du déroulement d'une simulation sont mises en place.

L'objectif de ce système 2D, à savoir une étude de faisabilité, ayant été atteint, nous avons entrepris l'implémentation 3D du système de particules orientées. Celle-ci a eu lieu dans le cadre de *Fabule* que nous présentons dans la section suivante.

6.2 *Fabule* : la plate-forme d'animation d'iMAGIS

À l'initiative de J.-D. Gascuel, le groupe Animation du laboratoire iMAGIS a fait le choix de développer une plate-forme commune, nommée *Fabule* [Gas93], regroupant les travaux des divers chercheurs et stagiaires concernés. L'objectif est de réduire au minimum le temps nécessaire pour

1. Toute les fonctions d'interaction ne sont pas implémentées, notamment, l'interaction de forme est équivalente à l'interaction co-sphérique en 2D.

passer de l'idée nouvelle à sa réalisation.
Fabule se présente comme une librairie C++ [Str92] interfacée avec le langage interprété *Tcl* [Ous94]. La visualisation repose sur la librairie graphique *OpenInventor* [Wer94] que nous utilisons sur station de travail *Silicon Graphics Inc.*.

OpenInventor propose une structure de graphe acyclique pour représenter la géométrie de la scène. La grande originalité de *Fabule* réside dans son interface avec *Tcl*. L'écriture d'une interface entre *OpenIventor* et *Tcl* permet de créer et de modifier interactivement la base de données graphique. Ces deux interfaces (*OpenInventor* $\leftrightarrow$ *Tcl* et *Fabule* $\leftrightarrow$ *Tcl*) permettent d'accéder interactivement aux objets lors du calcul d'une simulation, éventuellement de modifier leurs paramètres physiques et visuels, de détruire ou de créer de objets, et ceci sans avoir à recompiler le programme, sans même devoir nécessairement arrêter la simulation en cours.

Fabule est dédié à l'animation, essentiellement par modèles physiques, mais supporte aussi l'animation traditionnelle par interpolation de positions clefs. Une des classes principales proposée par *Fabule* est la classe des générateurs `FaGenerator`. Cette classe permet d'implémenter des lois de mouvement. Deux grand types sont dissociés, représentés par la classe `FaKeyFrame` pour l'animation par interpolation de positions clefs et les classes `FaPointDynamic` (dynamique du point) et `FaDynamic` (dynamique du solide) pour l'animation par modèles physiques. À chaque pas de simulation, les caractéristiques des objets de classe `FaGenerator` sont calculées (en mode pseudo-parallèle), puis la position des objets est mise à jour dans le graphe *OpenInventor*. La simulation est entièrement pilotée depuis l'interface graphique et la fenêtre interactive *Tcl*. Les scènes sont décrites sous forme de script *Tcl*.

Un mécanisme de *callback* permet d'appeler des fonctions définies en *Tcl* sur des événements remarquable de la simulation en C++. Par exemple, un callback permet d'exécuter une liste de fonctions *Tcl* lorsque toute les forces appliquées sont calculées, entre deux pas de simulation.

Fabule représente aujourd'hui environ 80 classes soit 40 000 lignes de *C++* et 8 000 de *Tcl*.

6.3 Implémentation 3D dans *Fabule*

L'introduction des particules orientées dans *Fabule* s'est fait par l'écriture d'une classe `FaOrientedParticle` dérivée de `FaDynamic`, et d'une classe `FaOrientedInteraction` dont dérive une classe pour chaque fonction d'interaction implémentée. La boucle d'animation a été modifiée pour prendre en compte le calcul des interactions. Une partie de ce travail a été réalisée en collaboration avec M. Desbrun, qui pour ses travaux sur les matériaux

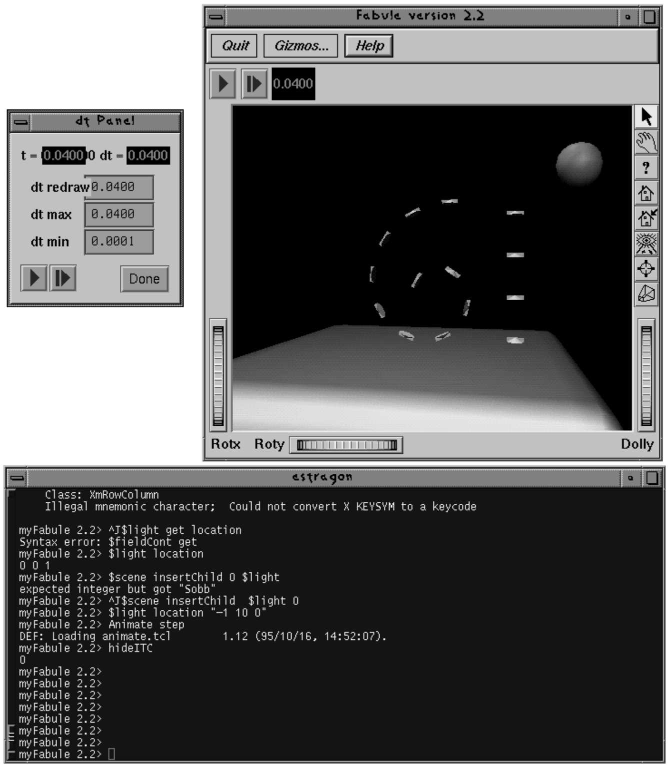


FIG. 6.2 - *Interface de* Fabule

hautement déformable (voir chapitre 4.6, page 70) implémentait à la même époque un système de particules non-orientées dans *Fabule*.
Des choix un peu différents de ceux de l'implémentation 2D ont été faits : un maximum d'informations est regroupé dans la fonction d'interaction elle-même et non dans la particule. Ceci permet une plus grande modularité. En effet, de cette façon, l'écriture d'une nouvelle fonction d'interaction ne nécessite pas de modifier la définition des particules (ce qui est le cas avec notre simulateur 2D). La loi d'interaction de chaque particule est codée sous la forme d'une liste de fonctions d'interaction. Lors de la simulation d'une particule, toutes les fonctions d'interactions sont successivement appelées, et leurs résultats sous forme de force et de moment de torsion sont combinés.

Nous avons fait une utilisation importante de l'interface *Tcl*. Par exemple, l'algorithme de reconstruction d'objet décrit au chapitre 5 est entièrement mis en œuvre en *Tcl*. Un effort important a été fourni pour offrir de bonnes possibilités de déboguage. De petites fonctions *Tcl*, utilisées comme *callback*, ont été écrites pour afficher tout au long de la simulation des caractéristiques des particules, telles que forces extérieures et d'interaction appliquées, vitesse, ou encore zone d'influence et de repos des interactions. Il est également possible d'afficher le nom *Tcl* de chaque particule afin de pouvoir accéder interactivement et sans ambiguïté aux données de la particule souhaitée.

À titre d'exemple, nous incluons ci-dessous le script *Tcl* utilisé pour la réalisation du test de robustesse de l'interaction de cohésion présenté au chapitre 3.3 (page 41) :

```
#[--------------- Initialisations diverses ------------------
set maxGen 3 #-- Nombre de generateurs
set c [SoSphere { radius 0.1}]  #-- Creation de la primitive graphique

#][---------- Creation des fonctions d'interaction ----------
set FchIt [FaCohesionIt]    #-- interaction de cohesion
$FchIt set rMax 2.5
$FchIt set amplitude 1000
$FchIt set damping 4
$FchIt set r0 1.0

set FrcIt [FaOrientedFrictionIt]  #-- interaction de frottement
$FrcIt set amplitude 1
$FrcIt  set rMax 5
$FrcIt  set weight 1

#][--------------- creation des generateurs ----------------
for {set i 0} {$i <$maxGen} {set i [expr $i+1]} {
    set p($i) [FaOrientedParticle new 1]
       #-- graphe OpenInventor
    set s($i) [SoSeparator]
    $s($i) addChild $c
    set jclPartObj($i) [FaRigidSolid]
    $jclPartObj($i) setPrimitive $s($i)
    $jclPartObj($i) setGenerator $p($i)
       #-- caracteristiques physiques
    $p($i) setMass 0.1
    $p($i) setInertia 0.1
    $p($i) addInteraction $FchIt
    $p($i) setFrictionFunction $FrcIt
}
#-- positionner les particules en fonction de la distance saisie
puts -nonewline  "Entrez la distance entre les particules (CR ^D): "
set d [expr [read stdin]]
set a [expr $d / 2.0]
set b [expr $a * sqrt(3)]
$p(0)  setPosition "-$a  0 0  0 0 0 1"
$p(1)  setPosition " $a  0 0  0 0 0 1"
$p(2)  setPosition "  0 $b 0  0 0 0 1"

#][---- Affichage des particules et des interactions -------
showPart
showITC
```

```
#][------------- sauvegarde des oscillations ---------------
traceOscil [set fich [open fch_$d.data w]] 5

#]----------------- Debut de la simulation -----------------
FaGenerator setAllTime    #-- initialisation du simulateur
Animate start             #-- la simulation commence
```

Les fonctions `showPart` et `showITC` permettent respectivement de visualiser les particules (ajout des nœuds `s($i)` à la scène) et les forces et moment d'interaction par ajout à la scène de vecteurs (pour chaque particule, un vecteur blanc montre la force et un vecteur rouge montre le moment de torsion) et d'un *callback* qui met à jour les coordonnées des vecteurs en fonction de la position des particules et des forces d'interaction à chaque pas de simulation.

```
proc showPart {} {
    global scene s maxGen areParticleShown
    if {$areParticleShown == 1} {hidePart}
    set areParticleShown 1
    for {set i 0} {$i <$maxGen} {set i [expr $i+1]} {
        $scene addChild $s($i)
    }
}

proc showITC {} {
    global forceScale forceCol forceCoord
    global torqueScale torqueCol torqueCoord
    global p forceCB forceSep scene
    global torqueSep maxGen
    global showITCenabled

    if {$showITCenabled == 1} {hideITC}
    set showITCenabled 1

    for {set i 0} {$i <$maxGen} {incr i} {
        set forceSep($i) [SoSeparator] #-- creation du vecteur force
        set forceCoord($p($i)) [SoCoordinate3 { point [0 0 0, 0 0 0]}]
        $forceSep($i) addChild $forceCol
        $forceSep($i) addChild $forceCoord($p($i))
        $forceSep($i) addChild [SoLineSet]

        set torqueSep($i) [SoSeparator] #-- creation du vecteur moment
        set torqueCoord($p($i)) [SoCoordinate3 { point [0 0 0, 0 0 0]}]
        $torqueSep($i) addChild $torqueCol
        $torqueSep($i) addChild $torqueCoord($p($i))
        $torqueSep($i) addChild [SoLineSet]
```

```
        #-- creation du callback de  mise a jour des vecteurs
        set forceCB($i) [ $p($i) addParticleCallback {
            global forceCoord torqueCoord forceScale torqueScale
            set pos [lindex [%who getPosition] 0]
            set forcePoint  [ concat "\[" $pos ","
                [vexpr $pos + $forceScale*[%who getParticleForce]]
                "\]" ]
            set torquePoint [ concat "\[" $pos ","
                [vexpr $pos + $torqueScale*[%who getParticleTorque]]
                "\]" ]
            $forceCoord(%who) point $forcePoint
            $torqueCoord(%who) point $torquePoint
        } ]

        #-- les vecteurs sont affiches
        $scene addChild $forceSep($i)
        $scene addChild $torqueSep($i)
    }
}
```

La procédure `traceOscil` ajoute un *callback* qui écrit dans le fichier passé en paramètre la mesure des oscillations à chaque pas de simulation. Ces données sont examinées et traitées ultérieurement.

Il est difficile d'évaluer exactement ma participation au projet *Fabule*. J'estime à environ 4000 lignes de *C++* l'implémentation des particules orientées et des interactions anisotrope, et à environ 2000 lignes de *Tcl* les divers outils que j'ai développés.

Chapitre 7

Vers un modèle de muscle

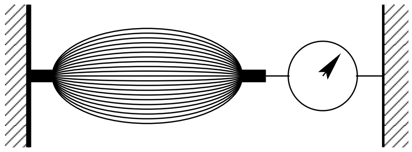
La modélisation de la totalité ou d'une partie du corps humain reste un défi pour les années à venir. L'intérêt d'une telle maquette informatique pour les gestes médicaux et chirurgicaux assistés par ordinateur (*GMCAO*) est indubitable : elle permettrait, par exemple, aux étudiants en médecine de s'initier à l'anatomie, aux chirurgiens de s'entraîner sur un patient virtuel, ou encore de simuler au préalable une intervention délicate, voire de planifier les trajectoires des robots qui commencent à pénétrer dans les salles d'opérations.

Les modèles de muscles ont été introduits en informatique graphique pour fournir un contrôle intuitif, que ce soit pour le contrôle de solides articulés ou la simulation de corps déformables. Ce type de paramétrisation permet de plus d'utiliser des connaissances acquises antérieurement dans d'autres domaines. Par exemple le système de codage FACS, qui associe à chaque expression du visage des fonctions d'activation d'ensembles de muscles (*Action Unit*), créé par des psychologues [EF78], est couramment utilisé pour l'animation faciale [PB81, Wat87, Wat92, Via92].
Si l'action musculaire est bien connue [Zaj89], il existe peu de modèles de muscles complets (c'est-à-dire alliant les déformations géométriques et les actions mécaniques) [CZ92]. À notre connaissance il n'existe aucun modèle capable de s'intégrer dans l'environnement complexe que serait celui d'un patient virtuel.

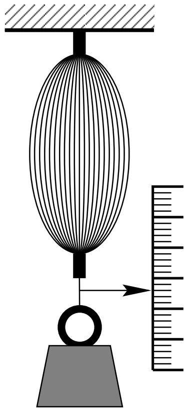
Après avoir passé en revue les différents modèles existants, nous montrons comment notre modèle d'objets déformables peut être adapté à la modélisation de muscles.

7.1 État de l'art

Selon qu'ils aient été créés pour des corps déformables ou pour des solides articulés, il existe deux types de modèles de muscles. Le premier définit les déformations d'une surface sous l'action d'une force. L'effet re-



a) Myographie isométrique



b) Myographie isotonique

FIG. 7.1 - *Mesure des grandeurs caractéristiques des muscles*

cherché est essentiellement visuel, les paramètres (amplitude de la force, zone d'influence, ...etc) sont réglés de manière plus ou moins empirique [Wat87, Via92, CHP89]. Ces modèles, surtout utilisés pour l'animation faciale, laissent une grande liberté à l'animateur, notamment celle de créer des mimiques impossibles.
Le second type de modèle caractérise la force exercée en réponse à une fonction d'activation, et déplace ainsi le problème du monde de l'animation à celui de la simulation. L'objectif recherché est la simulation d'un phénomène physique : la contraction musculaire. L'aspect visuel est secondaire.

7.1.1 Physiologie

Nous nous intéressons ici aux muscles participant à la vie de relation, c'est à dire les muscles striés autres que le cœur. Le muscle est composé de fibres musculaires, elles même composées de myofibrilles enveloppées d'une membrane (le sarcolemme). Les fibres musculaires se terminent aux deux extrémités par du tissu conjonctif, formant le tendon [Phy57].

Il existe deux méthodes classiques d'étude des propriétés mécaniques du muscle : les myographies isométriques et les myographies isotoniques. Ces techniques permettent de définir les grandeurs caractéristiques d'un muscle, nous en présentons une rapide description.

Myographie isométrique

Le muscle est fixé aux deux extrémités (Figure 7.1a), et un dynamomètre mesure la force exercée. La longueur du muscle est imposée. La force résultante est mesurée lorsque le muscle est au repos et lorsqu'il est soumis à un potentiel d'activation.

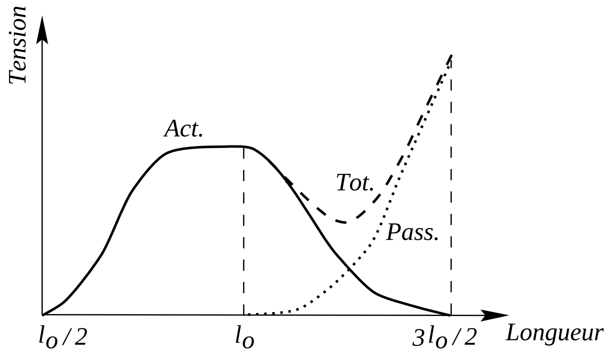


FIG. 7.2 - *Tensions active, passive et totale*

Myographie isotonique

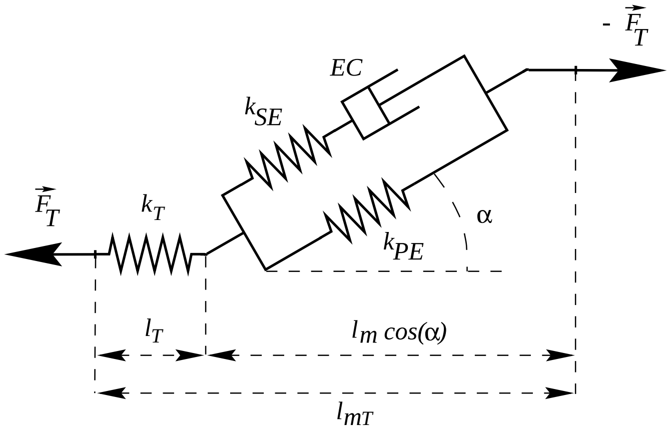
Une extrémité du muscle est fixée (Figure 7.1b), et sur l'autre est accrochée une masse. La force exercée (par la masse) est imposée. La longueur du muscle est mesurée.

Résultats et définitions

Les courbes tension/longueur obtenues permettent de définir la tension au repos ou tension passive et la tension totale (lorsque le muscle est activé). La différence entre ces deux courbes est appelée tension contractile nette ou tension active. La longueur d'équilibre (l_0) est définie comme étant la longueur d'apparition d'une force, pour un muscle au repos (Figure 7.2). Un autre phénomène important est évalué au cours de ces expériences : la dépendance entre la vitesse de contraction et la tension musculaire. Plus la contraction est rapide, moins la force développée est grande. La force maximale exercée par un muscle tétanisé de mammifère varie entre $2,5$ et $4kg/cm^2$ de section perpendiculaire à la direction des fibres.
D'autres expériences (un muscle est plongé dans un liquide et la variation du niveau du liquide est mesurée pendant la contraction) ont montré que le muscle se déforme à volume constant.

7.1.2 Modèle de l'action mécanique du muscle

Le premier équivalent mécanique de muscle, connu sous le nom de modèle de Hill [GH24], a été mis au point en 1924 par H. Gasser et A. Hill. En fait, il s'agit d'un modèle de l'ensemble muscle/tendon. Cet assemblage est composé d'un élément contractile et de deux ressorts : l'un monté en parallèle simule l'élasticité du muscle, l'autre monté en série simule l'élasticité du tendon. F. Zajac [Zaj89], E. Topp et P. Stevenson [ZTS86] ont repris et affiné ces travaux pour proposer un modèle générique de muscle. Quatre paramètres permettent de régler ce modèle pour simuler un muscle donné.



EC est l'élément contractile

FIG. 7.3 - *Équivalent mécanique du muscle*

Le dispositif obtenu, schématisé sur la figure 7.3, contient un ressort supplémentaire monté en série avec un système équivalent au modèle de Hill. Le modèle de Zajac prend en compte une donnée supplémentaire : l'angle α entre les fibres musculaires et le tendon.

Les paramètres spécifiques sont :

- la force maximale F_0 exercée par le muscle actif,
- la longueur l_0^M du muscle à laquelle F_0 est développée,
- l'angle entre le tendon et le muscle à l_0^M, et
- la longueur du tendon au repos.

Les équations différentielles (la vitesse de contraction joue un rôle important dans l'action musculaire) liant la force F_T exercée par le complexe muscle/tendon, sa longueur et la vitesse de contraction en réponse à une fonction d'activation du muscle peuvent être trouvées dans [ZTS86]. Une explication plus détaillée ainsi que la preuve de ce modèle se trouvent dans [Zaj89].

Ce modèle est beaucoup utilisé en animation, que ce soit pour le contrôle de solides articulés [PFPB92] ou pour la modélisation de muscles [CZ92, Che92].

7.1.3 Modèle par éléments finis

D. Chen propose, dans sa thèse de PhD [Che92], une implémentation du modèle de Zajac utilisant une méthode d'éléments finis. Prenant en compte la propriété de volume constant, D. Chen modélise aussi les déformations du muscle liées à la contraction ainsi que celle dues à la gravité. Toutefois, la résolution par éléments finis ne permet pas aisément de simuler conjointement le muscle et d'autre organes. En effet, la détection et le traitement des collisions entre les différentes parties constitutives sont très coûteux.

7.1.4 Autres modèles

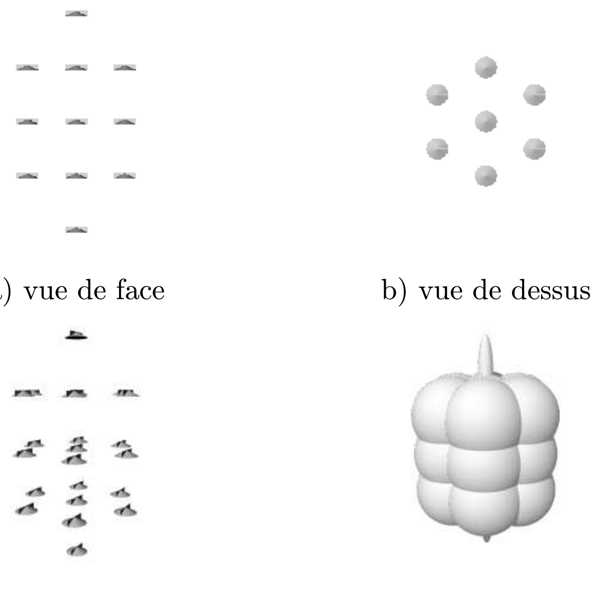
Le concept de muscle est beaucoup utilisé comme moteur pour animer des objets déformables ou articulés.
Dans le modèle proposé par K. Waters pour l'animation faciale [Wat87, Wat92, LTW95] , le tissu peaucier est modélisé par un reseau masses/ressorts à trois couches. Les masses de la couche interne sont fixes, elles représentent l'os. La couche médiane représente les muscles alors que la couche externe modélise la peau. Un muscle est un champ de forces appliqué sur la couche médiane, avec une zone d'application (l'action d'un muscle se fait sur un ensemble de masses). L'effet visuel est bien rendu, mais le muscle n'existe que par son action.

À l'opposé, J. Chadwick, D. Haumann et R. Parent ne simulent que la déformation de leur muscle [CHP89]. Ici, l'animation du personnage se fait par interpolation entre des positions clefs d'un squelette. Les positions relatives des segments du squelette sont utilisées pour créer une déformation du muscle grâce à l'utilisation de *Free Form Deformations*. Une *FFD* est un cube paramétré, et dans ce cas chaque arête comprend quatre points de contrôle d'une fonction de Bézier. Chaque point du muscle est exprimé dans un repère barycentrique lié aux points de contrôle du cube. Un déplacement de ces derniers entraîne une déformation du muscle. Les mouvements des points de contrôle sont liés aux angles entre les segments du squelette et à une fonction d'activation du muscle. La déformation du muscle est ensuite reportée sur la peau.

7.2 Les particules orientées : une solution ?

Comme nous venons de le voir, il n'existe pas à l'heure actuelle de modèle de muscle complet (comme celui de D. Chen) permettant facilement une simulation conjointe avec d'autres objets, déformables ou non.
Les systèmes de particules orientées, tels qu'ils ont été définis dans ce document, pourraient apporter une solution originale.

L'interaction co-linéaire permet de définir des alignements de particules. Ces alignements peuvent êtres utilisés pour simuler des fibres musculaires. En jouant sur les rayons de repos de l'interaction de cohésion, nous pouvons, comme on va le voir, allonger ou contracter le muscle, tout en maintenant le volume quasiment constant.



a) vue de face — b) vue de dessus

c) vue 3D — d) rayons de repos

FIG. 7.4 - *Définition du muscle*

7.2.1 Création

Le muscle est défini par un ensemble de fibres (Figure 7.4). Chaque fibre est définie par un ensemble de particules alignées selon leur axe z (Figure 7.4a). Une interaction co-linéaire est définie pour maintenir cet alignement. Les fibres sont arrangées suivant un pavage hexagonal du plan (Figure 7.4b). Les échantillons correspondants de chaque fibre sont maintenus dans le même plan par une interaction coplanaire. L'écartement entre les fibres et la distance entre chaque échantillon d'une même fibre sont gérés par une interaction de cohésion à rayon de repos ellipsoïdal (Figure 7.4d). Deux particules sont placées à chaque extrémité du muscle, elles servent de point d'ancrage. Leur connexion au reste du modèle est assurée par une force de cohésion.

7.2.2 Conservation du volume

La force de cohésion avec un rayon de repos ellipsoïdal peut être utilisée pour assurer une sorte de conservation de volume. Il faut lors de l'allongement ou de la contraction du muscle suivant un axe, compenser par une contraction ou un allongement sur les deux autres axes. Le volume d'une ellipse ayant comme rayon r_t^x, r_t^y et r_t^z suivant respectivement les axes orthonormés x, y et z est :

$$V_t = \frac{4\pi}{3} r_t^x r_t^y r_t^z \ . \tag{7.1}$$

a) Muscle contracté b) Muscle étendu

FIG. 7.5 - *Deux états du muscle*

Dans nos exemples, les fibres sont alignées suivant l'axe z des particules. Une action sur la longueur du muscle est compensée par une action sur sa largeur, de façon isotrope ($r_t^x = r_t^y$). D'après l'équation 7.1, la conservation du volume de l'ellipse au cours du temps impose la relation suivante entre les rayons :

$$V_t = V_0 \quad \Rightarrow \quad r_t^x = r_t^y = \sqrt{\frac{r_0^x r_0^y r_0^z}{r_t^z}} \ .$$

Ceci ne garantit pas une conservation du volume du muscle, mais assure cependant que des forces seront appliquées pour que les zones échantillonnées par chaque particule restent de volume constant. Le recouvrement entre ces zones n'est pas pris en compte. De plus, même au niveau de chaque particule, il peut y avoir des variations de volume. Si des forces extérieures compriment violemment les particules, elles n'auront pas une position relative de repos par rapport à l'interaction de cohésion.

La figure 7.5 montre le muscle dans deux états extrêmes : en a) il est comprimé de 50%, et en b) il est étiré de 150%. Les ellipsoïdes montrés sont les distances de repos de chaque particule.

7.2.3 Action mécanique

Afin de modéliser l'action musculaire, il faut définir une fonction d'activation du muscle qui a pour effet de produire des forces semblables à celles produites par le muscle. Ce travail reste à faire.

7.2.4 Problèmes

Afin d'obtenir un modèle utilisable, une fois les réponses aux questions posées dans la section précédente trouvées, il reste encore à modifier l'algorithme de simulation des particules. En effet, dans un pur système de particules comme le notre, une particule a une seule loi d'interaction[1] qui

1. Même si cette loi est la somme de plusieurs autres, il n'y a au total qu'une seule loi d'interaction par particule.

a) interaction co-linéaire b) interaction de cohésion

FIG. 7.6 - *Zones d'influence des interactions utilisées*

est appliquée pour calculer les interactions avec toutes les autres particules. Or, dans le cas du modèle de muscle, nous souhaitons diversifier les voisinages des particules pour chaque interaction. En effet, il est souhaitable que l'interaction co-linéaire ne s'applique qu'entre particules de la même fibre alors que les interactions de cohésion et de frottement doivent s'appliquer entre toutes les particules. Ceci n'est pas réalisable avec le mécanisme de typage des particules présenté à la section 4.2.2 (page 59). Ce dernier fonctionne en *tout ou rien* : il y a interaction ou il n'y a que répulsion. Le comportement nécessaire à la modélisation du muscle est plus complexe.

En définissant avec attention les zones d'influence des diverses interactions entrant en jeu, nous pouvons obtenir la différenciation souhaitée. La figure 7.6 montre les zones d'influence des interactions. Seules les particules placées les unes au dessus des autres sont soumises à une interaction colinéaire, ce qui n'es pas le cas de l'interaction de cohésion. Le système obtenu est cependant très fragile : dès que le mouvement des particules est suffisant pour changer les voisinages, le modèle n'est plus valable. Afin de résoudre ce problème, il faudrait explicitement créer des listes de voisins pour chaque interaction. Cette modification rapprocherait le système de particules d'un réseau masses/ressorts utilisant des masses orientées et des ressorts généralisés.

Chapitre 8

Conclusion

Les travaux présentés dans ce document ont donné lieu à deux présentations lors de conférences nationale [LP94] et internationale [LP95b], à la publication d'un article dans une revue internationale [LP95a] et à la présentation d'un poster [Lom95]. D'autres articles sont en préparation. La conclusion de ce document est l'occasion de souligner les principales innovations apportées par mon travail, mais aussi de critiquer le modèle proposé et enfin de proposer des ouvertures vers de nouveaux travaux de recherche.

Nous avons tout d'abord présenté une solution originale au problème des oscillations des systèmes de particules. Après avoir étudié les différentes techniques d'intégration numérique des équations de la dynamique, nous avons apporté la démonstration de la validité de notre proposition. Cette solution dépasse le cadre de la modélisation des objets à mémoire de forme. En effet, elle peut être utilisée directement dans tous les systèmes de particules utilisant une force d'attraction/répulsion et une intégration discrète des équations de la dynamique. Même si la solution apportée par l'interaction de cohésion n'est pas parfaite (les paramètres sont liés ce qui rend leur utilisation malaisée; un changement d'échelle de l'objet nécessite de modifier la valeur des paramètres utilisés), l'approche innovante qui consiste à définir la fonction d'interaction en fonction de ses propriétés mathématiques, et non plus en fonction de son lien avec la physique, ouvre la porte à un nouveau champ d'investigation. Plus précisément, modéliser l'interaction d'attraction/répulsion par une fonction polynomiale par morceaux (type *spline*), respectant les propriétés intuitives d'une force d'attraction/répulsion et la nouvelle contrainte de dérivée nulle à la position d'équilibre, devrait permettre d'obtenir à la fois un jeu de paramètres intuitifs et une indépendance par rapport à l'échelle tout en limitant l'amplitude des oscillations.

Nous avons ensuite présenté les composants d'un système complet, permettant de créer un modèle d'objet déformable à partir de données tridimensionnelles, puis de le simuler.
Le modèle, basé sur l'utilisation de particules orientées, permet de simuler

une grande variété de comportements, de la déformation élastique ou visco-élastique à la fracture. Ce modèle est aussi bien adapté à la modélisation volumique qu'à la modélisation surfacique. Dans ce dernier cas, l'écriture de l'interaction de forme a permis de modéliser tous types de surface d'après leurs propriétés mathématiques (courbures locales directionnelles).
La mise en place de zones d'influences (et de zones de repos pour l'interaction de cohésion) ellipsoïdales permet de répartir l'échantillonnage suivant les directions, et donc de diminuer le nombre de particules nécessaires pour la modélisation des objets. La répartition directionelle des échantillons est un idée qui a été suggérée par G. Turk [Tur92] en 1992 dans le cadre de la simplification de maillage déjà existants, mais qui n'a, à notre connaissance, jamais été mise en œuvre.
Nous proposons deux modes de construction des objets. Dans le premier, un ensemble de particules est initialisé dans une position arbitraire (sur un plan par exemple), et les interactions sont spécifiées. Le système évolue naturellement vers la configuration qui vérifie les propriétées géométriques spécifiées par les lois d'interaction. Le second mode de création des objets utilise notre algorithme de reconstruction. Dans ce cas la géométrie des objets est spécifiée de façon plus intuitive, par exemple en utilisant des outils de modelage [TG94]. Les particules sont automatiquement générées et les interactions spécifiées pour obtenir un modèle déformable de l'objet.

Finalement, l'utilisation conjointe des systèmes de particules orientées tels qu'ils ont été définis dans ce document et des surfaces implicites, a permis de créer un nouveau type d'objets déformables. Ce modèle allie la puissance des particules et des surfaces implicites, et permet de simuler des comportements variés : les déformations ont lieu à deux niveaux (au niveau de la surface et au niveau de la structure interne de l'objet), les objets sont capables de fractures, des surfaces de contact exactes sont calculés; enfin, la surface des objets est continue et la visualisation peut être de très grande qualité grâce à l'emploi d'algorithmes de rendu tels que par exemple le lancer de rayons.

Les différentes expérimentations menées avec ce modèle d'objet déformable nous ont aussi permis d'en apercevoir les limitations.
Dans un système de particules, la topologie de l'objet modélisé n'est pas fixée. Ceci permet de briser les objets puis de recoller les morceaux. Le recollement est très intéressant pour des pâtes plus ou moins fluides. Par contre, pour des objets plus complexes, ayant une forme propre, le comportemet intuitif attendu est plutôt que les parties d'un objet brisé se comportent comme autant d'objets et entrent en collision les unes avec les autres, plutôt que de se recoller. Le recollement incontrôlé des parties d'objets n'est donc pas un comportement très intéressant.
Dans un système de particules, le voisinage de chaque particule n'est pas plus fixé que la topologie de l'objet. Ceci signifie qu'une particule, suite à

une action extérieure telle qu'une collision par exemple, peut se déplacer à la surface de l'objet (ou à l'intérieur s'il s'agit d'une modélisation volumique). Ceci n'est pas gênant quand toutes les particules sont identiques. Par contre, quand chaque particule à des propriétés bien spécifiques, comme c'est le cas notamment avec notre interaction de forme, elle modélise une partie bien précise de l'objet. Voir cette portion de l'objet se déplacer à la surface de l'objet est très gênant. Ceci est de plus difficilement contrôlable.
Les deux remarques précédentes nous conduisent à imaginer une solution. Pour la modélisation d'objets complexes, où une particule véhicule beaucoup d'information, il serait plus judicieux de ré-écrire le système de particule comme un réseau masses/ressorts généralisé. Doublement généralisé en fait, puisque les masses de ce réseau doivent être orientées, et que les ressorts doivent simuler les interactions décrites dans ce document. Comme nous l'avons mentionné dans le premier chapitre, il est facile de modéliser une fracture avec un réseau masse/ressort. Il suffit au moment du calcul de chaque ressort, de décider si oui ou non il y a fracture. En cas de fracture, le ressort est détruit. Il est certainement intéressant d'approfondir ce nouveau concept de réseau de masses orientées/ressorts généralisés. Il faut, entre autres, régler le problème des interactions entre masses non connectées par des ressorts afin d'éviter les interpénétrations.

La première de nos perspectives est donc d'adapter le travail fait dans le cadre d'un système de particules à un réseau de masses orientées et de ressorts généralisés. Ceci permettra aussi de terminer notre modèle de muscle, à peine ébauché au chapitre 7, modèle qui contrairement à ceux existant pourra sans peine cohabiter avec d'autre objets (modélisés de la même façon ou par une fonction dedans/dehors). Peu de travail est nécessaire pour le transfert du système de particules vers un réseau masses/ressorts. Il suffit de modifier légèrement la définition des interactions et de ré-écrire l'algorithme de simulation. Transformer les interactions en ressort demande uniquement d'inclure dans chaque interaction les références des deux particules connectées par le ressort et de maintenir une liste de tous les ressorts actifs dans l'objet. L'algorithme de simulation doit être modifié de façon à traiter successivement chaque ressort au lieu de traiter successivement la loi d'interaction de chaque particule.
L'autre point laissé en suspens dans ce document est le test de notre algorithme de reconstruction d'objets sur des données réelles. Nous pensons que nous pourrions apporter une solution originale au problème de la segmentation de données. Ce domaine est encore très actif aujourd'hui. Actuellement, l'image 3D brute est d'abord traitée avec une succession de filtres qui permettent, par exemple de calculer le gradient de l'image. Puis, une détection de contours est effectuée. Selon les techniques employées, une image de distance à ces contours (en chaque point de l'image, la distance au point de contour le plus proche est mémorisée) est ensuite calculée. À partir de cette

carte de distance, un modèle statique de l'objet est créé. Le modèle d'objet déformable est calculé à partir du modèle statique. Nous postulons qu'avec notre modèle et notre algorithme de reconstruction, nous sommes capables de générer directement un modèle d'objet déformable à partir de la carte de distance. En effet, structurellement, une carte de distance n'est pas différente des fonctions implicites utilisées lors de nos tests.
Le second modèle d'objets déformables proposé (alliant particules orientées et surfaces implicites), permet d'utiliser des fonctions implicites complexes. L'emploi de fonction implicites anisotropes est une voie qui reste à explorer, il permettrait de diminuer encore le nombre de particules nécessaires à la modélisation d'un objet.

En résumé, l'utilisation de systèmes de particules orientés pour modéliser des objets déformables permet de simuler des comportements complexes et variés, tout en présentant des coùts raisonnables : chaque particule comporte beaucoup d'informations, seul un petit nombre de particule est donc nécessaire à la modélisation de l'objet. L'étude présentée dans ce document à permis de préciser un certain nombre de choix importants et de dégager des directions de recherche qui permettrons d'obtenir un modèle d'objet déformable aux propriétés très intéressantes.

Annexe A

Intégration de la dynamique

A.1 Évaluation numérique d'intégrales

L'application des lois de la dynamique pour calculer l'état courant d'un objet implique la résolution d'équations différentielles. Dans les paragraphes suivants, nous présentons les méthodes les plus classiques pour la résolution d'équations différentielles du type :

$$\frac{dy}{dt} = f(t) \ .$$

Résoudre pas à pas ce type d'équation revient à évaluer l'intégrale :

$$y(t + \Delta t) - y(t) = \int_t^{t+\Delta t} f(u)du$$

où $f(t)$ n'est pas connue mais peut être évaluée (avec un coût en temps de calcul élevé dès que la scène n'est pas triviale) en différents points d'échantillonnage.
Nous exposons finalement notre choix dans un contexte de résolution des équations de la dynamique.

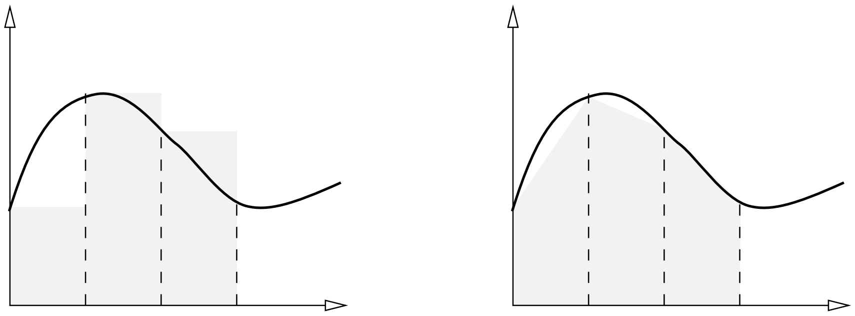
A.1.1 Méthode d'Euler

C'est la méthode de calcul numérique d'intégrale la plus simple. La méthode d'Euler est exacte lorsque $\vec{f}(t)$ est constante sur l'intervalle $[t, t + \Delta t]$ (Figure A.1a).

$$\int_t^{t+\Delta t} \vec{f}(u)du = \vec{f}(t)\Delta t + O(f'\Delta t^2) \ .$$

En intégrant les deux équations du système 1.13 (page 18) avec cette méthode (en supposant $\vec{f}(t)$ constante sur $[t, t + \Delta t]$), on obtient :

$$\begin{cases} \vec{v}(t + \Delta t) = \dfrac{\vec{f}_{ext}(t)}{m}\Delta t + \vec{v}(t) \\ \\ \vec{x}(t + \Delta t) = \vec{v}(t)\Delta t + \vec{x}(t) \ . \end{cases} \tag{A.1}$$



a) Intégration d'Euler *b) Intégration de Newton-Cotes*

L'intégrale de la fonction est approximée par l'aire grisée

FIG. A.1 - *Méthodes classiques d'intégration numérique*

On peut remarquer que même si la condition $\vec{f}_{ext}(t)$ = constante est vérifiée (par exemple pour un objet soumis aux seules actions de la gravité), une erreur est introduite. Dans le cas général ($\vec{f}(t) \neq \vec{0}$), $\vec{v}(t)$ n'est pas constante sur $[t, t+\Delta t]$ et l'erreur introduite pendant le calcul de $\vec{x}(t+\Delta t)$ est de l'ordre de $O(v'\Delta t^2)$.
Ce schéma d'intégration est cependant couramment employé.

A.1.2 Méthode de Newton-Cotes

Une méthode d'intégration un peu plus évoluée est celle de *Newton-Cotes* [Pa92, Act90].

$$\int_t^{t+\Delta t} f(u)du = (f(t) + f(t+\Delta t))\,\frac{\Delta t}{2} + O(f''\Delta t^3)\ . \tag{A.2}$$

Cette formulation est exacte quand $f(t)$ varie linéairement (Figure A.1b).
Dans ce cas :

$$\begin{aligned}
f(t) &= at + b\ , \\
\int_t^{t+\Delta t} f(u)du &= \left[\frac{au^2}{2} - bu\right]_t^{t+\Delta t} \\
&= at\Delta t + \frac{a\Delta t^2}{2} + b\Delta t \\
&= (2at + a\Delta t + 2b)\,\frac{\Delta t}{2} \\
&= (f(t) + f(t+\Delta t))\,\frac{\Delta t}{2}\ .
\end{aligned}$$

Cette technique d'intégration n'est cependant pas applicable pour calculer les vitesses, puisqu'elle requiert une évaluation des forces à l'instant $t+\Delta t$. Or $f(t+\Delta t)$ dépend de l'état de l'objet à l'instant $t+\Delta t$, ce qui est justement ce que l'on cherche à déterminer.

A.1.3 Schéma d'Euler modifié

Le schéma que nous utilisons est obtenu en couplant les deux méthodes d'intégration présentées ci-dessus.
La vitesse est calculée en appliquant le méthode d'Euler et la position en appliquant la méthode de Newton-Cotes. Le principe fondamental de la dynamique du point (éq.1.13) s'applique en calculant :

$$\left\{ \begin{array}{rcl} \vec{v}(t+\Delta t) & = & \vec{v}(t) + \frac{\Delta t}{m} \sum \vec{f}_{ext} \\ \vec{x}(t+\Delta t) & = & \vec{x}(t) \frac{\Delta t}{2} + (\vec{v}(t) + \vec{v}(t+\Delta t)) \ . \end{array} \right. \tag{A.3}$$

Ce schéma d'intégration présente l'avantage d'être exact lorsque les objets sont soumis à des forces constantes. Dans le cas contraire, l'ordre de grandeur de l'erreur commise (notée $\vec{\epsilon}_v$ pour la vitesse et $\vec{\epsilon}_x$ pour la position) est donné par :

$$\begin{array}{rcl} \vec{\epsilon}_v & = & O(\frac{d\vec{f}_{ext}}{dt} \Delta t^2) \ , \\ \vec{\epsilon}_x & = & O(\frac{d^2\vec{v}}{dt^2} \Delta t^3) + \int \vec{\epsilon}_v \\ & = & O(\frac{d\vec{f}_{ext}}{dt} \Delta t^3 + \vec{f}_{ext} \Delta t^2) \ . \end{array} \tag{A.4}$$

A.1.4 Méthodes de rang plus élevé

Il y a plusieurs méthodes pour évaluer plus précisement l'intégrale d'une fonction f sur l'intervalle $[a, b]$. Chacune de ces méthodes fait appel à l'évaluation de f en des points intermédiaires.

Pour mémoire, nous citons une famille de méthodes qui consistent à calculer f en plusieurs points d'échantillonnage régulièrement espacés sur $[a, b]$ et à approximer f par un polynôme. En posant $f_i = f(a+ih), i = 0..n$ où n et h sont tels que $b = x_n$, et en notant $I = \int_a^b f(x)dx$, on obtient [Act90] :

$$\begin{array}{llll} n=1 & I = \frac{h}{2}(f_0 + f_1) & +O(h^3 f'') & \text{(Newton-Cotes)} \\ n=2 & I = \frac{h}{3}(f_0 + 4f_1 + f_2) & +O(h^5 f^4) & \text{(Simpson)} \\ n=3 & I = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) & +O(h^5 f^4) & \text{(Simpson } \frac{3}{8}\text{)} \\ & & \cdots & \end{array}$$

Cette technique est applicable pour le calcul des équations de la dynamique en évaluant $\int_{t-(n-1)\Delta T}^{t+\Delta T} f(u)du$ et non $\int_t^{t+\Delta T} f(u)du$. Il nous faut nous ramener à notre contexte pour juger de la pertinence de telles méthodes.
Une scène classique d'animation par ordinateur peut être décomposée en instants où les objets sont soit soumis à des forces constantes (vol balistique

ou contact prolongé) soit en collision. Dans ce dernier cas, la fonction modélisant les forces appliquées f présente des changements importants (sauf dans le cas de chocs mous intégrés avec un pas de temps suffisement petit), voire des discontinuités. Approximer f par un polynôme suppose une certaine régularité de f, ce qui est faux dans le cas où des collisions se produisent.

Dans le cas d'objets soumis à des forces constantes, il est inutile de compliquer les calculs puisque le schéma d'Euler modifié donne une intégration exacte de la position.

L'utilisation de ces méthodes plus coûteuses en temps de calcul et en place mémoire ne se justifie donc pas pour l'animation d'objets par modèles physiques, en tout cas dans le contexte qui est le nôtre.

Annexe B

Quaternions

Les quaternions sont des outils puissants pour gérer les rotations dans un espace 3D [Sho85, Sho91]. Leur usage se développe en informatique graphique, où ils remplacent les classiques angles d'Euler et les matrices de rotation.

B.1 Définitions

L'ensemble des quaternions est noté $\mathcal{Q}$. Un quaternion peut être vu comme une généralisation des nombres complexes. C'est une combinaison linéaire de 1, $\vec{\imath}$, $\vec{\jmath}$ et $\vec{k}$ (où $\vec{\imath}$, $\vec{\jmath}$ et $\vec{k}$ forment un trièdre orthonormé direct) :

$$\mathsf{q} = \alpha + \beta\vec{\imath} + \gamma\vec{\jmath} + \delta\vec{k}\ ,$$

où α, β, γ et δ sont des réels[1]. Les opérations suivantes sont définies sur $\mathcal{Q}$:

+ Addition

$$\begin{aligned}\forall \mathsf{q}_1, \mathsf{q}_2 \in\ & \mathcal{Q}\\ \mathsf{q}_1 + \mathsf{q}_2 \ =\ & (\alpha_1 + \alpha_2) + (\beta_1 + \beta_2)\vec{\imath} + (\gamma_1 + \gamma_2)\vec{\jmath} + (\delta_1 + \delta_2)\vec{k}\ .\end{aligned}$$

· Produit externe

$$\begin{aligned}\forall \mathsf{q}_1 \in\ & \mathcal{Q} \quad ,\\ \forall a \in\ & \mathbb{R} \quad ,\\ a\mathsf{q} \ =\ & a\alpha + a\beta\vec{\imath} + a\gamma\vec{\jmath} + a\delta\vec{k}\ .\end{aligned}$$

$(\mathcal{Q}, +, \cdot)$ forme un espace vectoriel de dimension 4.

1. α est pappelé partie réelle du quaternion, tandis que le vecteur $\beta\vec{\imath}+\gamma\vec{\jmath}+\delta\vec{k}$ est appelé partie imaginaire.

$\star$ Produit interne[2]

$$\begin{aligned}
\vec{\imath}\star\vec{\jmath} &= -\vec{\jmath}\star\vec{\imath}=\vec{k}\\
\vec{\jmath}\star\vec{k} &= -\vec{k}\star\vec{\jmath}=\vec{\imath}\\
\vec{k}\star\vec{\imath} &= -\vec{\imath}\star\vec{k}=\vec{\jmath}\\
\vec{\imath}\star\vec{\imath} &= \vec{\jmath}\star\vec{\jmath}=\vec{k}\star\vec{k}=-1\ ,
\end{aligned}$$

ce qui donne pour deux quaternions quelconques :

$$\begin{aligned}
\mathsf{q}_1\star\mathsf{q}_2 &= (\alpha_1\alpha_2-\beta_1\beta_2-\gamma_1\gamma_2-\delta_1\delta_2)\\
&+ (\alpha_1\beta_2+\alpha_2\beta_1+\gamma_1\delta_2-\gamma_2\delta_1)\vec{\imath}\\
&+ (\alpha_1\gamma_2+\alpha_2\gamma_1-\beta_1\delta_2+\beta_2\delta_1)\vec{\jmath}\\
&+ (\alpha_1\delta_2+\alpha_2\delta_1+\beta_1\gamma_2-\beta_2\gamma_1)\vec{k}\ .
\end{aligned}$$

$(\mathcal{Q},+,\star)$ est un corps non commutatif.

$\overline{\mathsf{q}}$ Conjugaison

$$\begin{aligned}
&\forall \mathsf{q}\in\mathcal{Q}\ ,\\
&\overline{\mathsf{q}} = \alpha-\beta\vec{\imath}-\gamma\vec{\jmath}-\delta\vec{k}\ .
\end{aligned}$$

$|\mathsf{q}|$ Norme

$$\begin{aligned}
&\forall \mathsf{q}\in\mathcal{Q}\ ,\\
&|\mathsf{q}| = (\alpha^2+\beta^2+\gamma^2+\delta^2)^{\frac{1}{2}}=\sqrt{\mathsf{q}\star\overline{\mathsf{q}}}\ .
\end{aligned}$$

q^{-1} Inverse

$$\begin{aligned}
&\forall \mathsf{q}\in\mathcal{Q}^*\ ,\\
&\mathsf{q}^{-1} = \frac{\overline{\mathsf{q}}}{\mathsf{q}\star\overline{\mathsf{q}}}=\frac{\overline{\mathsf{q}}}{|\mathsf{q}\star\overline{\mathsf{q}}|^2}\ .
\end{aligned}$$

B.2 Représentation de vecteurs

$\mathbb{R}^3$ est plongé dans $\mathcal{Q}$: à tout vecteur $\vec{v}$ de $\mathbb{R}^3$ est associé le quaternion $0+\vec{v}$.
L'addition et le produit externe sont les mêmes que ceux habituellement définis sur $\mathbb{R}^3$. Pour le produit interne :

$$\begin{aligned}
&\forall \vec{u},\vec{v}\in\mathbb{R}^3\ ,\\
&\vec{u}\star\vec{v} = -(\vec{u}\cdot\vec{v})+(\vec{u}\wedge\vec{v})
\end{aligned}$$

où $\cdot$ et $\wedge$ sont respectivement les produits scalaire et vectoriel de $\mathbb{R}^3$.

2. Par abus de notation, $\vec{v}$ parfois identifié au quaternion de partie réelle nulle : $0+\vec{v}$.

B.3 Représentation des rotations

La rotation d'axe unitaire $\vec{n}$ et d'angle φ est notée $\mathcal{R}_{\vec{n},\varphi}$. Elle est représentée par le quaternion unitaire $\mathsf{q}_{\vec{n},\varphi}$:

$$\mathsf{q}_{\vec{n},\varphi} = \cos\frac{\varphi}{2} + \vec{n}\sin\frac{\varphi}{2} \ .$$

Il y a bijection entre l'ensemble des rotations et l'ensemble des quaternions unitaires. La composition de rotations s'obtient par un simple produit de quaternions :

$$\mathcal{R}_{\vec{n}_1,\varphi_1} \circ \mathcal{R}_{\vec{n}_2,\varphi_2} = \mathsf{q}_{\vec{n}_1,\varphi_1} \star \mathsf{q}_{\vec{n}_2,\varphi_2} \ .$$

L'intérêt de cette représentation par rapport à la notation matricielle est principalement un gain en stabilité numérique, mais aussi une économie en nombre d'opérations.
L'image du vecteur $\vec{v}$ par la rotation $\mathcal{R}_{\vec{n},\varphi}$ est donnée par la formule de Rodriguez (expression vectorielle) :

$$\mathcal{R}_{\vec{n},\varphi}(\vec{v}) = \vec{v}\cos\varphi + \vec{n}\wedge\vec{v}\sin\varphi + (1-\cos\varphi)(\vec{n}\cdot\vec{v})\vec{n} \ . \tag{B.1}$$

L'expression de cette rotation avec des quaternions est :

$$\mathcal{R}_{\vec{n},\varphi}(\vec{v}) = \mathsf{q}_{\vec{n},\varphi} \star \vec{v} \star \mathsf{q}_{\vec{n},\varphi}^{-1} \ .$$

B.4 Exponentielle et logarithme de quaternions

Exponentielle et logarithme de quaternions sont utiles pour l'intégration des rotations d'un solide. Avant de pouvoir les définir, il nous faut donner la définition de l'élévation à la puissance m d'un quaternion de rotation $\mathsf{q}_{\vec{n},\varphi}$ [3] :

$$\mathsf{q}_{\vec{n},\varphi}^{m} = \cos(\frac{m\varphi}{2}) + \sin(\frac{m\varphi}{2})\vec{n} \ .$$

L'exponentielle d'un quaternion est définie à partir de son développement en serie :

$$\exp(\mathsf{q}) = 1 + \frac{\mathsf{q}}{1!} + \frac{\mathsf{q}^2}{2!} + \frac{\mathsf{q}^3}{3!} + \ldots + \frac{\mathsf{q}^n}{n!} + \ldots \ .$$

Dans le cas particulier où q est un quaternion de partie réelle nulle ($\mathsf{q} = \vec{q}$), l'expression précédente devient [Han93] :

$$\exp(\vec{q}) = \cos\|\vec{q}\| + \frac{\vec{q}}{\|\vec{q}\|}\sin\|\vec{q}\| \ .$$

Le quaternion obtenu $\exp(\vec{q})$ est donc unitaire.

3. La démonstration de cette égalité peut être trouvée dans [Han93].

Dans le cas où $\mathtt{q} = \alpha + \vec{v}$ est un quaternion unitaire, le logarithme est défini par :

$$\log \mathtt{q} = \begin{cases} \dfrac{\arccos \alpha}{\sqrt{1-\alpha^2}} \vec{v} & \text{si } \alpha \neq 1 \\ 0 & \text{sinon} \end{cases} .$$

Dans le cas d'un quaternion de rotation, le logarithme s'écrit :

$$\log \mathtt{q}_{\vec{n},\varphi} = \frac{\varphi}{2} \vec{n} ,$$

d'où la notation condensée :

$$\mathtt{q}_{\vec{n},\varphi} = e^{\frac{\varphi}{2}\vec{n}} .$$

> **Remarque :** Du fait de la non commutativité de la multiplication des quaternions, il faut manipuler les logarithmes avec précaution. En effet, dans le cas général, $e^{(\log \mathtt{q}_1 + \log \mathtt{q}_2)}$ est différent de $e^{\log \mathtt{q}_1} \star e^{\log \mathtt{q}_1} = \mathtt{q}_1 \star \mathtt{q}_2$.

Bibliographie

[Act90] Forman S. ACTON. *Numerical Methods that Work.* The Mathematical Association of America, 1990.

[AF77] Marcello ALONSO et Edward J. FINN. *Physique Générale - Tome 1 : Mécanique.* InterÉditions, Paris, 1977.

[Che92] Davis Tzu-Wei CHEN. « *Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method* ». PhD thesis, Massachusetts Institute of Technology, February 1992.

[CHP89] John E. CHADWICK, David R. HAUMANN, et Richard R. PARENT. « Layered construction for deformable animated characters ». *Computer Graphics*, 23(3):243–252, July 1989.

[CZ92] David Tzu-Wei CHEN et David ZELTZER. « Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method ». *Computer Graphics*, 2(26), July 1992. SIGGRAPH '92 Proceedings.

[Des94] M. DESBRUN. « *Modélisation de matériaux hautement déformables en synthèse d'images* ». Rapport de DEA Informatique, Institut d'Informatique et de Mathématiques Appliquées de Grenoble, Juin 1994.

[DG94] Mathieu DESBRUN et Marie-Paule GASCUEL. « Highly Deformable Material for Animation and Collision Processing ». Dans *5th Eurographics Workshop on Animation and Simulation*, Oslo, Norway, septembre 1994.

[DG95] Mathieu DESBRUN et Marie-Paule GASCUEL. « Animating Soft Substances with Implicit Surfaces ». *Computer Graphics*, août 1995. SIGGRAPH '95 Proceedings.

[DSG91] Julie O'B. DORSEY, François X. SILLION, et Donald P. GREENBERG. « Design and simulation of opera lighting and projection effets ». *Computer Graphics*, 25(4):41–50, July 1991.

[EF78] Paul Ekman et Wallice Friesen. *Facial Action Coding System.* Consulting Psychologists Press, Inc., 1978.

[Gas93] Marie-Paule Gascuel. « An implicit formulation for precise contact modeling between flexible solids ». *Computer Graphics*, p. 313–320, August 1993. SIGGRAPH '93 Proceedings.

[Gas94] Jean-Dominique Gascuel. « Fabule : un environnement de recherche pour l'animation et la simulation ». Dans *Les Simulateurs, Troisième Séminaire du groupe de travail français Animation et Simulation*, Lille, octobre 1994.

[GH24] H. S. Gasser et A. V. Hill. « The dynamics of muscular contraction ». *Royal Society of London Proceedings*, 96:398–437, 1924.

[GTT89] Jean-Paul Gourret, Nadia Magnena Thalmann, et Daniel Thalmann. « Simulation of object and human skin deformations in a grasping task ». *Computer Graphics*, 23(3):21–30, July 1989. SIGGRAPH'89 Proceedings.

[Han93] Gabriel Hanotaux. « *Techniques de contrôle du mouvement pour l'animation* ». Thèse de doctorat, École Nationale Supérieure des Mines de Saint-Etienne, Université Jean Monnet de Saint-Etienne, Avril 1993.

[HWA+92] David Haumann, Jakub Wejchert, Kavi Arya, Bob Bacon, Al Khorasani, Alan Norton, et Paula Sweeney. « An application of motion design and control for physically-based animation ». *ACM SIGGRAPH '92 Course Notes*, 16:22–29, July 1992.

[Jim93] Stéphane Jimenez. « *Modélisation et simulation physique d'objets volumiques déformables complexes* ». Thèse de doctorat, Institut National Polytechnique de Grenoble, 1993.

[LC87] William E. Lorensen et Harvey E. Cline. « Marching cubes: a high resolution 3D surface construction algorithm ». *Computer Graphics*, 21(4):163–169, juillet 1987. SIGGRAPH'87 Proceedings.

[LFA75] Jacqueline Lelong-Ferrand et Jean-Marie Arnaudiès. *Cours de Mathématiques (tome 3) : Géométrie et Cinématique.* Dunod Université, Paris, 1975.

[LJF+91] A. Luciani, S. Jimenez, J-L. Florens, C. Cadoz, et O. Raoult. « Computational physics: a modeler simulator for animated physical objects ». Dans *Eurographics '91*, Vienna, Austria, septembre 1991.

[Lom92] Jean-Christophe LOMBARDO. « *Étude des solutions au problème de l'illumination globale faisant appel à la méthode d'intégration de Monte Carlo* ». Rapport de DEA Informatique, Institut d'Informatique et de Mathématiques Appliquées de Grenoble, Juin 1992.

[Lom95] Jean-Christophe LOMBARDO. « Construction d'objets déformables à partir de données 3D en utilisant un système de particules orientées ». colloque «Images de synthèses et application», *Entretiens Jacques Cartier*, Saint-Etienne, les 6, 7 et 8 Décembre 1995.

[LP94] Jean-Christophe LOMBARDO et Claude PUECH. « Modélisation d'objets déformables avec un système de particules orientées ». Dans *Deuxièmes journées de l'Association Française d'Informatique Graphique*, p. 161–175, Toulouse, 30 Novembre - 2 Décembre 1994.

[LP95a] Jean-Christophe LOMBARDO et Claude PUECH. « Modélisation d'objets déformables avec un système de particules orientées ». Dans *Revue internationale de CFAO et d'informatique graphique*, volume 10(4), p. 337–354. Hermès, Paris, 1995.

[LP95b] Jean-Christophe LOMBARDO et Claude PUECH. « Oriented Particles: A Tool for Shape Memory Objects Modelling ». Dans *Graphics Interface '95*, p. 255–262, Quebec city, Canada, 1995.

[LTW95] Yuencheng LEE, Demetri TERZOPOULOS, et Keith WATERS. « realistic modeling for facial animation ». *Computer Graphics*, p. 55–62, August 1995. SIGGRAPH '95 Proceedings.

[Mil88] Gavin S. P. MILLER. « The motion dynamics of snakes and worms ». *Computer Graphics*, 22(4):169–177, August 1988. SIGGRAPH'88 Proceedings.

[MP89] Gavin MILLER et Andrew PEARCE. « Globular Dynamics: A connected particles system for animating viscous fluids ». *Computer and Graphics*, p. 305–309, 89.

[Ous94] John K. OUSTERHOUT. *An introduction to Tcl and Tk*. Addison-Wesley, 1994.

[Pa92] W. H. PRESS et AL.. *Numerical Recipes in C*. Cambridge University Press, 2 édition, 1992.

[PB81] Stephen M. Platt et Norman I. Badler. « Animating facial expressions ». *Computer Graphics*, 15(3), August 1981. SIGGRAPH'81 Proceedings.

[PFPB92] Jihun Park, Donnald Fussel, Marcus Pandy, et James C. Browne. « Realistic animation using musculotendon skeletal dynamics and suboptimal control ». Dans *Third Eurographics Workshop on Animation and Simulation*, Cambridge, England, septembre 1992.

[Phy57] *Physiologie musculaire*. Éditions médicales et universitaire, 1957. Unité d'enseignement et de recherche en médecine de l'académie de Paris. Polycopié des étudiants en médecine.

[Pér95] Pérez José-Phillipe. *Mécanique ; points matériels, solides, fluides*. Masson, 1995.

[PW89] A. Pentland et J. Williams. « Good vibrations: modal dynamics for graphics animation ». *Computer Graphics*, 23(3):215–222, July 1989. SIGGRAPH'89 Proceedings.

[RB85] William T. Reeves et Ricki Blau. « Approximate and probabilistic algorithms for shading and rendering structured particles systems ». *Computer Graphics*, 19(3), July 1985.

[Ree83] William T. Reeves. « Particle Systems - A technique for modelling a class of fuzzy objects ». *Computer Graphics*, 17(3), July 1983.

[Sam90] Hanan Samet. *The design and analysis of spatial data strucures*. Addison-Wesley, 1990.

[Sho85] Ken Shoemake. « Animating rotation with quaternion curves ». *Computer Graphics*, 19(3):245–254, July 1985. SIGGRAPH'85 Proceedings.

[Sho91] Ken Shoemake. Quaternions and 4x4 matrices. Dans James Arvo, éditeur, *Graphics Gems II*, p. 351–354. Academic Press, 1991.

[ST92] Richard Szelisky et David Tonnesen. « Surface Modelling with Oriented Particle Systems ». *Computer Graphics*, 26(2):185–194, July 1992.

[Str92] Bjarne Stroustrup. *Le langage C++*. Addison-Wesley, 1992.

[TF88] Demetri Terzopoulos et Kurt Fleischer. « Modeling inelastic deformations: Viscoelasticity, plasticity, fracture ». *Computer Graphics*, 22(4):269–278, August 1988.

[TG94] Nicolas TSINGOS et Marie-Paule GASCUEL. « Un modeleur interactif d'objets définis par des surfaces implicites ». Dans *Deuxièmes journées de l'Association Française d'Informatique Graphique*, p. 151–159, Toulouse, 30 Novembre - 2 Décembre 1994.

[TG95] Nicolas TSINGOS et Marie-Paule GASCUEL. « Un Modeleur Intercatif d'Objets Définis par des Surfaces Implicites ». Dans *Revue Internationale de CFAO et d'Informatique Graphique*, volume 10(4), p. 355–366. Hermès, Paris, octobre 1995.

[TPBF87] Demetri TERZOPOULOS, John PLATT, Alan BARR, et Kurt FLEISCHER. « Elastically deformable models ». *Computer Graphics*, 21(4):205–214, 1987.

[TPF89] Demetri TERZOPOULOS, John PLATT, et Kurt FLEISCHER. « Heating and melting deformable models (from goop to glop) ». *Graphics Interface*, p. 219–226, 1989.

[Tur92] Greg TURK. « Re-tiling polygonal surfaces ». *Computer Graphics*, 2(26), July 1992.

[TW88] Demetri TERZOPOULOS et Andrew WITKIN. « Physically-based models with rigid and deformable components ». Dans *Graphics Interface '88*, Edmonton, Canada, June 1988.

[Via92] Marie-Luce VIAUD. « *Animation faciale avec rides d'expressions viellissement et parole* ». Thèse de doctorat, Université Paris Sud, centre d'Orsay, Novembre 1992.

[Wat87] Keith WATERS. « A muscle model for animating three-dimensional facial expression ». *Computer Graphics*, 21(4):17–24, July 1987. SIGGRAPH '87 Proceedings.

[Wat92] Keith WATERS. « A physical model of facial tissue and muscle articulation derived from computer tomography data ». *Visualization and Biomedical Computing*, 1808:574–583, 1992.

[Wer94] Josie WERNECKE. *The Inventor mentor*. Addison-Wesley, 1994.

[WH91] Jakub WEJCHERT et David HAUMANN. « Animation aerodynamics ». *Computer Graphics*, 25(4):19–22, July 1991.

[WH94] Andrew P. WITKIN et Paul S. HECKBERT. « Using Particles to Sample and Control Implicit Surfaces ». *Computer Graphics*, p. 269–278, juillet 1994. SIGGRAPH '94 Proceedings.

[WMW86] Geoff WYVILL, Craig MCPHEETERS, et Brian WYVILL. « Data Structure for Soft Objects ». *The Visual Computer*, p. 227–234, août 1986.

[WW89] Brian WYVILL et Geoff WYVILL. « Field Functions for Implicit Surfaces ». *The Visual Computer*, 5:75–82, décembre 1989.

[Zaj89] Felix E. ZAJAC. « Muscle and Tendon: Properties, Model, Scaling and Application to Biomechanics and Motor Control ». *Critical Review in Biomedical Engineering*, 17(Issue 4):359–410, 1989.

[ZTS86] F.E. ZAJAC, E.L. TOPP, et P.J. STEVENSON. « A dimensionless musculotendon model ». *Eight annual Conference of the Engineering in Medicine and Biology Society*, p. 601–604, 1986.