



**HAL**  
open science

# Modélisation et Animation de Matériaux Hautement Déformables en Synthèse d'Images

Mathieu Desbrun

► **To cite this version:**

Mathieu Desbrun. Modélisation et Animation de Matériaux Hautement Déformables en Synthèse d'Images. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1997. Français. NNT: . tel-00528678

**HAL Id: tel-00528678**

**<https://theses.hal.science/tel-00528678v1>**

Submitted on 22 Oct 2010

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

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

---

# Modélisation et Animation de Matériaux Hautement Déformables en Synthèse d'Images

---

Mathieu DESBRUN

Thèse préparée au sein du laboratoire *iMAGIS / GRAVIR-IMAG*, projet commun entre le *CNRS*, l'*INRIA*, l'*INPG* et l'*UJF*, et présentée pour obtenir le titre de :

**Docteur de l'Institut National Polytechnique de Grenoble  
spécialité Informatique**

*Arrêté ministériel du 30 mars 1992*

Soutenue le 9 Décembre 1997 devant la commission d'examen :

Demetri	TERZOPOULOS	Rapporteur
Gérard	HÉGRON	Rapporteur
Daniel	THALMANN	
Hervé	DELINGETTE	
Jean-Pierre	VERJUS	
Claude	PUECH	
Marie-Paule	CANI-GASCUEL	Directrice de thèse

*Science doesn't explain; science describes.*

*Science postulates models to describe how some part of nature behaves, then tests and refines that model till it works as well as we can measure (as evidenced by repeated, skeptical testing).*

*Science doesn't provide ultimate or absolute answers, but only proximate (good enough) answers.*

*Science can't find truth, but it can eliminate error, and identify things which aren't so, thereby narrowing the region in which truth may reside. In the process, science has produced more reliable knowledge than any other branch of human thought.*

*Pr. Donald E. Simanek, LHU.*

# Remerciements

Trois ans... Comment remercier nommément toutes les personnes dont j'ai croisé le chemin, et qui ont pu m'apporter un enrichissement aussi bien humain que scientifique? A mon grand regret, je ne pourrais être exhaustif, et me contenterais de détailler les principales contributions qui ont fait de moi un privilégié.

Je remercie dans un premier temps Gérard Hégron et Demetri Terzopoulos qui, malgré un emploi du temps surchargé, ont bien voulu être mes rapporteurs. Je remercie tout autant et pour les mêmes raisons les membres de mon jury : Jean-Pierre Verjus, Daniel Thalmann, Hervé Delingette, et Claude Puech.

Mes remerciements sont bien sûr aussi adressés à Marie-Paule Cani, dont le rôle ne se limita pas à la pure fonction de directrice de thèse. Ses talents comprennent en effet un enthousiasme blindé, un dévouement hors norme, ainsi qu'une clairvoyance étonnante. Il est évident, dans ces conditions, que mon travail sous sa tutelle ait été réellement stimulant. En outre, toutes les opportunités et les expériences professionnelles dont j'ai bénéficié au cours de ces trois ans sont presque systématiquement son œuvre.

Claude Puech aussi m'a constamment apporté ses encouragements et ses conseils avisés. Je tiens à le remercier pour sa gentillesse et son soutien de la première heure, et le féliciter pour la gestion de son équipe hors du commun.

Jean-Dominique Gascuel a largement contribué à ce que ma thèse se déroule convenablement. Magicien, il sait toujours tirer d'affaire ceux qui commencent à douter que l'informatique soit une science exacte. Dévoué, il donne souvent l'indispensable coup de pouce. Bref, il est inestimable.

George Drettakis m'a toujours été de précieux conseils depuis son arrivée au laboratoire, et a bien voulu réhausser mon anglais fort moyen lors de maintes relectures. Sa disponibilité à mon égard m'a été très profitable, preuve en est mon choix entre différents post-docs.

De même, François X. Sillion a été pour moi un exemple de par sa compétence et son amabilité. Lui aussi a su m'accorder du temps au dépend de ses occupations, et m'a soutenu et recommandé de façon grandiose.

Je tiens aussi à remercier Fabrice Neyret, dernier arrivé des permanents d'iMAGIS, pour tout le temps qu'il m'a consacré durant les derniers mois, entre autres pour relire le présent document.

Enfin, je salue Rosine Ah Tchou pour toutes ses qualités, et Jean-Luc Douville pour services rendus.

Je pense aussi à mes compagnons de route, avec par exemple la touche féminine indispensable qu'assument pleinement Céline Loscos et Agata Opalach (j'en profite pour les embrasser au passage ;-), ou la compagnie inégalable de Cyril Soler et Fédé Perez. Aussi bien au travail que dans la "vrai" vie, Nicolas Tsingos s'est révélé un ami dévoué, passionné et passionnant. Il en

fut de même pour Frédo Durand, qui mérite cependant une mention spéciale pour sa gentillesse et son omniscience. Il a par ailleurs partagé mon bureau, supportant tous les jours ma présence, ce qui est une performance notable.

Les anciens doctorants, donc récents docteurs, ont aussi laissé des traces indélébiles de leur passage au labo. Les conseils et la bonne humeur de François Faure (bises à braillette et à Florence), Frédéric Cazals, Stéphane Rivière, Alexiiiis Lamouret, Rachel Orti, Nicolas Holzschuch (bises à Myriam) ou de Jean-Christophe Lombardo (bises à la famille) ont été plus que profitables.

Restent encore tous les stagiaires de passage (Yann, Stéphane, etc...) et les nouveaux doctorants, dont Pierre-François Clerc, que je salue. Un grand merci entre autres à Eric Ferley, pour sa gentillesse absolue, et à Gilles Debunne pour sa bonne humeur communicative. Ils ont tous les deux été des relecteurs consciencieux, et vont, à n'en pas douter, assurer une relève de très haut niveau.

Toujours dans le cadre du travail, j'ai eu la chance de cotoyer par exemple Eric Leroi du BRGM, qui m'a permis d'enchaîner mon service national et ma thèse dans d'excellentes conditions. Je salue aussi quelques personnes telles que Arash Habibi, Annie Luciani, Eric Galin, Annick Montanvert, Jacques-Olivier Lachaud, ou Mahnu Promayon pour les diverses discussions que l'on a pu avoir. Enfin, je remercie Jules Bloomenthal, Brian Wyvill, Alain Fournier, Andrew Hanson, Julie Dorsey, Hans Pedersen et Eugene Fiume (je lui dois une partie des travaux présentés dans ce document) pour leurs conseils et leurs encouragements. Malgré leurs positions et leurs occupations, ces personnes ont pu me consacrer du temps au-delà de mes espérances.

Ce travail doit aussi beaucoup à mes parents et à ma famille au grand complet, que je veux donc remercier ici pour leur soutien affectif.

Enfin, pour l'ensemble de son œuvre passée, présente, et future, je tiens en dernier lieu à saluer Laure France, l'autre moi.

À tous, merci.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>11</b>
1.1	La Synthèse d'Images . . . . .	11
1.1.1	État actuel . . . . .	11
1.1.2	L'animation par images de synthèses . . . . .	12
1.2	Thème de la thèse . . . . .	13
1.2.1	But de nos travaux . . . . .	14
1.2.2	Motivations . . . . .	14
1.2.3	Critères importants . . . . .	14
1.3	Contenu . . . . .	15
<b>I</b>	<b>État de l'art</b>	<b>17</b>
<b>2</b>	<b>Animation de matières déformables</b>	<b>19</b>
2.1	Objets déformables structurés . . . . .	19
2.1.1	Approches issues de la mécanique des milieux continus . . . . .	19
2.1.2	Approches discrètes . . . . .	23
2.1.3	Approches globales . . . . .	26
2.1.4	Approches stratifiées . . . . .	27
2.2	Objets déformables non structurés . . . . .	29
2.2.1	Approches à base de particules . . . . .	30
2.2.2	Approches empruntées à l'Hydrodynamique . . . . .	35
2.3	Gestion des interactions entre objets déformables . . . . .	37
2.3.1	On dispose d'une surface externe . . . . .	37
2.3.2	On ne dispose pas de surface externe : le cas des particules . . . . .	40
2.4	Discussion . . . . .	41
<b>3</b>	<b>Modélisation par surfaces implicites</b>	<b>43</b>
3.1	Définition d'une surface implicite . . . . .	43
3.2	Historique des formulations . . . . .	44
3.2.1	Potentiel analytique . . . . .	44
3.2.2	Potentiel à points clés . . . . .	44
3.2.3	Potentiel à squelettes . . . . .	45
3.2.4	Potentiel procédural . . . . .	46
3.2.5	Potentiel intégral . . . . .	46
3.2.6	Potentiel composé . . . . .	47

3.2.7	Potentiel sous forme de R-fonctions . . . . .	47
3.2.8	Potentiel optimisé . . . . .	49
3.3	Échantillonnage pour la visualisation . . . . .	49
3.3.1	Discrétisation polygonale par partitionnement spatial . . . . .	49
3.3.2	Échantillonnage par particules . . . . .	50
3.4	Contrôle de la forme . . . . .	51
3.5	Outils divers . . . . .	52
3.5.1	Reconstruction à partir de points de donnée épars . . . . .	52
3.5.2	Décoration par placages de textures . . . . .	52
3.5.3	Rendu d'images de qualité . . . . .	52
3.5.4	Morphing entre objets implicites . . . . .	53
3.5.5	Conversion implicite/explicite, et vice-versa . . . . .	53
3.6	Ajout d'un modèle de déformation . . . . .	53
3.6.1	Présentation du modèle . . . . .	53
3.6.2	Modélisation des propriétés physiques d'un objet implicite . . . . .	54
3.6.3	Algorithme global . . . . .	56
3.6.4	Mise en œuvre . . . . .	56
3.6.5	Intérêts et limitations d'un tel modèle . . . . .	59
3.7	Discussion . . . . .	59
3.7.1	Implicite vs. Paramétrique . . . . .	59
3.7.2	Adéquation à l'animation d'objets hautement déformables . . . . .	60

## II Premier modèle stratifié d'objets très déformables 61

4	Mise au point d'une approche générique <span style="float: right;">63</span>
4.1	Positionnement du problème . . . . . <span style="float: right;">63</span>
4.1.1	Que signifie <i>déformable</i> et <i>hautement déformable</i> ? . . . . . <span style="float: right;">63</span>
4.1.2	Comportement macroscopique des matériaux déformables . . . . . <span style="float: right;">64</span>
4.1.3	Comportement des objets hautement déformables . . . . . <span style="float: right;">66</span>
4.2	Vers un modèle stratifié . . . . . <span style="float: right;">66</span>
4.2.1	Modèles minimalistes . . . . . <span style="float: right;">66</span>
4.2.2	Modèles hiérarchiques . . . . . <span style="float: right;">67</span>
4.2.3	Modèles à couches . . . . . <span style="float: right;">67</span>
4.2.4	Découpage en couches indépendantes . . . . . <span style="float: right;">68</span>
4.2.5	À la lumière de l'état de l'art . . . . . <span style="float: right;">68</span>
4.3	Proposition d'un modèle hybride à trois couches . . . . . <span style="float: right;">68</span>
4.3.1	<i>L'Ossature</i> : modèle interne . . . . . <span style="float: right;">69</span>
4.3.2	<i>La Chair</i> : habillage déformable . . . . . <span style="float: right;">69</span>
4.3.3	<i>La Peau</i> : interface extérieure . . . . . <span style="float: right;">69</span>
4.3.4	Échanges entre couches . . . . . <span style="float: right;">70</span>
4.3.5	Validité et efficacité . . . . . <span style="float: right;">70</span>
4.3.6	Généricité . . . . . <span style="float: right;">71</span>

<b>5</b>	<b>Un premier modèle de matériaux hautement déformables</b>	<b>73</b>
5.1	Une ossature définie par un système de particules . . . . .	73
5.1.1	Choix d'une formulation des lois d'interaction . . . . .	73
5.1.2	Mise au point d'un frottement visqueux . . . . .	74
5.1.3	Complexité algorithmique du bilan des forces . . . . .	75
5.2	Adaptation du modèle implicite élastique . . . . .	76
5.2.1	D'une ossature discrète à une surface continue . . . . .	76
5.2.2	Utilisation du modèle élastique implicite [Gas93] . . . . .	76
5.2.3	Gestion combinée des couches . . . . .	77
5.3	Échantillonnage de la couche implicite déformable . . . . .	78
5.3.1	Intérêt d'une méthode exploitant la cohérence temporelle . . . . .	78
5.3.2	Échantillonnage adapté aux objets implicites sécables à squelettes . . . . .	79
5.3.3	Avantages découlant d'un tel échantillonnage . . . . .	84
5.3.4	Discussion . . . . .	87
5.4	Gestion des mélanges indésirables . . . . .	87
5.4.1	Cas des objets ne changeant pas de topologie . . . . .	87
5.4.2	Cas des objets pouvant subir des fractures . . . . .	89
5.4.3	Modification du traitement des collisions . . . . .	90
5.4.4	Discussion . . . . .	91
5.5	Contrôle du volume . . . . .	91
5.5.1	Tentative de résolution analytique du problème . . . . .	92
5.5.2	Détection des variations locales de volume . . . . .	92
5.5.3	Correction du volume . . . . .	93
5.5.4	Discussion . . . . .	94
5.6	Champ d'applications . . . . .	94
5.6.1	Objets hautement déformables . . . . .	95
5.6.2	Personnages articulés . . . . .	95
5.6.3	Éboulement . . . . .	95
5.6.4	Autres exemples . . . . .	96
5.7	Discussion . . . . .	97
<b>III</b>	<b>Modélisation approfondie des différentes couches</b>	<b>99</b>
<b>6</b>	<b>Un modèle de particules adaptatives en temps et espace</b>	<b>101</b>
6.1	Introduction . . . . .	101
6.1.1	Défauts des modèles existants . . . . .	102
6.1.2	Apparition de modèles adaptatifs . . . . .	102
6.1.3	Nécessité d'un modèle adaptatif . . . . .	103
6.2	Un formalisme Lagrangien nommé SPH . . . . .	103
6.2.1	Discrétisation de l'espace: Euler vs. Lagrange . . . . .	104
6.2.2	Formulation discrète de champs continus . . . . .	104
6.2.3	Équation d'état - Forces de pression . . . . .	105
6.2.4	Discussion . . . . .	106
6.3	Un nouveau modèle déformable à base de densité . . . . .	106
6.3.1	Définition d'une équation d'état . . . . .	106
6.3.2	Choix d'un filtre de lissage . . . . .	107



6.3.3	Contrôle différentiel de la densité . . . . .	109
6.3.4	Implémentation de base . . . . .	110
6.4	Vers une simulation adaptative en espace-temps . . . . .	110
6.4.1	Idée de base d'un modèle adaptatif . . . . .	110
6.4.2	Intêret de la formulation SPH . . . . .	111
6.4.3	Simulation avec des particules non homogènes . . . . .	112
6.4.4	Corrélation entre espace et temps . . . . .	113
6.5	Adaptation automatique de la discrétisation spatiale . . . . .	113
6.5.1	"Importance Sampling" . . . . .	113
6.5.2	Critères d'adaptation . . . . .	114
6.5.3	Raffinement de la discrétisation . . . . .	114
6.5.4	Simplification de la discrétisation . . . . .	115
6.5.5	Critères annexes . . . . .	117
6.6	Adaptation automatique de la discrétisation temporelle . . . . .	117
6.6.1	Mécanisme global de l'adaptation . . . . .	117
6.6.2	Critères pour les pas de temps individuels . . . . .	118
6.6.3	Mise en œuvre . . . . .	118
6.7	Algorithme complet d'animation . . . . .	118
6.8	Implémentation et résultats . . . . .	119
6.8.1	Recherche des particules voisines . . . . .	119
6.8.2	Discrétisation initiale . . . . .	120
6.8.3	Résultats . . . . .	120
6.9	Conclusions et développements possibles . . . . .	122
<b>7</b>	<b>Peau implicite active pour l'habillage</b>	<b>125</b>
7.1	Notion de peau . . . . .	125
7.1.1	L'importance du paraître . . . . .	125
7.1.2	L'importance du rôle physique . . . . .	125
7.1.3	L'importance de l'efficacité . . . . .	126
7.2	Problématique . . . . .	126
7.2.1	Inadéquation d'un habillage purement géométrique . . . . .	126
7.2.2	Nécessité d'une peau active . . . . .	126
7.2.3	Autres caractéristiques souhaitées . . . . .	127
7.2.4	Formulation implicite . . . . .	127
7.3	Approches antérieures de surfaces déformables . . . . .	127
7.3.1	Approches Lagrangiennes . . . . .	128
7.3.2	Approches Euleriennes . . . . .	128
7.3.3	Discussion . . . . .	128
7.4	Peau Implicite Mouvante . . . . .	129
7.4.1	Définition d'une surface implicite mouvante . . . . .	129
7.4.2	Inadéquation d'une représentation par squelettes . . . . .	129
7.4.3	Intêret d'un champ discret . . . . .	129
7.4.4	Comportement différentiel du champ pour une surface animée . . . . .	130
7.4.5	Définition d'une stratégie pour la peau . . . . .	130
7.4.6	Considérations simplificatrices . . . . .	132
7.5	Mise en œuvre . . . . .	133
7.5.1	Approche générale de la mise en œuvre . . . . .	133

7.5.2	Structure de données . . . . .	133
7.5.3	Extensions du champ de vitesses . . . . .	134
7.5.4	Ajout de la tension de surface . . . . .	136
7.5.5	Mise à jour du voisinage tubulaire . . . . .	136
7.5.6	Construction du maillage et de ses normales . . . . .	137
7.5.7	Algorithme global . . . . .	137
7.5.8	Pas de temps . . . . .	137
7.6	Extensions des propriétés de la peau . . . . .	138
7.6.1	Détection d'appartenance accélérée . . . . .	138
7.6.2	Surface constante . . . . .	139
7.6.3	Volume constant . . . . .	139
7.7	Résultats . . . . .	139
7.7.1	Morphing . . . . .	139
7.7.2	Habillage de particules . . . . .	141
7.8	Conclusion et perspectives . . . . .	142
<b>8</b>	<b>Conclusions</b> . . . . .	<b>145</b>
8.1	Contributions . . . . .	145
8.1.1	Mise au point d'un modèle global stratifié . . . . .	145
8.1.2	Émergence d'un modèle de particules adaptatives . . . . .	145
8.1.3	Définition d'une peau active avec formulation implicite discrète . . . . .	146
8.2	Réflexions et bilan . . . . .	146
8.2.1	Synthèse du mouvement . . . . .	146
8.2.2	Des surfaces implicites . . . . .	148
8.2.3	Bilan . . . . .	149
8.3	Horizons . . . . .	149
8.3.1	Modèle adaptatif de simulation . . . . .	149
8.3.2	Peau implicite dynamique . . . . .	150
8.4	Conclusions . . . . .	150
<b>IV</b>	<b>Annexes</b> . . . . .	<b>151</b>
<b>A</b>	<b>Euler vs. Lagrange</b> . . . . .	<b>153</b>
A.1	Vision Lagrangienne . . . . .	153
A.2	Vision Eulerienne . . . . .	153
A.3	Expression de l'accélération en formalisme Lagrangien . . . . .	154
A.4	Expression de l'accélération en formalisme Eulerien . . . . .	154
A.5	Avantages et inconvénients des deux approches . . . . .	155
A.5.1	Approche Lagrangienne . . . . .	155
A.5.2	Approche Eulerienne . . . . .	155
<b>B</b>	<b>Notions d'hydrodynamique</b> . . . . .	<b>157</b>
B.1	L'hydrodynamique en général . . . . .	157
B.1.1	Grandeurs significatives . . . . .	157
B.1.2	Propriété notable: l'équation d'état . . . . .	158
B.1.3	Équations fondamentales . . . . .	158

---

B.2	Conservation de la masse . . . . .	158
B.2.1	Fluide incompressible . . . . .	159
B.3	Équation de la dynamique . . . . .	159
B.4	Vitesse de propagation à l'intérieur d'un fluide . . . . .	159
<b>C</b>	<b>Notions sur les équations différentielles</b>	<b>161</b>
C.1	Équations différentielles ordinaires du premier ordre . . . . .	161
C.1.1	Méthode d'Euler . . . . .	162
C.1.2	Méthode de Runge-Kutta . . . . .	162
C.1.3	Méthode implicite ou semi-implicite . . . . .	163
C.1.4	Méthode de Newton-Cotes . . . . .	163
C.2	Équations aux dérivées partielles . . . . .	164
C.2.1	Cas hyperbolique - Équation de propagation . . . . .	164
C.2.2	Cas parabolique - Équation de diffusion . . . . .	166
<b>D</b>	<b>Formulation des courbures d'une surface implicite</b>	<b>169</b>
D.1	Différentielles du second ordre et courbures . . . . .	169
D.2	Calcul direct des courbures moyenne et gaussienne . . . . .	170

# Chapitre 1

## Introduction

*Urge to come to terms with the "Outside",  
By absorbing, interiorizing it.  
I won't come out, you must come in to me.  
Into my womb-garden where I peer out.  
Where I can construct a universe within the skull,  
To rival the real.*

*James Douglas Morrison*

L'image est aujourd'hui plus que jamais un vecteur d'expression et de fascination de tout premier ordre. L'art permettait déjà de représenter une vision de la réalité ou une pure chimère issue de l'imagination de l'artiste. La Synthèse d'Images prolonge cette magie, permettant à la fois de représenter plus finement la complexité du réel et d'inventer des mondes nouveaux. Usant et abusant de notre sens le plus développé, la vue, l'image de synthèse n'en finit pas de rivaliser avec le réel.

### 1.1 La Synthèse d'Images

La Synthèse d'Images est née dès que les ordinateurs ont été capables de représenter des formes géométriques sur un écran. Très limitée à ses débuts par les capacités graphiques des machines, son essor a été foudroyant : en l'espace d'une trentaine d'années, le domaine a tellement progressé qu'il devient souvent impossible de distinguer les images calculées par ordinateur des images réelles. Les conséquences sont bien sûr multiples : tous les domaines scientifiques, techniques, ou médiatiques souhaitent utiliser ces compétences dans des buts très distincts, allant d'une meilleure visualisation de phénomènes complexes à l'utilisation de trucages cinématographiques.

#### 1.1.1 État actuel

Aujourd'hui, l'engouement pour la Synthèse d'Images est réel. Promue par les médias qui y trouvent un outil d'une puissance encore inconnue permettant de créer de nouvelles images, l'image de synthèse plaît à beaucoup pour son aspect ludique. Mais les perspectives réelles de

ce domaine sont bien plus vastes.

Le premier avantage de l'image de synthèse est son internationalité. La mondialisation des communications rend indispensable l'utilisation d'*images*, capables à elles seules de faire passer des messages complexes, sans barrière due au langage.

Plus généralement, l'image de synthèse permet plus que jamais la *visualisation* de toute forme de données. Dans les domaines tels que l'industrie automobile, ou l'architecture, visualiser un produit *avant même sa construction réelle* permet un progrès substantiel en évitant la phase lourde de fabrication de prototypes. Le développement d'un produit nouveau se fait donc plus vite, et nécessite bien moins d'investissements financiers.

Les possibilités de *simulation* ont été exploitées dès l'apparition des premières images de synthèse, mais gardent encore aujourd'hui un potentiel important de développement. Simulateurs de conduite, simulateurs d'opérations chirurgicales : autant d'outils pour lesquels la Synthèse d'Images permet déjà ou dans un proche avenir une solution peu onéreuse, supplantant fort avantageusement la formation initiale sur le terrain.

De nos jours, on ne peut pas négliger l'intérêt ludique des images de synthèse. Des sommes d'argent considérables sont investies pour le développement de jeux vidéo ou de trucages audiovisuels, promettant des retombées importantes pour tout le domaine. Enfin, les artistes commencent aussi à trouver que l'ordinateur peut faire un bon support d'œuvre grâce à l'étendue de ses capacités.

Mais l'un des principaux enjeux actuels est la *réalité virtuelle*. Derrière cette appellation antinomique se cache le fait que la Synthèse d'Images peut dès à présent dépasser le réel : grâce à l'ordinateur, et aux progrès réalisés, nous sommes capables de donner "vie" à des mondes inexistants, ou n'existant plus. Gérer l'existence virtuelle des objets ou des êtres est alors le rôle de l'*animation*.

### 1.1.2 L'animation par images de synthèses

Pour créer l'illusion du mouvement, l'animation fait se succéder une suite d'instantanés. Il n'y a pas si longtemps encore, ces animations se réduisaient à des dessins animés entièrement faits à la main. La mise en œuvre de tels films demandait une quantité de travail énorme, car il faut bien garder à l'esprit que les 25 images par seconde nécessaires à une relative fluidité du mouvement imposent 90000 dessins à l'heure ! Depuis, la production de dessins animés s'est automatisée avec l'utilisation d'interpolation entre des images clés : le graphiste en chef crée le scénario ainsi qu'environ une image sur vingt. Ce sont alors d'autres graphistes, appelés "in-betweeners", qui s'occupent des images intermédiaires.

#### Premiers pas

Ce travail d'intervalliste pouvant dans certains cas être très automatique, l'idée d'utiliser des ordinateurs pour remplacer l'homme fit son chemin. Les premières productions informatiques se limitaient à l'animation, par des techniques d'interpolation de plus en plus poussées, d'images ou de *positions clés*. Les recherches sur ce domaine de l'informatique graphique restent encore à ce jour actives mais malheureusement, de tels procédés restent limités. Notamment, les scènes complexes, où apparaissent un nombre élevé d'objets ou de personnages en mouvement, se prêtent difficilement à ce genre de techniques.

## Apparition des modèles générateurs

À l'inverse de ces procédés *descriptifs*, sont apparus des modèles dits *générateurs*, capables de générer automatiquement le mouvement et les déformations. La simulation de lois physiques en est une bonne illustration : par exemple, la trajectoire et les déformations d'un ballon peuvent être facilement simulées. De cette manière, le graphiste n'a plus qu'à fournir le modèle des objets, les conditions initiales et les contraintes qu'il souhaite voir respectées : le ballon prend alors "vie". Ainsi, les mouvements et les chocs sont *automatiquement gérés*, et peuvent être *modifiés* à l'aide de divers paramètres.

## Finalité et enjeux de l'animation

Il est délicat, voire stérile, de chercher à définir un but précis et unique pour l'animation par ordinateur à l'heure actuelle. *Définir des modèles d'animation plausibles et efficaces* peut être une définition suffisamment générale pour convenir. Or, cela semble aussi englober la simulation physique. Pourtant, l'animation se distingue considérablement de la simulation physique par *ses critères de qualités*.

La qualité d'une simulation physique, intégrant des équations modélisant le phénomène spécifique que l'on cherche à simuler, se jugera sur sa similarité avec les données numériques d'expérimentations réelles. Quant à l'animation en synthèse d'images, elle doit surtout tenir compte de l'**efficacité** et du **réalisme visuel** résultant : on doit donner l'illusion d'un comportement réel. Bien sûr, ces deux approches ne sont en rien antagonistes : l'idéal est évidemment une simulation précise avec un rendu réaliste. De toute façon, le réalisme visuel du mouvement demande bien souvent un respect des lois physiques fondamentales qui sont de rigueur tout autour de nous : cela explique que bien des techniques d'animation se classent dans la catégorie *modèles basés sur la physique*. Une autre grande classe d'approches peut être qualifiée de *phénoménologique*, dans la mesure où la modélisation cherche simplement à reproduire, de façon simple, un phénomène observé, proposant ainsi une alternative souvent fort efficace.

## 1.2 Thème de la thèse

On vient de le voir, la Synthèse d'Images est un domaine en plein essor. Le réalisme actuel des animations et des simulations d'éclairage a déjà des applications concrètes, et fait espérer une utilisation bien plus importante dans les prochaines années, ne serait-ce qu'au cinéma avec les effets spéciaux, mais plus généralement pour tout ce qui concerne la réalité virtuelle.

L'animation en Synthèse d'Images devient par exemple un outil de simulation incontournable, entre autres pour la chirurgie : simuler "virtuellement" une opération délicate permet de former les futurs praticiens de façon simple, efficace, peu onéreuse, et surtout moins risquée. De telles simulations nécessitent de modéliser et d'animer par ordinateur des objets déformables, voire très déformables, que l'on puisse saisir ou couper. De même, l'animation de matériaux très façonnables comme la pâte à modeler ou la terre glaise, qui servent à la mise au point de prototypes pour les voitures ou pour le design en général, demande des spécificités indispensables, comme la gestion de fortes déformations ou de séparations de la matière. Enfin, la simulation d'avalanches ou de glissements de terrains nécessite aussi de savoir aisément manipuler des matériaux très déformables à grande échelle.

Or, animer *efficacement* des objets très déformables n'est pas simple. Les modélisations classiques de la physique ne pouvant traiter en temps réel les calculs nécessaires à la simulation de mouvements complexes, il est du ressort de l'animation par Synthèse d'Images de proposer des modèles efficaces et robustes qui permettent d'obtenir des mouvements visuellement réalistes. Un certain nombre de modèles permettent d'ores et déjà d'obtenir des comportements élastiques, à savoir des objets déformables reprenant leur forme après un choc. D'autres, moins nombreux, simulent des comportements inélastiques, comme la visco-élasticité ou la semi-plasticité, c'est-à-dire des objets dont la structure interne peut être irréversiblement affectée lors d'un choc. Beaucoup de ces modèles sont issus de méthodes physiques simplifiées; pourtant, aucun modèle actuellement ne peut se prévaloir d'animer efficacement des objets hautement déformables capables de changement de topologie avec un traitement soigné des contacts avec d'autres objets de la scène. C'est dans ce cadre que s'inscrit cette thèse : *offrir des modèles d'animation générant automatiquement des mouvements réalistes et paramétrables de matériaux capables aussi bien d'épouser la forme des objets qui leur sont en contact que de se fracturer en plusieurs morceaux.*

### 1.2.1 But de nos travaux

Le but de cette thèse est de proposer une approche pour la modélisation et l'animation de matériaux hautement déformables, mi-solides, mi-liquides, particulièrement difficiles à synthétiser à l'aide des techniques traditionnelles utilisées en Images de Synthèse.

Il s'agira de présenter des modèles simples, efficaces et robustes permettant la synthèse de ces matériaux déformables à grand champ de déformation, capables de fractures et de fusions. Nous essayerons de montrer en quoi des techniques récentes, comme la modélisation par surfaces implicites, sont indispensables à la mise au point de tels modèles.

### 1.2.2 Motivations

Le choix de matériaux hautement déformables est particulièrement crucial. La *complexité* des mouvements et des formes dont ces objets sont capables en fait un but intéressant à plusieurs égards.

Tout d'abord, la physique ne sait que difficilement traiter de telles déformations, à mi-chemin entre physique du solide et mécanique des fluides. Le recours à la synthèse d'images est donc de toute façon une nécessité. Mais même en animation, manipuler des objets sans forme et tellement changeants devient problématique. Des problèmes difficiles, comme la conservation au cours du temps d'un volume constant, le maintien d'une représentation fine quelque soit la forme courante du matériau, et la gestion des caractéristiques topologiques susceptibles d'évoluer, apparaissent de façon flagrante et incontournable.

C'est aussi pour des objets aux comportements complexes que la nécessité d'algorithmes d'animation efficaces, capables de concentrer automatiquement les calculs dans les régions qui en ont besoin tout en épargnant du travail dans les zones moins critiques, apparaît réellement. Adapter la résolution des simulations en Synthèse d'Images est un défi actuel en simulation d'éclairage de scènes fixes, mais est encore très peu répandu en animation.

### 1.2.3 Critères importants

Les critères que l'on souhaite voir remplis sont directement issus de la problématique générale de la synthèse de mouvement. On insistera donc en premier lieu sur l'aspect **réalisme visuel** que tout modèle de synthèse doit avoir, aussi bien pour le rendu que pour le mouvement lui-même.

La **simplicité de modélisation et de paramétrage** est aussi très importante pour en faire un outil utile aux animateurs : sans des paramètres intuitifs, un modèle ne pourra être manipulé que par des spécialistes, ce qui restreindrait fortement son intérêt.

Un troisième critère incontournable est l'**interactivité** que doit offrir la mise en œuvre. Un modèle n'est humainement utilisable que s'il permet une rapidité suffisante pour offrir des résultats en des temps raisonnables. Sans parler de contraintes temps-réel, l'efficacité des calculs est l'un des points importants à aborder.

Le côté **générique** d'un modèle est bien sûr souhaitable, dans la mesure où l'on désire ne pas submerger les animateurs de modèles ad-hoc pour chaque type de comportement : l'étendue des possibilités d'un modèle en fait souvent son intérêt pratique.

Enfin, un dernier critère qui, en tant que programmeur, ne nous laisse pas du tout indifférent, est la **facilité d'implémentation et de développements futurs**. Ce critère est dans les faits une règle fondamentale pour les entreprises développant des logiciels d'animation : la gestion du développement et de l'évolution d'un produit n'est pas tâche aisée, et mérite une réflexion initiale indispensable à la rentabilité et à la pérennité du logiciel.

## 1.3 Contenu

Ce document s'articule en trois parties où huit chapitres se succèdent comme suit. Dans la première partie, nous passerons en revue l'état de l'art actuel en animation d'objets déformables au chapitre 2, ce qui nous amènera à considérer aussi les travaux en modélisation par surfaces implicites au chapitre 3.

Dans une deuxième partie, cette présentation des travaux antérieurs, orientée vers nos besoins, permettra lors du chapitre 4 de présenter une approche globale sous forme de *modélisation stratifiée*, superposant plusieurs modèles à différentes échelles. Le chapitre suivant proposera alors une première mise en œuvre d'un *modèle générique*, où des solutions seront apportées à bon nombre de problèmes classiques en animation et en modélisation par surfaces implicites.

Enfin, la troisième partie consistera à présenter des alternatives permettant d'améliorer encore le modèle obtenu. Ainsi, le chapitre 6 expose la mise au point d'un *modèle de simulation adaptatif en temps et en espace*, optimisant les temps de simulation en concentrant la puissance de calcul dans les régions en mouvement. Le dernier chapitre présente un modèle de *peau implicite active*, capable de simuler une tension de surface autour du modèle adaptatif tout en offrant des temps de calcul et de rendu réduits.

Ce document s'achèvera par une conclusion générale au chapitre 8, suivie de quatre annexes expliquant quelques notions hors du champ classique de la Synthèse d'Images, qui nous ont pourtant été indispensables pour mener à bien notre recherche.





Première partie

État de l'art



## Chapitre 2

# Animation de matières déformables

L'intérêt pour les modèles de matières déformables en animation par systèmes générateurs n'a cessé de croître ces dernières années. On tente par divers moyens de générer des mouvements réalistes en utilisant des paramètres physiques tels que la raideur ou le taux de compressibilité. Les comportements élastiques aussi bien que plastiques sont abordés sous des angles très divers, créant ainsi de nombreux modèles bien distincts.

Nous tenterons dans ce chapitre de résumer les principales approches d'animation de matières déformables existantes à ce jour en Synthèse d'Images. La classification a volontairement été axée sur les méthodes de résolution utilisées, pour permettre de voir les avantages et inconvénients de chaque modèle sous un angle algorithmique. Le champ des modèles passés en revue est sciemment plus vaste que celui qui nous concerne, puisqu'il va des objets élastiques jusqu'à la simulation de l'eau. Mais il permet ainsi d'avoir une vue plus globale des techniques existantes en animation, ce qui est évidemment souhaitable à défaut d'être obligatoire.

### 2.1 Objets déformables structurés

Les objets déformables structurés, c'est-à-dire dont la topologie ne change pas en cours de déformation, ont été les premiers modèles déformables développés en animation, comme extension des modèles rigides. Une première série de ces modèles d'objets déformables relève de ce que l'on peut nommer une "approche nodale" : la résolution numérique nécessite un maillage des objets. Les "nœuds" de discrétisation vont propager les déformations à travers tout l'objet, suivant les équations physiques choisies pour la modélisation. Toute déformation sera donc exprimée par le déplacement d'un certain nombre de nœuds à l'intérieur de l'objet. Deux types d'approches se distinguent à ce stade : les modèles issus d'équations différentielles physiques de milieux continus, et les modèles construits dès le départ à partir de composants mécaniques discrets.

#### 2.1.1 Approches issues de la mécanique des milieux continus

Un bon nombre d'approches pour l'animation de matériaux déformables en Synthèse d'Images se sont inspirés des équations de déformation issues de la théorie mécanique des milieux continus. La résolution se fait en discrétisant la matière constituant l'objet, puis en ré-exprimant les équations de manière discrète en chaque nœud du maillage. Pour ce faire, on utilise généralement le schéma des différences finies qui approximent les dérivées successives d'une fonction par des

combinaisons linéaires des valeurs de cette dernière aux nœuds voisins. Cela peut s'exprimer par ces équations :

$$\frac{df}{dx}(x_0, y_0, z_0) = \frac{f(x_0 + \Delta x, y_0, z_0) - f(x_0 - \Delta x, y_0, z_0)}{2 \cdot \Delta x} + \mathcal{O}(\Delta^2 x)$$

$$\frac{df}{dy}(x_0, y_0, z_0) = \frac{f(x_0, y_0 + \Delta y, z_0) - f(x_0, y_0 - \Delta y, z_0)}{2 \cdot \Delta y} + \mathcal{O}(\Delta^2 y)$$

$$\frac{d^2 f}{dx^2}(x_0, y_0, z_0) = \frac{f(x_0 + \Delta x, y_0, z_0) + f(x_0 - \Delta x, y_0, z_0) - 2 \cdot f(x_0, y_0, z_0)}{\Delta^2 x} + \mathcal{O}(\Delta^2 x)$$

et ainsi de suite. De la sorte, et des fois au prix d'une linéarisation, de toute équation physique modélisant le comportement d'un objet continu on obtient une *équation matricielle* correspondante, permettant de simuler ce comportement à l'aide d'une discrétisation de la matière. Le même type d'équation matricielle est obtenu en utilisant les *éléments finis* [Bat82], technique désormais classique en ingénierie.

Les modèles qui en résultent animent souvent de façon très réaliste des objets dans un premier temps élastiques, mais aussi des objets plastiques, c'est-à-dire absorbant une partie des déformations subies, comme nous allons le détailler.

### a) Un premier modèle pour objets élastiques

Le premier modèle de matériau régi par des équations différentielles caractérisant l'élasticité est présenté en 1987 par Terzopoulos, Platt, Barr et Fleischer [TPBF87]. L'objet, de paramétrisation  $\mathbf{r}(\mathbf{a}, t)$  où  $\mathbf{a}$  est un vecteur de coordonnées intrinsèques, est gouverné par l'équation Lagrangienne suivante :

$$\frac{\partial}{\partial t}(\mu(\mathbf{a}) \frac{\partial \mathbf{r}}{\partial t}(\mathbf{a}, t)) + \gamma(\mathbf{a}) \frac{\partial \mathbf{r}}{\partial t}(\mathbf{a}, t) + \frac{\partial \varepsilon(\mathbf{r})}{\partial \mathbf{r}} = \mathbf{F}(\mathbf{r}, t) \quad (2.1)$$

- $\mathbf{r}(\mathbf{a}, t)$  représentant la position du point  $\mathbf{a}$  de l'objet à l'instant  $t$ .
- $\mu(\mathbf{a})$  est la masse volumique de l'objet au point  $\mathbf{a}$ .
- $\gamma(\mathbf{a})$  est la densité de force de frottement au point  $\mathbf{a}$ .
- $\mathbf{F}(\mathbf{r}, t)$  représente la somme des forces extérieures.
- $\varepsilon(\mathbf{r})$  est une fonction qui représente l'énergie potentielle de déformation.

Il apparaît donc que, dans l'équation 2.1 :

- $\frac{\partial}{\partial t}(\mu \frac{\partial \mathbf{r}}{\partial t})$  traduit l'inertie de l'objet.
- $\gamma \frac{\partial \mathbf{r}}{\partial t}$  traduit les forces de frottement, dissipant de l'énergie.
- $\frac{\partial \varepsilon(\mathbf{r})}{\partial \mathbf{r}}$  représente la réponse élastique du matériau qui cherche à minimiser son énergie potentielle.

On voit que l'équation du mouvement met en œuvre une énergie potentielle de déformation  $\varepsilon$ , qui tend à redonner sa forme initiale à l'objet, à l'instar du ressort qui tend à reprendre sa longueur au repos. Pour quantifier les déformations de l'objet, les auteurs utilisent le tenseur métrique  $G$  associé à l'objet, c'est-à-dire :

$$G_{ij}(\mathbf{r}(\mathbf{a}, t)) = \frac{\partial \mathbf{r}}{\partial a_i} \cdot \frac{\partial \mathbf{r}}{\partial a_j}, \quad i, j \in \{1, 2, 3\}.$$

En effet, si  $G^0$  est le tenseur du matériau dans sa forme naturelle,  $\|G - G^0\|$  augmente avec les déformations. On peut alors écrire l'énergie potentielle de déformation  $\varepsilon(\mathbf{r})$  sous la forme suivante:

$$\varepsilon(\mathbf{r}) = \int_{\Omega} \|G - G^0\| da_1 da_2 da_3$$

Cette formulation entraîne bien une énergie nulle pour un simple déplacement rigide de l'objet, et provoque une force tendant à restaurer la forme initiale. À partir de cette équation, plusieurs types de comportements élastiques peuvent être obtenus en variant simplement la norme calculée. Mais comme le font remarquer les auteurs,  $\varepsilon$  ne peut s'exprimer que sous la forme d'une intégrale d'une fonction non linéaire.

### Algorithme d'animation

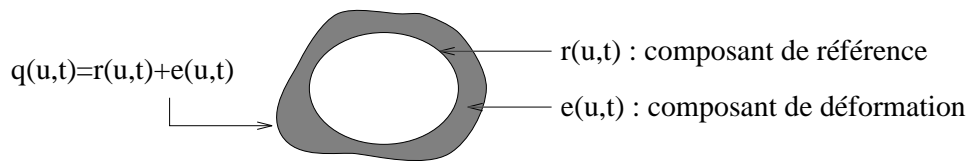
Simuler un objet physique avec ce modèle consiste donc à résoudre l'équation différentielle 2.1, soit à l'aide des différences finies comme vu précédemment puis d'une discrétisation temporelle, soit par éléments finis, ramenant dans les deux cas le problème à la résolution d'un très gros système matriciel (de l'ordre du nombre de nœuds de la discrétisation) du type  $A(t)x(t) = b(t)$ . *Animer un objet élastique revient donc à inverser à chaque instant la matrice  $A(t)$ .*

Des simulations sur différents types de matériaux (tissus élastiques, caoutchouc) ont montré les capacités et l'étendue d'un tel modèle. On peut cependant regretter que les comportements élastiques soient définis par un critère aussi peu intuitif qu'une norme de tenseur. De plus, aucune interactivité n'est possible, vu la lourdeur des calculs et le mauvais conditionnement des équations pour des objets assez rigides : c'est d'ailleurs ce qui a amené les auteurs à modifier le modèle initial.

### b) Méthode optimisée

C'est un an plus tard que Demetri Terzopoulos et Andrew Witkin proposent une nouvelle formulation du modèle [TW88]. Cette fois, l'objet est la somme de deux composants :  $\mathbf{r}(u)$  et  $\mathbf{e}(u, t)$ , exprimés dans le repère barycentrique de l'objet.  $\mathbf{r}(u)$  est dit "composant de référence" car il est indéformable et représente donc la forme au repos de l'objet ainsi que sa position au cours du temps. Par contre,  $\mathbf{e}(u, t)$ , immobile par rapport à  $\mathbf{r}(u)$ , est le "composant de déformation" qui représente la déformation actuelle de l'objet par rapport à sa forme initiale.

Une telle formulation permet, par rapport à l'ancien modèle, de définir plus simplement l'énergie potentielle de déformation. Plus précisément, les auteurs proposent d'écrire  $\varepsilon$  comme l'intégrale d'une combinaison linéaire des dérivées partielles successives de  $\mathbf{e}(u, t)$ , ce qui est maintenant possible puisque le cas d'un déplacement rigide n'est plus à envisager : la forme

FIG. 2.1 - *Modèle de [TW88]*

au repos est stockée par le composant de référence. Autre avantage : le fait de considérer un composant rigide permet une simplification des équations du mouvement, puisque le mouvement est maintenant régi par les lois de la dynamique du solide, et le composant déformable n'est plus utilisé que pour le calcul de la forme courante.

### Algorithme d'animation

Les deux apports de ce modèle ont pour conséquence directe que la simulation se ramène, comme auparavant, à la résolution d'un gros système linéaire, mais pour lequel la matrice  $A$  est indépendante du temps. *Animer des objets par cette méthode revient donc à calculer à chaque instant le produit  $x(t) = A^{-1}.b(t)$ , ce qui est beaucoup plus efficace.* Bien que mal adaptée à l'animation des tissus (absence d'un composant de référence), cette méthode permet par contre des simulations d'objets volumiques élastiques avec des temps de calcul raisonnables. Elle a en outre inspiré une généralisation pour le traitement d'objets inélastiques.

### c) Extension à des objets inélastiques

En effet, Terzopoulos et Fleischer publient un peu plus tard un article qui généralise la méthode hybride précédente à la simulation de différents types de déformations inélastiques [TPF89]. L'idée globale consiste à rendre le composant de référence non rigide, et à lui faire absorber peu à peu certaines déformations. Ainsi, l'objet est paramétré par  $\mathbf{q}(\mathbf{a}, t) = \mathbf{r}(\mathbf{a}, t) + \mathbf{e}(\mathbf{a}, t)$ , et l'on peut rendre l'objet :

- **visco-élastique** en utilisant une équation d'absorption du type :

$$\frac{d\mathbf{r}}{dt}(\mathbf{a}, t) = \frac{1}{\eta(\mathbf{a})} \cdot \mathbf{e}(\mathbf{a}, t)$$

où  $\eta(\mathbf{a})$  est la valeur de la visco-élasticité en  $\mathbf{a}$ . On obtient alors un objet avec une sorte de noyau visqueux entouré d'une couche élastique.

- **semi-plastique** en faisant absorber par le composant de référence uniquement les trop grandes déformations.
- **capable de fractures**. Les auteurs, pour ce cas, ne détaillent pas les problèmes d'implémentation qui paraissent pourtant délicats ; ils précisent simplement que les points de fractures doivent être *pré-déterminés* (par exemple, les intersections des mailles d'un tissu). Les fractures auront lieu en ces points en cas de trop fortes déformations.

### d) Modèles utilisant les éléments finis

La méthode dite des *éléments finis* est la plus courante dans le domaine de l'ingénierie pour toute simulation de déformation. L'objet simulé est discrétisé en mailles élémentaires formées d'un ensemble de noeuds. Après avoir choisi une base de fonctions d'interpolation, on peut se ramener à une équation classique de type :

$$M\ddot{\mathbf{X}} + C\dot{\mathbf{X}} + K\mathbf{X} = \mathbf{F}$$

permettant de trouver les déplacements des noeuds en fonction des forces en jeu. Ces principes ont été utilisés par exemple par Demetri Terzopoulos et Dimitri Metaxas [TM91].

Un autre modèle propose cependant une méthode un peu particulière : la particularité de leur approche est qu'elle n'est pas dynamique. Pour modéliser une main saisissant une balle déformable, Jean-Paul Gourret, Daniel Thalmann et Nadia Magnenat-Thalmann [GMTT89, GTT89] utilisent les éléments finis pour résoudre les équations mécaniques trouvées, en appliquant le principe des travaux virtuels. Et c'est l'utilisateur qui décide du mouvement de la main, les éléments finis servant uniquement à calculer les déformations créées. L'animation est en fait une succession d'états d'équilibre entre la main et la balle.

### e) Discussion

Toutes ces méthodes reposent sur un modèle physique dont on déduit les équations physiques du comportement en écrivant la conservation de l'énergie du système (par le biais du Principe Fondamental de la Dynamique, des équations de Lagrange, ou bien encore du Principe des Travaux Virtuels). Enfin, elles intègrent le mouvement des noeuds de discrétisation : des déformations locales apparaissent, et se propagent dans la matière des objets. Cela demande à chaque pas d'animation des calculs matriciels plus ou moins coûteux, qui semblent interdire toute interactivité. De plus, il n'existe pas d'approche *unifiée* permettant divers comportements. Chaque comportement (élastique, inélastique, plastique) a ses propres équations et souvent sa propre implémentation. Il est donc clairement impossible de fournir un logiciel d'animation à un graphiste, puisque l'on ne dispose pas non plus de paramètres intuitifs pour modifier le comportement d'un matériau. C'est en ce sens que ces modèles sont peu maniables, et que des modèles plus intuitifs et plus simples à mettre en œuvre sont souvent recherchés.

#### 2.1.2 Approches discrètes

Au lieu de partir d'une description continue de l'espace et du temps, puis de la discrétiser pour résoudre des équations différentielles du mouvement, une approche lancée par Greenspan en 73 propose de décrire des modèles directement discrets, enlevant ainsi une étape intermédiaire de modélisation. Il en découle par exemple qu'il est raisonnablement possible de considérer directement les objets déformables comme des structures construites à partir d'éléments mécaniques discrets. Typiquement, la matière structurée peut se concevoir comme une composition de masses ponctuelles reliées entre elles par des lois d'interaction simples type ressort ou amortissement. Cela n'est d'ailleurs conceptuellement pas très éloigné d'un objet discrétisé en maillage, aperçu au paragraphe précédent. Mais ce mode de résolution et les multiples choix d'interactions possibles se révèlent être un modèle beaucoup plus *maniable*.

En effet, si l'on connaît la disposition des masses et leurs lois d'interaction (figure 2.2), il est très facile de faire, *indépendamment sur chaque masse*, un bilan des forces exercées par les autres



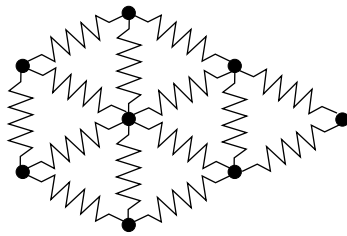


FIG. 2.2 - *Un exemple simple de réseau Masses-Ressorts*

masses, puis d'intégrer le mouvement par des méthodes d'intégration numérique. Les équations à résoudre pour chaque masse sont alors de simples équations différentielles du premier ordre, à savoir les relations fondamentales de la Dynamique :

$$\frac{d\vec{v}}{dt} = \vec{a} = \frac{\vec{F}_{total}}{m}$$

$$\frac{d\vec{x}}{dt} = \vec{v}$$

Une méthode d'intégration simple, comme le schéma de Newton-Cotes, résout ces équations de la manière suivante :

$$\vec{v}_t = \vec{v}_{t-dt} + \vec{a}_t dt$$

$$\vec{x}_t = \vec{x}_{t-dt} + \frac{1}{2}(\vec{v}_t + \vec{v}_{t-dt})dt.$$

Pour éviter les divergences qui peuvent survenir pour des pas  $dt$  d'intégration trop grands, d'autres moyens, comme la méthode de Runge-Kutta d'ordre 4 ou les méthodes implicites, ont été développés et permettent de bien meilleurs résultats, mais avec un coût de calcul plus important : voir l'annexe C pour un peu plus de détails.

### a) Exemples de modèles en réseaux fixes

Lorsque l'objet à simuler ne change pas de topologie au cours du temps, il est intéressant de le construire à l'aide de masses élémentaires agencées en réseau de topologie fixe. Cela permet de limiter le nombre d'interactions à calculer, et donc *d'accélérer le bilan des forces* sur chaque masse.

De tels réseaux ont été d'abord utilisés par Annie Luciani [Luc85] pour l'animation de marionnettes, ou encore par Chadwick pour animer la chair déformable de personnages [CHP89]. Dans le cadre d'une simulation de reptiles, Gavin Miller en 88 [Mil88] propose aussi d'utiliser un réseau fixe de masses et de ressorts comme constituant du corps des serpents.

Ces réseaux, faciles d'emploi, permettent, par les interactions entre les masses, une approximative conservation du volume du corps du serpent. En les joignant bout à bout (voir figure 2.3), on permet une propagation des déformations de ressort en ressort. En y rajoutant des stimulations sinusoïdales, on simule ainsi le déplacement caractéristique des reptiles.

On peut aussi citer des liaisons inter-masses plus générales, comme des ressorts angulaires contrôlant l'angle entre un triplet de masses, ou des ressorts de torsion [Jou97]. Un formalisme général appelé CORDIS-ANIMA, proposé par Claude Cadoz et Annie Luciani [LC86], permet de mettre en œuvre quantité de simulations différentes de type masses-ressorts : les liaisons peuvent être alors définies algorithmiquement, enrichissant grandement le potentiel de ces modèles discrets.

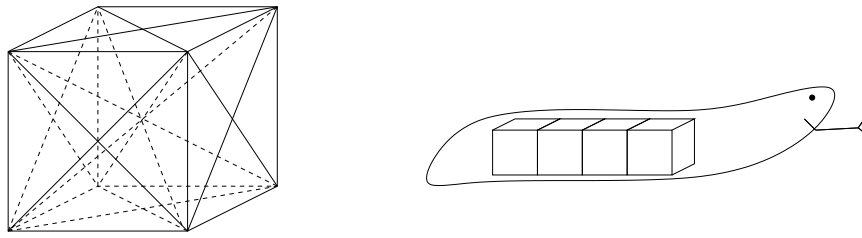


FIG. 2.3 - L'élément hexaédrique à la base du modèle de serpent.

### b) Stabilité de l'intégration

Comme nous l'avons évoqué plus haut, des techniques d'intégration évoluées comme les *méthodes implicites*, inconditionnellement stables, permettent de pallier en partie le problème de la stabilité de l'intégration des systèmes discrets au cours du temps. Mais ces solutions sont très coûteuses à l'emploi : c'est pourquoi des modèles plus efficaces, et souvent plus ad hoc, ont été développés. Par exemple, Xavier Provot [Pro95] propose de tout simplement seuiller la distance entre deux masses de façon à éviter toute force trop importante dans le système. Dans le même ordre d'idée, Ammar Joukhaddar [Jou97] divise le pas de temps lorsque l'énergie mécanique du système change brusquement, limitant ainsi les possibilités de divergence.

### c) Visualisation

Le fait de disposer d'un voisinage fixe entre masses permet une représentation simple par facettes : un maillage s'appuyant sur les masses et suivant les ressorts offre une visualisation de l'objet simulé. Cela permet une visualisation temps réel, bien utile dans les cas de simulateurs chirurgicaux par exemple [Mes97]. Cependant, un certain nombre d'applications en synthèse d'images nécessite une représentation beaucoup plus fine. Miller, par exemple, utilise une surface spline dont les masses sont les points de contrôle : l'aspect est donc lisse et les masses invisibles [Mil88]. Cette façon de procéder était d'ailleurs déjà utilisée en 2D par Annie Luciani pour l'animation de pantins [Luc85].

### d) Discussion

Les réseaux masses-ressorts sont très proches des modèles continus vus précédemment. On peut d'ailleurs montrer qu'un maillage régulier de masses-ressorts tend, pour un nombre infini de ressorts de longueurs infiniment petites et de masses infinitésimales, vers le comportement d'une membrane. Même algorithmiquement, *les calculs effectués dans un cas comme dans l'autre se ramènent finalement à des interactions simples entre noeuds de discrétisation, quelque soit le modèle de départ choisi*. La principale différence entre les deux approches vues jusqu'ici est donc plutôt d'ordre pratique : implémenter un modèle discret est simplement plus aisé et se prête très bien à la parallélisation.

De tels réseaux ne sont cependant pas si simples à définir pour un animateur. Quelle géométrie de réseau choisir en fonction des déformations voulues ? Combien de masses suffisent à obtenir un mouvement souhaité ? Quel doit être le pas de temps de simulation assurant à la fois stabilité et rapidité ? De plus, il est évident que des réseaux fixes ne peuvent engendrer des objets se cassant ou changeant simplement de topologie, ce qui limite les possibilités du modèle.

### 2.1.3 Approches globales

Dans toutes les approches vues jusqu'ici, le temps de calcul est souvent un obstacle pour l'animation en temps réel. Pourtant se sont développées, en marge des autres approches, des méthodes qui pallient ce problème. Elles évitent en effet les lentes propagations des déformations au travers de nœuds internes en calculant directement des déformations globales, ce qui évite les instabilités numériques. Malheureusement, aucun de ces modèles n'a abouti jusqu'à présent à une extension pour les objets hautement déformables. Essayons d'en comprendre la raison.

#### a) Dynamique modale

Alex Pentland et John Williams proposent en 89 le modèle d'objets déformables utilisant la dynamique modale [PW89]. Pour ce faire, ils partent de l'équation de l'élasticité déjà citée, mais pour un objet déformable discrétisé en éléments finis, à savoir :

$$M\ddot{\mathbf{u}} + D\dot{\mathbf{u}} + K\mathbf{u} = \mathbf{f} \quad (2.2)$$

où  $M$ ,  $D$  et  $K$  sont des matrices représentant respectivement la masse, l'amortissement et la raideur du matériau, et  $u$  est le vecteur déplacement des  $n$  nœuds de discrétisation, de taille  $3n$ . En remarquant que dans les cas courants,  $M, D$  et  $K$  sont toutes symétriques définies positives, et que l'on utilise fréquemment  $D = sM$  (où  $s$  est un scalaire), on bénéficie du fait que ces 3 matrices admettent une base commune de diagonalisation. L'intérêt immédiat consiste à transformer l'équation 2.2 en la multipliant par la transposée de la matrice de passage  $P$ , ce qui donne avec  $u = P\tilde{u}$  :

$$\tilde{M}\ddot{\tilde{\mathbf{u}}} + \tilde{D}\dot{\tilde{\mathbf{u}}} + \tilde{K}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}.$$

Puisque  $\tilde{M}$ ,  $\tilde{D}$  et  $\tilde{K}$  sont maintenant diagonales, on dispose donc de  $3n$  équations différentielles *indépendantes*. Elles représentent l'évolution des *modes vibratoires* de l'objet, d'où le nom d'analyse modale. La  $i^{\text{ème}}$  colonne de  $P$  (notée  $P_i$ ) représente justement le  $i^{\text{ème}}$  mode vibratoire, c'est-à-dire la déformation de l'objet en réponse à la force  $f_i$ . Les déformations peuvent donc s'écrire :

$$\mathbf{u} = \sum_i \tilde{\mathbf{u}}_i P_i.$$

Cependant, on sait que les modes de hautes fréquences ne changent que peu l'apparence de l'objet. Ainsi, une première approximation proposée par Pentland et Williams est de ne tenir compte que des trois premiers modes, ce qui signifie la prise en considération des déformations linéaires et quadratiques uniquement. Par ce fait, le pas d'intégration peut donc être augmenté puisque seules les basses fréquences sont retenues. Une seconde optimisation consiste à remarquer que les modes de vibration à basse fréquence d'un objet dépendent plus de ses dimensions que de sa forme précise. On pré-calculé donc des modes vibratoires pour un solide rectangulaire, et on les interpole selon les dimensions de la boîte englobante de l'objet.

Enfin, une dernière approximation est faite en associant à chaque mode une fonction polynômiale de déformation (voir [Bar84]) qui l'approxime, et qui pourra être appliquée à la surface de l'objet au repos pour obtenir l'objet déformé. Cela permet un gain de vitesse important à l'affichage et une gestion des interactions facilitée puisque l'on dispose d'une équation précise de la surface.

Cette méthode dispose comme incontestable avantage d'un coût de calcul très faible, permettant une visualisation temps réel. En revanche, les diverses approximations utilisées font que le modèle n'est plus une réelle simulation dynamique. Par contre, on remarquera que de tels procédés sont très pratiques pour des graphistes de par l'interactivité et du fait aussi que l'on peut jouer sur les paramètres pour obtenir les effets classiques d'animation, comme l'exagération des mouvements par exemple.

### b) Utilisation de matrices de déformations

Un an après, Andrew Witkin et William Welch [WW90] proposent une approche un peu similaire, mais sans utiliser la dynamique modale. Ils éliminent en effet les modes de hautes fréquences en restreignant les déformations d'un objet à une certaine classe de transformations globales matricielles. L'objet est défini au départ de façon paramétrique, puis subit des déformations par "placage" (composition avec des matrices de déformation), selon la méthode de Barr [Bar84]. La simulation est alors calculée en déduisant de ces déformations les énergies potentielles ( $V$ ) et cinétiques ( $L$ ) résultantes, et en intégrant l'équation Lagrangienne obtenue :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} - \mathbf{Q} = 0$$

où  $\mathbf{q}$  sont les coordonnées Lagrangiennes et  $\mathbf{Q}$  les forces extérieures exprimées dans ces coordonnées.

En 1992, Dimitri Metaxas et Demetri Terzopoulos [MT92] proposent de mélanger à la fois l'approche par éléments finis [TF88a] qui permet des déformations locales et une méthode type Witkin et Welch utilisant des déformations globales. Pour résumer, ils appliquent des déformations globales à leur composant de référence, et utilisent la résolution classique par maillage sur le composant de déformation. Le modèle de départ est donc optimisé en temps de calcul.

### c) Inconvénients des méthodes globales

Toutes ces approches réalisent des économies en temps de calcul en restreignant le comportement d'un objet à certaines classes de déformation. On tronque les déformations locales pour ne garder que le comportement global. On peut donc d'ores et déjà reprocher le manque de réalisme sur de tels modèles. En limitant ainsi le nombre de déformations, on voit mal comment généraliser ce type d'approches à la modélisation d'objets hautement déformables capables de se couper en plusieurs morceaux : les matrices de déformation ne peuvent en aucun cas changer la topologie d'un objet.

#### 2.1.4 Approches stratifiées

Partant des modèles existants en animation d'objets rigides et déformables, une famille de méthodes utilise une approche *modulaire* pour construire leur modèle global. L'idée directrice est de *combiner plusieurs couches constituées de modèles existants* afin d'obtenir un modèle complet plus riche, et plus intuitif à manipuler.

### a) Premier modèle de Burtnik

Dès 76, Nicholas Burtnik [BW76] propose une approche originale pour améliorer le contrôle de l'utilisateur dans le cadre de l'animation de personnages : au lieu de faire bouger les muscles

d'un personnage et d'engendrer alors le mouvement du squelette, il préfère directement animer le squelette. Puis, il rajoute les mouvements des muscles et de la peau en conséquence. L'animateur y gagne bien évidemment en facilité, et le modèle des muscles peut, de plus, être très simplifié.

### b) Améliorations de Chadwick et Parent

John Chadwick [CHP89] reprend cette idée en proposant un modèle plus complet. Il découpe son modèle de personnage en trois couches imbriquées :

- Un squelette, qui sera animé par l'utilisateur à l'aide de positions clés par exemple.
- De la chair (joues, muscles, ...), constituée de muscles contrôlés par des boîtes de FFD [SP86] qui définissent leur déformation courante. Ces boîtes peuvent être contrôlées par des masses reliées entre elles par des ressorts, et raccordées au squelette. Ainsi, à chaque mouvement du squelette, la chair s'ébranle et se déforme en conséquence.
- Une peau, purement géométrique, qui recouvre la chair.

On voit bien ici l'intérêt des trois couches : le squelette permet à l'utilisateur de facilement faire évoluer son personnage, la chair ajoute un degré de réalisme au squelette (les paramètres des ressorts et de l'amortissement sont réglables aussi par l'animateur), et une peau permet une représentation continue de l'ensemble.

### c) Généralisation de Gascuel, Verroust et Puech

Marie-Paule Gascuel, Anne Verroust et Claude Puech généralisent encore cette approche en 1990 [GVP90, Gas90, Ver90]. Les trois niveaux sont conservés, mais chacun est plus élaboré. Le squelette, par exemple, est dynamique et constitué de solides articulés. Ce qui implique que l'objet génère entièrement son mouvement : chutes, rebonds, sont entièrement gérés par le modèle lui-même. La chair est, dans ce modèle, constituée de ressorts répartis en étoile tout autour du centre de masse de chaque squelette : cela définit ainsi une chair déformable capable de détecter et répondre aux collisions. De plus, cela ajoute du réalisme par exemple dans les cas de flexion : deux squelettes ne pourront se plier trop car leurs chairs déformables respectives se collisionneront et créeront une force de rappel. Enfin, une peau, toujours purement géométrique, passe par les sommets des ressorts pour habiller le tout.

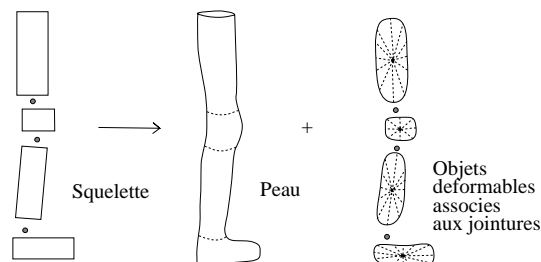


FIG. 2.4 - Construction de l'enrobage d'un squelette, d'après [GVP91]

### d) Modèle de Turner

Plus récemment, Turner propose à son tour un modèle plus spécifique pour l'animation de personnages [Tur95]. Cette fois, les couches internes (squelettes et chair) sont modélisées de façon purement géométrique et cinématique, laissant ainsi du temps de calcul pour simuler la peau, seul élément visible. Le squelette est toujours une structure articulée, que l'on peut déplacer par cinématique inverse, en spécifiant des chaînes cinématiques. Les muscles, modélisés par des objets implicites déformables simples de type super-quadriques<sup>1</sup>, sont attachés aux jointures du squelette. Ces muscles servent de contraintes de réaction, puisque l'on souhaite que la peau reste à l'extérieur de la chair. La peau est enfin ajoutée, fixée aux muscles par quelques points d'attaches pour éviter qu'elle ne glisse trop sur la chair. Cette peau est finalement simulée selon la méthode de Terzopoulos [TPBF87] pour les surfaces déformables.

Un effort est fait dans la modélisation des muscles pour que le personnage créé ait vraiment forme humaine. L'ajout de textures sur la peau augmente encore le réalisme.

### e) Modèle de Shen

Dans le même temps, Jianhua Shen et Daniel Thalmann présentent un modèle de personnages [ST95]. Là encore, un effort tout particulier est fait sur la modélisation des muscles, faite de primitives implicites de type metaballs. Par contre, la peau est purement géométrique. Shen profite de la simplicité topologique du corps humain pour habiller les muscles d'une B-spline de façon bien particulière : des points de contrôle sont définies sur une série de sections successives le long des membres. En recollant les différents patches obtenus pour chaque membre, il dispose finalement d'un maillage globale rapide à mettre à jour, permettant un enrobage particulièrement esthétique.

### f) Discussion

Ces approches, profitant des avantages de chacun des modèles qui le compose, permettent d'augmenter la complexité des objets simulés. Ils offrent aussi un temps de calcul réduit, puisque partagé entre différentes couches simples et appropriées. Ce découpage convient bien à l'utilisateur, qui y trouve des paramètres bien différenciés entre chaque couche et donc très intuitifs. Et les temps de calculs sont optimisés du fait de la simplification que peut apporter l'utilisation de couches, comme vu lors du modèle de Terzopoulos [TF88a] au paragraphe 2.1.1.

Pourtant, tous ces modèles fort séduisants exploitent le fait que les objets simulés sont *structurés* : cette structure fixe permet de déduire une modélisation composite optimisée. Ils ne sont donc aucunement capables de changements de topologie comme nous le souhaiterions. Une dernière remarque est que bien souvent, il manque de rétroaction entre chaque couche : une couche supplémentaire s'appuie sur la couche précédente, mais rares sont les modèles où l'information circule aussi dans l'autre sens : seul [GVP91] offre un retour de l'extérieur vers l'intérieur par le biais des ressorts, qui transmettent ainsi les forces extérieures aux squelettes.

## 2.2 Objets déformables non structurés

Le passage aux objets non structurés amène des problèmes bien particuliers. Le fait qu'un objet puisse se séparer en morceaux et/ou se réorganiser différemment à la façon d'une pâte à

---

1. Les super-quadriques seront vues plus en détail au chapitre 3.

modéliser entraîne des difficultés de gestion de topologie et de relation de voisinage entre points de discrétisation. Les techniques développées pour les objets structurés sont difficilement généralisables dans les cas extrêmes de déformation que sont la séparation ou la fusion par exemple. Ceci explique sûrement le peu de modèles existants en Synthèse d'Images capables de traiter de tels objets.

### 2.2.1 Approches à base de particules

Outre l'animation de réseaux masses-ressorts, le système CORDIS-ANIMA permet de modéliser de la matière non structurée [Luc89]. Pour cela, les masses élémentaires sont reliées par des lois d'interaction *pouvant évoluer au cours du temps*. L'évolution temporelle permet une plus grande gamme de comportements, puisque cette fois des liaisons de type ressort peuvent disparaître puis ré-apparaître. À la limite, toutes les masses du réseau peuvent être interconnectées : il s'agit alors d'un cas particulier, baptisé "modèle particulaire", que nous allons développer. Ce genre d'interaction s'inscrit dans le cadre plus vaste de la simulation du comportement : Reynolds [Rey87], par exemple, modélise le comportement de bancs de poissons ou de vols d'oiseaux en considérant que chaque membre d'un groupe réagit à son entourage et son environnement selon des lois simples. Et de cette multitude de comportements individuels simples naît un comportement global complexe, dit *émergent*.

Il est à remarquer que dans les premières utilisations de systèmes de particules, la modélisation n'utilisait pas forcément des lois physiques. En 1983, William Reeves [Ree83] utilisa des systèmes de particules, pour simuler des phénomènes comme le feu, la fumée, les feux d'artifice ainsi que l'herbe. Il proposa une modélisation très simple de particules contrôlées par des processus stochastiques pour gérer leur nombre, leurs vitesses ou encore leurs couleurs. Cette approche est intéressante, car elle compense le grand nombre de particules par des lois très simples, permettant des coûts de calcul tout à fait convenables. De plus, l'intérêt de tels systèmes est leur implémentation aisée sur des machines parallèles.

#### a) Premiers modèles

Cette idée de modèle particulaire est mise en application par Demetri Terzopoulos, dont l'approche de l'élasticité utilisait jusqu'alors des modèles continus régis par diverses équations différentielles. Il propose en 1989 un modèle hybride qui simule des objets déformables capables de fondre par des masses agencées en réseau, reliées par des ressorts thermo-conducteurs (dont la raideur diminue à haute température). À très haute température, ces ressorts "fondent". Une fois les masses libérées, elles sont soumises entre elles à des forces d'attraction-répulsion inspirées de la physique moléculaire [TPF89].

La même année, Gavin Miller et Andrew Pearce [MP89] exposent comment modéliser des objets allant de la poudre aux fluides visqueux à l'aide de forces inter-particulaires dites de Lennard-Jones. Ces forces, issues de mesures physiques, dérivent du potentiel  $\phi$  suivant:

$$\phi(r) = \frac{A}{r^{12}} - \frac{B}{r^6}$$

où  $A$  et  $B$  sont des constantes et  $r$  la distance entre deux particules. Elles ont depuis été généralisées comme nous le verrons par la suite. La figure 2.5 montre la forme caractéristique de ces forces d'attraction-répulsion pour une paire de particules, où l'on remarque une zone de répulsion pour des distances très faibles, une distance  $r_0$  d'équilibre stable, et une zone d'attraction à moyenne distance, pour enfin avoir une interaction quasi nulle à longue distance.

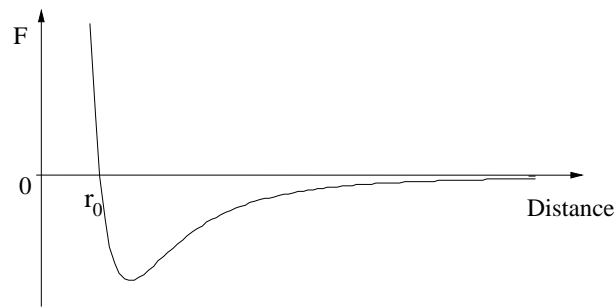


FIG. 2.5 - Un potentiel de Lennard-Jones, et la force dérivée.

En 1991, David Tonnesen [Ton91] poursuit le travail de Terzopoulos, et propose une loi d'interaction entre particules dépendant de la température. Ainsi, le modèle est identique que l'objet soit à l'état solide ou liquide. C'est uniquement la température qui modifie son comportement physique. Il s'agit maintenant d'un matériau constitué d'atomes, capable de se briser, de rebondir ou de fondre, uniquement selon la température de son environnement. Les particules offrent donc une approche unifiée pour toute une série de comportements.

Dans le même ordre d'idée, Annie Luciani et Stéphane Jimenez [LJF<sup>+</sup>91, LJR<sup>+</sup>91, Jim93] proposent de modéliser des objets déformables par des agglomérats de particules de matière — les billes (masses ponctuelles munies d'une sphère de non-pénétration), constitués en 3 couches:

- un ou plusieurs noyaux représentent la partie centrale de l'objet, souvent indéformable.
- des petites billes, attirées par les noyaux mais se repoussant entre elles, forment le derme ou la chair de l'objet.
- enfin, de minuscules billes, attirées par le derme et s'attirant entre elles, forment l'épiderme, simulant une tension de surface (voir figure 2.6).

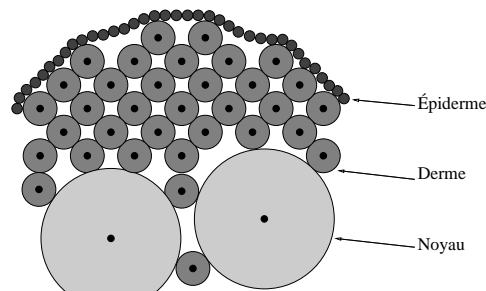


FIG. 2.6 - Le modèle de Jimenez (d'après [Jim93])

De bonnes modélisations des interactions entre billes permettent alors de simuler des comportements divers, et ceci de manière simple puisque, encore une fois, le calcul du mouvement d'une masse ponctuelle ne réclame aucun calcul complexe. Par exemple, des simulations de type tas de sable peuvent être effectuées avec succès par le biais de forces d'interaction très simples [LHVD95].



## b) Extensions récentes

Enfin, plus récemment, David Tonnesen et Richard Szeliski [ST92] ont présenté, dans le cadre de la modélisation de surfaces déformables, une nouvelle forme de particules : les particules orientées. Cette fois, les particules ne sont plus de simples points matériels, mais deviennent des sortes de mini-facettes orientées : cela laisse donc six degrés de liberté à la particule. Les auteurs ajoutent alors aux forces inter-particulaires classiques des forces de trois types : une force tendant à aligner les facettes, une autre visant à aligner les normales aux facettes, et une dernière ayant pour but de mettre les particules sur un cercle. Comme on peut le constater, cette approche est parfaite pour modéliser la surface d'un objet et toute déformation qui pourrait s'y produire, mais ne permet malheureusement aucun contrôle de la variation du volume de l'objet au cours du temps.

Ce modèle a été depuis enrichi par Jean-Christophe Lombardo et Claude Puech [LP95]. Il permet maintenant d'animer des objets à *mémoire de forme*, puisque les particules orientées tentent de garder leurs relations angulaires entre voisines. Une reconstruction d'objets de classe  $\mathcal{C}^1$  a même été développée, afin d'automatiquement initialiser les particules orientées en position stable sur une forme donnée.

Diverses extensions des forces d'interaction ont été aussi proposées. Lombardo et Puech rajoutent par exemple un palier à la fonction de Lennard-Jones autour de la position d'équilibre pour diminuer les oscillations lors d'une intégration par schéma d'Euler. Hugh Reynolds [Rey97], quant à lui, utilise une fonction s'annulant en des points régulièrement espacés au lieu de prendre une force d'interaction avec un seul point d'équilibre. Cela permet de simuler des phénomènes comme la fracture par seuils de rupture, les particules s'écartant en sautant de point d'équilibre en point d'équilibre.

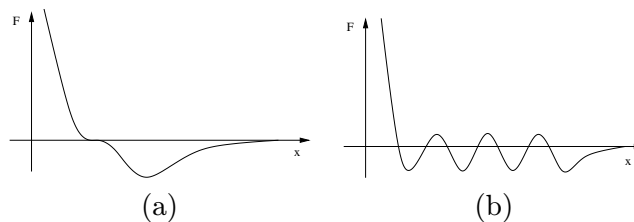


FIG. 2.7 - Les forces d'interaction en fonction de la distance pour le modèle de Lombardo (a), et de Reynolds (b)

## c) Importance du mode de résolution

Dans [MP89], Miller et Pearce détaillent les problèmes d'intégration qui se posent particulièrement quand la matière est rigide (la raideur, et donc les forces en jeu étant beaucoup plus importantes), et soulignent l'importance de la méthode d'intégration pour y remédier.

Mais, comme pour les réseaux masses-ressorts, et même si l'on manipule des lois du mouvement très simples, il apparaît que tous les modèles à système de particules décrits jusqu'ici peuvent devenir instables si le pas d'intégration n'est pas assez petit. C'est pourquoi toutes les mises en œuvre jouent sur la *force de frottement*, opposée à la vitesse, qui freine les particules, permettant une stabilisation plus rapide du système. C'est en effet la seule force *dissipative* du

système qui évite que le système oscille sans fin. Cette solution freine cependant la chute de l'objet simulé sous l'effet de la gravité, ce qui peut nuire au réalisme des simulations.

En 1992, Donald H. House et David Breen présentent un nouvel algorithme de simulation de comportement d'un système de particules [HBG92]. Ils proposent deux phases bien distinctes pour chaque pas de simulation. Dans un premier temps, ils calculent le mouvement des particules provoqué uniquement par les forces extérieures aux particules (poids, et réaction en cas de choc). Puis, ils ajustent les positions des particules de façon à satisfaire les contraintes inter-particulaires, définies par des fonctions de potentiel : c'est une phase de minimisation d'énergie. Pour ce faire, une descente de gradient stochastique (SGD) est utilisée, qui déplace les positions des particules de façon à diminuer l'énergie potentielle totale du système. On peut schématiser ce procédé par une phase de mouvements à grande échelle, suivie d'une phase d'affinage à l'échelle de la structure du système. Si, cette fois, le pas de simulation a moins d'importance, c'est le temps que demande la SGD qui devient critique : les auteurs annoncent 3 jours de calculs sur une station classique pour un ensemble de 51x51 particules !

#### d) Importance du mode d'évaluation des forces

Une fois le problème de résolution réglé, le dernier point à aborder est le mode d'évaluation des forces. C'est en effet le goulot d'étranglement de tout système de particules. L'évaluation de toutes les forces binaires consomme un temps très important si l'on n'y prend garde.

Or ce problème est fortement relié au problème physique classique dit des  $\mathcal{N}$  corps. Il est dès lors intéressant de répertorier les méthodes existantes dans ce domaine pour découvrir les moyens d'évaluation rapides possibles.

##### Particule-Particule

La méthode Particule-Particule (PP) est la façon la plus simple d'évaluer les forces binaires inter-particulaires, mais c'est malheureusement la plus coûteuse. Il s'agit, pour chaque particule, de calculer les forces dues aux autres particules en les passant en revue une à une. Autant la mise en œuvre est très rapide, autant le coût de calcul est élevé, puisqu'en  $O(N^2)$ , où  $N$  est le nombre de particules.

##### Particule-Grille

Bien souvent, les forces physiques inter-particulaires dérivent d'un potentiel. Une autre façon d'évaluer toutes les forces est alors de définir une grille régulière dans l'espace englobant toutes les particules. Puis, à chaque nœud, on calcule la densité de masse selon le nombre de particules à proximité. De ce champ discret de densité, on en déduit le champ potentiel résultant par l'équation de Poisson à l'aide d'une transformée de Fourier. De ces valeurs de potentiel, les forces aux nœuds de la grille sont approximées par différences finies. Enfin, une dernière phase consiste alors à calculer le bilan final des forces sur chaque masse en interpolant les forces stockées sur les nœuds proches. Le coût global est finalement en  $O(N_g \log(N_g))$ , où  $N_g$  est le nombre de nœuds dans la grille. Mais, en contre-partie, les interactions sont évidemment approximées très grossièrement.

### Par partitionnement spatial : Barnes et Hut

Un autre moyen de procéder à l'évaluation des forces utilise un algorithme de type *diviser pour régner*. Après avoir trouvé la boîte englobante des particules, on construit un octree que l'on subdivise jusqu'à qu'il n'y ait plus qu'une particule par feuille. Il suffit alors pour chaque particule de parcourir cet octree depuis la racine : si le sous-octree en cours de traitement s'avère très éloigné, la force due aux groupes de particules de ce sous-octree est approximée en fonction du centre de masse et de la masse totale du groupe. Pour les autres sous-octrees, on parcourt l'arbre jusqu'à la racine et le calcul classique de la force binaire inter-particulaire a lieu.

En évitant des calculs fins pour des parties peu influantes, on améliore ainsi grandement le temps de calcul de l'évaluation totale : l'algorithme a maintenant un coût total de  $O(N \log(N))$  pour  $N$  particules.

### Méthode de développements multipôles

La méthode de développements multipôles (*Fast Multipole Method*) est une technique introduite par Greengard en 1990, qui encore une fois passe par l'utilisation d'un potentiel plutôt que par des vecteurs force. L'idée est cette fois de faire une sorte de développement de Taylor sur le potentiel, à la différence que ce développement, dit multipôle, est d'autant plus précis que la distance est grande. En utilisant le même octree que pour la méthode de Barnes et Hut, on arrive alors à un coût algorithmique théorique de  $O(N)$  avec une précision numérique bien plus importante. Cependant, les formules mises en jeu par cette technique mathématique sont assez lourdes : sa mise en œuvre semble peu rentable pour des modèles aussi simples que ceux de la Synthèse d'Images.

### Par partitionnement spatial simple

Par contre, une technique classique d'accélération du bilan des forces est courante en animation, même si elle ne peut s'appliquer que pour des particules de même taille et des forces d'interaction binaires à support fini fixe. Si on sait que les forces s'annulent à partir d'une distance  $d$ , il suffit de créer une grille régulière dont chaque case a une taille égale à  $d$ , et chaque particule est affectée à une des cases de cette grille selon sa position. Alors l'évaluation d'une particule se fera en évaluant les forces binaires dues aux particules des cases voisines uniquement. Le coût théorique moyen total devient  $O(N)$ .

Mais, et c'est une remarque valable pour toutes les méthodes sauf la PP, les coûts théoriques moyens peuvent très facilement être mis en défaut : dans la dernière méthode par exemple, si presque toutes les particules se retrouvent dans la même case, cette technique fera perdre plus de temps que la méthode classique PP.

## e) Visualisation des particules

La représentation des systèmes à particules paraît problématique : un ensemble de masses discrètes désordonnées est peu propice à la création d'une image de qualité. Et même si parfois, le comportement d'un objet est plus important que sa forme précise, une des contraintes de la synthèse d'images est l'obtention d'*images* réalistes.

### Enrobage purement géométrique

Pour créer une représentation surfacique à partir d'un système de particules, il faut recourir à un *enrobage*, ou *habillage* du système de particules. Ce thème a été abordé dans divers articles. Une solution simple est d'utiliser une représentation de chaque particule sous forme de sphère [LJF<sup>+</sup>91]. Une approche fournissant des surfaces lisses plus adéquates à la représentation visuelle de liquides visqueux, utilise des surfaces implicites [MP89, TPF89, Ton91] : par analogie aux particules chargées, on considère que les masses émettent un potentiel dans l'espace, décroissant avec la distance. Une surface lisse entourant le système est alors trouvée en considérant une isopotentielle du champ résultant (voir le chapitre 3 pour plus de détails sur les surfaces implicites).

### Écran d'épingles

Enfin, Arash Habibi a présenté récemment une méthode d'habillage non purement géométrique. Afin d'offrir une visualisation plus esthétique des systèmes de particules en 2D constituant le modèle interne, il propose l'utilisation d'un *écran d'épingles*, défini par une grille de masses discrètes (les épingles) qui sont reliées entre elle par des ressorts. Le passage d'une particule fait soulever les masses sur son chemin, et l'ensemble de l'écran se déforme en conséquence. La hauteur des masses permet de définir un champ discret, qui, représenté sous forme de niveaux de gris, offre une visualisation originale.

Cet habillage des particules confère une représentation enrichie. Une particule seule en mouvement créera une traînée plus ou moins longue derrière elle selon les paramètres de l'écran, enrichissant ainsi le mouvement créé. Contrairement aux méthodes disposant d'une surface purement géométrique, l'écran ajoute, par sa simulation physique de la "peau", de la complexité au modèle particulaire.

### f) Discussion

Malgré des problèmes d'algorithmique, d'intégration numérique, et de visualisation, les particules forment un modèle séduisant de par leur généralité et leur simplicité. On peut en effet modéliser tout changement de topologie ou fortes déformations par ce modèle unifié dont l'implémentation est très aisée.

On peut cependant formuler une remarque : l'algorithme de Barnes et Hut optimise les calculs d'évaluation de forces en centralisant ses efforts dans les régions où des calculs précis sont inévitables, et en approximant les autres régions d'influence négligeable. Mais le système de particules, quant à lui, n'applique pas cette stratégie pourtant séduisante. On peut remarquer en effet que simuler des particules au repos nécessite autant de calculs que des particules subissant de forts déplacements : dans les deux cas, on sera obligé de passer par des évaluations de forces et par des intégrations à chaque pas de temps.

## 2.2.2 Approches empruntées à l'Hydrodynamique

Un cas bien particulier d'objet hautement déformable est l'eau, et plus généralement les liquides. La simulation de l'eau, matière extrêmement déformable par excellence, est un domaine actif en Synthèse d'Images. Un bon nombre de modèles utilise une approche très pragmatique,

en ne simulant par exemple que la forme des vagues de la mer. Mais pour pouvoir faire des animations d'eau s'écoulant dans un évier ou tourbillonnant dans un siphon, le meilleur moyen actuel est de nouveau de partir des équations différentielles vérifiées par les fluides incompressibles, à savoir principalement l'équation de Navier-Stokes<sup>2</sup>.

Ici encore, les modèles simulant les équations classiques d'hydrodynamique font se propager les déformations sur les nœuds de leur discrétisation. Mais contrairement aux approches précédentes, ce n'est pas l'objet qu'ils discrétisent, mais *l'espace* : en chaque petit élément de volume de l'espace, ils déterminent quelle quantité d'eau entre et sort en fonction du champ de vitesses, et tiennent ainsi à jour les lieux où l'eau se trouve. Il maintiennent ainsi un champ de vitesse et de densité, permettant de caractériser entièrement l'emplacement et la vitesse de l'eau.

### a) Un premier modèle très simplifié

Un premier modèle d'animation d'eau fut proposé en 90 par Michael Kass et Gavin Miller [KM90]. Une hypothèse simplificatrice est faite dès le début : on considérera que l'eau se représente par une fonction à deux dimensions,  $z = f(x, y)$ , discrétisée en colonnes d'eau de hauteurs différentes, comme schématisé sur la figure 2.8.

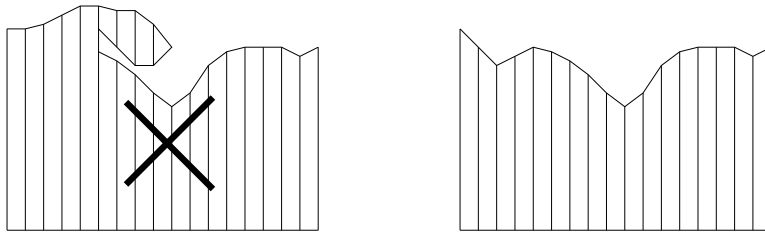


FIG. 2.8 - Une approximation de l'eau en colonnes d'eau ne convient pas toujours

Après avoir linéarisé l'équation de Navier-Stokes, ils se ramènent à une équation de propagation d'onde en deux dimensions. L'animation est donc fort simple : on trouve par différences finies les variations de hauteurs d'eau de chaque colonne de discrétisation. Les déformations se propagent alors naturellement, créant ainsi des vagues. Mais la linéarisation des équations du comportement a eu raison d'un certain nombre de propriétés intéressantes de l'eau, comme les écoulements tourbillonnaires ou les fameux rouleaux de bord de mer. Les simulations créées, bien que pratiques car très rapides, ne correspondent qu'à des situations bien particulières.

### b) Une simulation plus réaliste

Récemment, un modèle d'eau beaucoup plus fin a été proposé par Nick Forster et Dimitri Metaxas [FM96]. Cette fois, il s'agit d'une pure simulation 3D de l'équation de Navier-Stokes. Les auteurs discrétisent en effet l'espace en éléments de volume élémentaires, puis, toujours en utilisant les différences finies, y intègrent le champ des vitesses de l'eau dû au transport de l'eau de cellule en cellule. On est donc très proche des simulations physiques très précises, et le réalisme s'en ressent.

Quant à la représentation graphique de l'eau, elle est faite par un ensemble de marqueurs dont le mouvement est intégré dans le champ de vitesses de l'eau : l'affichage de ces très nombreuses

2. Une introduction sommaire à l'hydrodynamique est donnée dans l'annexe B.

petites particules permet de visualiser l'eau dans l'espace.

Ce modèle, relativement coûteux mais très convaincant, a été étendu depuis à la simulation de gaz, sans besoin particulier d'innovation puisque seules les équations régissant le mouvement changent [FM97].

### c) Discussion

L'animation de l'eau met en œuvre des techniques fort différentes de celles évoquées jusqu'à présent dans ce document : utiliser des variables de type pression ou densité, et partitionner l'espace plutôt que l'objet diffère grandement des approches animant des objets déformables. Pourtant, les résultats sont dignes d'intérêts et de telles méthodes peuvent apporter beaucoup aux méthodes traditionnelles.

Leur principal défaut est sûrement d'être difficilement combinables avec d'autres objets animés : les obstacles par exemple doivent être définis en terme de voxels pour être traités correctement [FM96]. La matière n'est en effet pas vraiment localisée, seuls les transferts de matière sont traités. C'est pourquoi, pour visualiser la matière en cours de simulation, on est obligé d'utiliser un nombre important de petits marqueurs qui suivent le champ de vitesses. Il est alors délicat de faire interagir des modèles de type différents, empêchant de créer des animations complexes.

Pendant, le point important ici est de remarquer la différence d'approche dans la discrétisation : alors que les techniques classiques de simulation, dites Lagrangiennes, discrétisent la matière et suivent l'évolution des nœuds de discrétisation, les techniques d'hydrodynamique utilisent une approche dite Eulérienne considérant une discrétisation fixe de l'espace d'où l'on observe le transport de la matière. Nous reviendrons dans ce document sur les différences théoriques entre ces deux approches a priori antagonistes; on pourra se référer à l'annexe A qui expose leurs principales caractéristiques.

## 2.3 Gestion des interactions entre objets déformables

Modéliser un objet aussi complexe soit-il reste vain si le problème de l'interaction avec le reste du monde n'est pas pris en compte. En effet, l'un des intérêts principaux des modèles générateurs est le traitement automatique des collisions, sans lequel deux objets pourraient allègrement passer au travers l'un de l'autre. Des solutions satisfaisantes ont déjà été conçues pour les contacts entre objets solides, grâce entre autres à la théorie des impulsions [Bar92, Fau97]. Mais pour ce qui concerne les objets déformables, les lois physiques capables de décrire l'état d'équilibre d'un objet soumis à des contraintes surfaciques données ne sont valables que pour de petites déformations, et de plus, fournissent des systèmes d'équations trop lourds à résoudre.

C'est pourquoi beaucoup de méthodes ont été développées au cas par cas, chacune pour une application bien précise. Nous allons essayer de dégager le principe des méthodes qui reviennent le plus souvent.

### 2.3.1 On dispose d'une surface externe

S'il existe une frontière entre les différents objets, la méthode dite "nodale" est en général appliquée. Le terme "nodale" provient du fait que l'on discrétise la surface en nœuds. Tout

consiste alors à faire une détection des interpénétrations nœud par nœud pour connaître les zones en interaction, puis en déduire des forces à appliquer à ces nœuds pour modéliser les diverses natures de chocs.

### a) Détection des collisions

Dans le cadre de sa simulation de reptiles, Miller utilisait des splines (surfaces paramétrées) pour couvrir ses masses en réseaux. Cela permet en effet d'obtenir des surfaces lisses pour le rendu. Mais quand les splines sont utilisées comme composant géométrique dans un modèle hybride comme pour [Mil88] ou [GVP91], c'est le module sous-jacent qui détecte les interactions, puisque de toute façon, on ne saurait pas gérer la déformation si la détection se faisait au niveau de la surface spline. Voyons donc comment traiter les détections dans le cas où l'on dispose d'une facétisation de la surface.

#### Principe de Moore et Wilhelms

La première chose à faire lors d'un pas d'animation est de détecter les points de contact. On a supposé que l'on dispose de points se situant sur la surface ; on peut donc définir un ensemble de facettes triangulaires constituant la "peau" de l'objet déformable. Une technique très employée consiste alors à vérifier si le chemin parcouru par une masse ne traverse pas une facette d'un autre objet déformable.

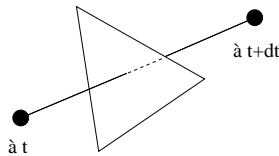


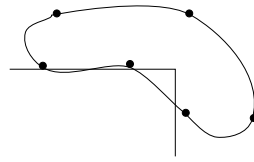
FIG. 2.9 - Détection de collision sur facette fixe

Si l'idée est simple, la mise en œuvre requiert un temps en  $O(n^2)$  où  $n$  représente le nombre de sommets du maillage. Il faut en effet se rendre compte que l'on doit trouver l'intersection d'un segment (la trajectoire du point) et d'un triangle en mouvement. Le système à résoudre se ramène à une équation de degré 5. On pourra noter que l'usage de boîtes englobantes pour une "prédétermination" accélère la détection, de même que l'usage de codages hiérarchiques des points dans l'espace comme les octrees.

#### Problèmes engendrés

Comme le fait remarquer Gavin Miller dans [Mil88], ce type de détection reste imprécis. En effet, pour ses serpents, l'utilisation d'une telle méthode interdit un obstacle de type escalier : dans ce cas, une particule pourrait être hors du sol aux instants  $t$  et  $t + dt$ , mais avoir traversé l'arête de l'escalier pendant ce laps de temps. Cela pourrait avoir comme fâcheuse conséquence que la peau pénètre dans une marche. Il faudrait alors calculer les contacts entre la spline de recouvrement et les obstacles, ce qui devient très lourd.

Le même type de problème d'interpénétrations locales voire de rebond avant le contact effectif se produit avec le modèle de [GVP91], qui utilise des B-splines contrôlées par des nœuds pour

FIG. 2.10 - *Problème de détection sur des splines*

visualiser la surface. On doit de toute façon bien être conscient que pour pouvoir détecter des collisions, il faut connaître des points se situant sur le contour de l'objet. Seulement, si cela est relativement aisé sur des objets élastiques, la tâche est plus ardue quand on manipule des objets plastiques, qui peuvent changer de topologie. Les points en surface peuvent varier constamment.

### Utilisation de surfaces implicites

Pentland et Williams [PW89] proposent une toute autre approche pour leur modèle. Ils utilisent en effet des surfaces implicites pour la représentation de leurs objets<sup>3</sup>. L'avantage est alors de disposer d'un test numérique permettant de savoir si un point est hors de l'objet ou non. Ainsi, la phase de détection se fait en  $O(n)$  pour  $n$  points de discrétisation, puisqu'il suffit de tester si les points de la surface d'un objet ne sont pas dans l'autre. Nous reverrons cette méthode plus en détail au chapitre 3.

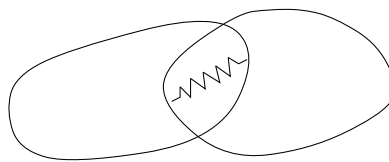
### b) Réponse aux collisions

Détecter précisément et rapidement les collisions entre objets est un premier pas. Reste à traiter la collision qui déformera les objets en interaction de façon cohérente et efficace.

### Méthode de pénalisation

Demetri Terzopoulos [TPF89] propose de pénaliser la pénétration des objets déformables dans des obstacles rigides (sphères, cylindres, plans) en les entourant d'un champ de force exponentielle.

Plus généralement, [MW88] simule une force de réaction entre deux objets rigides ou déformables en incorporant temporairement un ressort fictif entre les points de plus forte interpénétration. Le ressort est de longueur nulle au repos, et disparaît dès que le contact cesse. Cette méthode a, depuis, été utilisée avec succès par nombre d'auteurs ([TF88b, PW89]), qui l'ont appliquée telle quelle ou avec quelques modifications.

FIG. 2.11 - *Adjonction d'un ressort fictif*


---

3. Une définition précise des surfaces implicites sera vue au chapitre 3.



Un inconvénient peut cependant être relevé : ce ressort étant souvent assez dur, il faut, pour ne pas voir apparaître d'oscillations parasites, diminuer le pas d'intégration comme on l'a vu durant l'étude des réseaux masses-ressorts (voir 2.1.2).

### Les contraintes de réaction

John Platt et Alan Barr proposèrent en 1988 une méthode traitant les chocs entre un objet déformable discrétisé et un solide polygonal rigide [PB88]. Le principe est de rajouter une force supplémentaire ramenant l'objet à la surface du solide. Pour ce faire, après avoir détecté une interpénétration imminente entre une masse élémentaire (ou un nœud de discrétisation) et une facette du solide, on modifie la résultante des forces en remplaçant la composante normale à la facette par une force qui évitera la collision. Cette force est calculée pour que la masse élémentaire s'arrête à la surface du solide (sur la facette). Les calculs sont donc simples pour un solide, mais sont dissuasifs en ce qui concerne les collisions entre objets déformables. La force mise en jeu est une force de contrainte, et non une "force de réaction" car elle apparaît avant le contact et occulte artificiellement une composante de la résultante des forces.

### Approches spécifiques : exemple de Miller

Plusieurs techniques ont été utilisées dans des situations bien précises, et ne peuvent donc point être généralisées. Un exemple : Miller, dans sa simulation de serpents [Mil88], utilise une astuce pour éviter que ses serpents ne pénètrent dans le sol sur lequel ils rampent. Le corps du serpent est, comme on l'a vu en 2.1.2, constitué de masses élémentaires. Si après une étape de simulation, une des masses se retrouvent sous le niveau du sol, on change sa position en prenant la symétrique de la position calculée par rapport au sol. Ainsi, comme le montre la figure 2.12, on a simulé un rebond.

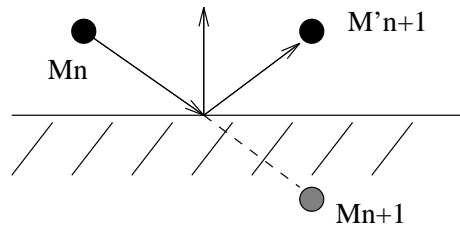


FIG. 2.12 - Méthode du rebond de Miller

Cette même approche est utilisée dans [MP89], mais pour des particules. Cependant l'utilisation d'une telle méthode entraîne une discontinuité dans le mouvement des particules. Vu la discontinuité de vitesse que cette méthode crée, les schémas d'intégration d'ordre élevé ne conviennent plus. Il faut alors se rabattre sur l'intégration d'Euler, d'où un pas d'intégration forcément très petit.

### 2.3.2 On ne dispose pas de surface externe : le cas des particules

Les particules sont un cas particulier en ce qui concerne les collisions. En effet, les forces d'interaction entre deux matériaux différents sont généralement de type répulsives à courte distance. Ainsi, les collisions sont *automatiquement* évitées. On ne peut même pas à proprement

parler de contacts ni de collisions, puisque les particules ne peuvent se toucher. En fait, le modèle particulaire possède l'avantage d'avoir une gestion unifiée à la fois des comportements physiques et des collisions. Ces avantages permettent des temps de simulation minimales, et ouvrent la porte aux possibilités d'interaction temps réel avec les matériaux simulés.

Cependant, les systèmes de particules sont souvent enrobés afin de disposer d'une surface externe à afficher. Mais dans tous les cas, *ces surfaces ne servent qu'à l'affichage* et ne sont pas considérées dans la détection de collisions, d'où une multitude d'interpénétrations parasites lors d'animations.

## 2.4 Discussion

Après avoir passé en revue les méthodes existantes, il apparaît clairement que la plupart des modèles sont bien souvent optimisés pour telle ou telle application, mais ne permettent pas d'aborder une grande gamme de comportements de façon unifiée. Seuls les *systèmes de particules* offrent à la fois des temps de calcul raisonnables et une vision unifiée des comportements allant de l'élastique au plastique. Ils sont en outre les seuls qui offrent un modèle général pour les fractures (objets se cassant en morceaux), et semblent donc les plus aptes à simuler un matériau hautement déformable.

Cependant, ils ne répondent pas tels quels à notre but, qui est de calculer une représentation surfacique d'un matériau hautement déformable pouvant être utilisée lors du rendu d'une animation. De plus, dans le cadre du calcul d'une animation visuellement réaliste, il est essentiel d'assurer des contacts surfaciques cohérents entre deux objets déformables en interaction, ce qui n'est le cas d'aucun des modèles présentés.

L'utilisation de surfaces splines pour modéliser la surface des objets ne peut convenir, du fait des changements de topologie que l'on aimerait pouvoir simuler. On ne peut en effet pas disposer constamment d'un réseau de points dont les adjacences sont connues. Il suffit pour s'en convaincre de penser à un objet capable de fractures : la connexité originelle se perd à la première cassure. Reconstruire une surface sur un tel objet demande une phase de calcul loin d'être négligeable. Enfin, on a pu voir au paragraphe 2.3.1 que l'absence d'une caractérisation mathématique simple de l'intérieur d'un objet rend la détection des interactions lente. Reste alors l'usage des *surfaces implicites* pour laquelle nous avons vu que la détection des chocs est facilitée. Nous allons découvrir lors du prochain chapitre les autres avantages de ces surfaces, en détaillant les progrès de ce domaine récent en Synthèse d'Images.



## Chapitre 3

# Modélisation par surfaces implicites

Modéliser un objet, c'est en donner une représentation. En synthèse d'images, la technique la plus courante de modélisation d'un objet est la définition de sa surface, ou *B-rep* pour *boundary representation*. À base de polygones ou de patches paramétrés, on peut ainsi définir un modèle synthétique et l'afficher sur écran.

Depuis maintenant une quinzaine d'années, une autre représentation prend peu à peu ses marques dans le monde de la modélisation. Cette fois, la définition de l'objet est obtenue en caractérisant son intérieur par une fonction mathématique : on parle alors de *V-rep* pour *volume representation*. La surface de l'objet, quant à elle, est définie de façon implicite, comme l'interface entre l'intérieur et l'extérieur. Si cette méthode paraît de prime abord fort peu intuitive, les développements incessants des techniques dans ce domaine ont permis d'en faire un outil de modélisation très simple d'usage.

Nous examinerons dans ce chapitre l'évolution des surfaces implicites depuis leur apparition en synthèse d'images, et tenterons de faire apparaître leur intérêt dans le cadre de l'animation d'objets déformables.

### 3.1 Définition d'une surface implicite

Une surface implicite est définie comme l'ensemble des points vérifiant une équation :

$$S = \{ \mathbf{P} \in \mathbb{R}^3 \mid f(\mathbf{P}) = iso \} \quad (3.1)$$

où *iso* est une valeur réelle appelée *isovaleur*, et *f* une fonction quelconque de  $\mathbb{R}^3$  dans  $\mathbb{R}$ , appelée *potentiel*. Les points intérieurs à l'objet peuvent être définis par la propriété :

$$f(\mathbf{P}) \geq iso \quad (3.2)$$

Trivialement, l'extérieur possède alors la caractérisation duale :  $f(\mathbf{P}) < iso$ . On dispose ainsi d'une fonction dedans/dehors, facilitant le rendu d'images composées d'objets implicites. Les techniques de lancer de rayons ont en effet besoin de connaître l'intersection entre un rayon et un objet. La normale d'un point à la surface d'un objet implicite, définie par une fonction *f*, est calculée par le gradient de *f* (noté  $\nabla f$ ) en ce point.

## 3.2 Historique des formulations

Les surfaces implicites, introduites en synthèse d'images dans le début des années 80, ont évolué régulièrement pour arriver aujourd'hui à des techniques attrayantes et efficaces, en faisant un outil de modélisation puissant.

### 3.2.1 Potentiel analytique

Les premières approches utilisèrent principalement des polynômes pour le potentiel  $f$ . Les objets ainsi définis étaient essentiellement des ellipsoïdes, des super-ellipsoïdes [WW89], des hyper-quadrriques [Han88], ou des super-quadrriques [Bar81, Bar84], comme sur la figure 3.1, qui se formulent ainsi :

$$f(x, y, z) = \frac{x^k}{a^k} + \frac{y^k}{b^k} + \frac{z^k}{c^k}.$$

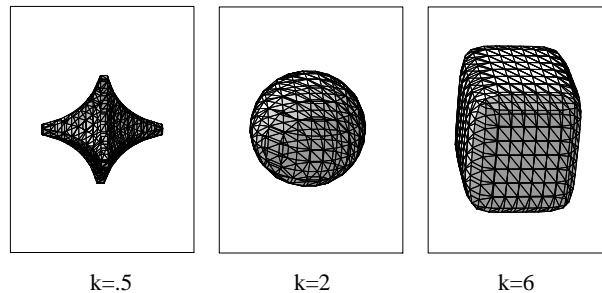


FIG. 3.1 - Exemples de super ellipsoïdes

### 3.2.2 Potentiel à points clés

Les formes trop limitées et le manque de paramètres intuitifs ont rapidement amené à une reformulation complète. Cette fois, les objets sont constitués de squelettes ponctuels et de fonctions potentiel. L'idée provient des champs électriques : un ensemble de particules chargées dessinent des isosurfaces pouvant être complexes, mais de forme toujours lisse. Ainsi, l'isosurface d'un ensemble de points clés émettant de l'énergie définit une surface implicite. Les points clés jouent le rôle de squelettes pour l'objet ainsi défini, et permettent de contrôler sa forme finale de façon intuitive.



FIG. 3.2 - Une fonction potentiel et la surface engendrée par deux points clés.

Trois méthodes principalement se sont distinguées, proposant à chaque fois leur fonction potentiel propre :

- **les Blobs** [Bli82], pour Blinn Objects : ici la fonction potentiel est une exponentielle

$$f(\mathbf{P}) = \sum_{\text{points clés}} e^{-\frac{ad_i^2}{2}},$$

où  $d_i$  est la distance de  $\mathbf{P}$  au point clé  $i$ . Avec cette formulation, le potentiel induit par un point ne s'annule jamais. Cela implique une modélisation délicate : l'ajout d'un point clé influence toute la scène.

- **les Meta-Balls** [NHK<sup>+</sup>85] : Nishimura, quant à lui, supprime l'exponentielle pour mettre une fonction polynomiale, décroissante et à support fini (c'est-à-dire qui s'annule au delà d'une certaine distance). Triple avantage : le calcul du potentiel est plus rapide car plus simple; on peut économiser le calcul du potentiel pour les points trop éloignés; et enfin, la modélisation devient plus locale : l'ajout d'un point ne modifie la surface que dans une sphère dite d'influence.
- **les Soft Objects** [WMW86] : De son côté, Brian Wyvill optimise lui aussi la fonction potentiel en utilisant un polynôme en la distance au carré. Cela évite des évaluations de racines carrées, très lentes, et permet ainsi des temps de calcul plus raisonnables. Le modèle résultant est très proche des meta-balls.

Ces modèles permettent déjà une modélisation beaucoup plus aisée, où l'on peut contrôler la forme par simples déplacements des points clés. Wyvill a étendu encore cette modélisation en ajoutant des points clés à influence négative : cette fois, le point clé *creuse* l'objet en repoussant la surface tout autour de lui [WW89].

### 3.2.3 Potentiel à squelettes

Plus généralement, Jules Bloomenthal et Brian Wyvill [BW90] proposent de définir des surfaces implicites engendrées par des squelettes. Ces squelettes peuvent être n'importe quelle primitive géométrique pour laquelle on sait calculer la distance à un point. Chaque squelette crée dans l'espace un potentiel qui décroît avec la distance. On considère alors qu'à chaque point de l'espace est associé un potentiel, résultant de la somme des potentiels créés par chacun des squelettes. Une définition des surfaces définies par un ensemble de squelettes  $s_i$  est alors :

$$S = \left\{ \mathbf{P} \in \mathbb{R}^3 \mid \sum_{s_i} f_i(r_i) = iso \right\}$$

où  $r_i$  est la distance de  $\mathbf{P}$  à  $s_i$  et  $f_i$  est une fonction décroissante.

On remarque que la caractérisation de l'intérieur de l'objet reste identique (voir équation 3.2), et que la construction d'objets même complexes est assez intuitive. Un squelette ponctuel donnera une sphère, un segment donnera un cylindre arrondi aux extrémités, etc. Le mélange de squelettes fournira un moyen simple et économique de création d'objets complexes.

Une fonction potentiel  $f_i$  peut être modélisée par toute fonction décroissante à support compact (voir figure 3.2). Le rayon d'influence  $R$  assure qu'un squelette ne changera pas toute

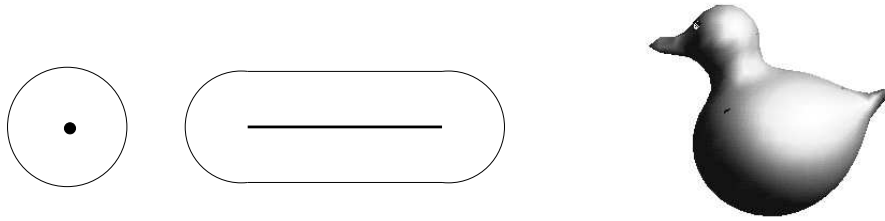


FIG. 3.3 - Exemples de surfaces implicites

la forme d'un objet, mais seulement une zone locale, ce qui permet une construction squelette par squelette d'objets complexes.

On appellera  $e$  l'épaisseur d'"enrobage" que représente une fonction potentiel, c'est-à-dire le rayon qu'aurait la sphère engendrée par un squelette ponctuel muni de cette fonction. On a bien sûr la relation :  $e = f^{-1}(iso)$ .

### 3.2.4 Potentiel procédural

Jules Bloomenthal [BW90] fait aussi remarquer que le potentiel peut être défini procéduralement, comme une boîte noire. Ceci peut d'ailleurs simplifier la tâche de l'utilisateur dans quelques cas. Par exemple, si l'on souhaite un embranchement lisse entre deux branches, on peut modifier le calcul de la distance aux squelettes en utilisant la méthode géométrique de la figure 3.4 pour arriver à nos fins. À remarquer d'ailleurs que cette formulation n'est  $\mathcal{C}^1$  que pour des angles d'embranchement inférieurs à  $\pi/2$ . Une solution plus générale à ce problème spécifique peut être trouvée dans [FGA96].

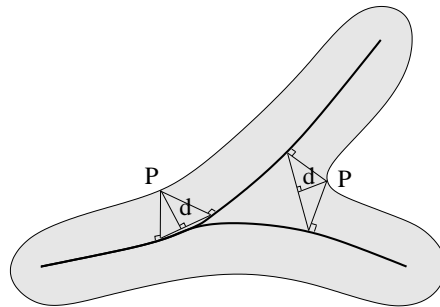


FIG. 3.4 - Construction géométrique pour un embranchement lisse

### 3.2.5 Potentiel intégral

Bloomenthal et Shoemake ont proposé un type de surfaces implicites encore différent, les *surfaces de convolution* [BS91]. À la base de cette modélisation, le problème des bosses (*bulges* en anglais) : si deux squelettes segments se coupent en croix, la surface implicite générée par les modèles précédents aura une bosse, autrement dit une épaisseur plus importante à la jonction des deux segments (voir figure 3.5). Pour éliminer ce défaut, il est possible d'écrire le potentiel

sous forme de sommes d'intégrales :

$$f(\mathbf{P}) = \sum_{\text{squelettes}} \int_{\text{squelette } S_i} e^{-\|\mathbf{P}-\mathbf{X}\|^2/2} d\mathbf{X}$$

Cette formulation est en fait une convolution du squelette par une gaussienne, permettant de tenir compte des contributions tout le long du squelette. Et comme la convolution est un opérateur linéaire, la somme de convolutions est équivalente à la convolution de la somme des squelettes ; il n'y donc pas de bosse à l'intersection de deux squelettes en croix.

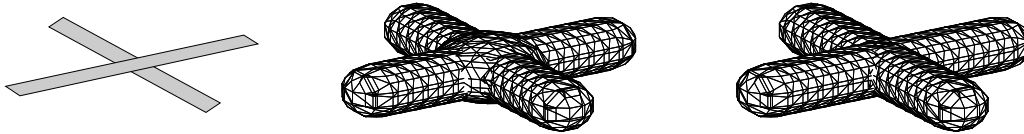


FIG. 3.5 - Un squelette, la forme bosselée qu'il engendre, et sa version convoluée.

Ce type de surface nécessite bien sûr plus de calculs, mais peut s'implémenter tout de même de façon efficace, d'autant plus que n'importe quelle fonction de filtre, plus simple que l'exponentielle, peut être utilisée. Cette technique n'est cependant utilisée qu'en dernier recours si l'on ne peut éviter l'utilisation d'une intégrale.

### 3.2.6 Potentiel composé

Tous les potentiels vus jusqu'ici peuvent être composés à une *déformation* quelconque de l'espace (en anglais : *warping*), sous la forme suivante :  $S = \{ \mathbf{P} \in \mathbb{R}^3 \mid f \circ W(\mathbf{P}) = iso \}$ .

La transformation  $W$  peut être quelconque. Par exemple, elle peut être écrite comme combinaison d'une translation et d'une rotation sous la forme  $W(\mathbf{P}) = R^{-1}(\mathbf{P}-\mathbf{t})$ , ce qui permet de décaler et tourner toute surface implicite [SP91]. Des transformations moins triviales [RCGG<sup>+</sup>97], comme un vissage autour d'un axe, permettent de modéliser très facilement des formes plus complexes, comme le prouve la colonne de l'image 3.6.

### 3.2.7 Potentiel sous forme de R-fonctions

Alexander Pasko [PASS95] a vulgarisé l'utilisation des R-fonctions pour la modélisation des surfaces implicites. Cette technique est intéressante car elle permet, de façon analytique, d'approcher des unions ou des intersections d'objets implicites.

En effet, on remarque facilement que l'union de deux surfaces implicites définies par  $f_1$  et  $f_2$  est tout simplement :  $Union_{f_1, f_2}(\mathbf{P}) = \max(f_1(\mathbf{P}), f_2(\mathbf{P}))$ . Par le même principe, on a :  $Intersection_{f_1, f_2}(\mathbf{P}) = \min(f_1(\mathbf{P}), f_2(\mathbf{P}))$ . Or, un mathématicien russe, Rvachev, avait introduit dès 1967 les fonctions suivantes :

$$f_1 \vee_{\alpha} f_2 = \frac{f_1 + f_2 + \sqrt{f_1^2 + f_2^2 - 2\alpha f_1 f_2}}{1 + \alpha}$$



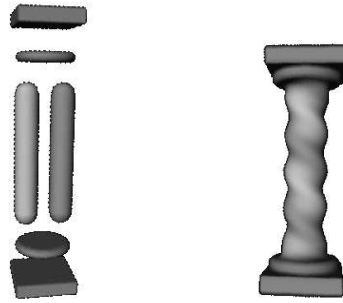


FIG. 3.6 - Une colonne constituée de deux segments, deux disques, et deux parallépipèdes. Le potentiel utilisé pour les segments est composé d'une déformation de type vissage (d'après [RCG+97]).

$$f_1 \wedge_\alpha f_2 = \frac{f_1 + f_2 - \sqrt{f_1^2 + f_2^2 - 2\alpha f_1 f_2}}{1 + \alpha}$$

qui ont les propriétés remarquables :

$$\lim_{\alpha \rightarrow 1} f_1 \vee_\alpha f_2 = \frac{f_1 + f_2 + |f_1 - f_2|}{2}$$

$$\lim_{\alpha \rightarrow 1} f_1 \wedge_\alpha f_2 = \frac{f_1 + f_2 - |f_1 - f_2|}{2}$$

On aura reconnu respectivement la fonction maximum et la fonction minimum dans ces deux limites. Ces deux R-fonctions généralisent donc la notion d'union et d'intersection, ce qui permet d'obtenir des objets  $\mathcal{C}^1$  très proches d'objets CSG (*Constructive Solid Geometry*).

Dans la même lignée, Wyvill a proposé très récemment une modélisation CSG (cette fois pour *Constructive Soft Geometry*) d'objets implicites, mais en utilisant un arbre d'évaluation. Le principal défaut reproché aux surfaces implicites, à savoir leur aspect trop lisse, a donc trouvé deux solutions bien distinctes, et permet maintenant une modélisation très complète. Ce progrès est illustré par la figure 3.7, montrant les évolutions sur dix ans.

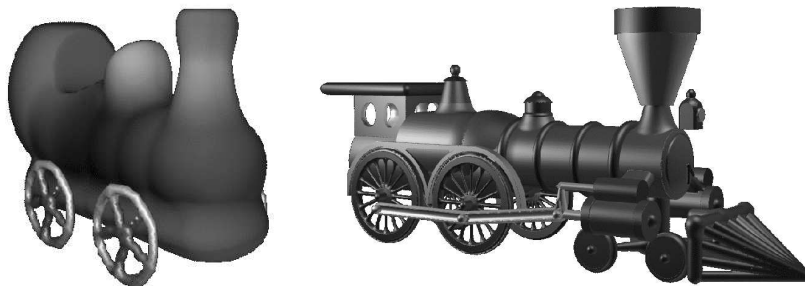


FIG. 3.7 - Le désormais fameux train de Wyvill : versions 1985 et 1995.

### 3.2.8 Potentiel optimisé

Afin de simplifier au maximum le temps d'évaluation du potentiel, des formulations simples et génériques ont été proposées [BS95]. Par exemple, l'utilisation de fonctions polynômiales rationnelles par morceaux permet des potentiels à raideur (pente à l'isovaleur) paramétrable en seulement une division, deux multiplications et une addition.

## 3.3 Échantillonnage pour la visualisation

La formulation implicite de la surface semble de prime abord un handicap pour sa visualisation. Contrairement aux surfaces paramétrées, les points d'échantillonnage répartis sur la surface permettant la polygonalisation ne sont pas donnés explicitement. Dans le cas d'une surface implicite quelconque, chaque point que l'on souhaite trouver sur la surface nécessite donc une recherche dichotomique sur le potentiel.

Différentes techniques d'échantillonnage ont été développées, permettant une visualisation rapide de surfaces implicites quelconques. Ces méthodes sont de deux types bien distincts : par partitionnement de l'espace, et par simulation de particules sur la surface.

### 3.3.1 Discrétisation polygonale par partitionnement spatial

#### Approche générale

L'idée à la base de cette approche décrite par Brian Wyvill [WMW86] est de partitionner l'espace en voxels<sup>1</sup> cubiques réguliers et de trouver les intersections de la surface de l'objet à discrétiser avec les arêtes de ces voxels. Une arête contient une intersection si ses extrémités sont de part et d'autre de la surface. On trouve donc les arêtes concernées en testant le potentiel de chacun des sommets des cubes. Puis une dichotomie permet de connaître les intersections exactes. Selon le nombre d'intersections, entre un et trois triangles sont tracés de façon à approcher la surface localement : quelques cas (parmi les  $2^8 = 256$  existants) sont représentés sur la figure 3.8. Malgré le nombre de cas à traiter, des implémentations simples à base de tables pré-calculées ont été proposées [Blo94].

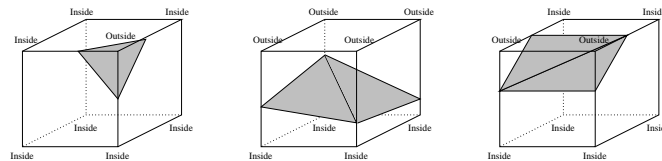


FIG. 3.8 - Trois cas de polygonalisation d'un voxel selon les valeurs aux sommets.

#### Traque de la surface

On peut bien sûr éviter de parcourir l'ensemble des cubes de l'espace pour discrétiser la surface implicite. Une fois le premier cube intersectant la surface trouvé, il suffit de gérer une liste des cubes à polygoniser : en effet, en examinant les voisins de chaque cube traité, on peut longer la surface et ne traiter que l'ensemble des cubes autour de la surface [WMW86, LC87]. Cette méthode est maintenant bien connue sous le nom de *Marching Cubes*.

1. voxel, pour *volume element*, par analogie à pixel, pour *picture element* et/ou *picture cell*.

## Problèmes d'ambiguïté topologique

La discrétisation polygonale par partitionnement spatial pose cependant des problèmes d'imprécisions et de non respect de topologie [VGW90]. Tout d'abord, si une composante connexe d'une surface implicite est strictement incluse dans un voxel, elle sera purement et simplement oubliée dans la polygonalisation finale. Comme on peut régler la taille des voxels, ce problème n'est, dans certains cas, que peu dérangeant. Par contre, pour la même raison, la représentation polygonale pourra être de topologie différente de la vraie surface. Pour s'en convaincre, il suffit de penser à une surface avec au milieu, un fin goulot d'étranglement. Si la section de ce goulot est plus petite qu'une face de voxel, il se peut que la surface soit alors coupée en deux parties distinctes, le goulot n'étant pas détecté.

Une grande partie des problèmes de topologie peut être éliminée en décomposant chacun des voxels en six tétraèdres. Non seulement la polygonalisation de chaque tétraèdre devient alors plus simple (plus que  $2^4 = 16$  cas), mais de plus, la polygonalisation est plus précise, gérant mieux les topologies délicates (voir figure 3.9).

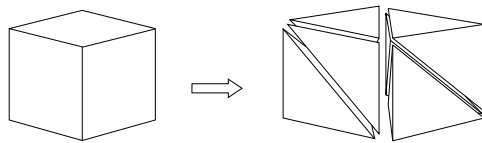


FIG. 3.9 - Décomposition du cube en six tétraèdres

Le défaut de cette technique est évidemment de fournir un modèle avec beaucoup plus de polygones que la méthode classique. Et pourtant, cette polygonalisation ne permet toujours pas d'assurer un respect scrupuleux de la topologie [SH97], sauf si bien sûr, on dispose d'informations supplémentaires sur la surface. Récemment, Jacques-Olivier Lachaud a proposé une technique basée sur la géométrie discrète, qui permet d'obtenir une triangulation cohérente, sans singularité ni auto-intersection [Lac96].

### 3.3.2 Échantillonnage par particules

#### Idée de base

La non-régularité des triangles fournis par une polygonalisation de type Marching Cubes a amené Figueiredo à proposer une méthode de simulation permettant de converger vers une discrétisation quasi-régulière [FGTV92]. Il s'agit tout simplement de considérer un système de masses/ressorts, comme vu au paragraphe 2.1.2, auquel on ajoute une force contraignant les masses à rester sur la surface. Une fois le système à l'équilibre, on obtient une discrétisation quasi-régulière de la surface grâce aux ressorts.

Par la suite, Witkin et Heckbert vont améliorer cette méthode en utilisant des particules reliées par des forces de répulsion uniquement [WH94]. Un mécanisme très simple de subdivisions automatiques et de destructions aléatoires permet aux particules d'envahir rapidement la surface à discrétiser, et d'atteindre interactivement une position d'équilibre, comme l'illustre la figure 3.10. Alors, si la surface implicite se déforme lentement, les particules sont capables de suivre le mouvement et s'adaptent automatiquement à tout changement de topologie, chose que ne pouvait faire la méthode à base de ressorts. On obtient ainsi une discrétisation régulière et robuste à tout mouvement. D'un autre côté, on ne dispose plus d'une véritable polygonalisation : il faudrait alors passer par un diagramme de Voronoï pour relier les particules correctement.

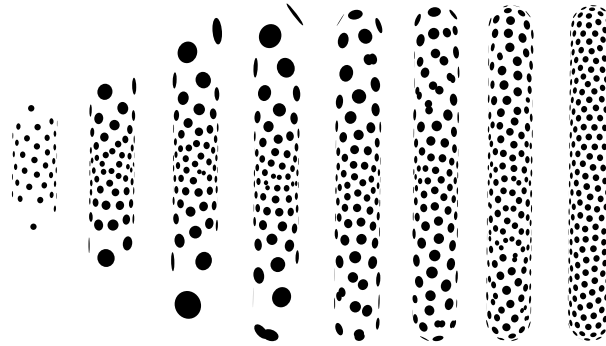


FIG. 3.10 - *Évolution des particules et discrétisation finale (d'après [WH94])*

### Développement récent : recherche de points singuliers

La méthode particulière a permis un développement intéressant : la recherche de points singuliers sur les surfaces [RRS96, SH97]. Les points singuliers, ou critiques, sont les points de la surface vérifiant  $\nabla f = 0$ . Et puisque les changements de topologie créent toujours au moins un point singulier, les problèmes d'ambiguïté topologique trouvent une solution élégante : il suffit de traquer l'apparition de tels points et de corriger localement la polygonalisation.

## 3.4 Contrôle de la forme

Le contrôle de la forme est un problème fondamental en modélisation. L'utilisation de surfaces implicites, même si elle règle bien des problèmes (comme les embranchements), apporte en la matière un problème connu sous le nom de *unwanted blending*, ou mélange indésirable. La figure 3.11 l'illustre simplement : autant les bras écartés, le personnage implicite à base de squelettes est plaisant, autant le bras baissé, le mélange qui se produit entre le corps et le bras rend le résultat surréaliste. Dans le même ordre d'idée, on peut aussi vouloir modéliser une main implicite, où les doigts fusionnent correctement avec la paume, mais sans que les doigts puissent se fondre les uns dans les autres.

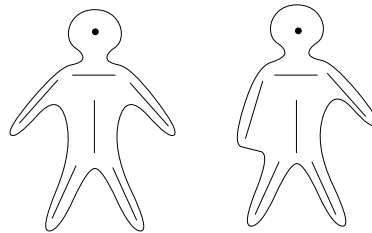


FIG. 3.11 - *Le problème du mélange indésirable*

Brian et Geoff Wyvill [WW89] ont suggéré une solution permettant de gérer ce genre de défaut, qui fut largement développée par la suite par Agata Opalach et Steve Maddock [OM93]. La solution passe par la définition d'un graphe de mélange de potentiels, permettant à tels et tels squelettes de mélanger leur potentiel respectif, alors que tel et tel ne doivent en aucun cas le faire. La procédure de calcul du potentiel en un point  $\mathbf{P}$  quelconque est alors :

- Calculer le potentiel dû à chacun des squelettes de l'objet.

- Calculer les sommes des potentiels par groupes de squelettes voisins dans le graphe de mélange.
- Retourner la valeur maximale.

Cette façon de faire pour le mélange des potentiels évite alors le problème cité. Cette solution comporte quand même un inconvénient : des discontinuités  $\mathcal{C}^1$  peuvent intervenir [GW95].

## 3.5 Outils divers

Les surfaces implicites disposent maintenant d'un nombre conséquent d'outils facilitant leur utilisation. En voici quelques uns susceptibles de nous intéresser dans ce document.

### 3.5.1 Reconstruction à partir de points de donnée épars

Il est important de pouvoir modéliser des objets réels à partir de données obtenues par scan-nerisation. Un certain nombre de techniques existent, qui permettent de trouver une représentation implicite en partant de points épars à la surface d'un objet. Muraki propose ainsi d'ajouter squelette après squelette afin d'affiner la reconstruction d'un objet, en minimisant l'écart entre surface reconstruite et points de données [Mur91]. Ont suivi nombre d'optimisations qui offrent aujourd'hui des outils de reconstruction efficaces [TBG95, BTG95].

### 3.5.2 Décoration par placages de textures

Une autre difficulté lorsque l'on travaille avec des surfaces implicites est le placage de textures. Faites pour décorer une surface, ces textures nécessitent une paramétrisation de la surface, chose qu'une formulation implicite ne fournit pas.

Cependant, il existe maintenant une méthode très au point qui remédie à ce problème [Ped95, Ped96]. Si l'on dispose d'une discrétisation de la surface obtenue par la méthode des particules décrite en 3.3.2, on peut paramétrer localement n'importe quel carreau dessiné sur la surface. En effet, la discrétisation régulière permet de définir des géodésiques discrètes sur la surface, que l'on peut lisser pour obtenir une vraie paramétrisation à deux dimensions via un maillage régulier. Ainsi, on peut décorer n'importe quelle surface implicite de façon interactive.

Dans le cas d'une surface animée, les décorations obtenues par cette méthode peuvent être maintenues, puisque l'échantillonnage suit la déformation. Une autre méthode permet d'affecter à ces textures plusieurs comportements en fonction du mouvement des squelettes qui le définissent : textures glissant sur la surface, textures fixes, et autres. Pour plus de détails sur cette technique, voir [SS96]. Mais le problème du texturage des surfaces implicites mouvantes reste cependant encore aujourd'hui un problème largement ouvert.

### 3.5.3 Rendu d'images de qualité

Les surfaces implicites se prêtent très bien au rendu par lancer de rayons, technique maintenant classique en Synthèse d'Images. Il est en effet possible de trouver l'intersection entre un rayon et une surface par dichotomie, ainsi que son rayon réfléchi, puisque la normale à la surface est donnée par le gradient du potentiel. Si d'un point de vue théorique, l'utilisation de surfaces implicites en lancer de rayons ne pose pas de problème particulier, il est cependant nécessaire de disposer de méthodes permettant d'accélérer le calcul des intersections entre rayons et surfaces,

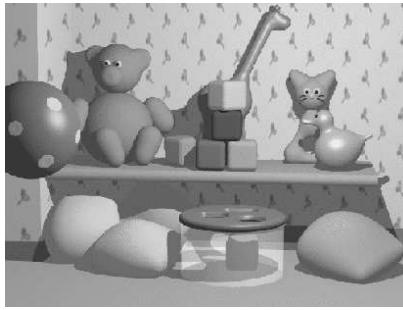


FIG. 3.12 - *Un exemple de rendu de qualité d'objets implicites : Simply Implicit [FGGL93]*

tout en garantissant de ne pas en rater [KB89, Gas95, Gal97] : les temps de rendu sont sans cela impraticables. Les images créées par lancer de rayons sont alors de très bonne qualité comme le prouve la figure 3.12.

### 3.5.4 Morphing entre objets implicites

Le morphing, ou métamorphose géométrique, trouve dans le cas de surfaces implicites à squelettes une solution simple et efficace : en associant les squelettes initiaux et finaux, on peut transformer un objet en un autre en déplaçant leurs squelettes respectifs [Wyv97]. Si le nombre et le type des squelettes n'est pas le même entre l'objet de départ et celui d'arrivé, une méthode faisant intervenir des sommes de Minkowski peut être utilisé [GA96, Gal97].

### 3.5.5 Conversion implicite/explicite, et vice-versa

Le problème de conversion entre modèle implicite et explicite est aussi étudié en détail depuis quelques temps. Une approche par analyse multiéchelle de la surface permet d'ores et déjà de convertir une forme donnée explicitement en une hiérarchie de squelettes définissant une surface implicite de plus en plus proche de la surface de départ [VTdMG94, VG96].

## 3.6 Ajout d'un modèle de déformation

Les surfaces implicites ont été toutefois peu utilisées en animation d'objets déformables avant 93. Avant cette date, un certain nombre d'animations avaient déjà fait un peu connaître les surfaces implicites : citons "The Great Train Robbery" de Wyvill, d'où est tirée la locomotive à gauche de la figure 3.7, ou encore le petit bonhomme en pâte dentifrice de Beier [Bei93].

Après ces modèles d'animation par mouvement de squelettes, un modèle déformable à base de surfaces implicites a été proposé [Gas93]. Conçu pour simuler des objets élastiques, il est très attrayant grâce à une modélisation fine et précise des contacts entre surfaces implicites.

### 3.6.1 Présentation du modèle

Le but est de simuler des matériaux élastiques qui assurent des contacts surfaciques exacts et dont les déformations restent locales. Plutôt que d'utiliser des équations Lagrangiennes générales, on néglige la variation de la distribution de masse au cours des déformations. Ceci permet d'utiliser la mécanique du solide pour le calcul du mouvement comme dans [GVP91], et donc

de gagner du temps de calcul. Pour ce faire, l'objet déformable est séparé en deux composantes distinctes, identiquement à [TF88a] :

- Une partie rigide définie par un repère, une masse et une matrice d'inertie correspondant à la forme de l'objet à l'état initial.
- Une partie déformable qui s'ajoute à la première pour modéliser la forme courante de l'objet.

### 3.6.2 Modélisation des propriétés physiques d'un objet implicite

Nous allons étudier attentivement dans cette section ce modèle de comportement élastique, reposant sur une définition implicite des objets : en effet, ce modèle est à la base d'une partie de notre contribution, développée dans le chapitre 5.

La forme d'un objet implicite ne dépend, comme on l'a vu, que de l'ensemble des points de l'iso-surface. Les variations de la fonction potentiel autour de cette isosurface peuvent donc être exploitées pour modéliser d'autres propriétés. L'idée à la base de [Gas93] est d'utiliser ces variations pour stocker les propriétés physiques de l'objet. On améliore ainsi encore un peu plus le taux de compression des informations à sauvegarder pour l'objet.

Nous allons voir comment faire une analogie cohérente entre la fonction implicite et la raideur, en trouvant une correspondance entre forces et déformations, ce qui permettra de définir le modèle déformable proprement dit.

#### Correspondance entre forces et déformations

Pour qu'un point  $\mathbf{P}$  de la surface d'un objet élastique, de raideur locale  $k_P$ , passe de la position  $\mathbf{P}_0$  à la position  $\mathbf{P}_1$ , il faut lui appliquer une force :

$$R(\mathbf{P}) = \int_{\mathbf{P}_0}^{\mathbf{P}_1} k_P(\mathbf{Y}) d\mathbf{Y} \quad (3.3)$$

En pratique, la correspondance entre forces et déformations sera utilisée lors de collisions, pour intégrer les forces de réaction colinéaires aux vecteurs normaux à la surface de contact. Ainsi, on peut se limiter à définir les propriétés élastiques uniquement selon des directions de déformation radiales.

Prenons alors une force  $dR(Y)$  radiale infinitésimale, et  $dY$  le déplacement radial provoqué par la force. D'après l'équation 3.3, on doit avoir :

$$dR(\mathbf{Y}) = k_P(\mathbf{Y}) \cdot d\mathbf{Y}$$

Si l'on exprime les déformations par les variations de la fonction potentiel  $f$ , on obtient :

$$df(\mathbf{Y}) = \nabla f(\mathbf{Y}) \cdot d\mathbf{Y} = \nabla f(\mathbf{Y}) \cdot \frac{dR(\mathbf{Y})}{k_P(\mathbf{Y})} \quad (3.4)$$

Puisque que l'on souhaite modéliser les propriétés physiques par les variations de la fonction potentiel à l'intérieur de l'objet implicite, il est judicieux de choisir (en rappelant que la direction de la normale est opposée à celle du gradient) :

$$\forall Y, \nabla f(\mathbf{Y}) = -k_P(\mathbf{Y}) \cdot \mathbf{n}(\mathbf{Y}) \quad (3.5)$$

Ce choix simplifie par la même occasion l'équation 3.4, et l'on aboutit à :

$$\int_{\mathbf{P}_0}^{\mathbf{P}_1} df(\mathbf{Y}) = - \int_{\mathbf{P}_0}^{\mathbf{P}_1} \mathbf{n}(\mathbf{Y}) \cdot d\mathbf{R}(\mathbf{Y}) = -\mathbf{n}(\mathbf{P}) \cdot \mathbf{R}(\mathbf{P})$$

puisque le vecteur normal  $\mathbf{n}(\mathbf{P})$  reste constant par définition lors d'une déformation radiale. Si on appelle alors  $g(\mathbf{P}) = f(\mathbf{P}) - f(\mathbf{P}_0)$  le *champ de déformation*, on a la relation immédiate :

$$g(\mathbf{P}) = -\mathbf{n}(\mathbf{P}) \cdot \mathbf{R}(\mathbf{P}) \quad (3.6)$$

grâce à l'équation précédente.

Si on modélise donc tous les objets de notre scène par des surfaces implicites dont les fonctions de potentiel vérifient l'équation 3.5, la relation  $g(\mathbf{P}) = -\mathbf{n}(\mathbf{P}) \cdot \mathbf{R}(\mathbf{P})$  définit parfaitement la correspondance *force/déformation radiale* recherchée, et permet comme on le verra par la suite de déduire d'une déformation donnée les forces de réaction résultantes.

### Modéliser la raideur avec les fonctions implicites

Nous venons de voir que la fonction potentiel  $f$  devait vérifier l'équation 3.5. Donc pour un point  $P$  de la surface d'un objet engendré par un squelette  $S$  et la fonction  $f$ , on doit avoir, après simplification par  $\mathbf{n}(\mathbf{P})$  :

$$k(\mathbf{P}) = -f'(d(\mathbf{P}, S)) = -f'(e)$$

Ainsi, il est très facile pour un graphiste de choisir la raideur d'un objet : il suffit de changer la pente de la courbe au point  $e$  (vérifiant  $f(e) = iso$ ). De plus, la pente en tout point de la portion de courbe sur  $[0, e]$  définit le comportement élastique à l'intérieur de l'objet. Deux choix sont souvent faits : soit on suppose que le matériau possède une élasticité linéaire (dans ce cas, la force de réaction est constante lors d'une déformation), soit on préfère appliquer une élasticité quadratique (la raideur de l'objet augmente alors progressivement lors d'une compression). Les deux solutions sont montrées sur la figure 3.13, mais un comportement quelconque peut aussi bien être modélisé.

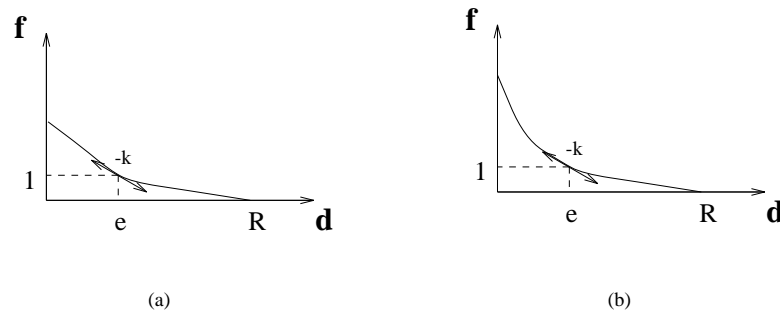


FIG. 3.13 - (a) Élasticité linéaire - (b) Élasticité non linéaire



### 3.6.3 Algorithme global

Le modèle étant construit, il faut maintenant planifier les différentes étapes d’animation à mettre en œuvre. Voici l’algorithme d’animation général, que nous détaillerons au fur et à mesure dans les paragraphes suivants :

1. Intégrer les équations du mouvement pour le composant rigide en tenant compte des forces  $\mathbf{F}$  et des moments  $\mathbf{T}$  externes :

$$\sum \mathbf{F} = m\mathbf{a}$$

$$\sum \mathbf{T} = I\dot{\Omega} + \Omega \wedge I\Omega$$

avec  $I$  la matrice d’inertie (supposée constante) du solide au repos,  $\mathbf{a}$  l’accélération et  $\Omega$  la vitesse angulaire.

2. Déplacer les squelettes en conséquence.
3. Traiter les interactions comme suit :
  - Détecter les interpénétrations entre objets implicites.
  - Déformer les composants implicites de manière à engendrer des surfaces de contacts exactes entre les objets.
  - Intégrer les forces de réaction et de friction sur ces surfaces. Réactualiser le bilan des forces en conséquence.
4. Afficher les objets tels qu’ils ont été déformés.

Cet algorithme est utilisé avec un pas d’intégration adaptatif. En effet, au cas où l’intervalle de temps soit trop grand, une importante interpénétration pourrait créer des déformations, et donc des forces complètement anormales. Ce problème est détecté et le système revient alors en arrière et recommence les calculs avec un intervalle de temps plus petit.

### 3.6.4 Mise en œuvre

#### Discrétisation initiale

L’algorithme présenté ici nécessite de discrétiser les objets implicites pour à la fois la détection des collisions, l’intégration numérique des forces de réaction le long des surfaces en contact, et la visualisation. Il s’agit donc de trouver un moyen économique de maintenir une discrétisation, c’est-à-dire une série de points de discrétisation à la surface de l’objet. Cependant, s’il existe plusieurs méthodes de discrétisation de surfaces implicites, aucune ne permet le ré-échantillonnage à chaque pas d’intégration en un coût raisonnable, comme on l’a vu au paragraphe 3.3.1.

Puisque les objets simulés dans [Gas93] ne changent pas de topologie au cours du mouvement, la méthode proposée consiste à initialiser des points sur la surface une seule fois, avant la simulation, et se contente de réactualiser localement la position de chacun de ces points de discrétisation au cours de l’animation. La méthode retenue pour cette première tâche est la technique dite des “Marching cubes”, ou polygonalisation par partitionnement de l’espace.

Cette discrétisation étant réalisée, on dispose maintenant de points sur la surface de l’objet. Or ces objets sont supposés élastiques : ils reprennent leur forme initiale après un choc ; ils ne

nécessitent donc pas un ré-échantillonnage à chaque pas de calcul. En effet, une sauvegarde après la phase des “Marching cubes” de la position de chaque point d'échantillonnage et de leur normale dans un repère local à l'objet, permet de les garder à leur place en cas de rotation ou de translation. Puis, en cas de chocs, on calcule la nouvelle position en cherchant, le long de leur normale de départ, où se situe la nouvelle surface. Ce choix de direction invariante assure que le point de discrétisation retrouvera sa position d'origine après un choc.

### Détection des collisions

Comme on l'a vu en 2.3.1, l'usage de surfaces implicites pour les objets permet un gain de temps considérable pour la détection des interpénétrations. En effet, en testant en un point  $\mathbf{P}$  de l'espace le potentiel induit par un objet (en fait, par ses squelettes), on sait que :

- $\mathbf{P}$  se trouve à l'intérieur si le potentiel est plus grand que *iso*,
- $\mathbf{P}$  est à l'extérieur sinon.

Si l'on a pris soin de discrétiser la surface en un certain nombre de points, on procède à une phase de détection de collisions en testant le potentiel des point d'échantillonnage de chaque objet. Une optimisation immédiate est bien sûr l'utilisation de boîtes englobantes pour chaque objet, permettant de ne calculer le potentiel que plus rarement. Une autre optimisation, plus fine, consiste à commencer les tests d'interpénétration sur la graine la plus “pénétrante” du pas de simulation précédent : on a alors de fortes chances de prouver tout de suite qu'il y a toujours pénétration. On dresse ainsi, et pour chaque objet, une liste des objets en contact en un temps  $O(n)$ , où  $n$  est le nombre de graines.

### Traitement des collisions

On dispose maintenant pour chaque objet d'une liste de solides en interaction. Il faut alors déformer les objets de manière à éviter les interpénétrations. Cette déformation doit tenir compte de la raideur relative des objets dans la zone de contact, de façon à ce que des forces de compression opposées soient appliquées de part et d'autre de leur surface (loi d'action-réaction).

### Déformation dans la zone d'interpénétration

On a vu au paragraphe 3.6.2 que la fonction potentiel permet de stocker non seulement les informations géométriques, mais aussi le modèle physique de l'objet. Ces fonctions vont ainsi nous servir pour créer de manière physique le contact, *en ajoutant un potentiel négatif dans les zones d'interaction de façon à comprimer le matériau.*

Supposons que deux objets  $S_1$  et  $S_2$  s'interpénètrent localement. Cherchons alors la valeur des potentiels de déformation  $g_{12}$  et  $g_{21}$  à ajouter aux potentiels  $f_1$  et  $f_2$  dans la zone d'interpénétration, de façon à créer une surface de contact entre les deux objets. On veut que, dans cette zone d'interpénétration, on ait comme équation de la surface de  $S_1$ :

$$f_1(\mathbf{P}) + g_{21}(\mathbf{P}) = iso \quad (3.7)$$

et :

$$f_2(\mathbf{P}) + g_{12}(\mathbf{P}) = iso \quad (3.8)$$

pour la surface de  $S_2$ . Le choix :

$$g_{12}(\mathbf{P}) = iso - f_1(\mathbf{P})$$

$$g_{21}(\mathbf{P}) = iso - f_2(\mathbf{P})$$

convient alors parfaitement : on a en effet une surface de contact définie par  $f_1(\mathbf{P}) = f_2(\mathbf{P})$  et cette surface se raccorde au reste de l'objet puisque pour tous les points  $\mathbf{P}$  vérifiant :  $f_1(\mathbf{P}) = f_2(\mathbf{P}) = iso$ , on a bien  $g_{12}(\mathbf{P}) = g_{21}(\mathbf{P}) = 0$ .

De plus, la relation 3.6 nous permet d'écrire, pour un point  $P$  à la frontière entre les objets  $S_1$  et  $S_2$  :

$$\mathbf{R}_1(\mathbf{P}) = -g_{21}(\mathbf{P})\mathbf{n}_1(\mathbf{P})$$

$$\mathbf{R}_2(\mathbf{P}) = -g_{12}(\mathbf{P})\mathbf{n}_2(\mathbf{P}).$$

Or  $\mathbf{P}$  est sur la surface de contact, donc  $\mathbf{n}_1(\mathbf{P}) = -\mathbf{n}_2(\mathbf{P})$  et  $g_{21}(\mathbf{P}) = g_{12}(\mathbf{P})$ . On vérifie bien :  $\mathbf{R}_1(\mathbf{P}) = -\mathbf{R}_2(\mathbf{P})$ . L'expression des potentiels de déformation est donc conforme au principe d'action-réaction.

Dans la pratique, connaissant la liste des objets en interaction, il est très simple de parcourir les points d'échantillonnage de ces objets et de les faire se déplacer sur la nouvelle surface, en tenant compte des potentiels de déformation. Le calcul des forces de réaction peut alors être fait dans un même temps, grâce à la relation 3.6. Si l'on multiplie chacune de ces forces par un petit élément de surface  $ds$  (représentant la finesse de l'échantillonnage), on profite des points d'échantillonnage à la surface pour faire une *intégration discrète* de ces forces. Les phases de déformation et de calcul des réactions sont confondues.

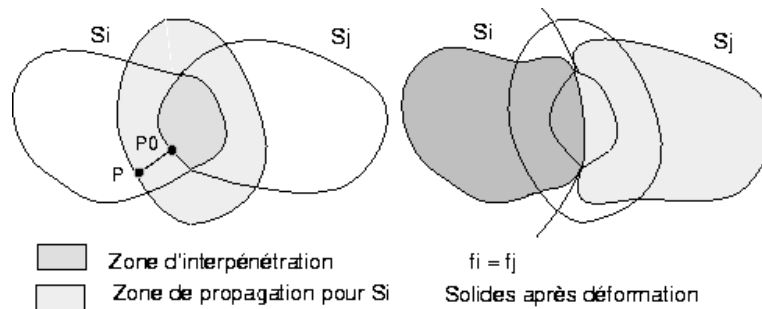


FIG. 3.14 - *Objet avant et après traitement*

### Propagation de la déformation

Une fois la zone d'interpénétration traitée, il convient, pour plus de réalisme, de déformer aussi les alentours de manière à ce que la forme de l'objet reste lisse. En utilisant encore un potentiel de déformation, mais cette fois positif pour simuler une dilatation, on peut modéliser une propagation qui s'atténue avec la distance à la zone de contact, sans pour autant que le mouvement ne soit affecté par cette modification locale. Cela permet une "compensation" de la compression à moindre coût, et ne requiert que peu de paramètres : l'utilisateur ne doit préciser que la taille de la zone de propagation et l'amplitude de la déformation par rapport à la

compression. On traite ainsi la propagation des déformations par un simple *artifice géométrique* : c'est en cela que l'approche de [Gas93] peut être taxée de *globale* selon la nomenclature utilisée au chapitre 2. Pour plus de détails sur le choix de la fonction d'atténuation, voir l'article [Gas93]. Un article récent propose une extension de cette fonction, simplifiant son usage [OCG97].

### Chocs avec des objets implicites rigides

Le formalisme implicite permet d'étendre la simulation à des interactions entre des objets élastiques et des objets rigides. Ainsi, si un objet mou  $S_1$  entre en collision avec un objet rigide  $S_2$ ,  $S_1$  doit manifestement prendre localement la forme de  $S_2$ . Ce qui signifie que les points  $\mathbf{P}$  vérifiant  $f_1(P) + g_{21}(\mathbf{P}) = iso$  doivent aussi vérifier  $f_2(\mathbf{P}) = iso$ . La solution :

$$g_{21}(\mathbf{P}) = (iso - f_2(\mathbf{P})) + (iso - f_1(\mathbf{P}))$$

convient alors pour traiter ce cas particulier.

#### 3.6.5 Intérêts et limitations d'un tel modèle

Ce modèle est une approche originale pour modéliser des objets déformables élastiques. Il approxime le comportement élastique [CG97] de façon simple et efficace. Son formalisme implicite permet des formulations simples et intuitives de la géométrie et des caractéristiques physiques d'un objet, et assure en complément un contact surfacique exact entre objets. Il permet des collisions même multiples et en calcule les forces de réaction intégrées sur les surfaces de contact.

Cependant, l'approximation faite prend en compte seulement les déformations radiales, limitant les capacités de réalisme à des déformations simples ou superficielles. C'est la principale contre-partie de cette formulation très simple et directe. En cela, on ne peut à proprement parler de modèle de matière déformable : les transferts de matière sont superficiellement traités, ce qui amène à classer encore une fois cette méthode parmi les approches globales. Quant à la façon dont sont traitées les collisions, elle peut être qualifiée de méthode de pénalisation étendue, puisqu'à la place d'un seul ressort entre deux objets s'interpénétrant, une série de points de discrétisation fourniront chacun une force de répulsion.

Par contre, un avantage important par rapport aux modèles physiques déformables vus au chapitre 2 est l'aspect visuel. Quelque soit la finesse de discrétisation utilisée pour les calculs de déformation, le rendu final est fin et réaliste grâce à la formulation implicite utilisée pour la modélisation surfacique.

## 3.7 Discussion

### 3.7.1 Implicite vs. Paramétrique

Les surfaces implicites, dont nous venons de passer rapidement en revue les techniques existantes, offrent des avantages certains sur leurs homologues paramétrées. Les splines, qui sont les surfaces paramétriques les plus utilisées en synthèse d'images, ne peuvent par exemple pas représenter de façon simple des formes canoniques comme une sphère, un cylindre, etc... Plus généralement, force est de constater que les surfaces implicites définissent un mode de modélisation très attrayant par bien des aspects :

- **Modélisation intuitive** : grâce aux divers types de potentiel existants, et en particulier aux potentiels engendrés par squelettes, la modélisation devient rapide et intuitive.

- **Stockage réduit** : la modélisation se faisant souvent avec très peu de squelettes, le coût de stockage du modèle est habituellement faible.
- **Définition d'une fonction caractérisant l'intérieur** : contrairement aux surfaces paramétrées, il est aisé de déterminer si un point est à l'intérieur ou hors d'un objet implicite, ce qui accélère à la fois la détection des collisions et le rendu par lancer de rayons.
- **Formulation indépendante de la topologie** : les surfaces implicites peuvent engendrer des objets de topologie quelconque sans que leur définition ne soit en quoi que ce soit plus complexe, alors que les surfaces paramétriques posent de délicats problèmes de recollements dès que des embranchements ou des trous sont présents dans l'objet à modéliser.

Les deux premiers avantages sont bien illustrés par l'exemple du canard de la figure 3.3. Avec seulement deux points pour le corps, un point pour la tête et deux segments pour le bec et la queue, on construit un canard. Le gain de stockage de l'objet par rapport à l'utilisation de splines ou autres est, il va sans dire, considérable, puisque seules les positions des squelettes et les paramètres des fonctions potentiels ont besoin d'être sauvegardés. Quant aux deux derniers avantages, il est aisé de voir que cela permet, principalement en animation, des gains de temps importants.

### 3.7.2 Adéquation à l'animation d'objets hautement déformables

Ce chapitre présente aussi une approche originale pour modéliser des objets déformables élastiques. Même si ce modèle ne peut convenir à la modélisation de matière très déformable de par son manque de réalisme physique, il semble que cette méthode réponde fort bien aux critères que nous nous étions fixés au départ. Il est en effet l'unique modèle décrit permettant une détection rapide des collisions sans problème d'interpénétration, et un rendu de qualité grâce à sa formulation implicite. Et dans la mesure où les surfaces implicites s'adaptent bien à la définition d'objets qui peuvent changer de topologie, une généralisation de ce modèle semble particulièrement indiquée pour la simulation de matière hautement déformable.

## Deuxième partie

# Premier modèle stratifié d'objets très déformables



## Chapitre 4

# Mise au point d'une approche générique

Le but de cette thèse est d'établir un modèle simulant des matériaux hautement déformables en images de synthèse. Nous proposons, dans ce chapitre, de préciser tout d'abord les propriétés caractéristiques des matériaux que l'on souhaite restituer à la simulation. Puis, à la lumière de l'état de l'art, une approche générique sera formulée, nous permettant d'aborder ce problème difficile en combinant des modèles plus simples.

### 4.1 Positionnement du problème

Nous proposons ici de commencer notre étude empiriquement, en définissant exactement le type de matériau que l'on souhaite simuler et en observant ses principales caractéristiques et propriétés, afin de mieux cibler le problème que l'on cherche à résoudre.

#### 4.1.1 Que signifient *déformable* et *hautement déformable*?

On entend par matériau déformable un matériau susceptible de changer de forme en fonction des forces extérieures qui lui sont appliquées. Cette définition très générale couvre la quasi totalité des objets qui nous environnent. En effet, les matériaux dits solides ne sont en fait qu'un cas de matériau très peu déformable à notre échelle. Une table en bois par exemple sera souvent qualifiée d'indéformable, alors qu'il suffit de frapper sur la table pour entendre une onde sonore en résulter, prouvant qu'une déformation a bien eu lieu.

D'autres objets, moins litigieux, sont clairement déformables et qualifiés comme tels dans le langage courant : un élastique, une balle en plastique, un ballon de baudruche, ou tout simplement un coussin en mousse se déforment aisément lors de manipulations. On donne aussi quelque fois le qualificatif de "mou" pour ce type d'objets. Dans une utilisation normale, ces objets conservent une même topologie au cours des déformations : leur structure ne change pas, et les relations de voisinage au sein de la matière sont préservées.

Certains objets peuvent même être tellement malléables que l'on peut les modeler, les mélanger en les repliant sur eux-mêmes comme par exemple de la pâte à modeler. Nous retrouvons aussi dans cette catégorie la pâte à pain, ou le dentifrice. Les fluides peuvent aussi être mis dans cette catégorie, mais cette fois, ils n'ont pas du tout de forme propre : posés sur un sol, ils s'éta-



leront sous forme de flaques. Ces objets, allant de la pâte malléable aux fluides visqueux seront qualifiés dans ce document de **matériaux hautement déformables** : cette fois, deux éléments de matière voisins à un moment peuvent être totalement séparés par une forte déformation. Ces substances peuvent donc se séparer en plusieurs morceaux, et se recombinaison.

#### 4.1.2 Comportement macroscopique des matériaux déformables

Après avoir vu la signification intuitive des termes déformables et très déformables, examinons comment les physiciens ont appréhendé ces phénomènes macroscopiques. Tout d'abord, l'observation scientifique a amené à définir deux comportements antagonistes des objets déformables :

- Un matériau est dit **élastique** s'il retourne à sa forme et sa taille originales après compression. Sa forme courante dépend alors uniquement des forces auxquels il est actuellement soumis.
- Un matériau est dit **plastique** parfait si il ne montre aucune tendance à revenir à sa forme et sa taille originales après une déformation. Sa forme courante dépend cette fois de l'historique des forces qu'il a subi.

Puis, des séries répétées et coïncidentes de tests sur différents matériaux ont amené à remarquer des comportements généraux, plus fins. Par exemple, la loi empirique de Hooke stipule que tout corps s'allonge proportionnellement à l'amplitude de la force l'étirant, jusqu'à un certain seuil appelé *limite de proportionnalité*. De même, une autre loi empirique montre qu'un matériau reprend toujours sa forme et sa taille initiales pour des forces inférieures à une valeur limite, appelée *limite d'élasticité*. Il a aussi été remarqué qu'il existe un *seuil de plasticité* au dessus duquel un matériau "cède" et connaît une déformation importante de sa structure interne. Enfin, si on augmente encore la pression exercée, on atteint souvent un *seuil de rupture* où la matière, trop sollicitée, casse purement et simplement.

Ces caractéristiques sont qualitativement les mêmes pour presque tous les matériaux, mais c'est bien sûr quantitativement, en fonction des valeurs des seuils, que l'on peut distinguer les innombrables types de matériaux.

#### Loi de contrainte/déplacement d'un matériau déformable

Au vu des observations précédentes, il est commun de définir un matériau déformable macroscopiquement par une courbe, appelée *loi de contrainte/déplacement*, qui permet de connaître quantitativement les *déformations relatives* d'un objet en fonction de la *pression* (force par unité de surface) qu'on y exerce. L'allure générale d'une telle courbe est représentée sur la figure 4.1 ; on y trouve alors les observations citées précédemment.

En effet, le point *A* représente la limite de proportionnalité, puisque la partie avant ce point est bien linéaire. Le point *B* indique la limite d'élasticité, c'est-à-dire qu'une pression plus forte engendrera des dommages à la forme et la taille du matériau : dans ce cas, même après arrêt de la pression, l'objet ne reviendra pas à sa forme initiale. À partir du point *C*, la structure interne du matériau commence à subir des changements importants, ce qui explique que les déformations commencent à largement s'amplifier à partir de ce point. Enfin, le point *E* marque le seuil de rupture du matériau, qui ne peut supporter plus de contraintes. Dans la zone plastique, entre *B*

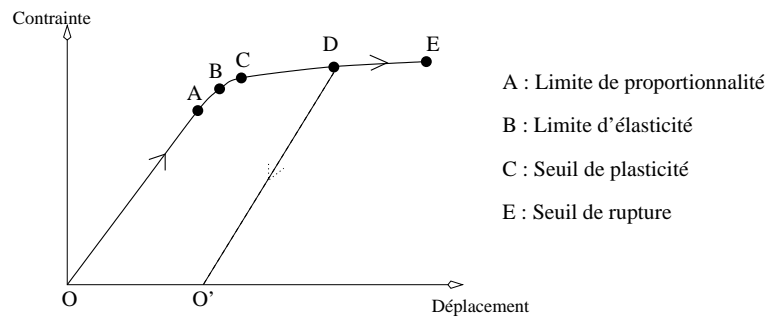


FIG. 4.1 - Une loi typique de contrainte/déplacement, et ses seuils.

et  $E$ , si on relâche la pression exercée en  $D$  par exemple, l'objet reviendra à une position stable  $O'$  selon les pointillés, conservant donc une déformation irréversible.

En pratique, ces lois ont des formes très variables, respectant cependant cette structure. Par exemple, dans le cas du verre, les points  $A$ ,  $B$ ,  $C$  et  $E$  sont confondus : le verre est élastique jusqu'à un certain point, où il casse.

### Justification du comportement à l'échelle microscopique

Ces lois de contrainte/déplacement modélisant le comportement macroscopique d'un objet s'expliquent relativement bien quand on descend à la structure moléculaire. Tout d'abord, des mesures moyennes des forces inter-particulaires entre deux atomes ont montré une forme caractéristique, dont la force de Lennard-Jones (voir paragraphe 2.2.1) est un bon représentant. Ces forces représentent le bilan d'un ensemble d'interactions d'ordre atomique aussi bien électriques que quantiques. Elles exhibent à l'échelle moléculaire une zone linéaire autour de la position d'équilibre (figure 4.2), qui explique la loi macroscopique de Hooke. En effet, la force de réaction entre chaque paire d'atomes sera proportionnelle à la compression. Par contre, passée une certaine valeur de compression, la réponse ne sera plus du tout linéaire.

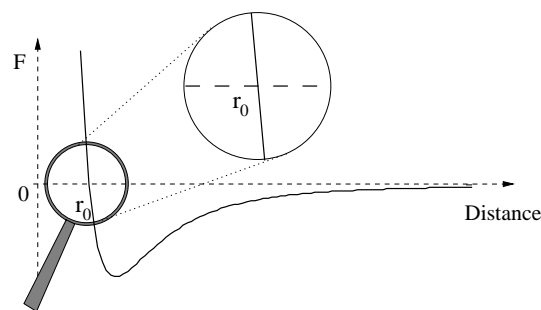


FIG. 4.2 - Zone linéaire de la loi de Lennard-Jones, autour de la position d'équilibre

Dans ces cas de forte compression, la structure cristalline de la matière sera alors affectée, et si par exemple la matière est organisée par couches d'atomes, ces couches vont glisser les unes par rapport aux autres, créant des changements irréversibles au sein de l'objet.

### 4.1.3 Comportement des objets hautement déformables

Quant aux objets hautement déformables, ils sont encore à part puisque même leurs lois de contrainte/déplacement sont particulières : du fait d'une faible cohésion interne, l'objet peut être déformé quasiment autant que l'on souhaite. De plus, de tels matériaux se prêtent très mal à des mesures de déformation : ils n'ont souvent même pas de forme au repos, et s'écoulent facilement. Il est par conséquent très difficile d'établir des critères quantitatifs que devra vérifier la matière à simuler : sa nature souple et ultra malléable sort du cadre physique bien établi. Seuls les critères qualitatifs sont clairs : ces substances peuvent se déformer, changer de forme au repos, se couper en morceaux, voire s'écouler.

L'animation de synthèse doit donc créer son propre modèle de matériaux très déformables, dont la validation sera principalement visuelle. On souhaitera ainsi animer de la lave, des avalanches de neiges, des coulées de boue, ou de la pâte à modeler. Il s'agira de trouver un compromis entre réalisme visuel, réalisme physique, et efficacité. Or, l'état de l'art des techniques d'animation traditionnelles vu au chapitre 2 montre à quel point les difficultés que soulèvent la modélisation et l'animation de substances très malléables sont complexes. Tous les problèmes classiques tels que la conservation du volume au cours du temps ou la gestion robuste des changements de topologie deviennent cruciaux dans ce cas précis, puisque les déformations peuvent être très importantes. La mise au point d'un modèle d'animation ad hoc et unique qui pourrait répondre à nos attentes paraît aussi difficile que peu pratique : une approche plus générale est nécessaire.

## 4.2 Vers un modèle stratifié

Les deux principaux critères souhaitables pour notre modèle sont la *généricité* et l'*efficacité*. Il s'agit donc de trouver une approche combinant ces deux qualités, en gardant à l'esprit le fait qu'un animateur doit pouvoir utiliser ce modèle très intuitivement et aisément.

### 4.2.1 Modèles minimalistes

Il y a bien des façons d'appréhender le monde qui nous entoure. Parmi les approches possibles, certains physiciens ont souvent cherché à définir un modèle *générique* global, qui pourrait modéliser quantitativement la réalité à toutes ses échelles. En connaissant le fonctionnement de l'infiniment petit, on peut ainsi connaître le comportement de toute chose. Cette modélisation, que l'on peut qualifier de *minimaliste*, a l'immense avantage d'être intellectuellement satisfaisante dans la mesure où des règles simples régissent entièrement et précisément l'évolution de tout objet animé.

Une telle modélisation a cependant un défaut majeur : même si le modèle est parfaitement valide, il n'est absolument pas utilisable en terme de simulation. D'une part, il demanderait bien trop de calculs pour le moindre mouvement, mais de plus, il devient quasiment impossible de paramétrer intuitivement un comportement souhaité. Toutes les caractéristiques visibles d'un objet, forme et mouvement, *émergeront* de sa composition et de son fonctionnement à l'échelle microscopique [Luc89]. Il sera alors difficile à un animateur de donner naissance à un comportement désiré. En animation de synthèse, un tel modèle minimaliste correspondrait à vouloir tout modéliser au moyen de particules en interaction. Régler un comportement global, comme

la conservation du volume, en manipulant les paramètres des forces fixes de type Lennard-Jones nous semble tout à fait impraticable.

La synthèse de mouvement et de déformations doit en fait obéir à une autre contrainte : simuler le plus rapidement possible des mouvements réalistes aisément paramétrables par un animateur, à savoir à l'aide de paramètres *macroscopiques*.

### 4.2.2 Modèles hiérarchiques

L'approche la plus pragmatique en physique est de trouver une modélisation simple d'un phénomène qui approche quantitativement le mieux une série d'observations effectuées. On crée ainsi un modèle, valide uniquement à une échelle donnée, qui approche macroscopiquement tous les phénomènes d'ordre microscopique<sup>1</sup>. Si pour chaque échelle d'observation, on construit un modèle approprié, le plus simple possible, on obtient ainsi une *hiérarchie* de modèles simples, aisément paramétrables.

L'animation en synthèse d'images se doit de procéder de la même façon. Pour simuler un diapason par exemple, il ne viendrait pas à l'esprit de simuler le comportement quantique des molécules mises en jeu. Ne serait-ce que pour des raisons d'efficacité, il vaut bien mieux simuler l'aspect macroscopique du phénomène, c'est-à-dire des oscillations régulières amorties à 440 Hertz. Par contre, si le diapason est susceptible de se casser en morceaux en cours d'animation, il faudra adapter le modèle pour qu'il puisse gérer un tel événement. Toute la difficulté est d'utiliser le modèle valide le plus simple (et donc le plus *efficace*) à chaque domaine d'utilisation : c'est la notion de *niveau de détail*, idée encore très peu mise en pratique en animation [CH97]. Cela oblige en effet à changer souvent de modèle en cours d'animation, ce qui n'est pas toujours facile sans créer de discontinuités visuelles.

### 4.2.3 Modèles à couches

Un moyen agréable pour éviter d'avoir à changer constamment de modèle tout en conservant l'idée d'échelle de validité est d'utiliser un modèle à couches. On fait ainsi coexister différents niveaux de modélisation dans un seul et même modèle global. Chaque couche a un rôle précis pour une échelle donnée, et est optimisée pour accomplir cette tâche spécifique. C'est ce que Demetri Terzopoulos [TPBF87, TW88], John Chadwick [CHP89] ou Marie-Paule Gascuel [GVP91] font en décomposant leur modèle en un système interne modélisant les déformations principales, complété par un système optimisé pour les déformations locales. Cette façon de faire combine la généralité et l'efficacité ; elle est de plus très intuitive pour un animateur. On peut en effet penser à faire un ballon, constitué d'une couche interne dont la trajectoire est spécifiée par positions clés, permettant à l'animateur de désigner où le ballon doit aller exactement. Une couche déformable "par dessus" la couche interne pourra gérer les déformations locales lors des rebonds, que l'animateur pourra régler indépendamment du modèle interne. Les paramètres, qui se rattachent à des échelles intuitives et distinctes, gagnent ainsi en simplicité.

---

1. Le modèle physique est en fait une approximation de toutes les interactions d'échelles différentes, et pas seulement microscopiques. Par exemple, définir la gravité comme une force constante est une simplification tout à fait valable à l'échelle humaine, mais la mécanique céleste dont elle résulte devra être modélisée plus finement à l'échelle planétaire.

#### 4.2.4 Découpage en couches indépendantes

Ceci nous amène à réfléchir sur la façon dont on peut découper un système donné en une série de modèles indépendants, ou tout au moins en interactions simples. C'est ce que s'est attaché à faire Arash Habibi dans sa thèse [Hab97]. En partant d'analogies entre mécanique et électricité, il s'est penché sur les caractéristiques que doivent vérifier un système pour pouvoir être découpé en deux sous-systèmes quasi-indépendants. Les résultats sont rassurants : deux phénomènes sont d'autant plus simulables indépendamment que l'impédance de l'un est très inférieure à l'impédance de l'autre. Ce qui, traduit en termes physiques, signifie que l'action de l'un sur l'autre est faible. Par exemple, les vêtements n'influent en principe en rien le mouvement de la personne qui les porte. La simulation du vêtement peut donc être faite "par-dessus" celle du mouvement de la personne, de façon indépendante.

Il n'est cependant pas clair qu'un objet hautement déformable puisse être découpé en de tels sous-systèmes indépendants. Ce type d'objet étant en général de consistance uniforme, on ne peut pas, à proprement parler, distinguer différents ensembles le composant. De toute façon, aucun outil automatique, capable d'extraire un ensemble de sous-systèmes équivalents, n'existe : le découpage doit donc être fait de manière intuitive, en des sous-systèmes interagissant entre eux si nécessaire.

#### 4.2.5 À la lumière de l'état de l'art

Comme nous l'avons vu lors de l'état de l'art sur l'animation, le modèle physique le mieux adapté à la modélisation d'objets hautement déformables semble être le système de particules. Pourtant, il est clair que ce modèle est très limité aussi bien du point de vue qualité visuelle que du point de vue maniabilité pour l'animateur. Les particules semblent cependant incontournables dans un premier temps pour qui souhaite simuler des matériaux capables de se séparer et/ou de fusionner aisément.

Elles peuvent par contre gagner en intérêt si elles font parties d'un modèle hybride, à couches. Reste encore à trouver les modèles qui permettront d'"habiller" le système particulaire, afin d'obtenir un modèle complet et cohérent, que l'on qualifiera de *stratifié* pour sa composition en couches inter-agissantes. C'est cette idée de modèle composite que nous nous proposons de développer.

### 4.3 Proposition d'un modèle hybride à trois couches

Le découpage proposé par Jimenez et Luciani, vu au paragraphe 2.2.1, est intéressant de par sa généralité. Intuitivement, il est en effet assez général de penser qu'un modèle déformable est constitué d'un "noyau", enrobé du "derme" et recouvert par un "épiderme". Cette décomposition en trois couches semble convenir particulièrement aux objets hautement déformables.

Nous avons donc choisi cette décomposition comme base de notre modélisation, même si l'appellation de chaque couche change pour mieux coller à leur rôle respectif. Nous allons détailler chaque niveau en insistant sur la généralité et la possibilité d'avoir des modèles performants pour chaque couche, permettant d'aboutir à un modèle global optimisé.

### 4.3.1 *L'Ossature* : modèle interne

La première couche de notre modèle représente l'*ossature*, c'est-à-dire la couche profonde. C'est le composant essentiel puisque qu'il définit réellement les mouvements globaux de l'objet simulé. Selon les désirs de l'animateur, ce modèle pourra être totalement décrit à la main (à l'aide de positions clés par exemple), ou pourra générer son propre mouvement en fonction des forces extérieures.

De même, selon la nature de la matière à simuler, l'*ossature* pourra aussi bien être un modèle très simple ou assez complexe : simuler un objet hautement déformable nécessitera un modèle capable de gérer de très fortes déformations, ce qui ne sera pas du tout nécessaire pour un objet élastique par exemple. La nature de ce modèle peut donc aller de la spécification de positions clés jusqu'au système de particules en passant, par exemple, par une arborescence d'objets articulés : tout système générant du mouvement est valable. Il suffit, pour optimiser les calculs, de choisir le modèle suffisant pour créer les mouvements désirés. Par le biais de ce choix, l'animateur pourra donc contrôler le mouvement global de l'objet animé.

### 4.3.2 *La Chair* : habillage déformable

Le modèle interne étant optimisé pour gérer le mouvement général et les fortes déformations si nécessaire, il s'agit maintenant de le couvrir, de l'enrober de matière déformable, qui correspond en quelque sorte à la *chair* de l'objet. Cette couche de matière déformable confèrera à l'*ossature* un véritable volume de matière (ce qui n'était pas forcément le cas de l'*ossature* seule : le modèle interne peut fort bien être ponctuel) comme la chair humaine le fait pour le squelette osseux.

Cette deuxième couche devra gérer les déformations locales survenant au cours du mouvement, tout en communiquant les forces mises en jeu à l'*ossature*, susceptible de se déformer globalement en conséquence. Ici encore, un modèle bien adapté est nécessaire pour à la fois remplir le rôle souhaité et optimiser les calculs : une approche globale, faisant de fortes approximations sur les déformations, peut tout à fait suffire. L'animateur devra aussi pouvoir agir simplement sur les caractéristiques d'élasticité de cette couche par exemple, ou encore sur sa capacité de déformation à volume constant.

### 4.3.3 *La Peau* : interface extérieure

La *peau*, fine pellicule entre l'extérieur et l'intérieur, est la seule partie visible d'un modèle déformable. Cette couche, véritable interface entre le monde extérieur et l'objet, représente le contour de l'objet. Elle définit la représentation de l'objet, l'aspect extérieur, puisque seule cette peau du modèle apparaît. Ses attributs seront principalement visuels, comme la couleur ou les textures qui y seront plaquées.

Elle ne se limite pas pour autant à un rôle purement géométrique : nombreux sont les exemples où la peau, munie d'une certaine élasticité, influence vraiment la forme et l'état d'un objet. La tension de surface de l'eau, responsable de la forme d'une goutte sur une surface plane, en est un bon exemple.

Une dernière tâche importante lui incombe, due à son rôle d'interface : la détection des collisions avec le monde extérieur, phase de calcul souvent critique comme nous l'avons vu au paragraphe 2.3.1. Tous ces rôles devront être assurés de façon performante par le modèle, qui devra toujours être aisément paramétrable par un animateur.

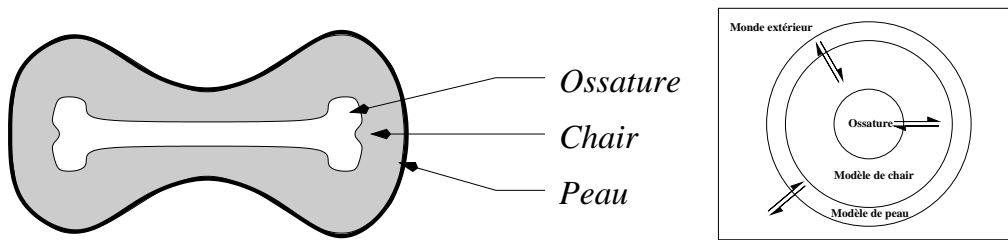


FIG. 4.3 - Schéma des couches et de leurs interactions.

#### 4.3.4 Échanges entre couches

Une fois ces couches définies, il faut regarder de plus près leur interdépendance. Comme le montre la figure 4.3, chaque couche est définie relativement à la couche inférieure : la chair enrobe l'ossature, et la peau recouvre la chair. Mais les relations entre couches ne doivent pas se limiter à cette dépendance à sens unique : les échanges entre les couches sont à chaque fois bilatéraux. Les différentes couches ne peuvent, sauf cas très précis, être simulées indépendamment. Les échanges peuvent se résumer comme suit :

- La chair s'appuyant sur l'ossature, cette dernière devra communiquer tout changement de forme et de position à la chair.
- À l'inverse, la chair devra propager les forces de compression qui lui sont appliquées (dus par exemple aux collisions) afin que la couche interne bouge en conséquence.
- La chair indiquera à la peau chacune de ses déformations afin que la peau épouse toujours sa forme.
- Inversement, la peau devra, dès qu'une collision sera détectée, en informer la chair de façon à ce que cette collision soit traitée correctement par la chair.
- Enfin, le monde extérieur agira sur la peau et la peau devra agir, en réponse, sur le monde extérieur.

Cette bi-latéralité des communications entre couches est importante, et c'est d'ailleurs ce qui manquait principalement aux modèles vus dans l'état de l'art au paragraphe 2.1.4.

#### 4.3.5 Validité et efficacité

Le découpage en trois couches spécifiques peut s'analyser comme une *décomposition multi-échelles simplifiée*. La décomposition d'un phénomène à plusieurs échelles procure en effet une description hiérarchique des variations de comportement sur des régions de taille variable, c'est-à-dire une décomposition du comportement en espace et en temps. Or, dans le cas de matières déformables homogènes, les fréquences d'un stimulus vont correspondre à des régions spatiales déterminées à cause de l'absorption progressive du choc par la matière : les fréquences élevées ne dépasseront pas une couche superficielle du matériau, les moyennes pénétreront sur une couche plus importante, alors que les basses, correspondant au stimulus filtré par la matière en périphérie, n'atteindront que le cœur de l'objet stimulé.

Les trois couches hiérarchiques concernent justement trois échelles de fréquences. L'ossature, par exemple, ne s'occupera que des basses fréquences, celles qui pénètrent vraiment la matière et définissent la forme globale de l'objet animé. La chair, quant à elle, traitera les moyennes fréquences, à effets plus localisés. La peau, enfin, sera plus apte à gérer les hautes fréquences des stimuli extérieurs, en se ridant localement par exemple. Ce découpage en couches permet dès lors de simuler chaque phénomène avec un modèle *optimisé pour chaque échelle*. Le nombre de couches, à savoir trois, semble suffire au travers des tests effectués pour décrire bon nombre de types d'objets en animation, comme nous le montrerons au chapitre suivant.

#### 4.3.6 Généricité

Ce découpage en trois couches n'est qu'un *cadre de travail*: du choix des modèles utilisés pour chacune des couches découlera une modélisation efficace de tel ou tel type d'objet. C'est un peu un *jeu de construction*, dont l'efficacité du modèle complet dépend du choix judicieux des pièces de l'assemblage. C'est en cela que notre modèle peut être qualifié de générique.





## Chapitre 5

# Un premier modèle de matériaux hautement déformables

**F**orts de l'état de l'art et de l'analyse du problème abordé, nous proposons dans ce chapitre de définir un premier modèle découlant des principes vus au chapitre précédent. En combinant des modèles adéquats, nous définissons un premier modèle cumulant les avantages acquis pour l'animation d'objets hautement déformables [CGD97]. Aussi, ce chapitre montre qu'un bon nombre des problèmes évoqués dans l'état de l'art de l'animation au chapitre 2 trouvent une solution élégante grâce à l'usage combiné des surfaces implicites et des particules. Bien sûr, différents problèmes liés à la combinaison d'une surface implicite et d'une structure interne déformable doivent être réglés. En premier lieu, travailler avec des surfaces implicites, donc sans formulation explicite, nous oblige à maintenir à tout instant de la simulation un bon échantillonnage de la surface [DTG95, DTG96]. Le problème du mélange entre potentiels et des variations de volume que cela entraîne au cours du mouvement sont aussi des points délicats [DG94, DG95]. Les paragraphes suivants exposent les solutions que nous avons proposées pour éliminer ces différents problèmes sans pour autant perdre les avantages intrinsèques des surfaces implicites.

### 5.1 Une ossature définie par un système de particules

La conclusion du chapitre 2 montrait bien à quel point les systèmes de particules semblent l'approche la plus appropriée pour la modélisation de matériaux hautement déformables. Nul autre modèle permet en effet une si large gamme de comportements élastico-plastiques, tout en traitant aisément les changements de topologie.

#### 5.1.1 Choix d'une formulation des lois d'interaction

Pour donner à nos objets des comportements inélastiques, nous avons dans un premier temps repris les approches de David Tonnesen [Ton91] et de Stéphane Jimenez [Jim93]. Cela consiste à considérer un ensemble de particules munies de lois d'interaction de type Lennard-Jones. La formulation que nous avons retenue pour la force qu'exerce une particule  $S_1$  sur une particule  $S_2$  est :

$$\mathbf{F} = K \cdot \left( \frac{r_0^n}{r^{n+1}} - \frac{r_0^m}{r^{m+1}} \right) \frac{\mathbf{r}}{r} \quad (5.1)$$

où  $\mathbf{r}$  est la distance entre  $S_1$  et  $S_2$ , et  $r = \|\mathbf{r}\|$ . Le paramètre  $K$  représente la raideur du matériau, et  $r_0$  la distance “au repos” entre deux particules (voir figure 5.1). On remarque en effet que pour  $r = r_0$ , la force inter-particulaire est nulle. C’est donc la distance à laquelle les particules tendent à demeurer. La signification du paramètre  $K$  est aussi assez intuitive puisqu’il est clair que plus  $K$  est élevé, plus la force qui tend à maintenir deux particules à une distance  $r_0$  est grande. On aura dans ce cas un système de particules peu malléable, puisque très cohérent. Quant aux paramètres  $n$  et  $m$ , leur rôle est plus délicat à définir. De nombreux auteurs utilisent  $n = 2m$ , avec  $n = 4, 6$  ou  $8$ . Ces valeurs permettent en effet de définir des lois d’interaction conduisant à des comportements adéquats.

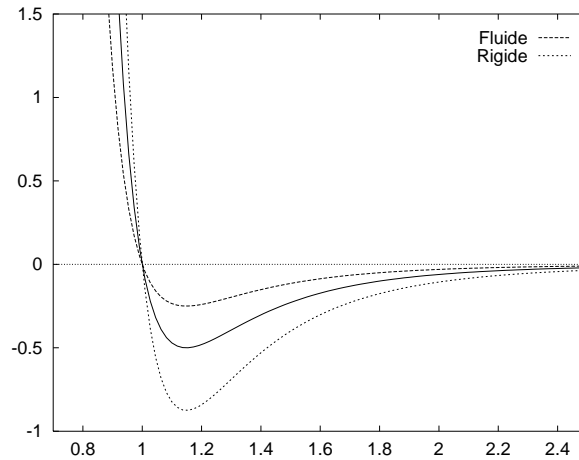


FIG. 5.1 - Lois d’interaction pour différents  $K$  ( $r_0 = 1$ )

Cependant, comme on l’a vu au paragraphe 2.1.2, intégrer ces forces ne conduisent généralement pas à un système stable. Si le pas d’intégration est trop grand, deux particules peuvent se rapprocher trop près, créant au pas suivant une force gigantesque. De toute façon, même lorsque le système ne diverge pas, les particules oscillent indéfiniment autour de leur position d’équilibre, puisqu’aucune force dissipative n’a encore été modélisée.

### 5.1.2 Mise au point d’un frottement visqueux

Un moyen de diminuer les défauts de stabilité est de rajouter une force de frottement fluide (type résistance de l’air), simulant l’évolution des particules dans un milieu ambiant et non dans le vide. Cette force, qui s’exerce sur chacune des particules, peut s’écrire :  $\mathbf{f}_r = -b \cdot \mathbf{v}$ , où  $b$  est un coefficient de frottement très faible. Cela permet un amortissement des oscillations, mais des tests sur machines montrent que la stabilisation n’est atteinte que pour des valeurs du pas  $dt$  inférieures à  $10^{-4}$ . Une telle fréquence de calcul n’est pas toujours acceptable. D’autre part, on ne peut pas trop augmenter le coefficient  $b$  : une particule en chute libre pourrait en effet voir son mouvement ralenti de façon peu naturelle, atteignant même une vitesse limite très rapidement.

L’idée d’une autre force de frottement dépendant, cette fois, de la *densité locale de particules* paraît être un palliatif naturel à ces problèmes d’amortissement. Il paraît en effet logique de considérer que le mouvement d’une particule est tributaire de la position et de la vitesse de ces voisins immédiats, qui l’empêchent de se mouvoir dans une direction quelconque comme décrit

par la figure 5.2.

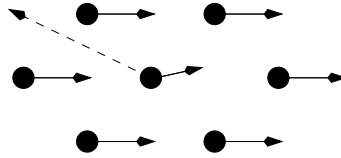


FIG. 5.2 - *Le frottement visqueux freine une particule dont le mouvement (en pointillé) serait contraire au mouvement local.*

Une manière simple pour définir une telle force de frottement, que nous appellerons “frottement visqueux”, consiste à appliquer sur chaque particule une force d’interaction avec chacune des particules environnantes. La somme de ces contributions décroissantes avec la distance dépend ainsi de la densité de particules dans le voisinage. Le frottement visqueux se modélisant en général par une force proportionnelle au carré de la vitesse, on souhaite utiliser pour chaque interaction de frottement le carré de la vitesse relative entre deux particules. En effet, cette force de frottement est due au mouvement relatif d’une particule par rapport à une autre, gênant le déplacement libre des deux. Nous choisissons donc d’exprimer la force de frottement visqueux totale, pour une particule de vitesse  $\mathbf{v}$  qui interagit avec des particules de vitesse  $\mathbf{v}_i$ , par :

$$\mathbf{F} = \sum_i \delta_i(\|S_i - S\|) \|\mathbf{v}_i - \mathbf{v}\| (\mathbf{v}_i - \mathbf{v}).$$

La fonction  $\delta_i$  est prise décroissante, à support compact. Ajuster cette fonction (en particulier son rayon d’action) permet de simuler des comportements plus ou moins visqueux, et offre une bonne stabilisation du système particulaire. Nous avons choisi une fonction dépendant de deux paramètres : le coefficient de frottement visqueux  $\alpha$ , et le rayon d’influence  $R$ . Nous avons fait le choix suivant :

$$\delta_i(r) = \begin{cases} \alpha \cdot \frac{(r-R)^2}{R^2} & \text{Si } r \in [0, R] \\ 0 & \text{Sinon} \end{cases}$$

Des tests 2D, puis 3D ont été effectués, et il s’avère qu’une fonction de ce type apporte amplement satisfaction puisque l’on peut alors utiliser des pas d’intégration de l’ordre de  $10^{-2}$  au lieu des  $10^{-4}$  de départ. Et ceci bien sûr sans les défauts qu’amènent les frottements fluides, comme le ralentissement exagéré en cas de chute libre.

### 5.1.3 Complexité algorithmique du bilan des forces

Pour  $n$  particules, une implémentation directe du bilan des forces entraîne une complexité de  $\mathcal{O}(n^2)$ , puisque l’on calcule pour chaque particule les  $(n - 1)$  forces exercées par les autres. On a vu au chapitre 2 que des méthodes permettent de faire baisser cette complexité. Ces techniques, décrites au paragraphe 2.2.1, exploitent le fait que les lois d’interaction décroissent rapidement avec la distance, et approximent donc la force créée par un ensemble de particules éloignées en un seul calcul. Mais ces approches ne sont réellement efficaces que lorsque la répartition de la matière est non uniforme.

Au contraire, dans notre cas, les particules sont relativement bien réparties au sein du matériau. Par contre, les lois d’interaction décroissent rapidement en fonction de la distance. Nous avons donc choisi de seuiller la fonction d’interaction quand elle devient très faible, de telle sorte qu’elle ait seulement un rayon d’influence  $R$ . On peut alors subdiviser l’espace en cubes de

taille  $R$ ; en assignant chacune des particules au cube où elle est actuellement, le bilan des forces pourra alors se réduire au balayage de voisins, réduisant ce calcul à un ordre de  $\mathcal{O}(n)$ . Ainsi, l'animation est aisément paramétrable et rapide.

## 5.2 Adaptation du modèle implicite élastique

Une fois le système de particules au point, il s'agit de l'enrober d'une couche de matière élastique implicite de façon à obtenir une surface lisse, jouant le rôle d'interface avec l'extérieur, et pouvant assurer des contacts surfaciques exacts. Cela correspond, dans le formalisme défini au chapitre précédent, à la fois à la chair et à la peau.

### 5.2.1 D'une ossature discrète à une surface continue

Nous avons vu que l'un des défauts des systèmes de particules est l'absence de frontière, de contour net autour du matériau. Il nous faut donc définir un moyen de construire aisément une surface qui représente l'état des particules. Cette surface n'est bien sûr pas forcément constituée d'un seul morceau, car les systèmes de particules peuvent très bien modéliser des séparations d'un matériau très déformable en plusieurs morceaux.

Pour y parvenir, l'idée est la suivante: les particules, qui sont dans un premier temps des points de l'espace, peuvent être considérées comme des squelettes d'objets implicites (ou points clés). Une particule seule engendre alors une sphère, et un système entier définit un objet pouvant être connexe ou non, selon la dispersion des particules dans l'espace. Le mélange des potentiels, vu lors de l'état de l'art des surfaces implicites, fait que l'objet ainsi créé est de toute façon lisse, et permet d'associer une surface à notre système de particules (voir figure 5.3). Cette surface pourra servir d'interface avec les autres objets, c'est-à-dire permettre la détection de collision et se déformer en conséquence.

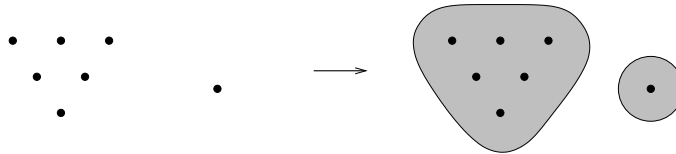


FIG. 5.3 - L'enrobage implicite engendre un contour net et lisse autour des particules.

### 5.2.2 Utilisation du modèle élastique implicite [Gas93]

Pour rendre cette surface localement déformable, le modèle implicite déformable détaillé au paragraphe 3.6 paraît être un excellent choix. En effet, le principal reproche que nous pouvions faire à ce modèle était sa limitation aux déformations radiales élastiques, et l'on ne pouvait alors parler de véritable matière déformable. Mais en tant que couche superficielle autour d'un modèle de matière lui-même déformable comme les particules, l'utilisation de [Gas93] prend tout son sens.

Sa formulation permet effectivement d'enrober les particules en engendrant une surface, mais de plus, ses approximations de l'élasticité lui permettent de gérer les déformations avec des temps de calcul réduits. Enfin, la notion de surface continue est conservée même en cas de collision,

et permet en outre une gestion fine des contacts qui évite tous les problèmes d'interpénétration locale ou de rebonds avant contact qui pouvaient se produire avec les modèles précédents. Il semble donc que ce modèle déformable implicite puisse servir à la fois de chair et de peau.

### 5.2.3 Gestion combinée des couches

#### a) Transmission entre couches

##### De l'ossature à la chair

La méthode d'enrobage par surfaces implicites permet de traduire en terme de surface la disposition des particules de la structure interne : en d'autres termes, la chair implicite s'appuie sur l'ossature et lui procure une surface cohérente. La seule donnée des positions des particules suffit donc à construire une forme, connexe ou non, reflétant l'état du système interne : défini ainsi, l'habillage est purement géométrique.

##### De la chair à l'ossature

La structure externe, chair et peau, est chargée de gérer les interactions avec l'extérieur. Lors d'un choc, elle détecte l'interpénétration, se déforme en conséquence, et doit alors communiquer les forces de contact à la structure interne. C'est ce processus qui permettra un comportement inélastique, puisqu'une fois les forces transmises, le système particulaire réagira en conséquence et sa forme au repos pourra en être affectée.

Si une force externe  $\mathbf{F}$  radiale est appliquée en un point  $\mathbf{P}$  de la surface (vérifiant donc  $\sum_i f_i(\mathbf{P}) = iso$ ), une première solution consiste à distribuer  $\mathbf{F}$  aux particules voisines en utilisant la décomposition du vecteur  $\mathbf{n}$  normal à la surface en  $\mathbf{P}$  en la somme des contributions des différents squelettes. En effet, par définition des surfaces implicites, on a :

$$\mathbf{n}(\mathbf{P}) = \frac{\sum_i \nabla f_i(\mathbf{P})}{\|\sum_i \nabla f_i(\mathbf{P})\|}$$

Ainsi, si l'on nomme  $\mathbf{n}_i$  le vecteur :  $\mathbf{n}_i(\mathbf{P}) = \nabla f_i(\mathbf{P}) / \|\sum_i \nabla f_i(\mathbf{P})\|$ , on peut alors écrire :

$$\mathbf{F} = \|\mathbf{F}\| \mathbf{n}(\mathbf{P}) = \|\mathbf{F}\| \sum_i \mathbf{n}_i(\mathbf{P}) = \sum_i \mathbf{f}_i$$

Cette première méthode permet une très bonne répartition des forces radiales appliquées à l'objet (figure 5.4.a). Cependant, utiliser systématiquement cette répartition alors que des forces sont généralement appliquées sur une zone non négligeable de la surface n'est pas forcément nécessaire. Une solution plus simple a été envisagée.

En se rappelant que la structure interne est constituée d'un système de particules, on peut penser que si une force est appliquée à une particule seulement, le jeu des échanges de forces inter-particulaires *propagera* rapidement cette force dans toute la structure. Transmettre la force de réaction au squelette le plus proche paraît donc suffisant. A posteriori, les essais d'animation effectués en cours de recherche sont convaincants, justifiant pleinement l'utilisation de cette solution simple.

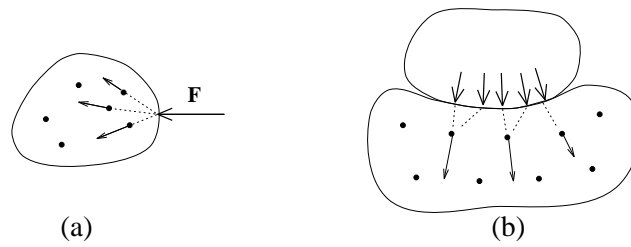


FIG. 5.4 - Deux solutions pour la répartition des forces extérieures.

Ces échanges entre les deux structures nous assurent alors un comportement d'objet hautement déformable. La méthode retenue permet en outre l'utilisation de squelettes quelconques. On peut utiliser des éléments mécaniques non ponctuels, en supposant bien sûr que l'on sait gérer le calcul des forces internes entre de tels éléments.

### b) Problèmes généraux des surfaces implicites

Une fois les échanges entre nos deux sous-modèles réglés, il demeure encore un certain nombre de points à aborder pour rendre le modèle complet utilisable. En effet, si l'emploi des surfaces implicites apporte d'incontestables avantages, des problèmes spécifiques découlant de la formulation implicite de la surface restent à résoudre. En premier lieu, maintenir une discrétisation surfacique d'un objet en cours de déformation n'est pas chose aisée, surtout si des changements topologiques peuvent survenir. Cette discrétisation est pourtant nécessaire pour détecter les collisions et intégrer les forces s'exerçant sur la peau. Elle sert également à l'affichage interactif des matériaux pendant le calcul du mouvement.

La gestion des mélanges entre blocs d'un même matériau n'est pas simple à modéliser de manière visuellement correcte : une fois un objet séparé en deux, il peut être indispensable, selon la nature du matériau, que les deux parties ne se re-mélangent pas, mais au contraire se collisionnent, ce que l'emploi direct d'une formulation implicite ne permet pas. Enfin, le contrôle de volume d'un objet implicite, problème ouvert jusqu'à présent, s'avère dans notre but une caractéristique indispensable pour la crédibilité des animations finales [Las87].

Les sections suivantes proposent des solutions pour ces trois problèmes généraux, importants pour qui veut travailler avec des surfaces implicites.

## 5.3 Échantillonnage de la couche implicite déformable

L'échantillonnage des surfaces implicites est un point crucial : des points d'échantillonnage sont constamment nécessaires pour la détection des collisions et l'intégration des forces de contact. Malheureusement, la majorité des techniques développées pour échantillonner les surfaces implicites demandent un temps de calcul important à chaque mouvement des squelettes, ce qui est problématique pour la gestion de la couche déformable de notre modèle.

### 5.3.1 Intérêt d'une méthode exploitant la cohérence temporelle

Les méthodes traditionnelles d'échantillonnage par partitionnement spatial [WMW86], souvent regroupées sous le terme "Marching Cubes", ont été étudiées pour être utilisées en modélisation. Pourtant, l'une des spécificités de notre application est que l'on souhaite maintenir un

échantillonnage de nos objets implicites au cours du temps. Or, dans la majorité des cas, les objets ne se déforment que très peu entre deux images consécutives d’une animation.

Cette “cohérence temporelle” n’est absolument pas exploitée par les méthodes type *Marching Cubes*, puisqu’elles nécessitent un recalcul complet de la discrétisation à chaque pas de temps. Il paraît donc préférable de choisir une méthode adaptative, comme celle présentée par Andrew Witkin et Paul Heckbert [WH94], qui échantillonne la surface à l’aide de particules répulsives, assurant ainsi à la fois une bonne répartition des points et un temps de calcul raisonnable entre deux pas de temps consécutifs. Cependant, outre le fait que le temps de calcul supplémentaire pour la gestion du mouvement des particules en surface n’est pas négligeable, l’inconvénient majeur de ce moyen de discrétisation est la perte de la notion de voisinage entre les points d’échantillonnage, rendant toute polygonalisation délicate.

### 5.3.2 Échantillonnage adapté aux objets implicites sécables à squelettes

Nous avons préféré adopter une approche différente, dédiée aux surfaces implicites générées par des squelettes. Les principales contraintes sont bien sûr de pouvoir gérer aisément les changements de topologie, tout en profitant le plus possible de la cohérence temporelle.

#### a) Principes de base

Il s’agit de développer une technique efficace permettant de garder des points, que nous appellerons “graines”, sur la surface de l’objet, et si possible répartis de façon uniforme pour permettre une bonne détection des collisions en tout endroit de l’objet. Nous<sup>1</sup> nous sommes inspirés d’une méthode évoquée par Bloomenthal et Wyvill dans [BW90]. Dans cet article, les auteurs préconisaient un moyen d’échantillonnage d’objets implicites fixes par “migration de graines” (*scattering*), où chacune des graines se déplaçait selon le gradient local du potentiel jusqu’à atteindre la surface.

Cependant, faire migrer les graines selon le gradient du potentiel ne paraît pas être optimal dans la mesure où les points d’échantillonnage pourront très bien ne plus recouvrir l’objet de façon uniforme après quelques déformations. De plus, en cas de changement de topologie, une re-génération de graines dans les régions devenues visibles sera souvent nécessaire.

Afin d’éliminer ces problèmes, nous avons décidé d’une stratégie tout autre. En voici les principes de base :

- À chaque squelette de la surface implicite est associé un nombre fixe de graines qui lui sont propres.
- Chaque graine migre en direction de la surface le long d’un axe fixé dans le repère local de son squelette. L’utilisation d’un axe fixe plutôt que la direction du gradient apporte deux avantages immédiats. Tout d’abord, l’évaluation du gradient devient inutile. Et surtout, tout mouvement global de l’objet implicite non déformé ne nécessite aucun calcul : les graines étant stockées dans le repère local, elles restent bien sur la surface et fournissent toujours une discrétisation valide.

---

1. Cette technique d’échantillonnage a été faite en collaboration avec Nicolas Tsingos, à l’époque en Magistère Informatique 2ème année, dans le cadre de son stage à iMAGIS sur la représentation interactive de surfaces implicites, en 93/94.



- Durant les déformations ou les changements topologiques, les graines migrent selon les axes et viennent toujours discrétiser toute la surface de l'objet, ne nécessitant donc pas d'ajout ou de suppression de graines. Afin d'optimiser les calculs et de permettre une bonne répartition de graines, nous avons mis au point une notion de validation et d'invalidation dynamique de graines, que nous développerons plus loin.

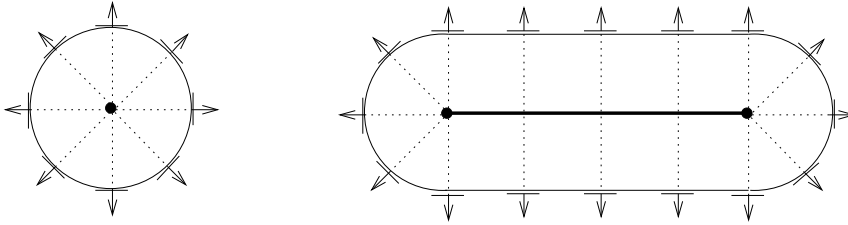


FIG. 5.5 - *Echantillonnage par migration de graines issues d'un point et d'un segment*

Il suffit donc de sélectionner des directions de migration uniformément autour des squelettes (figure 5.5) pour assurer une discrétisation simple et efficace, comme nous allons le détailler. Ces principes simples exploitent aussi la cohérence temporelle des images calculées. En effet, les objets ne se déforment que peu d'une image sur l'autre. Donc, si l'on dispose au début d'une série de graines (et de directions) bien réparties sur les contours de l'objet, on peut réactualiser à chaque mouvement la position de ces graines le long de leur axe directeur et ainsi maintenir une discrétisation de l'objet à peu de frais : l'ancienne position de la graine dans le repère local sera un bon point de départ pour rechercher la surface, puisqu'il y a beaucoup de chance pour que la nouvelle position ne soit guère éloignée.

## b) Initialisation

On doit commencer par trouver une répartition des directions pour les graines. Deux méthodes ont été envisagées. La première idée utilise la subdivision récursive d'un maillage : en partant du barycentre du squelette, on peut commencer par choisir les axes du repère comme premières directions. En envoyant des graines de part et d'autre de ces axes, on obtient déjà six points d'échantillonnage de base. Ils définissent un hexaèdre dans l'espace, soit huit facettes triangulaires approximant l'objet. Puis chacune de ces facettes est subdivisée en quatre nouvelles pour améliorer la discrétisation, et ainsi de suite récursivement, jusqu'à ce que les triangles soient tous d'aire inférieure à un paramètre donné, comme l'indique la figure 5.6.

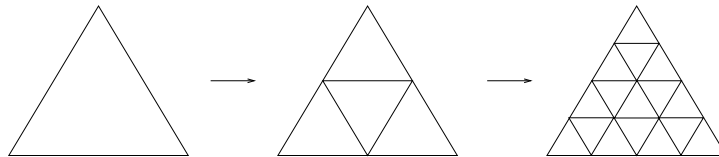


FIG. 5.6 - *Subdivision récursive d'une facette triangulaire*

Il est cependant délicat de prouver que cette méthode conduit à une répartition correcte des graines sur la surface quelque soit le squelette de départ. C'est une des raisons qui nous a conduit

à adopter un autre moyen. Il n'est pourtant pas impossible que dans l'avenir, cette technique puisse être utile à d'autres fins.

Nous avons préféré partir de la boîte englobante du squelette, augmentée de l'épaisseur d'enrobage  $e$ , et discrétiser uniformément ses faces. On trouve alors les directions des graines que l'on cherche en projetant ces nœuds de maillage sur le squelette (voir figures 5.7 et 5.8). Cette fois, il est clair que les directions ne sont pas uniformément réparties. On pourrait d'ailleurs y remédier en créant un maillage non uniforme sur les faces, avec plus de mailles sur les centres des faces que sur les côtés. Mais, les résultats obtenus par la méthode simple exposée ci-dessus étant convenables, nous n'avons pas développé plus cette technique.

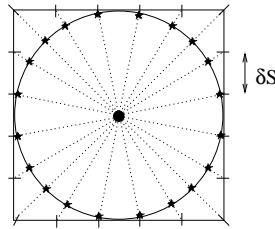


FIG. 5.7 - *Discrétisation uniforme de la boîte englobante en 2D*

Pour éviter à l'animateur de devoir spécifier directement le nombre de graines, nous lui demandons de fixer un paramètre de discrétisation qui correspond à l'élément de surface  $\delta S$  que représentera une graine : cela définira la discrétisation des faces de la boîte. Ainsi, quelque soit l'épaisseur d'enrobage des squelettes, la discrétisation est toujours de la même finesse. Enfin, il suffit de positionner des graines à une distance adéquate du squelette le long de chacun des axes pour disposer d'un échantillonnage du squelette traité. Les différentes étapes d'initialisation sont récapitulées sur la figure 5.8

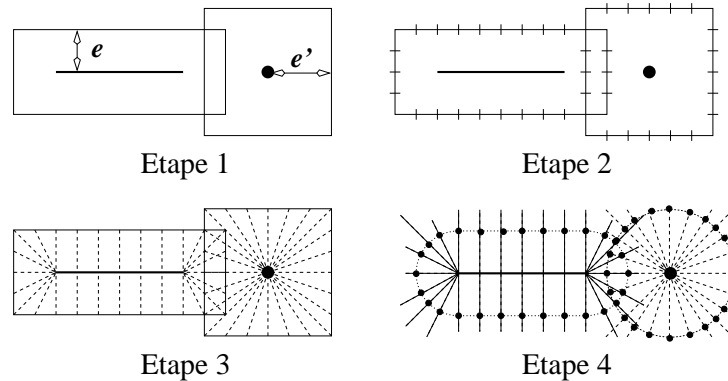


FIG. 5.8 - *Les différentes étapes d'initialisation pour une surface implicite engendrée par deux squelettes*

Quant à l'emploi hypothétique de squelettes non convexes où cette méthode poserait quelques problèmes, deux solutions sont envisageables. Soit on obtient des points de discrétisation par une méthode type Marching Cubes, puis on obtient les axes par projection sur le squelette. Soit on découpe le squelette en une série de composants convexes pour se ramener à un cas simple.

### c) Migration des graines

Après la phase d'initialisation, les graines migrent chacune le long de leur axe afin de rejoindre la surface implicite résultant du mélange des contributions de chaque squelette. En cas de déformation, elles partiront de leur ancienne position stockée dans le repère local de leur squelette, et migreront à la recherche de la nouvelle surface : la cohérence temporelle en animation nous permet d'assurer un coût de calcul réduit, puisque la graine n'aura pas à beaucoup changer de position d'une image à l'autre.

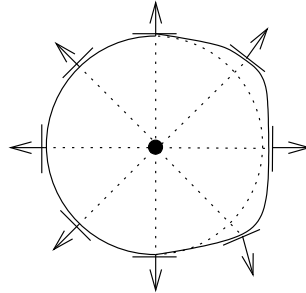


FIG. 5.9 - Migration de graines lors d'une déformation

#### Recherche d'un point de la surface

Pour trouver la nouvelle position, la méthode que nous avons retenue consiste à faire des petits pas de recherche dont la taille dépend de l'épaisseur du squelette considéré, de manière à trouver un point intérieur et un point extérieur à l'objet. La taille des pas est choisie en fonction de l'épaisseur d'enrobage et du pas de temps pour être adaptée au mieux. Par exemple, supposons que notre graine se trouve à l'extérieur de l'objet (ce qui se vérifie aisément en testant le potentiel de l'objet en ce point). Nous allons alors avancer par étapes le long de l'axe de la graine jusqu'à trouver deux points consécutifs dont l'un est à l'extérieur et l'autre à l'intérieur. On procède ensuite par simple méthode de recherche de zéro (dichotomie ou Regula Falsi [PTVF92]) entre ces deux points pour trouver la surface. Grâce toujours à la cohérence temporelle, nos tests ont montré que le nombre moyen de pas à effectuer pour chaque graine se situe autour de 3, et le nombre d'itérations de la dichotomie autour de 5. Ceci est bien sûr une moyenne, car s'il n'y a pas de chocs, ces valeurs sont nulles.

#### Validation et Invalidation des graines

Un problème de répartition des graines se pose lorsque les objets sont constitués de plusieurs squelettes. Une solution intuitive voudrait que l'on affecte des graines à chacun des squelettes le constituant, et que l'on pratique la migration de façon indépendante. Mais des effets indésirables apparaissent : à la fois, le nombre de graines à traiter peut devenir considérable, et de plus, on s'expose à avoir des lieux de concentration de graines, ce qui est contraire au souhait de répartition uniforme.

Nous avons donc incorporé dans l'algorithme de recherche de la surface un test permettant d'*invalidier* certaines graines. Il fallait trouver un test qui garantisse un compromis entre le nombre de graines restant valides (devant être suffisant pour la détection des collisions) et

une bonne répartition de ces graines. Un test simple nous a suffi : si une graine entre dans une zone où le potentiel de son squelette associé n'est plus prépondérant, on stoppe simplement la progression de la graine qui devient invalide. Intuitivement, cela signifie que si une graine d'un squelette s'aventure trop près d'autres squelettes, elle n'est plus utile puisque les graines des autres squelettes se chargent d'échantillonner cette région. La figure 5.10 montre un exemple où deux squelettes mélangent leur potentiel, ce qui a pour effet d'invalider les graines dans la zone d'intersection.

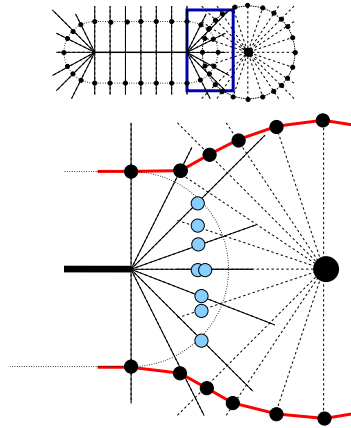


FIG. 5.10 - Gros plan sur la zone de mélange après le processus de migration/invalidation (les graines valides sont en noir, les invalides en gris)

Ce test s'implémente aisément : lors du calcul du potentiel de l'objet au point courant de la graine, on doit faire une somme de potentiels pour connaître le potentiel total (on calcule  $\sum_{s_i} f_i(r_i)$ ). Il suffit de vérifier au fur et à mesure du calcul que le potentiel dû au squelette propriétaire de la graine est plus important que chacun des autres potentiels.

Ces graines invalides peuvent très bien quelques temps après être de nouveau validées, puisque la topologie de l'objet peut changer. On a ainsi défini une méthode souple et peu coûteuse pour disposer d'une discrétisation sur des objets à topologie variable tout en gardant un nombre de graines valides raisonnable et une répartition globalement satisfaisante. Ce critère définit d'ailleurs une zone de prépondérance propre à chaque squelette : nous avons choisi d'appeler ces zones *territoires*, puisque qu'elles représentent les parties où chaque squelette est prépondérant. Plus formellement, le territoire  $T_i$  associé au squelette  $s_i$  est défini par :

$$T_i = \{P \in \mathbb{R}^3 / (f(P) \geq c) \text{ and } (\forall j f_i(P) \geq f_j(P))\}, \quad (5.2)$$

et l'ensemble de ces territoires *pave* le volume de l'objet. Ce découpage en territoires est en fait une *partition de Voronoï* de l'objet pour la distance définie par le maximum des contributions de potentiels. On maintient, par la méthode que l'on vient de décrire, ce partitionnement au cours du temps.

### 5.3.3 Avantages découlant d'un tel échantillonnage

#### a) Polygonalisation interactive

Parmi les caractéristiques nécessaires à tout logiciel d'animation, la visualisation interactive des animations en cours de calculs est un point essentiel. Maintenir un échantillonnage de la surface doit aussi permettre de visualiser les objets rapidement.

#### Représentation par écailles

Witkin et Heckbert proposent d'afficher sur chaque point d'échantillonnage une facette (ou *écaille*), dont la normale est alignée au gradient local de la surface implicite. Cette représentation, illustrée sur le dessin gauche de la figure 5.11, est effectivement rapide, mais ne permet pas une visualisation claire dès que la scène est un tant soit peu complexe : on distingue mal les objets en avant-plan des objets en arrière-plan.

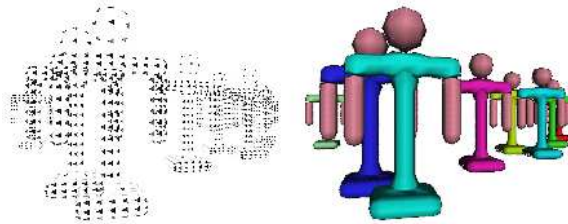


FIG. 5.11 - *Comparaison entre écailles et représentation opaque par territoires pour une scène complexe*

#### Représentation opaque par territoires

Notre technique d'échantillonnage dispose d'une particularité exploitable : pour chaque squelette, *les relations de voisinage entre les graines qu'il gère sont toujours conservées*, même lorsque l'objet se déforme. Lors de l'initialisation, on définit un maillage polygonal s'appuyant sur les graines. Ce maillage va rester valide tout le temps (figure 5.12); cela ne pouvait bien sûr pas se faire avec une technique à base de particules [WH94].

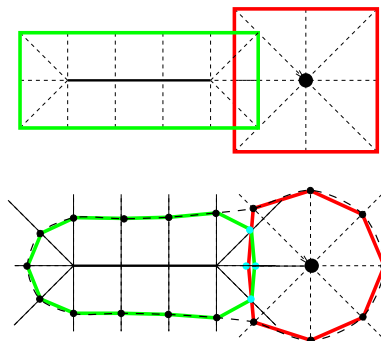


FIG. 5.12 - *La polygonalisation par primitives (en 2D) : initialement (a) puis une fois les potentiels mélangés (b)*

On ne dispose pas alors d'une polygonalisation complète de tout l'objet, mais d'une polygonalisation pour chaque primitive constituant l'objet implicite. La visualisation opaque obtenue n'est pas parfaite : on distingue bien les "raccords" entre chaque territoire, comme le prouvent les images (c) et (d) de la figure 5.13. Mais elle ne demande aucun temps de calcul et s'avère largement suffisante pour la prévisualisation des animations. De toute façon, la réalisation d'une séquence animée se terminera par une phase de lancer de rayons en post-processing, de manière à produire des images de haute qualité.

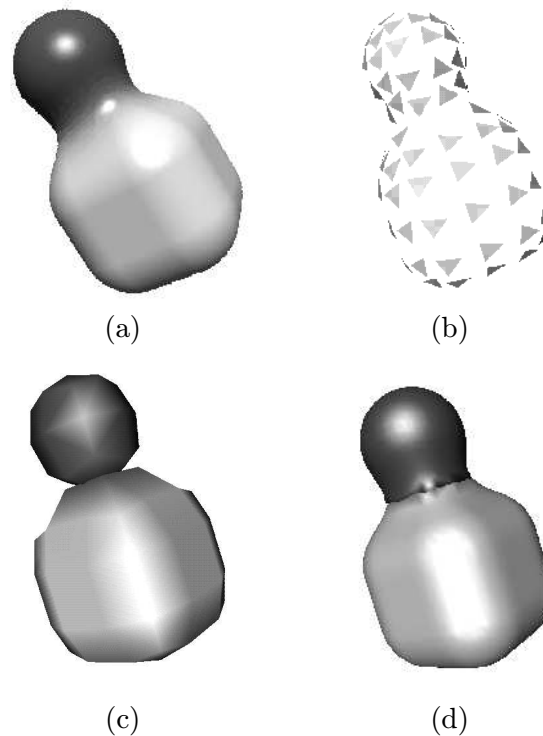


FIG. 5.13 - *Quatre versions d'un même objet : par lancer de rayons (a), par écailles (b), puis par la représentation opaque interactive pour deux précisions d'échantillonnage différentes (c-d)*

### b) Hiérarchie de boîtes englobantes

Disposer des graines valides pour chaque squelette fournit un deuxième avantage propre à notre technique d'échantillonnage. En effet, elle permet de définir une hiérarchie de boîtes englobantes qui accélère les détections de collision, ainsi que le rendu final.

#### Hiérarchie du volume

En réutilisant la propriété dont dispose notre objet implicite, à savoir être partitionné en territoires propres à chaque primitive, il est particulièrement aisé de trouver des boîtes parallèles aux axes qui englobent chaque volume : ce sont les boîtes minimales contenant toutes les graines de chacune des primitives, augmentées de la taille d'échantillonnage. Ainsi, un objet peut être approximé par une véritable hiérarchie de boîtes englobantes : des boîtes des squelettes peut

être déduite une boîte par composante connexe, et de ces boîtes sera encore déduite une boîte globale, comme le montre le schéma 5.14.

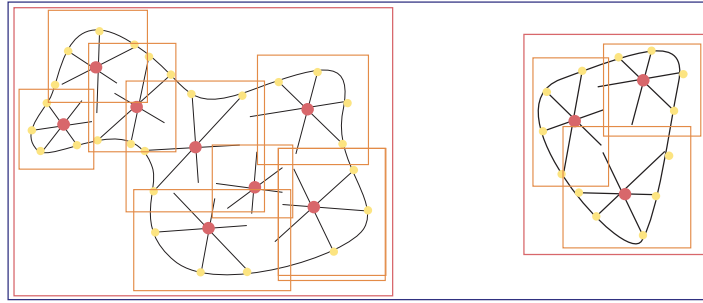


FIG. 5.14 - Une hiérarchie de boîtes englobantes pavant le volume d'un objet implicite

### Hiérarchie de la surface

Une hiérarchie plus fine peut même être construite en ne considérant que les graines *valides* de chaque squelette : on pave ainsi de boîtes locales la surface de l'objet au lieu de paver son volume.

### Application à la détection de collision

Une première application de ce découpage hiérarchique de l'espace est l'accélération de la détection de collision au cours d'une animation. La façon canonique de détecter une collision entre deux objets  $I_1$  et  $I_2$  était d'évaluer le potentiel de  $I_2$  sur tous les points de discrétisation de  $I_1$  et de le comparer à la valeur *iso* pour décider si interpénétration il y a [Gas93]. Un test aussi précis mais bien plus rapide peut se faire de la façon suivante : on teste tout d'abord s'il y a intersection des boîtes englobantes des deux objets. Si oui, alors on teste l'intersection entre les sous-boîtes englobantes de  $I_1$  avec la boîte englobante de  $I_2$ . Pour celles où il y a intersection, on teste enfin si elles intersectent une sous-boîte de  $I_2$ . Si c'est toujours le cas, on peut alors effectuer le test classique portant sur les potentiels. L'intersection de boîtes parallèles aux axes se testant très rapidement, on allège les calculs de façon drastique dans la majorité des cas. Même dans le pire des cas, à savoir si  $I_1$  est dans  $I_2$ , cette méthode entraîne un surcoût relativement faible.

### Application au lancer de rayons

Le lancer de rayons sur les surfaces implicites nécessite de calculer l'intersection entre une demi-droite et un objet défini implicitement, comme nous l'avons vu au paragraphe 3.5.3. Or, encore une fois, notre structure à trois niveaux de boîtes englobantes permet d'éviter des calculs d'intersection coûteux, et accélère donc le rendu comme le détail [Gas95].

### 5.3.4 Discussion

Notre approche, détaillée dans [DTG95, DTG96], se distingue de la méthode de Witkin et Heckbert [WH94] sur principalement trois points :

- Tout d’abord, notre idée de base est que chaque squelette dispose de ses propres *graines*, ou points d’échantillonnage. Ces graines se déplacent sur des axes partant du squelette pour venir se “coller” à la surface implicite après une recherche dichotomique. On profite bien sûr énormément de la cohérence temporelle au cours d’une animation, chaque graine partant de sa position précédente dans le repère local de son squelette.
- Le deuxième point de différence est la mise au point d’un critère d’invalidation des graines : pour non seulement éviter de perdre du temps dans l’échantillonnage, mais aussi pour respecter une bonne répartition des échantillons, on stoppe la migration des graines le long de leurs axes quand elles pénètrent dans une zone où le potentiel créé par leur squelette émetteur n’est plus prépondérant.
- Enfin, cet échantillonnage permet une représentation polygonale de l’objet à faible coût, utilisée pour la visualisation interactive des animations, ainsi qu’un découpage hiérarchique de l’espace pour le rendu ou la détection de collision.

Ces différences en font une méthode bien plus adaptée à nos besoins. Bien sûr, notre méthode ne peut rivaliser avec la qualité de la discrétisation obtenue par [WH94], mais en relâchant la contrainte d’échantillonnage uniforme, nous avons pu gagner en temps de calcul et en retombées pour l’animation de notre modèle.

## 5.4 Gestion des mélanges indésirables

Abordé au paragraphe 3.4, le problème du mélange systématique de toutes les primitives d’un objet entre elles apporte un inconvénient majeur : on ne peut pas, par exemple, modéliser un personnage avec les bras le long du corps (voir figure 3.11). Dans notre application aussi, cette particularité des surfaces implicites est un défaut à régler : typiquement, deux blocs séparés d’un matériau déformable vont mélanger leur potentiel, et donc se fusionner à distance, avant même qu’une collision ait pu être détectée. Nous proposons deux types de solutions, selon la nature des objets étudiés.

### 5.4.1 Cas des objets ne changeant pas de topologie

Le problème a été résolu pour les objets ne changeant pas de topologie grâce à l’utilisation d’un graphe de mélange [WW89, OM93], définissant quelles primitives doivent se mélanger et lesquelles ne le doivent pas. L’algorithme d’évaluation du potentiel devient alors : pour un point  $\mathbf{P}$  de l’espace, on retourne la plus forte contribution des potentiels sommés par groupe de squelettes voisins dans le graphe de mélange.

Pour traiter les problèmes de mélanges indésirables pour un matériau ne changeant pas de topologie, il suffit donc de définir un graphe de mélange en début d’animation. Chaque squelette disposera ainsi de la liste des squelettes voisins avec lesquels il somme son potentiel, et cela résoudra les inconvénients précédemment cités. Notons que cette méthode peut créer des discontinuités de tangente sur la surface de l’objet, ainsi que des fentes lorsque le calcul est mal effectué [GW95].



Pour assurer au minimum une continuité  $C^0$  en tout point de la surface, nous<sup>2</sup> avons remplacé le graphe de mélange par un *arbre d'évaluation*, beaucoup plus intuitif. En effet, il est plus facile de comprendre un potentiel écrit sous forme arborescente que sous forme algorithmique : grâce à cet arbre, on dispose d'un moyen systématique pour calculer le potentiel tout en évitant le problème de mélange indésirable. Prenons l'exemple d'un objet à trois squelettes, dont les deux squelettes extrêmes ne doivent pas pouvoir mélanger leur potentiel, comme indiqué sur la figure 5.15(a).

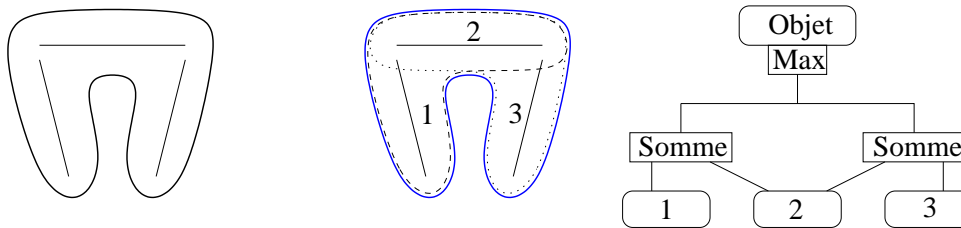


FIG. 5.15 - Un objet à trois squelettes, et son arbre d'évaluation (graphe acyclique orienté).

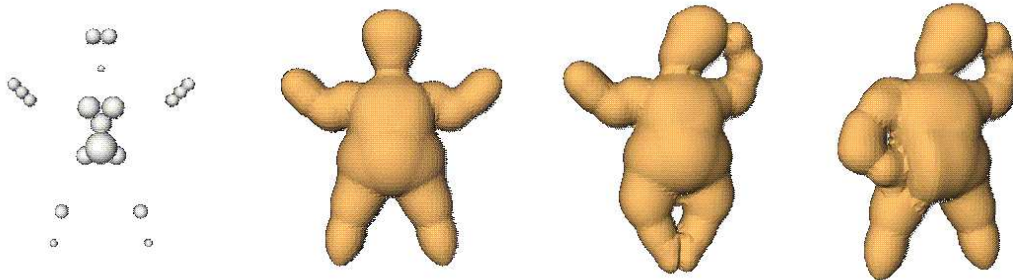


FIG. 5.16 - Personnage défini par arbre d'évaluation, et différentes postures prouvant la bonne gestion des mélanges (d'après [OCG97])

Alors le potentiel de l'objet devra être défini comme étant le maximum des potentiels issus de la somme des squelettes 1 et 2, et 2 et 3. De cette façon, les squelettes 1 et 3 ne mélangent pas leur potentiel et peuvent même se télescoper et se déformer en conséquence. Ce même exemple, un peu plus développé, est représenté sur la figure 5.16. Le graphe de mélange s'écrit alors comme le maximum des différentes sommes des sous-graphes complets, ou cliques [RCGG<sup>+</sup>97].

L'avantage de la représentation par arbre d'évaluation par rapport à un graphe de mélange est que cela unifie l'écriture des potentiels. En effet, toutes les façons de définir une surface implicite vues dans l'état de l'art au chapitre 3 peuvent se ramener à une représentation de type arbre d'évaluation de potentiel. Cette technique étant de plus facilement implémentable en classes dans un langage orienté objet, elle nous permet de disposer d'une véritable plate-forme d'utilisation et de tests des surfaces implicites [RCGG<sup>+</sup>97].

L'algorithme décrit ci-dessus convient donc parfaitement à l'animation de personnages, empêchant tout mélange indésirable ; malheureusement, dans le cadre plus général qui est le nôtre,

2. Cette partie sur les arbres d'évaluation a été faite en collaboration avec Dominique Rossin, stagiaire polytechnicien, et Marie-Paule Cani-Gascuel, Jean-Dominique Gascuel, et Agata Opalach.

un objet est susceptible de changer de topologie, en se cassant par exemple : il faut donc pouvoir changer dynamiquement le graphe de mélange ou l'arbre d'évaluation.

### 5.4.2 Cas des objets pouvant subir des fractures

L'un des avantages des surfaces implicites est de pouvoir modéliser les fractures : un objet peut passer d'une seule composante connexe à plusieurs, du fait du mouvement relatif de ses squelettes. Cependant, si deux morceaux d'un objet qui s'est fracturé viennent à se rapprocher, ils se mélangent à nouveau, à distance. On comprend donc que l'on ne peut se satisfaire d'un graphe de mélange fixe pour un tel objet : il faut mettre au point un graphe de mélange *dynamique*.

Notre solution consiste à définir pour chaque squelette une "liste de mélange", c'est-à-dire la liste de tous les squelettes devant se mélanger avec lui. On l'initialise comme un graphe complet, c'est-à-dire où tous les squelettes mélangent leur potentiel respectif. Ce graphe de mélange reste valide jusqu'à l'apparition d'une nouvelle composante connexe (fracture). On le maintient facilement à jour au cours du temps par l'algorithme suivant, exécuté à chaque pas de l'animation :

- Pour chaque paire de squelettes voisins dans le graphe de mélange courant, vérifier si leurs zones d'influence (définies par le rayon d'influence de leur potentiel) s'intersectent. La fermeture transitive de cette relation nous donne alors les différentes composantes connexes du graphe d'influence (voir figure 5.17). Par exemple, dans la figure 5.17, le point *A* sera détecté comme étant dans la même composante connexe que *B*, mais pas dans la même que *C*.

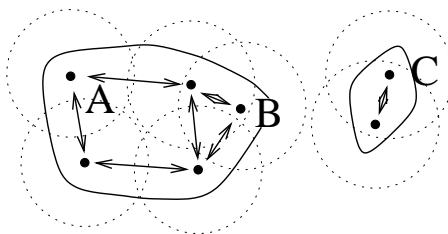


FIG. 5.17 - Le graphe d'influence et ses composantes connexes. Les squelettes *A* et *B* résident dans la même composante : leurs potentiels s'additionneront donc s'ils se rapprochent l'un de l'autre.

- Pour chaque squelette, enlever de sa liste de mélange les squelettes n'étant plus dans la même composante connexe que lui.

De cette manière, on tient à jour pour chaque squelette la liste des autres squelettes appartenant à sa composante connexe. Cette technique peut se transcrire en terme d'arbre d'évaluation si on le souhaite, et permet ainsi une gestion dynamique des séparations, ce qui permet que *deux composantes d'un même objet peuvent se collisionner* après s'être séparées. Il faut remarquer ici que les tests sur les zones d'influences ne permettent pas de détecter la séparation dès qu'elle se produit, mais de toute façon, aucune réduction de voisins ne peut se faire avant que les régions d'influence soient totalement disjointes, sous peine de créer une brutale altération de la forme des composants séparés.

### a) Fusion instantanée

Lorsque deux parties d'un même matériau entrent en collision, il peut être souhaitable de les refusionner sur l'instant. Cela se fait très simplement : dès qu'une collision se produit entre deux composantes distinctes, on fusionne leurs listes de voisinage de façon à ce que l'ensemble ne forme plus qu'une seule composante, dont tous les squelettes se mélangent. On effectue ainsi de façon simple une fusion instantanée, illustrée par la figure 5.24.

### b) Fusion progressive par compression

Une fusion plus progressive peut aussi être opérée. Il s'agit alors de ne faire fusionner des blocs de matériau entre eux que si la force les pressant les uns contre les autres dépasse une certaine valeur. Lors du traitement d'une collision entre deux composantes distinctes, on compare donc les forces de contact entre deux territoires en collision avec le seuil de fusion défini par l'utilisateur pour cette substance. Si le seuil est dépassé, on rajoute au graphe de mélange l'information que les deux squelettes concernés par la collision se mélangent à présent : cela créera une fusion locale entre les deux composantes. On peut alors aboutir à des matériaux se mélangeant localement, tout en se collisionnant partout ailleurs : c'est le cas de la figure 5.18 où un rouleau à pâtisserie fait se fusionner progressivement deux matériaux mous. Malheureusement, cette fusion progressive ne peut, dans le cas général, garantir une continuité  $C^1$  : des discontinuités de tangentes peuvent intervenir lors des pas successifs d'une telle fusion locale.



FIG. 5.18 - Fusion locale en cas de forte compression : progressivement, les squelettes concernés par la collision mélangent leur potentiel respectif (ossature : 4 particules en haut, 5 en bas).

### 5.4.3 Modification du traitement des collisions

Reste, une fois l'arbre d'évaluation des potentiels défini, à traiter correctement les collisions, y compris les auto-collisions entre différentes parties d'un même objet. Pour cela, on détecte et on traite les collisions *entre chaque paire de territoires* associés à des squelettes ne se mélangeant pas au sens du graphe de mélange. Ceci assure, dans le cas de la figure 5.16, un traitement de la collision entre le bras et le corps. L'algorithme de traitement des collisions s'implémente très aisément :

```

Pour tous les couples de squelettes (s1,s2) :
  Si des graines de s1 sont dans le territoire de s2
    Si s2 n'est pas parmi les voisins de mélange de s1
      alors COLLISION !

```

La collision in fine se fait en déplaçant les graines en collision et en calculant les forces résultant d'une telle déformation, comme expliqué en section 3.6.3. La procédure de traitement des collisions est donc extrêmement simple, et très générale puisqu'elle traite aussi sans cas particulier les auto-collisions, étendant ainsi la technique de [Gas93].

#### 5.4.4 Discussion

En gérant correctement les apparitions et disparitions de composantes connexes, notre modèle permet un contrôle précis et varié des mélanges entre objets, permettant plusieurs politiques de séparations/fusions via la modification dynamique d'un graphe de mélange ou d'un arbre d'évaluation. De plus, cette façon de faire permet d'étendre le traitement des collisions du modèle original, en permettant de détecter les auto-collisions sans traitement particulier. On gagne ainsi à la fois en généralité et en efficacité.

### 5.5 Contrôle du volume

Pouvoir contrôler le volume lors d'une animation est souvent capital [PB88, Las87]. Certaines simulations physiques se doivent par exemple de conserver le volume au cours du mouvement. Pourtant, une surface implicite définie par des primitives simples voit son volume varier lorsque les primitives se déplacent les unes par rapport aux autres, comme le montre la figure 5.19. Bien que souvent constaté [WW89], ce problème n'a jamais été abordé dans le cas général qui est le nôtre, où l'on peut avoir affaire à des fractures ou autres changements de topologie importants. Nous allons voir que grâce à notre méthode d'échantillonnage, ce problème trouve une solution élégante.

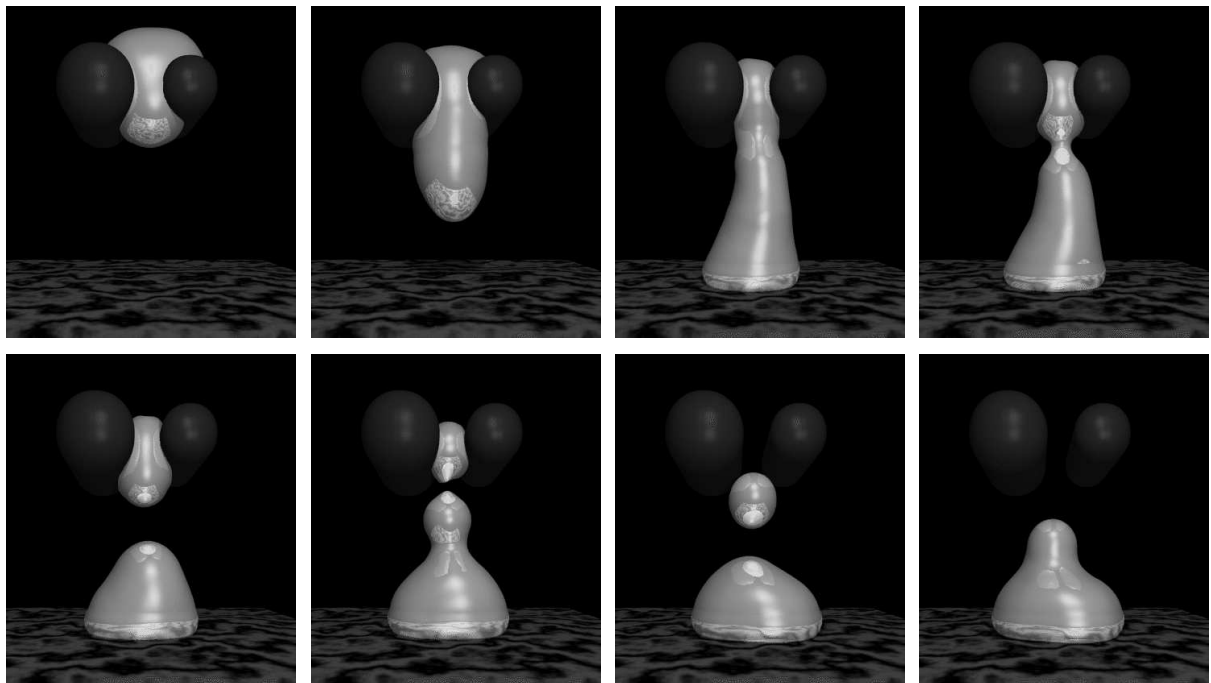


FIG. 5.19 - Variations de volume lors de la chute d'un matériau hautement déformable composé de 19 particules en ossature.

### 5.5.1 Tentative de résolution analytique du problème

La première idée pour résoudre ce problème a été d'essayer des fonctions potentielles susceptibles de procurer un volume constant. En écrivant que le volume du potentiel sommé pour  $n$  particules concentrées en un seul point doit être égal à la somme des volumes du potentiel de chacun des squelettes pris indépendamment, on contraint la fonction potentiel à conserver le volume entre deux positions extrêmes [DG94]. Malheureusement, cette solution s'est révélée inutile dans le cas général, les variations de volume pouvant être énormes dans les positions intermédiaires comme le prouve le diagramme de la figure 5.20 obtenu par découpage de l'espace en voxels. En fait, force est de constater que le volume ne peut en aucun cas rester constant si la fonction implicite elle-même n'est pas modifiée en fonction du mouvement.

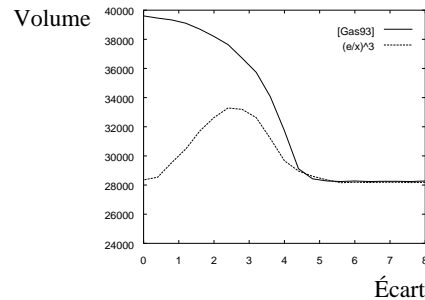


FIG. 5.20 - Variations de volume lors du mouvement relatif entre deux squelettes, pour le potentiel de [Gas93] et pour un potentiel contraint à garder un volume égal lors d'un écart nul et d'un écart infini.

### 5.5.2 Détection des variations locales de volume

Contrôler le volume signifie en premier lieu pouvoir le déterminer. Or un calcul analytique du volume à chaque instant n'est pas envisageable dans notre cadre vu la complexité des intégrations à effectuer. Reste, parmi les solutions classiques, la discrétisation de l'espace en voxels pour approximer le volume. Cependant cette méthode souffre de deux problèmes différents : premièrement, elle est relativement coûteuse, et deuxièmement, elle ne permet pas de déterminer aisément les régions où le volume a changé.

Il semble pourtant souhaitable de pouvoir agir de façon locale sur le volume : comme le montre la figure 5.21, dans le cas d'un mouvement relatif entre squelettes, il est préférable de contrôler le volume localement près des squelettes concernés plutôt que d'effectuer une correction globale qui déformerait tout l'objet.

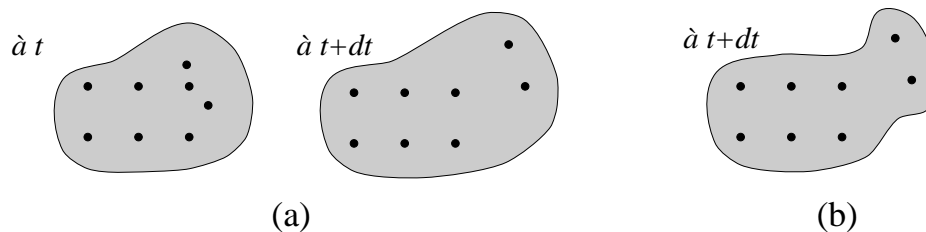


FIG. 5.21 - (a) Variation de volume. (b) Correction locale du volume.

C'est pourquoi nous avons repris l'idée des *territoires*, définis pour l'échantillonnage (voir équation 5.2). Chaque squelette dispose alors d'un volume, le volume de son territoire, qui lui est propre : l'ensemble de ces volumes forme une partition du volume global de l'objet. Approximer ce volume est chose aisée, encore une fois grâce à la méthode d'échantillonnage choisie : il suffit de sommer les volumes des petites pyramides centrées sur le squelette et qui s'appuient sur les graines comme le dépeint la figure 5.22. Chacun de ces volumes s'écrit simplement :

$$V_i = \sum_{s \in S_i} k d(s, P_i)^3$$

Le facteur  $k$  ne dépend que de la finesse de la discrétisation  $\delta S$ , et peut donc être omis dans le cadre de la conservation du volume.

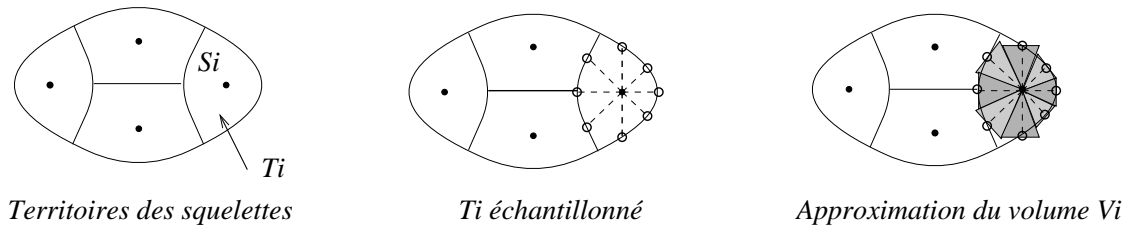


FIG. 5.22 - Territoires de squelettes et calcul du volume grâce aux graines.

### 5.5.3 Correction du volume

À partir de cette approximation du volume, on peut gérer les variations de volume de chaque squelette grâce à un contrôleur Proportionnel-Dérivé, technique classique en robotique servant à asservir la sortie d'un système (linéaire ou non) en modifiant l'entrée.

La méthode que nous avons mise au point consiste à changer la fonction potentiel de chaque primitive de façon à équilibrer les variations de volume observées lors de l'échantillonnage [DG95]. Pour ne pas modifier les paramètres physiques de notre objet qui sont stockés dans les variations de la fonction potentiel  $f$  (voir section 3.6.2), on agira sur l'enrobage par une simple *translation* de  $f$ . Ainsi, en cas de baisse de volume, on translatera  $f$  vers la droite ce qui provoquera une dilatation de la composante implicite du squelette. Au contraire, en cas d'excès de volume, on translatera le potentiel vers la gauche de façon à réduire la taille de l'enrobage.

Nous avons choisi de contrôler la dérivée de la translation plutôt que la translation elle-même, afin d'obtenir des variations de volume plus régulières. Notre contrôleur sur chaque squelette  $s_i$  prend alors comme entrées à l'instant  $t$  :

$$\Delta_i^t = \frac{V_i^t - V_i^0}{V_i^0} \quad \dot{\Delta}_i^t = \frac{V_i^t - V_i^{t-dt}}{V_i^0 dt}$$

(où  $V_i^t$  est le volume approximé à l'instant  $t$ ) et rend la valeur de la dérivée de la translation par rapport au temps à l'instant  $t$ ,  $\dot{\epsilon}_i^t$  :

$$\dot{\epsilon}_i^t = \alpha \Delta_i^t + \beta \dot{\Delta}_i^t$$

La valeur du potentiel  $f(\|P - s_i\|)$  est donc remplacée par :  $f(\|P - s_i\| - \epsilon_i^t)$ . Le fonctionnement du contrôleur est alors simple : si le volume dérive du volume souhaité, on compensera la

différence en modifiant l'épaisseur d'enrobage en conséquence. Ainsi, on est assuré de contrôler le volume localement comme le montre la figure 5.23 où deux squelettes ponctuels se rapprochent l'un de l'autre, avec ou sans contrôle de volume. L'écart est considérable : les variations de volume sans contrôleur montent à 400% du volume initial, alors qu'un bon choix des paramètres du contrôleur permet de rester au dessous des 5% de variation relative.

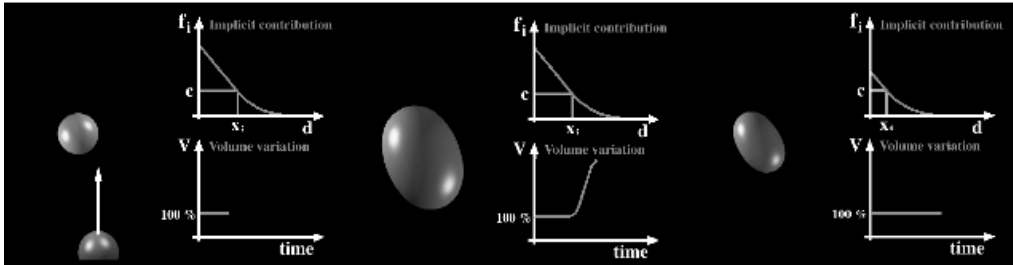


FIG. 5.23 - Deux squelettes en mouvement se mélangent : à gauche sans contrôleur, à droite avec contrôleur.

#### 5.5.4 Discussion

Le problème délicat du contrôle du volume d'objets implicites trouve ici une solution rapide et efficace. En exploitant la discrétisation qui nous était de toute façon nécessaire au bon fonctionnement du modèle, cette technique simple assure à nos objets une conservation du volume au cours des déformations, ce qui peut être *primordial* dans le cas de fusion immédiate de matériaux hautement déformables où des variations importantes pourraient se produire sinon (voir figure 5.24). Cette méthode est évidemment aussi bien utilisable pour le maintien d'un volume constant que pour des variations de volume imposées par l'utilisateur. Il suffit pour cela de faire varier la valeur objectif  $V_i^0$  de chaque squelette au cours du temps : le contrôleur agira sur l'épaisseur d'enrobage de la même façon, de sorte que les variations de volume imposées seront prises en compte pour l'objet implicite.

## 5.6 Champ d'applications

Ce premier modèle [DG95, CGD97] d'objets déformables offre un grand nombre d'avantages : on dispose en effet d'une méthode permettant d'animer des matériaux subissant de fortes déformations, avec une gestion aisée des changements de topologie, une modélisation fine des contacts surfaciques, et un contrôle local du volume. Le tout peut être visualisé interactivement sans surcoût de calcul. Enfin, une visualisation très fine peut se calculer par lancer de rayons, apportant une qualité de rendu visuel très importante pour l'ensemble.

Grâce à l'emploi d'un modèle stratifié, le champ d'applications dépasse le cadre que nous nous étions fixé au départ : en changeant un peu les modèles présents dans les différentes couches, on étend le champ des objets simulés par ce modèle. En voici quelques exemples que nous avons développés ou auxquels nous avons participé.

### 5.6.1 Objets hautement déformables

Le but premier était, comme nous l'avons dit, l'animation d'objets hautement déformables. Dans ce cas, toutes les techniques développées dans ce chapitre s'appliquent intégralement et directement : les changements de topologie, séparations ou fusions, ainsi que le contrôle de volume en font un modèle complet et efficace. Des animations prouvant le bon fonctionnement du tout ont été calculées. Parmi elles, un petit film intitulé *Kitchen Fiction* [GDGR96] incorpore tous les aspects intéressants de notre modèle. Les figures ci-dessous illustrent en partie ces animations.

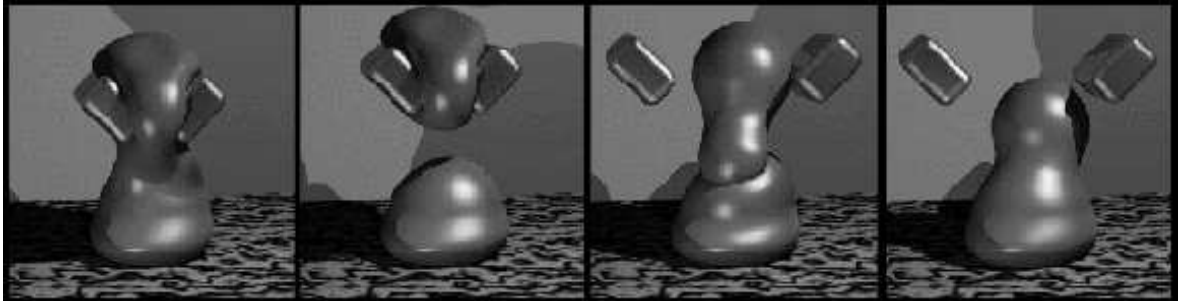


FIG. 5.24 - Une pince virtuelle saisit une pâte, puis la relâche : pendant ce temps, une séparation, puis une fusion immédiate se produisent au sein de la matière.

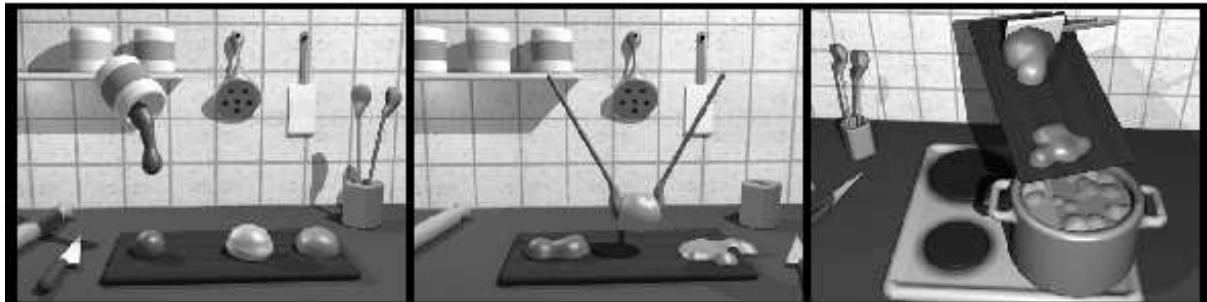


FIG. 5.25 - Extraits de *Kitchen Fiction* [GDGR96] : différentes substances sont manipulées par des ustensiles rigides, également modélisés par surfaces implicites.

### 5.6.2 Personnages articulés

Si l'on souhaite animer des personnages, il faudra cette fois choisir un modèle interne de type solides articulés. Ainsi, selon les forces externes fournies par la couche déformable implicite, le squelette s'anime tout en respectant les contraintes de liaisons aux articulations, ce que nous avons fait en utilisant l'algorithme dit de "displacement constraints" [GG94]. Quant à la couche déformable implicite, elle modélisera les contacts en simulant une chair à un coût relativement faible. Les intercollisions sont aussi détectées et traitées, comme on le voit sur la figure 5.26.

### 5.6.3 Éboulement

Notre modèle a aussi été utilisé dans un cadre tout autre, qui est la simulation de risques naturels comme les éboulements en particulier. Les simulations classiques en géophysique peuvent





FIG. 5.26 - *Un personnage simplifié, tombant à terre.*

difficilement se faire en 3D en des temps raisonnables : le BRGM (Bureau des Recherches Géologiques et Minières) a donc choisi de prendre un modèle d'animation simplifié. Pour augmenter le réalisme géophysique, le sol a été modélisé à l'aide de pistons qui se déforment selon des lois de contraintes/déplacements connues. Ainsi, des blocs de pierre modélisés par notre technique dévalent des pentes munies de lois déterminées par l'utilisateur : la trajectographie résultante est alors étudiée pour déterminer les régions à risque lors d'éboulements. Ce travail fait actuellement l'objet d'un contrat de collaboration entre le BRGM et iMAGIS, supervisé respectivement par Eric Leroi et Jean-Dominique Gascuel.



FIG. 5.27 - *Un exemple de trajectoire pour une pierre dévalant la colline.*

#### 5.6.4 Autres exemples

La couche déformable mise au point dans ce chapitre a aussi accessoirement servi dans d'autres modèles physiques nécessitant un habillage. Ce fut par exemple le cas pour le modèle d'objets à mémoire de forme à base de particules orientées, développé par Lombardo et Puech [LP95], ou pour un modèle de lèvres parlantes développé lors d'une collaboration Institut de la Communication Parlée / iMAGIS [GMTA<sup>+</sup>96]. Dans les deux cas, les surfaces implicites permettent de visualiser un ensemble de squelettes discrets qui servent d'ossature. Le modèle déformable implicite procure une surface à l'ensemble, tout en s'occupant de ce qui est détection des collisions et transfert des forces de contact.

## 5.7 Discussion

Ce premier modèle, et particulièrement la couche déformable implicite, permet d'atteindre les objectifs que nous nous étions fixés. Notre modèle, très général puisqu'extensible à différents types d'objets, permet de simuler les matériaux hautement déformables de façon efficace et facilement paramétrable. Les changements de topologie sont assurés grâce à la fois au système de particules et à la formulation implicite de l'habillage, les contacts sont gérés correctement par la couche implicite, le volume est contrôlé aisément, les mélanges progressifs ou instantanés sont prévus, et le tout avec une visualisation fine. Pour mettre au point ce premier modèle, il a fallu résoudre certains problèmes propres aux surfaces implicites (échantillonnage, gestion des mélanges et du volume). Grâce à ces solutions, les surfaces implicites deviennent un outil d'animation simple d'utilisation et très efficace.

La mise en pratique de ces différentes techniques révèle pourtant des points faibles. Il n'est par exemple pas évident de trouver le bon nombre de particules à utiliser pour un matériau. Pour un faible nombre de particules, l'animation risque de trop faire apparaître la granularité du modèle : on devine les particules sous la couche déformable. Mais trop augmenter le nombre de particules dans l'espoir d'obtenir un comportement plus fin induit des temps de calcul très importants. Les particules sont donc peu maniables telles quelles : on souhaiterait pouvoir adapter leur nombre en fonction des mouvements recherchés.

On peut aussi regretter l'absence de propriété physique pour la peau : le réalisme pourrait être accru si notre modèle pouvait gérer une tension de surface en plus, ce qui n'est pas prévu dans le modèle actuel.

Nous proposons donc dans la dernière partie de remédier à ces défauts en modifiant dans un premier temps la définition du modèle de particules pour à la fois le rendre plus pratique et plus rapide. Un modèle de peau munie de propriétés physiques sera présenté dans un second temps, enrichissant ainsi notre modèle générique.



## Troisième partie

# Modélisation approfondie des différentes couches



## Chapitre 6

# Un modèle de particules adaptatives en temps et espace

Depuis maintenant quelques années, un certain nombre de travaux en Synthèse d'Images ont porté sur une résolution adaptative de différents problèmes. En simulation de l'éclairage par exemple, une méthode hiérarchique permet de focaliser les efforts de calcul sur les zones à forte variation de luminosité. Un résultat précis est alors atteint après une succession de raffinements locaux de solutions approximatives. Ce concept *adaptatif*, qui consiste à ne pousser les calculs que dans les endroits où il y en a besoin, n'a jamais été exploité en animation d'objets déformables, et cela pour une raison simple : il est très difficile de simuler une même matière à différents niveaux locaux de résolution tout en conservant les mêmes propriétés physiques de déformation.

### 6.1 Introduction

Comme nous l'avons vu précédemment, la plupart des modèles d'objets très déformables utilisent des systèmes de particules animés par des lois physiques simplifiées. Ils évitent ainsi à l'animateur la lourde tâche de la spécification d'images clés. Cependant, trouver les valeurs du modèle physique qui entraîneront l'effet souhaité est souvent non trivial. Mains essais sont nécessaires afin d'obtenir une animation stable et finalisée. La difficulté de la mise au point est accrue par le fait que le nombre de particules utilisées et le pas de temps d'intégration choisi influent grandement sur le résultat obtenu, et de petites variations de ces valeurs peuvent entraîner des changements imprévisibles du comportement simulé.

Ce chapitre présente une tentative d'allègement du travail, et de l'animateur, et de l'ordinateur. Conçu pour des substances à densité constante pouvant aller de quasi-solides à liquides, le modèle présenté simule en effet un comportement donné indépendamment de la discrétisation spatiale et temporelle : ces résolutions seront automatiquement ajustées durant la simulation de façon à optimiser les performances (et donc, réduire les temps de calcul) ainsi qu'assurer une meilleure stabilité.

### 6.1.1 Défauts des modèles existants

Les systèmes de particules sont le seul modèle disponible en Synthèse d'Images pour qui souhaite simuler de la matière pouvant se séparer ou se réorganiser de différentes manières. Ces modèles discrétisent la matière en éléments de masses auxquels sont appliquées des forces d'action-réaction. Ces forces sont souvent dérivées du modèle microscopique dit de Lennard-Jones, qui décrit la force moyenne entre deux atomes en fonction de la distance les séparant. Elles sont pourtant utilisées à une échelle macroscopique.

Malgré une extrême simplicité d'implémentation, générer des simulations complexes à l'aide d'un système de particules n'est pas aisé. En premier lieu, la discrétisation spatiale, appelée *granularité* du modèle, doit être choisie *a priori*. Cette granularité devant être peu apparente lors du rendu final, il en résulte souvent un nombre très élevé de particules, et donc des temps de calcul en proportion, y compris pendant les étapes de l'animation où une telle discrétisation ne serait pas nécessaire. Le pas de temps est aussi délicat à choisir, puisque trop grand, il entraîne des instabilités, alors que trop faible, il fait perdre un temps précieux. Enfin, le réglage des forces d'interaction et de friction demande beaucoup de savoir-faire, principalement à cause de la grande sensibilité du résultat à la granularité et au pas de temps choisis.

En d'autres termes, le principal défaut des systèmes de particules conventionnels est que les discrétisations spatiale et temporelle restent constantes durant une animation. En plus d'être non intuitives à déterminer, ces discrétisations fixes sont loin d'être optimales pour des animations complexes.

### 6.1.2 Apparition de modèles adaptatifs

Des modèles adaptatifs sont de plus en plus développés en Synthèse d'Images. Ils sont l'expression de l'idée clé qui consiste à *automatiquement concentrer les efforts là où il y en a le plus besoin*. Ces méthodes raffinent ou simplifient un modèle de façon adaptative afin de gagner en précision tout en économisant les calculs inutiles. Ce paradigme très général est par exemple depuis longtemps utilisé avec succès en illumination globale. La technique de radiosité a énormément gagné en efficacité avec les algorithmes hiérarchiques qui subdivisent ou regroupent en clusters les éléments surfaciques pour assurer une précision donnée en un faible coût de calcul, et ceci de façon automatique [HSA91, Sil95]. En visualisation aussi, des polygonalisations adaptatives d'objets s'obtiennent en simplifiant localement leur maillage, dans les régions peu ou pas vues par l'utilisateur [Hop96, CVM<sup>+</sup>96].

Une approche adaptative a aussi été développée pour l'échantillonnage interactif des surfaces implicites [WH94], comme nous l'avons vu au paragraphe 3.3.2. Un ensemble de points d'échantillonnage se repoussant sont contraints de rester sur une iso-surface, et se répartissent rapidement grâce à des rayons de répulsion adaptatifs. L'équilibre est atteint rapidement après une série de dédoublements et de suppressions automatisés de certains de ces points.

Par contre, très peu nombreux sont les modèles proposant des simulations interactives en animation de modèles déformables. Récemment, un essai intéressant de résolution adaptative a été proposé dans le cadre de la simulation de tissus [HPH96]. Un réseau de masses-ressorts modélisant un morceau de tissu se raffine localement si deux ressorts forment un angle excédant une tolérance donnée, ce qui produit une description plus fine de la forme. La mise en œuvre

passer par l'ajout de masses, l'ajout de ressorts munis de raideurs doubles, et la division du pas de temps de simulation pour toutes les masses. Cependant, un certain nombre de problèmes sont laissés de côté. Par exemple, cette méthode ne garantit pas un même comportement physique pour deux niveaux de raffinement distincts. De plus, l'ajout de masse entraîne une augmentation du poids global, ce qui n'est pas sans amener d'inconvénients si le tissu était traîné sur le sol : il serait tout simplement plus lourd ! Pour finir, la simplification d'un tissu raffiné n'est pas traitée.

Toujours pour des matières structurées, une approche hiérarchique a été présentée dans le cadre de simulation par éléments finis [AH97]. L'objet à simuler est d'abord maillé sommairement, puis chaque maille est elle-même maillée plus finement. Par analogie avec les circuits électriques, on pré-calculer comment le maillage sommaire influence chaque sous-maillage fin en trouvant la matrice de raideur équivalente aux noeuds en commun entre les deux niveaux de discrétisation. Il est par la suite possible de simuler l'ensemble en gardant le maillage sommaire partout sauf dans la zone concernée par les déformations en cours : grâce au découplage effectué, la zone fine peut être simulée à des fréquences plus élevées que le maillage sommaire. Par contre, il existe des problèmes de recollement entre maillages grossiers et raffinés, et l'approximation induite par ce modèle n'est pas vraiment claire, vu par exemple que le modèle raffiné n'influence en rien le modèle sommaire. De plus, une seule maille peut être raffinée à la fois, ce qui limite fortement l'aspect hiérarchique de l'approche.

### 6.1.3 Nécessité d'un modèle adaptatif

Nous souhaitons, dans le cadre de ce document, animer des matières hautement déformables. Or, les approches hiérarchiques vues au paragraphe précédent ne peuvent convenir : la définition hiérarchique de la subdivision de la matière interdit les changements de topologie et les fortes déformations en général. Il s'agit en l'occurrence de définir un modèle *adaptatif*, qui permette des subdivisions ou des simplifications locales de la matière, *sans conserver de hiérarchie puisque la matière peut se réorganiser n'importe comment*.

Nous avons vu au paragraphe 6.1.1 que les critères souhaités pour un modèle particulaire plus efficace et plus maniable pouvaient être formulés comme suit :

- Nécessité de paramètres macroscopiques, facilement réglables,
- Comportement physique indépendant de la granularité,
- Discrétisations spatiale et temporelle adaptées localement, afin de localiser les calculs et optimiser le modèle.

Nous allons donc exposer dans ce chapitre un nouveau modèle déformable que nous avons mis au point en partant de ces critères de qualité. Nous verrons qu'en nous aidant d'un formalisme physique peu usité, il est possible de formuler un modèle particulaire en terme de *densité*, et ce indépendamment de toute discrétisation. Puis nous détaillerons comment ce modèle se prête particulièrement bien à une simulation adaptative à la fois en espace (subdivisions et regroupements de particules) et en temps (pas de temps d'intégration individuels).

## 6.2 Un formalisme Lagrangien nommé SPH

Le formalisme utilisé dans ce chapitre dérive des SPH (*Smoothed Particles Hydrodynamics*), modèle utilisé à l'origine par des astrophysiciens pour des simulations de fluides cos-



miques [Mon92]. Cette approche générale a aujourd'hui diverses applications en physique, depuis des simulations astronomiques aux problèmes de résistance des matériaux. Une version simplifiée a d'ailleurs déjà été utilisée en Synthèse d'Images pour la simulation de phénomènes gazeux [SF93, SF95].

### 6.2.1 Discrétisation de l'espace : Euler vs. Lagrange

La matière est un ensemble infini d'éléments de masse infinitésimaux. Toute simulation doit donc passer par une *discrétisation*, c'est-à-dire ne considérer qu'un nombre fini de points d'échantillonnage. Souvent, et en hydrodynamique particulièrement, les techniques de simulation classiques de fluides utilisent une discrétisation de type Eulerienne. Elles partitionnent l'espace par une grille fixe de voxels, et étudient ce qui pénètre et ce qui ressort de chacun des voxels au cours du temps. Ces techniques sont bien maîtrisées à l'heure actuelle, et permettent de faire des simulations très réalistes [FM96, FM97]. Des méthodes adaptatives, raffinant localement la grille, ont même été mises au point et donnent satisfaction aux physiciens désireux de simuler des phénomènes d'écoulements tourbillonnaires par exemple. Cependant, cette approche ne convient pas parfaitement à la Synthèse d'Images. La gestion des collisions avec des objets solides ou déformables est problématique : on ne connaît effectivement pas la localisation exacte de la matière. Seuls les champs de densité et de vitesse permettent d'avoir une connaissance sur l'emplacement du matériau, mais une représentation fine nécessitera par exemple de positionner une multitude de marqueurs au sein de la matière et les faire suivre le champ de vitesse.

Une alternative à cette technique est d'utiliser une approche Lagrangienne, comme les SPH : cette fois, on échantillonne directement le fluide par un ensemble de masses appelées *particules*, dont on suit l'évolution dans l'espace au cours du temps<sup>1</sup>.

Une particule  $j$  représente un petit volume de matière de masse fixe  $m_j$  à une position courante  $\mathbf{X}_j$ . En tant que point d'échantillonnage, elle porte diverses valeurs de champs physiques, comme sa vitesse  $\mathbf{v}_j$  et sa densité de masse  $\rho_j$ . Selon les applications, la pression ou la température peuvent être aussi intéressantes.

### 6.2.2 Formulation discrète de champs continus

Pour être représentative d'un champ continu  $f$ , la valeur  $f(\mathbf{X})$  portée par une particule en  $\mathbf{X}$  doit représenter une moyenne macroscopique des valeurs de cette fonction sur un petit volume autour de la particule. Ainsi, la valeur du champ est lissée par le filtrage des fréquences élevées par rapport à la taille de la particule. Cette valeur moyenne locale, notée  $\langle f(\mathbf{X}) \rangle$ , peut être écrite comme la convolution de  $f$  par un noyau de filtrage comme suit :

$$\langle f(\mathbf{X}) \rangle = \int_V f(\mathbf{X}') W_h(\mathbf{X} - \mathbf{X}') d\mathbf{X}' \quad (6.1)$$

où  $W_h$  est un filtre de lissage dont le paramètre  $h$ , appelé taille de lissage, contrôle l'étendue du filtrage. L'intégrale de  $W_h$  doit être égale à 1 quelque soit  $h$ , et  $W_h$  doit tendre vers l'impulsion de Dirac quand  $h$  tend vers zéro. En pratique, ces intégrales sont approximées par des sommes discrètes. Afin d'intégrer sur la masse et non sur le volume, la densité de masse  $\rho$  est d'abord

---

1. Quelques détails sur les différences entre ces deux approches duales peuvent être trouvés dans l'annexe A.

intégrée dans l'équation intégrale 6.1 :

$$\begin{aligned} \langle f(\mathbf{X}) \rangle &= \int_V \frac{f(\mathbf{X}')}{\rho(\mathbf{X}')} W_h(\mathbf{X} - \mathbf{X}') \rho(\mathbf{X}') d\mathbf{X}' \\ &= \int_M \frac{f(\mathbf{X}')}{\rho(\mathbf{X}')} W_h(\mathbf{X} - \mathbf{X}') dm \end{aligned} \quad (6.2)$$

Une approximation de l'équation précédente obtenue par une méthode de Monte-Carlo grâce aux valeurs des champs portées par les particules nous donne aisément :

$$\langle f(\mathbf{X}) \rangle \simeq \sum_j m_j \frac{f_j}{\rho_j} W_h(\mathbf{X} - \mathbf{X}_j) \quad (6.3)$$

où  $f_j$  est un raccourci d'écriture pour  $f(\mathbf{X}_j)$ . Le schéma de lissage fournit aussi une approximation du gradient des champs. En effet, une intégration par partie nous donne :

$$\int_V \nabla f(\mathbf{X}') W_h(\mathbf{X} - \mathbf{X}') d\mathbf{X}' = \int_V f(\mathbf{X}') \nabla W_h(\mathbf{X} - \mathbf{X}') d\mathbf{X}'. \quad (6.4)$$

De cette dernière égalité résulte qu'une approximation du gradient d'un champ peut s'écrire :

$$\langle \nabla f(\mathbf{X}) \rangle \simeq \sum_j m_j \frac{f_j}{\rho_j} \nabla W_h(\mathbf{X} - \mathbf{X}_j) \quad (6.5)$$

Si nous disposons d'un ensemble, même désordonné, de points d'échantillonnage dans la matière à simuler, on peut, à l'instar des méthodes à grilles régulières, calculer des champs et leurs dérivées de façon simple. Ainsi, les valeurs de densité aux particules seront déterminées en fonction de l'emplacement de leurs masses voisines, ce qui paraît intuitivement raisonnable.

### 6.2.3 Équation d'état - Forces de pression

Une définition macroscopique d'un modèle déformable peut être donnée à travers l'utilisation d'un champ de pression  $P$ , duquel sont dérivées des forces conservatives, dites *forces de pression*. La façon dont la pression évolue dans le milieu est alors définie par une *équation d'état*. Pour un gaz parfait par exemple, on sait que le produit de la pression  $P$  et du volume  $V$  est constant à une température fixée ( $PV = kT$ ).

Les forces de pression étant proportionnelles au gradient de la pression [Bat73], la force de pression exercée sur une particule  $i$  s'écrit alors :

$$\mathbf{F}_i^{\nabla P} = -\nabla P_i dV = -m_i \frac{\nabla P_i}{\rho_i} = \frac{m_i}{\rho_i} \sum_j m_j \frac{P_j}{\rho_j} \nabla W_h(\mathbf{X} - \mathbf{X}_j).$$

Mais une observation rapide nous montre que le principe d'action-réaction entre deux particules ne sera pas vérifié, puisque il n'y a aucune raison dans le cas général pour que  $m_i \nabla P_i / \rho_i$  soit égal à  $m_j \nabla P_j / \rho_j$  pour  $i \neq j$ . Pour symétriser la formulation des forces de pression, il suffit d'utiliser la règle de dérivation suivante :  $\nabla P / \rho = \nabla (P / \rho) + P \nabla \rho / \rho^2$ . Cela nous amène à écrire :

$$\frac{\nabla P_i}{\rho_i} = \sum_{j \neq i} m_j \frac{P_j}{\rho_j^2} \nabla_i W_h^{ij} + \frac{P_i}{\rho_i^2} \sum_{j \neq i} m_j \nabla_i W_h^{ij}$$

où la notation  $\nabla_i W_h^{ij}$  représente le gradient de  $W_h(\mathbf{X}_i - \mathbf{X}_j)$  par rapport à la position de la particule  $i$ . Finalement, l'évaluation des forces de pression agissant sur cette même particule s'écrit :

$$\mathbf{F}_i^{\nabla P} = -m_i \sum_{j \neq i} m_j \left( \frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_h^{ij} \quad (6.6)$$

en remplaçant  $P_i$  et  $P_j$  par leurs valeurs respectives en fonction de l'équation d'état.

### 6.2.4 Discussion

Une remarque importante à faire est que l'équation précédente se trouve être une somme de forces modélisant des interactions binaires entre les particules :

$$\mathbf{F}_i^{\nabla P} = \sum_{j \neq i} \mathbf{F}_{ij}^{\nabla P} \quad \text{avec} \quad \mathbf{F}_{ij}^{\nabla P} = -m_i m_j \left( \frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_h^{ij}. \quad (6.7)$$

Les particules s'appuyant sur une formulation SPH peuvent donc s'implémenter exactement comme les systèmes de particules conventionnels en Synthèse d'Images : à chaque pas de temps, les forces internes d'interaction ainsi que les forces extérieures comme la gravité sont évaluées; puis, pour chaque particule, l'équation du mouvement est intégrée. La grande différence est que les forces internes n'ont pas été réglées arbitrairement par l'utilisateur : *elles dérivent de l'équation d'état qui gouverne les variations de pression*. Par exemple, dans le cas d'un gaz parfait, la pression est proportionnelle à la densité  $\rho$  du gaz. L'équation (6.6) ne produira donc que des forces répulsives, confortant le fait qu'un gaz a tendance à occuper tout l'espace libre.

Ainsi, *l'approche SPH nous permet d'approximer un matériau défini indépendamment de toute discrétisation spatiale ou temporelle*. Autrement dit, une particule n'est plus un élément unitaire de matière, elle représente un *échantillon de matière* permettant de connaître des informations sur la densité locale par exemple. De plus, la technique de Monte-Carlo dont elle dérive nous assure une stabilité croissante pour un nombre de particules croissant, aspect dont on ne peut négliger l'importance. Enfin, il faut signaler qu'il a été prouvé récemment l'équivalence entre les SPH et la méthode d'éléments finis de Galerkin appliquée en temps et en espace [Dil96].

## 6.3 Un nouveau modèle déformable à base de densité

Nous allons ici décrire pourquoi et comment adapter l'approche SPH à la simulation de matériaux déformables de consistance uniforme. Nous ferons dans toute cette partie l'hypothèse que toutes les particules sont de même masse  $m_j$ , comme dans les systèmes de particules conventionnels.

### 6.3.1 Définition d'une équation d'état

Notre but ici est de simuler une substance hautement déformable qui revienne à une densité constante dès que les forces extérieures sont nulles. En conséquence, nous choisissons de définir l'équation d'état, qui gouverne les variations de pression à l'intérieur de notre matériau, par :

$$P = k(\rho - \rho_0). \quad (6.8)$$

Cette équation est assez explicite : elle génère des forces de pression qui tendent à restaurer une densité de masse  $\rho_0$  dès que la densité locale s'écarte de cette valeur. Le paramètre  $k$  contrôle l'amplitude des forces de restauration, et est donc tout à fait analogue à une raideur. D'ailleurs, il est à noter que la vitesse maximale de propagation des ondes de pression dans un milieu, appelé communément *vitesse du son*, vérifie l'équation  $c = \sqrt{\partial P / \partial \rho}$  comme démontré en annexe B. Et justement, dans notre cas, la vitesse du son  $c$  se calcule aisément :

$$c = \sqrt{k} \quad (6.9)$$

On retrouve donc que le matériau propagera d'autant plus vite les déformations que la raideur sera grande.

Ce choix d'équation d'état garantit que le matériau tendra toujours à revenir au même volume d'équilibre, égal à la masse totale divisée par  $\rho_0$ , ce qui n'était pas évident dans notre ancien modèle [DG95]. Contrairement au cas des gaz parfaits, les forces d'interaction découlant de cette équation d'état pourront aussi bien être attractives que répulsives : par conséquent, le matériau simulé aura bien une cohésion interne.

Il reste encore à ajouter une force dissipative modélisant la viscosité. Pour cela, nous avons utilisé l'approche la plus usitée dans la littérature SPH. De façon analogue à la force de dissipation utilisée dans notre approche précédente (voir paragraphe 5.1.2), la viscosité s'écrit en fonction des vitesses relatives entre chaque paire de particules :

$$D_i = -m_i \sum_{j \neq i} m_j \Pi_{ij} \nabla_i W_h^{ij} \quad (6.10)$$

avec :

$$\Pi_{ij} = \begin{cases} \frac{-c\mu_{ij} + 2\mu_{ij}^2}{\bar{\rho}_{ij}} & \text{si } \mu_{ij} < 0 \\ 0 & \text{si } \mu_{ij} \geq 0 \end{cases} \quad \mu_{ij} = h \frac{\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{\mathbf{r}_{ij}^2 + h^2/100}$$

$$\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j \quad \mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j \quad \bar{\rho}_{ij} = (\rho_i + \rho_j)/2$$

Ce choix assure une bonne approximation de la viscosité naturelle d'un matériau, et de plus est réglable à souhait. On pourra au passage remarquer que le deuxième terme de  $\Pi_{ij}$  est comparable à un terme de viscosité numérique comme abordé dans l'annexe C.

### 6.3.2 Choix d'un filtre de lissage

Le filtre utilisé pour le calcul des valeurs discrètes des champs, et en particulier pour l'évaluation des forces de pression dans l'équation (6.6), doit être choisi avec discernement. Dans la littérature SPH [HK92, Mon92], une approximation spline de la Gaussienne, avec un support fini de rayon  $2h$ , est classiquement utilisée. Pourtant, ce choix de filtre ne donnerait pas de bons résultats pour notre application. En effet, la valeur des forces d'interaction étant proportionnelle au gradient du filtre, une dérivée tendant vers zéro en zéro provoquerait une force tendant vers zéro quand deux particules s'approchent, ce qui se traduirait par des regroupements de particules.

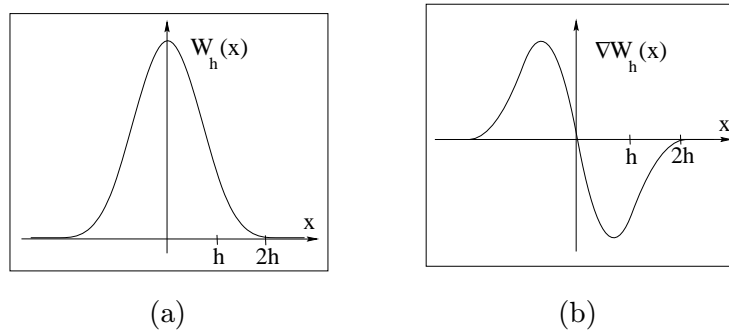


FIG. 6.1 - Le filtre Gaussien approximé par une spline (a), et sa dérivée en 1D (b).

Nous avons donc préféré utiliser un filtre qui n'a pas cette particularité, comme le montre la figure 6.2 :

$$W_h(\mathbf{X}) = \frac{15}{\pi(4h)^3} \begin{cases} (2 - \frac{\|\mathbf{X}\|}{h})^3 & \text{if } 0 \leq \|\mathbf{X}\| \leq 2h \\ 0 & \text{if } \|\mathbf{X}\| > 2h \end{cases} \quad (6.11)$$

Comme la Gaussienne ou le filtre spline, ce filtre garantit une bonne stabilité d'intégration puisque sa transformée de Fourier chute rapidement en fréquence [Mon92], limitant donc les oscillations parasites. De plus, la force  $\mathbf{F}_{ij}^{\nabla P}$  (6.6) qu'il produit entre deux particules, comme tracé en figure 6.3, a une allure tout à fait similaire à la classique force de Lennard-Jones, fort souvent utilisée par les systèmes de particules habituels. Il est donc normal que le type de substances simulées soit qualitativement le même. Au lieu d'une force fixe entre chaque couple de particules, *cette approche calcule les forces d'interaction en fonction de milieu avoisinant*. C'est en cela que ce modèle particulière est totalement différent des systèmes conventionnels. Les forces d'interaction *s'adaptent* à leur environnement, et cet avantage, comme nous le verrons plus tard, va nous permettre de mettre au point une simulation adaptative.

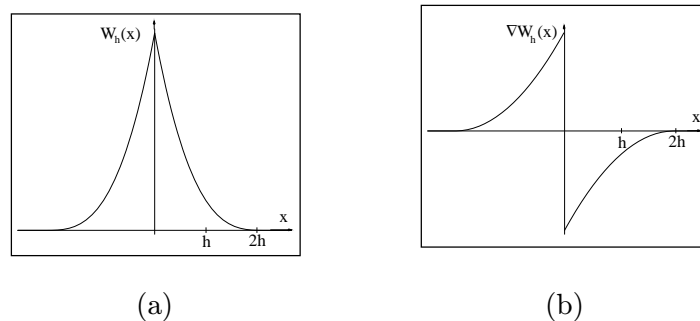


FIG. 6.2 - La fonction de lissage (a) et son gradient en 1D (b)

Le paramètre  $h$  doit lui aussi être bien choisi. Alors qu'un filtre trop large adoucirait trop les variations des champs, un filtre trop étroit ajouterait des oscillations. Notons d'ailleurs que  $h$  contrôle le rayon d'influence des particules, puisque deux particules distantes de plus de  $2h$  n'interagissent pas. Le volume réel qu'occupe une particule de masse  $m_j$  dans un matériau de

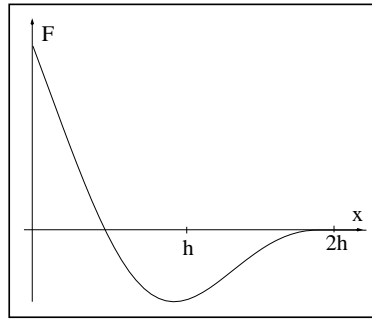


FIG. 6.3 - Une force d'interaction typique, appliquée sur une particule  $i$  par une particule  $j$ , en fonction de la distance entre les deux particules

densité  $\rho_0$  est  $m_j/\rho_0$ . En considérant une particule isotrope, ce volume peut être vu comme une sphère de rayon  $r_j$  telle que :

$$\frac{4}{3}\pi r_j^3 = \frac{m_j}{\rho_0}. \quad (6.12)$$

Nous avons donc choisi de prendre  $h$  proportionnel à  $r_j$ , de telle sorte que cela procure un nombre de voisins suffisant pour chaque particule. Cela signifie d'ailleurs que plus la particule est lourde, plus elle représentera une valeur moyennée, ce qui est intuitivement raisonnable. Ainsi, on définit un paramètre  $\alpha$  sans dimension, et on choisit la formulation suivante :

$$h = \alpha \sqrt[3]{m_j/\rho_0}.$$

En pratique, on souhaite qu'une particule ait au plus une vingtaine de particules voisines, ce qui résulte, en faisant l'hypothèse de particules sphériques empilées au mieux, un paramètre  $\alpha$  entre 1.15 et 1.3.

### 6.3.3 Contrôle différentiel de la densité

Calculer la pression par l'équation 6.8 demande une évaluation de la densité de masse locale en chaque particule. Une utilisation directe de l'équation 6.3 pour la densité nous donne :

$$\rho_i = \sum_j m_j W_h(\mathbf{X}_i - \mathbf{X}_j)$$

Cette expression, bien que fort simple, n'est pourtant pas appropriée en pratique. La densité de masse étant évaluée selon le nombre de voisins, l'approximation de densité chutera sur les bords en cas de distribution uniforme. Cet effet de bord amènera des configurations stables tout à fait non uniformes : les particules auront tendance à s'entasser près de la surface du matériau pour compenser le manque de voisins sur les bords.

Une méthode alternative existe cependant. Il faut savoir que la densité dans un matériau vérifie l'équation suivante, dite de *continuité*<sup>2</sup> :

$$\dot{\rho} = -\rho \operatorname{div} \mathbf{v} \quad (6.13)$$

2. Pour plus de détails, voir la démonstration dans l'annexe B

où  $\operatorname{div} \mathbf{v}$  représente la divergence de la vitesse, qui peut tout simplement s'exprimer par :

$$(\operatorname{div} \mathbf{v})_i = \frac{1}{\rho_i} \sum_{j \neq i} m_j (\mathbf{v}_j - \mathbf{v}_i) \cdot \nabla_i W_h^{ij}. \quad (6.14)$$

L'équation de continuité, exprimant la conservation de masse [Bat73], offre ainsi un contrôle différentiel de la densité : après avoir défini une valeur initiale de la densité pour chaque particule, il suffit d'intégrer cette relation différentielle au cours du temps pour disposer d'une densité valide, sans problème d'imprécision sur les bords.

### 6.3.4 Implémentation de base

Le nouveau modèle déformable présenté ci-dessus n'a que peu de paramètres : seuls la densité au repos, un paramètre de raideur, et un paramètre de viscosité doivent être fixés par l'utilisateur. Ce modèle de base est de plus très simple à implémenter : une fois que l'on a fixé la masse  $m_j$  et la densité initiale  $\rho_0$  de toutes les particules, les forces de pression et de viscosité sont calculées à chaque pas de temps, ainsi que les variations de densité. Le mouvement est alors calculé classiquement en intégrant vitesses et positions [DG96]. Si une grille régulière est utilisée pour le stockage des particules, le coût total de l'algorithme est linéaire en le nombre de particules : en effet, la recherche des plus proches voisins d'une particule se fait en temps constant grâce à cette structure de donnée [MP89].

## 6.4 Vers une simulation adaptative en espace-temps

Puisque les caractéristiques de la substance simulée ne dépendent d'aucune hypothèse quant à la discrétisation spatiale ou temporelle, *différents niveaux d'approximation de la même matière* peuvent être obtenus simplement en changeant la granularité du modèle courant. C'est un avantage prépondérant par rapport aux systèmes de particules classiques.

### 6.4.1 Idée de base d'un modèle adaptatif

Produire des déformations complexes demande un nombre élevé de particules. Assurer alors un pas de temps qui évite toute divergence du système demande des temps de calcul très importants. Pour alléger cette lourdeur des calculs, des *simulations adaptatives* seraient très profitables. De plus, cela libérerait l'utilisateur du dilemme du choix de la discrétisation spatiale et temporelle, tâche souvent ingrate.

Notre idée de base est donc de subdiviser automatiquement les particules qui se trouvent dans des régions subissant de fortes variations de pression, et de regrouper au contraire celles qui se trouvent dans des régions très stables. Une région au repos ne nécessite effectivement que peu de points d'échantillonnage pour la définir, alors qu'une région soumise à une forte déformation nécessite un sur-échantillonnage pour à la fois bien intégrer et décrire les mouvements : on optimise ainsi les calculs tout en automatisant l'adaptation de la discrétisation.

Cette idée n'est pas sans rappeler l'approche de Witkin et Heckbert pour l'échantillonnage adaptatif d'une surface, vue en détail au paragraphe 3.3.2. Pourtant, en comparant attentivement, on s'aperçoit que cette approche est tout à fait le contraire de la nôtre : alors que leur

système de particules cherche à avoir une discrétisation constante, nous cherchons à l'adapter en fonction du mouvement. Ce qui se traduit par exemple par le fait que dans leurs simulations, ce sont les particules les plus grosses qui bougent le plus vite, contrairement à ce que l'on souhaite dans notre application.

De plus, la comparaison ne peut guère être poussée plus loin : nous sommes dans un contexte d'animation à base de physique, et une méthode de suicides des particules par exemple ne peut être valide puisqu'elle revient purement et simplement à supprimer de la matière.

### 6.4.2 Intêret de la formulation SPH

Dans une optique de modèle adaptatif, on s'aperçoit rapidement que l'utilisation des systèmes de particules conventionnels serait très délicate : comment garantir le même comportement à différents niveaux de raffinement avec des forces d'interaction figées ?

#### La densité, grandeur intensive

Une grandeur physique est dite *intensive* si, quand deux systèmes strictement identiques sont ajoutés, cette grandeur ne change pas de valeur. Tel est le cas par exemple pour la température (deux milieux de même température mélangés resteront à la même température), ou la pression, la densité. Dans le cas contraire, on parle de grandeur *extensive*, pour la masse, la vitesse, ou le volume par exemple.

Or justement, dans le cas d'un modèle adaptatif, on doit remplacer localement un système par un autre, raffiné. Si l'on dispose d'un système de particules classique, seules la masse et les vitesses peuvent être réglées de telle sorte que les deux systèmes, l'ancien et le raffiné, soient à peu près équivalents. Mais alors, les forces d'interaction, de formes fixées, risquent fort de totalement changer le comportement du système.

Pour notre modèle de particules, les forces d'interaction ne sont pas fixes, elles dépendent des variations locales de densité. Donc, si l'on peut aussi contraindre le modèle raffiné à conserver sa densité locale, les deux systèmes seront vraiment équivalents dans le sens où les forces de pression y seront strictement identiques. La grandeur intensive macroscopique *densité* est donc d'un intérêt capital pour les systèmes adaptatifs.

#### Euler et Lagrange

Autant les simulations Euleriennes peuvent très facilement calculer la densité d'un milieu, autant les simulations Lagrangiennes type particules ne le font quasiment jamais. Trouver la masse locale par unité de volume dans un ensemble de masses *désordonnées* est en effet délicat.

C'est là l'un des avantages majeurs du formalisme SPH : permettre une simulation Lagrangienne tout en pouvant exprimer des grandeurs typiquement Euleriennes comme la densité. On conserve alors les qualités des deux types d'approches. Ce modèle introduit au paragraphe 6.3 semble donc le meilleur candidat pour y intégrer une approche adaptative. Cependant, les adaptations en temps et en espace que nécessite ce genre d'approche vont nous obliger à calculer à chaque instant un pas de simulation avec des particules de tailles différentes en présence. Le paragraphe suivant présente les modifications nécessaires pour gérer ce cas.



### 6.4.3 Simulation avec des particules non homogènes

Jusqu'à présent, nous utilisons une largeur de filtre  $h$  fixe. Pourtant, il semble qu'un filtrage grossier suffise amplement dans les régions stables, alors qu'un filtre étroit décrit mieux des changements de pression importants. De plus, avoir une description plus précise (car sur-échantillonnée) dans les régions fortement instables et une description plus vague (car sous-échantillonnée) dans les régions stables offre *une description optimisée d'un phénomène*. L'échantillonnage de la matière, caractérisé par le nombre de particules et leurs masses respectives, doit alors changer au cours du temps, rendant le système de particules non homogène car composé de particules de masses différentes à chaque instant donné.

Cette remarque soulève deux questions qui doivent être traitées avant de s'attaquer au schéma proprement dit de l'algorithme adaptatif. Premièrement, les équations gouvernant la simulation doivent être étendues aux systèmes de particules non homogènes. Deuxièmement, la discrétisation spatiale ne peut (ou plutôt ne doit) pas se faire indépendamment de la discrétisation temporelle : comme on le verra, cela pourrait créer des comportements aberrants et des instabilités.

La nouvelle formulation de notre système de particules est tout aussi valable pour des échantillonnages non uniformes de la masse, grâce à la formulation des forces de pression dérivant directement de l'équation d'état et de la discrétisation locale actuelle. Seules quelques adaptations doivent être faites pour étendre la méthode vue précédemment à un système de particules non homogène.

#### Interprétations duales : *Gathering et Shooting*

Maintenant que chaque particule a sa propre masse  $m_j$ , et donc sa propre largeur de filtre toujours définie par  $h_j = \alpha \sqrt[3]{m_j/\rho_0}$ , nous devons regarder comment les formules utilisant  $h$  doivent être modifiées. Prenons la formule 6.6 comme exemple. Si l'on change le  $W_h$  par un  $W_{h_i}$ , cela signifie que l'estimation du gradient local de pression au niveau de la particule  $i$  sera obtenue en pondérant les valeurs de pression voisines selon le filtre  $W_{h_i}$  (voir figure 6.4). Par contre, si l'on met cette fois un  $W_{h_j}$ , le gradient local sera estimé d'après les contributions de pression induites par les particules voisines. On peut alors utiliser les termes *gathering* et *shooting*, utilisés en rendu d'images, pour décrire ces deux formulations. En effet, dans le cas "gathering", une particule collecte les valeurs de pression voisines et les pondère selon sa propre distribution de masse. Dans le cas contraire, la particule somme directement les contributions données par les particules voisines. On retrouve ainsi les deux méthodes de simulation en radiosité pour l'échange d'énergie lumineuse entre surfaces : soit un élément de surface somme les contributions des autres surfaces, soit il distribue sa valeur de radiosité aux autres [SP94].

#### Forces de pression symétriques

Les deux interprétations précédentes étant autant valables l'une que l'autre, nous avons choisi de combiner *shooting* et *gathering*. La force de pression appliquée en la particule  $i$  peut alors être définie comme la moyenne des forces calculées par l'équation 6.6 avec les deux interprétations possibles, ce qui nous donne :

$$\mathbf{F}_i^{\nabla P} = -km_i \sum_{j \neq i} m_j \left( \frac{\rho_i - \rho_0}{\rho_i^2} + \frac{\rho_j - \rho_0}{\rho_j^2} \right) \frac{\nabla_i (W_{h_i}^{ij} + W_{h_j}^{ij})}{2} \quad (6.15)$$

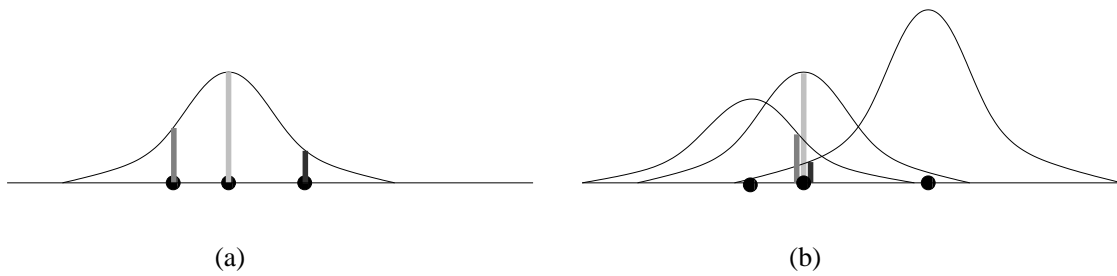


FIG. 6.4 - Illustration de l'interprétation (a) Gathering, et (b) Shooting

Ainsi, et contrairement à ce qu'aurait donné une interprétation purement “shooting” ou purement “gathering”, cette formulation a l'avantage de vérifier la loi d'action/réaction de par sa symétrie. L'ensemble des particules interagissant avec la particule  $i$  est donc maintenant caractérisé par l'ensemble de  $j$  vérifiant :  $\|\mathbf{X}_i - \mathbf{X}_j\| < 2 \max(h_i, h_j)$ .

De façon similaire, le terme  $h$  est remplacé par  $(h_i + h_j)/2$  dans l'équation de viscosité. Une fois que ces deux changements ont été faits, le modèle déformable décrit dans la section précédente peut être appliqué sans plus de modifications.

#### 6.4.4 Corrélation entre espace et temps

Pour obtenir des résultats stables et fiables, les pas de temps et la résolution spatiale de notre modèle doivent être réglés de concert. En effet, des éléments de masse plus faibles, soumis aux mêmes forces de pression, devront avoir un pas de temps plus petit que les autres afin d'éviter toute divergence de l'intégration. Il faut donc définir une stratégie cohérente d'adaptation de l'espace et du temps qui nous garantisse à la fois efficacité et stabilité.

La discrétisation spatiale est cruciale aussi bien pour disposer d'un bon compromis entre précision et efficacité de l'approximation discrète des champs, que pour permettre une description adaptative. Nous proposons donc d'adapter en premier lieu la discrétisation spatiale en fonction des variations locales de pression et de la précision souhaitée des calculs. Puis, la discrétisation temporelle sera adaptée en conséquence pour assurer stabilité et fiabilité. Les deux prochaines sections détaillent les algorithmes utilisés pour cette stratégie.

## 6.5 Adaptation automatique de la discrétisation spatiale

La subdivision spatiale doit être adaptée localement selon la complexité des déformations qui se produisent. Dans notre modèle, des variations locales de densité impliquent des forces de recouvrement de densité, et donc du mouvement. Une discrétisation spatiale adéquate semble donc être une description optimisée de la densité de masse.

### 6.5.1 “Importance Sampling”

Le formalisme SPH repose en grande partie sur l'approximation d'intégrales par une méthode de Monte-Carlo (voir paragraphe 6.2.2). Plus précisément, une technique dite d'*importance sampling* [PTVF92] est utilisée, puisque la différentielle  $dV$  est transformée en différentielle  $dm$  en introduisant la densité de masse dans l'équation 6.2. L'approximation est alors, par analogie

avec l'intégration par la méthode des rectangles, *d'autant plus précise que les particules ont une densité quasi constante dans leur volume propre* (voir figure 6.5(a)).

### 6.5.2 Critères d'adaptation

Il semble donc que le critère de densité localement constante soit à la fois important en terme de qualité de discrétisation et en terme de précision de calcul. Il est donc tout désigné pour gouverner les adaptations de notre modèle : là où la densité varie beaucoup, de multiples "petites" particules sont nécessaires, alors que quelques "grosses" particules suffisent pour décrire une densité quasi constante. L'intérêt de cette stratégie apparaît clairement sur la figure 6.5 : pour un même nombre d'échantillons, l'intégrale sera bien mieux approximée par une distribution non uniforme adaptée.



FIG. 6.5 - (a) *Discrétisation uniforme* contre (b) *discrétisation adaptée*

### 6.5.3 Raffinement de la discrétisation

Une façon de vérifier le critère d'adaptation est de remplacer une particule  $i$  par plusieurs particules plus petites lorsque la différence de pression avec l'une de ses voisines est trop grande par rapport à son volume. Le critère de subdivision peut donc s'écrire :

$$\text{diviser la particule } i \text{ ssi: } \exists j \text{ voisin de } i / |\rho_j - \rho_i| \frac{m_i}{\rho_i} > \Delta \quad (6.16)$$

avec  $\Delta$  un seuil choisi par l'animateur. La subdivision de la particule se traduit par la suppression de la particule courante et la création de  $n$  particules qui ré-échantillonnent l'ancien volume propre de la particule défunte, comme le schématise la figure 6.6(a).

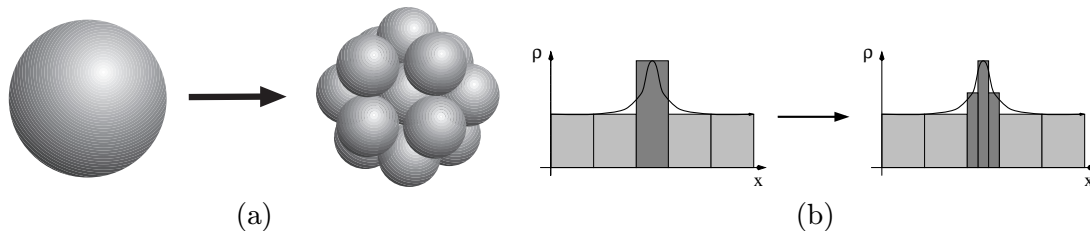


FIG. 6.6 - *Le raffinement spatial s'effectue en divisant une particule en plusieurs dans les régions où la densité varie beaucoup.*

### Mise en œuvre

Les conservations de la masse et de la quantité de mouvement imposent que les masses  $m'_j$  et les vitesses  $\mathbf{v}'_j$  des nouvelles particules créées doivent vérifier :

$$m'_j = \frac{m_i}{n} \quad \mathbf{v}'_j = \mathbf{v}_i$$

Quant aux  $h'_j$ , ils sont toujours calculés selon  $m'_j$ . Enfin, les densités  $\rho'_j$  sont interpolées linéairement entre la densité de la particule défunte et les densités de ses voisines. Ce processus peut être mieux appréhendé à l'aide du cas 1D représenté sur la figure 6.6(b).

Le contrôle différentiel de la densité rend ici encore un grand service. L'ajout de particules ne créera aucun changement brusque dans le mouvement : comme la masse est conservée et que la distribution de densité est affinée, les mouvements qui suivent la subdivision décrivent simplement mieux la déformation en cours.

#### 6.5.4 Simplification de la discrétisation

Les régions de densité presque constante sont très stables, puisque les déformations internes de notre modèle sont dues aux différences de pression. Il y a donc moyen d'économiser un certain nombre de particules en regroupant ces particules stables en une seule grosse particule. Le critère que nous utiliserons pour caractériser la stabilité d'une région est tout à fait le symétrique du précédent :

$$\text{regrouper toutes les voisines de } i \text{ ssi : } \forall j \text{ voisin de } i, |\rho_j - \rho_i| \frac{m_i}{\rho_i} < \delta \quad (6.17)$$

où  $\delta$  est un paramètre réglable par l'animateur.

#### Critère de sphéricité

Pourtant, dans le cas de la simplification, ce critère n'est pas suffisant. Comme le montre la figure 6.7(a), on ne peut regrouper tout ensemble stable en une seule particule : les particules étant isotropes, il faut que la région stable soit approximativement une sphère, au risque sinon de dégrader la description géométrique et inertielle localement.

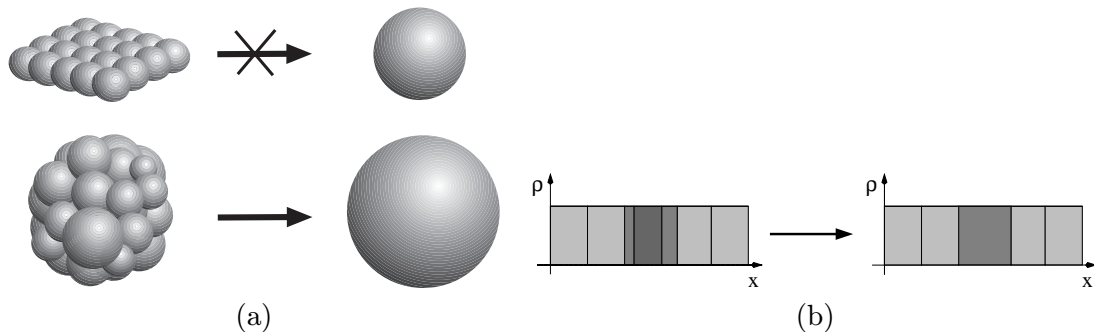


FIG. 6.7 - Une simplification peut se faire quand un groupe de particules stables est proche d'une forme sphérique

Un moyen efficace de vérifier cette propriété est de calculer la *matrice d'inertie locale*  $I_M$  pour cet ensemble de particules stables, en considérant les particules comme des masses ponctuelles. Si  $\tilde{\mathbf{X}} = (\tilde{x}, \tilde{y}, \tilde{z})$  est le barycentre du groupe de particules, la matrice s'écrit :

$$I_M = \begin{pmatrix} I_{Ox} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{Oy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{Oz} \end{pmatrix}$$

avec :

$$\begin{aligned} I_{Ox} &= \sum_j m_j ((y_j - \tilde{y})^2 + (z_j - \tilde{z})^2) & I_{Oy} &= \sum_j m_j ((x_j - \tilde{x})^2 + (z_j - \tilde{z})^2) \\ I_{Oz} &= \sum_j m_j ((x_j - \tilde{x})^2 + (y_j - \tilde{y})^2) & I_{xy} &= \sum_j m_j (x_j - \tilde{x})(y_j - \tilde{y}) \\ I_{yz} &= \sum_j m_j (y_j - \tilde{y})(z_j - \tilde{z}) & I_{xz} &= \sum_j m_j (x_j - \tilde{x})(z_j - \tilde{z}) \end{aligned}$$

Comme toute matrice d'inertie est symétrique définie positive, la matrice  $I_M$  est diagonalisable. Si les trois valeurs propres sont à peu près égales, la distribution de masse est quasi sphérique. On peut alors simplifier les particules en une seule particule équivalente, aussi bien en masse qu'en densité, et donc en volume. La mise en œuvre de ce critère a été optimisée, en évitant le calcul des valeurs propres : une fois la matrice d'inertie  $I_M$  trouvée, on en calcule simplement le déterminant et la trace. Puisque le déterminant est égal au produit des trois valeurs propres et que la trace est, quant à elle, égale à la somme des valeurs propres, nous vérifions simplement le critère suivant :

$$\left| \det(I_M) - \left( \frac{\text{trace}(I_M)}{3} \right)^3 \right| < \epsilon. \quad (6.18)$$

Ce critère est vérifié si et seulement si les trois valeurs propres, qui sont toujours positives, sont quasiment identiques. On détecte ainsi rapidement les configurations sphériques de particules.

### Mise en œuvre

La nouvelle particule est positionnée au centre de gravité des particules supprimées, sa masse est fixée à la somme des anciennes masses, sa vitesse calculée pour conserver la quantité de mouvement, et sa densité déduite de la nouvelle masse et de son encombrement spatial :

$$\begin{aligned} m'_i &\leftarrow \sum_j m_j \\ \mathbf{X}'_i &\leftarrow (\sum_j m_j \mathbf{X}_j) / m'_i \\ \mathbf{v}'_i &\leftarrow (\sum_j m_j \mathbf{v}_j) / m'_i \\ \rho'_i &\leftarrow \frac{m'_i}{\sum_j (m_j / \rho_j)}. \end{aligned}$$

Dans l'implémentation effective de nos particules adaptatives, nous testons les critères de simplification autour d'une particule  $i$  juste avant d'en intégrer les forces. Les recherches des plus proches voisins étant déjà faites, tester si une particule  $i$  doit *fusionner avec ses voisines* ne rajoute pas beaucoup de calculs : la comparaison des densités (équation 6.17) est faite dans une

boucle, en même temps que le calcul du barycentre de l'ensemble. Si aucun des tests de densité n'a échoué, nous éliminons les distributions clairement non sphériques en testant d'abord si le barycentre calculé est proche de la particule  $i$ . Si ce test est encore vérifié, nous calculons la matrice  $I_M$  et utilisons l'équation 6.18 pour décider définitivement d'effectuer ou non la simplification.

### 6.5.5 Critères annexes

Les raffinements et les simplifications peuvent être “forcés” en rajoutant d'autres critères. On peut par exemple ajouter un critère qui force le sur-échantillonnage lors d'une collision détectée. Toutefois, il y a fort à parier que de toute façon, la subdivision aurait eu lieu automatiquement peu de temps après, puisque le choc aurait automatiquement créé une sur-pression. Nous avons aussi utilisé un critère de plus : nous arrêtons la subdivision si les tailles des particules créées sont inférieures à un *grain* donné. Cela permet d'éviter les subdivisions infinies lors de gros chocs, et de garantir donc des temps de calcul raisonnables, quelque soit l'action simulée. Les deux critères combinés assurent par exemple qu'un matériau posé sur une surface trouée même finement pourra s'écouler si le grain du matériau est plus petit que le trou.

## 6.6 Adaptation automatique de la discrétisation temporelle

Comme nous l'avons exposé au paragraphe 6.4.4, notre stratégie consiste à adapter la discrétisation temporelle à l'actuelle discrétisation spatiale. Puisque l'évaluation des forces, qui inclut une recherche des plus proches voisins, est une partie très gourmande en temps machine (d'autant plus qu'un système inhomogène ne permet plus de bilan des forces en temps linéaire), il semble intéressant d'associer à chaque particule un *pas de temps individuel* plutôt que de choisir le plus petit pas de temps pour toutes.

Le plus petit pas de temps sera d'ailleurs atteint par les plus petites particules : leur masse étant très faible, leur accélération est très forte pour une force de pression donnée. Une intégration précise nécessite donc un pas de temps faible, à l'instar des hautes fréquences dans un signal.

### 6.6.1 Mécanisme global de l'adaptation

Au cours d'une animation, il est souhaitable qu'une particule n'intègre ses équations de mouvement seulement lorsque cela est nécessaire, selon son propre pas de temps actuel. Entre deux évaluations de forces, sa position et sa vitesse seront mises à jour à l'aide de la dernière accélération calculée, de telle sorte que les autres particules plus petites puissent effectuer leurs propres évaluations de forces en utilisant des positions valides de leurs voisines.

Un point de détail est que cette façon de faire ne vérifiera pas exactement le critère d'action-réaction, puisqu'une petite particule  $j$  ré-évaluera ses forces d'interaction chaque  $dt_j$ , alors que la force d'une de ses voisines  $i$ , plus grosse, sera considérée constante pendant un intervalle de temps  $dt_i > dt_j$ . Heureusement, comme le critère de sélection de pas de temps nous assurera une bonne approximation du mouvement pour ces particules, ce problème ne sera ni conséquent, ni perceptible en pratique.

On contraindra enfin les pas de temps à être égaux à la fréquence d'affichage divisée par une puissance de deux : ainsi, toutes les particules se re-synchroniseront à chaque affichage du système de particules.

### 6.6.2 Critères pour les pas de temps individuels

Durant une animation, le pas de temps d'une particule doit être suffisamment petit pour éviter toute instabilité lors de l'intégration, mais pas trop petit pour des raisons d'efficacité. Un critère important pour la précision d'intégration des équations de propagation est le critère de Courant (voir annexe C). Principalement utilisé pour les différences finies, ce critère stipule simplement qu'un front d'onde ne doit pas "sauter" une particule en un pas de temps ; autrement dit, une onde doit bien être propagée de proche en proche. Puisque l'on a vu que la vitesse maximale d'une onde de pression dans notre système était la vitesse du son  $c$  (voir section 6.2.3), il suffit alors d'écrire :

$$dt_i \leq \lambda_1 \frac{h_i}{c} \quad \text{où } \lambda_1 < 1 \quad (6.19)$$

De plus, l'animation de notre modèle déformable repose pour une grosse partie sur de bonnes approximations des vitesses et des densités échantillonnées aux particules. Puisque ces deux valeurs peuvent varier très rapidement par exemple en cas de choc, nous avons rajouté deux conditions supplémentaires, qui bornent respectivement le taux de variation de la vitesse et le taux de variation de la densité en un pas de temps :

$$dt_i \leq \lambda_2 \sqrt{\frac{h_i}{\|\mathbf{a}_i\|}} \quad dt_i \leq \frac{\lambda_3}{|(\text{div } \mathbf{v})_i|} \quad (6.20)$$

où  $\lambda_2$  and  $\lambda_3$  sont des constantes, et où  $\mathbf{a}_i$  est l'accélération courante de la particule  $i$ . Notez que ces trois constantes  $\lambda_1, \lambda_2$  and  $\lambda_3$  sont bien sans dimension, ce qui garantit la validité des critères.

### 6.6.3 Mise en œuvre

Pendant une simulation, le prochain pas de temps à utiliser pour une particule donnée est calculé à chacune de ses évaluations de forces, puisque l'accélération et la divergence de la vitesse peuvent changer. Le même calcul est bien sûr fait lors d'un raffinement ou d'une simplification spatiale.

Afin de pouvoir synchroniser l'animation tous les  $\Delta t$  dans l'optique d'un affichage des particules, le pas de temps individuel  $dt_j$  d'une particule  $j$  est pris à la plus grande valeur de  $\Delta t/2^q$  vérifiant les trois critères décrits plus haut, avec  $q$  un entier positif ou nul.

## 6.7 Algorithme complet d'animation

Comme certains calculs doivent être faits tous les plus petits pas de temps pour toutes les particules, nous utilisons une simple boucle pour l'implémentation de l'algorithme d'animation plutôt qu'une queue d'événements. En conséquence, le schéma algorithmique ressemble très fortement à la version des systèmes de particules classiques :

À chaque temps  $t$ :

– Pour chaque particule nécessitant une ré-évaluation des forces :

1. Trouver la liste des voisins d'après les positions courantes, et calculer la nouvelle divergence de la vitesse.

2. Tester les critères de raffinement/simplification, et procéder à la subdivision ou au regroupement si besoin
  3. Évaluer les forces de pression et de viscosité d'après les équations 6.15 et 6.10
  4. Après y avoir ajouté les forces de gravité et de collision, en déduire la nouvelle accélération de la particule :  $\mathbf{a}_i \leftarrow \mathbf{F}_i/m_i$
  5. Calculer la divergence de la vitesse selon l'équation 6.14, et en déduire par l'équation 6.13 la valeur de  $\dot{\rho}_i$
  6. Utiliser les nouvelles valeurs de l'accélération et de la divergence pour déterminer le laps de temps  $dt_i$  où la particule peut se passer de ré-évaluation, selon les critères 6.19 et 6.20
  7. Si besoin est, mettre à jour  $dt_{min}$ , le plus petit pas de temps en cours.
- Pour toutes les particules, mettre à jour la vitesse, la position, et la densité d'après les valeurs courantes  $\mathbf{a}_i$ ,  $\mathbf{v}_i$ , et  $\dot{\rho}_i$  et  $dt_{min}$  par les relations :  $\mathbf{v}_i \leftarrow \mathbf{v}_i + \mathbf{a}_i dt_{min}$ ,  $\mathbf{X}_i \leftarrow \mathbf{X}_i + \mathbf{v}_i dt_{min}$ , et  $\rho_i \leftarrow \rho_i + \dot{\rho}_i dt_{min}$
  - $t \leftarrow t + dt_{min}$

Comme la partie la plus coûteuse des calculs, à savoir la recherche des voisins et l'évaluation des forces, n'est faite que lorsque cela est strictement nécessaire, cet algorithme permet un gain substantiel en temps. Diverses petites optimisations ont été ajoutées pour encore gagner du temps, comme par exemple le codage du pas de temps par l'entier  $q$  tel que  $dt_j = \Delta t/2^q$ . Ainsi, le test permettant de savoir si la particule doit être ré-évaluée peut se faire simplement avec des décalages et des "ou" logiques.

## 6.8 Implémentation et résultats

L'algorithme d'animation ci-dessus demande l'utilisation de structures de données spécifiques pour accélérer les calculs, ainsi que quelques initialisations pour pouvoir démarrer la simulation.

### 6.8.1 Recherche des particules voisines

L'évaluation des forces et des divergences nécessite en premier lieu une recherche des particules voisines, ce qui se fait en  $O(n^2)$  par une implémentation naïve (si  $n$  est le nombre de particules). Mais cette fois, l'utilisation d'une grille régulière pour accélérer la recherche ne semble pas optimale puisque chaque particule a sa propre taille de filtrage.

À la place, nous avons développé une technique de type octree qui permet un temps algorithmique théorique de  $O(n \log(n))$ . Dans chaque sous-octree, on stocke deux listes, l'une indiquant quelles particules sont dans cette portion de l'espace, et l'autre indiquant les particules qui influent sur cette partie. Cet octree se ramifie au fur et à mesure des recherches effectuées, et permet encore une fois de localiser les calculs. Cependant, en pratique, cet algorithme est plus lent que l'implémentation naïve. En effet, il n'est pas difficile de constater qu'une telle structure de données nécessite des calculs qui ne peuvent s'amortir que pour un nombre élevé de particules. Or, dans ce cas, la taille et le nombre de listes de stockage des particules à chaque échelle de l'octree prennent un temps de gestion mémoire très important. Finalement, les temps obtenus par cette méthode lors de nos test ne rivalisent même pas avec une implémentation naïve.



Le meilleur compromis semble être encore une fois l'utilisation d'une grille régulière, permettant dans une grande majorité des cas d'accélérer la recherche des particules voisines, comme le prouve la figure 6.8.

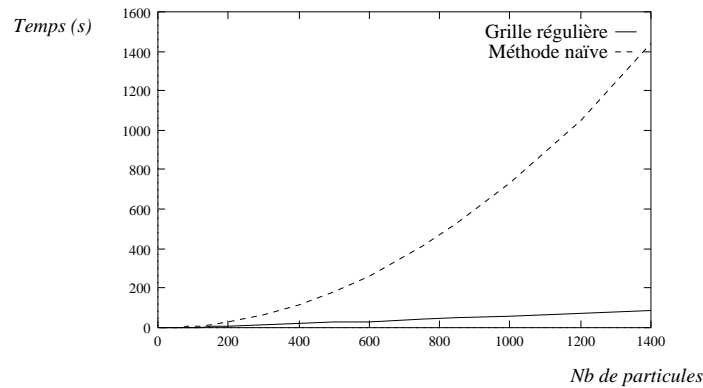


FIG. 6.8 - Comparaison des temps de recherches des voisins selon la méthode utilisée.

### 6.8.2 Discrétisation initiale

Avant de commencer toute simulation, une discrétisation initiale doit être mise en place. La forme initiale souhaitée est remplie de petites particules par empilement hexagonal. Chaque particule est supposée de même taille, masse, et densité initiale. Si l'on souhaite avoir une position d'équilibre, alors la pression interne doit compenser la gravité :

$$\nabla P = \rho \mathbf{g}.$$

Cette équation implique par exemple que pour une profondeur  $H$  de matière, où la gravité s'exerce selon l'axe des  $z$ , la densité initiale doit cette fois être fixée à :

$$\rho(z) = \rho_0 \exp(g(H - z)/k)$$

Quand les particules sont en place, on effectue alors une passe de simplification selon des critères de géométrie similaires à ceux décrits au paragraphe 6.5.4. La substance est ainsi prête à être simulée.

### 6.8.3 Résultats

Nous avons testé notre modèle sur divers exemples avec des jeux de paramètres variés résumés dans le tableau suivant :

Paramètre	Valeur(s)	Signification
$\rho_0$	$10 - 10^6$	Densité à l'équilibre
$k$	$10 - 10^6$	Raideur
$\eta$	0.1-10	Viscosité
$\alpha$	1.3	Rayon d'influence
$\Delta$	0.1-1	Seuil pour le raffinement
$\delta$	.01-0.1	Seuil pour la simplification
$\lambda_1$	.3	Nombre de Courant
$\lambda_2$	.5	Intégration précise de la vitesse
$\lambda_3$	.001	Intégration précise de la densité

Dans les exemples ci-dessous, les particules sont affichées comme des sphères de rayon  $r_i$  (voir équation 6.12). Ces sphères, qui représentent le volume de matière qu'échantillonnent les particules, servent aussi à détecter les collisions. Le raffinement se fait en remplaçant une particule par sept autres. Un détail important : toutes les simulations ci-dessous ont été réalisées sans frottement fluide.

Le premier exemple consiste à tester le nouveau modèle déformable avec un nombre fixe de particules. Pour cela, nous avons choisi une substance presque liquide, qui s'écoule dans une boîte transparente. La figure 6.9 indique les évolutions des pas de temps au cours du temps : on s'aperçoit que les moments de fortes perturbations, comme quand le liquide vient frapper la face opposée en (b), nécessitent plus d'étapes de calculs (sur-échantillonnage temporel) en  $1/25^e$  de seconde que lors du reste de l'animation. Ainsi, la stabilité est bien assurée malgré des chocs importants lors de la simulation.

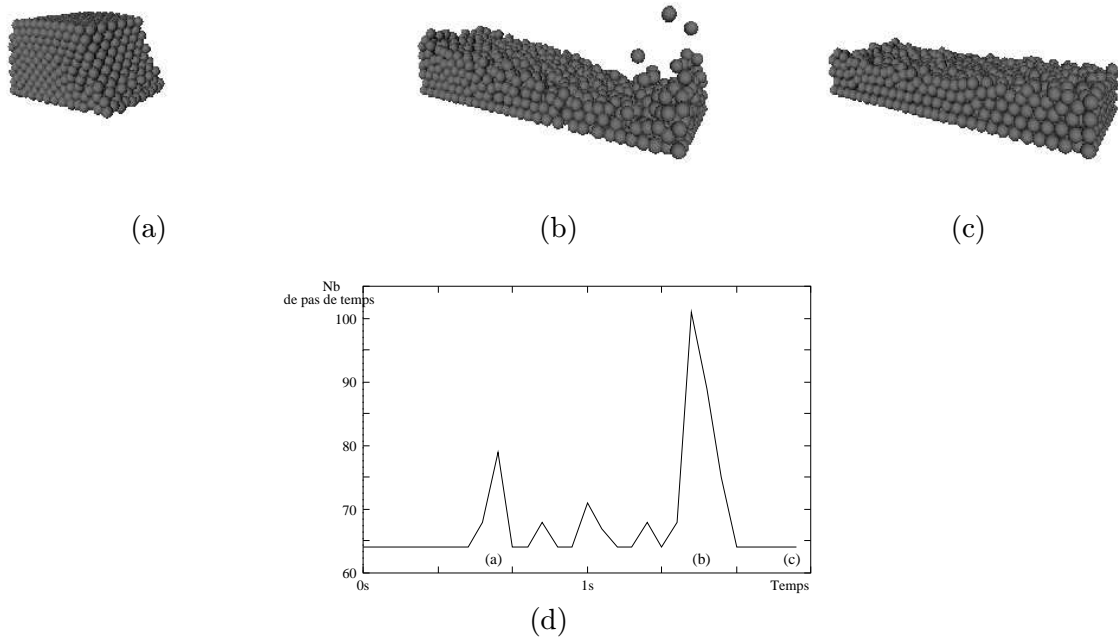


FIG. 6.9 - Animation de 1440 particules adaptées seulement en temps

Le deuxième test compare le comportement de l'objet simulé avec une discrétisation fixe ou adaptative. Un même objet mou, constitué de 105 particules dans sa version fixe et de 15 particules pour la simulation adaptative, tombe sous l'effet de la pesanteur. Un obstacle aigu le coupe en deux avant qu'il ne touche le sol. On peut alors vérifier que les résultats visuels ne sont guère différents grâce à notre équation de comportement global. Dans la simulation adaptative, le nombre de particules varie entre 15 et 99, donnant lieu à de multiples subdivisions et regroupements comme le prouvent la disparité des diamètres des particules en cours d'animation sur la figure 6.10. Le temps d'animation pour la discrétisation spatiale fixe avec le plus petit pas de temps utilisé dans la simulation adaptative est de 5 minutes 20. La version optimisée, utilisant des discrétisations spatiales et temporelles adaptatives, ne demande que 1 minute de calcul.

Enfin, des tests dans des environnements plus complexes ont été réalisés comme le montre la figure 6.11 par exemple. Ici, nous avons pris un matériau avec des caractéristiques proches de la

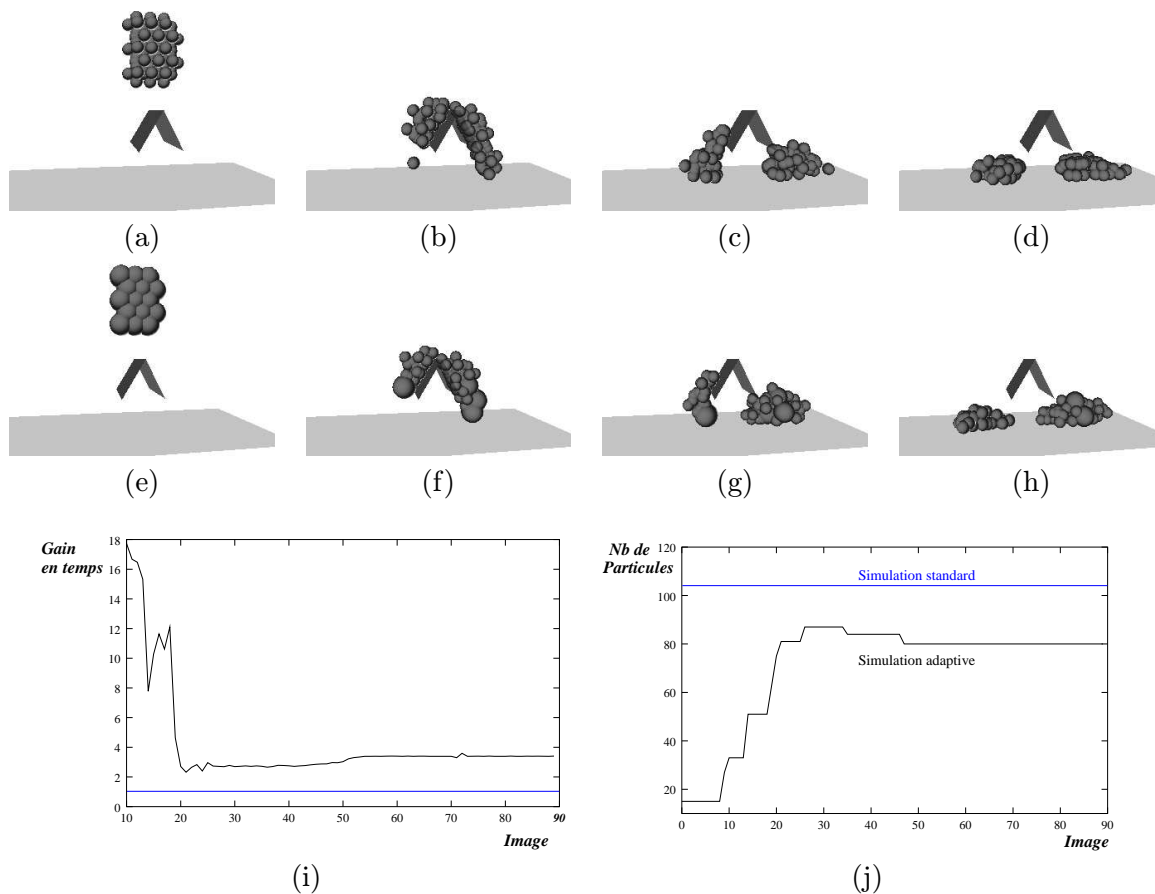


FIG. 6.10 - Comparaison entre discrétisation fixe (a, b, c, d) et adaptative (e, f, g, h): deux objets définis par la même équation d'état sont animés. Le gain en temps de calcul, ainsi que le nombre de particules utilisées pour chaque image dans la version adaptative sont indiqués en (i) et (j).

boue sur une colline simplifiée pour simuler une coulée de boue. C'est dans de grandes simulations de ce type que les particules adaptatives se révèlent fort adéquates, dans la mesure où des calculs non optimisés demanderaient des semaines de calculs. Pour l'exemple de la figure 6.12, le temps de calcul moyen entre deux images est de une seconde.

## 6.9 Conclusions et développements possibles

Nous avons présenté dans ce chapitre un nouvel algorithme qui permet une adaptation spatiale et temporelle automatique de la résolution d'un système de particules au cours d'une animation. Les avantages sont bien sûr une simplification du travail de l'animateur qui n'a plus à choisir une discrétisation fixe, mais surtout une optimisation des calculs due à des raffinements et des simplifications automatiques qui, à la fois, permettent de garantir précision et stabilité des calculs tout en offrant une description accrue des déformations.

Ces caractéristiques découlent de la formulation particulière de notre modèle déformable, qui définit le comportement d'une substance indépendamment de toute discrétisation. Une équation

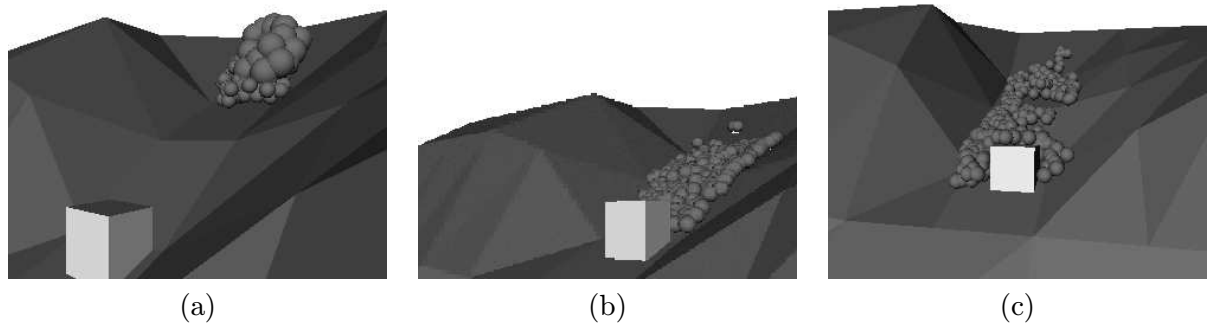


FIG. 6.11 - *Une coulée de boue sur un relief montagneux*

d'état lui confère en effet de façon macroscopique un comportement donné.

Il est à noter qu'une telle adaptation de la résolution a déjà été utilisée pour la simulation de gaz [SF93, SF95]. En effet, dans le cas particulier d'une équation d'advection/diffusion/dissipation, une optimisation peut être mise en œuvre. Si on représente la matière par des particules SPH, on peut simuler le processus d'advection en déplaçant les particules selon le vent. On peut aussi simuler le processus de diffusion en augmentant la taille de lissage de la particule au cours du temps. Enfin, la dissipation peut être simulée par une perte de poids exponentielle des particules au cours du temps. Ainsi, on dispose d'une discrétisation optimale sans calcul compliqué. Notre méthode s'inspire donc de cette approche, tout en se plaçant dans un cadre plus général.

Divers développements peuvent être envisagés à court terme. Par exemple, l'ajout de température dans le modèle déformable serait idéal pour créer des comportements plus complexes, comme celui de la lave. En effet, si le matériau a une raideur dépendant de la température, on pourra créer des animations où la matière durcit en refroidissant, à l'instar de [Ton91]. Des tests sur d'autres équations d'état simples peuvent être aussi envisagés.

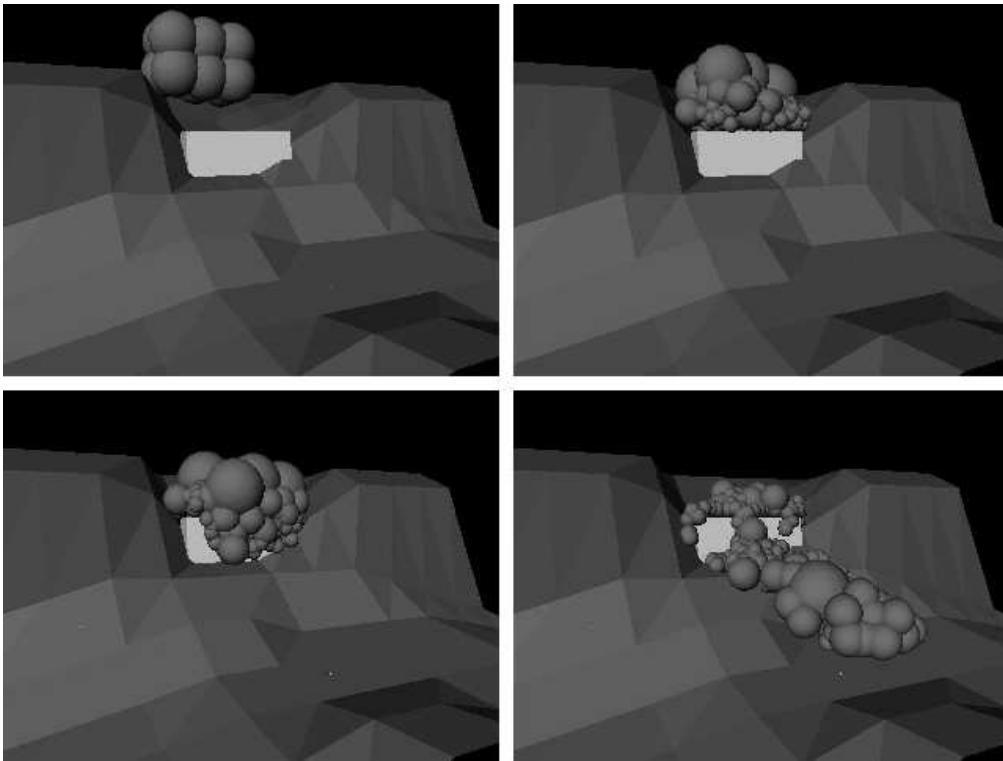


FIG. 6.12 - *Substance dévalant un relief montagneux ( $t=0.3s, 1.64s, 2.6s$  et  $4.1s$ ).*

## Chapitre 7

# Peau implicite active pour l'habillage

L'habillage externe d'un modèle déformable est la partie non immergée de l'iceberg : on ne voit que lui, et pourtant, il est supposé n'être qu'une fine couche, juste une interface entre l'intérieur et l'extérieur. On attend ainsi de cette ultime couche, appelée *peau*, deux propriétés antagonistes. L'essentiel en animation étant les mouvements créés, sensés être en grande partie gérés par le modèle interne, la peau ne doit pas être gourmande en temps de calcul. Mais la visualisation est pourtant fondamentale pour le rendu final et l'impression générale qui ressortira de l'animation : l'aspect géométrique et les déformations produites doivent être de qualité. Ce chapitre propose donc un modèle de peau active vérifiant ces contraintes restrictives.

### 7.1 Notion de peau

La peau est, comme nous l'avons déjà évoqué au chapitre 4, l'interface entre le modèle déformable et l'extérieur. Voyons en détail quel est son rôle exact.

#### 7.1.1 L'importance du paraître

Une peau sert en premier lieu à visualiser le modèle animé. Cette tâche est évidemment fondamentale puisque la représentation graphique, indépendamment du mouvement, est en grande partie responsable de l'impression globale ressentie lors d'une animation. Dans certains cas, l'habillage peut même améliorer un modèle interne peu convaincant, de même que les textures peuvent gommer certaines insuffisances de la géométrie en rendu d'images.

Les particules et les réseaux masses/ressorts illustrent bien le besoin d'habillage : ces points discrets ne peuvent suffire pour visualiser l'animation, une représentation continue est nécessaire. Qui plus est, cette représentation doit être cohérente avec le modèle physique, ce qui, comme on l'a vu au paragraphe 2.3, n'est pas souvent le cas. Des artefacts visuels apparaissent alors indubitablement. Le rendu visuel de la peau est donc un aspect essentiel.

#### 7.1.2 L'importance du rôle physique

La peau d'un modèle physique est aussi sensée détecter les collisions avec le reste de la scène puisqu'elle est l'interface entre intérieur et extérieur. En pratique, ceci est pourtant rarement respecté comme nous avons pu le constater en passant en revue les modèles existants dans le chapitre 2.

Plus généralement, une peau dotée de propriétés physiques permet d'*enrichir* grandement un modèle. La tension de surface est par exemple une caractéristique notable dans le cas de substances hautement déformables : on peut citer les effet classiques qui lui sont dus tels que la goutte d'eau sur une table, ou le ménisque dans les éprouvettes. Ce type de phénomène apporte un réalisme conséquent à une animation en synthèse de mouvement. La peau ne doit donc pas se contenter d'être *passive* uniquement, son rôle physique est aussi d'importance.

### 7.1.3 L'importance de l'efficacité

L'animation de la peau, aussi capitale soit-elle pour l'esthétisme et le réalisme, doit pourtant pouvoir se faire avec des temps de calcul très faibles. L'animation proprement dite est en théorie produite par les couches internes gérant les fortes déformations : la gestion de la peau doit donc être optimisée pour ne pas devenir un goulot d'étranglement lors de l'animation du modèle complet. En d'autres termes, la peau est sensée être la couche finale, et non principale.

Ce n'est pas forcément le cas de tous les modèles déformables, puisque certains auteurs prennent le point de vue inverse qui veut que l'essentiel des calculs soit affecté à la peau, seul élément visible du modèle [Tur95]. Mais encore une fois, la simulation de matériaux hautement déformables impose des contraintes inhabituelles, qui, sans ternir son caractère fondamental, placent le modèle surfacique à une moindre priorité en terme de temps de calcul.

## 7.2 Problématique

Au vu des contraintes imposées au modèle de peau, il semble que la couche implicite de notre premier modèle présenté au chapitre 5 n'offre qu'une solution partielle. Cette utilisation de surfaces implicites pour l'habillage apporte certes beaucoup d'avantages du point de vue rapidité de la gestion de la topologie et de la détection des collisions, ou encore pour la qualité de la visualisation, mais un petit nombre de points gênants demeurent.

### 7.2.1 Inadéquation d'un habillage purement géométrique

Le fait de disposer d'un modèle interne *adaptatif* rend obsolète le modèle développé précédemment. En effet, un habillage *purement géométrique* ne convient plus : il ne dépend que de la position des squelettes du modèle interne, sans aucune "inertie". Il est "passif", même si des déformations lui sont ajoutées localement lors de collisions. Si ceci n'est pas forcément un handicap pour la plupart des modèles internes, les modèles adaptatifs comme notre système de particules décrit au chapitre 6 ne peuvent être habillés que par une véritable peau physique. Sinon, chaque changement de granularité du modèle interne sera immédiatement répercuté par la surface, créant un brutal et désagréable changement de forme (voir figure (7.1)). Ce phénomène est parfois nommé *popping effect*.

### 7.2.2 Nécessité d'une peau active

Si l'enrobage est au contraire une interface *active*, cette peau viendra épouser *lentement* la nouvelle forme du matériau sous-jacent quelles que soient les subdivisions ou les simplifications qui ont lieu. Il convient donc de créer une peau physique, attirée par le modèle sous-jacent. De plus, le réalisme final du modèle sera grandement amélioré en conférant des propriétés physiques à la peau, comme une *tension de surface* par exemple.

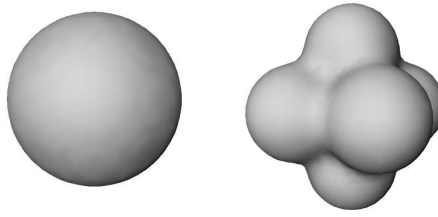


FIG. 7.1 - *Exemple de popping : une particule se raffine en six particules.*

Cela n'est pas sans rappeler les "snakes" utilisés en vision [KWT88]. À partir d'informations extraites d'une image, ces modèles de surfaces déformables extraient des contours minimisant des critères d'énergie bien choisis. On peut tout à fait transposer cette approche à notre cas 3D, en appliquant des forces pour modéliser la peau selon la chair sous-jacente, et en minimisant l'énergie de déformation pour simuler la tension de surface ; mais comme notre champ d'action comprend des objets susceptibles de se découper en plusieurs morceaux, les snakes classiques, de topologie fixe, ne sont plus du tout adaptés.

### 7.2.3 Autres caractéristiques souhaitées

D'autres raisons nous poussent à trouver un nouvel habillage. L'aspect granuleux du matériau, délicat à supprimer dans le modèle du chapitre 6, peut facilement s'effacer si la peau dispose d'une tension de surface réglable. De plus, diminuer les temps de calcul d'une animation passe par la suppression de l'étape la plus lente : le rendu par lancer de rayons. Si la peau pouvait fournir, à peu de frais, une représentation polygonale fine, on éliminerait ainsi les heures de calculs de rendu qui étaient, jusqu'à présent, un passage obligé.

### 7.2.4 Formulation implicite

On a vu dans les chapitres précédents tous les avantages que peut procurer une formulation implicite pour l'habillage d'un modèle très déformable. Par exemple, les surfaces implicites gèrent automatiquement tout changement de topologie ; ne serait-ce que pour cette raison, une peau définie implicitement s'avère adaptée à nos besoins. Dans ce chapitre, nous nous proposons donc de définir un modèle de peau alliant surfaces implicites et comportement dynamique avec inertie. De cette manière, on disposera d'une peau venant épouser le modèle interne, simulant une tension de surface, offrant une polygonalisation très rapide tout en gérant parfaitement les changements de topologie, et enfin permettant encore d'accélérer le processus de détection des collisions.

## 7.3 Approches antérieures de surfaces déformables

La peau que l'on désire modéliser peut être tout simplement vue comme une surface déformable. Pour la simuler, deux familles d'approches distinctes sont envisageables.



### 7.3.1 Approches Lagrangiennes

C'est peut-être la solution la plus intuitive : pour simuler une peau, on pourrait la discrétiser en éléments de masse, reliés entre eux par des liaisons appropriées de type ressort ou force de Lennard-Jones, comme vu au paragraphe 2.1.2. Il suffit alors d'intégrer la dynamique de l'ensemble pour obtenir une surface déformable du genre tissu [HBG92, ST92, VCT95, Pro95, WTT95]. Cette méthode est aussi utilisée dans un domaine connexe de la synthèse d'images, à savoir l'analyse d'images, dans le cadre de segmentation de données volumiques [LM96]. Une manière équivalente de procéder est de formuler les équations du mouvement en termes de surfaces déformables continues comme l'a proposé Demetri Terzopoulos [TPBF87]. Par discrétisation de la surface et utilisation des différences finies, on simule de façon analogue une peau élastique venant épouser une forme donnée [Tur95].

Pourtant, cette première façon de faire pose un gros problème de maintien des liaisons entre noeuds de discrétisation pour l'application qui nous intéresse. Notre peau devant servir à l'animation d'objets très déformables, les nombreux changements de topologie qui peuvent apparaître rendent délicat la conservation d'une polygonalisation cohérente au cours du temps [LB94]. La méthode envisagée dans [WH94] qui consiste à calculer à chaque affichage le diagramme de Voronoï des particules surfaciques pour en déduire une polygonalisation n'est pas envisageable, car bien trop gourmande en temps de calcul.

### 7.3.2 Approches Euleriennes

Au lieu de discrétiser la matière en éléments, puis de suivre leurs mouvements, les approches Euleriennes partitionnent l'espace en volumes fixes, puis regardent ce qui entre et sort de chacun de ces volumes. Les deux approches sont bien sûr équivalentes, mais diffèrent dans leur façon d'écrire les équations et leur mises en œuvre (voir, pour plus de précisions, l'annexe A "Euler vs. Lagrange".)

Des approches Euleriennes ont été utilisées en animation, particulièrement pour la simulation d'eau et de gaz [KM90, FM96], mais jamais à notre connaissance pour une simulation de surface déformable. Un modèle d'habillage particulier utilise aussi ce type d'approche, mais avec une formulation originale : l'écran d'épingles d'Arash Habibi, vu dans le paragraphe 2.2.1. Cette technique permet bien plus que de représenter une simple surface : elle offre la possibilité d'un véritable rendu volumique, même si sa mise en œuvre actuelle n'est que 2D. Réduire son rôle uniquement à la définition d'une surface gaspillerait beaucoup de temps de calcul.

D'une manière générale, le formalisme Eulerien semble bien adapté à un traitement rapide d'une part de l'évolution d'une surface, et d'autre part de sa représentation, grâce à des *voisinages fixes* connus. La phase la plus coûteuse des approches Lagrangiennes, à savoir le calcul à chaque pas de temps des nouvelles relations de voisinages entre points d'échantillonnage, est ainsi évitée.

### 7.3.3 Discussion

Dans l'optique d'un rendu surfacique performant et rapide, il semble que les approches Euleriennes soient bien mieux adaptées à notre application. Leur avantage majeur est qu'elles conservent intrinsèquement la notion de *voisinage* entre points d'échantillonnage de par leur découpage fixe de l'espace, contrairement aux approches Lagrangiennes qui obligent une recherche de voisins coûteuse en temps de calcul. Cette particularité promet *a priori* de permettre une polygonalisation rapide même lors de changements de topologie. Malheureusement, aucun modèle surfacique basé sur une telle approche n'a été proposé jusqu'à présent. Le chapitre suivant

présente une solution possible, alliant une approche Eulerienne et les caractéristiques attendues pour notre modèle de peau.

## 7.4 Peau Implicite Mouvante

Dans cette section, nous nous attachons à décrire comment modéliser et animer de manière efficace une peau implicite ayant les qualités requises. Les détails de mise en œuvre seront vus dans la section suivante.

### 7.4.1 Définition d’une surface implicite mouvante

La définition formelle d’une surface implicite  $S$  qui bouge, se déforme, se coupe au cours du temps peut s’écrire fort simplement :

$$S(t) = \left\{ \mathbf{X} \in \mathbb{R}^3 / f(\mathbf{X}, t) = 0 \right\}.$$

où  $f$  est une fonction dite “potentiel” de  $\mathbb{R}^3 \times \mathbb{R}$  dans  $\mathbb{R}$ . Nous avons fait l’hypothèse d’une isovaleur nulle pour simplifier les explications qui suivent. Dans le reste de ce chapitre, nous prendrons la convention de définir l’intérieur d’un objet par le lieu où  $f$  prend des valeurs positives, et de diriger les normales unitaires  $\mathbf{n}$  vers l’extérieur. On a alors la relation suivante :  $\mathbf{n} = -\nabla f / \|\nabla f\|$ .

### 7.4.2 Inadéquation d’une représentation par squelettes

Notre ancien modèle d’habillage utilisait des surfaces implicites engendrées par squelettes. Le mouvement de ces squelettes entraînait la déformation de la surface de façon induite. Dans le cadre qui nous intéresse à présent, cette technique n’est que peu appropriée. En effet, son premier défaut est que la complexité de la surface résultante est directement liée au nombre de squelettes simples la générant. Si l’on souhaite *dissimuler la granularité du modèle physique*, il faudra un nombre élevé de primitives, amenant rapidement une saturation des calculs pour des formes complexes.

Son deuxième défaut est plus lié à notre modélisation de la couche interne. Les particules que l’on utilise étant adaptatives, et donc susceptibles de subdivisions et de fusions, les utiliser comme squelettes amène au problème déjà évoqué du *popping*, dû à un habillage purement géométrique. Toutefois, grâce au formalisme SPH, on dispose d’un véritable champ de densité qui paraît être un bon point de départ pour construire une délimitation surfacique : la surface d’un objet pourrait être définie comme le lieu où la densité de masse devient inférieure à un certain seuil. Il convient donc de trouver une formulation adaptée qui puisse tirer profit de cette information sans avoir les inconvénients de la modélisation par squelettes.

### 7.4.3 Intérêt d’un champ discret

Dans le paragraphe 7.3.2, on a vu que l’intérêt principal des approches Euleriennes était de faciliter la polygonalisation grâce à des relations de voisinage fixes. On peut dans cet esprit songer à utiliser une *fonction implicite discrète* : le potentiel n’a alors aucune formulation analytique, mais est simplement stocké sur une grille régulière. Une version continue peut facilement être interpolée par une fonction tri-linéaire à l’intérieur de chaque cube de la grille par exemple.

Cette formulation apporte bon nombre d'avantages. En premier lieu, le coût de l'évaluation du potentiel d'un point de l'espace se fait en temps constant, *indépendamment de la complexité de la surface manipulée*. Cette qualité permet de garantir un temps de traitement borné quelque soit la surface simulée.

En second lieu, la discrétisation de la surface engendrée par un potentiel discret est largement facilitée par rapport au cas général des surfaces implicites quelconques. Puisque l'on dispose déjà d'une grille régulière où le potentiel est stocké, les cubes par lesquels passe la surface sont facilement trouvés en recherchant les endroits où le potentiel change de signe. Cela correspond en tout point à l'algorithme des "Marching Cubes" [WMW86, LC87] sauf que les parties les plus longues, à savoir calcul du potentiel et traque de la surface, sont ici inutiles. Une fois ces cubes trouvés, on obtient une discrétisation en interpolant entre les sommets du cube afin de trouver des points de potentiel nul. Notre choix d'interpolation tri-linéaire permet d'effectuer la recherche de points à la surface de façon extrêmement simple : la position du point de potentiel nul entre deux sommets de signes opposés se fait par une simple règle de trois, *sans aucune dichotomie*. De plus, une polygonalisation à partir des points d'échantillonnage ainsi calculés est triviale à construire du fait que la grille permet de connaître les relations de voisinage : c'est exactement la méthode de polygonalisation par partitionnement spatial, vue au paragraphe 3.3.1.

Enfin, un dernier avantage apporté par cette formulation discrète est la possibilité d'un *contrôle localisé* de la surface. On peut en effet agir sur chaque valeur du potentiel discret pour affecter très localement la surface avoisinante, sans pour autant augmenter la complexité de la formulation [FV97].

#### 7.4.4 Comportement différentiel du champ pour une surface animée

Puisque le potentiel définit une surface mouvante, il suit une équation différentielle. En écrivant que le chemin d'un point  $\mathbf{X}(t)$  restant sur la surface au cours du temps a la propriété de vérifier  $f(\mathbf{X}(t), t) = 0$  pour tout  $t$ , on obtient :

$$\forall \mathbf{X}(t) \in S(t), \quad \frac{df}{dt}(\mathbf{X}(t), t) = \frac{\partial f}{\partial t}(\mathbf{X}(t), t) + \nabla f(\mathbf{X}(t), t) \cdot \frac{d\mathbf{X}(t)}{dt} = 0 \quad (7.1)$$

Une façon d'animer la surface est alors de définir un champ de vitesses  $\frac{\partial \mathbf{X}}{\partial t}$  (puisque c'est la seule inconnue dans l'équation différentielle) et d'intégrer l'équation (7.1) au cours du temps. Ce type d'approche a d'ailleurs déjà été utilisé par Witkin et Heckbert [WH94] pour une application de type échantillonnage interactif, ainsi qu'en animation de textures [SS96] ou en vision [MSV95, Whi95].

Contrôler l'évolution de la surface revient donc à affecter à chaque point de la surface une *vitesse instantanée* de déplacement. Par intégration de l'équation différentielle (7.1), *la valeur du potentiel  $f$  sera modifié pour prendre en compte le déplacement souhaité*. Ce contrôle *indirect* du mouvement de la surface, par le biais du potentiel la définissant, peut paraître une méthode quelque peu tortueuse. On verra pourtant par la suite que cette approche offre bien des avantages.

#### 7.4.5 Définition d'une stratégie pour la peau

Pour donner finalement vie à la peau, il faut et il suffit de définir un *champ de vitesses* à chaque instant. Il s'agit donc de trouver une stratégie qui permette de définir ce champ en tout point et à chaque instant, de façon à ce que la peau bouge comme on le souhaite.

### Proposition d’une stratégie

Notre peau est sensée recouvrir le modèle physique intérieur. Dans le cas des particules adaptatives, le modèle physique fournit aussi un champ de densité de masse, permettant de localiser la matière dans l’espace. Une bonne définition du contour de l’objet serait alors de choisir une isodensité comme frontière entre intérieur et extérieur. Malheureusement, cette frontière n’est que purement géométrique, et souffre de discontinuités temporelles lors de subdivisions ou de fusions de particules.

Une idée immédiate est d’alors utiliser cette iso-densité comme *cible*, c’est-à-dire que la peau sera constamment *attirée vers cette frontière*. On contourne ainsi le problème de l’habillage purement géométrique en introduisant une simulation de surface déformable tendant à englober un volume donné. De plus, la stratégie permettant ce comportement peut s’écrire très simplement :

$$\frac{d\mathbf{X}}{dt}(t)_{cible} = \alpha (G_{cible}(\mathbf{X}) - iso) \mathbf{n}(\mathbf{X}) \quad (7.2)$$

où  $\alpha$  est un scalaire,  $G_{cible}$  la densité de masse induite par le modèle interne, et  $iso$  l’iso-densité choisie. La signification de cette équation est assez intuitive : si la surface actuelle est à l’intérieur de la matière, elle devra gonfler selon sa normale pour rejoindre l’iso-densité cible. Inversement, si la surface est localement à l’extérieur de la cible, elle devra se dégonfler pour aller épouser le bord. Cette façon de gonfler et dégonfler fait bien sûr penser à une bulle ou à un ballon (figure 7.2). Le concept est d’ailleurs très proche des “balloon snakes” de Laurent Cohen en analyse d’images [Coh91].

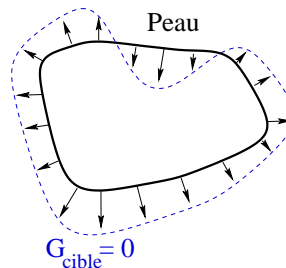


FIG. 7.2 - *Stratégie de la bulle : gonflements et dégonflements locaux*

Cette stratégie permet le parfait respect du modèle stratifié défini au chapitre 4 : la peau “recouvre” effectivement le modèle de chair. D’ailleurs, plus généralement, la stratégie ainsi définie reste valable pour un potentiel  $G_{cible}$  quelconque. On peut par exemple penser à prendre des potentiels très simples, qui seraient par exemple des unions de sphères engendrées par les particules. Par le biais de la tension de surface, la peau reconstruite sera bien lisse, malgré un potentiel cible seulement  $\mathcal{C}^0$ .

De même, rien n’impose dans cette stratégie que le potentiel  $G_{cible}$  soit fixe. Il peut changer au cours du temps, ce qui est le cas de la densité de masse au cours d’une animation : la peau sera alors souvent en train de poursuivre la cible. Il serait d’ailleurs plus correct de noter le potentiel :  $G_{cible}(\mathbf{X}, t)$ . La notation simple, sans indication de temps, sera conservée uniquement pour ne pas compliquer les formules.

### Prise en compte d'une tension de surface

La stratégie peut aussi conférer à la peau une simulation de tension de surface. Par analogie à une peau élastique sous tension, il suffit d'ajouter un terme de pénalisation de la courbure pour simuler finalement une tension de surface. La stratégie devient alors:

$$\begin{aligned} \frac{d\mathbf{X}}{dt} &= \frac{d\mathbf{X}}{dt}(t)_{cible} + \frac{d\mathbf{X}}{dt}(t)_{tension} \\ &= \alpha (G_{cible}(\mathbf{X}) - iso)\mathbf{n}(\mathbf{X}) + \beta \kappa(\mathbf{X}) \mathbf{n}(\mathbf{X}) \\ &= [\alpha (G_{cible}(\mathbf{X}) - iso) + \beta \kappa(\mathbf{X})] \mathbf{n}(\mathbf{X}) \end{aligned} \quad (7.3)$$

où  $\kappa$  est la courbure moyenne, et  $\beta$  un coefficient permettant de régler l'importance de la tension de surface. Par cette dernière équation, on impose à la peau à la fois de poursuivre l'iso-densité et de minimiser sa courbure en tout point.

### 7.4.6 Considérations simplificatrices

Pour réduire le nombre de calculs, il est important de se débarrasser de tout le superflu pour ne garder que l'essentiel. La peau n'ayant qu'un rôle surfacique, il est logique de limiter le calcul du potentiel discret  $f$  à une zone proche de la surface.

#### Voisinage Tubulaire

Comme Velho et Gomes l'ont fait remarquer [VG96], les caractéristiques fondamentales d'un potentiel sont ses variations autour de la surface implicite qu'il définit. En dehors de ce voisinage tubulaire de la surface, les valeurs du potentiel n'ont aucune influence sur la surface elle-même.

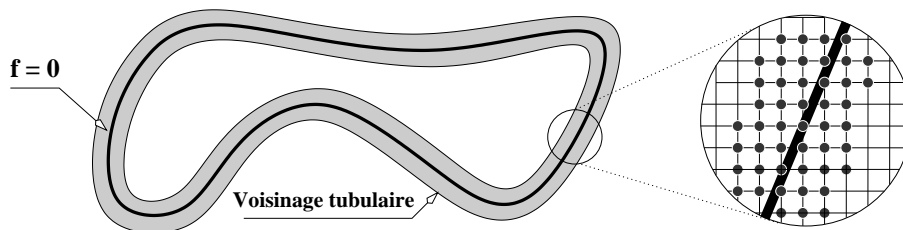


FIG. 7.3 - Voisinage tubulaire d'une surface implicite définie par  $f = 0$ .

Dans notre cas, où le potentiel est discret, ce voisinage se traduit par un ensemble de points de la grille tout autour de la surface (figure 7.3). L'information contenue dans le voisinage tubulaire permet ainsi de déduire le gradient du potentiel sur la surface par différences finies. On s'aperçoit qu'il n'est donc pas nécessaire d'intégrer les variations de  $f$  sur toute la grille : il suffit de *propager un front de quelques voxels d'épaisseur* pour bénéficier de toutes les informations nécessaires à l'évolution de la surface.

#### Nœuds autour de la surface

Restreindre les calculs à un voisinage proche de la surface va permettre à tout instant d'optimiser à la fois les opérations d'intégration du champ et celles nécessaires à sa polygonalisation. En limitant les calculs au voisinage strictement nécessaire, on gagne non seulement en efficacité, mais de plus, *le coût de l'animation de la peau devient proportionnel à la surface couverte*, et

non au nombre total de nœuds de discrétisation de  $f$  à l'intérieur du volume qui nous intéresse, qui est  $O(n^3)$  pour une grille de  $n \times n \times n$ . En pratique, cela fait gagner un ordre de grandeur et ramène la complexité moyenne à  $O(n^2)$ .

### Seuillages intérieur et extérieur

Les régions hors du voisinage tubulaire de la surface ne sont d'aucun intérêt. Il est pourtant bon de savoir si elles sont internes, ou externes à l'objet : cela permettra une détection des collisions efficace. On seuille donc le potentiel comme le propose Ross Whitaker [Whi95] : tous les nœuds dont la valeur de  $f$  dépasse 1 sont seuillés à 1, et donc considérés comme intérieurs. Inversement, tous les nœuds dont la valeur passe en dessous de  $-1$  sont seuillés à  $-1$  et considérés comme extérieurs. Le choix de la valeur seuil est bien sûr arbitraire : toute autre valeur changerait l'échelle du potentiel discret, mais pas la surface.

L'ensemble de ces méthodes et de ces simplifications permet alors de définir, au moins formellement, un modèle discret de peau implicite active à potentiel discret comme nous le souhaitons. Reste à mettre le tout en œuvre.

## 7.5 Mise en œuvre

Les éléments exposés au paragraphe précédent peuvent difficilement s'implémenter directement : arriver à un modèle robuste et performant demande un certain nombre d'améliorations.

### 7.5.1 Approche générale de la mise en œuvre

L'équation différentielle (7.1), de type Halmilton-Jacobi, a été étudiée en détail par James Sethian [Set96]. Il propose une méthode mathématique un peu coûteuse, passant par une évaluation des dérivées autres que les classiques différences finies, qui permet d'intégrer cette équation de façon précise et cohérente.

Cette théorie est maintenant très élaborée. Nous avons cependant préféré développer une méthode plus rapide et très stable, même si mathématiquement moins précise, dont nous allons discuter les points délicats de la mise en œuvre. Cette approche est bien adaptée à la Synthèse d'Images puisqu'elle permet une simulation de qualité visuelle suffisante pour une complexité de traitement faible.

### 7.5.2 Structure de données

#### Codage du potentiel

Le potentiel étant un ensemble de valeurs discrètes sur une grille 3D, nous utilisons pour le stocker un tableau  $n \times n \times n$  de nombres réels, où  $n$  sera fixé en fonction de la finesse de polygonalisation souhaitée [FV97].

De plus, le seuillage entre  $-1$  et  $1$ , décrit au paragraphe 7.4.6, peut conduire à une petite optimisation : les valeurs de  $f$  peuvent être codées en virgule fixe, sur par exemple 2 octets, assurant une précision de  $1/32768$  pour un gain de place mémoire d'un facteur 4 par rapport à un codage en double precision.

### Liste chaînée des nœuds autour de la surface

De façon à pouvoir accéder rapidement aux seuls nœuds intéressants, à savoir ceux qui se trouvent dans un voisinage tubulaire de la surface, nous utilisons une simple liste chaînée comprenant les coordonnées de ces nœuds dans le tableau du potentiel. En effet, cette liste peut être tenue à jour dynamiquement de façon très rapide par suppressions et additions de nœuds au cours du mouvement de la surface. Une structure de données plus complexe n'est pas nécessaire : l'ordre de stockage des nœuds n'a aucune importance, puisque le traitement de chaque nœud est indépendant. Ainsi, nous conservons les coordonnées des nœuds *utiles* au cours du mouvement de la surface, et l'on verra par la suite que cette structure suffit.

### 7.5.3 Extensions du champ de vitesses

La stratégie de vitesse instantanée exposée au paragraphe 7.4.5 n'était définie que pour des points sur la surface de la peau. Or l'intégration se fait uniquement sur les nœuds de la grille proches de la surface : il s'agit donc d'étendre la notion de champ de vitesses au voisinage de la surface.

#### Problème d'oscillation

Une intégration naïve de l'équation différentielle sur tous les points de la grille voisins de la surface amène d'inévitables oscillations. Pour s'en convaincre, il suffit de comprendre que la vitesse d'un nœud ne peut être nulle que si ce nœud est juste sur la surface cible fixée : alors,  $G_{cible}(\mathbf{X}) = iso$  et  $\frac{d\mathbf{X}}{dt}$  est nul. Les nœuds étant sur une grille régulière, ce cas de figure est évidemment très rare : cela signifie que le système n'arrivera jamais à un état de repos, oscillant toujours autour de la position cible.

En effet, l'équation différentielle 7.1 n'a de sens que pour des points *sur la surface*. Pour remédier à cet inconvénient, il est judicieux de formuler la stratégie aux sommets de la grille *en fonction du point le plus proche de la surface actuelle*. Ainsi, si la peau passe exactement par la surface cible, les sommets voisins seront bien affectés d'une vitesse nulle. Cela équivaut à *faire bouger les isopotentiels voisines à la même vitesse que la surface*, comme le représente la figure 7.4 : c'est d'ailleurs la méthode proposée, mais non utilisée en pratique, par Malladi et Sethian [MSV95], car cette solution amène un nouveau problème : comment trouver rapidement le point le plus proche ?

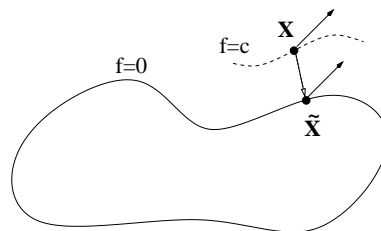


FIG. 7.4 - Utilisation du point de la surface le plus proche (d'après [MSV95])

## Évaluations sur le maillage polygonal

Nous proposons, plutôt que de réellement déterminer la projection  $\tilde{\mathbf{X}}$  sur la surface actuelle d'un point  $\mathbf{X}$  de la grille, d'approximer  $G_{cible}(\tilde{\mathbf{X}})$  grâce aux informations locales dont on dispose déjà. Si, lors de la dernière polygonalisation de la peau, des voxels autour de  $\mathbf{X}$  étaient à cheval sur la surface, un ensemble de  $p$  points  $\{\mathbf{P}_i\}$  sur la surface actuelle ont été déterminés par interpolation linéaire<sup>1</sup>. Alors, nous remplaçons tout simplement l'évaluation de  $G_{cible}(\tilde{\mathbf{X}})$  par :

$$\frac{1}{p} \sum_{i=1..p} G_{cible}(\mathbf{P}_i).$$

Ceci peut s'implémenter fort simplement en examinant si, dans les six arêtes partant de  $\mathbf{X}$ , il y a des points trouvés comme appartenant à la surface lors la polygonalisation précédente. Cela nécessitera donc jusqu'à six évaluations du potentiel cible par sommets de la grille, mais un système d'estampillage évitera de refaire des évaluations lors du traitement des nœuds voisins. Cette méthode nécessite du coup autant d'évaluations de  $G_{cible}$  que de points d'échantillonnage, évitant ainsi une surcharge de calculs. Dernier cas : si le nœud  $\mathbf{X}$  ne possède aucun point sur la surface dans son voisinage, on se contente de la classique évaluation de  $G_{cible}(\mathbf{X})$  sans que cela ne prête à conséquence :  $\mathbf{X}$  est "assez" loin de la surface.

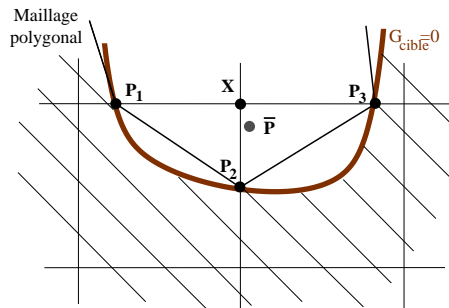


FIG. 7.5 - *Evaluation de la stratégie en utilisant les points de la surface*

Par cette méthode, nous sommes alors assurés de la bonne intégration du mouvement, et surtout, d'un état d'équilibre stable une fois que les polygones passent bien par la surface cible. Cela aurait été moins valable si l'on avait plutôt choisi d'évaluer  $G_{cible}(\bar{\mathbf{P}})$ , avec  $\bar{\mathbf{P}} = \frac{1}{p} \sum \mathbf{P}_i$ , car on aurait alors évalué un point hors du maillage polygonal représentant la peau, comme l'indique la figure 7.5 : la vitesse en  $\mathbf{X}$  est non nulle même si les points de discrétisation trouvés sont parfaitement sur la surface à reconstruire, ce qui aurait donné lieu à des oscillations.

On remarque qu'ainsi, à chaque pas de temps, on se sert de *tous* les points de discrétisation courants pour définir comment le potentiel (et par effet de bord, la surface) doit exactement évoluer.

## Convergence et stabilité

Ainsi traité, le modèle défini ci-dessus permet une convergence aussi bien pour une surface cible fixe que mouvante. Dans le cas de vitesse élevée de la cible, il faudra prendre garde que la peau ne sorte jamais totalement de l'objet cible : sinon, elle caverait (s'écraserait sur elle-même) et disparaîtrait. Il faut donc régler le paramètre  $\alpha$  à des valeurs non infimes, pour être assuré que

1. Ce cas est relativement courant, puisque l'on limite tous les calculs au voisinage tubulaire de la surface.



la peau puisse suivre tous les mouvements de la cible. Quant à la stabilité, on remarquera que l'évaluation de  $G_{cible}$  sur des sommets du maillage polygonal évite de recourir à une viscosité artificielle<sup>2</sup> lors de l'intégration, puisque sans ajout d'aucun terme dans l'équation différentielle, on obtient une intégration convergente et stable sur une grille régulière (dans le cas d'une cible fixe bien sûr).

#### 7.5.4 Ajout de la tension de surface

Puisque simuler une tension de surface revient à minimiser la courbure de la peau en tout point, il faut pouvoir exprimer la courbure moyenne locale en tout point, à savoir la moyenne des deux courbures principales. La surface est définie implicitement par le potentiel  $f$ , il est possible alors d'écrire la courbure moyenne comme suit :

$$\begin{aligned} \kappa(\mathbf{X}) &= \operatorname{div}(\mathbf{n}(\mathbf{X})) \\ &= \operatorname{div}\left(-\frac{\nabla f(\mathbf{X})}{\|\nabla f(\mathbf{X})\|}\right) \\ &= -\frac{f_x^2(f_{yy}+f_{zz})+f_y^2(f_{xx}+f_{yy})+f_z^2(f_{xx}+f_{yy})-2f_x f_y f_{xy}-2f_y f_z f_{yz}-2f_x f_z f_{xz}}{(f_x^2+f_y^2+f_z^2)^{3/2}} \end{aligned} \quad (7.4)$$

en fonction des différentes dérivées partielles ( $f_x, f_{xx}, \dots$ ) approximées en  $\mathbf{X}$  par différences finies. Tout autre choix de courbure peut aussi être valable, voire préférable. Aussi, l'annexe D explique en détail comment exprimer une combinaison quelconque des courbures principales en tout point d'une surface implicite.

Les courbures étant connues aux sommets de la grille, on peut tout simplement calculer la composante  $\beta\kappa(\mathbf{X})$  en chacun de ces sommets. En effet, il s'avère en pratique qu'une telle évaluation suffit à une bonne animation de la peau, sans recourir à des évaluations plus fines comme c'est obligatoirement le cas pour la composante  $\alpha(G_{cible} - iso)$ .

#### 7.5.5 Mise à jour du voisinage tubulaire

Un point important à aborder est la gestion dynamique des nœuds du voisinage tubulaire de la surface. Il s'agit en effet de savoir quand ajouter un nœud dans le voisinage, et quand doit-il être enlevé.

Le "réveil" d'un nœud doit tout simplement avoir lieu quand la surface arrive à proximité immédiate de l'un de ses voisins. Une mise en œuvre simple peut donc se faire lors du calcul de la vitesse  $d\mathbf{X}/dt$  sur un nœud : si la surface passe dans un des voxels adjacents, on vérifie que tous nos voisins sont déjà actifs, et, au besoin, on réveille ceux qui ne le sont pas. Autrement dit, on propage l'information à ses proches voisins : par ce principe de "bouche à oreille", on anticipera l'évolution de la surface en activant les nœuds quand nécessaire.

Quant à la suppression d'un nœud, elle s'effectue lors du seuillage : les nœuds de la grille sortant de l'intervalle  $] -1, 1[$  sont seuillés et sortis de la liste chaînée, car ils ne font plus partie du voisinage tubulaire de la surface.

Par ces principes très simples, on tient à jour la liste des nœuds dans le voisinage tubulaire, avec un surcoût minime de calcul.

---

2. Voir l'annexe C pour une introduction à la viscosité numérique.

### 7.5.6 Construction du maillage et de ses normales

Comme on l’a vu précédemment, l’affichage de la peau à la fin de chaque pas de temps peut se faire de façon simple et rapide. En parcourant les voxels compris dans le voisinage tubulaire de la surface, on génère des polygones en traitant chacun des voxels selon les valeurs du potentiel aux sommets. On construit ainsi une liste de points d’échantillonnage et une liste de facettes s’appuyant sur ces points.

Cependant, pour permettre un rendu agréable de type “shading<sup>3</sup>”, il faut aussi associer à chaque point une normale. Comme l’on dispose par différences finies des normales à la surface, il suffit de les interpoler pour trouver la normale à chaque point du maillage [LC87]. On dispose ainsi de toutes les données nécessaires pour soit faire un rendu immédiat, soit stocker les informations en vue d’y appliquer un lancer de rayons pour améliorer encore la qualité des images.

Appliqué à des surfaces cibles très lisses, l’algorithme décrit ci-dessus exhibe malheureusement une sorte d’*acné de surface* quand la tension de surface est nulle : autant la polygonalisation est irréprochable, autant le shading crée des effets de petites bosses sur toute la surface, comme du bump mapping. C’est en effet un défaut qui est dû à des petites variations de la normale par rapport à la véritable normale : on reconstruit bien la surface sur la grille, mais les normales ne sont qu’approximées par différences finies. Cet inconvénient peut être réglé par un simple *lissage*. Et ce lissage est justement obtenu par une tension de surface, même très faible. En conséquence, la peau générée par le modèle muni d’une tension de surface apparaît parfaitement lisse, sans avoir recours à un traitement particulier des normales.

### 7.5.7 Algorithme global

La mise en œuvre des points détaillés ci-dessus peut se faire en deux passes : il faut en effet faire l’évaluation des vitesses sans toucher aux valeurs de  $f$  dans un premier temps, puis, lors d’une deuxième passe, les mettre à jour en intégrant. Ce qui résulte en l’algorithme suivant :

À chaque pas de temps  $dt$ ,

- Pour tous les points  $\mathbf{X}$  de la grille référencés par la liste chaînée (*voisinage tubulaire*)
  - Calculer  $\nabla f(\mathbf{X}, t)$  par différences finies.
  - Calculer  $\frac{d\mathbf{X}}{dt}$  comme décrit aux sections 7.5.3 et 7.5.4.
  - En déduire  $\frac{\partial f}{\partial t}(\mathbf{X}, t) = -\nabla f(\mathbf{X}, t) \cdot \frac{d\mathbf{X}}{dt}$
- Pour ces mêmes points  $\mathbf{X}$ 
  - Mettre à jour :  $f(\mathbf{X}, t + dt) = f(\mathbf{X}, t) + \frac{\partial f}{\partial t}(\mathbf{X}, t) dt$

Visualiser la nouvelle surface en polygonalisant les voxels du voisinage tubulaire.

### 7.5.8 Pas de temps

Le dernier point important à aborder est le problème du choix du pas de temps pour l’intégration, afin de disposer finalement d’une méthode stable et précise.

#### Cas de la stratégie sans tension de surface

L’équation différentielle 7.1 est de type purement hyperbolique selon la classification décrite dans l’annexe C dans le cas où il n’y a pas de tension de surface, c’est-à-dire si  $d\mathbf{X}/dt =$

3. Rendu de la surface tenant compte des normales pour le calcul de l’éclairage.

$(d\mathbf{X}/dt)_{cible}$ . Son pas de temps doit donc suivre le classique critère de Courant-Friedrichs-Levy, à savoir, dans notre cas :

$$dt \leq \frac{\Delta x}{v} \quad (7.5)$$

où  $v = \max \|\frac{d\mathbf{X}_i}{dt}\|$  représente la vitesse maximale courante des nœuds. Ce critère est en tout point analogue au théorème de Shannon, qui stipule qu'il faut échantillonner le temps au double de la fréquence la plus élevée d'un signal pour ne pas perdre en qualité.

Au vu de l'algorithme que nous utilisons, il faut aussi vérifier que la surface ne puisse "traverser" plus d'un nœud en un pas de temps : sinon, notre liste chaînée stockant les nœuds du voisinage tubulaire risquerait de n'être plus valide. Il faut et il suffit, pour éviter cela, que  $V_{cible} \cdot dt$  soit inférieur à  $\Delta x$  : on retrouve exactement le critère de Courant. Ce n'est donc pas une contrainte supplémentaire.

### Cas de la stratégie avec tension de surface

L'ajout d'une tension de surface modifie la taille du pas de temps à utiliser. Le terme de courbure change en effet le type d'équation différentielle : on rajoute une contribution *parabolique* à une équation hyperbolique. La contrainte sur le pas de temps devient alors :

$$dt \leq \frac{\Delta x^2}{2D}$$

où  $\Delta x$  représente la taille de la discrétisation spatiale, et  $D = \beta \max \|\nabla f(\mathbf{X}_i)\|$  représente la diffusion due à la tension de surface.

Finalement, le pas de temps choisi dans notre implémentation s'écrit :

$$dt = \min\left(\frac{\Delta x}{v}, \frac{\Delta x^2}{2D}\right) \quad (7.6)$$

## 7.6 Extensions des propriétés de la peau

Le modèle de base étant défini, il est possible de l'enrichir de nouvelles propriétés physiques, sans que l'algorithme ne change. Ces propriétés, comme la surface ou le volume constant, étaient délicates à mettre en œuvre dans les modèles implicites précédents [DG95] : elles trouvent pourtant ici une solution simple.

### 7.6.1 Détection d'appartenance accélérée

La mise en œuvre de la peau décrite jusqu'ici amène une première retombée, d'importance : la détection des collisions avec la peau est particulièrement optimisée. En effet, pour un point de l'espace donné, il est aisé de déterminer s'il est à l'intérieur ou à l'extérieur de la peau, puisqu'il suffit d'interpoler entre les valeurs discrètes de  $f$ , puis de comparer le résultat à l'isovaleur. Contrairement à notre ancien modèle où le temps d'évaluation du potentiel dépendait amplement du nombre de squelettes, on obtient ici une évaluation *en temps constant*, et faible.

### 7.6.2 Surface constante

Lors de la polygonalisation, il est aisé de sommer la surface de chaque triangle généré afin d'obtenir une approximation de l'aire totale de la surface. On peut alors, à l'instar de la tension de surface, rajouter une pénalisation qui permettra par exemple de conserver une aire donnée. Il suffit pour cela d'ajouter le terme :

$$\frac{d\mathbf{X}}{dt}(t)_{surface} = \gamma \frac{(A_0 - A(t))}{A_0} \mathbf{n}(\mathbf{X})$$

à la stratégie de champ de vitesses. Cela fera enfler ou désenfler globalement l'objet qui gardera une surface constante, ou qui peut, sans plus de difficultés, suivre une évolution donnée si l'on fait varier  $A_0$  au cours du temps.

### 7.6.3 Volume constant

Le volume total approximé par la représentation polygonale peut être tenu à jour très simplement aussi. À chaque changement de signe d'un nœud, il suffit d'ôter ou d'ajouter le volume d'un voxel pour disposer d'une approximation suffisante du volume. Une implémentation plus approfondie peut calculer le volume finement en tenant compte de la polygonalisation de chaque cube. Dans les deux cas, il suffit d'ajouter le terme de pénalisation :

$$\frac{d\mathbf{X}}{dt}(t)_{volume} = \gamma \frac{(V_0 - V(t))}{V_0} \mathbf{n}(\mathbf{X})$$

pour que la stratégie tienne également compte du contrôle de volume.

Il est intéressant de remarquer ici que cette méthode de conservation de volume est très similaire à la technique proposée par Emmanuel Promayon [Pro97]. Il a en effet prouvé que le gradient de la contrainte d'incompressibilité d'un objet maillé en un sommet du maillage est colinéaire à la moyenne des normales des facettes adjacentes. C'est-à-dire, dans notre cas, que la direction maximisant la variation de volume est *celle de la normale à notre peau*: le choix du déplacement selon la normale, fait au paragraphe 7.4.5, est donc justifié *a posteriori* dans ce cas.

## 7.7 Résultats

Cette peau active a bien entendu été testée dans une optique d'habillage de modèles physiques, mais d'autres applications ont été envisagées.

### 7.7.1 Morphing

Une application directe de la technique décrite dans ce chapitre est le *morphing* (métamorphose) entre objets implicites. En effet, si la peau est initialisée sur une forme de départ, et que l'on change la définition de sa surface cible, la peau se met à évoluer de façon autonome vers la nouvelle forme.

La spécificité d'une telle méthode vis-à-vis des techniques classiques de métamorphose entre objets implicites définis par squelettes est qu'il n'est absolument pas nécessaire de faire une phase de correspondance entre squelettes [Gal97]. La peau se transforme d'une forme à une autre sans aucune indication. C'est bien sûr aussi un défaut : on ne pourra pas contrôler comment la métamorphose doit se faire. Seuls les variations des potentiels et les paramètres  $\alpha$  et  $\beta$  pourront être

modifiés pour changer le comportement obtenu.

La figure 7.6 montre une métamorphose entre deux objets de même topologie. L'objet de départ est la composition d'un vissage et d'un potentiel défini par un segment, alors que l'objet cible est un potentiel analytique. Les paramètres utilisés sont respectivement  $\alpha = 2.0$ ,  $\beta = .01$ , et le potentiel discret est stocké sur une grille de taille  $50 \times 50 \times 50$  pour obtenir une représentation très fine. Le temps de calcul moyen entre deux images est inférieur à la seconde sur une *Indy R4600*.

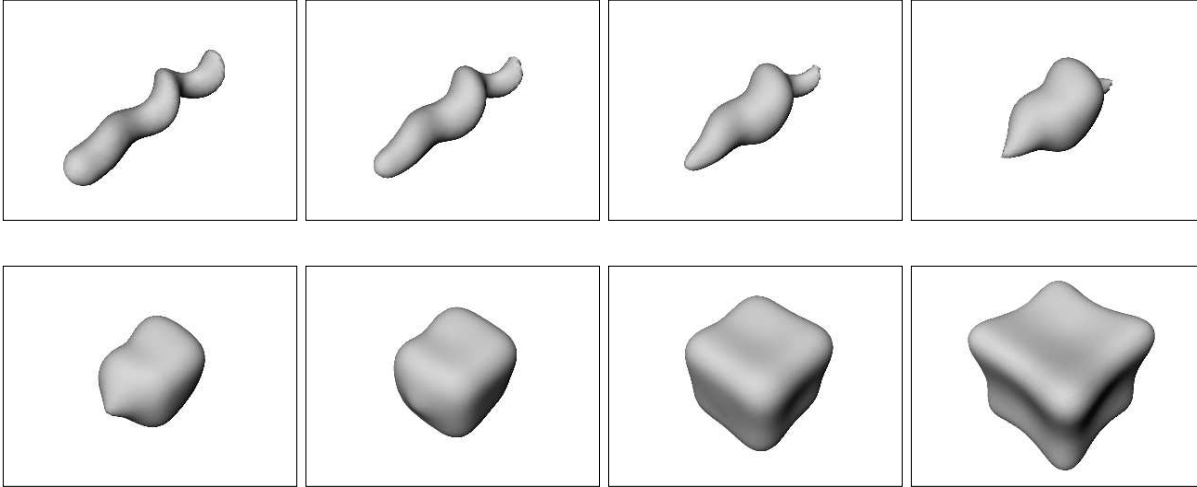


FIG. 7.6 - Exemple de métamorphose entre deux objets de même topologie.

Le deuxième exemple illustré par la figure 7.7 montre une métamorphose entre deux objets de topologies différentes. L'objet de départ est un mélange de six squelettes ponctuels, et l'objet cible est une paire de sphères, les paramètres restant inchangés par rapport à l'exemple précédent. La configuration des deux objets implicites offre ici un effet intéressant : quatre des six lobes de l'objet initial semblent caviter, alors que les deux derniers lobes paraissent s'écarter sur les côtés. Les surfaces intermédiaires exhibent alors des arêtes presque vives. Le filament joignant les deux sphères disparaît peu à peu, pour finalement donner lieu à une séparation.

Le nombre d'images générées dépend du pas de temps choisi pour la simulation de la peau : plus le pas est petit, plus la métamorphose sera lente, passant par une multitude de formes intermédiaires.

On peut penser à d'autres stratégies, apportant d'autres métamorphoses. La formulation utilisée par Figueiredo [FGTV92] peut également être appliquée dans notre cas, résultant en un champ de vitesses défini par :

$$\frac{d\mathbf{X}}{dt}(t)_{cible} = \alpha \text{sign}(G_{cible}(\mathbf{X}) - iso) \nabla G_{cible}(\mathbf{X})$$

où *sign* est une fonction retournant 1 ou  $-1$  selon le signe de son argument. La peau n'aura plus le comportement d'une bulle gonflant et dégonflant localement : elle "remontera" désormais le champ de gradient de l'objet cible pour atteindre la surface, ce qui peut créer des effets visuels tout aussi intéressants.

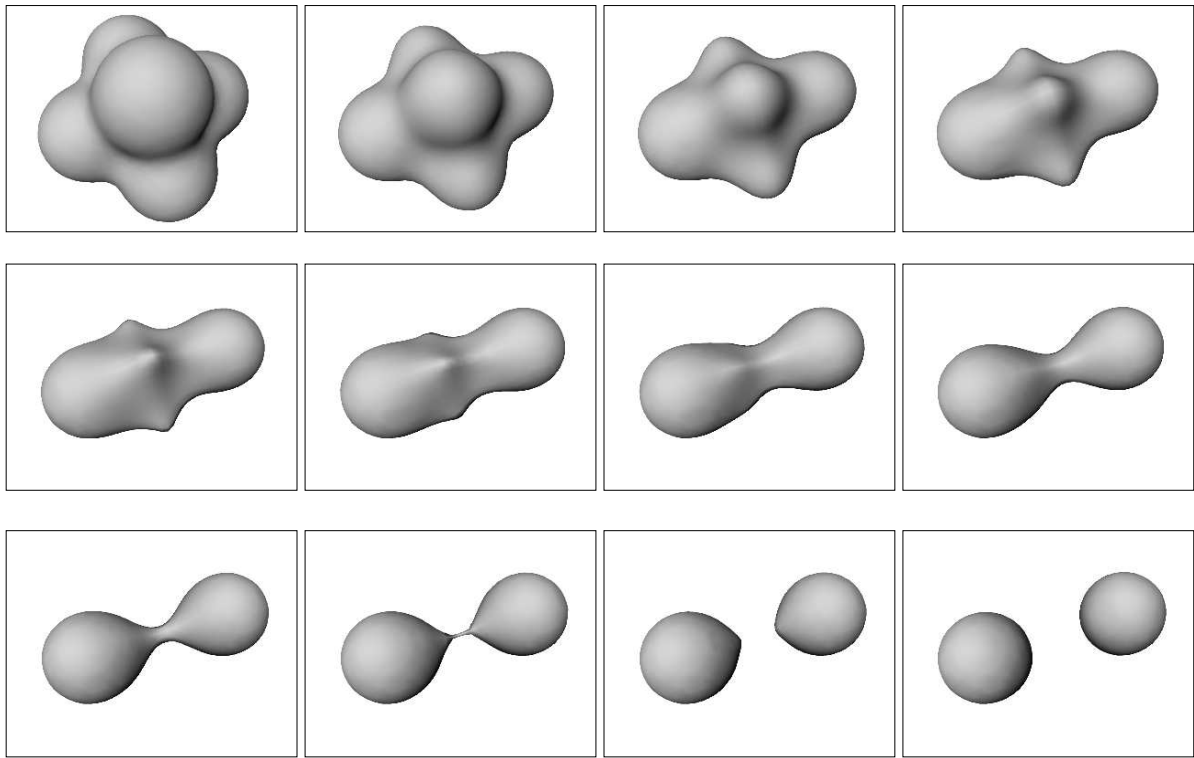


FIG. 7.7 - *Exemple de métamorphose entre objets de topologies différentes*

### 7.7.2 Habillage de particules

Notre modèle de peau peut aussi être incorporé dans notre modèle générique défini au chapitre 4. Cette fois, c'est la chair qui tient le rôle de cible mouvante.

#### Particules conventionnelles

Les premiers essais ont été faits avec des systèmes de particules conventionnels. Ces particules servant de squelettes à une surface implicite définissant la chair, il est alors simple de faire converger la peau vers cette surface implicite qui évolue en fonction du mouvement des particules.

Dans cet exemple, nous avons simulé la peau à partir des positions de particules obtenues lors de la simulation dépeinte sur la figure 6.9. Il n'y a donc pas d'interaction entre peau et particules. On voit cependant sur la figure 7.8 que la peau enrobe les particules et dissimule ainsi la granularité du modèle. On remarque aussi que la tension de surface permet d'obtenir divers aspects pour une même animation de particules.

#### Particules adaptatives

Enfin, des tests ont été menés pour l'enrobage de particules adaptatives. La figure 7.9 montre un des résultats obtenus, gommant effectivement la granularité intérieure du matériau. Le surcoût dû à cet habillage est cependant minime, car l'utilisation d'une grille même relativement grossière suffit souvent. La figure 7.10 permet de voir le nombre de sommets actifs de la grille, et l'on comprends mieux au vu de ce petit nombre que l'enrobage des particules adaptatives est un

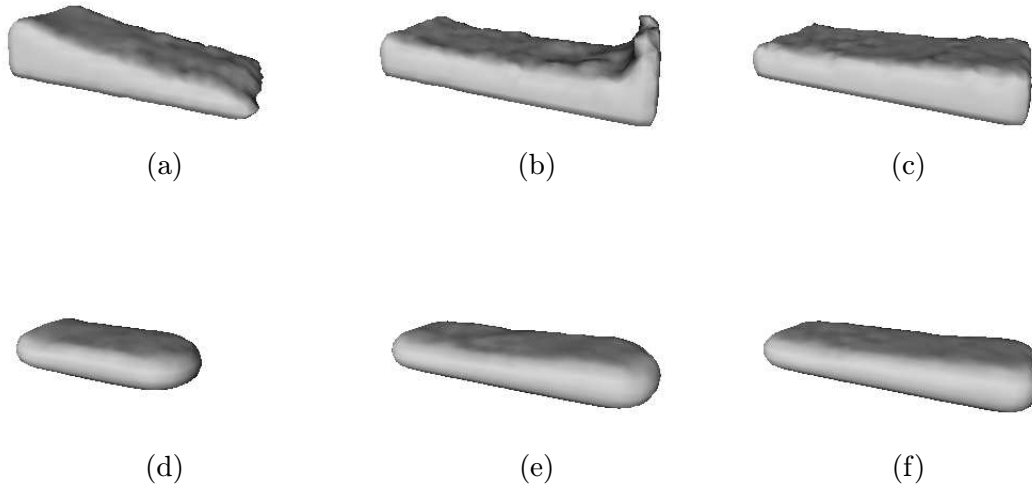


FIG. 7.8 - Exemples d'habillage de particules, pour  $\beta = .15$  (a-b-c) et  $\beta = .3$  (d-e-f), et une taille de grille de  $40 \times 20 \times 20$  (voir figure 6.8).

processus au temps de calcul négligeable par rapport à la simulation des particules elles-mêmes. Cela est d'ailleurs confirmé par les temps effectifs de la peau implicite, variant du temps réel à maximum une seconde de traitement entre deux images successives.

Pour être conforme à notre modèle générique, il faudra encore dans un proche avenir pouvoir transmettre les forces au système de particules sous-jacent pour rendre la rétroaction entre peau et chair effective. Cela semble tout à fait possible, lors des évaluations du potentiel cible, grâce aux points d'échantillonnage. Nous n'avons pas encore développé cette dernière mise au point.

## 7.8 Conclusion et perspectives

Ce chapitre a présenté une nouvelle technique de simulation de peau [DCG97]. Le modèle utilise une formulation implicite discrète, apte à gérer tout changement de topologie. À partir des points d'échantillonnage trouvés au pas précédent, le potentiel discret définissant la peau est modifié en fonction du champ de vitesses imposé, provoquant donc de façon indirecte le mouvement souhaité pour la peau. Ce comportement, déterminé par le champ de vitesses à chaque instant, peut être défini par plusieurs stratégies. Nous avons choisi, dans le cadre de l'animation de matériaux hautement déformables, d'attirer la peau vers la chair tout en simulant une tension de surface. Les paramètres d'attraction ( $\alpha$ ) et de tension ( $\beta$ ) étant macroscopiques, leurs effets sur le mouvement sont très intuitifs.

Cette méthode rapide, car réduisant ses calculs à une proche périphérie de la surface, permet une gestion de surfaces implicites mouvantes de complexité variable sans surcoût de calcul contrairement aux méthodes à base de squelettes. De plus, le temps de détection de collision est encore réduit grâce à l'emploi d'un potentiel simplissime. Enfin, la surface générée pouvant être munie de propriétés physiques supplémentaire comme une surface ou un volume constant, son

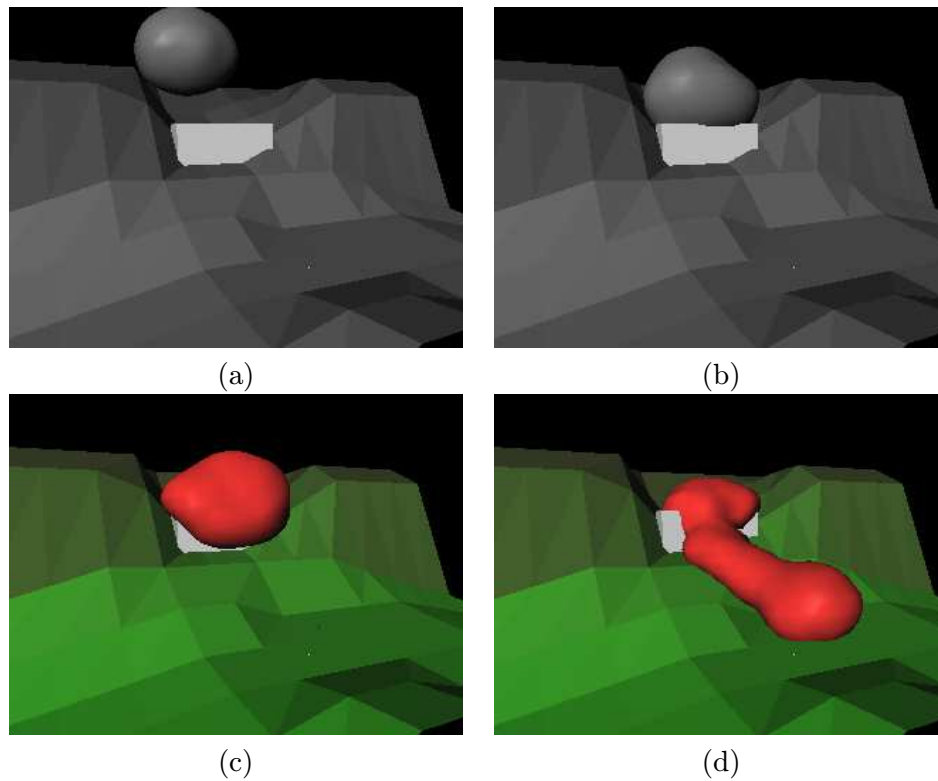


FIG. 7.9 - *Peau implicite appliquée sur l'animation adaptative de la figure 6.12.*

usage permet d'espérer une gamme importante d'applications.

Utilisée en tant qu'enrobage d'un système de particules quelconque, cette peau assure un filtrage temporel intéressant. Selon la tension de surface et la vitesse d'attraction définies, un mouvement pré-calculé pourra être grandement filtré par la simulation du mouvement de la peau.

Une autre utilisation immédiate de cette peau est l'habillage de modèles Euleriens [FM96]. En effet, dans la description Eulerienne, le champ de vitesses du matériau est déjà géré : la peau pourra ainsi suivre les mouvements générés de façon aussi simple qu'avec les particules.

Enfin, on peut songer à ajouter une détection des changements de topologie, qui peut être utile à la gestion de la chair. Bien qu'un parcours du tableau des valeurs du potentiel puisse permettre de trouver le nombre de composantes connexes, une méthode simple comme la caractéristique d'Euler-Poincaré permet, à partir du nombre courant de triangles, d'arêtes et de sommets dans la polygonalisation de la peau, de déterminer les changements de topologie, comme utilisé dans [LM96].

Une autre voie de développement possible est la parallélisation de l'algorithme : une grande partie de cette méthode porte en effet sur des calculs indépendants pour chaque noeud de la grille. Les temps de calcul seraient dans ce cas encore réduits si plusieurs processeurs pouvaient travailler de concert.



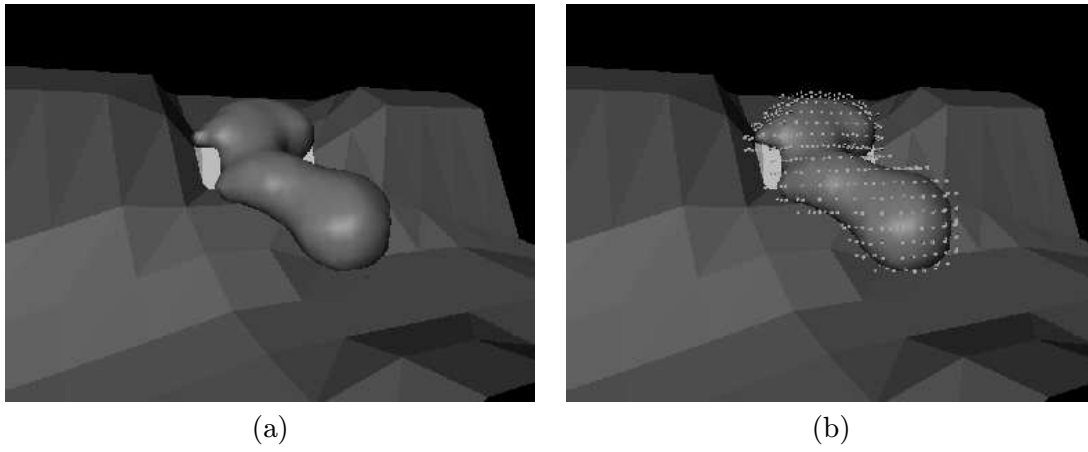


FIG. 7.10 - Une image de peau active (a), et ses sommets actifs de la grille régulière (b).

## Chapitre 8

# Conclusions

Après avoir décrit l'évolution de nos recherches, nous souhaitons dans ce chapitre résumer nos différentes contributions et dresser un bilan du travail effectué. Il sera alors temps de réfléchir aux perspectives qui s'offrent à nous.

### 8.1 Contributions

Nous avons discuté tout au long de cette thèse des moyens pour modéliser et animer des matériaux hautement déformables. Pour ce faire, nous avons commencé par proposer l'utilisation de modèles hybrides combinant plusieurs techniques dont les systèmes de particules et les surfaces implicites, en exploitant au mieux leurs particularités respectives. Dans une deuxième phase, nous avons développé de nouvelles méthodes pour améliorer la facilité d'utilisation de notre modèle d'une part, et pour optimiser les calculs d'autre part. Ce processus a donné lieu à des contributions d'ordres divers et d'importances variées.

#### 8.1.1 Mise au point d'un modèle global stratifié

Nous avons présenté dans un premier temps un modèle général qui permet d'enrichir n'importe quel modèle physique existant en l'interfaçant avec un enrobage implicite définissant la surface de l'objet. La représentation implicite utilisée permet une détection rapide doublée d'une modélisation fine et efficace des collisions entre objets, ainsi que des auto-collisions entre différentes parties d'un même objet. Une méthode adaptative d'échantillonnage de surface permet de faire tous ces calculs en des temps faibles, grâce à l'exploitation de la cohérence temporelle [DTG95, DTG96]. Les objets ainsi modélisés peuvent subir des séparations ou des fusions, et leurs variations de volume peuvent être contrôlées aisément lors de l'animation [DG94, DG95]. Utilisée avec succès pour la simulation de substances hautement déformables par habillage d'un système de particules, cette approche peut être combinée avec divers modèles et offre ainsi une méthode générique pour l'animation d'objets déformables [CGD97].

#### 8.1.2 Émergence d'un modèle de particules adaptatives

Dans un deuxième temps, l'utilisation de systèmes de particules s'est révélée aussi délicate pour l'animateur qu'inutilement gourmande en temps de calcul. Après avoir analysé les défauts constatés, nous avons proposé un nouveau modèle de particules, analogue à une méthode d'éléments finis, dont on peut définir le comportement macroscopique à l'aide d'une équation d'état.

Les forces inter-particulaires sont alors calculées sur chaque particule en fonction de l'état de son voisinage immédiat, de façon à suivre la contrainte sur le comportement global [DG96].

Puis, cette méthode a été étendue à des particules adaptatives, permettant de raffiner automatiquement en temps et en espace la description du matériau simulé en fonction des déformations subies. Grâce à l'équation de comportement macroscopique portant sur la densité, le système ainsi géré anime une substance à différents niveaux de résolution spatiale et temporelle afin de minimiser les calculs à produire, principe encore rare en animation.

### 8.1.3 Définition d'une peau active avec formulation implicite discrète

L'ancien habillage implicite ne convenant plus à un modèle adaptatif, notre dernière contribution a été de définir un modèle de peau active, capable entre autres de simuler une tension de surface, tout en gardant une formulation implicite, mais en utilisant un potentiel discret cette fois. Similaire à un modèle de snake, il offre une régularisation de l'habillage en ne nécessitant que des calculs locaux et simples. On dispose ainsi d'un modèle générant une polygonalisation fine de nos matériaux quels que soient les changements de topologie, tout en restreignant les calculs d'évolution de la surface à un proche voisinage [DCG97].

## 8.2 Réflexions et bilan

Au cours des trois années qu'a duré cette thèse, notre vision des modèles en images de synthèse a peu à peu évolué au rythme de notre progression. Nous proposons dans cette section d'exposer le cheminement des idées pour mieux recadrer notre contribution dans le domaine.

### 8.2.1 Synthèse du mouvement

#### De la maniabilité des modèles très déformables

Nous avons vu en passant l'état de l'art en revue que les systèmes de particules semblaient un des modèles les mieux adaptés à la simulation d'objets hautement déformables. En effet, la capacité de fractures ou de fusions de tels matériaux rend les techniques classiques issues de modèles continus incapables de les simuler, nécessitant donc une approche discrète où la gestion des changements de topologie n'est plus problématique.

Cependant, les approches discrètes développées ces dernières années ont toutes proposé l'utilisation de forces binaires fixes entre masses, ce qui se révèle être totalement non intuitif pour un animateur. La forme de la force inter-particulaires renseigne en effet peu sur le comportement du système en cours d'animation. Les résultats sont pourtant intéressants dans la mesure où des lois fixes simples entre chaque paire de particules entraînent un mouvement global complexe.

Mais pouvoir définir le mouvement d'un objet *de façon macroscopique* est fondamental en animation. Sans un tel contrôle sur les objets à animer, l'animateur devra passer des heures à régler des paramètres peu intuitifs, dont l'effet sur le comportement n'est absolument pas évident. Ce problème est d'autant plus important pour les objets à grand champ de déformation dont les mouvements complexes seront très difficiles à affiner.

#### Vers une définition macroscopique du comportement

Les systèmes de particules conventionnels sont une généralisation des réseaux masses-ressorts. Comme les ressorts, les particules disposent de lois d'interaction binaire de formes fixées. Ces

interactions indépendantes de l'état courant du voisinage nous ont paru très restrictives. Tout d'abord, elles ne permettent pas d'établir des critères locaux de stabilité : il faut par exemple calculer l'énergie totale d'un système pour vérifier la bonne intégration du mouvement [Jou97]. De plus, le potentiel de Lennard-Jones étant une moyenne empirique des forces réelles à l'échelle microscopique, choisir une force fixe de ce type pour une échelle macroscopique paraît être une forte approximation. Cela nous semble tout à fait similaire à l'hypothèse d'*état constant d'équilibre statique*, faite dans la théorie de l'élasticité, puisque les forces de Lennard-Jones ont pu difficilement être mesurées dans le cas dynamique; mais c'est une hypothèse qu'il reste à vérifier. Il est en tout cas plus intuitif que les forces mises en jeu entre petits éléments de masse dépendent de l'état actuel (compression, dilatation) de la matière localement.

Nous avons donc proposé l'utilisation d'un nouveau modèle de particules, peu différent du point de vue implémentation, mais dont les bases permettent de ne plus avoir à définir les forces d'interaction. On peut en effet considérer les Smoothed Particles comme une généralisation des systèmes de particules classiques qui approximaient les interactions par une force fixe (voir figure 8.1). *Le mouvement est maintenant appréhendé de façon macroscopique* par une équation d'état, véritable *loi de comportement global*. De cette équation sont dérivées les forces d'interaction entre les points d'échantillonnage actuels, ce qui engendrera le mouvement du matériau souhaité. Ainsi, nous utilisons un modèle à la fois *phénoménologique* dans la mesure où l'équation d'état peut être totalement "aphysique", mais tout de même *basée sur la physique* puisque l'on dérive de l'équation d'état des forces de pression dont on intègre les effets.

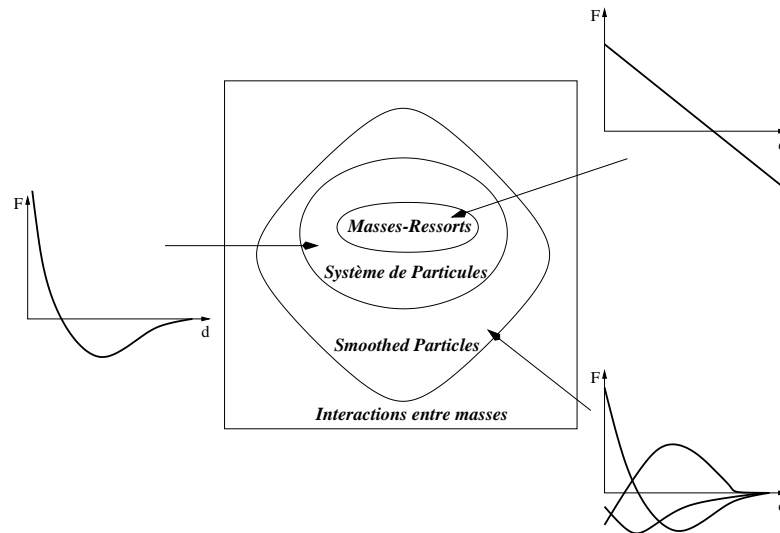


FIG. 8.1 - Généralité des différentes interactions entre masses.

### Vers une localisation des calculs

Actuellement, on remarque une émergence de nouvelles techniques d'analyse en images de synthèse : multi-résolution, multi-échelles, ou niveaux de détails, autant de termes prouvant que la recherche en Synthèse d'Images analyse finement les besoins réels de calcul pour atteindre un effet voulu. En fixant des critères de qualité à respecter, on cherche à proposer des *résolutions efficaces* de phénomènes *complexes* en ne fournissant que l'effort de calcul indispensable. L'exemple de la radiositivité [SP94] est frappant : en partant d'un modèle physique d'échanges

radiatifs, des techniques de simulation d'éclairage ont été développées, combinant accélérations algorithmiques et gestion multi-échelles des scènes, qui aboutissent maintenant à des images d'un réalisme inégalé en des temps raisonnables. Ces techniques sont d'ailleurs maintenant en partie reprises par les physiciens, prouvant au passage que la transversalité des domaines scientifiques n'est pas un vain mot.

Notre modèle de particules adaptative propose une première approche en ce sens : l'animation d'objets déformables se fait en raffinant automatiquement selon les besoins un échantillonnage de la matière et du temps. Les calculs sont ainsi optimisés car localisés dans les régions subissant de fortes déformations. Pour cela, et contrairement aux particules conventionnelles, respecter une loi globale se fait par des calculs de forces d'interactions au niveau local. C'est cette approche consistant à dériver les forces locales d'une équation d'état qui permet non seulement de déduire des critères de stabilité locale en termes de pas de temps adaptés, mais qui entraîne aussi une capacité à adapter la résolution spatiale sans changer le comportement global.

### 8.2.2 Des surfaces implicites

Le cas des objets hautement déformables montre de façon évidente qu'une définition surfacique sous forme B-rep n'est ni intuitive, ni adaptée à l'animation. Les changements de topologie nous poussent en effet à revenir à la notion de surface comme *interface* entre l'intérieur et l'extérieur d'un matériau. Les surfaces implicites sont parfaites pour une telle modélisation, puisqu'elles permettent de ne plus considérer les changements de topologie comme un cas problématique au cours d'une animation.

#### Complexité et intérêt de la formulation implicite

Utiliser une modélisation par surfaces implicites, nécessitant la définition d'un potentiel grâce auquel la surface sera finalement construite, paraît de prime abord une solution complexe et bien peu directe. Mais cette apparente complexité a des retombées importantes en terme d'efficacité : nous avons montré tout au long de cette thèse combien il est efficace de pouvoir par exemple déterminer aisément l'appartenance à un objet grâce à une formulation définissant en fait le volume. En décrivant la surface non comme une équation paramétrique qui devra changer du tout au tout en cas de changement de topologie, mais comme l'interface d'un champ de présence spatiale qui peut évoluer naturellement vers une topologie différente, les surfaces implicites offrent une formulation pertinente.

Bien sûr, la rançon de la formulation implicite est l'absence de paramétrisation des surfaces. Cela demande donc de développer un certain nombre de méthodes pour maintenir une discrétisation ou gérer les mélanges. C'est pourquoi nous avons proposé dans cette thèse des solutions à ces problèmes spécifiques aux surfaces implicites, de façon à rendre leur utilisation plus simple.

#### Vers un modèle implicite à potentiel discret

L'emploi d'un nombre élevé de particules nous a amené à modifier l'utilisation de surfaces implicites engendrées par squelettes. Nous avons trouvé préférable de définir un *potentiel discret* pour définir notre surface. En effet, dans la majorité des cas, il est inutile de garder une représentation *continue* du potentiel : une bonne discrétisation de la surface suffit à la fois au modèle d'animation dédié au calcul des mouvements, et à la visualisation fine. Il nous a donc semblé opportun de simplifier la définition du potentiel en le considérant (tri)linéaire par morceaux,

et de ne stocker que des valeurs discrètes sur une grille. Si cette grille est suffisamment fine, le rendu visuel sera suffisant, et les calculs seront optimisés par rapport à une définition continue.

Notre modèle de peau s'appuie sur ce principe : à partir d'un ensemble de points à la surface (qui ont été déterminés au pas précédent), on modifie les valeurs du potentiel localement pour que, par effet de bord, la surface ait le comportement souhaité. Ces modifications pouvant se limiter à un voisinage de la surface, nous optimisons ainsi le nombre de calculs nécessaires au mouvement de la peau.

### 8.2.3 Bilan

La méthode sous-jacente à ces diverses approches peut se résumer simplement : nous avons trouvé préférable de *définir les objets et leur comportement de façon macroscopique*, puis de les *animer par le biais d'une discrétisation adaptative en dérivant localement les interactions* nécessaires à l'obtention du comportement souhaité.

Dans le cas des particules par exemple, une équation d'état définit le comportement global de la matière; puis la discrétisation spatiale et temporelle de la matière est localement adaptée en fonction de la finesse nécessaire à une bonne description du mouvement et à la bonne intégration des forces inter-particulaires dérivées de la contrainte sur le comportement.

De même, le modèle implicite discret maintient une discrétisation adaptée sur et autour de la peau selon les mouvements en cours, et choisit au mieux le pas de temps qui convient de façon à respecter l'attraction et la tension de surface définies globalement par l'utilisateur.

Ces techniques de raffinement/simplification sont couramment utilisées en rendu sur des scènes fixes, mais *l'intervention du temps complexifie grandement le problème*. Nous proposons ici des modèles adaptatifs pour l'animation qui prolongent l'emploi de ces techniques, et essaient d'optimiser le nombre de calculs nécessaires en *offrant une gestion plus fine de la complexité des mouvements*.

## 8.3 Horizons

Cette dernière section est une tentative de projection dans le futur. Le modèle de particules adaptatives et le modèle de peau utilisent en effet tous les deux des méthodes nouvelles, qui peuvent sûrement être étendues et améliorées.

### 8.3.1 Modèle adaptatif de simulation

Le nouveau modèle de particules paraît propice à de nombreux développements dans la mesure où il est, à notre connaissance, le seul à pouvoir assurer une simulation d'un même comportement à des échelles de résolution spatiale et temporelle adaptées.

Il peut intéresser à ce titre les géologues pour des *simulations de coulée de lave, d'éboulements ou d'avalanches de neige*, pour lesquelles des modèles macroscopiques existent. Des contacts avec le BRGM et le laboratoire 3S de Grenoble sont actuellement en cours afin de juger l'adéquation de notre modèle à leur besoin.

La *simulation d'organes* pour opérations chirurgicales est aussi un domaine en plein essor, où les modèles optimisés sont prisés pour permettre l'animation temps-réel. Le modèle des particules adaptatives tel que nous l'avons montré ici n'est peut-être pas idéalement adéquat pour la manipulation d'objets structurés, mais paraît une base de méthodologie prometteuse. Là encore,

une étude de faisabilité est en cours dans le cadre d'une action incitative INRIA. Gilles Debunne commence en effet sa thèse sur le niveau de détail en animation, et travaille actuellement sur un modèle hiérarchique d'animation d'objets structurés.

Enfin, l'idée même de l'animation par échantillonnage adaptatif en temps et en espace peut aisément être développée plus avant. L'utilisation de particules *anisotropes* (non sphériques), ou au contraire de modèles beaucoup plus simples respectant les lois de l'adaptation peut tout à fait être envisagée.

### 8.3.2 Peau implicite dynamique

La peau, issue d'un modèle très simple et donc efficace, peut aussi trouver des applications autres que l'animation.

Par exemple, il est connu qu'en échographie, la reconnaissance des contours d'un organe est beaucoup plus aisée si l'objet est en mouvement. Si la segmentation manuelle d'une image de cœur est très délicate, les médecins sont par contre capables de la distinguer immédiatement si le cœur est en train de battre : les informations dynamiques fournissent des indications subtiles, filtrant en quelque sorte les bruits dus à l'échographie. Il y a donc fort à parier que l'utilisation de la peau en 2D, en tant que contour dynamique, puisse permettre une *segmentation de séquences échographiques*, grâce au lissage qu'induit la peau. Les séquences de données volumiques 3D sont aussi envisageable, bien qu'encore peu répandues actuellement. Des images tests, fournies par le laboratoire TIMC avec lequel nous sommes en contact, permettront de vérifier ces hypothèses.

En *reconstruction d'objets* à partir de points épars, le modèle de peau peut aussi permettre une gestion robuste de la reconstruction d'objets de forme et de topologie quelconques.

Enfin, cette peau implicite semble être un point de départ intéressant pour la *sculpture interactive*. La combinaison d'une formulation implicite et d'un maillage rapide permet d'envisager une manipulation de la peau en réalité virtuelle, laquelle pourra alors être déformée à l'aide d'outils, avec une résolution variable, lissée par filtrage, etc... Des industriels comme Renault s'intéressent à cet aspect pour la mise au point de formes en design, qui actuellement se fait ... en terre glaise. Là encore, une thèse qui vient de débiter, menée par Eric Ferley, devrait étudier les avantages et les inconvénients de notre modèle dans ce cadre.

## 8.4 Conclusions

Cette thèse nous a donné l'occasion de définir et d'utiliser divers modèles de matériaux hautement déformables. Certaines parties ont d'ores et déjà donné lieu à des applications, comme dans le domaine de prévisions des risques naturels. Mais les autres parties sont plus théoriques et novatrices, et bien que des collaborations commencent à fleurir, il reste à finaliser leurs utilisations. Ce sera alors l'occasion de peut-être simplifier ou plus généralement modifier nos modèles initiaux pour les rendre tout à fait simples et rapides.

Mais le principal intérêt à long terme de cette étude nous semble être *la faisabilité de l'animation adaptative*. L'aspect dynamique de l'animation avait jusque là grandement empêché le développement de techniques optimisés, comme il existe depuis plusieurs années dans le rendu d'images. Le modèle de particules tend à prouver que l'obstacle du temps peut être affronté. Il faudra alors continuer l'effort dans ce sens pour arriver à des modèles directement utilisables en Réalité Virtuelle, perspective motivante de par ses retombées diverses.

## Quatrième partie

### **Annexes**





---

## Annexe A

# Euler vs. Lagrange

Ou deux façons différentes de voir les mêmes choses

**D**écrire un mouvement de matière, c'est mesurer les variations locales de position, de vitesse, ou autres, se produisant au cours du temps. Pour mesurer ces variations, deux approches distinctes existent, connues sous le nom d'approches Eulerienne et Lagrangienne.

### A.1 Vision Lagrangienne

Une première façon d'appréhender le mouvement de la matière est de se mettre "à la place" de la matière. On considère donc une petite quantité de masse de matière, et *on suit son évolution dans le temps*. On pourra ainsi relever l'évolution de la position, de la vitesse, de la pression, ou de tout autre grandeur physique mesurée sur l'élément que l'on trace. Modéliser un phénomène par cette approche consiste alors à définir une discrétisation initiale de la matière, puis à décrire l'évolution temporelle de chacun des petits éléments de masse.

Une simulation Lagrangienne fait donc évoluer un ensemble de masses élémentaires dans l'espace, calculant à chaque pas de temps leurs interactions selon les propriétés du matériau simulé et intégrant les déplacements créés. On dispose alors d'une description discrète du matériau dont la position des points d'échantillonnage évoluent au cours du temps.

### A.2 Vision Eulerienne

Une seconde façon d'appréhender un phénomène complexe est de le regarder évoluer depuis un repère fixe. Si chaque petite portion de l'espace est scrutée afin de déterminer ce qui s'y passe, le phénomène entier pourra être décrit.

De façon plus précise, on peut par exemple considérer une partition de l'espace en volumes élémentaires juxtaposés. Si pour chacun de ces volumes, on quantifie ce qui entre et ce qui sort au cours du temps, on disposera d'une bonne description des transports de la matière, et donc de l'ensemble du mouvement dans l'espace.

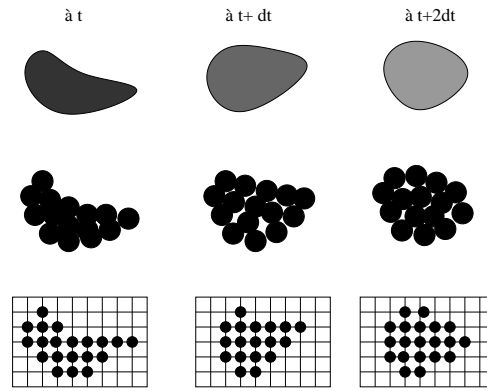


FIG. A.1 - De haut en bas : un phénomène quelconque, puis une approche Lagrangienne et une approche Eulerienne pour simuler son mouvement

Une simulation Eulerienne classique utilisera donc une grille fixe pour stocker dans chaque nœud la densité et la vitesse du matériau en ce point. Les calculs du mouvement se feront en chaque voxel en actualisant la densité en fonction, entre autres, des mouvements induits par le champ de vitesses. On dispose alors aussi d'une description discrète, mais cette fois, les points d'échantillonnage sont fixes dans l'espace.

### A.3 Expression de l'accélération en formalisme Lagrangien

L'accélération d'une parcelle de matière dans une simulation Lagrangienne est facile à exprimer : la vitesse  $\mathbf{v}(t)$  d'une particule donnée ne dépend en effet que du temps. L'accélération de la particule s'écrit alors très simplement :

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} \quad (\text{A.1})$$

Plus généralement, pour n'importe quel champ  $\phi$  porté par une particule, sa dérivée temporelle est :  $d\phi/dt$ , et peut donc être très aisément approximée par  $(\phi(t + \delta t) - \phi(t))/\delta t$ .

### A.4 Expression de l'accélération en formalisme Eulerien

Dans le formalisme Eulerien par contre, la dérivée  $\partial\mathbf{v}(\mathbf{X}, t)/\partial t$  n'est pas l'accélération d'un élément à la position  $\mathbf{X}$  au temps  $t$ , car cet élément n'occupe cette position que de façon instantanée. La vitesse  $\mathbf{v}(\mathbf{X}, t)$  est fonction à la fois du temps et de la position de l'élément considéré. Sa différentielle s'écrit alors :

$$d\mathbf{v} = \frac{\partial\mathbf{v}}{\partial t}dt + \frac{\partial\mathbf{v}}{\partial x}dx + \frac{\partial\mathbf{v}}{\partial y}dy + \frac{\partial\mathbf{v}}{\partial z}dz$$

Il en résulte que l'accélération est définie par :

$$\begin{aligned} \mathbf{a} &= \frac{d\mathbf{v}}{dt} = \frac{\partial\mathbf{v}}{\partial t} + \frac{\partial\mathbf{v}}{\partial x} \frac{dx}{dt} + \frac{\partial\mathbf{v}}{\partial y} \frac{dy}{dt} + \frac{\partial\mathbf{v}}{\partial z} \frac{dz}{dt} \\ &= \frac{\partial\mathbf{v}}{\partial t} + v_x \frac{\partial\mathbf{v}}{\partial x} + v_y \frac{\partial\mathbf{v}}{\partial y} + v_z \frac{\partial\mathbf{v}}{\partial z} \\ \Rightarrow \mathbf{a} &= \frac{\partial\mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \end{aligned} \quad (\text{A.2})$$

Cette relation est en fait valable quelque soit le champ considéré dépendant et de la position et de la vitesse. On peut, par simplicité, écrire qu'en formalisme Eulerien, la dérivé temporelle totale est :

$$\frac{d}{dt} = \frac{\partial}{\partial t} + (\mathbf{v} \cdot \nabla) \quad (\text{A.3})$$

## A.5 Avantages et inconvénients des deux approches

Ces approches ne sont que deux façons distinctes d'aborder un problème. On ne peut en qualifier une de meilleure : pour faire une analogie avec l'algorithmique, elles expriment simplement la matière selon deux "structures de données" différentes. Selon les spécificités du problème abordé, il pourra donc y avoir des avantages à utiliser l'une par rapport à l'autre.

### A.5.1 Approche Lagrangienne

Une formulation Lagrangienne permet de disposer à tout instant de masses intérieures à la matière, permettant de la localiser (et donc de la *représenter*) rapidement. La détection des collisions est donc facilitée. Par contre, toute recherche de variations de champ nécessite une étape de recherche des plus proches voisins, qui peut selon les cas se révéler coûteuse.

### A.5.2 Approche Eulerienne

Au contraire, l'approche Eulerienne permet d'avoir des informations sur le voisinage en un temps réduit : la grille portant les valeurs des champs permet d'accéder en temps constant aux informations des noeuds voisins. À l'inverse, la grille porte des informations sur la densité de la matière présente : visualiser la matière nécessite donc un traitement particulier, allant de la simple interpolation à la simulation de marqueurs traçant les déplacements [FM96]. Les collisions ne peuvent être traités facilement qu'avec des objets voxelisés.



## Annexe B

# Notions d'hydrodynamique

Le terme de *fluide* rassemble à la fois les liquides et les gaz. En effet, leur propriété caractéristique commune est la facilité avec laquelle ils peuvent se déformer, contrastant avec les solides. Un matériau solide possède une forme donnée, qui change seulement lorsque les conditions extérieures varient. Un volume de fluide, au contraire, n'a pas de *forme* par défaut, et peut s'organiser de diverses façons. L'étude du comportement des fluides en fonction des forces agissant sur le système est l'objet de l'hydrodynamique [Bat73].

Dans cette annexe, nous verrons un aperçu des bases d'hydrodynamique qui nous ont été utiles lors du développement de cette thèse. Une simplification de taille a été opérée de façon à alléger la présentation : nous considérons les fluides comme *isothermes*, c'est-à-dire que le mouvement n'entraîne aucune modification de température au sein de la matière.

### B.1 L'hydrodynamique en général

L'hydrodynamique, ou mécanique des fluides, est l'étude du comportement des fluides. Le terme de fluide regroupe de nombreux matériaux, forts distincts : le métal en fusion, l'huile, l'air, ou l'eau sont tous des fluides avec pourtant des différences de comportement marquées. On comprend alors bien à quel point cette branche des sciences est complexe, nécessitant presque toujours des connaissances physiques et mathématiques importantes. Cependant, les bases de l'hydrodynamique sont relativement simples.

#### B.1.1 Grandeurs significatives

Tout d'abord, les grandeurs manipulées sont macroscopiques. Bien qu'un fluide, comme tout élément de matière, soit constitué d'un nombre élevé d'atomes en interaction, on fait l'hypothèse que le fluide est un milieu continu, ce qui à notre échelle paraît une bonne approximation. Ainsi, la *densité*  $\rho$  par exemple sera une propriété importante d'un volume de fluide : elle joue le même rôle dans le mouvement du fluide que la masse d'un solide dans la mécanique Newtonienne. La *viscosité*  $\mu$  est aussi une variable intéressante pour les fluides, qui peut affecter beaucoup le comportement globale. Enfin, la *pression interne*  $P$  est une grandeur macroscopique représentant la valeur statistique moyenne de la force qu'exerce une partie du fluide sur une autre. Cette force est en fait due aux mouvements des atomes qui créent un échange de quantité de mouvement entre régions voisines.

### B.1.2 Propriété notable : l'équation d'état

L'hydrodynamique modélise les différents fluides existants par une *relation liant les variations des différentes grandeurs au sein du fluide*. Cette relation, appelée *équation d'état*, est obtenue à partir de mesures lors d'expérimentations. Par exemple, on a remarqué que pour beaucoup de gaz, la pression était proportionnelle à la densité : on en a fait un modèle dit du "gaz parfait", dont l'équation d'état est  $PV = \text{constante}$ . Dans le cas général, on définira un fluide par son équation d'état  $P = P(\rho)$ <sup>1</sup>.

### B.1.3 Équations fondamentales

Les équations de base de l'hydrodynamique sont relativement peu nombreuses. C'est pourquoi nous nous proposons de les décrire dans les paragraphes qui suivent. L'idée sous-jacente est très simple : rien ne se perd, rien ne se crée, tout se transporte (pour parodier Lavoisier). Par exemple, une perte de matière locale devra être compensée par un gain ailleurs.

## B.2 Conservation de la masse

La conservation de la masse d'un fluide impose certaines restrictions sur son champ de vitesses. Pour trouver les contraintes induites, considérons une surface fermée  $A$  dont la position est fixe dans le repère du monde et qui entoure un volume  $V$  de fluide. Si  $\rho$  est la densité du fluide, alors la masse de fluide entourée par la surface s'exprime naturellement par :  $\int_V \rho dV$ . Par contre, le débit de masse sortant de la surface par unité de temps est :  $\int_A \rho \mathbf{v} \cdot \mathbf{n} dA$ , où  $\mathbf{v}$  est le champ de vitesses. Alors, la conservation de masse se traduit par l'égalité suivante :

$$\frac{\partial \int_V \rho dV}{\partial t} = - \int_A \rho \mathbf{v} \cdot \mathbf{n} dA.$$

La formule d'Ostrogradski (stipulant que le flux d'un champ à travers une surface est égal à l'intégrale sur le volume intérieur à la surface de la divergence du champ) nous permet de transformer cette équation en une seule intégrale totalement volumique :

$$\int_V \left( \frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{v}) \right) dV = 0.$$

Cette relation étant valide pour tout choix de surface, il faut que l'intégrande, continu, soit identiquement nul. Nous en déduisons donc la relation valide en tout point du fluide :

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{v}) = 0 \tag{B.1}$$

Cette équation, exprimant la *conservation de la masse*, est appelée *équation de continuité* (pour d'obscures raisons historiques).

En utilisant la relation A.3, cette équation prend une expression un peu plus simple :

$$\frac{d\rho}{dt} + \rho \text{div} \mathbf{v} = 0. \tag{B.2}$$

En exprimant l'équation ainsi, on peut interpréter la divergence de la vitesse comme une mesure locale de dilatation.

---

1. La température  $T$  intervient souvent dans les équations d'état, mais ici, nous avons fait l'hypothèse d'une température constante.

### B.2.1 Fluide incompressible

Un fluide est dit *incompressible* si sa densité n'est pas affectée par les changements de pression. La variation de densité étant donc nulle, on en déduit grâce à l'équation de continuité que la relation spécifique des fluides incompressibles est :

$$\operatorname{div} \mathbf{v} = 0,$$

autrement dit, il n'y a aucune dilatation ou compression au sein d'un fluide incompressible.

## B.3 Équation de la dynamique

Pour trouver l'équation gérant le mouvement d'un fluide, considérons une parcelle de fluide  $dV$ , de densité  $\rho$ , sous une pression locale de  $P$ . La force de pression exercée sur la parcelle est par définition  $\nabla P dV$ . La gravité exerce, quant à elle, une force égale à  $\rho dV \mathbf{g}$ . Enfin, la viscosité interne crée une force quantifiable par  $(\mu \nabla^2 \mathbf{v} + \frac{\mu}{3} \nabla(\operatorname{div} \mathbf{v}))dV$ . Le bilan des forces fait, il suffit d'appliquer la loi fondamentale de la dynamique pour obtenir l'équation de Navier-Stokes, après avoir éliminé le  $dV$  intervenant de part et d'autre de l'égalité :

$$\rho \frac{d\mathbf{v}}{dt} = \rho \mathbf{g} - \nabla P + \mu \nabla^2 \mathbf{v} + \frac{\mu}{3} \nabla(\operatorname{div} \mathbf{v})$$

Pour un fluide incompressible, le dernier terme n'a pas de raison d'être puisque la divergence de la vitesse est toujours nulle.

## B.4 Vitesse de propagation à l'intérieur d'un fluide

Il est intéressant de remarquer que tout fluide a une vitesse maximale de propagation des déformations qui lui est propre. Pour le comprendre, considérons un fluide parfait compressible non visqueux et non soumis à l'influence gravitationnelle (pour simplifier les calculs), vérifiant donc les équations suivantes :

$$\frac{d\rho}{dt} + \rho \operatorname{div} \mathbf{v} = 0 \tag{B.3}$$

$$\rho \frac{d\mathbf{v}}{dt} = -\nabla P \tag{B.4}$$

et une équation d'état de type  $P = P(\rho)$ .

On suppose que le fluide est initialement au repos ( $\mathbf{v} = 0$ ), avec des caractéristiques uniformes ( $P = P_0, \rho = \rho_0$ ). Étudions alors la dynamique des perturbations infinitésimales au sein du fluide, c'est-à-dire le comportement du système pour des champs :

$$\rho(\mathbf{X}, t) = \rho_0 + \rho'(\mathbf{X}, t)$$

$$P(\mathbf{X}, t) = P_0 + P'(\mathbf{X}, t)$$



où les fluctuations  $\rho'$  et  $P'$  sont de très faible amplitude. Il est donc légitime, au voisinage de l'état au repos, de linéariser les équations fondamentales B.3, B.4 et l'équation d'état, ce qui conduit tout naturellement à :

$$\frac{\partial \rho'}{\partial t} + \rho_0 \operatorname{div} \mathbf{v} = 0 \quad (\text{B.5})$$

$$\rho_0 \frac{d\mathbf{v}}{dt} = -\nabla P' \quad (\text{B.6})$$

$$P' = \left( \frac{\partial P}{\partial \rho} \right) (\rho_0) \rho'. \quad (\text{B.7})$$

L'équation B.5 se réécrit alors :

$$\frac{\partial P'}{\partial t} + \rho_0 \left( \frac{\partial P}{\partial \rho} \right) (\rho_0) \operatorname{div} \mathbf{v} = 0.$$

Enfin, en prenant la divergence de l'équation B.6, on peut transformer l'équation précédente pour obtenir :

$$\frac{\partial^2 P'}{\partial t^2} - \left( \frac{\partial P}{\partial \rho} \right) (\rho_0) \Delta P' = 0. \quad (\text{B.8})$$

On reconnaît l'équation classique de propagation des ondes, avec une vitesse :

$$c^2 = \left( \frac{\partial P}{\partial \rho} \right) (\rho_0).$$

Cette vitesse, dite *vitesse du son du matériau*, est donc une caractéristique du matériau que l'on peut déduire de l'équation d'état qui lie la pression et la densité.

## Annexe C

# Notions sur les équations différentielles

Une équation différentielle décrit une relation entre une fonction inconnue et ses dérivées. Résoudre une équation différentielle signifie donc trouver une fonction vérifiant cette relation. Dans le cadre de l'animation par synthèse du mouvement, la résolution des équations différentielles les plus courantes passe rarement par des solutions analytiques : nous sommes presque exclusivement confrontés à des problèmes nécessitant une résolution numérique. En partant de conditions initiales, on procède par estimation des différentes dérivées en jeu et on approxime pas à pas les variations de la fonction cherchée. Maintes solutions existent pour ce problème classique d'algorithmique numérique. Cette annexe ne se veut absolument pas un résumé (et encore moins un cours) des différentes méthodes existantes et de leurs avantages respectifs. Nous allons plus modestement essayer de passer en revue les notions de base nécessaires à la bonne compréhension de certains points importants abordés dans cette thèse, ou plus généralement en synthèse du mouvement.

### C.1 Équations différentielles ordinaires du premier ordre

Une première famille d'équations différentielles fort utilisées en animation, dites ordinaires du premier ordre, se formule de la façon suivante :

$$\dot{\mathbf{X}} = f(\mathbf{X}, t) \tag{C.1}$$

L'importance de cette famille est d'autant plus grande que toute équation différentielle ordinaire peut se ramener à un système d'équations du premier ordre couplées. Par exemple, l'équation classique du mouvement :

$$\ddot{\mathbf{X}} = \mathbf{F}/m$$

peut se ré-écrire en le système d'équations couplées suivant :

$$\dot{\mathbf{v}} = \mathbf{F}/m$$

$$\dot{\mathbf{X}} = \mathbf{v}$$

Examinons alors les différentes façons de résoudre de telles équations différentielles.

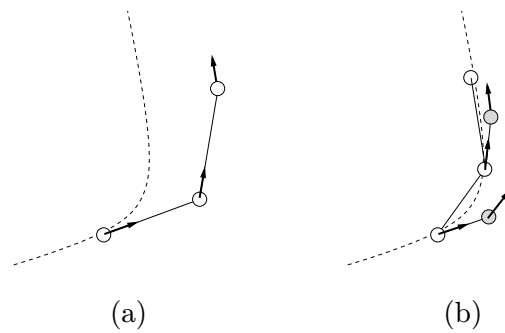


FIG. C.1 - Deux pas d'Euler (a), en comparaison à deux pas de Runge-Kutta (b)

### C.1.1 Méthode d'Euler

La méthode la plus simple pour intégrer cette famille d'équations est appelée la méthode d'Euler. Sa formulation est très simple : en supposant que l'on dispose d'une position  $\mathbf{X}$  à l'instant  $t$ , notée donc  $\mathbf{X}(t)$ , alors l'estimation d'Euler pour la position à l'instant  $t + dt$  est :

$$\mathbf{X}(t + dt) = \mathbf{X}(t) + f(\mathbf{X}(t), t) dt \quad (\text{C.2})$$

Si on se remémore la série de Taylor d'une fonction dérivable :

$$\mathbf{X}(t + h) = \mathbf{X}(t) + h\dot{\mathbf{X}}(t) + \frac{h^2}{2}\ddot{\mathbf{X}}(t) + \dots + \frac{h^n}{n} \frac{d^n \mathbf{X}}{dt^n}(t) + O(h^{n+1}),$$

on s'aperçoit alors que la méthode d'Euler consiste à négliger toutes les dérivées d'ordre supérieur à un : la méthode est dite d'ordre un.

La simplicité de cette approximation apporte malheureusement tout un tas de problèmes numériques : l'imprécision de l'approximation entraîne bien souvent des divergences. En fait, la formule est tout simplement dissymétrique : on intègre la fonction sur un intervalle de temps  $dt$  mais on utilise comme information seulement la dérivée initiale. Même en jouant sur la valeur de  $dt$ , appelé *pas de temps*, ou pas d'intégration, la fonction approximée dérive souvent très rapidement de la fonction réelle.

Pour augmenter la précision de l'approximation, on pourrait penser à diminuer la valeur du pas d'intégration  $dt$ ; mais alors, l'intégration sur un laps de temps donné demandera bien plus d'évaluations de  $f$  pour un résultat guère meilleur. Bref, en pratique, la méthode d'Euler est à la fois peu précise et peu efficace.

### C.1.2 Méthode de Runge-Kutta

Pour rendre plus symétrique l'intégration, on peut envisager d'évaluer la dérivée de  $\mathbf{X}$  au milieu du parcours afin de corriger un peu l'approximation :

$$\mathbf{X}(t + dt) = \mathbf{X}(t) + dt f(\tilde{\mathbf{X}}(t + dt/2), t + dt/2)$$

Une approximation simple de  $\tilde{\mathbf{X}}(t + dt/2)$ , sensé être le point à mi-parcours de l'évaluation, s'obtient par la méthode d'Euler :

$$\tilde{\mathbf{X}}(t + dt/2) = \mathbf{X}(t) + \frac{dt}{2} f(\mathbf{X}(t), t)$$

Cette méthode revient donc à faire une première évaluation par Euler, puis à approximer la dérivée en ce point pour finalement en déduire une bonne approximation finale. On montre, toujours à l'aide des développements limités, que cette méthode dite de *Runge-Kutta* est d'ordre deux cette fois. Mais on a dû faire une nouvelle évaluation de  $f$  à mi-parcours : la précision est à ce prix. La figure C.1 montre sur un cas simple que cette dernière technique est souvent bien plus précise que la méthode d'Euler.

Cette technique de base peut être aussi utilisée pour obtenir des approximations d'ordre quatre, et peut même se combiner avec l'utilisation de pas de temps adaptatifs, à un coût évidemment d'autant plus élevé [PTVF92].

### C.1.3 Méthode implicite ou semi-implicite

Après avoir formulé l'approximation sur un pas d'intégration en fonction de la dérivée initiale, puis de la dérivée à mi-parcours, une dernière possibilité peut s'écrire :

$$\mathbf{X}(t + dt) = \mathbf{X}(t) + dt f(\mathbf{X}(t + dt), t + dt)$$

Cette formulation est dite implicite, car le résultat de l'approximation est lui-même utilisé lors de l'approximation [PTVF92]. Ce n'est pas forcément un problème si la fonction  $f$  est connue; malheureusement, en animation, la connaissance de  $f$  demande souvent une évaluation des forces en fonction des positions.

### C.1.4 Méthode de Newton-Cotes

L'inconvénient des méthodes plus évoluées que la technique d'Euler pour la synthèse de mouvement est que chaque pas d'animation nécessite plusieurs évaluations des forces du système: un retour en arrière dans le temps (pour traiter correctement une collision par exemple) n'est alors souvent pas facile à gérer efficacement. Une solution intermédiaire est bien souvent préférée. Dans le cas fréquent de l'intégration des équations fondamentales de la dynamique, à savoir :

$$\mathbf{a} = \mathbf{F}/m$$

$$\dot{\mathbf{v}} = \mathbf{a}$$

$$\dot{\mathbf{X}} = \mathbf{v}$$

une solution, dite de Newton-Cotes, est d'écrire à chaque pas de temps  $dt$  :

$$\begin{aligned} \mathbf{a}_t &= \frac{\mathbf{F}}{m} \\ \mathbf{v}_t &= \mathbf{v}_{t-dt} + \mathbf{a}_t dt \\ \mathbf{X}_t &= \mathbf{X}_{t-dt} + \frac{1}{2}(\mathbf{v}_t + \mathbf{v}_{t-dt}) dt \end{aligned} \tag{C.3}$$

Ce choix d'intégration est très simple à mettre en œuvre, et a la bon goût d'aboutir à une résolution *exacte* dans le cas de chute libre (accélération constante). Ce schéma est donc très souvent utilisé en pratique.

## C.2 Équations aux dérivées partielles

Les équations aux dérivées partielles sont au cœur de maints problèmes de simulation, en hydrodynamique par exemple. Cette fois, ces équations font intervenir des fonctions dépendant de plusieurs paramètres : typiquement, le temps et la position. On classe généralement ces équations en trois catégories : hyperbolique, parabolique, ou elliptique. Cette dernière catégorie, dont le prototype est l'équation de Poisson, ne sera pas abordée ici puisque nous ne l'avons pas utilisée dans ce document. Par contre, les deux premières catégories méritent de s'y arrêter, car un bon nombre de notions fréquemment utilisées en animation découlent de leur analyse.

### C.2.1 Cas hyperbolique - Équation de propagation

L'équation type de la catégorie hyperbolique est l'équation de *propagation* d'onde suivante :

$$\frac{\partial^2 u}{\partial t^2} = -v^2 \frac{\partial^2 u}{\partial x^2}.$$

Elle peut se ramener en fait à un système couplé de deux équations du premier ordre qui se formulent ainsi :

$$\frac{\partial u}{\partial t} = -v \frac{\partial u}{\partial x}. \quad (\text{C.4})$$

Même si cette équation a une solution analytique connue, elle est un bon exemple pour tester les schémas d'intégration puisque l'on pourra comparer les résultats à la vraie solution analytique.

#### Première résolution

L'approche numérique la plus usitée pour la résolution d'une équation de ce type est l'utilisation des différences finies. On choisit pour cela une grille régulière en  $x$  et en  $t$ . Par simplicité, nous écrivons :

$$\begin{aligned} x_j &= x_0 + j\Delta x, \\ t_n &= t_0 + n\Delta t, \\ \text{et } u_j^n &= u(x_j, t_n). \end{aligned}$$

Une façon simple d'exprimer les dérivées partielles, tout à fait analogue à la méthode d'Euler, s'écrit :

$$\frac{\partial u}{\partial t} \Big|_{j,n} = \frac{u_j^{n+1} - u_j^n}{\Delta t} \quad (\text{C.5})$$

$$\frac{\partial u}{\partial x} \Big|_{j,n} = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \quad (\text{C.6})$$

L'équation C.4 s'approxime alors aisément :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x},$$

d'où l'on déduit le schéma d'intégration suivant :

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{2\Delta x} (u_{j+1}^n - u_{j-1}^n) \quad (\text{C.7})$$

Ce schéma est donc très simple à mettre en œuvre. Malheureusement, à l'instar du schéma d'Euler, cette solution est peu précise et assez instable.

### Analyse de la stabilité

Une solution simpliste (mais efficace) pour analyser la stabilité d'un schéma d'intégration est la méthode dite de von Neumann. Elle consiste à considérer les coefficients des équations différentielles quasi constants en temps et en espace : dans ce cas, les solutions indépendantes (*modes propres*) des équations différentielles sont toutes de la forme :

$$u_j^n = \xi(k)^n e^{ikj\Delta x} \quad (\text{C.8})$$

où  $k$  est le nombre d'onde, proportionnel à l'inverse de la longueur d'onde. Puisqu'une solution quelconque sera une combinaison linéaire de ces modes propres, les équations différentielles seront instables (car s'amplifiant exponentiellement) s'il existe un  $k$  pour lequel  $|\xi(k)|$  est strictement supérieur à 1 :  $\xi(k)$  est en effet le coefficient d'amplification de l'onde  $k$  dans le temps.

Appliqué au schéma de l'équation C.7, on obtient en y substituant C.8 et en divisant par  $\xi(k)^n$  :

$$\xi(k) = 1 - i \frac{v\Delta t}{\Delta x} \sin(k\Delta x)$$

dont le module est toujours supérieur à 1 quelque soit  $k$ . Le schéma est donc *inconditionnellement instable*.

### Méthode de Lax et condition de Courant

Pour accroître la stabilité sans pour autant allourdir le schéma, on peut avoir l'idée de substituer le terme en  $u_j^n$  par  $(u_{j+1}^n + u_{j-1}^n)/2$ , rendant l'équation plus symétrique. Le schéma, dit de Lax, devient alors :

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{v\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) \quad (\text{C.9})$$

Le coefficient d'amplification s'écrit maintenant :  $\xi(k) = \cos(k\Delta x) - i \frac{v\Delta t}{\Delta x} \sin(k\Delta x)$ . Le critère de stabilité stipulant  $|\xi(k)| < 1$  nécessite alors d'avoir :

$$\frac{|v|\Delta t}{\Delta x} < 1 \quad (\text{C.10})$$

On trouve ainsi le classique critère de stabilité de Courant-Friedrichs-Lewy, souvent appelé *condition de Courant*. Intuitivement, cela se comprend très bien si l'on se remémore que l'équation C.4 de départ correspond à une propagation d'onde à une vitesse  $v$  : le critère signifie donc que le pas de temps doit être suffisamment petit pour bien propager les ondes de cette vitesse à travers le maillage régulier. Ce n'est pas sans rappeler le théorème de Shannon en théorie du signal.

### Viscosité

À ce stade, il est intéressant de chercher à comprendre pourquoi le simple changement effectué par la méthode Lax garantit la stabilité. Si on ré-écrit le schéma sous la forme :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + \frac{1}{2} \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta t},$$

on s'aperçoit que cela représente exactement l'approximation d'Euler de l'équation :

$$\frac{\partial u}{\partial t} = -v \frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{2\Delta t} \frac{\partial^2 u}{\partial x^2}. \quad (\text{C.11})$$

Le terme supplémentaire est analogue à une diffusion : il entraîne un effet dissipatif, appelé *viscosité*. D'ailleurs, on le constate en regardant le coefficient d'amplification : il est inférieur à 1, ce qui signifie que l'amplitude des ondes décroît rapidement avec le temps.

Cette viscosité est d'autant plus forte pour les fréquences élevées comme l'indique la formule C.11. Cela signifie que ce sont surtout les hautes fréquences qui sont mal intégrées par ce schéma. Mais sans viscosité, c'est-à-dire dans le schéma simple du début, ces fréquences s'amplifieraient tellement qu'elles viendraient perturber les fréquences basses, les plus visibles. Mieux vaut donc atténuer fortement les hautes fréquences si cela permet d'avoir une bonne approximation des autres fréquences.

### Autres méthodes classiques

La stabilité d'intégration peut aussi profiter d'informations quant à la nature des perturbations. Dans le schéma de Lax par exemple, une perturbation au nœud  $j$  va se propager aux nœud  $j - 1$  et  $j + 1$  au pas de temps suivant. Mais si l'on sait que, au moins localement, la perturbation va dans le sens des  $x$  croissants, seul le nœud  $j + 1$  devrait être modifié : c'est l'idée directrice de la méthode dite *upwind*, que l'on pourrait traduire par "dans le sens du vent".

Un autre schéma classique améliore l'approximation du schéma numérique en utilisant les informations de temps aux deux pas précédents : cette méthode, dite "staggered leapfrog", rappelle dans le principe Runge-Kutta, puisque l'on prend en compte une approximation à "mi-temps" pour en déduire une meilleure approximation au pas de temps suivant.

### C.2.2 Cas parabolique - Équation de diffusion

Le prototype du cas parabolique est l'équation de diffusion, largement rencontrée en physique :

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}. \quad (\text{C.12})$$

#### Schéma simple

Cette fois, une approximation directe de cette équation nous donne :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}.$$

Ce type d'approximation était inconditionnellement instable dans le cas hyperbolique; ici, une analyse de von Neumann nous montre que ce schéma est conditionnellement stable. En effet, le coefficient d'amplification s'écrit :

$$\xi(k) = 1 - \frac{4D\Delta t}{(\Delta x)^2} \sin^2\left(\frac{k\Delta x}{2}\right),$$

ce qui entraîne qu'un schéma stable nécessite le critère :

$$\frac{2D\Delta t}{(\Delta x)^2} < 1. \quad (\text{C.13})$$

À l'instar du critère de Courant pour les équations paraboliques, ce critère de stabilité est assez général pour les équations paraboliques.

### **Autres méthodes**

Comme pour les équations différentielles ordinaires, l'utilisation de schéma implicite améliore la précision et la stabilité de l'intégration. Il existe même un certain nombre de schémas inconditionnellement stables, comme celui de Crank-Nicholson [PTVF92].





## Annexe D

# Formulation des courbures d'une surface implicite

Cette annexe aborde le problème du calcul de la géométrie du second ordre d'une surface implicite, à savoir ses courbures. Pour une surface implicite définie par un potentiel  $f$ , on sait que la normale  $\mathbf{n}(P)$  à la surface est selon le gradient  $\nabla f$  du potentiel. Par contre, trouver en un point  $P$  quelconque les deux courbures principales n'est pas aussi simple. Examinons quelles relations permettent de trouver ces informations.

### D.1 Différentielles du second ordre et courbures

La différentielle d'ordre deux du potentiel  $f$  est la matrice Hessienne :

$$D^2 f = H = \begin{bmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{yx} & f_{yy} & f_{yz} \\ f_{zx} & f_{zy} & f_{zz} \end{bmatrix}$$

où  $f_{xx} = \frac{\partial^2 f}{\partial x^2}$ ,  $f_{xy} = \frac{\partial^2 f}{\partial x \partial y}$ , et ainsi de suite.

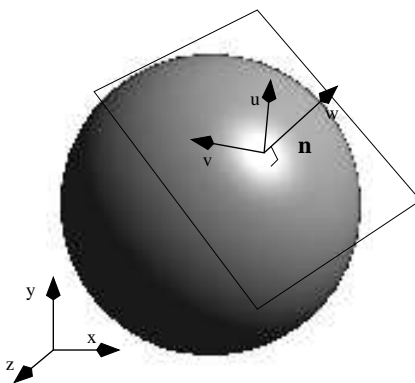


FIG. D.1 - Le repère de Frénet  $(u, v, w)$  et le plan tangent en un point  $P$

En tout point  $P$  régulier de la surface implicite, c'est-à-dire où le gradient de  $f$  n'est pas nul, un repère local digne d'intérêt est le repère dit de Frénet, représenté sur la figure D.1. Si l'on

exprime la matrice Hessienne dans ce repère, où l'axe  $z$  est devenu la normale à la surface, et où les deux autres axes forment une base du plan tangent à la surface en  $P$ , avec  $R$  la matrice de rotation qui transforme le repère cartésien  $(P, \vec{x}, \vec{y}, \vec{z})$  en le repère de Frénet  $(P, \vec{u}, \vec{v}, \vec{w})$ , on obtient :

$$H = RW = R \begin{bmatrix} f_{uu} & f_{uv} & f_{uw} \\ f_{vu} & f_{vv} & f_{vw} \\ f_{wu} & f_{wv} & f_{ww} \end{bmatrix}$$

Puisque  $R$  est une rotation, on a bien sûr égalité entre les déterminants de  $H$  et  $W$ . Il en va de même pour leur trace et leur norme euclidienne. Parmi les termes de  $W$ , seul le terme  $f_{ww}$  peut s'exprimer en fonction de  $H$  et  $\nabla f$ . En effet,  $f_{ww} = (001)W(001)^t$ , ce qui s'exprime dans le repère cartésien par  $\mathbf{nHn}$ . On a donc la relation :

$$f_{ww} = \frac{1}{|\nabla f|^2} \nabla f H \nabla f.$$

La sous-matrice supérieure gauche caractérise quant à elle les informations du second-ordre de  $f$  restreintes au plan tangent :

$$W_{|2} = \begin{bmatrix} f_{uu} & f_{uv} \\ f_{vu} & f_{vv} \end{bmatrix}$$

Les vecteurs propres de cette matrice sont en effet les axes des courbures principales, et les valeurs propres  $\omega_1$  et  $\omega_2$  sont justement ces courbures principales à un facteur  $|\nabla f|$  près. Si  $P$  est la matrice de passage dans cette base propre, on a donc :

$$PW_{|2} = |\nabla f| \begin{bmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{bmatrix}$$

## D.2 Calcul direct des courbures moyenne et gaussienne

Cependant, la courbure moyenne  $\kappa_M = (\omega_1 + \omega_2)/2$  et la courbure gaussienne  $\kappa_G = \omega_1\omega_2$  peuvent être calculées sans exprimer la matrice  $W_{|2}$ . On a effectivement vu que :

$$\text{trace}(W_{|2}) = |\nabla f| (\omega_1 + \omega_2) = 2|\nabla f| \kappa_M$$

$$\det(W_{|2}) = |\nabla f|^2 \omega_1 \omega_2 = |\nabla f|^2 \kappa_G$$

Les relations vues au paragraphe précédent permettent d'écrire :

$$\begin{aligned} \text{trace}(W_{|2}) &= f_{uu} + f_{vv} \\ &= f_{uu} + f_{vv} + (f_{uw} - f_{wu}) \\ &= \text{trace}(W) - f_{ww} \\ &= \text{trace}(H) - f_{ww} \\ &= f_{xx} + f_{yy} + f_{zz} - f_{ww}, \end{aligned}$$

ce qui permet d'écrire la courbure moyenne directement. De façon similaire, la courbure gaussienne peut s'exprimer par :

$$\begin{aligned} \det(W_{|2}) &= |\nabla f|^2 \omega_1 \omega_2 \\ &= \frac{1}{2} |\nabla f|^2 ((\omega_1 + \omega_2)^2 - \omega_1^2 - \omega_2^2) \\ &= \frac{1}{2} |\nabla f|^2 (\text{trace}(W_{|2})^2 - \|W_{|2}\|^2) \end{aligned}$$

La norme de  $W|_2$  peut aussi s'écrire directement en fonction de  $H$  et de  $\nabla f$  :

$$\begin{aligned}
 \|W|_2\|^2 &= \|W\|^2 - 2f_{uw}^2 - 2f_{vw}^2 - f_{ww}^2 \\
 &= \|W\|^2 - 2(f_{uw}^2 + f_{vw}^2 + f_{ww}^2) + f_{ww}^2 \\
 &= \|H\|^2 - 2\|\nabla f_w\|^2 + f_{ww}^2 \\
 &= \|H\|^2 - 2\|H \nabla f\|^2 + f_{ww}^2
 \end{aligned}$$

Une fois les deux courbures trouvées, on peut même en déduire la valeur respective de  $\omega_1$  et  $\omega_2$ . À noter aussi que la courbure dite moyennée, définie par  $\kappa_w = (\omega_1 + \omega_2)\omega_1\omega_2/(\omega_1^2 + \omega_2^2)$ , peut aussi s'écrire directement :

$$\kappa_w = \frac{1}{2}(f_{xx} + f_{yy} + f_{zz} - f_{ww}) \left( \frac{(f_{xx} + f_{yy} + f_{zz} - f_{ww})^2}{\|H\|^2 - 2\|H \nabla f\|^2 + f_{ww}^2} - 1 \right).$$



# Table des figures

2.1	Modèle de [TW88]	22
2.2	Un exemple simple de réseau Masses-Ressorts	24
2.3	L'élément hexaédrique à la base du modèle de serpent.	25
2.4	Construction de l'enrobage d'un squelette, d'après [GVP91]	28
2.5	Un potentiel de Lennard-Jones, et la force dérivée.	31
2.6	Le modèle de Jimenez (d'après [Jim93])	31
2.7	Les forces d'interaction en fonction de la distance pour le modèle de Lombardo (a), et de Reynolds (b)	32
2.8	Une approximation de l'eau en colonnes d'eau ne convient pas toujours	36
2.9	Détection de collision sur facette fixe	38
2.10	Problème de détection sur des splines	39
2.11	Adjonction d'un ressort fictif	39
2.12	Méthode du rebond de Miller	40
3.1	Exemples de super ellipsoïdes	44
3.2	Une fonction potentiel et la surface engendrée par deux points clés.	44
3.3	Exemples de surfaces implicites	46
3.4	Construction géométrique pour un enbranchement lisse	46
3.5	Un squelette, la forme bosselée qu'il engendre, et sa version convoluée.	47
3.6	Une colonne constituée de deux segments, deux disques, et deux parallépipèdes. Le potentiel utilisé pour les segments est composé d'une déformation de type vissage (d'après [RCG+97]).	48
3.7	Le désormais fameux train de Wyvill : versions 1985 et 1995.	48
3.8	Trois cas de polygonalisation d'un voxel selon les valeurs aux sommets.	49
3.9	Décomposition du cube en six tétraèdres	50
3.10	Évolution des particules et discrétisation finale (d'après [WH94])	51
3.11	Le problème du mélange indésirable	51
3.12	Un exemple de rendu de qualité d'objets implicites : <i>Simply Implicit</i> [FGGL93]	53
3.13	(a) Élasticité linéaire - (b) Élasticité non linéaire	55
3.14	Objet avant et après traitement	58
4.1	Une loi typique de contrainte/déplacement, et ses seuils.	65
4.2	Zone linéaire de la loi de Lennard-Jones, autour de la position d'équilibre	65
4.3	Schéma des couches et de leurs interactions.	70
5.1	Lois d'interaction pour différents $K$ ( $r_0 = 1$ )	74

5.2	Le frottement visqueux freine une particule dont le mouvement (en pointillé) serait contraire au mouvement local. . . . .	75
5.3	L'enrobage implicite engendre un contour net et lisse autour des particules. . . . .	76
5.4	Deux solutions pour la répartition des forces extérieures. . . . .	78
5.5	Echantillonnage par migration de graines issues d'un point et d'un segment . . . . .	80
5.6	Subdivision récursive d'une facette triangulaire . . . . .	80
5.7	Discrétisation uniforme de la boîte englobante en 2D . . . . .	81
5.8	Les différentes étapes d'initialisation pour une surface implicite engendrée par deux squelettes . . . . .	81
5.9	Migration de graines lors d'une déformation . . . . .	82
5.10	Gros plan sur la zone de mélange après le processus de migration/invalidation (les graines valides sont en noir, les invalides en gris) . . . . .	83
5.11	Comparaison entre écailles et représentation opaque par territoires pour une scène complexe . . . . .	84
5.12	La polygonalisation par primitives (en 2D): initialement (a) puis une fois les potentiels mélangés (b) . . . . .	84
5.13	Quatre versions d'un même objet: par lancer de rayons (a), par écailles (b), puis par la représentation opaque interactive pour deux précisions d'échantillonnage différentes (c-d) . . . . .	85
5.14	Une hiérarchie de boîtes englobantes pavant le volume d'un objet implicite . . . . .	86
5.15	Un objet à trois squelettes, et son arbre d'évaluation (graphe acyclique orienté). . . . .	88
5.16	Personnage défini par arbre d'évaluation, et différentes postures prouvant la bonne gestion des mélange (d'après [OCG97]) . . . . .	88
5.17	Le graphe d'influence et ses composantes connexes. Les squelettes <i>A</i> et <i>B</i> résident dans la même composante: leurs potentiels s'additionneront donc s'ils se rapprochent l'un de l'autre. . . . .	89
5.18	Fusion locale en cas de forte compression: progressivement, les squelettes concernés par la collision mélangent leur potentiel respectif (ossature: 4 particules en haut, 5 en bas). . . . .	90
5.19	Variations de volume lors de la chute d'un matériau hautement déformable composé de 19 particules en ossature. . . . .	91
5.20	Variations de volume lors du mouvement relatif entre deux squelettes, pour le potentiel de [Gas93] et pour un potentiel contraint à garder un volume égal lors d'un écart nul et d'un écart infini. . . . .	92
5.21	(a) Variation de volume. (b) Correction locale du volume. . . . .	92
5.22	Territoires de squelettes et calcul du volume grâce aux graines. . . . .	93
5.23	Deux squelettes en mouvement se mélangent: à gauche sans contrôleur, à droite avec contrôleur. . . . .	94
5.24	Une pince virtuelle saisit une pâte, puis la relâche: pendant ce temps, une séparation, puis une fusion immédiate se produisent au sein de la matière. . . . .	95
5.25	Extraits de <i>Kitchen Fiction</i> [GDGR96]: différentes substances sont manipulées par des ustensiles rigides, également modélisés par surfaces implicites. . . . .	95
5.26	Un personnage simplifié, tombant à terre. . . . .	96
5.27	Un exemple de trajectoire pour une pierre dévalant la colline. . . . .	96
6.1	Le filtre Gaussien approximé par une spline (a), et sa dérivée en 1D (b). . . . .	108
6.2	La fonction de lissage (a) et son gradient en 1D (b) . . . . .	108

6.3	Une force d'interaction typique, appliquée sur une particule $i$ par une particule $j$ , en fonction de la distance entre les deux particules . . . . .	109
6.4	Illustration de l'interprétation (a) Gathering, et (b) Shooting . . . . .	113
6.5	(a) Discrétisation uniforme contre (b) discrétisation adaptée . . . . .	114
6.6	Le raffinement spatial s'effectue en divisant une particule en plusieurs dans les régions où la densité varie beaucoup. . . . .	114
6.7	Une simplification peut se faire quand un groupe de particules stables est proche d'une forme sphérique . . . . .	115
6.8	Comparaison des temps de recherches des voisins selon la méthode utilisée. . . . .	120
6.9	Animation de 1440 particules adaptatées seulement en temps . . . . .	121
6.10	Comparaison entre discrétisation fixe (a, b, c, d) et adaptative (e, f, g, h): deux objets définis par la même équation d'état sont animés. Le gain en temps de calcul, ainsi que le nombre de particules utilisées pour chaque image dans la version adaptative sont indiqués en (i) et (j). . . . .	122
6.11	Une coulée de boue sur un relief montagneux . . . . .	123
6.12	Substance dévalant un relief montagneux ( $t=0.3s, 1.64s, 2.6s$ et $4.1s$ ). . . . .	124
7.1	Exemple de popping: une particule se raffine en six particules. . . . .	127
7.2	Stratégie de la bulle: gonflements et dégonflements locaux . . . . .	131
7.3	Voisinage tubulaire d'une surface implicite définie par $f = 0$ . . . . .	132
7.4	Utilisation du point de la surface le plus proche (d'après [MSV95]) . . . . .	134
7.5	Evaluation de la stratégie en utilisant les points de la surface . . . . .	135
7.6	Exemple de métamorphose entre deux objets de même topologie. . . . .	140
7.7	Exemple de métamorphose entre objets de topologies différentes . . . . .	141
7.8	Exemples d'habillage de particules, pour $\beta = .15$ (a-b-c) et $\beta = .3$ (d-e-f), et une taille de grille de $40 \times 20 \times 20$ (voir figure 6.8). . . . .	142
7.9	Peau implicite appliquée sur l'animation adaptative de la figure 6.12. . . . .	143
7.10	Une image de peau active (a), et ses sommets actifs de la grille régulière (b). . . . .	144
8.1	Généralité des différentes interactions entre masses. . . . .	147
A.1	De haut en bas: un phénomène quelconque, puis une approche Lagrangienne et une approche Eulerienne pour simuler son mouvement . . . . .	154
C.1	Deux pas d'Euler (a), en comparaison à deux pas de Runge-Kutta (b) . . . . .	162
D.1	Le repère de Frénet (u,v,w) et le plan tangent en un point P . . . . .	169





# Bibliographie

- [AH97] Oliver R. ASTLEY et Vincent HAYWARD. « Real-time Finite-elements Simulation of General Visco-elastic Materials for Haptic Presentation ». Dans *IROS'97, workshop on Dynamic Simulation*, p. 52–57, Grenoble, France, septembre 1997.
- [Bar81] Alan H. BARR. « Superquadrics and Angle-Preserving Transforms ». *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.
- [Bar84] Alan H. BARR. « Global and Local Deformations of Solid Primitives ». *Computer Graphics*, 18(3):21–29, 1984.
- [Bar92] David BARAFF. « Dynamic Simulation of Non-penetrating Rigid Bodies ». *PHD Thesis*, Cornell University, mai 1992.
- [Bat73] G.K. BATCHELOR. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1973.
- [Bat82] Klaus-Jürgen BATHE. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, 1982.
- [Bei93] Thaddeus BEIER. « Practical Uses for Implicit Surfaces in Animation ». Dans *Modeling, Visualizing and Animating with Implicit Surfaces (SIGGRAPH'93 course notes Number 25)*, Anaheim, CA, août 1993.
- [Bli82] Jim BLINN. « A generalization of Algebraic Surface Drawing ». *ACM Transactions on Graphics*, p. 235–256, juillet 1982.
- [Blo94] Jules BLOOMENTHAL. « An Implicit Surface Polygonizer ». Dans *Graphic Gems IV*. Paul Heckbert (ed.), Academic Press, 1994.
- [BS91] Jules BLOOMENTHAL et Ken SHOEMAKE. « Convolution Surfaces ». *Computer Graphics*, 25(4):251–256, juillet 1991. Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 1991).
- [BS95] Carole BLANC et Christophe SCHLICK. « Extended Field Functions for Soft Objects ». Dans *Implicit Surfaces'95—the First Eurographics Workshop on Implicit Surfaces*, p. 21–32, Grenoble, France, avril 1995.
- [BTG95] Eric BITTAR, Nicolas TSINGOS, et Marie-Paule GASCUEL. « Automatic Reconstruction of Unstructured 3D data: Combining Medial Axis and Implicit Surfaces ». Dans *Eurographics'95*, septembre 1995.

- [BW76] Nicholas BURTONYK et Mark WEIN. « Interactive Skeleton Technique For Enhancing Motion Dynamics in Key Frame Animation ». *Communications of the ACM*, 19(10):564–569, octobre 1976.
- [BW90] Jules BLOOMENTHAL et Brian WYVILL. « Interactive Techniques for Implicit Modeling ». *Computer Graphics*, 24(2):109–116, mars 1990. Proceedings of Symposium on Interactive 3D Graphics.
- [CG97] Marie-Paule CANI-GASCUEL. « *Introduction to Implicit Surfaces* », Chapitre Implicit Surfaces in Physically-Based Animation, p. 271–293. Edited by Jules Bloomenthal, Morgan Kaufmann Publishers, 1997.
- [CGD97] Marie-Paule CANI-GASCUEL et Mathieu DESBRUN. « Animation of Deformable Models using Implicit Surfaces ». *IEEE Transactions on Visualization and Computer Graphics*, 3(1):39–50, mars 1997.
- [CH97] Deborah A. CARLSON et Jessica K. HODGINS. « Simulation Levels of Detail for Real-time Animation ». Dans *Graphics Interface'97*, p. 1–8, Kelowna, British Columbia, 1997.
- [CHP89] John E. CHADWICK, David R. HAUMANN, et Richard E. PARENT. « Layered Construction for Deformable Animated Characters ». *Computer Graphics*, 23(3):243–252, juillet 1989.
- [Coh91] Laurent D. COHEN. « On active contour models and balloons ». *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, mars 1991.
- [CVM+96] Jonathan COHEN, Amitabh VARSHNEY, Dinesh MANOCHA, Greg TURK, Hans WEBER, Pankaj AGARWAL, Frederick P. BROOKS, JR., et William WRIGHT. « Simplification Envelopes ». Dans Holly RUSHMEIER, éditeur, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, p. 119–128. ACM SIGGRAPH, Addison Wesley, août 1996.
- [DCG97] Mathieu DESBRUN et Marie-Paule CANI-GASCUEL. « Peau Implicite Active pour l'Habillage de Modèles Hautement Déformables ». Dans *Cinquième Journées de l'AFIG*, Rennes, décembre 1997.
- [DG94] Mathieu DESBRUN et Marie-Paule GASCUEL. « Highly Deformable Material for Animation and Collision Processing ». Dans *5th Eurographics Workshop on Animation and Simulation*, p. 89–102, Oslo, Norway, septembre 1994.
- [DG95] Mathieu DESBRUN et Marie-Paule GASCUEL. « Animating Soft Substances with Implicit Surfaces ». Dans *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, p. 287–290. ACM SIGGRAPH, Addison Wesley, août 1995. Los Angeles, CA.
- [DG96] Mathieu DESBRUN et Marie-Paule GASCUEL. « Smoothed Particles: A New Approach for Animating Highly Deformable Bodies ». Dans *7th Eurographics Workshop on Animation and Simulation*, Poitiers, France, septembre 1996.

- [Dil96] Gary A. DILTS. « Equivalence of the SPH Method and a Space-Time Galerkin Moving Particle Method ». Technical Report LA-UR 96-134, Los Alamos National Laboratory, janvier 1996.
- [DTG95] Mathieu DESBRUN, Nicolas TSINGOS, et Marie-Paule GASCUEL. « Adaptive Sampling of Implicit Surfaces for Interactive Modeling and Animation ». Dans *Implicit Surfaces'95—the First Eurographics Workshop on Implicit Surfaces*, p. 171–185, Grenoble, France, avril 1995.
- [DTG96] Mathieu DESBRUN, Nicolas TSINGOS, et Marie-Paule GASCUEL. « Adaptive Sampling of Implicit Surfaces for Interactive Modelling and Animation ». *Computer Graphics Forum*, 15(5):319–327, décembre 1996. A preliminary version of this paper appeared in *Implicit Surfaces'95*, Grenoble, France, may 1995.
- [Fau97] François FAURE. « Deux problèmes physiques pour la Synthèse d'Images ». *Thèse de Doctorat*, Université Joseph Fourier, Grenoble, septembre 1997.
- [FGA96] Eric FERLEY, Marie-Paule GASCUEL, et Dominique ATTALI. « Skeletal Reconstruction of Branching Shapes ». Dans *Implicit Surfaces'96, The Second International Workshop on Implicit Surfaces*, p. 127–142, Eindhoven, The Netherlands, octobre 1996.
- [FGTV92] Luiz Henrique de FIGUEIREDO, Jonas de Miranda GOMEZ, Demetri TERZOPOULOS, et Luiz VELHO. « Physically-Based Methods for Polygonization of Implicit Surfaces ». Dans *Graphics Interface'92*, p. 250–257, Vancouver, Canada, mai 1992.
- [FM96] Nick FOSTER et Dimitri METAXAS. « Realistic Animation of Liquids ». *Graphical Models and Image Processing*, 58(5):471–483, 1996.
- [FM97] Nick FOSTER et Dimitri METAXAS. « Modeling the Motion of a Hot, Turbulent Gas ». *Computer Graphics*, p. 181–188, 1997. Proceedings of SIGGRAPH'97 (Los Angeles, California).
- [FV97] Max FROUMENTIN et Eric VARLET. « Dynamic Implicit Surface Tessellation ». Dans *ACM Symposium on Virtual Reality Software and Technology'97 Proceedings*, Lausanne, Switzerland, 1997.
- [GA96] E. GALIN et S. AKKOUCHE. « Blob Metamorphosis based on Minkowski Sums ». Dans *Eurographics'96*, Poitiers, France, août 1996.
- [Gal97] Eric GALIN. « Métamorphose et visualisation de Blobs à Squelettes. ». *Thèse de Doctorat*, Université Claude Bernard, Lyon, juillet 1997.
- [Gas90] Marie-Paule GASCUEL. « Déformations de surfaces complexes : techniques de haut niveau pour la modélisation et l'animation ». *Thèse de doctorat*, Université Paris XI, octobre 1990.
- [Gas93] Marie-Paule GASCUEL. « An implicit formulation for precise contact modeling between flexible solids. ». *Computer Graphics*, 27:313–320, août 1993. Proceedings of SIGGRAPH'93 (Anaheim, CA).

- [Gas95] Jean-Dominique GASCUEL. « Implicit Patches: An Optimized and Powerful Ray Intersection Algorithm for Implicit Surfaces ». Dans *Implicit Surfaces'95—the First Eurographics Workshop on Implicit Surfaces*, p. 143–159, Grenoble, France, avril 1995.
- [GDGR96] Marie-Paule GASCUEL, Mathieu DESBRUN, Jean-Dominique GASCUEL, et Stéphane RÉHEL. « Kitchen Fiction ». *Eurographics'96*, Video Competition, août 1996.
- [GG94] Jean-Dominique GASCUEL et Marie-Paule GASCUEL. « Displacement constraints for interactive Modeling and Animation of Articulated Structures ». *The Visual Computer*, 10(4):191–204, mars 1994.
- [GMTA<sup>+</sup>96] Thierry GUIARD-MARIGNY, Nicolas TSINGOS, Ali ADJODANI, Christian BENOÎT, et Marie-Paule GASCUEL. « 3D Models of the Lips for Realistic Speech Animation ». Dans *Computer Animation'96*, p. 80–89, Geneva, Switzerland, juin 1996.
- [GMTT89] Jean-Paul GOURRET, Nadia MAGNENAT THALMANN, et Daniel THALMANN. « Simulation of Object and Human Skin Deformations in a Grasping Task. ». *Computer Graphics*, 23(3):21–29, juillet 1989. Proceedings of SIGGRAPH'89 (Boston, MA, July 1989).
- [GTT89] Jean-Paul GOURRET, Nadia Magnenat THALMANN, et Daniel THALMANN. « The Use of Finite Element Theory for Simulating Object and Human Body Deformations and Contacts ». Dans *Eurographics 89*, p. 477–487, septembre 1989.
- [GVP90] Marie-Paule GASCUEL, Anne VERROUST, et Claude PUECH. « Animation with collisions of deformable articulated bodies ». Dans *Eurographics Workshop on Animation and Simulation*, septembre 1990.
- [GVP91] Marie-Paule GASCUEL, Anne VERROUST, et Claude PUECH. « A modeling system for complex deformable bodies suited to animation and collision processing ». *Journal of Visualization and Computer Animation*, 2(3):82–91, août 1991. A shorter version of this paper appeared in *Graphics Interface'91*.
- [GW95] Andrew GUY et Brian WYVILL. « Controlled Blending for Implicit Surfaces using a Graph ». Dans *Implicit Surfaces'95—the First Eurographics Workshop on Implicit Surfaces*, p. 107–112, Grenoble, France, avril 1995.
- [Hab97] Arash HABIBI. « Visualisation d'objets très déformables: relations mouvement - forme - image ». *Thèse de Doctorat*, INPG, janvier 1997.
- [Han88] Andrew HANSON. « Hyperquadrics: Smoothly Deformable Shapes with Convex Polyhedral Bounds ». *Computer Vision, Graphics, and Image Processing*, 44:191–210, 1988.
- [HBG92] Donald HOUSE, David BREEN, et Philipp GETTO. « On the Dynamic Simulation of Physically-Based Particle-System Models ». Dans *Third Eurographics Workshop on Animation and Simulation*, Cambridge, England, septembre 1992.
- [HK92] Lars HERNQUIST et Neal KATZ. « TREESPH: A unification of SPH with the hierarchical tree method ». *App. J. Supp.*, 70:419, 1992.

- [Hop96] Hugues HOPPE. « Progressive Meshes ». Dans Holly RUSHMEIER, éditeur, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, p. 99–108. ACM SIGGRAPH, Addison Wesley, août 1996.
- [HPH96] Dave HUTCHINSON, Martin PRESTON, et Terry HEWITT. « Adaptive Refinement for Mass/Spring Simulation ». Dans *7th Eurographics Workshop on Animation and Simulation*, p. 31–45, Poitiers, France, septembre 1996.
- [HSA91] Pat HANRAHAN, David SALTZMAN, et Larry AUPPERLE. « A Rapid Hierarchical Radiosity Algorithm ». *Computer Graphics*, 25(4):197–206, août 1991. Proceedings SIGGRAPH '91 in Las Vegas (USA).
- [Jim93] Stéphane JIMENEZ. « Modélisation et Simulation Physique d'Objets Volumiques Déformables Complexes ». Thèse de doctorat, Institut National Polytechnique de Grenoble, novembre 1993.
- [Jou97] Ammar JOUKHADAR. « Modélisation dynamique et applications robotiques ». *Thèse de Doctorat*, mai 1997.
- [KB89] Devendra KALRA et Alan H. BARR. « Guaranteed Ray Intersections with Implicit Surfaces ». *Computer Graphics*, 23:297–306, novembre 1989.
- [KM90] Michael KASS et Gavin MILLER. « Rapid, Stable Fluid Dynamics for Computer Graphics ». *Computer Graphics*, 24(4):49–55, 1990.
- [KWT88] Michael KASS, Andrew WITKIN, et Demetri TERZOPOULOS. « Snakes: Active Contour Models ». Dans *1st Conference on Computer Vision*, p. 321,331, London, U.K., juin 1988.
- [Lac96] Jacques-Olivier LACHAUD. « Topologically Defined Isosurfaces ». Dans *Discrete Geometry for COmputer Imagery (Springer Verlag)*, Lyon, France, 1996.
- [Las87] John LASSETER. « Principles of Traditional Animation Applied to 3D Computer Animation ». *Computer Graphics*, 21(4):35–43, juillet 1987. Proceedings of SIGGRAPH'87 (Anaheim, California, July 1987).
- [LB94] Jacques-Olivier LACHAUD et Eric BAINVILLE. « A discrete adaptive model following topological modifications of volumes ». Dans *Discrete Geometry for Computer Imagery*, p. 183–194, septembre 1994.
- [LC86] Annie LUCIANI et Claude CADOZ. « Utilisation de Modèles Mécaniques et Géométriques pour la Synthèse et le Contrôle d'Images Animées ». Dans *Deuxième Colloque Image, CESTA*, Nice, avril 1986.
- [LC87] William LORENSEN et Harvey CLINE. « Marching cubes: a high resolution 3D surface construction algorithm ». *Computer Graphics*, 21(4):163–169, juillet 1987. Proceedings of SIGGRAPH'87 (Anaheim, California, July 1987).
- [LHVD95] Annie LUCIANI, Arash HABIBI, Alexis VAPILLON, et Yves DUROC. « A physical Model Of Turbulent Fluids ». Dans *6th Eurographics Workshop on Animation and Simulation*, Maastricht, Netherlands, septembre 1995.

- [LJF<sup>+</sup>91] Annie LUCIANI, Stéphane JIMENEZ, Jean-Loup FLORENS, Claude CADOZ, et Olivier RAOULT. « Computational physics: a modeler simulator for animated physical objects ». Dans *Eurographics'91*, Vienna, Austria, septembre 1991.
- [LJR<sup>+</sup>91] Annie LUCIANI, Stéphane JIMENEZ, Olivier RAOULT, Claude CADOZ, et Jean-Loup FLORENS. « A unified View of Multitude Behaviour, flexibility, plasticity, and fractures: balls, bubbles and agglomerates ». Dans *IFIP WG 5.10 Working Conference*, Tokyo, Japan, avril 1991.
- [LM96] Jacques-Olivier LACHAUD et Annick MONTANVERT. « Volumic Segmentation using Hierarchical Representation and Triangulated Surface ». Dans *European Conference on Computer Vision*, p. 137–146, Cambridge, U.K., avril 1996.
- [LP95] Jean-Christophe LOMBARDO et Claude PUECH. « Oriented Particles: A Tool for Shape Memory Objects Modelling ». Dans *Graphics Interface'95*, Quebec, Canada, mai 1995.
- [Luc85] Annie LUCIANI. « Un Outil Informatique de Création d'Images Animées ». *Thèse de docteur ingénieur d'électronique*, Institut National Polytechnique de Grenoble, novembre 1985.
- [Luc89] Annie LUCIANI. « Modèles physiques pour l'animation ». *Rapport de Recherche ACROE*, 89(14), novembre 1989.
- [Mes97] Philippe MESEURE. « Modélisation de corps déformables pour la simulation d'actes chirurgicaux ». *Thèse de Doctorat*, Université des Sciences et Technologies de Lille, janvier 1997.
- [Mil88] Gavin MILLER. « The Motion Dynamics of Snakes and Worms. ». *Computer Graphics*, 22(4):169–177, août 1988. Proceedings of SIGGRAPH'88 (Atlanta, Georgia).
- [Mon92] J. J. MONAGHAN. « Smoothed Particle Hydrodynamics ». *Annu. Rev. Astron. Astrophys.*, 30:543, 1992.
- [MP89] Gavin MILLER et Andrew PEARCE. « Globular Dynamics: A connected particle system for animating viscous fluids ». *Computers and Graphics*, 13(3):305–309, 89. This paper also appeared in SIGGRAPH'89 Course notes number 30.
- [MSV95] Ravikanth MALLADI, James A. SETHIAN, et Baba C. VEMURI. « Shape Modeling with Front Propagation: A Level Set Approach ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, février 1995.
- [MT92] Dimitri METAXAS et Demetri TERZOPOULOS. « Dynamic Deformation of Solid Primitives with Constraints ». *Computer Graphics*, 26(2):309–312, juillet 1992. Proceedings of SIGGRAPH'92 (Chicago, Illinois, July 1992).
- [Mur91] Shigeru MURAKI. « Volumetric Shape Description of Range Data using Blobby Model ». *Computer Graphics*, 25(4):227–235, juillet 1991.
- [MW88] Matthew MOORE et Jane WILHELMS. « Collision Detection and Response for Computer Animation ». *Computer Graphics*, 22(4):289–298, août 1988. Proceedings of SIGGRAPH'88 (Atlanta, August 1988).

- [NHK<sup>+</sup>85] H. NISHIMURA, M. HIRAI, T. KAWAI, T. KAWATA, I. SHIRAKAWA, et K. OMURA. « Objects Modeling by Distribution Function and a Method of Image Generation (in japanese) ». *The Transactions of the Institute of Electronics and Communication Engineers of Japan*, J68-D(4):718–725, 1985.
- [OCG97] Agata OPALACH et Marie-Paule CANI-GASCUEL. « Local Deformation for Animation of Implicit Surfaces ». Dans *SCCG'97, Bratislava, Slovakia*, 1997.
- [OM93] Agata OPALACH et Steve MADDOCK. « Implicit Surfaces: Appearance, Blending and Consistency ». Dans *Fourth Eurographics Workshop on Animation and Simulation*, p. 233–245, Barcelona, Spain, septembre 1993.
- [PASS95] A. PASKO, V. ADZHIEV, A. SOURIN, et V. SAVCHENKO. « Function Representation in Geometric Modeling: Concepts, Implementation and Applications ». *The Visual Computer*, 11(8):429–446, 1995.
- [PB88] John PLATT et Alan BARR. « Constraint Methods for Flexible Models ». *Computer Graphics*, 22(4):279–288, août 1988. Proceedings of SIGGRAPH'88 (Atlanta, Georgia).
- [Ped95] Hans K. PEDERSEN. « Decorating Implicit Surfaces ». *Computer Graphics*, p. 291–300, août 1995. Proceedings of SIGGRAPH'95 (Los Angeles, California, August 1995).
- [Ped96] Hans K. PEDERSEN. « A Framework for Interactive Texturing on Curved Surfaces ». *Computer Graphics*, p. 295–302, août 1996. Proceedings of SIGGRAPH'96 (New Orleans, Louisiana, August 1996).
- [Pro95] Xavier PROVOT. « Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior ». Dans *Graphics Interface'95*, p. 147–154, Quebec, Canada, mai 1995.
- [Pro97] Emmanuel PROMAYON. « Modélisation et Simulation de la Respiration ». *Thèse de Doctorat*, novembre 1997.
- [PTVF92] William PRESS, Saul TEUKOLSKY, William VETTERLING, et Brian FLANNERY. *Numerical Recipes in C, second edition*. Cambridge University Press, New York, USA, 1992.
- [PW89] Alex PENTLAND et John WILLIAMS. « Good Vibrations: Modal Dynamics for Graphics and Animation ». *Computer Graphics*, 23(3):215–222, juillet 1989. Proceedings of SIGGRAPH'89 (Boston, MA, July 1989).
- [RCGG<sup>+</sup>97] Dominique ROSSIN, Marie-Paule CANI-GASCUEL, Jean-Dominique GASCUEL, Agata OPALACH, et Mathieu DESBRUN. « Plate-forme d'expérimentation pour la modélisation par surfaces implicites ». Dans *Journées Modeleurs Géométriques*, septembre 1997.
- [Ree83] W. T. REEVES. « Particle systems—a technique for modeling a class of fuzzy objects ». *Computer Graphics*, 17(3):359–376, 1983.



- [Rey97] Hugh REYNOLDS. « An Alternative Inter-Particle Force Model for Coupled System Flexible Body Dynamics ». Dans *8th Eurographics Workshop on Animation and Simulation*, Buydapest, Hungary, septembre 1997.
- [Rey87] Craig W. REYNOLDS. « Flocks, Herds and Schools: A Distributed behavioral Model ». *Computer Graphics*, 21(4):25–34, juillet 87.
- [RRS96] Angela R OSCH, Matthias RUHL, et Dietmar SAUPE. « Interactive Visualization of Implicit Surfaces with Singularities ». Dans *Implicit Surfaces'96—Eurographics and ACM-Siggraph Workshop*, p. 73–87, Eindhoven, the Netherlands, octobre 1996.
- [Set96] James A. SETHIAN. *Level Set Methods*. Cambridge Press, 1996.
- [SF93] Jos STAM et Eugene FIUME. « Turbulent Wind Fields for Gaseous Phenomena ». Dans James T. KAJIYA, éditeur, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, p. 369–376, août 1993.
- [SF95] Jos STAM et Eugene FIUME. « Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes ». Dans Robert COOK, éditeur, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, p. 129–136. ACM SIGGRAPH, Addison Wesley, août 1995. held in Los Angeles, California, 06-11 August 1995.
- [SH97] Barton T. STANDER et John C. HART. « Guaranteeing the Topology of an Implicit Surface Polygonalisation ». *Computer Graphics*, p. 279–286, août 1997. Proceedings of SIGGRAPH'97i (Los Angeles, California, August 1997).
- [Sil95] François SILLION. « A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters ». *IEEE Transactions on Visualization and Computer Graphics*, 1(3), septembre 1995. (a preliminary version appeared in the fifth Eurographics workshop on rendering, Darmstadt, Germany, June 1994).
- [SP86] Thomas W. SEDEBERG et Scott R. PARRY. « Free-form Deformations of Solid Geometric Models ». *Computer Graphics*, 20(4):151–160, 1986.
- [SP91] Stan SCLAROFF et Alex PENTLAND. « Generalized Implicit Functions For Computer Graphics. ». *Computer Graphics*, 25(4):247–250, juillet 1991. Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 1991).
- [SP94] François X. SILLION et Claude PUECH. *Radiosity and Global Illumination*. Morgan Kaufmann Publishers, Inc., 1994.
- [SS96] Jean-Paul SMETS-SOLANES. « Vector Field Based Texture Mapping of Animated Implicit Objects ». Dans *Eurographics'96*, p. 289–300, Poitiers, France, août 1996.
- [ST92] Richard SZELISKI et David TONNESEN. « Surface Modeling with Oriented Particle Systems ». *Computer Graphics*, 26(2):185–194, juillet 1992. Proceedings of SIGGRAPH'92 (Chicago, Illinois, July 1992).
- [ST95] Jianhua SHEN et Daniel THALMANN. « Interactive shape design using metaballs and splines ». Dans *Implicit Surfaces'95—the First Eurographics Workshop on Implicit Surfaces*, p. 187–195, Grenoble, France, avril 1995.

- [TBG95] Nicolas TSINGOS, Eric BITTAR, et Marie-Paule GASCUEL. « *Computer Graphics: Developments in virtual environments* », Chapitre Semi-automatic Reconstruction of Implicit Surfaces for Medical Applications. Earnshaw and Vince. Academic Press., june 1995. Proceedings of Computer Graphics International'95, Leeds, UK.
- [TF88a] Demetri TERZOPOULOS et Kurt FLEISCHER. « Deformable Models ». *The Visual Computer*, 1988(4):306–331, 1988.
- [TF88b] Demetri TERZOPOULOS et Kurt FLEISCHER. « Modeling Inelastic Deformations: Viscoelasticity, Plasticity, Fracture. ». *Computer Graphics*, 22(4):269–278, août 1988. Proceedings of SIGGRAPH'88 (Atlanta, Georgia).
- [TM91] Demetri TERZOPOULOS et Dimitri METAXAS. « Dynamic 3-D models with local and global deformations: deformable superquadrics ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, July 1991.
- [Ton91] David TONNESEN. « Modeling Liquids and Solids using Thermal Particles ». Dans *Graphics Interface '91*, p. 255–262, Calgary, AL, juin 1991.
- [TPBF87] Demetri TERZOPOULOS, John PLATT, Alan BARR, et Kurt FLEISCHER. « Elastically Deformable Models. ». *Computer Graphics*, 21(4):205–214, juillet 1987. Proceedings of SIGGRAPH'87 (Anaheim, California).
- [TPF89] Demetri TERZOPOULOS, John PLATT, et Kurt FLEISHER. « Heating and Melting Deformable Models (From Goop to Glop) ». Dans *Graphics Interface '89*, p. 219–226, London, Ontario, juin 1989.
- [Tur95] Russel TURNER. « LEMAN: A system for Construsting and Animating Layered Elastic Characters ». Dans *Computer Graphics- Developments in Virtual Environments*, p. 185–203, Academic Press, San Diego, CA, juin 1995.
- [TW88] Demetri TERZOPOULOS et Andrew WITKIN. « Physically Based Model with Rigid and Deformable Components ». *IEEE Computer Graphics and Applications*, p. 41–51, décembre 1988.
- [VCT95] Pascal VOLINO, Martin COURCHESNE, et Nadia Magnenat THALMANN. « Versatile and Efficient Techniques for imulating Cloth and Other Deformable Objects ». *Computer Graphics*, p. 137–144, août 1995.
- [Ver90] Anne VERROUST. « Etude de problèmes liés à la définition, la visualisation et l'animation d'objets complexes en informatique graphique ». *Thèse d'état*, Université Paris XI, décembre 1990.
- [VG96] Luiz VELHO et Jonas GOMEZ. « Approximate conversion of parametric to implicit surfaces ». *Computer Graphics Forum*, 15(5):327–337, décembre 1996. A preliminary version of this paper appeared in *Implicit Surfaces'95*, Grenoble, France, may 1995.
- [VGW90] Allen VAN GELDER et Jane WILHELMS. « Topological Considerations in Isosurface Generation ». *Computer Graphics*, 24(5):79–86, novembre 1990.

- [VTdMG94] Luiz VELHO, Demetri TERZOPOULOS, et Jonas de MIRANDA GOMEZ. « Multiscale Implicit Models ». Dans *Sibgrapi'94*, Curitiba, Brazil, Nov 1994.
- [WH94] Andrew WITKIN et Paul HECKBERT. « Using Particles to Sample and Control Implicit Surfaces ». *Computer Graphics*, p. 269–278, juillet 1994. Proceedings of SIGGRAPH'94.
- [Whi95] Ross WHITAKER. « Algorithms for implicit deformable models ». Dans *The International Conference of Computer Vision*, Boston, Mass, 1995.
- [WMW86] Geoff WYVILL, Craig MCPHEETERS, et Brian WYVILL. « Data Structure for Soft Objects ». *The Visual Computer*, 2(4):227–234, août 1986.
- [WTT95] Yi WU, Daniel THALMANN, et Nadia Magnenat THALMANN. « *Computer Graphics: Developments in virtual environments* », Chapitre Deformable Surfaces using Physically Based Particle Systems, p. 205–215. Earnshaw and Vince. Academic Press., jun 1995. Proceedings of Computer Graphics International'95, Leeds, UK.
- [WW89] Brian WYVILL et Geoff WYVILL. « Field Functions for Implicit Surfaces ». *The Visual Computer*, 5:75–82, décembre 1989.
- [WW90] Andrew WITKIN et William WELCH. « Fast Animation and Control for Non-rigid Structures. ». *Computer Graphics*, 24(4):243–252, août 1990. Proceedings of SIGGRAPH'90 (Dallas, Texas, August 1990).
- [Wyv97] Brian WYVILL. « *Introduction to Implicit Surfaces* », Chapitre Animation and Special Effects, p. 243–269. Edited by Jules Bloomenthal, Morgan Kaufmann Publishers, 1997.

## Résumé

Cette thèse porte sur la modélisation et l’animation de matériaux hautement déformables en Synthèse d’Images. Le but est de créer des modèles virtuels de matière, à base de physique, capables de se déformer et d’interagir avec leur environnement de façon automatique. La difficulté principale est d’assurer l’efficacité des calculs et la qualité visuelle résultante même dans les cas de déformations extrêmes, pour alors permettre d’utiliser ces modèles dans le cadre de simulateurs, et plus généralement, en réalité virtuelle.

Après avoir passé en revue les modèles existants, nous proposons un premier *modèle hybride* de matériaux hautement déformables combinant système de particules et surfaces implicites. Les avantages de ces deux techniques sont cumulés pour aboutir à un modèle global capable aussi bien de séparations que de fusions, en assurant aussi d’autres propriétés comme la conservation de volume. Nous discutons dans un second temps de l’inadéquation des systèmes de particules classiques. Une alternative est alors présentée, sous forme d’un nouveau modèle permettant une *simulation adaptative en temps et en espace*, où les particules peuvent se subdiviser dans les régions subissant de fortes déformations, et au contraire se regrouper dans les régions stables. L’adaptation de la discrétisation de la matière permet ainsi de répartir les calculs de façon *efficace*, tout en assurant la stabilité de la simulation. Enfin, un modèle de *peau implicite active* est exposé. Ce modèle permet d’animer une surface déformable chargée d’enrober un modèle physique quelconque pour à la fois lui fournir une visualisation et des propriétés physiques comme une tension de surface : on obtient ainsi une solution efficace et peu onéreuse à la visualisation du modèle particulaire adaptatif précédent, en filtrant les changements internes de discrétisation.

Ainsi, ces nouveaux modèles ouvrent la voie à la simulation adaptative d’objets déformables, qui permet de minimiser les calculs pour une précision donnée.

**Mots clef :** Images de synthèse, animation et simulation d’objets déformables, système de particules, surfaces implicites, simulation adaptative.

## Abstract

In this thesis, we are dealing with computer graphics modeling and animation of highly deformable materials. Our work is aimed at creating virtual physically-based models of matter, able to automatically deform due to interactions with the environment. Coping with large deformation is known to be time consuming, so efficiency and visual quality are the key issues as they can open avenues such as surgery simulation or virtual reality in general.

After a review of previous related work, we present a *hybrid model* for highly deformable materials that combines implicit surfaces and a particle system. It results in a global model that gathers the advantages of the two approaches, and other valuable properties as volume preservation. Then we discuss conventional particle system weaknesses. An alternative model is thus proposed allowing a *space-time adaptive simulation*, where particles can subdivide to better discretize fast deforming areas, or merge to simplify stable regions. Computations are therefore stable and optimized as discretization is automatically adapted. Lastly, an *active implicit skin* model is introduced. This deformable surface can coat any deformable model both providing a neat visualization and conferring physical properties such as surface tension. More generally, it offers an efficient yet low-cost technique to visualize adaptive models, avoiding “popping” effects through smoothing of sudden internal change of granularity.

The new models we developed seem to promise a fully adaptive simulation of deformable objects as computations are minimized to ensure a given accuracy and focussed where needed.

**Keywords:** Computer graphics, animation synthesis, deformable objects simulation, particle system, implicit surfaces, adaptive simulation.