



**HAL**  
open science

# Modélisation et visualisation de phénomènes naturels simulés pas système physique particulière

Claire Guilbaud

► **To cite this version:**

Claire Guilbaud. Modélisation et visualisation de phénomènes naturels simulés pas système physique particulière. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 2002. Français. NNT: . tel-00004458

**HAL Id: tel-00004458**

**<https://theses.hal.science/tel-00004458>**

Submitted on 3 Feb 2004

**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.

# ***INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE***

*N° attribué par la bibliothèque*

□□□□□□□□□□

## **THESE**

pour obtenir le grade de

**DOCTEUR DE L'INPG**

***Spécialité: « Imagerie, Vision et Robotique »***

préparée au laboratoire d'Informatique et de Création Artistique de l'Ecole Nationale Supérieure  
Informatique et de Mathématiques Appliquées de Grenoble, dans le cadre de l'**Ecole Doctorale**

**« Mathématiques, Sciences et Technologie de l'information, Informatique »**

présentée et soutenue publiquement

par

Claire GUILBAUD

le 23 Septembre 2002

### **Titre :**

Modélisation et visualisation de phénomènes naturels simulés par système physique particulière

### ***Directeur de thèse :***

Annie Luciani

### **JURY**

M. Augustin Lux	, Président
M. Marc Neveu	, Rapporteur
M. Jean Louchet	, Rapporteur
Mme. Annie Luciani	, Directeur de thèse
M. Arash Habibi	, Examineur



# Table des matières

---

<b>Table des matières.....</b>	<b>3</b>
<b>Liste des tableaux.....</b>	<b>7</b>
<b>Liste des figures.....</b>	<b>9</b>
<b>Remerciements.....</b>	<b>13</b>
<b>PARTIE I - INTRODUCTION ET ÉTAT DE L'ART.....</b>	<b>15</b>
<b>Chapitre 1 Entrée en matière.....</b>	<b>17</b>
1. Cadre de la thèse .....	18
2. Les attentes.....	19
3. Organisation du document.....	19
4. Convention d'écriture.....	20
<b>Chapitre 2 Etat de l'art en modélisation et visualisation .....</b>	<b>21</b>
1. Vers un modèle générique pour la modélisation de phénomènes naturels.....	22
2. Différentes méthodes de visualisation pour répondre à la problématique.....	30
3. Discussion .....	41
<b>PARTIE II - MODELISATION PHYSIQUE DE PHENOMENES NATURELS.....</b>	<b>43</b>
<b>Chapitre 3 CORDIS-ANIMA, un modelleur-simulateur .....</b>	<b>45</b>
1. Le formalisme CORDIS-ANIMA .....	46
2. Les états de la matière .....	49
3. Le modèle générique .....	50
<b>Chapitre 4 Caractéristiques dynamiques de phénomènes .....</b>	<b>53</b>
1. Les fluides .....	54
2. Les matériaux granulaires .....	56
3. Croissance de cellules végétales.....	57
4. Les fluides turbulents .....	58
5. Les gouttes de fluide .....	60
6. Tableau des dynamiques caractéristiques.....	61
<b>Chapitre 5 Simulations de phénomènes naturels .....</b>	<b>63</b>
1. La croissance de cellules végétales .....	63
2. Le sable .....	67
3. La pâte.....	76
4. Fluides turbulents.....	83
5. Simulation à différentes fréquences .....	91
6. Conclusion.....	93

<b>Chapitre 6</b>	<b>Analyse des données de simulation .....</b>	<b>95</b>
1.	Modèle de croissance de cellules végétales.....	96
2.	Modèle de fluide visqueux .....	96
3.	Modèle de sable.....	99
4.	Modèle de fluides turbulents .....	103
5.	Conclusion.....	106
<b>PARTIE III - VISUALISATION DE PHENOMENES NATURELS.....</b>		<b>107</b>
<b>Chapitre 7</b>	<b>Construction massique récursive .....</b>	<b>109</b>
1.	Elaboration de la méthode CMR en 2D .....	110
2.	La méthode en trois dimensions .....	112
3.	La visualisation par création d'un maillage.....	116
4.	Amélioration de la méthode CMR .....	119
<b>Chapitre 8</b>	<b>Visualisation par structuration du flux .....</b>	<b>121</b>
1.	Le graphe de voisinage.....	122
2.	Visualisation de la surface.....	127
3.	Conclusion.....	135
<b>Chapitre 9</b>	<b>Optimisation des calculs d'habillage : vers un rendu temps réel.....</b>	<b>139</b>
1.	Artefacts visuels .....	139
2.	Discontinuités temporelles .....	140
3.	Limitation de la vitesse de calcul .....	140
4.	Algorithme de rendu de surfaces implicites pour le temps réel .....	152
5.	Conclusion.....	157
<b>Chapitre 10</b>	<b>Visualisation par artefacts .....</b>	<b>159</b>
1.	Principe de base.....	160
2.	Visualisation par artefacts visuels .....	161
3.	Conclusion.....	165
<b>PARTIE IV - CONCLUSION ET REFERENCES BIBLIOGRAPHIQUES .....</b>		<b>167</b>
<b>Chapitre 11</b>	<b>Conclusion .....</b>	<b>169</b>
1.	Bilan .....	169
2.	Perspectives .....	172
3.	Conclusion.....	172
<b>Chapitre 12</b>	<b>Bibliographie.....</b>	<b>175</b>
<b>ANNEXES .....</b>		<b>185</b>
<b>Annexe A</b>	<b>Croissance de cellules végétales.....</b>	<b>187</b>
1.	Quelques notions de biologie végétale.....	187
2.	Définition des courbes splines cardinales .....	188
3.	Modélisation des composants.....	189
4.	Croissance .....	190
5.	Déplacement.....	191
6.	Conclusion.....	191

<b>Annexe B</b>	<b>Spécifications CFP2000.....</b>	<b>193</b>
1.	171.swim .....	193
2.	172.mgrid .....	194



# Liste des tableaux

---

Tableau 1 : Types de configurations utilisées en fonction du phénomène voulu.....	51
Tableau 2 : Images de gel.....	56
Tableau 3 : Paramètres de la cellule en forme de pentagone.....	66
Tableau 4 : Observations pour différentes fréquences de simulation.....	93
Tableau 5 : Conditions pour déduire si un sommet est en bordure de surface .....	118
Tableau 6 : Débit et latence d'instructions pour six microprocesseurs.....	142
Tableau 7 : Evaluation de la vitesse maximal théorique de calcul de la fonction de densité .....	146
Tableau 8 : Evaluation de la vitesse réelle de calcul de la fonction de densité .....	146
Tableau 9 : Classement des 20 meilleurs processeurs aux tests 171.swim et 172.mgrid (du plus rapide au plus lent).....	148





# Liste des figures

---

Figure 1 : Principe de l'écran d'épingles.....	41
Figure 2 : Schématisation de CORDIS-ANIMA.....	45
Figure 3 : Point L (liaison) et point M (matériel).....	47
Figure 4 : MAT et LIA.....	48
Figure 5 : Les différents états de la matière du plus solide au plus gazeux.....	50
Figure 6 : Caractéristiques Dynamiques d'un matériau fluide.....	55
Figure 7 : Schéma de stabilité d'un tas de sable.....	57
Figure 8 : Caractéristiques dynamiques des matériaux granulaires.....	57
Figure 9 : Caractéristiques Dynamiques de la croissance de cellules.....	58
Figure 10 : Formation d'un tourbillon de Kelvin-Helmholtz.....	59
Figure 11 : Formation de l'allée de von Karman derrière un obstacle.....	60
Figure 12 : Tourbillon de Kelvin-Helmholtz et Allée de von Karman.....	60
Figure 13 : Caractéristiques d'une goutte.....	61
Figure 14 : Différentes formes de la cellule.....	64
Figure 15 : Liaisons structurelles des cellules.....	66
Figure 16 : Modèles de croissance de cellules végétales.....	67
Figure 17 : Modèle 2D de l'environnement physique à sol rugueux.....	69
Figure 18 : Modèle 2D de l'environnement physique à sol lisse.....	69
Figure 19 : Liaison entre particules de la matière.....	70
Figure 20 : Liaisons entre l'environnement physique et la matière.....	70
Figure 21 : structure du modèle 2D à sol rugueux.....	71
Figure 22 : structure du modèle 2D à sol lisse.....	71
Figure 23 : Modèle 2D de sable à sol rugueux.....	72
Figure 24 : Modèle 2D de sable à sol lisse.....	73
Figure 25 : Environnement du modèle 3D à sol rugueux.....	74
Figure 26 : Environnement du modèle 3D à sol lisse.....	74
Figure 27 : Modèle de sable 3D à sol lisse.....	76
Figure 28 : Modèle 3D de sable à sol rugueux.....	76
Figure 29 : Liaison entre particules de matière.....	77
Figure 30 : Liaison entre l'environnement physique et la matière.....	78
Figure 31 : Modèle 2D de pâte.....	79
Figure 32 : Modèle 2D de fluide expansif.....	79
Figure 33 : Modèle 3D de pâte.....	81
Figure 34 : Modèle 3D de gel.....	82
Figure 35 : Modèle 3D de goutte-à-goutte de pâte.....	82
Figure 36 : Modèle de couche simulée.....	84
Figure 37 : Principe de l'injection et du recyclage.....	84
Figure 38 : Interaction visqueuse source de rotation.....	85
Figure 39 : Modèle à quatre injecteurs.....	86
Figure 40 : Modèle à cinq injecteurs.....	86

Figure 41 : Obstacle dans le sillage d'un écoulement de fluide.....	87
Figure 42 : Modèle d'allée de von Karman .....	88
Figure 43 : Modèle de colonne de fumée .....	88
Figure 44 : Environnement physique du modèle de colonne de fumée.....	89
Figure 45 : Colonne de fumées.....	90
Figure 46 : Modèle de sable 2D simulé à 687Hz .....	93
Figure 47 : Vitesse des particules de cellule lors de la simulation .....	96
Figure 48 : Vitesses (point) et accélération (vecteur) de la pâte 2D (indication de direction : rouge pour l'axe X, vert pour Y, bleu pour Z) .....	97
Figure 49 : Structuration des données vitesses et accélérations .....	97
Figure 50 : Vitesses (point) et accélération (vecteur) de la pâte 3D.....	98
Figure 51 : Vitesses (point) et accélération (vecteur) du fluide expansif 2D .....	98
Figure 52 : Vitesses (point) et accélération (vecteur) du gel 3D.....	99
Figure 53 : Vitesses (point) et accélération (vecteur) du sable 2D à sol rugueux .....	100
Figure 54 : Vitesses (vecteur) et accélération (point) du sable 2D à sol rugueux .....	100
Figure 55 : Organisation du tas avant restructuration .....	101
Figure 56 : Vitesses (point) et accélération (vecteur) de sable 3D (sol rugueux).....	101
Figure 57 : Vitesses (point) et accélération (vecteur) de sable 2D à sol lisse .....	102
Figure 58 : Vitesses (point) et accélération (vecteur) du sable 3D à sol lisse .....	103
Figure 59 : Vitesses (point) du tourbillon de Kelvin-Helmholtz (indication de direction : rouge pour l'axe X, vert pour Y, bleu pour Z) .....	104
Figure 60 : Vitesses (point) de l'allée de von Karman.....	104
Figure 61 : Accélérations (point) de l'allée de von Karman .....	104
Figure 62 : Schématisation de l'allée de von Karman dans le sillage d'un obstacle .....	105
Figure 63 : Vitesses (point) du jet .....	105
Figure 64 : Image issue d'une simulation 3D .....	109
Figure 65 : Masque de convolution de taille 2 .....	112
Figure 66 : Voisinage d'un voxel en 3D .....	112
Figure 67 : Initialisation du volume discret.....	113
Figure 68 : les 8 sous-régions autour d'un voxel .....	114
Figure 69 : Construction vers les plus fortes densités les plus proches (coupe du volume).....	115
Figure 70 : volume construit (les voxels sont des sphères) .....	116
Figure 71 : Les 16 cas de base.....	117
Figure 72 : Volumes construits par facettes .....	119
Figure 73 : Volumes construits avec la méthode modifiée .....	120
Figure 74 : Quelques graphes de voisinage 2D et 3D .....	124
Figure 75 : Définition d'un vecteur voisin moyen.....	126
Figure 76 : Vecteurs voisins moyens .....	127
Figure 77 : Particules aux bords.....	127
Figure 78 : Diagrammes de Delaunay et de Voronoï.....	128
Figure 79 : Quelques surfaces implicites obtenues par lancer de rayons .....	133
Figure 80 : Quelques surfaces implicites obtenues par polygonalisation.....	135
Figure 81 : Structure de stockage du maillage .....	150
Figure 82 : Première approximation de la surface.....	154
Figure 83 : Affinement de la surface.....	155
Figure 84 : Exemple de filtre de taille $Z=3$ (ici filtre moyennneur).....	162

Figure 85 : Exemples de filtres gaussiens .....	163
Figure 86 : Filtre médian .....	163
Figure 87 : traitement de la carte de densité.....	164
Figure 88 : Décomposition d'un voxel en élément graphique .....	165
Figure 89 : Quelques exemples de visualisation par artefacts.....	165
Figure 90 : Coupe d'une cellule végétale.....	188
Figure 91 : Forme de base des différents composants internes .....	190
Figure 92 : Fonction d'évolution de la croissance des composants internes .....	191
Figure 93 : Structuration de la cellule avec ces composants .....	192
Figure 94 : Différentes visualisations possibles pour différentes formes de cellule .....	192



# Remerciements

---

Je remercie Annie Luciani pour m’ avoir fait connaître le domaine de la modélisation physique, paradigme passionnant pour la multitude des créations réalisables, Jean-Loup Florens pour son aide et ses connaissances en de nombreux domaines. Merci également à Claude Cadoz, directeur de l’ACROE.

Je tiens à remercier Marc Neveu et Jean Louchet qui ont accepté d’être rapporteurs, les membres du jury Arash Habibi, Annie Luciani et Augustin Lux qui ont participé à ma soutenance de thèse.

Je remercie tous les stagiaires, thésards et permanents de l’ICA et de l’ACROE pour la bonne humeur qu’ils ont apportée. Une pensée particulière pour Nicolas (l’autre thésard du labo), Yannick, Cécile, Jean-Loïc, Maria, Julie, Geoffrey, Claire, Cyrille, Pierre, et tous les autres qui m’ont supportée durant ces trois années.

Un grand merci à Ambroise qui a bien voulu relire cette thèse à plusieurs reprises, et à toute ma famille pour leur soutien.



**Partie I - INTRODUCTION ET ETAT DE  
L'ART**





# Chapitre 1 Entrée en matière

---

*"God Himself, sir, does not propose  
to judge a man until his life is over.  
Why should you and I?"*

*Dale Carnegie, "How To Win Friends And Influence People"*

« L'ordinateur est une machine automatique de traitement de l'information<sup>1</sup> ». L'ordinateur tel que nous le connaissons aujourd'hui est le fruit de l'évolution sur plusieurs siècles de recherches sur la gestion des données. Depuis la carte perforée, la machine de Turing, l'objet s'est métamorphosé en un ensemble de composants obéissant à une suite d'instructions formées d'opérations arithmétiques et logiques.

L'avènement des ordinateurs pour le public s'est fait dans les années 70. Dès lors son développement s'est accru de manière spectaculaire en envahissant nombre de corps de métiers. Il s'est peu à peu avéré être plus qu'un outil de gestion des données. Il est devenu un outil de conception à part entière. Il nous a permis de rédiger ce mémoire, de développer les techniques exposées dans celui-ci, de faire nos recherches sans courir à droite et à gauche pour trouver le précieux document indispensable, d'écouter le « Requiem en ré mineur KV 626 » de Wolfgang Amadeus Mozart lors de la rédaction de ce chapitre : c'est un outil multifonctionnel de création, d'élaboration.

Dès sa mise à disposition pour le plus grand nombre, il fut utilisé pour créer. L'image de synthèse, et la musique par ordinateur étaient nées. Une image de synthèse est une image engendrée par l'ordinateur. Son origine *in vitro*, ou par manipulation en fait un outil d'expression à part. C'est une sculpture, une peinture, une caricature, une photographie.

---

<sup>1</sup> Définition du Petit Larousse, édition 2000

L'immobilité de l'image peut être un frein à notre imagination. Même si certaines œuvres figées racontent à elles seules une histoire, nous ne pouvons voir évoluer ses acteurs, ses paysages. L'ordinateur est alors devenu un outil d'animation, de conception de dynamiques. Ce nouveau domaine d'application est fort prisé par le 7<sup>ème</sup> art. Il permet de mêler fantastique et réalité à l'intérieur d'un même film sans que les deux soient différenciables. Le cinéma se confond de plus en plus avec l'animation par ordinateur.

L'étape suivante est au-delà du photo-réalisme. La puissance de calculs des nouvelles machines permet de simuler de mieux en mieux l'apparence des choses naturelles. Cette avancée technologique nous autorise à concevoir un univers entièrement virtuel modélisé dans sa globalité. C'est un monde à part, où divers domaines scientifiques côtoient les arts plastiques.

Les recherches effectuées au cours de ces trois années de thèse sont en étroite relation avec une partie infime du domaine de l'animation naturelle. Nous nous sommes penchés sur l'élaboration de modèles d'écoulements de matière par système physique particulière, et la conception d'outils pour représenter visuellement ces modèles purement dynamiques.

Créer un modèle d'un phénomène naturel, c'est simuler, reproduire sa dynamique, sa forme.

## 1. Cadre de la thèse

A l'ICA et dans les domaines de l'image de synthèse, de nombreux travaux sont réalisés pour modéliser des objets à topologie variable ou fixe, des objets dynamiques complexes. Depuis plusieurs années, les acteurs de la recherche développent des logiciels pour simplifier la création d'animations entièrement développées par ordinateur.

Il existe diverses manières de reproduire un comportement. Ce chapitre ne listera pas l'ensemble des méthodes existantes, mais présentera de manière brève et, espérons-le, concise le formalisme CORDIS-ANIMA de modélisation dynamique et sonore développé au sein du laboratoire. Le concept sera expliqué en détail au Chapitre 3. Cette présentation se fait dans l'optique de fournir un aperçu des problèmes rencontrés au cours des travaux de recherche.

Simuler un phénomène naturel, ce n'est pas copier, ou imiter ses moindres mouvements mais reproduire les caractéristiques qui nous permettent de l'identifier. CORDIS-ANIMA ([cadoz93]) est un modéleur-simulateur physique particulière se situant à un niveau mésoscopique ([luciani00]) de représentation. Un objet, un phénomène naturel est transformé en un réseau de masses ponctuelles interagissant les unes avec les autres selon certaines lois de la physique. C'est un entrelacement de masses et d'interactions aux caractères variés. L'assemblage de ces matériaux de base fournit une dynamique et donc une forme pour l'objet modélisé.

Disposer de la simulation, c'est disposer des coordonnées des masses ponctuelles, masses réparties dans l'espace sans information de topologie les reliant sur une durée. Qu'en faire pour les visualiser, i.e. pour créer une forme spatiale adéquate qui les enveloppe sans dissimuler les dynamiques obtenues lors de la modélisation ? L'être humain que nous sommes est en mesure de bâtir mentalement une

surface pour cet ensemble de particules sans avoir connaissance de la topologie ayant conduit à sa création. Cela permet de situer subjectivement une surface. Nous avons donc cherché à réaliser un ensemble d'outils de visualisation prenant pour entrée ce type de données.

## 2. Les attentes

Nous nous intéressons plus particulièrement aux phénomènes naturels et notamment à ceux qui ne présentent pas de topologie spatiale fixe tels que les fluides, pâtes, sable, mousses et fumées et plus généralement aux processus de morphogenèse tels que la croissance cellulaire. Ces phénomènes ont fait l'objet de simulations 2D et 3D (sauf pour les cellules pour lesquelles les simulations n'ont été que bidimensionnelles). Ils constituent une base de données pour évaluer et valider les différentes méthodes de visualisation mises en place. Une question fondamentale se pose : que voulons-nous visualiser et dans quel but ?

L'outil idéal dans l'animation par ordinateur que nous avons envisagé devrait permettre, à terme, de composer une séquence en piochant dans une base de données proposant divers objets et phénomènes naturels. Ces composants de la scène seraient alors liés par des relations d'interactions appropriées pour les faire évoluer de façon « naturelle ».

La représentation visuelle des modèles doit rendre compte de ce qu'elle représente en créant une forme, un volume à partir d'un ensemble de points sans structure disséminés dans l'espace.

Les recherches se sont organisées ainsi : dans un premier temps, nous avons réalisé un état des lieux sur les différentes méthodes de rendu, de modélisation, de visualisation en rapport avec l'animation des phénomènes naturelles.

Nous avons ensuite élaboré le modèle d'un premier écoulement de pâte en trois dimensions. Ce modèle de base fut un test pour la première méthode d'habillage de particules.

Nous avons par la suite conçu un ensemble de modèles de phénomènes naturels pour valider et améliorer les méthodes implémentées pour qu'elles aient à disposition une large gamme de caractéristiques dynamiques, et soient ainsi à même de représenter n'importe quel écoulement.

## 3. Organisation du document

Le manuscrit de thèse est structuré de la façon suivante. Nous passons en revue différentes techniques de simulation et d'animation par ordinateur, en mettant en avant avantages et inconvénients de chacune d'entre elles. Puis nous abordons la partie simulation de phénomènes naturels de nos recherches. Nous commençons par expliquer la simulation physique particulière avec CORDIS-ANIMA, puis détaillons et analysons les modèles réalisés. Dans un troisième temps, nous présentons les méthodes réalisées pour répondre à notre problématique de visualisation de particules dispersées, sans information de structure ou topologie. Nous concluons alors en résumant les apports de nos travaux.

## 4. Convention d'écriture

Nous définissons ici différents termes qui reviendront à de nombreuses reprises dans le manuscrit.

**Masse ponctuelle** : point matériel sans taille, sans masse. Le terme décrit un point matériel du domaine de la modélisation physique.

**Particule** : point géométrique dont les coordonnées sont celles enregistrées dans le film comme résultat de la simulation physique. Le terme décrit un point matériel du domaine de la visualisation.

**Interaction** : liaison interparticulaire entre masses ponctuelles. Le terme se rapporte à la modélisation.

**Connexion** : liaison entre particules. Le terme se rapporte à la visualisation.

# Chapitre 2 Etat de l'art en modélisation et visualisation

---

Modéliser un phénomène dynamique<sup>2</sup>, c'est d'abord concevoir, élaborer un modèle théorique de ces évolutions. Il ne s'agit pas de recréer sa forme mais plutôt de reproduire sa dynamique. Pour ces phénomènes, si la dynamique est juste, alors la forme le sera aussi, car la forme n'est que la partie visible restituant les dynamiques internes.

Nous n'avons pas pour but l'élaboration d'un état de l'art exhaustif des méthodes de modélisation et de visualisation de phénomènes naturels, mais plutôt une constitution d'une base de connaissance générale pour poursuivre plus en avant nos recherches.

L'organisation de ces sections se base sur le cheminement de la conception d'une animation : observation, spécification, modélisation, puis visualisation du phénomène.

La représentation de phénomènes naturels fait partie de la quête de la reproduction du monde qui nous entoure, au même titre que l'animation de personnages, la simulation de foules, la modélisation d'objets manufacturés. L'élaboration de tous ces modèles ne pouvant être réalisée dans le temps imparti, nous nous sommes restreints à la modélisation et visualisation d'écoulements de fluides (turbulents ou non), de matériaux granulaires, de croissances de cellules végétales, qui représentent des cas typiques du domaine à explorer.

Pour juger de la pertinence de l'utilisation de la modélisation physique particulière telle qu'elle est pratiquée au sein du laboratoire ([[luciani86](#)], [[cadoz93](#)]), nous nous devons de connaître d'autres méthodes de modélisation de phénomènes naturels et de matériaux déformables.

Nous n'avons pas de méthode spécifique de visualisation, car jusqu'alors, peu de travaux se sont penchés sur la visualisation de points dispersés dans l'espace, points sans indication de topologie. Il nous faut donc aborder nos recherches avec un maximum de connaissances sur les méthodes de

---

<sup>2</sup> Un phénomène pour lequel l'évolution temporelle représente la caractéristique essentielle.

visualisation-rendu de phénomènes naturels. Cette étude nous permettra de sélectionner les méthodes susceptibles de nous convenir et/ou nous servir de base pour en concevoir de nouvelles.

## 1. Vers un modèle générique pour la modélisation de phénomènes naturels

Les écoulements de fluides, qu'ils soient turbulents, visqueux ou granulaires ne peuvent être modélisés par assemblage de formes géométriques simples ou par définition de fonctions d'évolution, de par les mouvements et les formes complexes qu'ils présentent. Ces complexités engendreraient un temps de création long pour obtenir une dynamique et une forme réaliste si le travail devait être réalisé à la main. Il vaut mieux s'en remettre à une génération automatique de l'évolution spatiale du phénomène par diverses méthodes.

Logan et al. ([logan94]) ont synthétisé nombre de méthodes de simulations d'objets déformables. Ils ont distingué deux catégories de méthodes : les méthodes contraintes et les méthodes physiques. La première catégorie ne peut s'adapter à la modélisation de phénomènes naturels, car la production de dynamiques complexes ne s'obtient pas facilement par la description d'un ensemble de règles ou de contraintes. Les dynamiques macroscopiques issues de mouvements microscopiques sont difficiles à déterminer de manière simple, celles résultant d'une interaction avec l'environnement du phénomène ne peuvent être contrôlées que par un ensemble de règles exhaustives prévoyant le moindre cas source de mouvement. La seconde est celle des méthodes physiques (« physically based model ») qui regroupent un ensemble d'approches basées sur les lois physiques.

Plus généralement, la dernière catégorie peut être scindée en deux entités : celles basées sur des *modèles de la physique*, et celles sur des *modèles physiques*. La première se sert des équations du phénomène élaborées par la connaissance physique pour simuler la dynamique et la forme, tandis que la seconde applique les lois mathématiques de base de la physique pour réécrire le modèle physique lui-même. Ces lois sont plus génériques que le modèle qu'elles permettent de formuler et sont en amont du phénomène modélisé.

La section qui suit présente les différentes méthodes de simulation sans préciser le type de visualisation utilisée, et ce même si la plupart de ces méthodes effectuent en parallèle simulation et habillage. Cette omission se fait dans un souci de clarté. Nous présentons, dans un premier temps, les méthodes de la physique (pour les fluides, il s'agit de méthodes dites CFD : « Computational Fluid Dynamics »), puis les méthodes contraintes (agents, comportementales, stochastiques), enfin nous aborderons les méthodes physiques.

### 1.1. Modèles de CFD (« Computational Fluid Dynamics »)

Les méthodes de CFD utilisent les équations de la dynamique des fluides (Navier-Stokes, Euler, etc.). Elles permettent la modélisation réaliste de fluides et de fluides turbulents (gaz, fumée).

En 1990, Kass et Miller ([kass90]) ont recours à une version simplifiée des équations de Navier-Stokes pour modéliser une étendue d'eau peu profonde. Cette méthode utilise une carte des hauteurs pour représenter la surface de l'eau. Une telle méthode simule en temps réel ; les dynamiques résultantes sont réalistes (vague, trou, courant) mais l'usage d'une carte des hauteurs ne permet pas de produire une vague qui se déroule dans sa totalité.

Foster et Metaxas ([foster97]) utilisent ces mêmes équations pour la modélisation de fluides. La grille de discrétisation est grossière, mais les mouvements sont correctement rendus. Cependant leur méthode n'est stable que si l'intervalle de temps entre deux pas de calculs est petit. Ils avaient utilisé une méthode similaire pour animer les mêmes phénomènes en 1996 ([foster96]). En 1997, Stam ([stam97]) propose un procédé de simulation plus stable utilisant une méthode d'intégration semi-lagrangienne des équations de la dynamique des fluides. Mais les petits mouvements sont problématiques. Dans [Stam99], l'auteur se sert d'une combinaison d'équations de Navier-Stokes, de principe de résolution basé sur la physique des fluides (« physics-based fluid solvers ») et de textures solides pour réaliser des modèles de fluides complexes réalistes. La méthode n'est pas assez précise pour être utilisée par les physiciens, car il s'agit là d'une méthode pour l'informatique graphique. Cette méthode a l'avantage d'introduire l'interactivité de l'utilisateur avec le fluide modélisé.

Witting ([witting99]) s'est penché sur une méthode de modélisation de fumées et d'eau basée sur ces mêmes équations auxquelles des contraintes (de conditions aux bords, de densité, de chaleur) ont été ajoutées. Les résultats obtenus sont satisfaisants. Cependant ces simulations sont en deux dimensions.

Weimer et al. ([weimer99]) modélisent des écoulements de fluides à travers des subdivisions, produisant ainsi un champ de vecteurs discrets arbitraires suivant les conditions initiales. Un tel système basé sur la partie linéaire des équations de Navier-Stokes a été réalisé dans un souci d'interaction et de visualisation temps réel d'écoulements de fluides. Cependant il ne traite que de fluides non turbulents ; la modélisation d'écoulements plus complexes nécessiterait la prise en compte des équations de Navier-Stokes dans leur globalité, rendant ainsi la méthode non temps réel.

Fedkiw et al. ([fedkiw01]) modélisent des fluides turbulents à l'aide des équations d'Euler incompressibles. La méthode proposée est stable et présente peu de dissipation numérique car elle emploie une technique d'intégration semi-lagrangienne et l'équation de pression de Poisson pour réinjecter de l'énergie au modèle.

Foster et al., dans [foster01], ont mis au point un système pour l'animation par ordinateur manipulant des liquides visqueux pouvant interagir avec leur environnement. Pour cela ils combinent une résolution semi-lagrangienne des équations avec une nouvelle approche pour calculer les dynamiques des fluides autour d'objets solides. Le fluide est séparé en deux parties : sa surface et son intérieur. La surface du fluide est une combinaison hybride entre particules sans inertie et une surface implicite (appelée courbe de niveau). Avec une telle approche, les auteurs reproduisent des liquides allant de l'eau à la boue épaisse.

Les méthodes utilisant les équations de Navier-Stokes ne se cantonnent pas à la modélisation de fluides ou de fluides turbulents. Yngve et al. dans [yngve00] proposent un modèle d'explosion dont



la dynamique se base sur les équations des écoulements de fluides visqueux et compressibles. Par cette méthode, il est possible de simuler différents types d'explosions, ainsi que l'impact de l'explosion sur son environnement. C'est par l'intermédiaire de techniques de rendu appropriées que l'on modélise différents types d'explosion (boules de feu, nuages de poussière).

Barrero et al. ([barrero00]) se sont servis de ces équations pour modéliser la combustion d'objets. Cependant ils utilisent une représentation compressible des équations pour que d'autres effets, tels que la compression, puissent être modélisés.

*Ces méthodes de la physique* utilisant les équations de Navier-Stokes, ne peuvent servir à la modélisation de matériaux granulaires ou de mouvement de foules. Ils ne sont pas génériques et ne sont pas applicables à tout phénomène dont la caractéristique principale et commune est la dynamique réglant les modifications de topologie spatiale.

## 1.2. Modèles comportementaux, stochastiques, à agents

### 1.2.1 Modèle à agents

En 1997, Gareau propose dans sa thèse ([gareau97]) une architecture pour l'interaction entre systèmes animés hétérogènes (système de particules, géométrie des solides, etc.). La modélisation se fait par agents. Ils communiquent entre eux par messages, l'interaction entre objet se fait au moyen d'une causalité unique, la force. Avec l'agent particulaire, des phénomènes naturels, comme les nuages, la pluie sont simulés.

### 1.2.2 Modèles stochastiques

Dès 1983, Reeves ([Reeves83]) propose une modélisation par système particulaire d'objets à contour flous (feu, nuage, eau). L'objet est défini par son volume, lui-même défini par un nuage de particules. Au cours de la simulation, les particules se déplacent sous l'action d'un processus stochastique, changeant ainsi la forme du volume. Elles peuvent aussi mourir. Il est alors possible d'obtenir une dynamique, un mouvement satisfaisant pour un ensemble de phénomènes mais les fumées n'en font pas partie leurs dynamiques sont trop complexes à reproduire.

### 1.2.3 Modèles comportementaux

Sims ([sims90]) propose une démarche ayant recours à un simulateur comportemental incluant des particules pour représenter de l'eau, de la neige, du feu, et le vent. Il est possible de contrôler les déplacements par un modèle physique ou un simple suivi de courbe. Les particules ont leur propre taille, forme, couleur et transparence, permettant ainsi la simulation de neige, de vent, et d'eau. La description du modèleur par Sims ne traite pas de la collision entre particules. La reproduction d'écoulements de fluides ne peut donc être possible.

Xin et al. ([xin93]) ont développé une méthode de modélisation de sols dans laquelle la simulation est conduite par les événements. Un événement peut être une collision entre particules représentant le sol ou une collision entre le sol est un obstacle. Le  $\Delta t$  est déterminé par l'instant du prochain événement. Donc lorsqu'il n'y a aucun événement, les particules suivent une trajectoire

parabolique dans le champ gravitationnel. Lors des contacts, des forces de friction et de dissipation apparaissent. Lorsqu'un événement se produit, le modéleur quantifie l'énergie perdue et place la particule en conséquence. Le sol peut posséder différentes caractéristiques. Une telle technique ne pourrait s'appliquer à la modélisation de phénomènes naturels de type écoulement de fluides, car le nombre de collisions rendrait le temps de calcul fort long.

Chiba et al. ([chiba95]) ont travaillé sur une méthode pour simuler des courants marins par modèle particulaire comportemental. Leur méthode doit pouvoir rendre les mouvements complexes de l'eau surtout en surface de celle-ci. L'eau consiste en un nombre disjoint de petites parties imaginaires appelées particules d'eau (« water-particules »). La particule d'eau est un point matériel qui se déplace selon les lois de Newton du mouvement : elle a un volume, interagit élastiquement avec un obstacle et avec les autres particules selon une loi de probabilité proportionnelle à la densité, et est soumise aux forces existantes entre particules (force attractive, force répulsive, force visqueuse). Ces forces n'ont qu'une portée réduite. Cette méthode bien que donnant des dynamiques convaincantes ne s'intéresse qu'à un seul type de phénomène. Elle ne semble pas être en mesure de modéliser d'autres phénomènes de manière simple : toutes les règles comportementales seraient à revoir.

La neige est un phénomène complexe lorsqu'elle s'accumule sur divers objets. Les différentes études parues s'intéressent à son accumulation sur différents objets d'une scène plutôt qu'à son mouvement lorsqu'elle entre en contact avec un support. Nishita et al. ([nishita97]) la modélisent par un ensemble de « metaballs » de différentes tailles pour distinguer les molécules d'air des molécules de glace. Fearing et al. ([fearing00]) ont préféré lancer des particules de neige sur les objets de la scène puis appliquer des règles d'arrangement de la matière pour définir la façon dont la neige s'accumule.

Les nuages évoluent selon les conditions atmosphériques. Leur aspect se trouve modifier par la position de l'observateur ou du soleil. Dobashi et al. ([dobashi00]) propose une méthode simple et non coûteuse en temps de calcul de modélisation de nuages en 3 dimensions (indispensable pour représenter la distribution de la densité). Leur évolution se fait à l'aide d'automates cellulaires prenant en compte plusieurs règles simples de transition. Ces automates simplifient la dynamique de formation des nuages.

## 1.3. Modèles physiques

La simulation physique pour l'animation par ordinateur débute au début des années 80. Elle a pour but la reproduction de phénomènes naturels complexes. Les chercheurs n'étant pas les personnes qui réalisent les animations, ces méthodes se doivent d'être intuitives et simples à manipuler par des non-physiciens.

### 1.3.1 Modèles dérivés des « Smooth Particles Hydrodynamics »

Dans l'optique d'un modéleur générique, Desbrun et al. ([desbrun96]) ont développé un nouveau type de méthode dérivée des SPH (« Smooth Particles Hydrodynamics ») définies dans [Monaghan92]. Cette technique a été réalisée pour des astrophysiciens pour l'aide à la résolution des problèmes de résistance des matériaux. Elle avait déjà servi sous une forme simplifiée pour la

modélisation de fluides turbulents et de feu dans [stam93] et [stam95]. Les matériaux reproduits sont gouvernés par une équation d'état macroscopique, c'est-à-dire que leur comportement global est indépendant de la fréquence d'échantillonnage de la simulation. Ce formalisme modélise les milieux continus hétérogènes en deux niveaux : une modélisation des dynamiques macroscopiques par système particulaire de masses ponctuelles en interaction, auquel se superpose un procédé d'interpolation de la matière en remplaçant chaque masse ponctuelle par une fonction de densité massique. Cette méthode est utilisée en particulier pour l'étude des interfaces milieux continus-milieux solides (fluide-aile d'avion) ou l'inverse (corps solides perturbateurs pénétrant dans les turbines). L'application de cette méthode à l'animation a permis la modélisation de corps présentant de grandes déformations et a mené à une définition de la surface de l'objet comme une isosurface d'une fonction masse/densité, renforçant ainsi l'intérêt de la modélisation par surfaces implicites présentée par le même laboratoire dans [desbrun95], principe repris et complété dans [gascuel\_mp97]. Ces travaux ont fait l'objet de la thèse de Mathieu Desbrun [desbrun97]. Le document reprend, point par point, la mise en place de la méthode des SPH.

Dans [stam93], les auteurs ont élaboré une méthode d'écoulements de fluides turbulents. Un composant déterministe se charge de simuler le comportement global de l'écoulement, un autre utilise une synthèse espace-temps de Fourier pour modéliser les micro-mouvements.

La simulation de phénomènes naturels peut s'avérer fort utile pour la prévention de risques naturels. L'étude menée par Stora et al. ([stora98]) s'est intéressée à la modélisation de coulées de lave par SPH, tout en prenant en compte des paramètres physiques comme la température des matériaux, les échanges thermiques. Ainsi, ils ont pu obtenir des simulations réalistes du point de vue de la dynamique et des déformations. Leur méthode modélise un changement d'état : le passage de l'état liquide à l'état solide de la lave. Cette technique a été reprise dans [gascuel\_jd98], mais les auteurs n'ont pu avoir recours à ce principe de simulation pour la modélisation de chutes de pierres. En effet, le sol, représenté par un tapis plastico-élastique, réagit en fonction des chutes de pierre à la manière de pistons, indépendamment du voisinage. La roche est un solide rigide défini par une surface implicite (une masse et un tenseur interne) soumise aux lois de la dynamique. Les gros blocs de roche étant modélisés comme un ensemble de roches, les fractures ont pu être obtenues.

La modélisation par SPH, qui relève dans son principe de la modélisation des dynamiques macroscopiques, ne peut aboutir à la simulation de nombreux phénomènes naturels, car nombre de mouvements microscopiques engendrent des mouvements macroscopiques (phénomènes d'émergence, de chaos déterministe, etc.), difficilement représentable directement à l'échelle macroscopique.

### 1.3.2 Modèles surfaces implicites

Dans [gascuel\_mp97] la modélisation se fait par surfaces implicites générées par squelette. Cette approche présente l'avantage de posséder un mécanisme de détection efficace des collisions ainsi qu'une modélisation de contacts entre corps déformables. Cependant, pour que les comportements des objets modélisés soient naturels, il a fallu se servir d'un graphe de voisinage pour que seuls les squelettes joints de l'objet fusionnent. Les collisions entre squelettes joints ne sont pas détectées. La modélisation par surface implicite générée par squelette ne peut représenter tous les types de phénomènes, notamment lorsque l'on est face à une dynamique complexe ou que l'on souhaite reproduire des fractures ou des mouvements browniens.

DaDalto dans [dadalto97] a consacré sa thèse à la modélisation de phénomènes naturels à l'aide de « blobs<sup>3</sup> ». Cette méthode, bien que rendant des surfaces lisses, ne peut donner entière satisfaction dans la modélisation de phénomènes turbulents.

### 1.3.3 Modèles par éléments finis

La méthode des éléments finis est souvent utilisée pour modéliser des corps déformables. Dans [obrien99], les auteurs proposent une façon de simuler une fracture dans un volume 3D en analysant les tenseurs des éléments finis pour déterminer où la fracture doit débiter et dans quelle direction elle doit se propager. Si l'on fait varier la forme du volume initial ou les propriétés du matériau ou les conditions initiales de la simulation, il est possible d'avoir différents types de fractures.

En 1999, Debonne et al. ([debunne99]) proposent un nouveau modèle à base d'éléments finis structuré par une élasticité linéaire (calculs d'opérateurs différentiels discrets) et un stockage dans une grille non régulière de l'objet. Cette solution, pour l'animation de matériaux élastiques déformables en temps interactif, offre la possibilité de raffiner le maillage de l'objet : plus la déformation est importante, plus celui-ci est discrétisé. Cependant, il faut assurer un comportement global similaire quelle que soit la taille de discrétisation. Cette méthode a été approfondie par la suite pour permettre la modélisation de corps déformables en temps réel ([debunne00]). De nouveaux opérateurs différentiels pour le mixage d'éléments finis ont été introduits. La technique est simple et générale pour traiter des maillages adaptatifs qui décrivent un objet. La complexité d'un tel système est du même ordre que celle des systèmes masse-ressort.

La modélisation par éléments finis, bien qu'imitant efficacement les matériaux déformables, est difficilement utilisable pour modéliser des phénomènes naturels de type écoulements de matière, dont la caractéristique essentielle est la dynamique du changement de topologie interne du matériau. Son principe de base repose sur la continuité du matériau par forces de contact interne.

### 1.3.4 Les modèles physiques particuliers : Héritage de Greenspan

Plusieurs études se sont inspirées des travaux de Greenspan ([greenspan73]). Dès 1973, Greenspan s'est penché sur la recherche d'un modèle physique générique permettant de simuler de nombreux phénomènes naturels. Pour lui, tous les comportements physiques visibles de notre monde (à l'échelle macroscopique) émergent de mouvements microscopiques. Ainsi la reproduction de phénomènes est possible en tirant profit uniquement des lois actives au niveau microscopique. L'objet est donc représenté comme un ensemble de particules interagissant de manière non dissipative par une loi d'interaction potentielle générique : la fonction de Lennard-Jones.

$$F(d) = -\frac{\alpha}{d^m} + \frac{\beta}{d^n}$$

---

<sup>3</sup> Terme de Blinn pour désigner un type de surface implicite ([blinn82]).

Lorsque  $d$  tend vers zéro, les particules se repoussent, lorsque  $d$  est grand, elles s'attirent. Entre ces deux états les particules se comportent comme ci elles étaient reliées par une interaction cohésive.

Dès lors plusieurs études basées sur ses travaux ont été menées : [miller89], [tonnesen91], [terzopoulos87], [terzopoulos88], [terzopoulos89], [luciani91]. Les différences entre ces études portent sur l'expression de la fonction potentielle (analytique pour [miller89], [tonnesen91], [terzopoulos89] et discrète pour [luciani91]), et l'ajout explicite ou non d'un terme dissipatif (non explicite pour [tonnesen91], [terzopoulos89] et explicite pour [miller89], [luciani91]).

Dans [miller89], les auteurs ajoutent au modèle de Greenspan un terme purement visqueux (dépendant de la vitesse) aux interactions inter-particulaires, introduisant ainsi une fonction d'atténuation entre particules. Tonnesen ([tonnesen91]) modélise des fluides et des solides à l'aide d'une unique méthode basée sur la dynamique de Newton auquel il a ajouté une énergie thermique. Le changement de géométrie ainsi que les mouvements des particules sont la conséquence de forces extérieures et d'énergie thermique appliquée au modèle. Un objet est solide si sa température est basse, tandis que lorsqu'elle est élevée, il devient liquide. Pour obtenir un effet dissipatif, l'auteur a introduit un terme de viscosité ambiante.

Terzopoulos, Platt et Fleisher ([terzopoulos87]) ont représenté des objets non rigides au comportement de fluide capable de conduction thermique, de thermo-élasticité, de mélange grâce à un système d'équations différentielles. En 1988, Terzopoulos et al. ([terzopoulos88]) proposent un modèle plus complet où un objet est composé d'une structure de référence et d'une structure de déformation. La méthode est moins coûteuse que précédemment puisque le mouvement est régi par les lois de la dynamique du solide. Le composant caractérisant la déformabilité de l'objet dans l'équation différentielle n'est calculé que pour connaître la forme courante de l'objet. Ce modèle est généralisé dans [terzopoulos89] où l'objet peut se déformer de manière inélastique (fracture possible).

Luciani et al., dans [luciani91] et [cadoz93], ont montré qu'un modèleur unique générique peut être réalisé si l'interaction entre particules est non seulement élastique mais plus généralement visco-élastique. L'ajout explicite dans les fonctions d'interaction d'un terme visqueux permet à la fois de régler explicitement et séparément les comportements potentiels (fonctions d'interaction élastiques) et les comportements dissipatifs. C'est pourquoi, contrairement aux autres méthodes, CORDIS-ANIMA, méthode présentée dans les deux articles précédemment cités et au Chapitre 3, n'ajoute aucun paramètre spécifique pour un type de phénomènes donnés. Il est ainsi possible de modéliser des matériaux granulaires ([manzotti95], [luciani95]), des fluides ([juneidi97], [luciani00]), des gaz et fluides turbulents ([vapillon95], [luciani95], [luciani00]), des matériaux déformables ([luciani91b]) en ne disposant que de 4 paramètres. Tous les états de la matière peuvent ainsi être représentés grâce à un modèleur unique générique.

Young et al. ([young96]) ont aussi mené des travaux en vue de la conception d'un modèleur générique. La matière est modélisée comme un ensemble de disques rigides reliés par des liaisons inélastiques (dissipatives). Le type de comportement obtenu est régi par un coefficient de restitution de la vitesse. Ils obtiennent ainsi différentes caractéristiques (un état cohésif, dispersif de la matière, des effondrements, des mouvements libres) que l'on retrouve dans les milieux granulaires, les fluides turbulents, les gaz.

Szeliski et al. ([Szeliski92]) utilisent un système de particules orientées en interaction pour simuler des surfaces élastiques soumises à des interactions avec des objets qui peuvent l'étirer, la fracturer, voire même la souder. Les particules sont soumises à des forces attractives de longue portée et à des forces répulsives de faible portée. Elles suivent les lois de la dynamique de Newton. Pour ne gérer que la surface de l'objet au lieu de masses ponctuelles, chaque particule possède, outre sa position dans l'espace et les interactions particulières, une orientation (vecteur) et un potentiel (équivalent physique du calcul du moment pour un solide) qui règle les arrangements volumiques (planaires ou sphériques). Il est possible d'introduire des particules en cours de simulation pour agrandir la surface (meilleure distribution planaire pour la visualisation). Cette technique peut modéliser bon nombre de surfaces et volumes très déformables, mais aucune tentative de modélisation de phénomènes naturels types écoulements de fluides n'a été effectuée.

L'industrie s'intéresse de plus en plus à la modélisation physique. Son application n'est pas seulement destinée à des sociétés de production de film d'animation, mais aussi à des sociétés concevant des jeux vidéo. La société irlandaise Havok ([www.havok.com](http://www.havok.com)) développe un moteur physique temps réel pour les jeux vidéo et l'animation. Ce moteur est capable de gérer des scènes comprenant des corps rigides, des corps déformables, des tissus, des systèmes de particules, des fluides, des cordes. Tous ces objets-corps peuvent interagir les uns avec les autres en temps réel. Ils ont opté pour un modèleur physique pour produire des dynamiques réalistes (que ce soit pour les collisions ou l'animation de personnages), pour que le monde virtuel dans lequel évolue le joueur soit interactif, pour que ces interactions soient réalistes, pour que les objets ne soient pas temporaires (un mur cassé, restera cassé tout au long du jeu). Le monde créé est ainsi plus riche et surtout plus proche du monde qui nous entoure et ce avec un coût de calcul et de développement peu élevé. Les objets ont une masse, une force de friction, une élasticité ; ils peuvent être pourvus de ressorts ou d'amortisseurs suivant ce que l'on cherche à représenter. Ce moteur a été utilisé pour la conception du jeu « Warcraft III », et vient d'être inclus dans la dernière version de Flash (MacroMedia). Alias/Wavefront n'est pas en reste et propose comme composant pour son logiciel Maya un moteur physique pour la modélisation de fluide.

## 1.4. Conclusion

Il existe un grand nombre de méthodes pour la modélisation de phénomènes naturels mais peu sont capables de reproduire l'ensemble des états de la matière, laissant ainsi de côté certains types de phénomènes. La modélisation physique particulière telle qu'elle est pratiquée au laboratoire ICA est à même de modéliser l'ensemble des états de la matière en modifiant un nombre restreint de paramètres (les paramètres servent à décrire les caractéristiques de la matière). Ainsi il nous est possible d'obtenir des modèles solides, gazeux, de fluides, de matériaux granulaires, mais aussi d'animer des personnages ou de modéliser du tissu.

## 2. Différentes méthodes de visualisation pour répondre à la problématique

Maintenant que nous avons indiqué la méthode de modélisation<sup>4</sup> dont nous disposons, il faut s'interroger sur la façon d'habiller (de visualiser, de rendre) un ensemble de particules. Bien entendu la technique employée pour l'habillage de croissance de cellules végétales sera probablement fort différente de celles que nous pourrions utiliser pour habiller les fluides. Avant d'explicitier les différentes méthodes et la manière dont il serait possible de les adapter à notre problème, précisons la problématique qui nous concerne.

*Comment visualiser un ensemble de points dispersés et non organisés dans l'espace (sans aucune information de topologie) dont les positions sont issues d'une simulation physique particulière représentant un phénomène naturel ? I.e. comment visualiser des points répartis à l'intérieur ou en bordure d'un modèle sans perdre d'informations sur la topologie, les caractéristiques dynamiques du phénomène imité ?*

Les points peuvent former tous types de structure, ce qui implique notamment que leurs positions relatives, les uns par rapport aux autres, évoluent constamment.

Nous distinguons les méthodes de visualisation des méthodes de rendu. Une méthode de visualisation est une représentation de données suite à un traitement d'informations : une construction de volume à partir d'un ensemble de points par exemple. Une méthode de rendu est une représentation d'un objet visualisé de manière plus poussée : on prend en compte des couleurs, des textures, des lumières. Le lancer de rayons est une méthode de rendu. Il existe aussi des méthodes traitant à la fois la visualisation et le rendu.

Nous allons répertorier différentes méthodes de visualisation et de rendu pouvant s'adapter à notre problématique. Cette liste n'est pas exhaustive et ne regroupe que les méthodes susceptibles d'être intéressantes. Nous avons décidé de ne pas nous restreindre uniquement à celles dont les chercheurs se servent pour modéliser des phénomènes, mais d'élargir nos recherches aux techniques de reconstruction de surface, de triangulation de surface.

Pour que la présentation de techniques existantes soit claire, nous les avons classées par type de représentation de la forme. Nous discutons des méthodes de rendu en même temps que certaines méthodes de visualisation. Puis nous abordons les méthodes de représentation volumique dans l'espace discret, puis les représentations par surfaces implicites, les représentations par contour puis les représentations par triangulation d'un ensemble de points.

Il est à noter que bon nombre des méthodes présentées dans cette section, traite de reconstruction de volume à partir de points de l'espace. Ces techniques ont des données initiales identiques mais des procédés d'approche fort différents. Nous appelons reconstruction le principe de visualiser une surface d'un objet à partir de données éparées, plus ou moins bruitées, réparties sur la surface à reconstruire.

---

<sup>4</sup> Cette méthode, CORDIS-ANIMA, est expliquée en détails au chapitre 3

## 2.1. Représentation de volume par kD-tree

La représentation par octree/quadtrees de solides géométriques est une façon de décrire le volume de l'objet dans l'espace discret à partir d'un nombre d'éléments de base minimal. Hirschfeld s'y est intéressé dans [hirschfeld97]. En ce qui nous concerne, nous n'avons pas de définition géométrique des corps mais il est tout à fait envisageable de se servir de ce principe de subdivision hiérarchique de l'espace pour construire un volume contenant l'ensemble de nos particules en suivant un ensemble de règles (caractérisant l'espace) de connexion-déconnexion entre sous-volumes de même niveau dans l'arbre de stockage du volume.

Artzy et al. proposent dans [artzy81] une méthode pour détecter un contour à partir d'éléments de volumes connectés entre eux. Il s'agit plus d'un suivi de contour puisque les auteurs disposent de la notion d'intérieur-extérieur. La surface est un graphe où les nœuds sont des facettes séparant les voxels (les éléments de volumes) de l'intérieur de ceux de l'extérieur.

Un autre moyen assez efficace de description de l'espace relatif à un ensemble de points, est l'application d'algorithmes « diviser pour conquérir » (« divide and conquer »). Ils permettent de définir les plus proches voisins, un partitionnement de l'espace selon leur répartition. Bentley ([bentley80]) fait un bilan des méthodes existantes et introduit leur utilisation en dimension  $n$ . Il propose des méthodes de partitionnement hiérarchique de l'espace en kD-tree (octree/quadtrees) pour accélérer les problèmes de recherches spatiales.

## 2.2. Représentation de volume par isosurface

### 2.2.1 Introduction et rendu

Les isosurfaces, ou surfaces implicites, permettent de formuler un volume en terme de fonctions implicites. Une fonction implicite est de la forme  $f(x, y, z) = c$  où  $c$  est la valeur pour laquelle le contour est considéré. Cette fonction est aussi appelée fonction de densité ou fonction potentielle. En 1982, Blinn ([blinn82]) l'a défini comme une fonction exponentielle (« blob »). Wyvill et al. ([wyvill86], « soft objects ») ainsi que Nishimura et al. ([nishimura85], « metaball ») la définissent par une fonction polynomiale pour avoir un contrôle local.

Toutes ces méthodes permettent une modélisation d'objets à partir d'un ensemble de fonctions implicites. Pour les visualiser, les techniques les plus connues sont le lancer de rayons et la polygonalisation<sup>5</sup> de la surface. La dernière technique regroupe un ensemble de méthodes. L'algorithme des « Marching Cubes » est une méthode présentée par Lorensen et Cline dans [lorensen87]. Elle est simple et rapide. Elle remplace les voxels par un ensemble de facettes triangulaires selon l'appartenance des sommets à l'isosurface. Les méthodes de pavages remplacent

---

<sup>5</sup> Nous appellerons polygonalisation le principe d'extraire d'un ensemble d'éléments de volume un ensemble de facettes (bi ou tri-dimensionnelle) rendant compte de la surface. On cherche à réaliser un canevas représentant au mieux la surface implicite.



les voxels par des éléments tétraédraux (e.g. [bloomenthal88]). La première méthode présente des configurations ambiguës ce qui fait apparaître des trous en surface des volumes construits ce que ne fait pas la seconde. Allogower et Schmidt ([allogower85]) partitionnent l'espace en cubes ou simplex tétraédraux, évaluent la fonction scalaire aux sommets des éléments de volume, puis en chacun des sommets de ces éléments approximent de façon linéaire la surface. La méthode des « Marching Cubes », lorsqu'elle est optimisée de manière adéquate peut se réaliser en temps réel ou interactif.

Dans [triquet01], les auteurs proposent une méthode de polygonalisation optimisée de surfaces implicites pour une application médicale gestuelle rétroactive. Les surfaces sont utilisées pour modéliser le sang s'écoulant dans les vaisseaux, donc l'algorithme de visualisation des surfaces se doit d'être interactive, c'est-à-dire d'afficher au moins dix images par secondes pour que l'animation restituée au médecin soit fluide, et qu'elle représente exactement l'interaction du manipulateur sur les vaisseaux. L'optimisation de la visualisation de la surface par algorithme de Lorensen et Cline se fait via une fonction de champ simplifiée, une accélération de l'évaluation de la valeur de la surface en chaque point et une résolution des trous dans la surface implicite.

Le lancer de rayons est un outil efficace de rendu de fonctions implicites. La méthode de distribution de rayons (« ray-casting ») s'occupe uniquement de déterminer les surfaces visibles, alors que le lancer de rayons va plus loin en essayant de déterminer à quoi ressemble réellement chaque surface visible. Il s'appuie sur des concepts mathématiques et physiques de la propagation de la lumière : les images ainsi produites peuvent être saisissantes de réalité. Cela permet de créer divers effets visibles difficiles à obtenir avec d'autres méthodes (la réflexion, la transparence et les ombres). Cette technique est très coûteuse en temps de calcul. Pour accélérer ce processus, le lancer de rayons discret travaille dans un espace discret avec des calculs sur des entiers. Bergad et al. dans [bergad94] et Delfosse et al. dans [delfosse94] proposent des méthodes de lancer de rayons discrets.

## 2.2.2 Visualisation

Voyons comme les chercheurs utilisent les surfaces implicites pour représenter des objets solides, des objets déformables et des phénomènes naturels. A la base, les surfaces implicites sont sans squelette. La surface est la somme de fonctions implicites ayant un centre et un rayon prédéfinis. Dans l'optique de construire des formes plus complexes et plus structurées, un certain nombre d'études ont introduit l'utilisation de squelettes.

### a. Représentation classique

La visualisation de matière capable de s'agglomérer sous l'action de la chaleur tire profit de la définition des isosurfaces. Dans [terzopoulos89], les auteurs appliquent la formulation simple et efficace de Blinn exposée dans [blinn82] et mettent plus en avant les dynamiques de leurs matériaux déformables en rendant les sphères implicites par lancer de rayons. Tonnesen fait de même dans [tonnesen91] pour son modèle de liquide et de solide basé sur un système de particules sensibles à la chaleur.

En 1995, Chiba et al. dans [chiba95] s'en servent pour visualiser leur modèle d'eau simulée en espace discret. Les auteurs ont préféré un découpage tétraédral plutôt que le classique triangulaire.

Pour leur modèle de neige ([nishita97]), Nishita et al. développent une méthode de visualisation basée sur des isosurfaces hiérarchiques. Ainsi ils peuvent décomposer la neige en cristaux de glace et en cristaux d'air telle qu'elle existe dans la nature. Une telle décomposition fournit donc une surface qu'il faut ensuite rendre de manière réaliste. Les auteurs ont donc mis au point un modèle d'illumination calculant la quantité de lumière renvoyée suivant l'épaisseur, la qualité, la densité de la neige.

Pour la première fois, les surfaces implicites sont impliquées dans une méthode de visualisation d'objets à contour flou. Dans [stam97], Stam représente le gaz comme une superposition de surfaces implicites floues (« fuzzy blobs ») sphériques définies par une masse et une taille. Si l'on souhaite un rendu temps réel pour juger de la qualité de la simulation, on se servira d'artefacts qui approximent de manière polygonale la projection sur le plan de vue des sphères implicites (« splats »), alors que si l'on souhaite une animation réaliste, on se servira d'un rendu par lancer de rayons.

Pour visualiser ces phénomènes naturels simulés par système particulaire, Gareau ([gareau97]) a recours aux isosurfaces. Il discrétise la matière, calcule la lumière directe émise et les rayons indirects. Cependant, il ne prend pas en compte la densité du phénomène.

DaDalto ([dadalto97]) ayant utilisé les surfaces implicites telles que Blinn les a définies pour la modélisation de phénomènes naturels, la visualisation coulait de source. Pour une animation plus réaliste, il les rend par radiosité ou lancer de rayons. Ces deux méthodes ont dû s'attaquer à l'inconvénient majeur des surfaces implicites : le *mélange*<sup>6</sup> non désiré.

Comme nous l'avons indiqué dans la section précédente, nombreuses sont les méthodes de modélisation-visualisation utilisant les surfaces implicites. L'intérêt d'une telle technique est un contrôle de la forme au cours de la simulation, donc la forme se nourrit des dynamiques internes pour évoluer de manière naturelle. Les auteurs de [stora98] et [gascuel\_jd98] ont mis à profit ces qualités. Dans [desbrun96], la surface des objets est une isosurface d'une fonction masse ponctuelles-densité de masse. Pour aboutir à une surface plus convaincante, les auteurs de [stora98] ont bruité la surface pour reproduire la rugosité de la lave au contact de l'air, et ont utilisé des facettes texturées.

Après Chiba en 1995, Fearing modélise de la neige ([fearing00]), et dès que les particules sont correctement placées (une fois la simulation terminée), met à profit les propriétés des surfaces implicites pour obtenir une enveloppe lisse, rendant les différences de hauteur dues à l'accumulation de matière à divers endroits de la scène.

La visualisation de nuages se sert elle aussi des surfaces implicites. Pour que le rendu soit réaliste, les auteurs de [dobashi00] se servent de la distribution de la densité de matière calculée lors de l'animation, appliquent des textures 2D et travaillent directement sur le tampon image.

La dernière étude de Foster sur les liquides ([foster01]) utilise une méthode de génération de contour à partir de fonctions implicites. Chaque particule représente le centre d'un objet volumique géométrique, une fonction implicite. La surface résultante doit être lissée spatialement et temporellement. Pour cela on applique plusieurs fois une fonction pour réduire les coins en modifiant la valeur de la fonction implicite  $\phi$  proche de l'isocontour. Cette technique n'est valable que si le

---

<sup>6</sup> Appelé aussi fusion entre primitives implicites (blend)

nombre de particules en surface est assez important. Comme on ne peut pas en placer énormément pour des raisons de temps de calcul, on crée  $\phi$  à partir des particules dont on dispose puis on l'étire grâce aux champs de vitesses qui nous indiquent les mouvements de particules. Les volumes ainsi restitués par lancer de rayons pour obtenir une visualisation impressionnante de réalisme.

## b. Représentation par squelettes

Dans [muraki91], l'auteur emploie les surfaces implicites pour reconstruire un objet 3D à partir de données bidimensionnelles issues de scanner. Dans un premier temps, il applique un filtre gaussien pour obtenir moins de points et la normale en chacun de ces points. De là, il cherche à créer un ensemble de primitives (squelettes des surfaces implicites) possédant des coefficients susceptibles de minimiser la fonction suivante :  $E = E_{value} + qE_{normale} + E_{shrink}$  où  $E_{value}$  est l'énergie de la fonction implicite,  $E_{normale}$  l'énergie de la normale,  $E_{shrink}$  une énergie ajoutée pour ne pas incorporer d'incohérences dans l'image finale. On part d'une primitive satisfaisant l'équation, puis on la divise en deux parties et on cherche de nouveaux coefficients satisfaisants. On réitère le processus jusqu'à l'obtention d'une énergie totale suffisamment minimale. Plus le nombre de primitives augmente, plus la forme devient complexe, et le coût de calcul important. La minimisation de la fonction d'énergie permet d'aboutir à une surface qui colle au mieux aux données initiales, une propagation et un mélange des couleurs lors de la fusion des primitives, mais l'information de luminosité - indispensable au calcul des normales - présente dans les données initiales nous est étrangère. Cependant l'usage une fois de plus des surfaces implicites pour reconstruire des volumes, nous montre que cette technique peut s'adapter à la visualisation de nombre d'objets. Il est à noter que cette méthode fait penser à celles des contours actifs (« snakes ») présentée par la suite.

L'article [bittar95] présente une méthode de reconstruction de volume à partir d'un ensemble de points répartis à la surface de l'objet. L'algorithme calcule un axe médian<sup>7</sup>, et réduit le nombre de sphères maximales de cet axe en déterminant si les sphères choisies atteignent bien tous les points initiaux, puis, à partir des sphères maximales précédemment choisies, crée des surfaces implicites. Grâce à cette méthode il est possible de reconstruire des objets complexes (avec branches ou cavités), mais la résolution du volume final est fonction de l'objet à reconstruire : si la résolution est trop élevée, les voxels en bordure de volume ne seront pas connectés, ainsi la surface reconstruite ne serait pas celle de l'objet initial. Les travaux ont été poursuivis sur ce sujet dans le cadre d'une thèse [bittar98].

Dans [ferley96], une nouvelle approche de construction par surfaces implicites est abordée. Les auteurs extraient un squelette géométrique à partir d'un diagramme de Voronoï où chaque point est le germe d'une région du dit diagramme. Pour ne pas façonner une surface angulaire, les arêtes du graphe sont transformées en « B-Spline ». Ces splines servent ensuite à la conception de squelettes de surfaces implicites. Pour éviter que plusieurs squelettes ne participent pas à la construction de isosurface en un point formant ainsi un renflement inapproprié (« bulge »), les auteurs ont recours à des patches de Bézier. Cette méthode prend en compte la topologie et la géométrie de la forme à reconstruire, mais la surface résultante contient du bruit généré par les positions imprécises de points initiaux.

---

<sup>7</sup> Positions des centres de chaque sphère maximale de l'objet auquel on a associé un rayon. Une sphère maximale est une sphère du squelette S qui n'est pas incluse dans une autre sphère de S.

Pour l'animation d'objets déformables, M. Desbrun choisit dans sa thèse ([desbrun97]), comme l'avait proposé Jimenez ([jimenez93]) d'utiliser des modèles tri-couches composés d'une ossature, d'une chair, et d'une peau. L'ossature définit les mouvements globaux, la chair les fortes déformations et la peau la surface de l'objet. Tous ces composants interagissent entre eux. La peau, la surface visible, est définie par une surface implicite apportant ses avantages et inconvénients. Côté avantages, on a une surface lisse, douce à l'aspect assez naturel. Côté inconvénients, il faut être en mesure de conserver une discrétisation correcte même lorsque l'objet se déforme fortement, de prévenir les mélanges indésirables entre portions de matières. Des solutions ont été proposées pour résoudre une partie de ces inconvénients.

L'animation de personnage dont le corps est formé de fonctions implicites, se contrôle de manière assez simple. Cette simplicité n'est obtenue que si le corps est formé de squelettes (bases des surfaces implicites) et qu'un graphe de voisinage est utilisé pour gérer les fusions de surface non voulues ([gascuel\_mp97]). La même année, Galin a travaillé sur la métamorphose de « blobs » à squelettes rendus par lancer de rayons ([galin97]).

Nous avons discuté, il y a peu, de méthodes de reconstruction par surfaces implicites. Ces méthodes ont des limites lorsqu'il s'agit de synthétiser des formes ne pouvant être définies de manière implicite. Pour obtenir d'autres formes, comme le cylindre ou le plan sans irrégularités en leur surface, Rigaudière et al. ([rigaudière00]) se sont intéressés à de nouvelles fonctions implicites. Ils redéfinissent les opérateurs de mélange (union et intersection) définis dans [pasko94], et décrivent de nouvelles primitives implicites pouvant habiller les squelettes : pour habiller un segment par un cylindre (représenté de manière implicite), on utilise deux fonctions de densité l'une influençant le cylindre longitudinalement, l'autre dans le plan de définition du cylindre. Ces nouvelles primitives permettent la modélisation d'une large gamme de formes.

[zhao01] expose une méthode de reconstruction de forme par surfaces implicites à partir d'un ensemble de données dispersées (points, courbes, surfaces) en utilisant la méthode de courbe de niveaux (« level set »). Dans un premier temps, une pseudo-surface pondérée est générée, puis est calculée sa courbe de niveaux de manière optimale. La courbe de niveaux sert à la déformation de surface implicite à l'aide d'équations aux dérivées partielles et d'une formulation variationnelle. Cette méthode efficace pour la reconstruction de données ne peut nous servir telle quelle puisque pour toutes méthodes de reconstruction, les points, courbes sont en surface de volume et placés aux endroits où la surface se modifie le plus, à la manière des courbes de niveaux pour les cartes géographiques.

### c. Représentation par F-Rep

Plus généralement, les fonctions implicites appartiennent à la représentation de type F-Rep (représentation de fonctions définissant un objet géométrique par une fonction réelle continue de plusieurs variables) qui permettent la modélisation d'objets géométriques complexes. On ne peut définir une sphère par une fonction implicite car cela impliquerait que pour une fonction  $f(x, y, z) = 0$  il existe une unique valeur de  $z$  pour chaque couple  $(x, y)$ . Mais il est possible d'obtenir une sphère si l'on considère un point et un rayon, et non un ensemble de points. La modélisation par F-Rep fait l'objet d'un logiciel HyperFun décrit dans [adzhiev99] et [pasko00]. C'est un langage et un outil de modélisation géométrique basé sur les R-fonctions, incluant plusieurs types de modèles

comme les fonctions implicites classiques, les fonctions implicites à squelettes, les solides définis par opérations d'assemblages.

### 2.3. Les « snakes » : un modèle de contour actif

Le modèle de contour déformable adaptatif, appelé « snake », a été développé par Terzopoulos et al. dans [terzopoulos88]. Le contour actif (« snake »), objet de référence, est constitué de masses, reliées entre elles par des interactions élastiques, formant ainsi un squelette. La forme initiale de l'objet (son anatomie) est prise lorsque la structure interne est au repos. Cette anatomie se déforme suivant les déplacements du squelette. Il s'agit à chaque instant de minimiser une fonction d'énergie pour que le contour actif colle au mieux au contour désiré. Cette méthode a aussi été utilisée dans travaux de Jimenez ([jimenez93]) et Habibi ([habibi91]).

La méthode originale du modèle de contour actif présente une instabilité numérique, que Berger dans [berger90] résout en introduisant l'idée que le contour actif grandit : il peut se diviser, ces morceaux de faible énergie peuvent grandir selon la direction de leur tangente tandis que ceux de forte énergie meurent. Après chaque pas de croissance, l'énergie de chaque morceau est minimisée. Cette méthode itérative de croissance du contour actif (« snake growing ») converge toujours, donc la solution numérique est stable, mais la robustesse de cet algorithme à un coût : le nombre de calculs.

Dans [miller91], les auteurs proposent une nouvelle approche au problème de génération de modèle géométrique topologiquement proche d'un ensemble de données échantillonnées. On part d'une graine et on la déforme selon un ensemble de contraintes jusqu'à englober les points. Cela revient à introduire un ballon au centre des points initiaux et à le gonfler jusqu'à ce que sa membrane se bloque sur ces points. C'est une méthode hybride des contours actifs.

Flickner et al. utilisent une formulation de modèle de contour actif en terme de spline paramétrique pour délimiter des formes et des régions dans des images bidimensionnelles ([flickner94]). Comme dans la version originale, ils cherchent les points en superficie de la forme en minimisant une fonction d'énergie, mais doivent ensuite effectuées un traitement pour définir les points de contrôle de la spline. McInerney et al. dans [mcinerney95] manipulent, dans un espace discret, un ensemble de contours actifs au lieu d'un contour actif unique pour coller au mieux au contour recherche.

La méthode des contours actifs est intuitive dans la recherche de contour de forme. Sa flexibilité et son adaptation à toutes sortes de topologie font qu'elle est capable de définir les formes les plus complexes. Cependant la convergence de la fonction d'énergie dans les régions non concaves n'est pas satisfaisante. La qualité du résultat dépend fortement du positionnement initial du contour actif (de sa forme de départ). Elle est de plus inadaptée à des changements de topologie importants et imprédictibles. C'est pourquoi Jimenez ([jimenez93]) et Habibi ([habibi97]) ont renoncé à le généraliser. Cependant, pour tenter de résoudre ces inconvénients, Xu et Price ([xu98]) introduisent une force externe.

Dans la même veine, le travail d'Olivier Lachaud au cours de sa thèse ([lachaud98]) a porté sur l'utilisation de techniques d'extraction de surface par modèle discret ou déformable à partir

d'images en niveaux de gris. Il a tout d'abord étudié l'extraction d'un modèle géométrique (une surface triangulée, une représentation par contour - B-rep - par exemple), mais ce procédé s'adapte peu à des objets complexes en perpétuelle évolution. La reconstruction discrète regroupe un ensemble de techniques, comme l'extraction isosurface, l'utilisation des contours actifs, plutôt rapide pour la reproduction de modèles à partir de données volumétriques. Cependant toutes ces études ne se sont pas faits pour des données dynamiques, mais il a permis de mettre en avant certaines méthodes de reconstruction susceptible de répondre à notre problématique. En effet la méthode des contours actifs, mise au point par Terzopoulos et al. [terzopoulos88], parvient à rendre une enveloppe non convexe d'un ensemble de points dispersés de l'espace sans topologie prédéfinie. Cependant la plupart des études à ce sujet sont appliquées à des images bidimensionnelles en niveaux de gris.

## 2.4. Représentation de volume par triangulation

Nous venons de décrire deux grandes catégories de visualisation de contour. Elles ont leurs inconvénients et leurs avantages. Qu'en est-il des méthodes de triangulation de points ? Appportent-elles des résultats susceptibles de nous convenir. C'est ce que nous allons voir.

### 2.4.1 L'emballage de formes (« gift-wrapping »)

Dans sa thèse ([favardin93]), Favardin cherche un moyen de déterminer automatiquement une structure géométrique (une triangulation) pour un ensemble de points quelconques. Elle propose une version modifiée de l'algorithme d'emballage de formes (« gift-wrapping »), qui construit une enveloppe convexe. En ne considérant que des points dans le voisinage local d'une arête pivot<sup>8</sup>, elle obtient des surfaces non convexes.

### 2.4.2 Les Graphes de Delaunay et Voronoï

Le diagramme de Delaunay et son dual, le diagramme de Voronoï, sont des algorithmes de construction de graphes géométriques parmi les plus connus. Ils décrivent l'espace recouvert par un ensemble de points et servent de base à bon nombre de méthodes de définition de surface en fonction des plus proches voisins. Le diagramme de Voronoï partitionne l'espace en régions d'influence.

Dès 1991, Barry Joe dans [joe91] propose une méthode de construction du diagramme de Delaunay en trois dimensions en utilisant des transformations locales et un découpage tétraédral de l'espace.

Etant donné un ensemble de points, le diagramme de Voronoï correspondant divise le plan en régions ; A chaque point est associé une région du plan plus proche de lui que de n'importe quel autre point. Pour étendre cette méthode à la troisième dimension, il faut que les droites divisant les régions deviennent des plans, tandis que les plans définis par les droites (les régions) deviennent des volumes. Aurenhammer dans [aurenhammer91] fait un bilan de l'utilisation de cette méthode ainsi qu'une unification de ses propriétés mathématiques et géométriques.

---

<sup>8</sup> A partir d'un point de position minimal **A**, on cherche un point **B** tel que tous les points sauf **A** et **B**, se trouve à gauche de l'arête [**AB**]. L'arête [**AB**] est dit arête pivot.

Les méthodes de reconstruction de formes ne s'appliquent pas uniquement à des données éparpillées en surface de l'objet. Nullans a dans sa thèse ([nullan98]) travaillé sur la reconstruction de structures géologiques à partir de données incomplètes et hétérogènes. Son idée est d'assembler les données selon leurs diagrammes de Voronoï. Il dispose d'un paramètre supplémentaire indiquant pour chaque portion de données les caractéristiques géologiques. Cette information, non négligeable, donne une solution première au problème puisqu'elle identifie et caractérise les bords et l'intérieur de la couche de sol en études.

### 2.4.3 Les « Alpha shapes »

Veltkamp ([veltkamp92\_thesis]) pense que les graphes géométriques décrivant les relations entre points sont des outils utiles pour la reconstruction-construction puisqu'ils décrivent l'intérieur et l'extérieur de la forme. Cependant certains graphes géométriques, comme le diagramme de Voronoï ou de Delaunay, construisent des enveloppes convexes. Plusieurs algorithmes de construction d'enveloppes non convexes à partir des graphes de Delaunay et de son dual Voronoï ont été mis au point, parmi eux les « alpha shapes », sous-graphe de la triangulation de Delaunay.

Ce sous-graphe de Delaunay a été défini en 1983 par Edelsbrunner dans [edelsbrunner83], et généraliser en 3D dans [edelsbrunner94]. Leur concept est une approche pour formaliser la notion intuitive de forme pour un ensemble de points de l'espace. La forme (« alpha shape ») d'un ensemble de points  $S$  est le polytope non nécessairement convexe ou connecté pour lequel  $\alpha$  contrôle le niveau de détails. Il s'agit donc de l'ensemble des simplex (des segments en 2D) pour lesquels il existe un cercle (une sphère en 3D) de rayon  $\alpha$  dont les deux points définissant le simplex sont supportés par le périmètre de ce cercle et tel que, à l'intérieur de ce cercle ne se trouve aucun autre point de  $S$ . Les données initiales doivent être non bruitées et représenter une surface suffisamment lisse.

Guo et al. dans [guo97] propose une technique de reconstruction d'une surface inconnue à partir d'un ensemble de données sur la surface de l'objet. L'approche est la suivante : extraire la surface, considérée comme un maillage, à partir de la méthode que l'on vient de présenter, puis simplifier ce maillage. Comme toute méthode y ayant recours, les surfaces reconstruites s'appliquent correctement aux données initiales, mais il demeure toujours le problème du choix de la valeur de  $\alpha$ .

Pour reconstruire des données issues de scanners, Bernardini et al. proposent une méthode de conception simple issue de la construction de graphes géométriques : le « ball-pivoting ». La technique fait rouler une balle sur la surface des points pour générer un maillage de l'objet ([bernardini99]). Le rayon de la balle est inférieur à la moitié du rayon de courbure maximale de la surface. La technique se synthétise ainsi : à partir de trois points (un triangle), on fait tourner la balle par rapport à une des arêtes déjà créées, jusqu'à ce qu'elle rencontre un point. Là, on forme un nouveau triangle, puis on recommence avec une arête non encore visitée jusqu'à ce que tous les points appartiennent au maillage. Le principal inconvénient d'une telle méthode se situe dans le choix du rayon de la balle : si il est trop grand, on perd de l'informations, si il est trop petit, le maillage comportera des trous. Côté avantage, c'est un algorithme avec un temps de calcul linéaire, possédant les propriétés des sous-graphes de Delaunay, et dont les données initiales sont à la surface de l'objet à reconstruire, mais il ne peut rendre de manière appropriée et convenable des formes complexes aux rayons de courbure forts variés.

#### 2.4.4 Le $\Gamma$ -graphe

L'article [veltkamp90] introduit le  $\gamma$ -graphe. Il s'agit d'un assemblage de plusieurs graphes bien connus comme l'arbre de recouvrement minimum (« spanning tree »), la triangulation de Delaunay, l'enveloppe convexe et le graphe de Gabriel. On ressort une enveloppe convexe d'un ensemble de points jusqu'à ce que les bords du  $\gamma$ -graphe soient une surface fermée passant par tous ces points.

#### 2.4.5 Les autres Triangulations

Dans [veltkamp92\_article], l'auteur cherche à connaître le degré nécessaire et suffisant d'un polynôme pour interpoler des données dispersées de l'espace. Une fois cette analyse réalisée, il construit une surface cubique à l'aide de patches de Bézier triangulaire pour parvenir à une surface  $G^1$ -continue. Afin d'obtenir suffisamment de degrés de liberté pour définir les points de contrôles de la surface de Bézier, des méthodes de division des triangles en 2, 3 et 6 morceaux ont été développées. Cette méthode est fastidieuse pour le problème qui nous concerne car même si nous arrivions à placer nos points sur la surface du volume, le degré du polynôme devrait être élevé pour que sa surface ce calque sur celle de la forme.

Comme dans [muraki91], [szeliski92] exploite des informations d'orientation. En effet, la modélisation par particules orientées donne, de part sa conception, une idée précise de la forme de l'objet que l'on cherche à visualiser. Par interaction locale, ces particules s'alignent elles-mêmes sur un manifold 2D. L'orientation initiale des particules est cruciale pour l'efficacité de cet algorithme.

Hoppe propose dans [hoppe94], une méthode pour reconstruire automatiquement une surface lisse, précise et concise à partir d'un ensemble de points  $\mathbf{X}$  où  $\mathbf{X}$  est un échantillonnage bruité et non organisé d'une surface inconnue  $\mathbf{U}$ . La surface inconnue  $\mathbf{U}$  peut avoir n'importe quelle topologie, et aucune information de structures ou d'orientation n'est fournie. Sa méthode se décompose en trois parties : dans un premier temps, il estime la surface initiale, puis il optimise le maillage précédemment obtenu, et lisse la surface par morceaux. Les résultats obtenus montrent que cette méthode est un outil robuste de reconstruction mais elle ne peut s'adapter à la construction de surface lorsque les points ne sont pas situés en sa surface. La première étape se base sur une construction d'un voisinage pour  $\mathbf{X}$ . Il minimise une fonction d'énergie par ajustements successifs à partir d'un maillage triangulaire jusqu'à l'obtention d'une surface concise et précise.

Le problème des méthodes de triangulation d'un ensemble de points est que le niveau de détail de la surface finale est fortement dépendant de la répartition des points dans l'espace ainsi que de leur nombre. Ces méthodes ne peuvent nous servir sans un prétraitement des données pour les placer de manière satisfaisante pour ne pas que le maillage résultant suggère autre chose que ce qu'il représente.

### 2.5. Représentation de formes à contour mal définis

Certains phénomènes sont dits à contour flou, c'est-à-dire qu'on ne peut tracer leurs formes avec précisions.



Dans sa méthode pour l'animation par système particulaire d'objets à contour flou présentée dans [sims90], Sims attribue à chaque particule une forme, une taille, une couleur, une transparence qui lui est propre.

Stam dans [stam93] se sert d'une méthode similaire pour visualiser ses fluides turbulents. Le rendu se sert des propriétés de transparence du phénomène, et tire profit d'une technique d'éclairage basée sur les lois physiques.

Il n'a pas été le seul à baser son rendu de particules sur la transparence de celles-ci. Reeves dans [reeves93] a fait de même, mais a choisi d'appliquer en plus un flou de mouvement (« motion blur ») dessus.

Dans [fedkiw01], les auteurs réalisent la visualisation de fumée en même temps que leur animation. Ce couplage des deux étapes permet l'utilisation de paramètres de la simulation pour la phase de rendu. Cette dernière pour être réaliste et en adéquation avec le modèle manipule des photons, des données de densité. A chaque pas de calcul, on détermine la quantité de lumière qui atteint chaque voxel de la grille de rendu en se basant sur la grille de densité, un coefficient d'extinction relatif à la densité de fumée, et le devenir des photons lancés (absorbés ou dispersés). Les phénomènes ainsi visualisés sont très réalistes et nécessitent un coût de calcul peu élevé (cette méthode de modélisation-visualisation de fumées fonctionnent sur des PDA).

## 2.6. L'écran d'épingles

L'écran d'épingles a été inventé vers le début des années 1930 par Alexandre Alexeïeff et Claire Parker. Il s'agit d'une plaque de 900mm sur 120mm composée de 500 000 cavités disposées, sur une surface blanche, en quinconce dans lesquels sont fixées des aiguilles noires dont la tête mesure 5mm. Lorsque cet écran est éclairé par un projecteur, chaque épingle saillante produit une ombre sur la surface de l'écran. L'œil étant incapable de distinguer l'aiguille noire de l'ombre sombre, il est possible d'obtenir toutes les nuances de gris entre le blanc et le noir : plus les épingles sont saillantes, plus la surface que leurs ombres couvrent est grande et moins la lumière ne vient frapper la surface blanche de l'écran. Les aiguilles peuvent être vues comme les pixels d'un écran.

Annie Luciani et Arash Habibi sont partis de ce principe pour élaborer une version informatique de l'écran d'épingles. Dans [luciani95] il a servi pour la visualisation de matériaux granulaires. Arash Habibi ([habibi97]) a montré que par ce procédé, on simulait en fait un procédé très générique de gravure et l'a ainsi utilisé pour visualiser n'importe quel phénomène modélisé à l'aide de particules. Pour cela, la surface blanche de l'écran est remplacée par un support, dont on choisit la nature.

L'écran d'épingle informatique est constitué d'une grille de masses ponctuelles reliées par interactions visco-élastique à chacune de ses voisines, et à un point ponctuel fixe placé à une hauteur inférieure. Les points de la grille sont appelés phyxels (« physical elements »). Chaque masse ponctuelle du modèle possède une zone d'influence qui agit sur les phyxels compris dans cette zone. Cette influence se traduit par un déplacement de l'ensemble des phyxels de la zone d'influence (ces déplacements servent ensuite au calcul de la couleur du pixel dans l'image finale). Par ce système de

ressorts liés entre eux, on obtient une rémanence de la dynamique du phénomène. Cette méthode n'est toute fois applicable, pour le moment, qu'à des ensemble de points de  $\mathcal{R}^2$

## 2.7. Conclusion

Ce que l'on retire de cette section est le fait que le type de phénomène à modéliser influence le type de méthode à utiliser, et que le type de technique de visualisation dépend du type de données dont on dispose. Aucune des méthodes que nous avons présentées ne traite de la visualisation de données parfaitement désordonnées, sans indication de topologie, d'orientation et de position par rapport à la forme à obtenir. Nous allons discuter dans la section suivante des méthodes susceptibles de convenir à nos exigences.

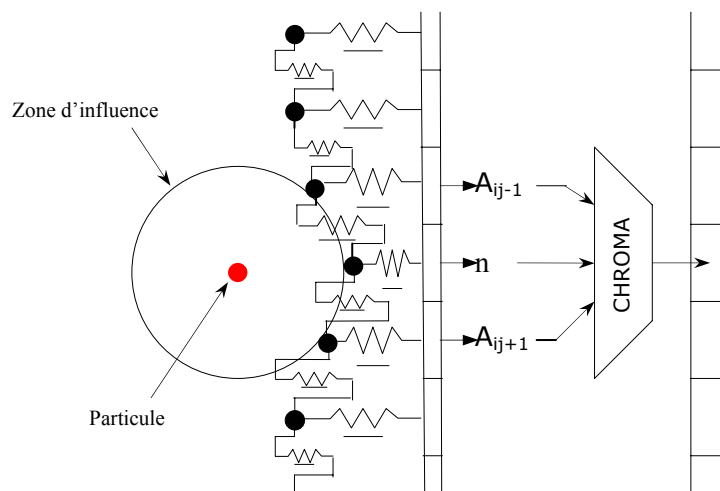


Figure 1 : Principe de l'écran d'épingles

## 3. Discussion

Si l'on considère les méthodes de modélisation de phénomènes naturels, d'objets dans leur globalité, on remarque que simulation et visualisation vont de pair. Ces deux étapes de l'animation se font en parallèle pour tirer profit des données dynamiques, informations supplémentaires, issues de la simulation en vue de les insérer dans la construction du volume pour être à même de donner des résultats plus que satisfaisants.

Notre démarche se démarque totalement de cette unification modélisation-visualisation dans la mesure où nous les séparerons de manière drastique. La modélisation des phénomènes choisis se fera par modèle physique particulière en utilisant la librairie CORDIS-ANIMA présentée au chapitre suivant. La simulation est l'ensemble de positions des particules au cours du temps. Le jugement de la qualité des dynamiques reproduites se base sur une évolution des particules dans l'espace physique. Il faut donc que l'utilisateur mette à profit son imagination pour voir un volume, reconnaître le phénomène par ces caractéristiques dynamiques.

Les positions des particules au cours du temps sont les seules informations dont on disposera pour la visualisation. Il faut donc essayer d'en tirer des informations susceptibles de caractériser le volume qui les englobe sans masquer les formes complexes résultant des dynamiques internes.

Les méthodes favorisant les relations de voisinage pour créer l'enveloppe de la forme ne sont pas aptes à modéliser les contours sinueux qui nous intéressent. Nous verrons au chapitre 4, que lorsque un fluide se replie sur lui-même la distance entre les deux portions de fluide est à peine plus grande que la distance entre particules de la même portion. Donc des algorithmes de triangulation ne peuvent nous convenir.

Les méthodes recherchant un contour par adaptation successive telles que les contours actifs, que nos prédécesseurs Jimenez et Habibi ont cherché à utiliser au laboratoire, posent des problèmes d'initialisation des données et d'adaptation des données à celles classiquement utilisées. De plus, une telle technique n'a pas lieu d'être si elle ne se réalise pas en même temps que la simulation pour qu'entre deux images le volume construit soit cohérent.

La construction d'un volume par surfaces implicites ou la construction d'un volume dans l'espace discret sont deux méthodes candidates pour la résolution de notre problématique de visualisation. Le travail dans l'espace discret permet une liberté de construction qui nous amènera à saisir les formes les plus complexes de notre phénomène, mais nous nous heurterons probablement à une non-cohérence volumique au cours du temps. Les surfaces implicites ont montré leur souplesse, leur robustesse à la représentation des objets les plus divers, et comme Bloomenthal l'a énoncé dans le livre édité chez Morgan Kaufman intitulé **Introduction to implicit surfaces**, « Les surfaces implicites à squelettes (...) sont particulièrement bien adaptées à la gestion des changements de topologie qui peuvent survenir au cours de l'animation ». Nous allons donc vérifier ses dires en procédant à la mise en place d'une méthode visualisant par surfaces implicites incluant un traitement sur les données dont nous disposons. Le rendu du volume défini par isosurface pourra être réalisé soit par lancer de rayons, soit en utilisant une méthode de polygonalisation.

**Partie II - MODELISATION PHYSIQUE  
DE PHENOMENES NATURELS**



# Chapitre 3 CORDIS-ANIMA, un modelleur-simulateur

CORDIS-ANIMA est un système de modélisation et de simulation numérique d'objets physiques visibles, audibles et manipulables ([cadoz93]). Un objet physique n'est ni une image, ni un son, ni aucun autre phénomène accessible par nos sens, mais une structure qui soumise à des actions (forces, couples, ...) peut être source d'image, de mouvement, de son.

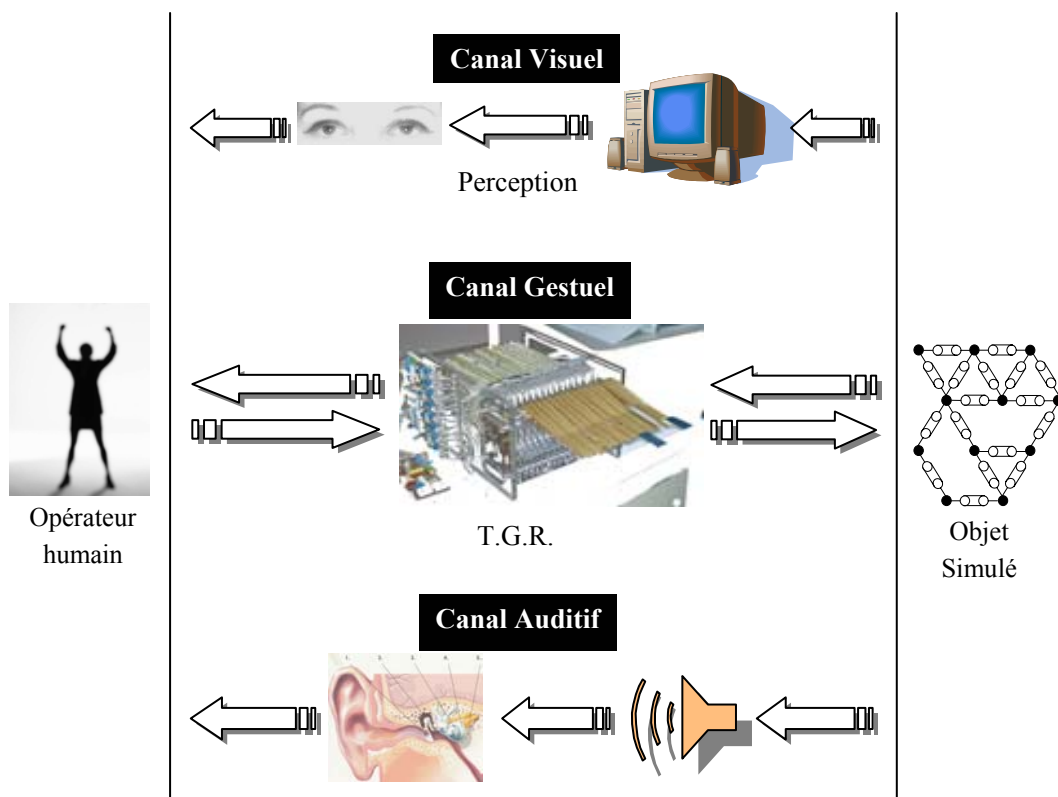


Figure 2 : Schématisation de CORDIS-ANIMA

La réplique de ces objets est réalisée à l'aide d'une machine numérique capable de communiquer avec l'opérateur humain par l'intermédiaire de trois canaux sensoriels : visuel, acoustique et gestuel. L'opérateur est alors placé dans une situation expérimentale simulée. Il agit sur l'objet, reçoit en retour sa réaction physique (proprioceptive, kinesthésique), le voit et l'entend. Le canal gestuel de la Figure 2 est bidirectionnel (par l'intermédiaire du Transducteur Gestuel Rétroactif - TGR) et les canaux visuel et auditif sont unidirectionnels.

CORDIS-ANIMA définit un ensemble d'éléments et de règles de combinaison qui forment un langage. Toutefois, en plus de sa fonction descriptive des objets du monde physique, il permet, dans l'espace virtuel, la construction d'objets qui n'ont pas d'équivalents réels. CORDIS-ANIMA acquiert le statut d'outil de création.

## 1. Le formalisme CORDIS-ANIMA

Deux types de modules composent CORDIS-ANIMA ([Luciani91]):

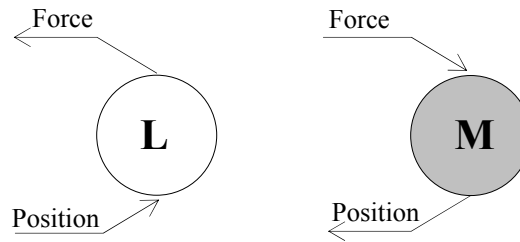
- ❑ **les modules physiques** : les entrées et sorties sont des forces et des déplacements. Les sorties sont élaborées à partir des entrées selon un calcul spécifique prenant modèle sur un processus physique élémentaire ;
- ❑ **les modules fonctionnels** : ils ne manipulent pas exclusivement des variables physiques et ne répondent pas nécessairement à un modèle physique immédiat, mais, utilisés rationnellement, étendent les possibilités de représentation. Les modules de variation structurelle dynamique ou de variation paramétrique dynamique sont des modules fonctionnels.

Nous ne développerons ici que les modules physiques.

### 1.1. Les modules physiques

La simulation d'un objet physique par ordinateur s'effectue au moyen d'un processus de calcul numérique. Cette technologie implique que la communication physique entre un opérateur et un objet simulé soit discrète et de dimension finie. Il est donc nécessaire d'adopter une représentation idéalisée de la communication physique entre objets sous forme d'une communication mécanique ponctuelle. De plus, l'interaction entre deux objets physiques est essentiellement non orientée, et fait intervenir deux variables duales : l'une intensive, la force, l'autre extensive, la position.

Aussi, si l'on appelle point de communication, pour l'objet ou pour l'opérateur, la paire d'entrée-sortie qui réalise la communication, il existe deux types de points de communication (Figure 3).



**Figure 3 : Point L (liaison) et point M (matériel)**

- **Les points M** : (matériel) admettent en entrée la variable force et génèrent en sortie la variable position ;
- **Les points L** : (liaison) au contraire, admettent en entrée la variable position et génèrent en sortie la variable force.

Ces points d'entrée-sortie permettent via un algorithme de calculer la variable de sortie en fonction de celle d'entrée. Les algorithmes des éléments linéaires se basent tous sur les équations différentielles de la mécanique classique. La masse  $M$ , la constante de raideur  $K$  et la constante de viscosité  $Z$  sont introduites sous forme de paramètres ( $x$  est la position d'un point).

$$Mx''(t) + Zx'(t) + Kx(t) = \sum \overrightarrow{F_{ext}}$$

## 1.2. Règles de construction

La construction d'un objet CORDIS-ANIMA revient à établir des connexions entre les points de communications des différents modules qui le composent.

- ❑ La connexion entre deux points consiste à utiliser la sortie de l'un comme entrée de l'autre et réciproquement. Par conséquent, seuls des points de type opposés sont connectables.
- ❑ Il est possible de connecter plusieurs points L à un même point M : la force appliquée à l'entrée du point M sera la résultante des forces provenant des sorties des différents points L auxquels il est connecté.
- ❑ Il est interdit de connecter plusieurs points M à un même point L. Ceci est dû au caractère extensif de la variable position : calculer la résultante de plusieurs positions n'a pas de sens. L'entrée de position d'un point de type L ne peut provenir de plusieurs positions de points M différents, il y aurait autrement un conflit de position.

Pour pouvoir construire la plus grande variété d'objets possibles, la base topologique retenue est constituée de deux types d'atomes : le premier est constitué de deux points L, l'autre est un point M.

L'atome à un point M, nommé **MAT**, a une localisation ponctuelle dans l'espace puisque son algorithme calcule des positions au cours du temps à partir de forces appliquées par son environnement. Il constitue alors la granulation de la matière. Nous l'appelons point matériel.



L'atome à deux points L, nommé **LIA**, n'a de localisation dans l'espace que via des points auxquels il est connecté. Si on calcule deux forces à destination de ces points à partir de leur position et variation de position, une interaction est établie entre les points MAT et LIA. Nous l'appelons élément de liaison. Les deux forces calculées par l'élément de liaison sont coaxiales, égales en intensité et de sens opposé, afin de respecter le principe d'action-réaction ([**cadoz90**]) :

$$\vec{F}_1 = -\vec{F}_2 = f(\vec{X}_1, \vec{X}_2)$$

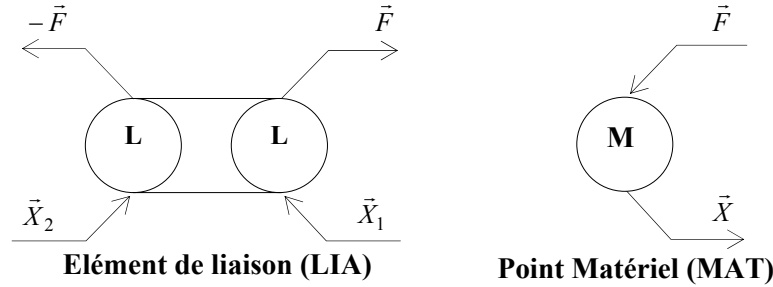


Figure 4 : MAT et LIA

### 1.3. Les algorithmes de CORDIS-ANIMA

Pour présenter toutes les valeurs discrètes des signaux d'entrées et de sorties, CORDIS-ANIMA adopte la notion de suite récurrente. Ainsi  $X_n$  et  $F_n$  désignent respectivement les variables position et force à l'instant  $n$ .

Par convention, l'unité de temps interne à CORDIS-ANIMA est égale à la période d'échantillonnage. La fréquence d'échantillonnage prend des valeurs différentes suivant que les applications sont dans le domaine des images animées ( $f_e=1050Hz$ , valeur obtenue expérimentalement à partir de l'ensemble des simulations précédentes de l'ACROE - ICA, et sous-multiples de certains standards) ou de la synthèse sonore ( $f_e=44,1 kHz$ , standard de l'industrie aujourd'hui).

#### 1.3.1 Algorithme de masse

A partir de la loi fondamentale de la dynamique  $\vec{F} = m\vec{A}$  où  $\vec{A}$  est l'accélération, et d'une approximation de la représentation discrète de la vitesse et de l'accélération, on a :

$$\begin{aligned} V_n &= X_n - X_{n-1} \\ A_n &= V_n - V_{n-1} \end{aligned}$$

L'écart entre deux échantillons étant unitaire, on obtient la loi d'évolution :

$$X_{n+1} = 2X_n - X_{n-1} + \frac{F_n}{m}$$

où  $X_{n+1}$  est la position calculée,  $X_n$  et  $X_{n-1}$  la mémoire des positions des deux instants précédents,  $F_n$  la force d'entrée et  $m$  la valeur de la masse.

#### 1.3.2 Algorithme d'interaction linéaire, liaison viscoélastique

Dans le cas d'une élasticité pure (un ressort), à partir de la loi temporelle :

$$F(t) = K(X_1(t) - X_2(t))$$

où  $K$  est le paramètre représentant la constante de raideur du ressort, on obtient :

$$F_{2n} = -F_{1n} = -K(X_{1n} - X_{2n})$$

De la même façon, dans le cas d'un frottement visqueux pur, l'algorithme est basé sur la loi d'évolution :

$$F_{2n} = F_{1n} = -Z(V_{1n} - V_{2n})$$

où  $Z$  est le coefficient de frottement visqueux.

CORDIS-ANIMA intègre les deux interactions (élastique et frottement visqueux) au sein d'un même module recevant les variables  $X_1$  et  $X_2$ , les paramètres  $K$  et  $Z$ , et générant les variables  $F_1$  et  $F_2$ .

### 1.3.3 Interaction non linéaire : la liaison conditionnelle

Il s'agit d'une interaction viscoélastique dont l'état peut changer en fonction de conditions sur ses variables d'entrées. Une condition souvent utilisée est la distance entre deux masses ponctuelles reliées par un module LIA, ce qui permet de modéliser une interaction seuillée. La modification de l'état de l'interaction s'effectue par modification des paramètres  $K$  et  $Z$ .

### 1.3.4 Modules dégénérés

La masse ponctuelle, la liaison viscoélastique et la liaison conditionnelle forment une base algorithmique du système CORDIS-ANIMA. Cette liste de modules n'est toutefois pas exhaustive. D'autres modules ont été développés dont notamment des modules dits dégénérés car incomplets, mais très pratiques :

- ❑ Le module SOL, homogène à un point matériel. Quelque soit la force reçue, il renvoie une position fixe. Il est utilisé comme point d'attache.
- ❑ Le module PES, homogène à un élément de liaison. Quelque soit la position, il renvoie une valeur constante de force. Il permet de simuler la pesanteur.

## 1.4. Méthodologie

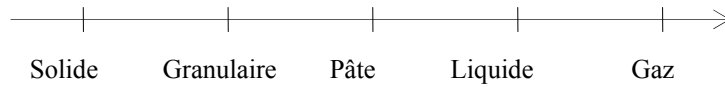
CORDIS-ANIMA va nous permettre de modéliser des phénomènes en les représentant comme un ensemble de masses ponctuelles liées entre elles par des interactions appropriées.

Après avoir défini un modèle de départ en se basant sur notre connaissance physique, l'expérimentation permettra de faire évoluer le modèle : c'est en comparant, en général sur des critères subjectifs, le comportement du phénomène de synthèse avec celui de phénomènes réels que l'on va affiner les paramètres du modèle et éventuellement le modèle lui-même, par touches successives.

## 2. Les états de la matière

De façon générale, quand on parle d'états de la matière, on n'en dénombre que trois : solide, liquide et gaz. Ces trois états sont insuffisants pour considérer toutes les différentes caractéristiques

dynamiques qui séparent ces états. Nous en considérons cinq ([Luciani00]) : solide, granulaire, pâte, liquide et gaz (Figure 5).



**Figure 5 : Les différents états de la matière du plus solide au plus gazeux**

Tous ces états présentent différentes caractéristiques qu'il est indispensable de retrouver lors de la simulation.

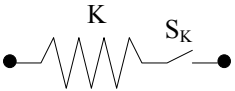

- ❑ Les solides se caractérisent par des interactions permanentes entre molécules qui déterminent la nature de la matière du solide ;
- ❑ Les matériaux granulaires sont soumis à des contraintes inhomogènes internes, ce qui permet de simuler des empilements, des effondrements, des avalanches de surface ;
- ❑ Les pâtes se caractérisent par un mouvement sinueux avec points d'inflexions, un tassement et un étalement de la matière ;
- ❑ Les liquides par des écoulements laminaires, des tourbillons de Kelvin-Helmholtz, des allées de von Karman ;
- ❑ Les gaz par des mouvements browniens, des mouvements libres.

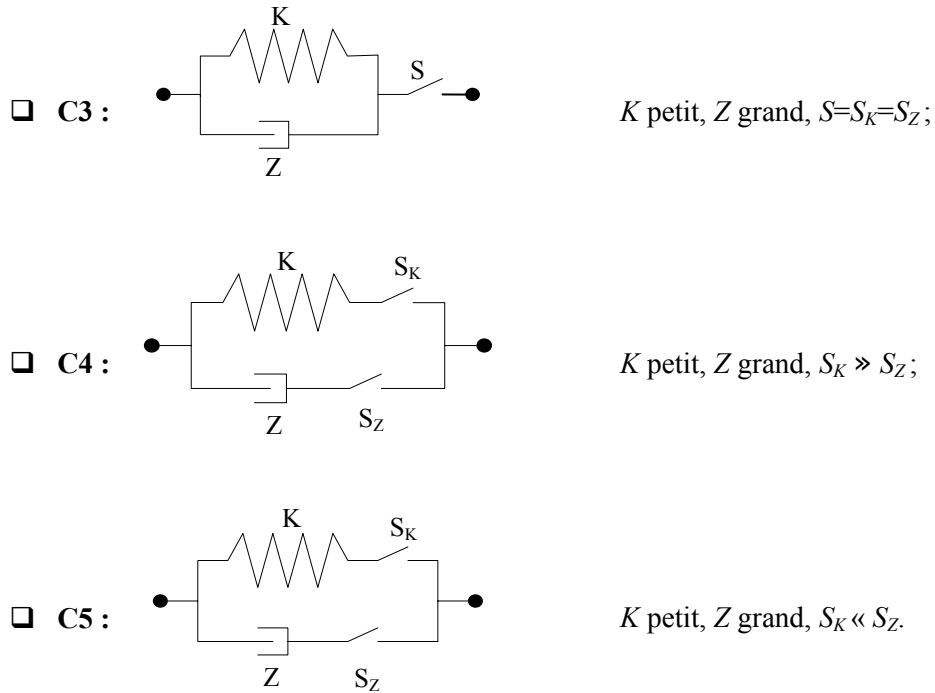
### 3. Le modèle générique

Dans ([Luciani91]), Luciani et al. ont montré que seuls quatre paramètres suffisent pour modéliser l'ensemble des états de la matière. Donc un modèle générique est constitué de masses reliées par des liaisons visqueuses seuillées ( $Z$ ,  $S_Z$ ), ainsi que des liaisons élastiques seuillées ( $K$ ,  $S_K$ ).

#### 3.1. Configurations possibles des liaisons inter-particulaires

Il existe cinq configurations possibles de liaisons inter-particulaires ([Luciani00]):

- ❑ C1 :   $S_k$  petit,  $K$  grand, sol rugueux ;
- ❑ C2 :   $S_Z$  petit,  $Z$  grand,  $S_Z < d_{intercouches}$  ;



### 3.2. Types de configurations utilisées

Phénomènes	Configuration	Particularités
Les matériaux granulaires	C1	$S_{Ksol} \ll S_{Kparticules}$ , viscosité ambiante
Des matériaux granulaires à la crème	C5	
Du sable sec vers le sable humide et la pâte expansive	C1	$S_{Ksol} \ll S_{Kparticules}$ , viscosité ambiante, mais $K$ plus petit et $Z_{ambient}$ plus grand
Du fluide visqueux compressible au matériau granulaire	C3	
Les pâtes	C5	
Les fluides	C4	
Les fluides turbulents	C2, C3, C4	pour avoir les tourbillons de Kelvin-Helmholtz, utiliser C2 ; pour avoir les allées von Karman, utiliser C4
Les gaz	C1	

**Tableau 1 : Types de configurations utilisées en fonction du phénomène voulu**



# Chapitre 4 Caractéristiques dynamiques de phénomènes

---

Lorsque l'on regarde un phénomène se développant dans la nature, nous l'identifions sur des critères visuels. Nous reconnaissons ses caractéristiques dynamiques, sa forme, sa texture. Comme nous l'avons exposé au chapitre précédent, nous modéliserons ce phénomène, ou plus précisément la catégorie à laquelle il peut appartenir par méthode physique particulière. Nous devons donc être en mesure de spécifier les principales caractéristiques dynamiques de chaque phénomène, pour pouvoir juger de sa qualité.

Nous nous attachons plus particulièrement à la modélisation de phénomènes naturels issus d'écoulements de matière. Comme nous disposons d'un modèleur générique, il nous est apparu normal d'essayer de mettre en œuvre une méthode de visualisation générique. Pour qu'une telle technique puisse être validée, il fallait que les phénomènes que nous avons choisis présentent une grande variété de dynamiques.

Les caractéristiques dynamiques sont un ensemble de particularités, plus ou moins complexes, de mouvements, de dynamiques internes, de formes, caractéristiques du phénomène. Les caractéristiques dynamiques citées ici se situent à différentes échelles de représentation : microscopique, mésoscopique et macroscopique. Quelle que soit l'échelle à laquelle la particularité se situe, le modèle physique et la visualisation doivent être capable de la rendre. Sans elles, il devient difficile d'authentifier notre mouvement.

La plupart des phénomènes choisis sont des agglomérats ([jimenez93]) : les particules significatives du modèle (i.e. la plupart du temps les particules du flux) interagissent toutes physiquement deux à deux entre elles. Donc si il y a  $n$  particules, il y aura  $\frac{n(n-1)}{2}$  interactions. Les agglomérats sont des objets à forme variable et indéfinissable en terme de géométrie. Pour montrer l'efficacité de la modélisation physique particulière, nous avons également tenté d'étendre l'étude à une autre catégorie de phénomènes : la morphogenèse, en prenant comme exemple de base la croissance de cellules végétales et la constitution d'un tissu cellulaire qui en découle.

Pour des raisons de lisibilité, les caractéristiques dynamiques ne sont présentées qu'en deux dimensions, mais il est évident que la plus part d'entre elles se produisent en trois dimensions.

## 1. Les fluides

Le fluide est sans doute l'un des états de la matière regroupant la plus grande diversité de mouvements, de dynamiques. La différence entre ces phénomènes vient de leur consistance qui varie entre l'état granulaire et l'état liquide. Leur point commun est leur flux continu déformable pouvant prendre (après plus ou moins de temps) la forme du vase qui les contient.

De la pâte à pain – proche des matériaux granulaires car formant un tas -, à la pâte de dentifrice, à la mousse en passant par le gel, tous ces phénomènes font partie des fluides. La large gamme de caractéristiques dynamiques qu'ils présentent nous est apparue d'un intérêt non négligeable pour la mise en place d'une méthode de visualisation générique et robuste.

Nous n'exposerons pas toutes les caractéristiques dynamiques pour chacun des types de fluides mais une synthèse regroupant tous les phénomènes de cet état de la matière. Les particularités se retrouvent dans tous les phénomènes sous des formes plus ou moins marquées.

### 1.1. Obtention du mouvement sinueux

Lorsqu'un flux de matière s'écoule d'un entonnoir, la vitesse d'écoulement est indépendante de la quantité de matière contenue dans l'entonnoir (*Caractéristique Dynamique 1 du matériau fluide : écoulement*).

Dès que le flux touche le sol, des tensions internes apparaissent, engendrant un mouvement sinueux pour une portion de flux, sous l'influence du sol. Sous la pression du reste du flux, le fluide s'étale (*Caractéristique Dynamique 2 du matériau fluide : étalement*) dans la direction de la sinuosité.

Les tensions internes des particules basses vont se répercuter sur les particules du dessus. Ces tensions nouvellement créées sont là pour stabiliser les sous-tas. Elles sont de direction opposée aux premières, créant ainsi un extremum suivi d'un point d'inflexion (*Caractéristique Dynamique 3 du matériau fluide : point d'inflexion*, *Caractéristique Dynamique 4 matériau fluide : extremum*). Le point d'inflexion est la zone où d'un côté du flux nous avons un angle saillant et de l'autre un angle rentrant. L'extremum est une sinuosité. Le changement de sens des forces inter-particulaires donne lieu à une répétition par symétrie horizontale du mouvement sinueux (suite d'extrema et de points d'inflexions).

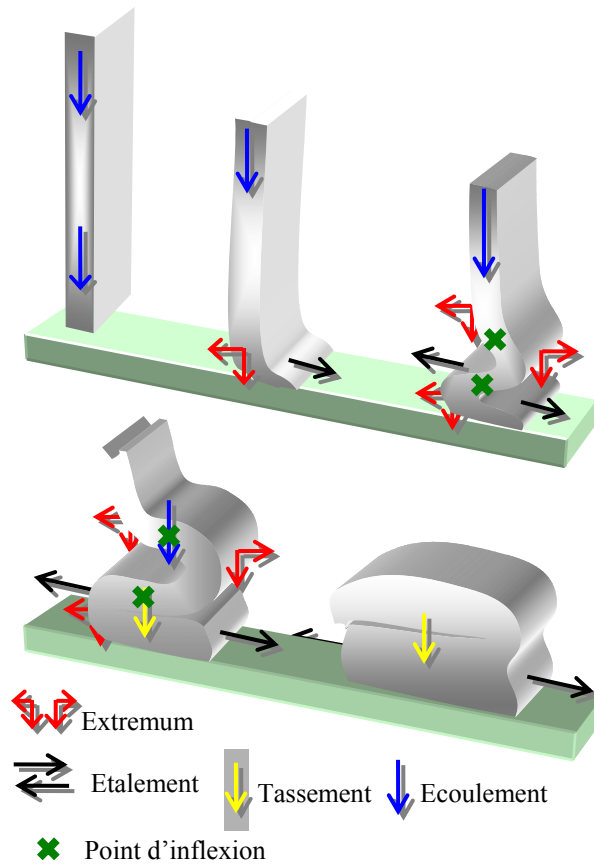


Figure 6 : Caractéristiques Dynamiques d'un matériau fluide

## 1.2. Le tassement

L'étalement continu des particules au sol et de celles sous son influence provoque un tassement que l'on appelle aussi mélange inter-couches (une couche est une portion de flux entre deux points d'inflexion) faisant ainsi disparaître le mouvement sinueux. (*Caractéristique Dynamique 5 du matériau fluide : tassement*).







Tableau 2 : Images de gel

## 2. Les matériaux granulaires

Les milieux granulaires sont très répandus dans la nature et présentent des caractéristiques dynamiques tout à fait surprenantes. Ces caractéristiques font d'eux un état de la matière à part entière. Par un simple changement dynamique de la configuration des grains (géométrie) il passe de l'état de fluide à celui de solide. En effet lorsqu'il s'écoule, il le fait comme un fluide, et lorsqu'un tas se forme, avec parfois la construction de voûtes, il devient solide.

Les évolutions des matériaux nous ont particulièrement intéressé. Ces phénomènes, présentent un lien forme-mouvement important, formes nous conduisant parfois à le qualifier de fluide.

Même si le tas de sable est organisé, cette organisation résulte d'un phénomène non structuré au niveau des composants élémentaires où les particules ne sont pas liées à leurs voisines initiales, mais évoluent de manière à former une structure stable.

### 2.1. Ecoulement et formation du tas de sable

Dans un récipient suffisamment étroit, la pression pour les matériaux granulaires est indépendante de la hauteur du matériau. Du sable dans la partie supérieure du sablier s'écoule à une vitesse indépendante de la quantité de sable contenu dans cette partie (*Caractéristique Dynamique 1 des matériaux granulaires : écoulement*).

Lorsque du sable s'écoule d'un entonnoir, il se forme un tas dont la pente augmente progressivement (*Caractéristique Dynamique 2 des matériaux granulaires : **empilement***) jusqu'à atteindre un niveau critique. Là, des avalanches de surface ont lieu (*Caractéristique Dynamique 3 des matériaux granulaires : **avalanches de surface***) pour que le tas retrouve sa stabilité. Ces deux pentes sont caractérisées par 2 angles (cf. Figure 7,  $\alpha$  est l'angle maximum de stabilité du tas, et  $\beta$  l'angle de repos).

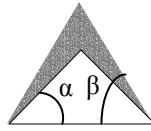


Figure 7 : Schéma de stabilité d'un tas de sable

## 2.2. Voûtes et effondrements

Les lignes de forces sont des indicateurs de la non répartition uniforme des forces dans le tas, forces assurant pourtant sa stabilité. Ces voûtes (*Caractéristique Dynamique 4 des matériaux granulaires : ligne de force*) sont des lignes de forces, appelés aussi réseaux de contraintes.

Une rupture dans les lignes de force entraîne un effondrement interne du tas en sous-tas non similaires au tas principal (*Caractéristique Dynamique 5 des matériaux granulaires : avalanche interne ou effondrement*).

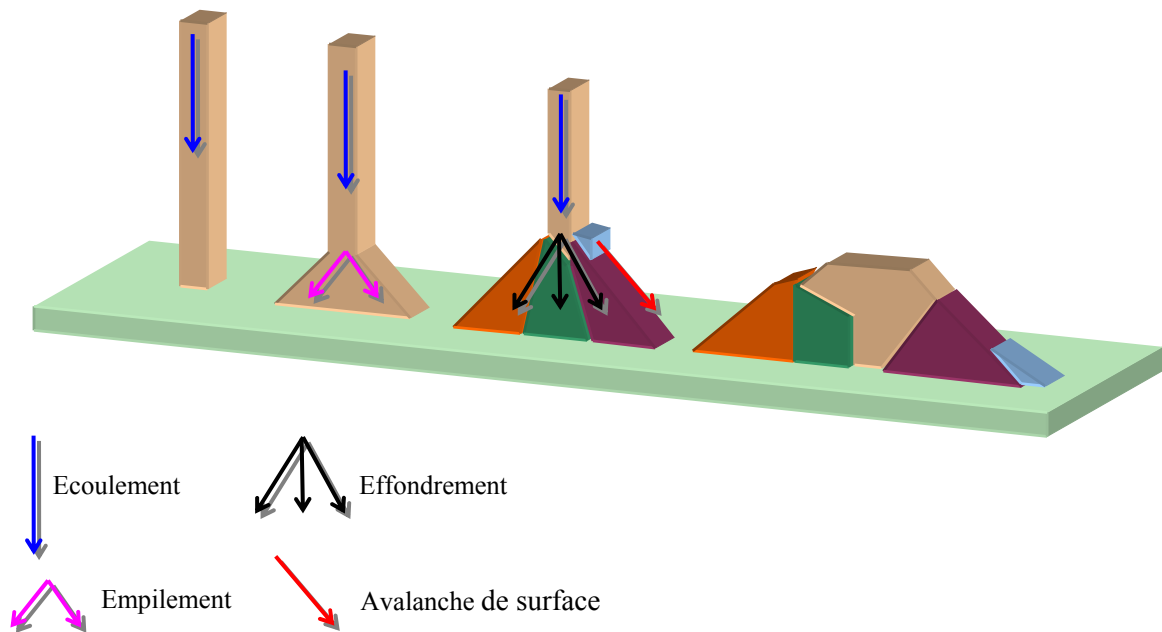


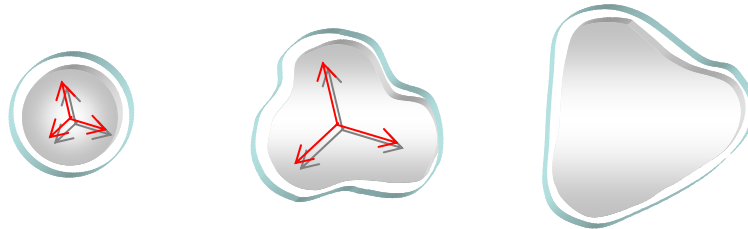
Figure 8 : Caractéristiques dynamiques des matériaux granulaires

## 3. Croissance de cellules végétales

La cellule est l'unité de base du vivant. Tout organisme, du plus simple au plus perfectionné, est constitué de cellules de différents types formants un ensemble coordonné.

La croissance des cellules végétales constitue un problème particulier de la biologie cellulaire : la cellule végétale est entourée d'une paroi rigide qui la protège des déformations et lui donne sa

forme. La croissance (son changement de taille et de forme) ne peut donc se réaliser que grâce à des changements de structure et des propriétés de cette paroi (*Caractéristique Dynamique 1 de la croissance de cellules : **changement de structure***).



**Figure 9 : Caractéristiques Dynamiques de la croissance de cellules**

Elle se fait par une extension progressive de l'encombrement spatial de la cellule basée sur une augmentation de la taille de la cellule (*Caractéristique Dynamique 2 de la croissance de cellules : **changement de taille***).

Les pressions internes du germe sont les moteurs d'une croissance souvent unidirectionnelle. La forme finale de la cellule est prédéfinie par des informations stockées dans le germe, et contrainte par l'environnement de la cellule (taille de l'espace libre disponible lors de la croissance) (*Caractéristique Dynamique 3 de la croissance de cellules : **forme à maturité prédéfinie***).

Un ensemble de cellules forme un tissu. Si le tissu vient à se déchirer lorsque les cellules sont à maturité, il ne pourra être reconstruit. (*Caractéristique Dynamique 4 de la croissance de cellules : **fracture non consolidable***)

## 4. Les fluides turbulents

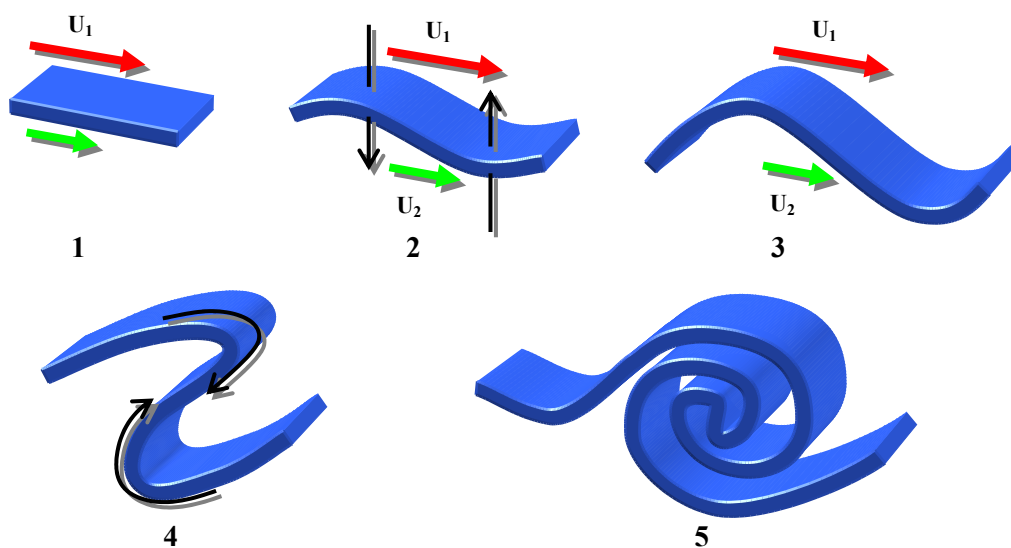
Les objets à contour flous (« fuzzy objects ») regroupent des phénomènes aussi variés que les nuages, la fumée, les gaz. Tous ont en commun un contour indéterminé, une forme indéfinissable par la géométrie car leurs évolutions sont très complexes à définir.

Nous allons définir les différentes caractéristiques de ces modèles même si ces derniers n'entreront pas dans notre étude en vue de la conception d'une méthode générique de visualisation. Leur contour flou n'oblige pas à la construction d'un volume, car on peut jouer sur la lumière et la transparence des particules qui les composent pour le visualiser. Cependant en tant que modèles physiques représentés par un agglomérat, et selon ce qui a été dit en introduction du présent chapitre, nous nous devons de les modéliser.

### 4.1. Tourbillons de Kelvin-Helmholtz

Une perturbation extérieure à un écoulement de fluide produit une légère oscillation des flux qui suffit à produire un tourbillon de Kelvin-Helmholtz. Mais ce n'est pas la seule façon d'obtenir des tourbillons de Kelvin-Helmholtz : une simple différence de vitesses entre deux fluides suffit. Nous ne n'explicitons pas ici toutes les configurations engendrant ces vortex.

Si nous considérons une interface (plan horizontale) coincée entre deux fluides  $U_1$  et  $U_2$ , et si la vitesse de  $U_1$  est supérieure à la vitesse de  $U_2$  (Figure 10.1), alors une légère oscillation se crée (Figure 10.2). La différence de pression provoquée par l'oscillation entre les parties concaves et convexes de l'interface augmente l'amplitude de celle-ci. Arrivée à une certaine amplitude, la partie concave de l'interface (partie haute) va subir l'action de  $U_1$ , tandis que la partie convexe (partie basse) celle de  $U_2$  (Figure 10.3).  $U_1$  étant plus rapide, la partie haute s'enroule provoquant ainsi une rotation de  $U_2$ ,  $U_2$  provoquant alors la rotation de  $U_1$  (Figure 10.4 et Figure 10.5). Le vortex engendré est un tourbillon de Kelvin-Helmholtz.



**Figure 10 : Formation d'un tourbillon de Kelvin-Helmholtz**

## 4.2. Allée de von Karman

Un fluide trouvant sur sa trajectoire un obstacle cylindrique crée après celui-ci une allée de von Karman, ensemble de tourbillons de Kelvin-Helmholtz de sens opposé.

Le fluide, de répartition de vitesse uniforme, peut au passage de l'obstacle s'assimiler à deux fluides de répartition de vitesse non uniforme (deux fluides à deux couches). On se retrouve alors avec deux configurations (configuration de la Figure 10) de génération de tourbillon de Kelvin-Helmholtz mais dont le sens de rotation est opposé.

Dans le cas d'un jet de fluide dans un milieu immobile, les tourbillons s'enroulent vers l'extérieur. Ce phénomène est aussi une allée de von Karman que l'on retrouve dans la formation de colonne de fumée. Cette caractéristique est un effet de couche limite se produisant dans le sillage de l'obstacle.

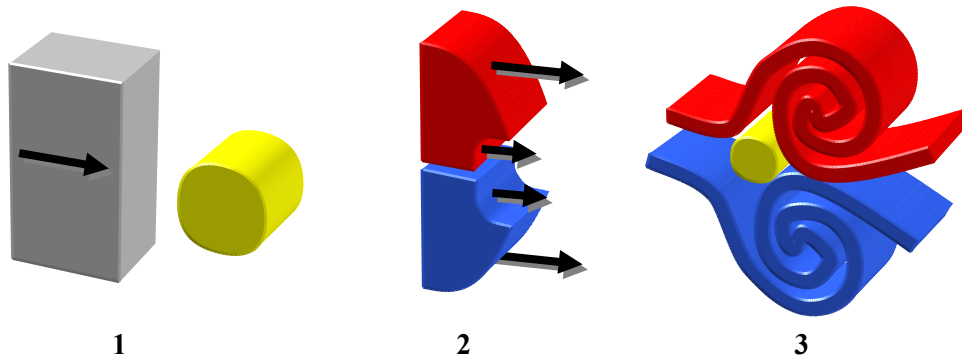


Figure 11 : Formation de l'allée de von Karman derrière un obstacle

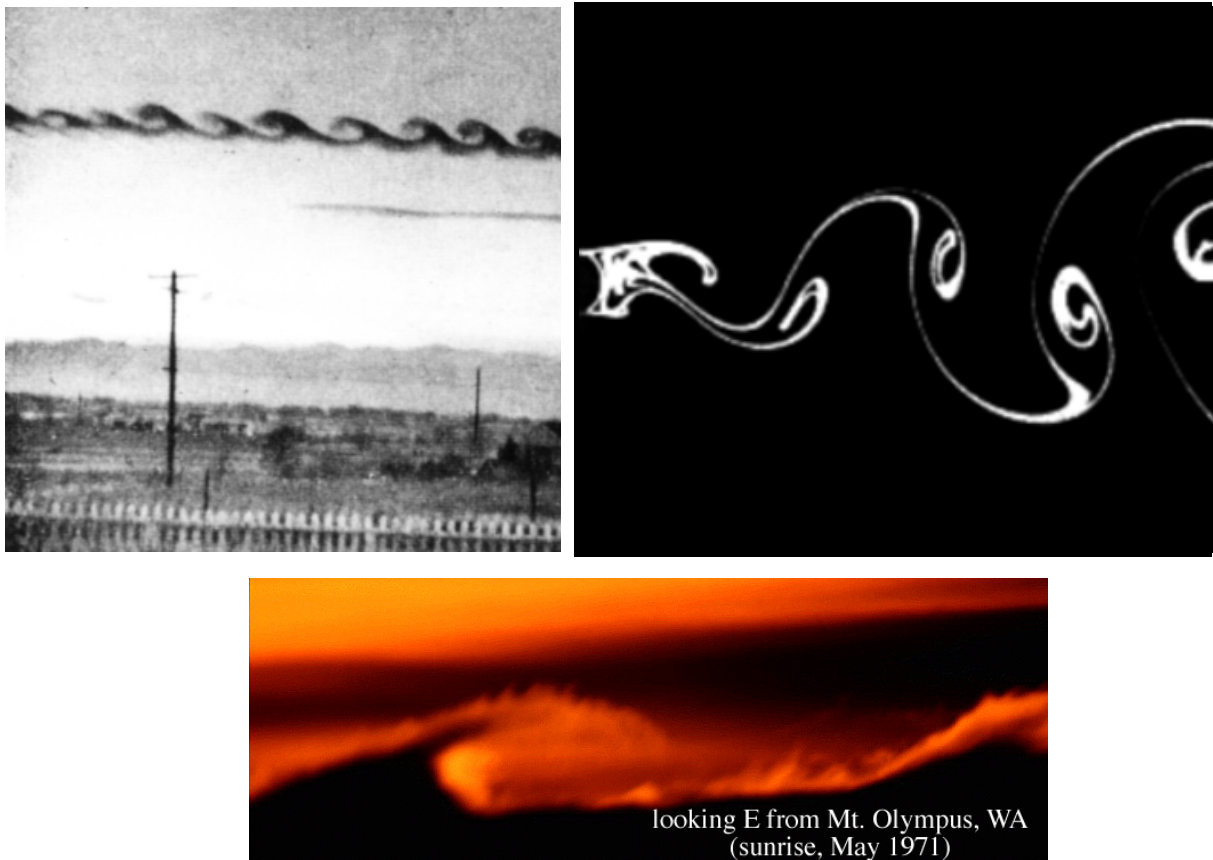
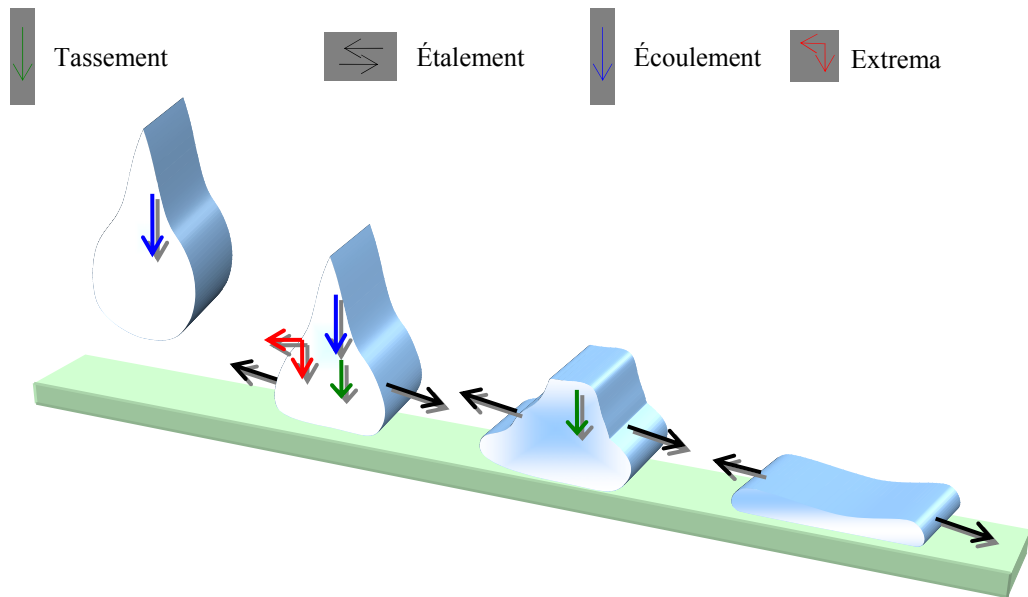


Figure 12 : Tourbillon de Kelvin-Helmholtz et Allée de von Karman

## 5. Les gouttes de fluide



**Figure 13 : Caractéristiques d'une goutte**

Nous avons ajoutées cette section car nous avons choisis de visualiser également un phénomène de goutte à goutte. Les caractéristiques dynamiques sont identiques à celles des fluides, mais ont lieu sur un petit nombre de particules. Le mouvement sinueux existe bel et bien mais les sinuosités sont moins visibles car peu de particules la forment.

## 6. Tableau des dynamiques caractéristiques

N°	Caractéristique	État de la matière	Explications
CD1	Écoulement	Fluide, Matériaux granulaires, Fluide turbulents	La vitesse d'écoulement dans l'entonnoir est indépendante de la quantité de matière faisant pression
CD2	Étalement	Fluide, Goutte	Glissement de la matière sur le sol
CD3	Point d'inflexion	Fluide, Goutte	Zone de changement de sens de courbure du flux (concave à convexe, ou inversement)
CD4	Extrema	Fluide, Goutte	Sinuosité entre deux points d'inflexion présentant un angle rentrant et un angle saillant
CD5	Tassement	Fluide, Goutte	Mélange de couches, disparition de deux zones de part et d'autre d'une sinuosité
CD6	Empilement	Matériaux granulaires	Formation de tas
CD7	Avalanche de surface	Matériaux granulaires	Particule (grain) roulant le long de la pente du tas car celle-ci est trop importante pour que le tas soit stable
CD8	Ligne de force	Matériaux granulaires	Séparation entre sous-tas dans un tas de sable
CD9	Avalanche interne, Effondrement	Matériaux granulaires	Réorganisation interne de la matière pour avoir un tas stable

<i>CD10</i>	Croissance par division cellulaire	Croissance de cellules végétales	Multiplication des cellules et constitution d'un tissu cellulaire
<i>CD11</i>	Fracture non consolidable	Croissance de cellules végétales	Lorsque le tissu se déchire, il ne peut se consolider seul
<i>CD12</i>	Modification de la structure de la paroi	Croissance de cellules végétales	Intervient pour que la croissance de la cellule puisse avoir lieu
<i>CD13</i>	Stockage d'informations sur la forme finale	Croissance de cellules végétales	Le germe, embryon de la cellule, possède des informations sur la façon dont il doit se développer
<i>CD14</i>	Tourbillon de Kelvin-Helmholtz	Fluides turbulents, Gaz	Vortex
<i>CD15</i>	Allée de von Karman	Fluides turbulents, Gaz	Double tourbillon de Kelvin-Helmholtz de sens de rotation opposé

# Chapitre 5 Simulations de phénomènes naturels

---

Comme nous l'avons indiqué à plusieurs reprises, nous avons sélectionné les phénomènes sur lesquels notre travail allait porter. Les modèles sont tous de type « agglomérat ». La matière y est constituée de  $n$  particules dont chacune est en interaction avec les  $(n-1)$  autres, ce qui suppose que la matière contient  $\frac{n(n-1)}{2}$  interactions. L'agglomérat est un modèle de composition-décomposition structurelle de la matière. La forme obtenue correspond à un état stable du groupe de particules, donc est le résultat d'un processus dynamique.

Nous exposerons tout d'abord le cas de la croissance de cellules végétales, puis les modèles d'écoulements de matière en commençant par le sable, puis en abordant les matériaux du plus solide aux plus turbulents : la pâte, le gel, les fluides turbulents. Toutes ces simulations sont calculées à une fréquence de 1050Hz. Nous avons cependant voulu adapter l'optimalité de cette valeur en simulant deux modèles - l'un de sable, l'autre de pâte - à des fréquences inférieures. Ces résultats seront présentés en fin de chapitre à la section 5.

Les modèles de matière (sable, pâte, gel, fluide turbulent) ont une conception commune que nous allons réexpliquer pour chacun des cas pour une question de clarté. Dans tous les cas, le flux de matière s'écoule d'un entonnoir vers un sol. Les résultats seront mis en parallèle avec les caractéristiques dynamiques du chapitre précédent. Nous les rappellerons sous la forme : *<phénomène, numéro de la caractéristique, intitulé>*.

## 1. La croissance de cellules végétales

La croissance de cellules végétales s'intègre dans le cadre du projet de modélisation de tissus végétaux, et plus généralement du processus de morphogenèse. Dans l'optique d'obtenir une diversité de comportements, nous devons être à même de simuler plusieurs formes de cellules. Ce travail s'est réalisé à partir des modèles de Laure France ([France95]).



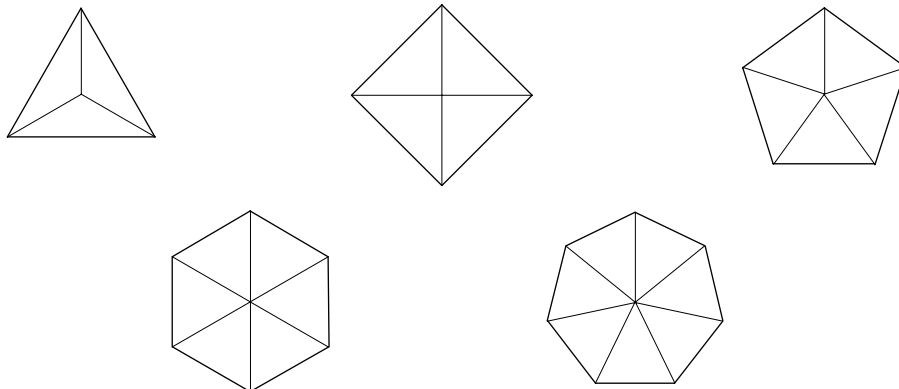
La cellule à laquelle on s'intéresse est celle susceptible de créer des tissus surfaciques. Elle ne sera donc pas représentée comme un objet tridimensionnel, mais bidimensionnel. Par rapport à une véritable cellule vivante, notre cellule correspondra à une *coupe* de la cellule. Du point de vue de la modélisation, la cellule est considérée comme un élément atomique indivisible pouvant se comprimer et se détendre selon son environnement. La notion d'indivisibilité est en contradiction avec ce qui a été dit au chapitre précédent : *<cellule, CD4, fracture non consolidable>*, mais nous souhaitons modéliser la croissance de la cellule du germe à la maturité, sans prendre en compte les déchirures, conséquence d'actions extérieures non en rapport avec l'environnement végétal de la cellule et le phénomène lui-même.

Le processus de croissance de la cellule lui procure de l'énergie qui lui permet de se développer d'un stade embryonnaire, appelé germe, à sa maturité. La maturité correspond à une taille prédéfinie si rien ne la contraint, à une taille moindre sinon. La cellule a la connaissance des procédés d'expansion auxquels elle est soumise, et possède des propriétés physiques élastiques.

## 1.1. Conception du modèle

### 1.1.1 Structure de la cellule

La croissance d'une cellule résulte d'un changement de la structure de sa paroi externe (CD12) et d'une extension de son encombrement spatial. La modélisation du phénomène se fera par augmentation du volume défini par les masses ponctuelles représentant la paroi cellulaire. Une telle représentation nous permettra d'aboutir à diverses formes en faisant varier le nombre de particules et leurs positions initiales. Les formes finales que nous avons choisies sont des polygones à 3, 4, 5, 6 ou 7 côtés.



**Figure 14 : Différentes formes de la cellule**

### 1.1.2 Les masses ponctuelles : les MAT

Le nombre de particules constituant la paroi de la cellule détermine la forme de celle-ci. Pour que sa forme ne soit pas trop malléable, nous avons ajouté une particule centrale. La masse de ce point matériel, barycentre de la cellule est prise égale à celle des particules de l'enveloppe, donnant ainsi une influence égale à toutes les masses ponctuelles, ou supérieure à cette dernière, modélisant ainsi

une forte densité au centre. Nous avons choisi une masse identique pour que la cellule s'épanouisse de manière équivalente. Les MAT constituant la paroi cellulaire doivent être assez légers pour ne pas dénaturer la forme obtenue au stade de la maturité. Car plus les particules ont une masse importante, plus l'énergie inter-particulaire est grande, ce qui entraîne un dépassement possible du stade de maturité.

Nous avons sélectionné deux configurations aux dynamiques répondant à notre attente.

- ❑ **Configuration 1** : les particules sont des MAT non-visqueux de paramètres  $\{M_i\}$
- ❑ **Configuration 2** : les particules sont des MAT<sup>9</sup> visqueux de paramètres  $\{M_i, Z_i\}$

### 1.1.3 Les interactions : les LIA

Les cellules lors de leur croissance présentent un comportement élastique qui doit être reproduit par les interactions entre masses ponctuelles. Les liaisons doivent être suffisamment rigides pour ne pas provoquer un phénomène de variation de taille, permanent ou brusque, qui perturberait la forme de la cellule. Mais la cellule doit être raisonnablement flexible pour arriver jusqu'à sa forme de maturité.

Nous pouvons classer les interactions structurant la cellule ainsi :

- ❑ Celles formant la paroi externe de la cellule : *LIA enveloppe*
- ❑ Celles reliant le barycentre à la paroi : *LIA centre-paroi*
- ❑ Celles structurant l'espace entre particule de la paroi : *LIA internes*

Les différentes expérimentations nous ont montré qu'une différenciation des valeurs des paramètres des liaisons n'était pas utile dans la mesure où les dynamiques résultantes étaient fort semblables.

Les valeurs de la composante élastique ainsi que les longueurs à vide sont identiques pour les deux configurations. La raideur ne doit être ni trop faible, ni trop forte pour que la forme se stabilise. Les longueurs à vide varient selon le type de liaisons. Elles ont été calculées pour le polygone qu'elles représentent sachant que les liaisons centre-paroi sont de longueur  $d$ . Les valeurs des paramètres pour la cellule en pentagone et pour les deux configurations retenues ont été regroupées dans le Tableau 3, page 66.

#### **Configuration 1**

Pour dissiper l'énergie potentielle des interactions élastiques gérant la croissance de la cellule, nous devons prendre des interactions visco-élastiques.

#### **Configuration 2**

Pour cette configuration, la dissipation s'effectue dans la fonction de viscosité de milieu incluse dans les MAT. Ce sont eux qui absorbent le surplus d'énergie élastique des liaisons. Il n'est alors plus nécessaire d'introduire la viscosité dans les interactions.

---

<sup>9</sup> Le module MAT visqueux est un module « intégré » de CORDIS-ANIMA qui permet de simuler le comportement inertiel d'une masse qui évolue dans un milieu visqueux immobile.

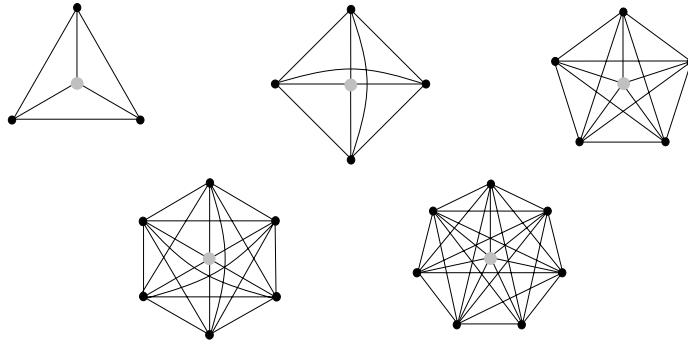


Figure 15 : Liaisons structurelles des cellules

		Configuration 1	Configuration 2
<b>MAT</b>	Gravité	0.00	0.00
	Masse $M_i$	0.05	0.05
	Viscosité $Z_i$	0.00	0.50
<b>LIA enveloppe</b>	Raideur K	0.01	0.01
	Viscosité Z	0.10	0.00
	Seuil (longueur à vide)	$2d \sin(\frac{\Pi}{5})$	$2d \sin(\frac{\Pi}{5})$
<b>LIA centre-paroi</b>	Raideur K	0.01	0.01
	Viscosité Z	0.10	0.00
	Seuil (longueur à vide)	$d$	$d$
<b>LIA internes</b>	Raideur K	0.01	0.01
	Viscosité Z	0.10	0.00
	Seuil (longueur à vide)	$2d \sin(\frac{2\Pi}{5})$	$2d \sin(\frac{2\Pi}{5})$

Tableau 3 : Paramètres de la cellule en forme de pentagone

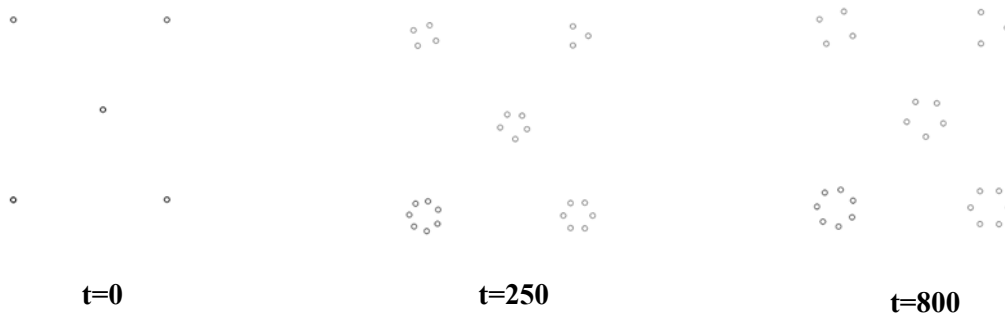
## 1.2. Simulations réalisées

Nous avons réalisé un certain nombre de simulations en modifiant les paramètres précisés dans le Tableau 3. Pour être en accord avec le caractère élastique de la paroi de la cellule, nous avons sélectionné comme simulation de référence la modélisation réalisée avec la configuration 2. La simulation de référence est celle qui servira à l'élaboration d'une méthode de visualisation pour les modèles de croissance de cellules végétales. La stabilisation de la cellule, se fait grâce à la nature visqueuse des MAT. Les trois images de la Figure 16 montrent les résultats de cette simulation.

La forme de la cellule à maturité, définie par le seuil des interactions, est obtenue par stabilisation de la structure à l'instant  $t_f$ , ce qui reproduit la caractéristique <cellule, 4, forme prédéfinie>. La première caractéristique (le changement de structure de la paroi) n'a pas été modélisée car il n'était *a priori* pas utile pour simuler ce modèle. De plus elle n'était pas issue d'un processus

dynamique. La croissance se fait uniquement par augmentation du volume de la cellule (*<cellule, 2, changement de taille>*).

A partir de ces cellules élémentaires et de ce type élémentaire de processus de croissance, l'objectif ultérieur est de construire un tissu cellulaire par accrochage dynamique de ces cellules dynamiques. Ce thème étant un thème de recherche en soi ne sera pas traité ici. Nous nous limiterons à la croissance cellulaire qui nous permettra d'initialiser le travail sur la visualisation de ce type de phénomènes dynamiques.



**Figure 16 : Modèles de croissance de cellules végétales**

## 2. Le sable

Le sable est un matériau granulaire riche en dynamiques variées et complexes. Sa modélisation est ardue lorsque la méthode utilise les lois physiques. Une étude a retenu notre attention celle de Luding et al. ([Luding96]) qui modélise du sable s'écoulant d'un tube. Les calculs sont complexes, les simulations ainsi produites ne présentent pas toutes les dynamiques pouvant résulter de l'écoulement de sable depuis un entonnoir vers le sol.

Une méthode par modélisation physique particulière paraît être une bonne approche pour la réalisation d'un modèle réaliste de sable. Dans un premier temps, nous concevrons ce modèle en deux dimensions pour régler correctement les paramètres des interactions entre particules, puis dès qu'un modèle serait jugé satisfaisant nous l'étendrons à la troisième dimension.

Les résultats obtenus par le laboratoire ICA à partir des travaux de [manzotti95] et [luciani95] ont permis de spécifier les conditions minimales nécessaires et suffisantes pour qu'il y ait formation d'un tas de sable. Il faut que le sol soit rugueux ou que les extrémités de la base du flux soient stoppées par un obstacle, les parois du réceptacle par exemple. En nous inspirant de ces études, nous avons donc mis au point deux modèles de sable l'un avec un sol rugueux, l'autre avec un sol lisse et un réceptacle pour caler la matière. Seul l'environnement physique et l'échelle des modèles varient.

## 2.1. Conception du modèle 2D

La réécriture et les tests du modèle en 2D ont été nécessaires pour passer de manière homogène au modèle 3D à partir de la même base. C'est pourquoi, nous avons réécrit le modèle original de Manzotti et effectué à partir de notre modèle nos propres simulations et observations, nécessaires pour le passage à l'étude des procédés de visualisation.

### 2.1.1 L'environnement physique du modèle

Nous appelons environnement physique du modèle l'ensemble des composants ne représentant pas la matière granulaire que l'on souhaite étudier, lui permettant de s'écouler de haut en bas dans un mouvement réaliste.

Les particules utilisées pour l'environnement sont des modules de types SOL de CORDIS-ANIMA. Ces modules ont une position fixée en début de simulation qui ne varie pas avec le temps.

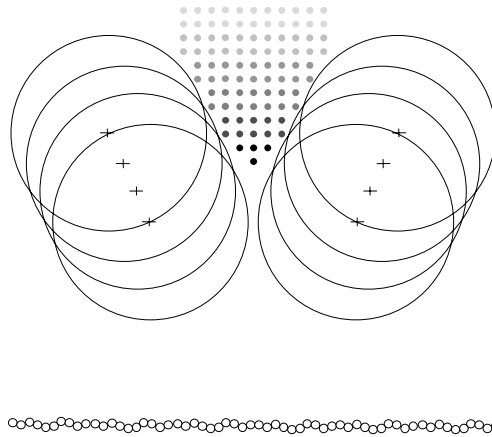
L'environnement est plongé dans un milieu de gravité non nulle, possédant une viscosité ambiante faible nécessaire à l'absorption d'une partie de l'énergie élastique du modèle.

#### a. Modèle à sol rugueux

L'entonnoir est modélisé par un ensemble de 8 modules SOL formant un cône non fermé en sa partie la plus basse (4 SOL pour chaque côté). Le goulot étroit terminant l'entonnoir n'a pas été modélisé car nous ne versions pas un liquide dans un récipient à embouchure étroite mais simplement en direction d'un sol.

Chaque module SOL possède un seuil de non pénétrabilité de son espace intrinsèque. Ces seuils sont identiques pour tous les modules de l'entonnoir, et a été choisi de manière à ce que la partie inférieure de l'entonnoir soit assez large pour laisser passer le flux de matière, i.e. d'une distance supérieure au seuil de l'interaction entre particules de sable.

L'entonnoir peut être vu comme un haut de sablier. La partie basse dans laquelle s'écoule le flux est dépourvue de parois et n'est constituée que d'un sol rugueux. Ce sol est formé d'un ensemble de modules SOL répartis de manière aléatoire selon une droite horizontale et de part et d'autres de celle-ci. La répartition des masses ponctuelles selon les abscisses se fait en fonction du volume des particules de sable, c'est-à-dire en fonction du seuil élastique des interactions. L'écart entre les positions des particules sur l'axe des ordonnées est minime. Cette irrégularité sert à donner au sol un aspect plus naturel. Le modèle à sol rugueux est présenté à la Figure 17 .

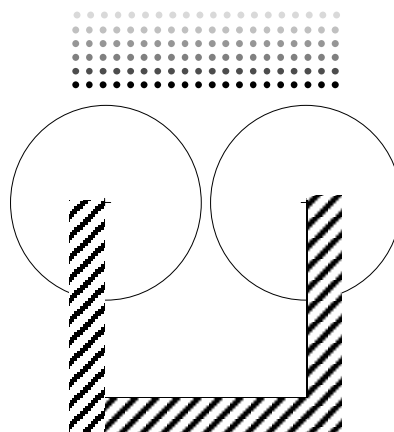


**Figure 17 : Modèle 2D de l'environnement physique à sol rugueux**

### **b. Modèle à sol lisse**

L'entonnoir est composé de deux modules SOL dont la distance les séparant, moins leur seuil caractéristique, laisse s'écouler la matière.

Comme nous voulions spécifier un modèle à sol lisse, nous avons construit un réceptacle. Le réceptacle est un récipient aux parois lisses verticales, dont le sol est lisse. Pour simuler des parois verticales rectilignes, nous avons placé deux modules SOL très éloignés par rapport à l'échelle de notre modèle dont le seuil d'interaction est largement supérieur à la taille que nous voulions donner à notre réceptacle. Dans le cas d'un sol lisse, il suffit d'ajuster les paramètres d'un SOL en suivant le principe utilisé pour les parois pour obtenir un sol horizontal. L'environnement physique du modèle avec sol lisse est représenté à la Figure 18.



**Figure 18 : Modèle 2D de l'environnement physique à sol lisse**

## **2.1.2 La matière**

### **a. Les masses ponctuelles : les MAT**

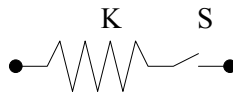
La matière est constituée de  $n$  modules MAT visqueux de masse 0.001. Nous avons testé différents placements initiaux des particules (en forme de triangle à l'intérieur de l'entonnoir, en

rectangle au dessus de l'entonnoir). La configuration en triangle a été préférée pour le sol rugueux tandis que l'autre pour le sol lisse. Le nombre de MAT composant la matière diffère mais cela n'a pas d'incidence sur les dynamiques perçues, la seule conséquence est une diminution du temps de calcul pour le modèle à sol lisse.

Nous avons expérimenté différentes positions de la matière à l'instant initial pour évaluer l'influence de ce positionnement sur les dynamiques d'un modèle. Quelle que soit la manière de placer les masses ponctuelles utilisées nous avons obtenu les mêmes caractéristiques mais un placement ordonné, régulier avec une distance entre particules supérieure au seuil élastique des liaisons les reliant, évite de simuler sur un temps trop long. En effet si les particules sont initialisées à des places aléatoires, avant que le flux ne s'écoule de l'entonnoir, il faut attendre une réorganisation de la matière.

### b. Les interactions : les LIA

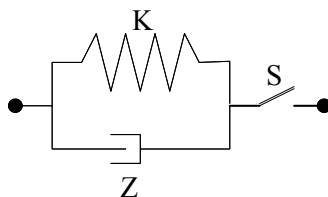
Les dynamiques sont issues du caractère élastique du sable. Donc les interactions entre particules de la matière sont élastiques et de seuil adapté à son environnement physique et à son positionnement. Les valeurs des paramètres ont été regroupées dans les tableaux de la section 2.1.4. La Figure 19 représente la liaison placée entre chacune des particules et les  $n-1$  autres.



**Figure 19 : Liaison entre particules de la matière**

### 2.1.3 L'interaction environnement matière

Les interactions entre les particules de matière et le sol, l'entonnoir et le réceptacle lorsqu'il en existe un, sont des liaisons visco-élastiques de raideur et de viscosité identiques. Les seuils correspondent au volume physique que l'on a voulu donner aux différents composants de l'environnement physique. Sans ce seuil, ces particules sont assimilables à un point matériel sans volume. La Figure 20 représente le type de liaison utilisée entre l'environnement et la matière.



**Figure 20 : Liaisons entre l'environnement physique et la matière**

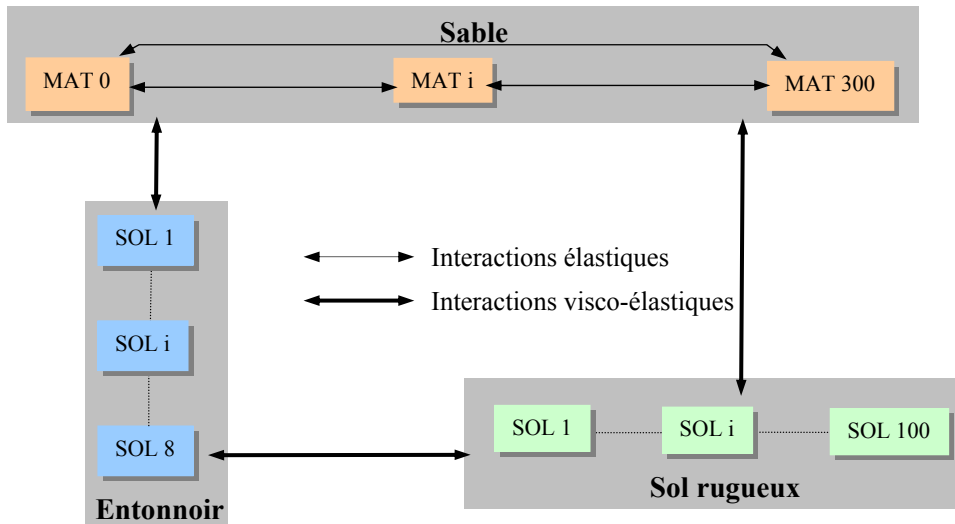


Figure 21 : structure du modèle 2D à sol rugueux

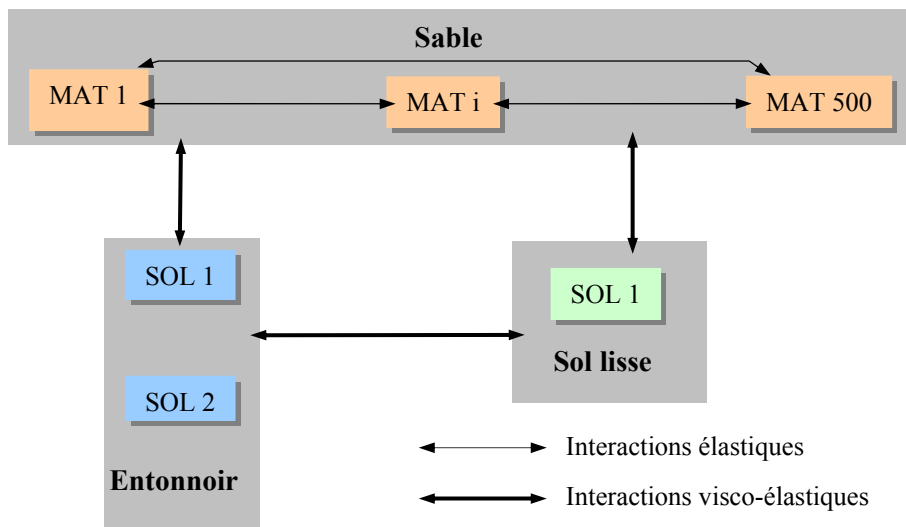


Figure 22 : structure du modèle 2D à sol lisse

## 2.1.4 Les caractéristiques retenues

### a. Le modèle du sol rugueux

	Nombre	Type	Masse	Viscosité
Entonnoir	8	SOL	-	-
Sol	100	SOL	-	-
Matière	500	MAT	0.001	0.2

	Raideur K	Viscosité Z	Seuil S
Matière-Sol	500	0.2	0.002
Matière-Entonnoir	500	0.2	0.050
Matière-Matière	500	0.0	0.002



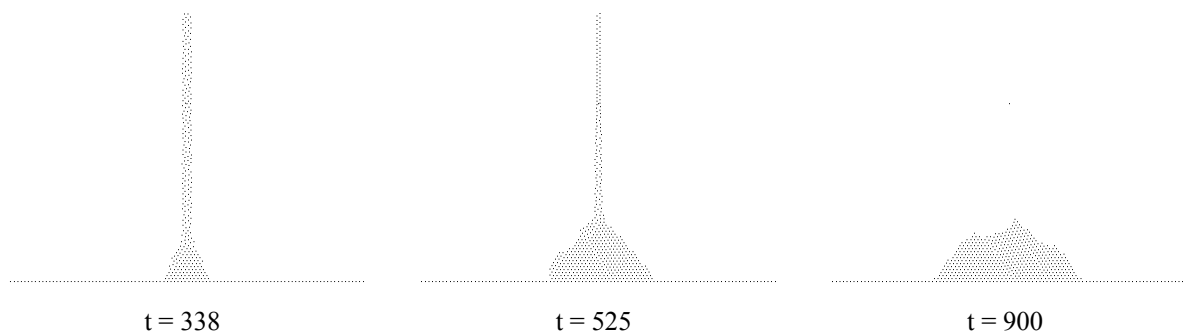
## b. Le modèle du sol lisse

	Nombre	Type	Masse	Viscosité
Réceptacle	2	SOL	-	-
Entonnoir	2	SOL	-	-
Sol	1	SOL	-	-
Matière	300	MAT	0.001	0.05

	Raideur K	Viscosité Z	Seuil S
Matière-Sol	500	0.05	128.020
Matière-Entonnoir	500	0.05	0.965
Matière-Réceptacle	500	0.05	128.020
Matière-Matière	500	0.00	0.035

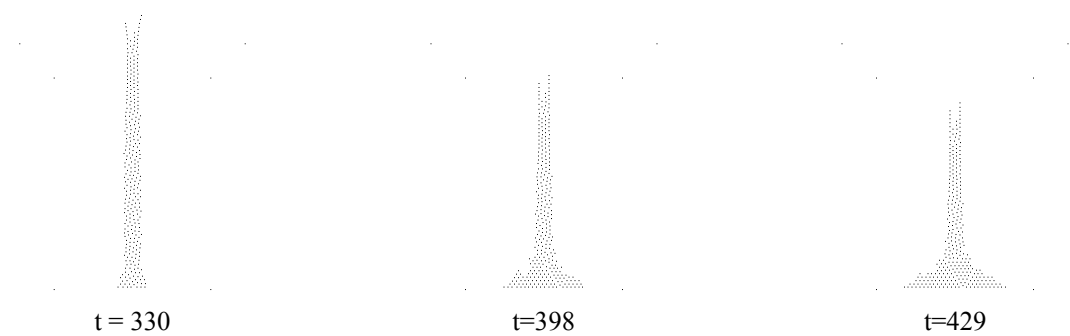
### 2.1.5 Validation des modèles

Les modèles créent des comportements sensiblement similaires. Dans nos simulations, le modèle à sol rugueux produit des tas formés (*<sable, 2, empilement>*) dans lesquels les restructurations internes (*<sable, 5, avalanche interne>*) sont plus rares malgré le flux nourri qui s'écoule de l'entonnoir vers ce dernier (*<sable, 1, écoulement>*). Mais on remarque cependant de nombreuses avalanches de surface (*<sable, 3, avalanche de surface>*). Entre deux sous-tas, et peu avant ces effondrements, on remarque des lignes de force (*<sable, 4, ligne de force>*) qui apparaissent et disparaissent au gré des dynamiques.



**Figure 23 : Modèle 2D de sable à sol rugueux**

En ce qui concerne le modèle à sol lisse, le tas est en permanente évolution car le sable glisse sur le sol et l'empilement de particules reçoit à chaque instant de la matière supplémentaire. Les avalanches internes ainsi que les lignes de forces sont plus nombreuses et visibles (les lignes de force sont parfaitement rectilignes). De ce fait, durant l'écoulement, le tas présente bien sa forme triangulaire caractéristique, avant les réorganisations internes puis la surface du tas tend à s'uniformiser.



**Figure 24 : Modèle 2D de sable à sol lisse**

C'est pourquoi, nous avons cependant choisi de garder pour de futures études ces deux modèles bien que celui à sol rugueux soit fort coûteux en calcul, puisqu'il y a un nombre plus important de liaisons et de particules.

## 2.2. Conception du modèle 3D

Le travail a principalement porté sur l'extension des composants de l'environnement physique de la 2D vers la 3D en leur ajoutant des portions selon la direction manquante. Par contre, il n'était *a priori* pas certains que les résultats obtenus en 2D se retrouveraient intégralement en 3D. Le travail a donc consisté en une véritable validation du modèle en 3D avec la recherche des paramètres associés aux phénomènes recherchés. Or, le problème de visualisation en 3D est critique pour la phase d'évaluation du modèle lui-même.

### 2.2.1 L'environnement physique du modèle

#### a. Modèle à sol rugueux

Nous avons choisi de garder un entonnoir en forme de sablier pour le modèle à sol rugueux. La Figure 25 représente le modèle. L'entonnoir n'a pas été représenté tridimensionnellement afin que les dessins restent compréhensibles.

Pour diminuer le temps de calcul requis, nous n'avons utilisé que la partie utile du sol (en contact avec le tas) – soit la partie centrale de 24\*24 modules SOL.

#### b. Modèle à sol lisse

Le réceptacle se compose d'un sol et de 4 parois verticales rectilignes. Pour la modélisation de l'entonnoir, nous nous sommes basés sur les résultats obtenus lors de l'étude de la modélisation de pâte tridimensionnelle (présentée dans la prochaine section), à savoir une duplication de l'entonnoir 2D, axé selon X, en une seconde portion d'entonnoir selon Z, et une obstruction des espaces entre ces modules SOL afin que la taille de l'entonnoir soit respectée et que celui-ci ne comporte pas de cavités. Le modèle à sol lisse en trois dimensions est synthétisé par la Figure 26.

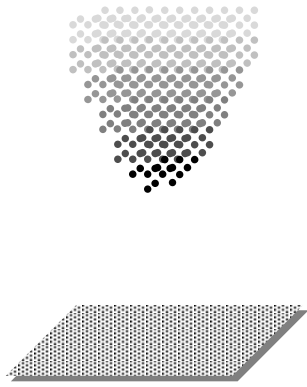


Figure 25 : Environnement du modèle 3D à sol rugueux

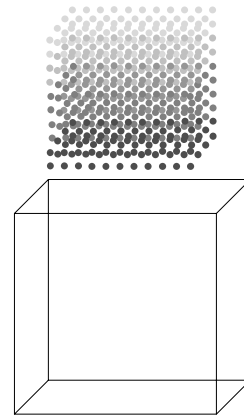


Figure 26 : Environnement du modèle 3D à sol lisse

### 2.2.2 La matière

Lors de la réalisation des modèles bidimensionnels, nous avons pu confirmer les résultats de [manzotti95] : l'obtention des caractéristiques de sable est possible avec peu de particules. Nos simulations 3D montrent que ce résultat se transpose en 3D puisque le modèle 3D de sol rugueux ne contient que 500 MAT pour le flux granulaire. Le modèle à sol lisse est composé 900 MAT (le nombre d'interactions avec l'environnement et notamment le sol étant bien moindre, il est possible de rajouter des MAT en gardant un temps de calcul acceptable). Trop de particules augmenterait sensiblement le temps de simulation, tandis que trop peu enlèverait la cohésion de la matière (le flux se réduirait à un fin filament) de laquelle résulte les dynamiques.

Les interactions entre particulaires sont *a priori* identiques à celles des modèles 2D. Le premier objectif est de voir si elles conduisent aux mêmes dynamiques.

### 2.2.3 L'interaction environnement matière

Nous avons rajouté les liaisons nécessaires à la connexion des éléments inexistants en 2D aux autres composants du modèle. Les valeurs des paramètres des liaisons entre composants du modèle sont inchangées.

### 2.2.4 Les caractéristiques retenues

Les modifications sont indiquées en gras.

#### a. Le modèle du sol rugueux

	Nombre	Type	Masse	Viscosité
Entonnoir	<b>16</b>	SOL	-	-
Sol	<b>576</b>	SOL	-	-
Matière	500	MAT	0.001	<b>0.1</b>

	Raideur K	Viscosité Z	Seuil S
Matière-Sol	500	<b>0.1</b>	0.002
Matière-Entonnoir	500	<b>0.1</b>	0.050
Matière-Matière	500	0.0	0.002

## b. Le modèle du sol lisse

	Nombre	Type	Masse	Viscosité
Réceptacle	4	SOL	-	-
Entonnoir	8	SOL	-	-
Sol	1	SOL	-	-
Matière	900	MAT	0.001	0.05

	Raideur K	Viscosité Z	Seuil S
Matière-Sol	500	0.05	128.020
Matière-Entonnoir	500	0.05	0.965
Matière-Réceptacle	500	0.05	128.020
Matière-Matière	500	0.00	0.035

### 2.2.5 Validation des modèles

L'analyse de modèles tridimensionnels est complexe lorsqu'il s'agit d'apprécier des dynamiques internes et ces types de réarrangements à stabilité limite. Nous espérons que, comme le type des interactions internes de la matière n'a pas été changé, les caractéristiques dynamiques que l'on obtiendrait seraient de même nature.

Les tendances à un comportement fluide du modèle à sol lisse (non formation d'amoncellements) se sont révélés être plus importants en 3D, validant *a posteriori* les résultats élaborés en 1995 par Manzotti, à savoir qu'une condition nécessaire et suffisante pour qu'un tas « bien formé » de matériau granulaire se forme est que le flux rencontre un obstacle rugueux. En effet, les effondrements internes ont semble-t-il été absorbés par la dynamique résultant du sol lisse à savoir l'étalement plus ou moins homogène du flux sur la surface du réceptacle. Nous n'observons aucune formation de tas, d'effondrements mais uniquement l'écoulement du sable, les avalanches de surface. Une alternative pour retrouver la formation de tas est de rajouter des parois pour limiter le glissement du flux et recréer les effets solides d'arcs-boutements conduisant à la formation de tas et de sous-tas internes. On obtient alors un tas bien structuré dès que la matière atteint les parois. Aucun effondrement interne ou création de ligne de force ne semblent se produire. Mais cette absence est peut-être aussi due à la difficulté d'analyser des structures à l'intérieur du tas, masquées par les particules en surface.

Le modèle à sol rugueux, malgré le peu de MAT formant le flux et un blocage dans l'entonnoir d'une partie de la matière, s'est avéré par contre fonctionner dans la continuité des résultats théoriques et pratiques du modèle 2D. Cependant nous ne distinguons non plus aucun effondrement interne, ni voûtes mais le tas est correctement formé.



**Figure 27 : Modèle de sable 3D à sol lisse**

Est-ce parce que nos simulations n'en présentent pas ? Nous pouvons en effet remarquer en 2D que pour certaines valeurs des paramètres physiques, les contraintes à l'intérieur du tas présentent bien une caractéristique essentielle des matériaux granulaires sans pour autant entraîner la formation d'effondrements. Le tas peut en effet rester dans un état d'équilibre instable. Pouvons-nous les déceler dans un amas 3D ? Les deux explications sont plausibles. Nous avons cependant commencé, en nous basant sur les analyses 3D, par apporter des modifications afin d'être certain d'avoir davantage de dynamiques caractéristiques des matériaux granulaires en modifiant les valeurs des paramètres physiques afin de rendre prépondérants les comportements élastiques sur les comportements visqueux.



**Figure 28 : Modèle 3D de sable à sol rugueux**

### 3. La pâte

Pour reproduire des écoulements de pâtes, en tant qu'état de la matière, nous partons des travaux de Zein Juneidi ([Juneidi97]). Nous avons sélectionné des types de phénomènes susceptibles de présenter l'ensemble des dynamiques décrites précédemment.

La modélisation de ces phénomènes s'est faite avant celle du sable. Aucun phénomène naturel tridimensionnel n'avait été jusqu'alors réalisé à l'aide de CORDIS-ANIMA, donc la validité

quantitative et qualitative du modèle n'était pas assurée. Il est possible qu'une ré-étude du champ paramétrique soit nécessaire.

Nous avons étudié deux modèles 2D : l'un de pâte, l'autre que nous avons appelé « fluide expansif ». Leur conception est présentée conjointement car seuls les paramètres des interactions inter-particules sont modifiés.

### 3.1. Conception du modèle 2D

#### 3.1.1 L'environnement physique du modèle

L'environnement physique est celui que l'on a exposé pour le modèle de sable à sol lisse. Rappelons-en les particularités. Il est composé d'un entonnoir fait de 2 modules SOL, d'un réceptacle aux parois verticales lisses (2 modules SOL), et au sol lisse (1 module SOL). Cet environnement est décrit à la Figure 18.

#### 3.1.2 La matière

##### a. Les masses ponctuelles : les MAT

La matière est composée de 300 modules MAT visqueux. Le nombre de molécules a été choisi pour que le flux de matière ne soit pas une particularité (ni trop fin, ni trop épais).

##### b. Les interactions : les LIA

La Figure 29 représente le type d'interaction inter-particulaire utilisé pour ces modèles de pâte. Outre des valeurs de seuil différentes pour les deux modèles proposés, celui assimilable à un fluide expansif possède une élasticité faible qui nous permet de modéliser la pression ([vapillon95]). Ainsi les particules se repoussent, et ont naturellement tendance à remplir les zones vides.

La viscosité a un double rôle : elle provoque des mouvements de rotation et des tourbillons, et du fait de la relative forte viscosité des interactions donnent un volume propre aux particules.

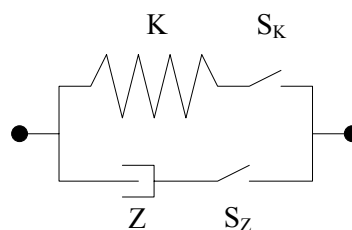


Figure 29 : Liaison entre particules de matière

La matière est disposée au dessus de l'entonnoir selon une forme rectangulaire. Les particules sont séparées par une distance supérieure au plus grand des seuils des interactions inter-particulaires.

#### 3.1.3 L'interaction environnement matière

La surface lisse du réceptacle est modélisée par des éléments de liaison, décrit à la Figure 30, qui sont régis par une interaction de type butée visco-élastique dont le seuil est choisi très supérieur à la taille de l'interaction inter-particulaires.

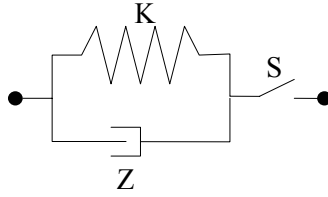


Figure 30 : Liaison entre l'environnement physique et la matière

### 3.1.4 Les caractéristiques retenues

#### a. Pâte

	Nombre	Type	Masse	Viscosité
Entonnoir	2	SOL	-	-
Réceptacle	3	SOL	-	-
Matière	300	MAT	0.05	0.2

	Raideur K	Viscosité Z	Seuil S
Matière-Entonnoir	500	0.2	0.960
Matière-Réceptacle	500	0.2	128.020
Matière-Matière 1	500	0	0.030
Matière-Matière 2	0	2	0.120

#### b. Fluide expansif

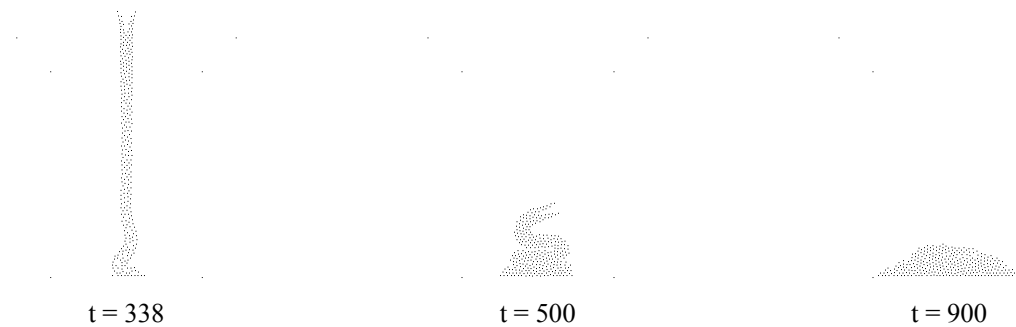
	Nombre	Type	Masse	Viscosité
Entonnoir	2	SOL	-	-
Réceptacle	3	SOL	-	-
Matière	300	MAT	0.05	0.2

	Raideur K	Viscosité Z	Seuil S
Matière-Entonnoir	500	0.2	0.955
Matière-Réceptacle	500	0.2	128.020
Matière-Matière 1	10	0	0.090
Matière-Matière 2	0	2	0.035

### 3.1.5 Validation des modèles

La matière s'écoule normalement de l'entonnoir (*<pâte, 1, écoulement>*) en formant quelques arches à l'amorce de cette dynamique (Elles se dilatent dans le sens du mouvement). Dès que le flux entre en contact avec le sol lisse, il s'étale (*<pâte, 2, étalement>*). Les particules en interaction directes avec celles recouvrant le sol forment les prémises d'un extremum (*<pâte, 4, extremum>*) qui s'accroît au fur et à mesure de l'accumulation de matière. Plus la quantité de matière s'écoulant de l'entonnoir est forte, plus les forces agissant sur les particules coincées entre le sol et l'écoulement (particules au dessus de l'extremum le plus haut) sont fortes. Il se crée alors un deuxième extremum au dessus et de sens opposé au premier avec un point d'inflexion entre les deux (*<pâte, 3, point d'inflexion>*). Ce processus se répète et génère un ensemble d'extrema tête-bêche et de points

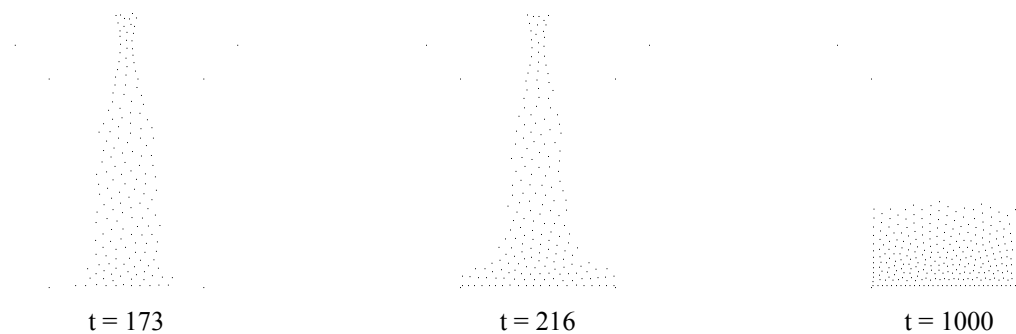
d'inflexion. Mais la matière glisse toujours sur le sol diminuant ainsi la distance entre deux extrema. Elle finit par se tasser sur elle-même (*<pâte, 5, tassement>*).



**Figure 31 : Modèle 2D de pâte**

Nous avons réalisé d'autres types de phénomènes que l'on a qualifié de pâte car ils utilisent le même type d'interaction inter-particulaire, mais ne présente pas les caractéristiques dynamiques de la pâte. Cependant la matière a une consistance similaire et peut être comprimée.

Pour le modèle dit de « fluide expansif », la compression est importante lors du passage de l'entonnoir. La matière s'expande ensuite (d'où son nom) en donnant l'impression de vouloir occuper un maximum d'espace dans les limites de sa cohésion. Tant que les parois du réceptacle ne sont pas atteintes, le fluide s'étale et se réorganise. Une fois les limites de l'espace atteintes le réaménagement n'a lieu que pendant un court instant et pour les particules proches du sol. Dès lors un mouvement de rotation apparaît dans le flux, et la matière se tasse à la manière de la pâte. Certains phénomènes de type solide comme les arcs-boutements créant les voûtes apparaissent lorsque les parois sont atteintes.



**Figure 32 : Modèle 2D de fluide expansif**

## 3.2. Conception du modèle 3D

### 3.2.1 L'environnement physique du modèle

Le passage à une dimension supérieure de l'environnement de la scène n'a pas posé de problème. La seule incertitude était de savoir si l'entonnoir laisserait passer suffisamment de matière à la fois (sans tomber dans l'excès) pour ne pas rompre le flux pour reproduire le phénomène obtenu en 2D. La solution proposée, à savoir positionner quatre modules SOL selon les coins d'un carré (dans le



plan X,Z) et en rajoutera au milieu des côtés pour colmater les brèches situées à la verticale des coins du réceptacle, s'avère être satisfaisante.

### 3.2.2 La matière

Les particules de matière sont disposées avec la même distance que pour la 2D entre chacune d'entre elles, et selon un parallélogramme rectangle dont les côtés ne dépassent pas de l'entonnoir. Les paramètres des interactions n'ont pas été modifiés.

### 3.2.3 L'interaction environnement matière

Il s'agit comme pour le modèle 2D de liaison visco-élastique à seuil unique.

### 3.2.4 Les caractéristiques retenues

Au cours de nos expérimentations, nous avons modifié certains paramètres : seuil de l'entonnoir et caractéristiques des liaisons inter-particulaires. Nous avons alors obtenu un modèle de gel en introduisant plus de viscosité, en diminuant l'élasticité et la valeur de son seuil, et en augmentant le seuil de l'entonnoir. En gardant les mêmes paramètres et en ajoutant de l'élasticité nous obtenons un modèle de goutte à goutte. Tous ces modèles nous ont semblé pertinents pour démontrer la validité du modèle générique de simulation physique de phénomènes naturels utilisant CORDIS-ANIMA.

Les noms donnés aux phénomènes sont basés sur les dynamiques obtenues, donc subjectifs.

#### a. Pâte

	Nombre	Type	Masse	Viscosité
Entonnoir	8	SOL	-	-
Réceptacle	5	SOL	-	-
Matière	900	MAT	0.05	0.2

	Raideur K	Viscosité Z	Seuil S
Matière-Entonnoir	500	0.2	0.960
Matière-Réceptacle	500	0.2	128.020
Matière-Matière 1	500	0	0.030
Matière-Matière 2	0	2	0.120

#### b. Gel

	Nombre	Type	Masse	Viscosité
Entonnoir	8	SOL	-	-
Réceptacle	5	SOL	-	-
Matière	300	MAT	0.05	0.2

	Raideur K	Viscosité Z	Seuil S
Matière-Entonnoir	500	0.2	0.995
Matière-Réceptacle	500	0.2	128.020
Matière-Matière 1	75	0	0.015
Matière-Matière 2	0	5	0.120

### c. Goutte à Goutte

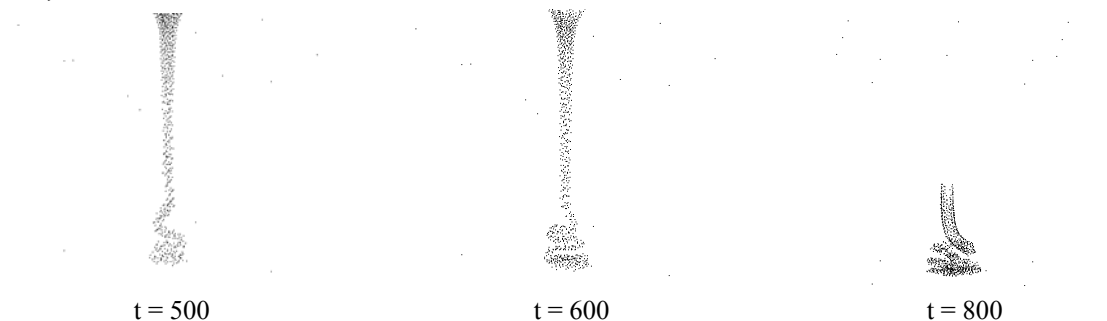
	Nombre	Type	Masse	Viscosité
Entonnoir	8	SOL	-	-
Réceptacle	5	SOL	-	-
Matière	200	MAT	0.05	0.2

	Raideur K	Viscosité Z	Seuil S
Matière-Entonnoir	500	0.2	0.995
Matière-Réceptacle	500	0.2	128.020
Matière-Matière 1	100	0	0.090
Matière-Matière 2	0	5	0.012

#### 3.2.5 Validation des modèles

Nous n'avons pas choisi d'étendre à la troisième dimension le modèle de fluide expansif car, de part les dynamiques peu marquées et peu diverses qui le caractérise, son intérêt ne nous a pas paru indispensable à la validation du modèle générique de simulation de phénomènes naturels.

Le modèle de pâte 3D se comporte conformément à son équivalent en 2D. Il présente les mêmes caractéristiques mais plus sur un seul plan. Le flux s'écoule de l'entonnoir de manière compact et dense malgré une division du flux en début de simulation. La matière est donc capable de se diviser puis de se recoller sans qu'on intervienne sur la structure physique de celle-ci. Les images sélectionnées montrent toutes les caractéristiques dynamiques de la pâte, mais pour pouvoir en détecter une, il faut analyser la simulation dans une direction non perpendiculaire au mouvement de celle-ci, ou tout au moins se trouver au dessus du sol.

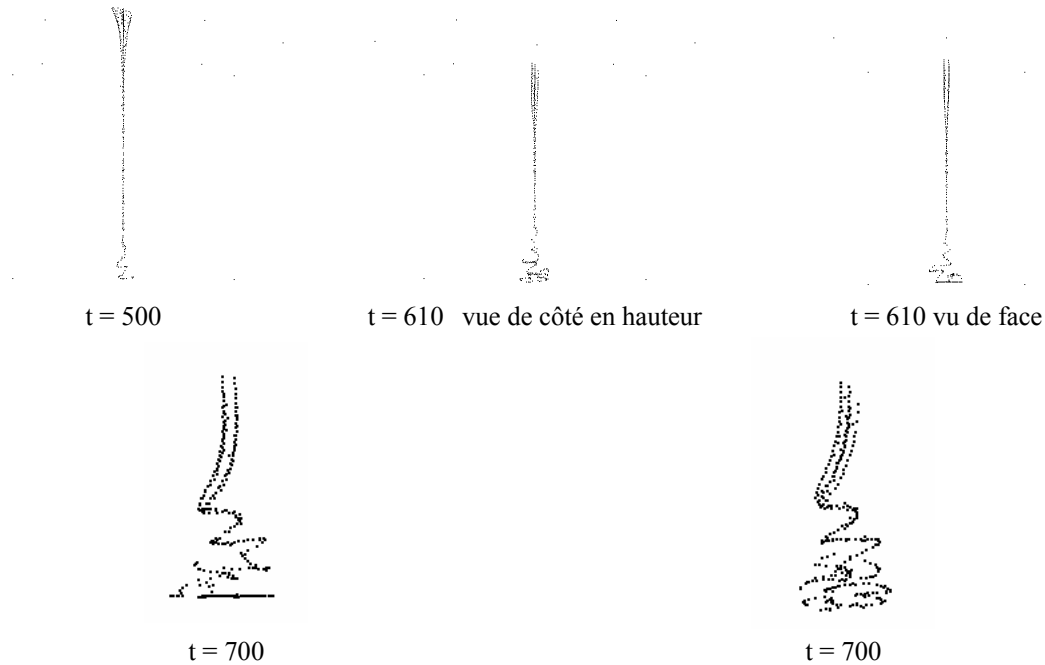


**Figure 33 : Modèle 3D de pâte**

Nous avons ensuite choisi d'expérimenter la simulation d'un fluide composé de moins de particules, en modifiant les paramètres des interactions de la matière ou de l'environnement physique.

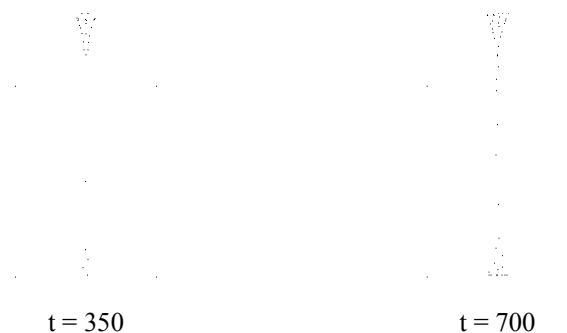
Nous avons obtenu un modèle que nous avons nommé « gel » tant son comportement est similaire au gel réel représenté sur les photographies du chapitre précédent. Le fluide est fortement déformable. Son flux est très fin. Avec peu de particules, nous avons modélisé les caractéristiques de la pâte de manière plus extrêmes dans la mesure où la matière forme un hélicoïde fin et ample dont le flux se compacte et s'étire. La densité du fluide est faible par rapport au modèle de pâte.

Le premier extremum apparaît avant que le fluide ne glisse sur le sol, laissant ainsi se répercuter sur les particules du flux une plus grande force qui atteint des particules plus hautes que sur le modèle 2D de pâte, ce qui accentue encore un peu plus les sinuosités des extrema. Le mouvement est réellement tridimensionnel et non symétrique, même lorsque le flux se tasse.



**Figure 34 : Modèle 3D de gel**

Pour le modèle suivant nous avons diminué la taille du goulot de l'entonnoir pour faire un modèle qui ressemble à un goutte-à-goutte de pâte. Est-ce que la formation des dynamiques de pâte est indépendante du nombre de particules du flux ? Lorsque le flux passe l'entonnoir il est morcelé en petits morceaux de matière qui une fois en contact avec le sol ou la matière qui le précède forme des extrema, des points d'inflexions puis se tasse tout en glissant sur le sol. Nous pouvons en conclure que les dynamiques de pâte sont plus marquées lorsque le flux est fin et long, et que le nombre de particules est peu significatif dans la modélisation d'un phénomène de pâte.



**Figure 35 : Modèle 3D de goutte-à-goutte de pâte**

## 4. Fluides turbulents

Les fluides turbulents ont des formes et des évolutions dynamiques complexes, non définissables par des techniques simples. Ils appartiennent aussi bien à l'état liquide que gazeux et incluent des phénomènes naturels comme la formation de nuages.

Les méthodes de modélisation existantes sont basées sur les équations de Navier-Stokes ou sur des méthodes propres au phénomène à reproduire. Avec cette classe de phénomène, nous abordons la modélisation de nouvelles dynamiques. Nous écarterons de notre champ d'étude les nuages dans leur effet d'amoncellement statique. Nous nous intéressons à la modélisation de tourbillons de Kelvin-Helmholtz, à la formation d'allée de von Karman, qui apparaissent dans l'état liquide, gazeux ou même dans certains amoncements nuageux.

Notre démarche de conception reste inchangée. Nous commencerons par concevoir l'environnement de la matière, puis la matière dont l'étude nous intéresse. De même que pour les sables, nous reprendrons les travaux effectués au préalable dans le laboratoire ([vapillon95]) pour réécrire le modèle et avoir une base homogène et que nous maîtrisons bien d'effets à visualiser.

Les fluides sont modélisés par un ensemble de particules potentiellement toutes en interaction. C'est donc encore une fois par agglomérat que la matière sera structurée.

### 4.1. Tourbillon de Kelvin-Helmholtz

#### 4.1.1 Mécanisme d'injection et de recyclage

Nous devons simuler des écoulements relativement rapides et d'assez longue durée qui font intervenir un grand nombre de particules. Le temps de calcul et l'occupation mémoire étant directement liés à la quantité de MAT utilisés, nous utilisons un système d'injection et de recyclage pour en limiter le nombre (Figure 37)

Les particules ont deux états possibles : en coulisse ou en scène. Celles en coulisses sont inactives, en réserve tandis que celles en scène sont actives et participent pleinement à la simulation.

Les particules sont émises des coulisses vers la scène à une fréquence  $f$  et en un point précis, l'injecteur. Elles sont réabsorbées vers les coulisses lorsqu'elles dépassent les frontières de la scène et sont réintroduites au niveau de l'injecteur.

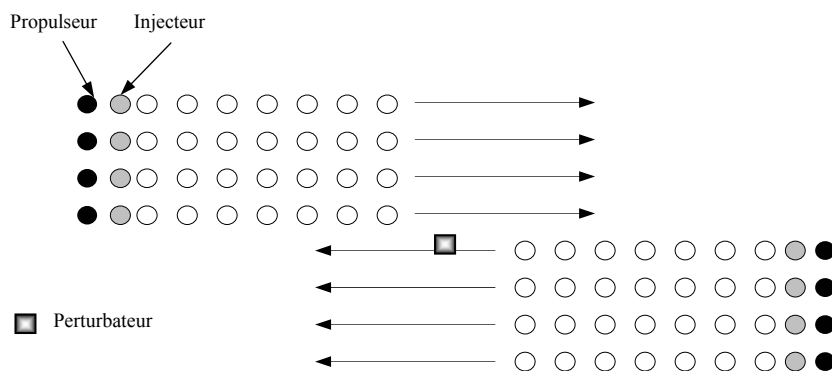
Une butée élastique, le propulseur, est placée proche de chaque injecteur pour repousser les particules et ainsi leur donner une vitesse initiale. C'est donc la distance injecteur-propulseur qui détermine leurs vitesses – module et direction. Pour simuler des écoulements d'une certaine largeur, plusieurs injecteurs sont placés côte à côte. Le propulseur, l'injecteur comme les coulisses sont modélisés par des modules SOL.

#### 4.1.2 Les masses ponctuelles : les MAT

On dispose de deux séries d'injecteurs-propulseurs, opposés et décalés verticalement projetant leurs particules l'une vers l'autre (Figure 36)

La différence entre le seuil d'interaction inter-particulaire et l'espace entre les couches de particules est importante. Elle détermine si les particules de deux couches adjacentes sont en interaction dès leur injection ou non. De même, la différence entre la vitesse des particules à l'injection (réglée par le propulseur) et la fréquence d'injection détermine si les particules consécutives d'une même couche sont en interaction ou non.

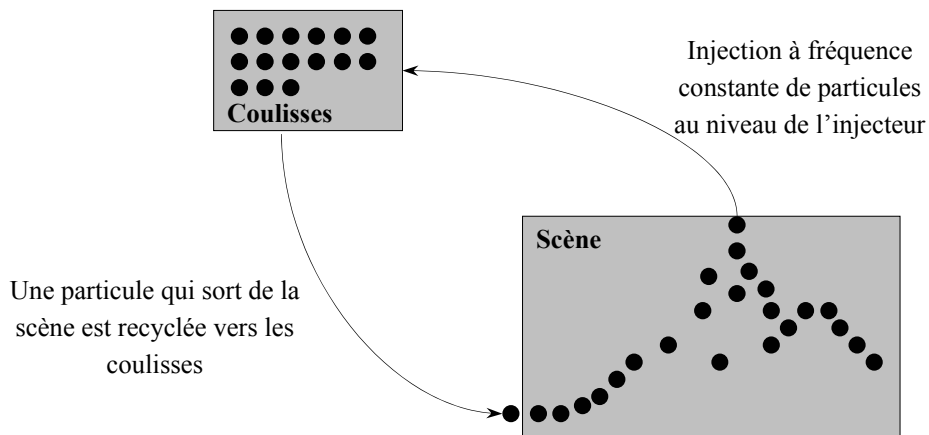
#### 4.1.3 Interactions entre particules



**Figure 36 : Modèle de couche simulée**

Les particules sont liées deux à deux par une interaction purement visqueuse. Lorsque la distance entre deux particules est supérieure à ce seuil, les particules sont indépendantes, tandis que si elle est inférieure, une force visqueuse non nulle circule entre elles.

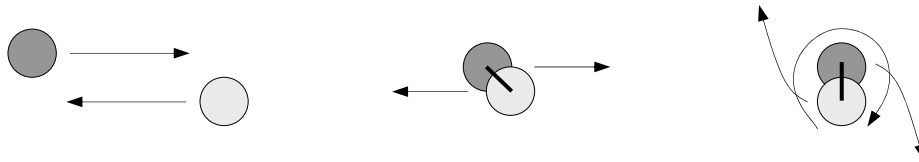
Les particules d'une même couche sont reliées dès le premier instant de la simulation, alors que les particules de couches distinctes le sont lorsque l'écoulement est suffisamment formé et le perturbateur non encore entré en action.



**Figure 37 : Principe de l'injection et du recyclage**

Dans certaines conditions de vitesse, ces interactions sont susceptibles de provoquer des mouvements de rotation : deux particules de vitesses différentes entrant en contact entament un mouvement de rotation l'une autour de l'autre (Figure 38).

Chaque particule est reliée à un injecteur et un seul. C'est par cet injecteur que la particule reviendra après être sortie de la scène.



**Figure 38 : Interaction visqueuse source de rotation**

#### 4.1.4 Le perturbateur

La perturbation déclenchant le tourbillon de Kelvin-Helmholtz est produite par une masse en interaction de butée élastique avec l'ensemble de la matière. Le MAT est placé de manière à repousser les couches centrales vers celle de direction opposée. Comme l'ont montré les résultats des expérimentations effectuées par [vapillon95], lorsque l'écoulement est établi et qu'il est dans un état d'équilibre instable, le déclenchement du tourbillon peut s'effectuer de manière simple par perturbation élémentaire comme un simple choc élastique ponctuel.

#### 4.1.5 Caractéristiques et Validation du modèle

Ces deux modèles ont été réalisés sans viscosité ni gravité ambiante.

##### a. Modèle à deux fois 2 injecteurs

	Nombre MAT	Masse MAT	K	Seuil K	Z	Seuil Z	Espace
Couches gauche	2x100	0.05	0	0	5.0	0.035	0.04
Couches droite	2x100	0.05	0	0	5.0	0.035	0.04
Liaisons différents inj.	-	-	0	0	10.0	0.035	-
Perturbateur	1	0.5	70	0.02	0	0	-
Force injecteur	-	-	8.0	-	-	-	-

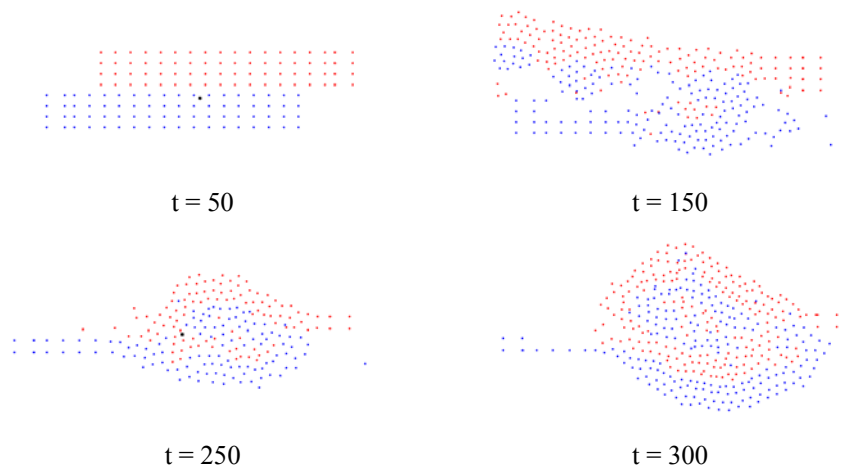
##### b. Modèle à deux fois 5 injecteurs

	Nombre MAT	Masse MAT	K	Seuil K	Z	Seuil Z	Espace
Couches gauche	5x100	0.05	0	0	10.0	0.035	0.04
Couches droite	5x100	0.05	0	0	10.0	0.035	0.04
Liaisons différents inj.	-	-	0	0	10.0	0.035	-
Perturbateur	1	0.1	10	0.02	0	0	-
Force injecteur	-	-	5.0	-	-	-	-

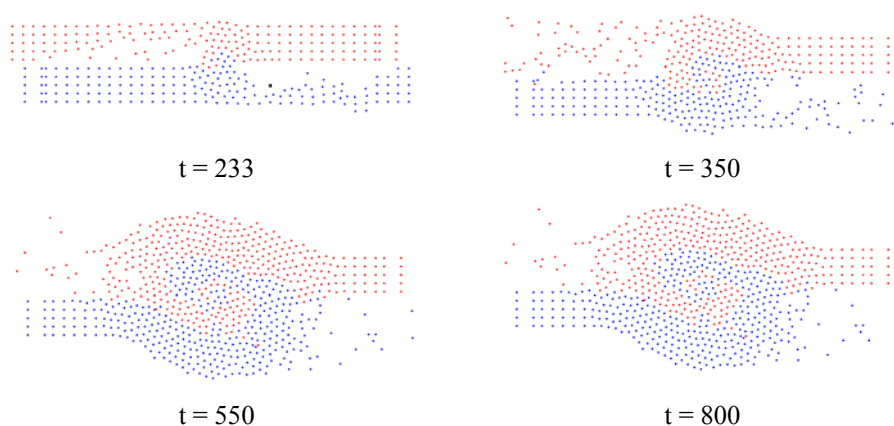
### c. Conclusion

Mis à part les paramètres du perturbateur, les deux modèles diffèrent par deux caractéristiques : la force de l'injecteur (le perturbateur a été conçu en fonction de cette force), la viscosité des interactions inter-particulaires d'une même couche. Le modèle à cinq injecteurs utilise toujours les mêmes types de liaisons tandis que l'autre s'appuie sur une viscosité des particules des couches plus faibles que celle entre les couches. Cela entraîne une mise en interaction et un début de tourbillon de Kelvin-Helmholtz plus chaotique lorsque le flux ne possède pas les mêmes caractéristiques entre les particules. Cela est visible sur la Figure 39, pour l'image 150.

Le modèle à cinq injecteurs (Figure 40) passe par toutes les phases caractérisant la formation d'un tourbillon de Kelvin-Helmholtz. Un modèle avec une force d'injection pas trop forte et une homogénéité des interactions inter-particulaires convient mieux à la formation d'un vortex.



**Figure 39 : Modèle à quatre injecteurs**



**Figure 40 : Modèle à cinq injecteurs**

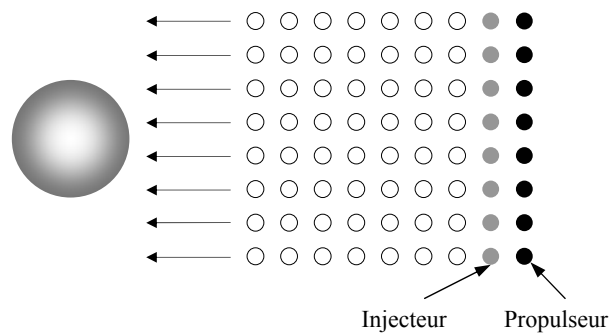


Figure 41 : Obstacle dans le sillage d'un écoulement de fluide

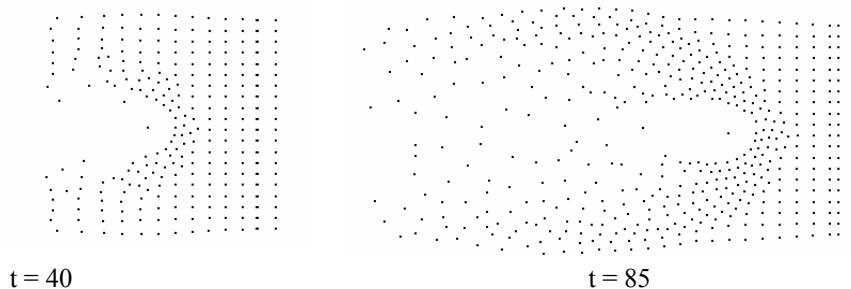
## 4.2. Allée de von Karman dans le sillage d'un obstacle

Nous n'avons pas réétudié le modèle que nous présentons dans ce paragraphe réalisé par le laboratoire lors du stage de DEA d'A. Vapillon ([vapillon95], [luciani95b]). Nous le présentons ici pour démontrer que l'existence d'un modèle générique pour la modélisation de l'ensemble des états de la matière est possible, généricité sur laquelle nous nous appuyerons pour recherche de la même façon un procédé de visualisation générique.

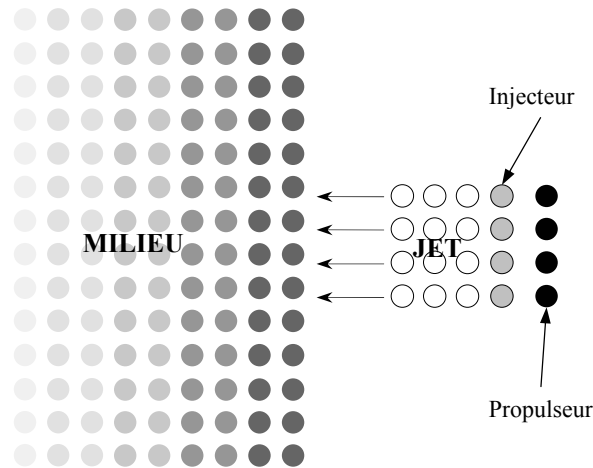
Ce modèle est similaire à celui des couches établies pour modéliser une allée de von Karman. Le perturbateur n'existe pas, et une seule couche de fluide est considérée. Mais la structure du reste du modèle est parfaitement identique (système injecteur-recyclage, structuration de la matière). Sur le parcours de la matière un obstacle rigide (un module SOL, et une interaction obstacle-matière élastique et raide). La composition du modèle est présentée à la Figure 41.

### 4.2.1 Caractéristiques et Validation du modèle

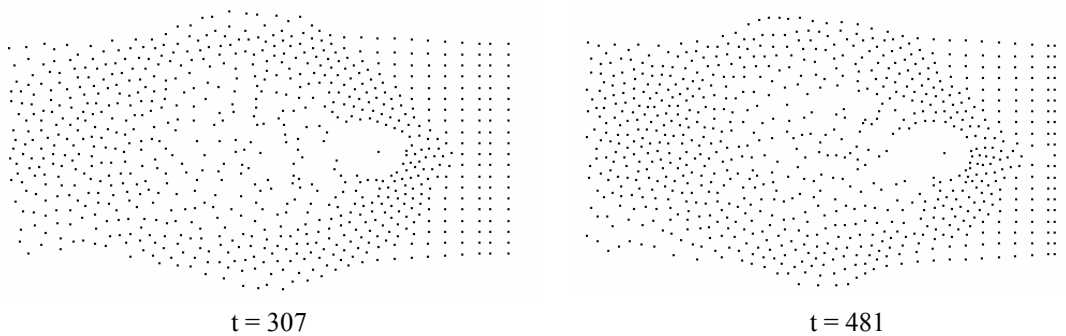
	Nombre MAT	Masse MAT	K	Seuil K	Z	Seuil Z
<b>Jet</b>	20x75	0.05	0.15	0.09	5.0	0.035







**Figure 43 : Modèle de colonne de fumée**



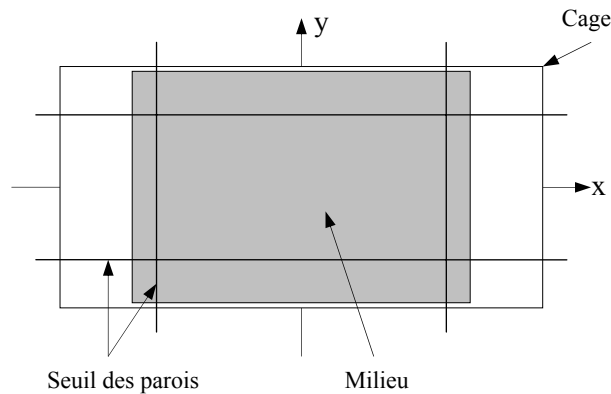
**Figure 42 : Modèle d'allée de von Karman**

Il n'est pas évident de constater la formation d'allée de von Karman sur des images fixes où chaque module du modèle est représenté par un point matériel. Lorsque l'on regarde l'animation, la matière forme des vortex derrière l'obstacle et de part et d'autres de l'axe des abscisses. Ils se déroulent pour laisser place à la formation d'autres volutes. Le modèle reproduit bien une allée de von Karman.

Le bord d'attaque de l'obstacle subit une élasticité compressée de la matière qui arrive à une vitesse constante. Cette compression augmente la densité de particules dans l'espace entourant l'obstacle. Une force d'injection plus faible (elle est de 7 pour ces images) engendrait des volutes plus légères.

### 4.3. Volute de fumée et allée de von Karman

La « colonne » de fumée est un modèle au comportement riche, mêlant écoulements laminaires, allées de von Karman, tourbillons de Kelvin-Helmholtz. La formation de ces dynamiques nous oblige à considérer deux milieux : la matière, et le milieu dans lequel elle évolue.



**Figure 44 : Environnement physique du modèle de colonne de fumée**

#### 4.3.1 L'environnement physique

Pour délimiter l'espace du modèle, on utilise une cage qui délimite la scène (pour le système d'interaction-recyclage) et des parois pour canaliser le milieu.

#### 4.3.2 Le milieu : particules et les interactions

Le milieu est un ensemble de particules disposées en une grille régulière de 20 sur 13 confinée par les parois. Les paramètres de liaisons visco-élastiques reliant ces particules ont été choisis de manière à ce qu'aucun mouvement ne se produise lorsqu'il n'y a pas de force extérieure qui s'y applique. La distance séparant deux particules de la grille est le double de celle séparant deux injecteurs.

#### 4.3.3 Le jet : particules et interactions

Le jet est produit par quatre injecteurs qui projettent les particules à la même vitesse. La dimension des particules du jet par rapport aux particules du milieu influe sur le comportement du modèle. Si les particules du milieu sont plus grandes alors celles du jet ne pourront former un fluide homogène. Pour que les volutes se forment, la taille des particules du milieu est identique à celles du jet.

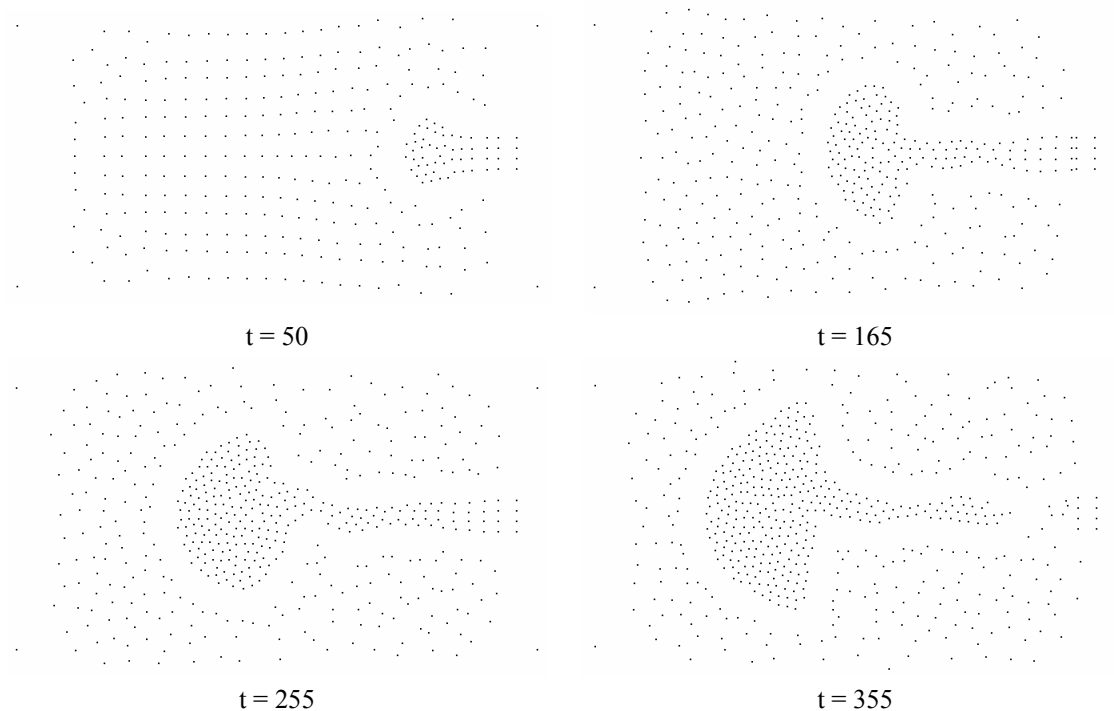
La viscosité du milieu influence la déformabilité du jet et l'amortissement du milieu, donc le comportement du jet. Les particules du jet sont connectées si elles sont issues du même injecteur, ou d'un autre injecteur (liaison type agglomérat). La force de l'injecteur est du même ordre de grandeur que celle du modèle à couche établie.

#### 4.3.4 Liaisons Jet-Milieu

Pour connecter ces deux modules nous nous servons de liaisons visco-élastiques aux paramètres identiques à ceux du milieu.

#### 4.3.5 Caractéristiques et Validations du modèle

	Nombre MAT	Masse MAT	K	Seuil K	Z	Seuil Z	Espace
<b>Milieu</b>	20x13	0.05	0.15	0.18	5.0	0.035	0.08
<b>Jet</b>	4x80	0.05	0.15	0.09	5.0	0.035	0.04
<b>Milieu-Jet</b>	-	-	0.15	0.18	5.0	0.035	-



**Figure 45 : Colonne de fumées**

Comme le milieu est moins dense que le jet, il se déforme sous l'interaction de la matière injectée, qui se déforme elle aussi. En effet le milieu est deux fois moins dense que le jet. La formation des tourbillons de Kelvin-Helmholtz et des allées de von Karman est difficilement décelable puisque les particules se concentrent dans un espace restreint. Ce modèle n'a pas le comportement exact d'une fumée légère s'élevant dans le ciel car la matière est trop cohésive limitant la génération de caractéristiques dynamiques (un jet purement visqueux serait préférable *a priori*). Ces propos font suite à des réflexions sur la manière d'analyser et de nommer les modèles réalisés. L'étiquetage n'est pas formel : il ne repose sur aucune notion physique propre à la dynamique mais sur une impression, une sensation mise en parallèle avec nos connaissances du phénomène dans la nature. Etablir une connexion entre le monde réel et celui de la simulation physique nous permettrait d'être catégorique sur la classification des modèles.

#### 4.4. Discussion sur les modèles de fluides turbulents

Les trois modèles présentés montrent la difficulté de la conception de phénomènes de fluides turbulents par modèle physique particulière. Le nombre de paramètres interdépendants à gérer est important. Ce que nous pouvons déduire de ces trois études est qu'une matière purement visqueuse semble être plus appropriée pour modéliser les effets tourbillonnaires.

Nos recherches nous ont amenés à réfléchir à la conception de modèle tridimensionnel. Nous nous y sommes essayés, mais les résultats ont été peu convaincants. Qu'est ce qu'un fluide turbulent 3D ? Un ensemble de modèles 2D dont les dynamiques sont plus marquées à mesure que l'on approche du centre du flux, ou tout autre chose.

D'après l'expérience du passage en 3D des modèles de pâtes, de sable, la nature de la matière ainsi que sa structure ne doit pas être modifiée. Mais peut-on en conclure qu'un phénomène réalisé en 2D n'est rien d'autre qu'une tranche de son équivalent 3D ? Il s'agit d'un sujet de discussion très actif en physique et nous ne pouvons en ce qui nous concerne être catégorique, en particulier à la seule vue des résultats de nos simulations. En ce qui nous nous concerne, nous considérerons un modèle 3D comme constitué d'un ensemble de tranches espacées exactement comme les lignes du modèle en 2D, le perturbateur étant introduit au centre de ces tranches.

Mais comment analyser un tel modèle avec les méthodes de visualisation dont nous disposons ? Ce problème sera à nouveau d'actualité pour la colonne de fumée et les allées de von Karman dans le sillage d'un obstacle.

Nous ne nous sommes pas engagées plus en avant dans la simulation de fluides turbulents, d'une part parce qu'il s'agit d'un sujet en soi avec ses propres difficultés de modélisation en 3D et d'autre part parce que les résultats que nous avons sont suffisants pour notre travail sur la visualisation. Mais nous chercherons à tenir compte de ces dynamiques lors de création de visualisation, car ces modèles sont des cas exemplaires de modèles de type « agglomérats ».

## 5. Simulation à différentes fréquences

La simulation à une fréquence de 1050Hz est la fréquence de simulation utilisée en standard dans le laboratoire pour simuler les phénomènes physiques non audibles. Elle se situe entre des phénomènes basse-fréquence (50Hz pour l'image) et haute-fréquence (44,1 KHz pour le son). Elle a été choisie car elle correspond à un multiple de la fréquence habituellement utilisée en animation (25Hz) et à un sous-multiple de la fréquence nécessaire pour produire des phénomènes audibles.

On est certain que les phénomènes naturels auxquels on s'intéresse ici ne nécessitent pas une fréquence supérieure à celle-ci pour être correctement restitués. Pour ces types de phénomènes, cette fréquence est empiriquement une borne supérieure, choisie pour mettre au point le modèle en toute sécurité. Cette fréquence impose le calcul des positions et des forces des particules du modèle tous les  $1/1050$  de secondes (0.000952 s). Pour la visualisation (observation phénoménologique visuelle), on ne stocke que les positions tous les  $1/25$  de secondes (0.04 s). Donc pour obtenir l'état du modèle pour l'image  $t$ , on a calculé  $42*t$  fois les valeurs pertinentes du modèle, soit 31500 boucles de calculs lors

de la simulation pour une animation de 30 secondes. Le temps de calcul d'une boucle dépend du nombre de particules et de liaisons qui composent le modèle. Le calcul de certains modèles volumineux en nombre de particules a pris près de 10 heures sur une Octane pourvue d'un processeur R10000 à 175 MHz pour une animation de 40 secondes.

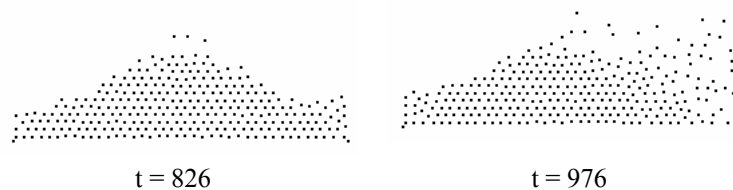
Il se peut que le modèle reste valide à des fréquences de simulation inférieure, ce qui nous permettrait d'accélérer la réalisation de modèles et l'étude sur un champ paramétrique plus large. Nous ne pouvons le savoir qu'empiriquement en utilisant la simulation car il n'existe aucune évaluation de la bande passante fréquentielle de ce type de phénomène. Nous avons donc simulé le modèle de sable à sol lisse 2D ainsi que le modèle de gel 3D à différentes fréquences. Deux frontières sont à déterminer : la fréquence en dessous de laquelle la simulation ne converge plus, la (les) fréquence(s) en dessous de laquelle (desquelles) on perd les caractéristiques dynamiques. Ce choix de phénomènes s'est fait pour disposer d'un maximum de dynamiques, qui nous permettrait de savoir si la formation de certaines d'entre elles est irréalisable pour certaines fréquences. Le Tableau 4 expose les résultats des expérimentations en fonction des modèles et des fréquences de simulation.

<b>Fréquence</b>	<b>Modèle de Sable</b>	<b>Modèle de gel</b>
250 Hz	Dès les premiers instants et l'entrée en contact des particules avec l'entonnoir, le modèle diverge (explosions permanentes sans moment d'accalmie).	La simulation est fluide et les dynamiques correctes, mais le dépôt de matière dans l'entonnoir est animé de soubresauts dus à une légère divergence des calculs lorsque la matière est dense et compressée.
375 Hz	-	La vibration de la matière dans l'entonnoir se produit toujours, mais les dynamiques caractéristiques sont toutes là.
500 Hz	Même divergences qu'à 250Hz mais plus tardivement et moins brusquement. La force de l'explosion (divergence) est proportionnelle au nombre de particules en interaction (en contact). Cependant entre chaque souffle les particules retombent lentement comme après un feu d'artifice.	A cette fréquence, on ne constate plus de divergence du modèle.
650 Hz	L'écoulement se déroule normalement malgré quelques déséquilibres dans l'entonnoir. Le tas formé n'est pas stable et peut atteindre un seuil qui le fait basculer dans la divergence, mais l'on retrouve des effondrements, des avalanches de surface et des lignes de force.	-
687 Hz	Le modèle est plus stable mais de petites explosions de surface amène le tas à disparaître.	-
706 Hz	Le tas est correct, vibre de plus en plus à mesure que l'on s'approche de la surface. Les vibrations sont moindres à l'intérieur, car contenues par la masse de la matière qui contraint le mouvement des particules. Toutes	-

	les dynamiques sont là. Ce modèle ne présente de divergences qu'en surface du tas.	
725 Hz	Les soubresauts sont légers autant en surface que dans le tas qui répond aux caractéristiques d'un matériau granulaire.	-
800 Hz	Le modèle est quasiment stable et contient toutes les dynamiques obtenues à 1050 Hz. Il peut être considéré comme satisfaisant.	-

**Tableau 4 : Observations pour différentes fréquences de simulation**

Les dynamiques peuvent être obtenues pour une fréquence moitié de celle actuellement utilisée, mais dès que des forces élastiques entrent en jeu cette fréquence ne peut être diminuée que de 25%. Nous n'avons pas réalisé cette étude sur d'autres modèles comme les objets rigides ou très déformables. Cette étude d'optimisation temporelle, qui se justifie lorsque l'on ne dispose pas de la bande passante des objets physiques de référence, est à faire au cas par cas. Nous ne pouvons pas en déduire si 1050Hz est *la* fréquence de simulation ou si l'utilisation d'une fréquence moindre entre 850Hz ou 900Hz serait suffisante pour simuler n'importe quel objet, phénomène. Si l'on simule à 850Hz, on effectue 19% de boucles de calculs en moins.



**Figure 46 : Modèle de sable 2D simulé à 687Hz**

## 6. Conclusion

Tous ces modèles ont été réalisés à l'aide de CORDIS-ANIMA en utilisant le précepte proposé dans [Luciani00] de modèle physique particulaire générique pour la simulation de phénomènes naturels.

Nous retenons des modèles de sable l'importance de la structure de l'environnement et notamment du sol, des modèles de pâte la formation des dynamiques avec peu de particules et la complexité des mouvements, des modèles à fréquence inférieure la nécessité de simuler à une fréquence permettant l'apparition des dynamiques, des modèles de fluides turbulents la difficulté de concevoir et visualiser des résultats faisant intervenir plusieurs flux de matière.



## Chapitre 6 Analyse des données de simulation

---

Maintenant que nous avons réalisé un ensemble de modèles de phénomènes naturels présentant différentes caractéristiques, nous devons étudier les différentes manières de les habiller. Visuellement, nous les reconnaissons grâce à leurs invariants, leurs dynamiques. Ces signes distinctifs sont détectables par les formes ou sous-formes que prennent un ensemble de particules. Nous cherchons à extraire ces spécificités de nos fichiers ne contenant que des positions pour s'en servir pour la visualisation.

Il s'agit de définir le flux et ses caractéristiques en ne prenant en compte que les masses ponctuelles de la matière. Nous sommes confrontées à un premier problème pour les modèles avec le système d'injection-recyclage qui, de part la présence de coulisses, vont fausser les calculs de vitesse, d'accélération. Nous choisissons de traiter de manière globale toutes ces simulations, *i.e.* de ne pas implémenter de programmes particuliers pour ceux posant des difficultés de gestion de part leur structure. Cependant on peut les restructurer en considérant plusieurs flux (groupes de particules aux destinées communes).

Lors de la simulation, nous ne récupérons que les positions des masses ponctuelles au cours du temps, mais disposer de leurs vitesses, accélérations, forces, ou toute autre donnée peut s'avérer fort utile pour la spécification de la forme exacte de la matière. Les forces sont impossibles à calculer à partir de positions sans connaître la structure du modèle étudié. Cette structure (type et paramètres des interactions de l'agglomérat) nous avons choisi de l'ignorer. Ainsi le processus de création peut être réalisé par différentes personnes ne communiquant point. Nous calculons les vitesses, accélérations et positions relatives (distance entre particules) pour chaque modèle. Dans un premier temps, nous l'avons aussi fait pour les vitesses et accélérations relatives mais ces données se sont avérées être extrêmement lourdes à gérer et à stocker. De plus elles ne semblaient pas être indispensables dans la mesure où de ce type d'information pouvait être retrouvé par l'analyse des vitesses et accélérations.

Les positions relatives sont une manière de caractériser le comportement interne du flux et de le structurer. La vitesse indique la direction et le sens du mouvement des particules. On sait que si toutes les vitesses sont homogènes nous nous trouvons en présence d'un solide. L'accélération révèle la diminution ou la augmentation de vitesse qui peut être consécutive à une collision. Etablir une



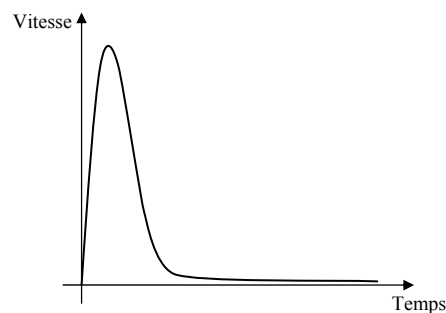
relation entre les écoulements de fluides CORDIS-ANIMA et le monde réel serait une preuve tangible sur la nature du modèle.

Le présent chapitre est organisé comme le précédent. Nous exposons nos résultats suivant les modèles du plus solide ou plus gazeux.

Nombre des images suivantes figurent la vitesse des particules par un point dont la taille représente la norme de celle-ci, et la couleur sa direction. Les accélérations sont symbolisées par des vecteurs. Il peut arriver que ce soit l'inverse, mais nous le précisons alors.

## 1. Modèle de croissance de cellules végétales

La courbe de vitesses en fonction du temps est typique de la liaison élastique reliant les particules du pourtour au barycentre de la cellule. Le mouvement premier de croissance est brusque, puis diminuer au fur et à mesure que l'énergie emmagasinée initialement s'épuise. Tout au long de la simulation les vitesses sont dirigées vers les positions finales des particules, *i.e.* dans l'axe de la liaison interne. Comme nous le constatons sur l'animation, les particules ne sont pas immobiles lorsque celle-ci se termine.



**Figure 47 : Vitesse des particules de cellule lors de la simulation**

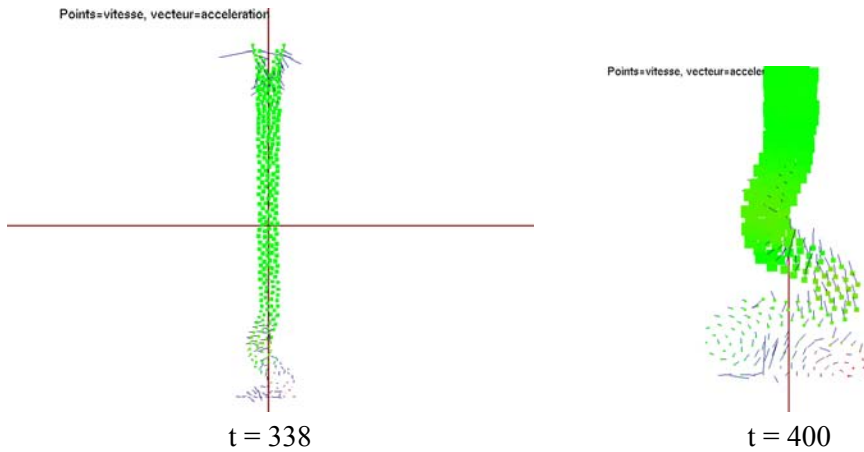
La croissance est bien un phénomène qui consomme de l'énergie, et la forme finale de la cellule est le résultat du processus de croissance pour une structure interne donnée.

## 2. Modèle de fluide visqueux

### 2.1. Modèle de pâte

Les vitesses et accélérations des particules témoignent de la cohésion de la matière.

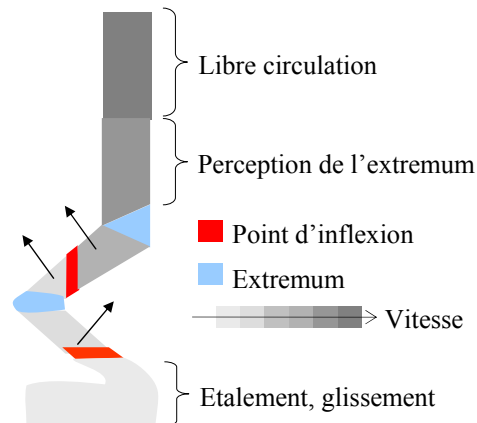
Les masses ponctuelles en contact avec l'entonnoir ont une vitesse tangente à l'entonnoir, et une accélération globalement dirigée vers le centre de la matière. Lorsqu'elles dépassent le goulot, leur vitesse se dirige vers le sol (attraction de la liaison sol-particule et de la gravité des MAT).



**Figure 48 : Vitesses (point) et accélération (vecteur) de la pâte 2D (indication de direction : rouge pour l'axe X, vert pour Y, bleu pour Z)**

La limite inférieure du flux agit alors comme un moteur qui amplifie le mouvement vers le sol. Lorsqu'elles atteignent le sol, leurs accélérations s'opposent à l'attraction du sol (collision), et leurs vitesses diminuent à mesure qu'elles se rapprochent du sol.

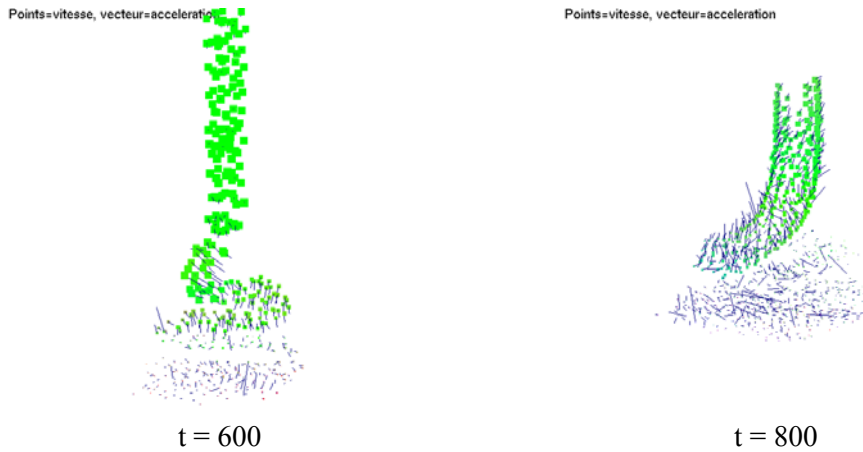
Les prémices des caractéristiques dynamiques sont perceptibles en analysant les vitesses et accélérations des particules de la matière. Un extremum se caractérise par un sens différent des accélérations des particules en dessous et au dessus de lui : elles forment un V (Figure 49). Le point



**Figure 49 : Structuration des données vitesses et accélérations**

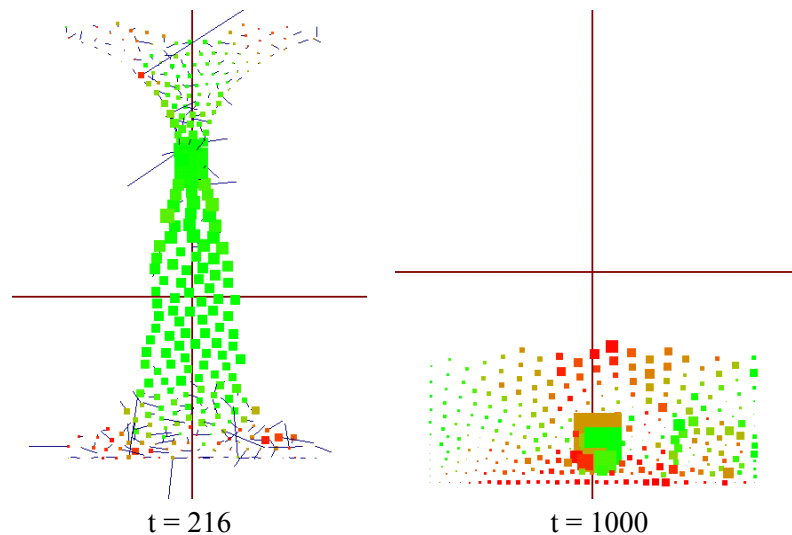
d'inflexion se situe entre deux extrema au niveau où les normes de vitesses se distinguent. Le tassement entraîne une forte diminution de la vitesse, et une accélération dirigée vers le centre du pli. L'accélération des extrema en forme de V se transforme en un mouvement circulaire. En fin de simulation, pour le phénomène de tassement des différentes portions entre extrema, les vitesses sont toutes dirigées vers le sol mais de manière non perpendiculaire, plutôt en direction des extrémités du flux sur le sol comme si elles suivaient les pentes d'une colline.

L'analyse du modèle 3D s'est avérée plus difficile, mais l'ensemble des analyses avancées en 2D semble se confirmer.



**Figure 50 : Vitesses (point) et accélération (vecteur) de la pâte 3D**

## 2.2. Modèle de fluide expansif



**Figure 51 : Vitesses (point) et accélération (vecteur) du fluide expansif 2D**

Le stationnement dans l'entonnoir engendre, pour les particules du flux, une accélération en direction de l'intérieur du flux. Lors du passage du goulot, le fluide se comprime fortement pour se détendre une fois l'influence de l'entonnoir dissipée et ainsi s'écouler dans un mouvement à vitesse homogène constante.

Le contact avec le sol est source d'accélération horizontale, mais principalement pour les particules extérieures qui entraînent les autres dans un mouvement de glissement sur la structure lisse. Cette caractéristique d'accélération forte se propage aux couches supérieures non en interaction directe avec le sol.

Le modèle de fluide expansif n'a pas de caractéristique complexe décrivant sa forme. Cette constatation, faite lors de simulation, se confirme ici. Mais l'on remarque qu'une fois l'écoulement terminé les particules internes cherchent toujours à s'étaler sur le sol, alors que celles proches des

parois ont tendance à se tasser sur leurs congénères. Ces actions de direction opposée favorisent l'apparition de mouvements tourbillonnaires de réorganisation de la matière en vue d'une répartition homogène et régulière des particules.

### 2.3. Modèle de gel

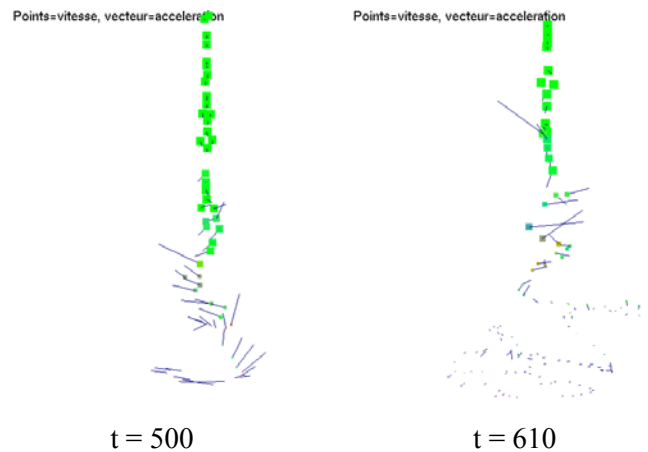


Figure 52 : Vitesses (point) et accélération (vecteur) du gel 3D

Le point d'inflexion est une région située entre deux groupes de particules aux vitesses différentes, l'extremum se situe entre deux groupes de particules aux vitesses identiques mais aux accélérations symétriques par rapport à la verticale. Nous ne sommes pas en mesure d'en dire plus car le flux de matière est fin, les groupes sont plus difficiles à détecter, et le mouvement est tridimensionnel. Les teintes de couleurs caractérisent la direction de la vitesse, mais l'on constate que les groupes sont formés d'une déclinaison de couleur ce qui signifie qu'entre deux particules d'un même groupe le vecteur vitesse a tourné autour d'un axe vertical central au flux. Les accélérations font de même mais vers l'extérieur de l'hélicoïdale.

### 2.4. Modèle de goutte-à-goutte

Une goutte est composée de très peu de particules, donc les caractéristiques en terme de vitesse et d'accélération ne sont pas attribuables à un type de phénomène en particulier. Nous pensons qu'il s'agit d'un goutte-à-goutte de pâte car nous connaissons les paramètres, et que nous détectons des dynamiques de ce phénomène à partir des positions des masses ponctuelles, mais rien du point de vue de données calculées ne nous permet de l'affirmer.

## 3. Modèle de sable

### 3.1. Modèle de sable à sol rugueux

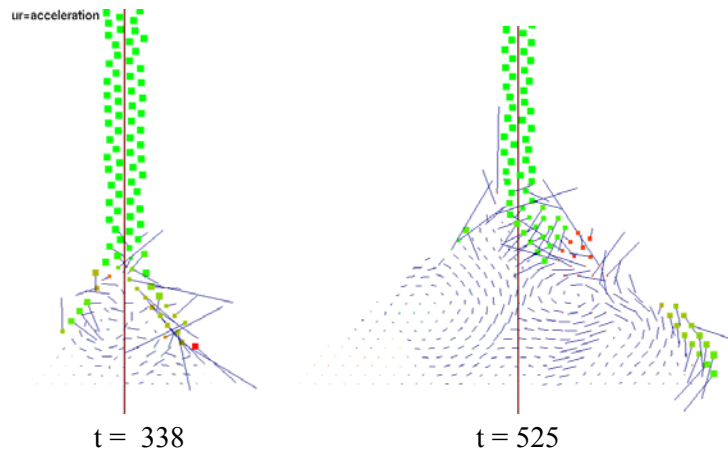


Figure 53 : Vitesses (point) et accélération (vecteur) du sable 2D à sol rugueux

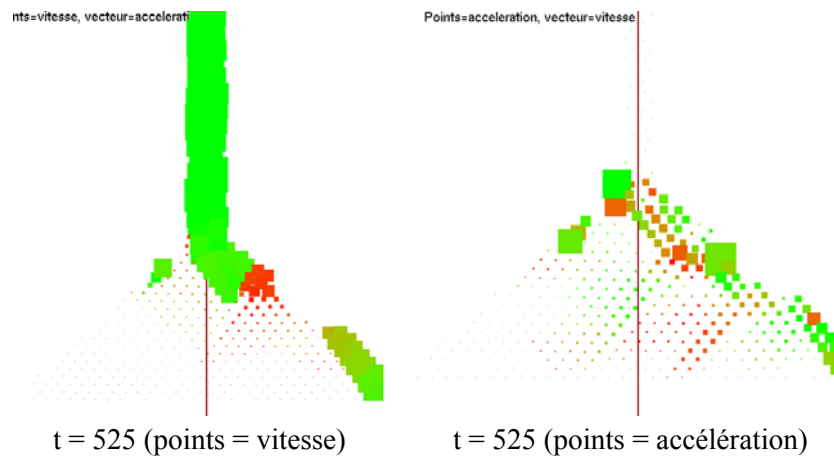


Figure 54 : Vitesses (vecteur) et accélération (point) du sable 2D à sol rugueux

Nous ne décrivons pas les instants de stationnement dans l'entonnoir ainsi que l'écoulement car il est en tout point identique au modèle de pâte. Le tas se forme par ajout successif de particules. Les vitesses au centre du tas sont quasiment nulles, tandis que les particules participant à son élargissement par glissement sur ses pentes ont une vitesse dirigée vers l'extérieur du tas et vers le sol, et une accélération uniquement vers le sol. Les particules prises dans le mouvement d'avalanche externe ont une vitesse et une accélération forte dirigée vers l'extérieur du tas, plutôt vers le haut.

Notre analyse nous a amené à spécifier les dynamiques déclenchant les caractéristiques du phénomène. Tout d'abord le tas se forme par un écoulement et des empilements successifs de particules de matière (seules les particules en contact avec l'extérieur possèdent une vitesse verticale et une accélération vers le centre du tas). Le tas atteint alors une maturité source de stabilité en son intérieur jusqu'à ce que sa pente l'oblige à se restructurer sous l'impulsion de mouvements tourbillonnaires, ou l'apparition de sous-structures distinguées par leur vitesse, leur accélération. Lorsque les avalanches internes ont eu lieu, le tas se stabilise à nouveau et le cycle d'augmentation de

taille et de réorganisation se poursuit. Les zones séparant deux sous-structures sont les arches décrites au chapitre 4.

L'évolution du tas n'est pas symétrique et les avalanches internes n'impliquant pas l'ensemble des sous-structures sont unilatérales.

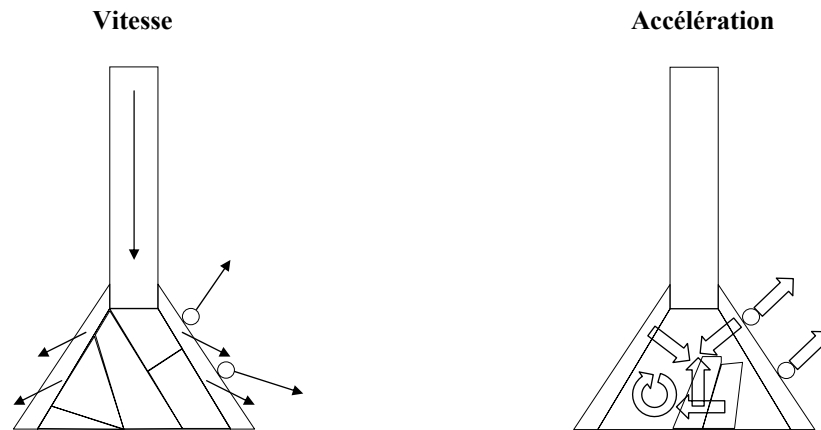


Figure 55 : Organisation du tas avant restructuration

### 3.1.1 Le modèle 3D

Comme le montre les images de la Figure 56, les particules glissent bien sur la structure existante comme pour le modèle 2D, les accélérations sont bien dirigées vers le centre, les avalanches externes ont les mêmes caractéristiques que précédemment mais la présence de mouvements tourbillonnaires ainsi que de sous-structures n'a pas été établi. Il nous faudrait pour cela être en rotation permanente autour du tas pour être à même de distinguer les différents plans dans lesquels les déplacements se produisent, et ainsi s'attacher aux destinées communes des particules.

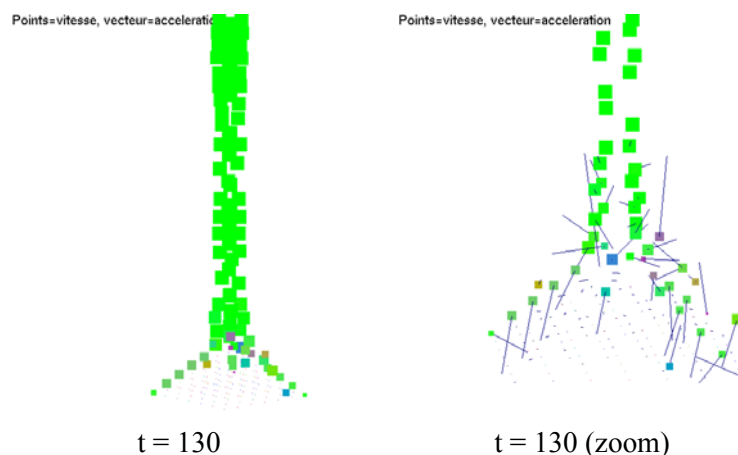
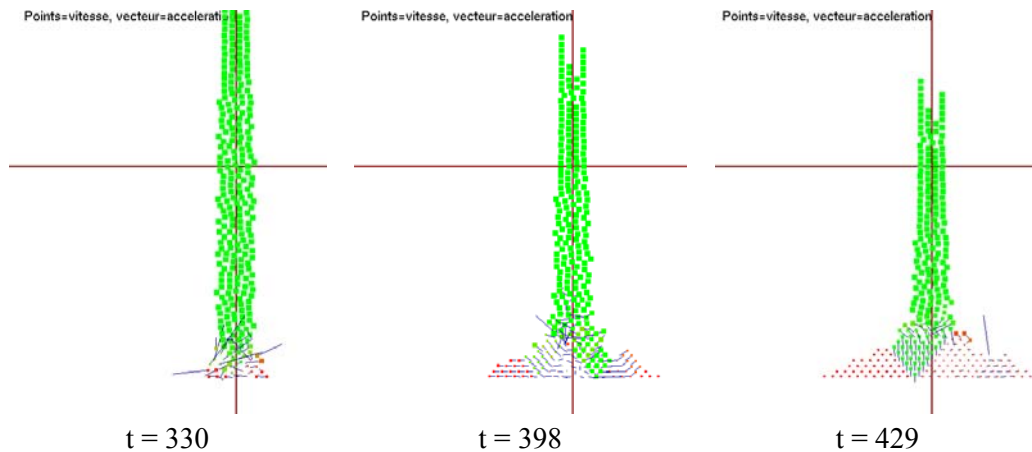


Figure 56 : Vitesses (point) et accélération (vecteur) de sable 3D (sol rugueux)

### 3.2. Modèle de sable à sol lisse

Lorsque les particules s'écoulent de l'entonnoir, les vitesses et accélérations sont semblables à celle présentes dans le modèle de pâte. L'écoulement se traduit par une vitesse homogène et une accélération nulle. Dès que les particules entrent en collision avec le sol, la vitesse diminue et l'accélération augmente en direction opposée à la gravité. Dès lors le sable se met à glisser sur le sol, donc les vitesses s'accroissent en direction du sens de glissement.



**Figure 57 : Vitesses (point) et accélération (vecteur) de sable 2D à sol lisse**

Cependant, une fois le tas suffisamment formé, le centre de celui-ci possède des accélérations en direction du sol, témoignant ainsi d'un effondrement. A partir de cet instant, le tas se décompose en sous-tas selon les couples vitesse-accélération possibles :

1. Vitesse forte, Accélération forte ;
2. Vitesse forte, Accélération faible ;
3. Vitesse faible, Accélération forte ;
4. Vitesse faible, Accélération faible.

Le cas 4, rencontré principalement en fin de simulation, correspond à une stabilisation de la position. Les particules ne participant pas à la formation du tas s'écoulent toujours de l'entonnoir avec la caractéristique 2. Les effondrements, conséquence de la restructuration du tas ont une accélération forte et une vitesse faible, tandis que les avalanches externes ont une vitesse et une accélération fortes. Les portions du tas glissant sur le sol ont une accélération forte dirigée vers l'intérieur du tas, et une vitesse faible vers l'extérieur.

L'analyse des données calculées en 3D est difficile à réaliser. Ce que nous pouvons constater est un fort glissement des particules en contact avec le sol.

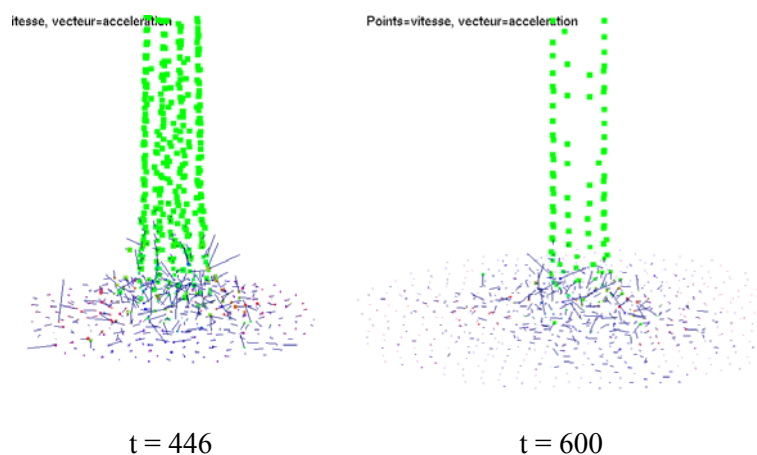


Figure 58 : Vitesses (point) et accélération (vecteur) du sable 3D à sol lisse

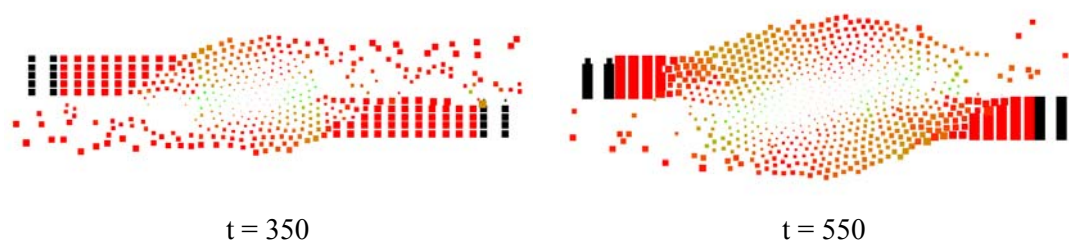
## 4. Modèle de fluides turbulents

### 4.1. Modèle de tourbillon de Kelvin-Helmholtz

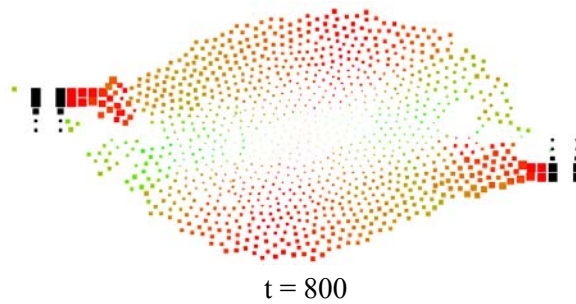
Ici les groupes se distinguent par la direction de leur vitesse. L'accélération n'est pas un critère de formation de groupe puisqu'elle y est non homogène tant en direction qu'en norme.

Dans un premier temps, les particules appartenant aux injecteurs centraux génèrent un vortex primaire. Les suivantes le heurtent (forte accélération de ces particules) et induisent un mouvement de rotation accentuant le mouvement de tourbillon existant.

Les laminarités, détectables par leurs caractéristiques de vitesse (norme régulière et dirigée vers la sortie de la scène), contournent le vortex en formation. L'accélération n'y est alors plus nulle. Le tourbillon grossit en absorbant ces particules, empêchant ainsi la formation de nouvelles laminarités. Les particules injectées par la suite sont immédiatement happées par ce tourbillon.



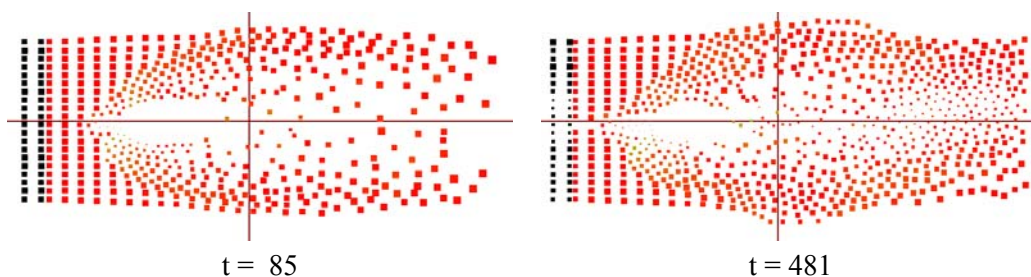




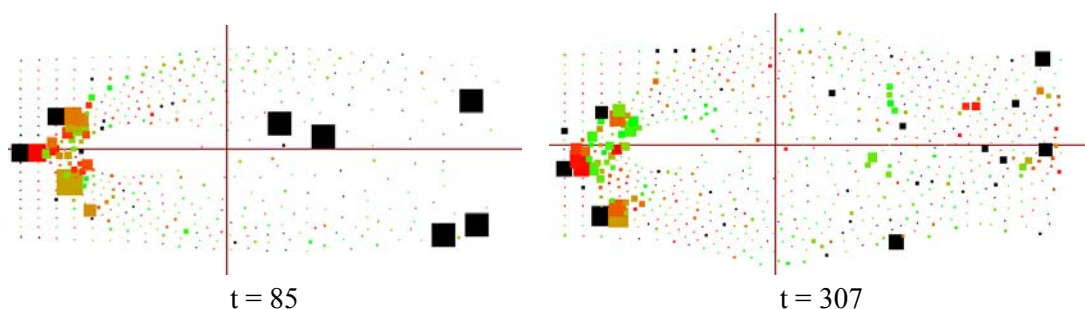
**Figure 59 : Vitesses (point) du tourbillon de Kelvin-Helmholtz (indication de direction : rouge pour l'axe X, vert pour Y, bleu pour Z)**

## 4.2. Modèle d'allée de von Karman dans le sillage d'un obstacle

L'obstacle crée un écoulement laminaire sur son bord d'attaque déviant les particules qui se trouvent ainsi prisent dans un mouvement giratoire qui les entraîne au-delà de l'influence de l'objet. En son front, se crée une zone de non mouvement remplie de particules coincées entre lui et le flux puissant et incessant de matière. Ces particules sont percutées par celles du fluide donc leurs accélérations sont fortes et non homogènes. Mais cela concerne aussi les particules longeant le bord d'attaque de l'obstacle. Lors de son contournement les masses ponctuelles de part et d'autres cherchent à se rassembler : les accélérations sont verticales.

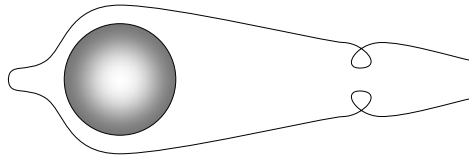


**Figure 60 : Vitesses (point) de l'allée de von Karman**



**Figure 61 : Accélérations (point) de l'allée de von Karman**

Elles semblent ensuite se désorganiser mais le fluide garde une certaine cohésion. En effet, elles ont une vitesse horizontale et une accélération verticale ce qui nous amène à penser qu'elles se déplacent en tournoyant sur elle-même tout en se dirigeant vers la sortie de la scène.



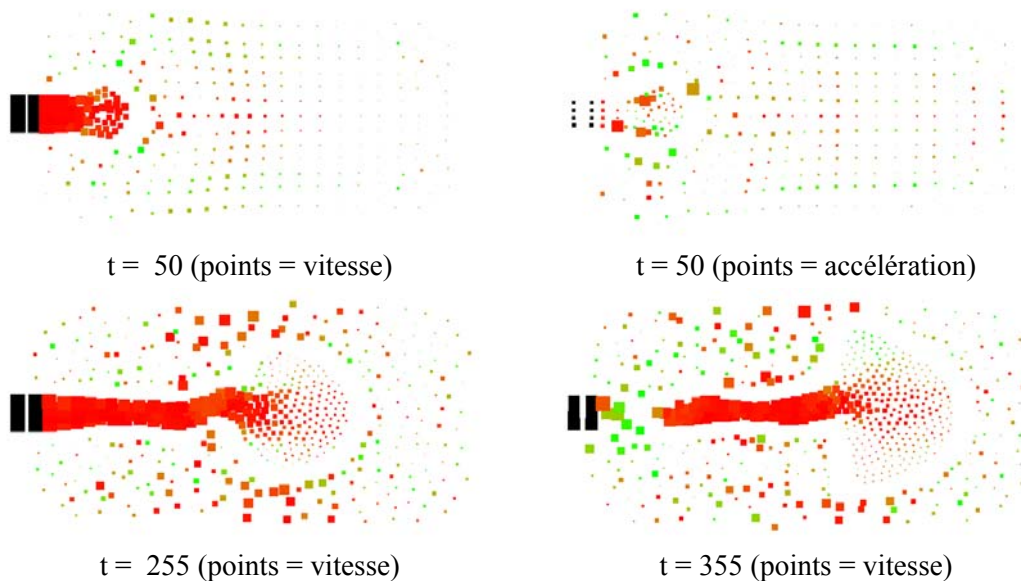
**Figure 62 : Schématisation de l'allée de von Karman dans le sillage d'un obstacle**

### 4.3. Modèle de volutes de fumée

Au début de la simulation, le milieu dans lequel le flux est introduit est organisé de manière régulière. Lorsque les injecteurs introduisent les particules dans le milieu, celui se comprime horizontalement et s'étend verticalement. Cette compression horizontale et dépression verticale vers l'extérieur (vitesse et accélération non nulles) se propage à travers le milieu.

Le jet crée autour de lui une zone vide de particules. Mais celles du milieu en bordure de cette région ont une vitesse et une accélération fortes. Comme pour l'allée de von Karman, l'accélération maximale se situe à l'apogée de la tête du jet (partie la plus haute ou la plus basse). Ces masses ponctuelles agissent par compression sur le jet, générant ainsi en arrière de la tête de petits tourbillons de sens opposés (visibles par de fortes accélérations).

De plus le bord d'attaque du jet est freiné par le milieu. Les vitesses sont donc plus faibles laissant ainsi l'environnement exercer une pression sur la matière. Mais celle-ci cherchant à se diriger vers la sortie, la tête est comprimée de part et d'autre de son centre périphérique. Cette partie n'est pas atteinte par cette déformation, puisque la vitesse  $y$  est plus importante, plus horizontale. Les tourbillons en arrière de la tête grossissent à mesure que le jet s'avance dans le milieu. Lorsqu'il n'y a plus de particules à injecter, la matière s'enroule sur elle-même en se désaxant : la tête du jet n'est pas symétrique en termes de norme, de direction des vitesses et des accélérations.



**Figure 63 : Vitesses (point) du jet**

## 5. Conclusion

Nous venons de voir qu'il est possible de détecter les caractéristiques dynamiques de fluide selon leurs vitesses et accélérations. Cette division des particules du flux en groupes spécifiques se fait selon des critères concrets mais suggérés par le visuel. Lorsque nous distinguons deux groupes selon la norme (taille du point, longueur du vecteur), la direction (couleur du point ou vecteur) des vitesses ou accélérations, notre œil est capable de séparer les deux sans avoir recours aux valeurs numériques. Comment faire pour que cette division se fasse de manière automatique sans mettre en place des méthodes spécifiques aux phénomènes ? L'analyse des données pour les modèles tridimensionnels nous aurait aidé si elles n'avaient pas été aussi complexes à examiner.

Notre programme d'analyse des données permettait aussi de visualiser la courbe des vitesses et accélérations de l'ensemble des particules du flux à un instant donnée et la courbe des vitesses et accélérations d'une particule donnée au cours de la simulation. De ce type de visualisation, nous pouvons déduire que les modèles de pâte et de sable gardent une structure relativement stable au cours du temps puisque deux masses ponctuelles initialement proches le restent plus ou moins (position, évolution des courbes de vitesses et accélérations). Cette caractéristique s'avère plus flagrante pour les pâtes. Le positionnement des particules n'influe pas sur le mouvement si elles sont séparées par une distance supérieure au plus grand des seuils des liaisons les reliant. Les matières modélisées ainsi et répondant à ces critères de structuration sont cohésives.

**Partie III - VISUALISATION DE  
PHENOMENES NATURELS**



## Chapitre 7 Construction massique récursive

---

Dans le but d'habiller une simulation CORDIS-ANIMA, nous avons mis au point un algorithme appelé Construction Massique Récursive ou algorithme CMR. Cette méthode doit répondre à la problématique de la thèse, à savoir : Comment habiller un ensemble de points de l'espace "sans structure prédéfinie" ? ([guilbaud00])



**Figure 64 : Image issue d'une simulation 3D**

Le résultat de la simulation d'un modèle physique CORDIS-ANIMA est un film constitué de l'ensemble des coordonnées des positions des masses ponctuelles du modèle simulé au cours du temps. Il n'y a aucune information quant à la position des particules les unes par rapport aux autres et par rapport à la surface de l'objet volumique modélisé. La visualisation, qui nous est proposée une fois la simulation réalisée, est limitée dans la mesure où la seule chose que l'on puisse faire c'est d'associer un cercle à un point ou relier deux particules par une ligne. Cependant, un oeil averti détecte un volume avec aussi peu d'informations.

Tous les phénomènes étudiés sont des écoulements de matière. Jusqu'à présent, ces modèles physiques n'existaient à l'ICA qu'en deux dimensions puisqu'il n'y avait aucune méthode de visualisation 3D. La simulation d'un modèle de pâte tridimensionnel nous a amené à réfléchir sur une

façon de l'habiller pour obtenir une impression de volume et ainsi pouvoir juger de la qualité du modèle réalisé.

Il existe de nombreuses méthodes permettant de passer d'un ensemble de point à un volume. On parle alors de méthodes de reconstruction. Mais ces méthodes se basent sur des mesures, des captures ou des évaluations de volume préexistant. Dans notre cas, nos particules ne décrivent pas explicitement le volume mais se contentent d'en faire partie. Autrement dit, nous n'aurons pas à implémenter une méthode de reconstruction volumique mais plutôt une méthode de construction qui créera un volume à partir d'informations partielles sur ce volume : nos particules ne sont pas à la surface du volume, mais à l'intérieur de celui-ci ; elles ne sont pas des descriptions du volume mais des informations sur son comportement.

Nous allons donc proposer une méthode de construction volumique qui générera un volume affichable à partir d'un ensemble de particules décrivant son comportement. La section 1 présente la mise en place de la méthode en deux dimensions, tandis que la section 2 explique le passage en trois dimensions de la méthode CMR.

## 1. Elaboration de la méthode CMR en 2D

La méthode CMR a besoin d'un certain nombre de paramètres pour fonctionner : il faut préciser la taille de l'espace discrétisé (ici un plan), exprimé en nombre de pixels. Elle représente la taille de discrétisation de l'espace des particules. Plus cette donnée est grande plus la résolution est élevée, donc plus la surface construite contiendra de détails et plus le temps de calcul sera important.

D'autres variables qui influencent la construction sont à fournir, mais pour des raisons de compréhension, elles seront expliquées lors de leur introduction dans l'algorithme. La méthode de construction se déroule dans l'espace discret.

Avant d'explorer en détail, le fondement de la phase de construction, expliquons son principe. Nous souhaitons relier chacune des particules du flux disposées dans un plan discret en remplissant les pixels séparant deux particules. Ces portions de matière créées servent par la suite à combler les espaces laissés vides entre les particules lors la première phase de la méthode. Cela revient à établir des chemins 8-connexes entre particules et à générer des portions de matière jusqu'à l'obtention d'un volume en se gardant de construire une surface convexe.

La méthode CMR est constituée de deux parties principales : le calcul de la carte de densité, qui nous guidera dans la construction de la surface, la construction de la surface. La carte de densité traduit l'encombrement spatial des particules dans la scène.

Ces deux étapes ont été développées en parallèle jusqu'à l'obtention d'une surface satisfaisante. Le calcul de la carte de densité débute par une phase d'initialisation qui consiste à attribuer à chaque particule du modèle physique un pixel dans le plan discrétisé. La densité n'est pas toujours calculée pour l'ensemble des pixels du plan car cela est assez coûteux en temps. En général on la calcule plutôt pour les voisins des pixels contenant une particule du flux.

Dans un premier temps, nous établissons une carte de densité simple. La densité en un pixel représente le nombre de masses ponctuelles présentes dans les 8 pixels voisins (le premier étage de voisins). Cette carte de base sert à l'amélioration de la phase de construction de la surface.

Les premiers tests n'ont pas été satisfaisants : l'expansion dans les régions pourvues de particules ne donne qu'une enveloppe convexe, tandis que l'expansion vers les plus fortes densités produisait un résultat peu satisfaisant. A partir de là, la notion de région close s'est imposée. Au lieu de chercher à s'étendre vers une forte densité pouvant se trouver fort éloignée de notre point courant, nous avons décidé de construire en direction des plus fortes densités les plus proches, i.e. en direction des plus fortes densités<sup>10</sup> situées dans une région fermée dont la taille est définie par l'utilisateur.

Plusieurs façons de calculer la distance entre deux voxels donnés ont été testées. Le calcul d'une distance euclidienne en espace discret n'étant pas très pertinent, nous avons choisi d'utiliser la distance discrète les séparant. Cette méthode est moins coûteuse en temps et en nombre de calcul ; de plus, son utilisation n'altère en rien le volume construit.

La dimension de la région close influence le volume construit. Celle-ci doit être en relation avec le positionnement des particules les unes par rapport aux autres. Une distance trop grande, lorsque le volume est petit, ne donne aucun résultat dans la mesure où la surface obtenue serait l'enveloppe convexe de l'ensemble des particules. A première vue, cette dimension correspond à la distance moyenne séparant deux pixels contenant un point du flux.

Le fait de s'étendre vers les régions de plus fortes densités les plus proches permet d'obtenir plusieurs enveloppes, ce qui est satisfaisant puisqu'un flux n'est pas obligatoirement constitué d'un seul bloc.

La méthode de construction semble prometteuse. Nous avons décidé d'améliorer le calcul de la carte de densité.

Le calcul de la carte de densité sur les 8 voisins les plus proches produit des voxels "parasites" en bordure de volume. Pour y remédier, la carte de densité subit un seuillage qui élimine ces voxels non désirés. Néanmoins, cette étape peut être la cause de « trous » dans la carte de densité, comme dans le volume final. Une phase de « pseudo-remplissage » a donc été instaurée. Comme nous le verrons par la suite, ces étapes de post-traitement ne sont pas toujours utiles.

La carte de densité se calcule à partir d'un masque de convolution qui prend en compte deux étages de voisins successifs (soit les 24 voisins les plus proches, cf. Figure 65). On se trouve confronté aux mêmes problèmes que précédemment : on obtient une enveloppe convexe si la dimension de la région est trop grande, et il est parfois nécessaire d'effectuer un post-traitement de la carte de densité.

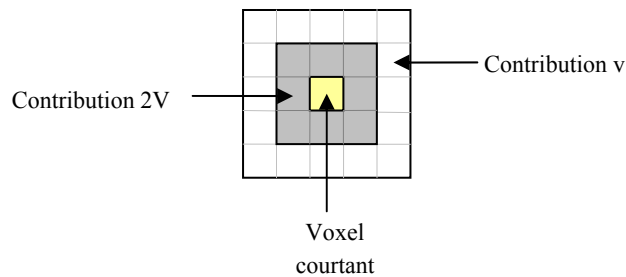
Les volumes construits sont proches de nos attentes, mais ils ne répondent pas entièrement à nos critères. Si on améliore la carte de densité, alors on améliorera les volumes puisque c'est sur elle que repose la phase de construction de la méthode.

---

<sup>10</sup> pixels dont le nombre de particules à l'intérieur ou en bordure est grand (forte valeur dans le tableau correspondant)



Nous utilisons un masque de convolution (cf. Figure 65) pour privilégier les plus proches voisins. La contribution d'un pixel contenant un point sur le premier niveau de voisins est double de celle d'un pixel sur le second niveau. Les premiers résultats montrent des progrès. Divers tests ont été réalisés et mènent aux conclusions suivantes : le post-traitement de la carte de densité n'est pas une nécessité, l'utilisation d'un masque de convolution de plus grande taille améliore la surface obtenue.



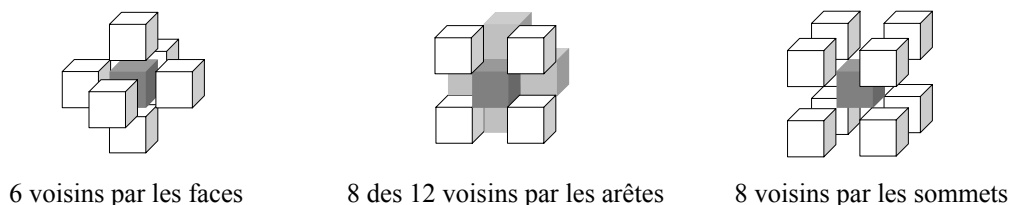
**Figure 65 : Masque de convolution de taille 2**

Une fois ces améliorations apportées, le travail a consisté à élaborer une manière de combler les trous dus au seuillage lors du post-traitement de la carte de densité. Différents essais ont été faits. Le post-traitement est fonction de la disposition des particules et des dimensions du plan (taille de discrétisation).

Il est donc difficile de trouver un bon compromis permettant de remplir de façon efficace ces trous pour tous les ensembles de particules sans considérer un cas particulier. Nous avons donc laissé la possibilité d'effectuer un post-traitement, mais nous ne pouvons garantir que ce remplissage sera efficace pour l'ensemble de particules considérées.

Ce problème peut être en partie résolu en jouant sur les paramètres qui régissent la construction de l'enveloppe (la taille du masque de convolution, la dimension d'une région, la taille du plan).

## 2. La méthode en trois dimensions

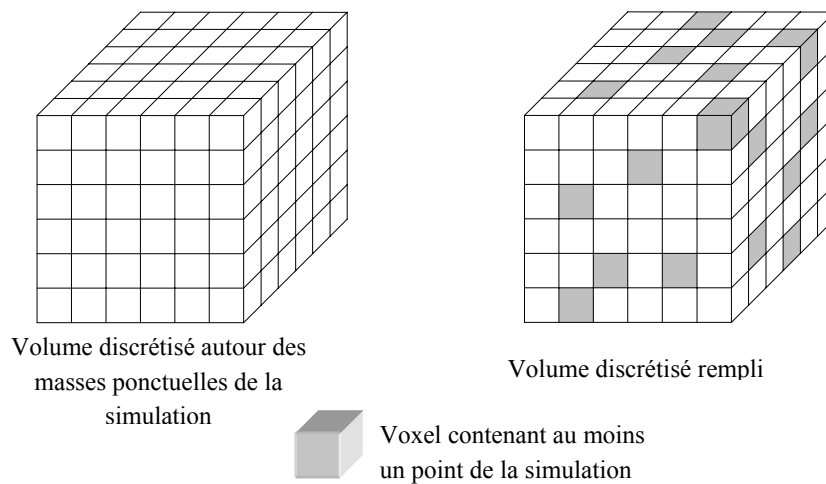


**Figure 66 : Voisinage d'un voxel en 3D**

Le développement de la méthode en 2D nous a permis de déterminer les principales caractéristiques utiles à la construction d'une surface non convexe. La méthode 3D découle de l'algorithme de base 2D en conservant les mêmes options de calcul pour la carte de densité, et la construction du volume. Un pixel devient un voxel, la surface résultant de l'algorithme de construction devient un volume (on parlera aussi de surface lorsque l'on considérera l'enveloppe du volume).

La présente section rappelle les différentes options choisies, et fait une synthèse de la méthode. Nous discuterons ensuite de la validité de la méthode vis-à-vis de la problématique.

La méthode commence par une phase d'initialisation qui insère les masses ponctuelles dans un volume englobant discret (cf. Figure 67). Ce volume est la base de la construction de la carte de densité et du volume. Il a été créé de manière à englober le nuage de particules à un instant donné. Les dimensions d'un voxel varient donc au cours du temps. Les paramètres de construction seront amenés à être changés en conséquence car ils représentent un nombre de voxels.



**Figure 67 : Initialisation du volume discret**

## 2.1. Détermination de la carte de densité

Pour calculer la densité en un point discret (un voxel), nous appliquons sur le volume discret un masque de convolution 3D de dimension 3x3. Il a pour fonction de déterminer la densité, fonction du nombre de particules qu'il comprend, et de leurs emplacements.

Il existe deux manières de calculer la carte de densité. Soit la densité est calculée pour tous les voxels du volume discret, soit elle l'est seulement pour les voxels présents dans la région clôturée par le masque de convolution (région close entourant les voxels contenant des particules).

Les résultats obtenus ne sont que très légèrement différents, car la densité est calculée pour les zones dans lesquelles *a priori* le volume va être construit. La seconde manière a été introduite pour diminuer le temps de calcul.

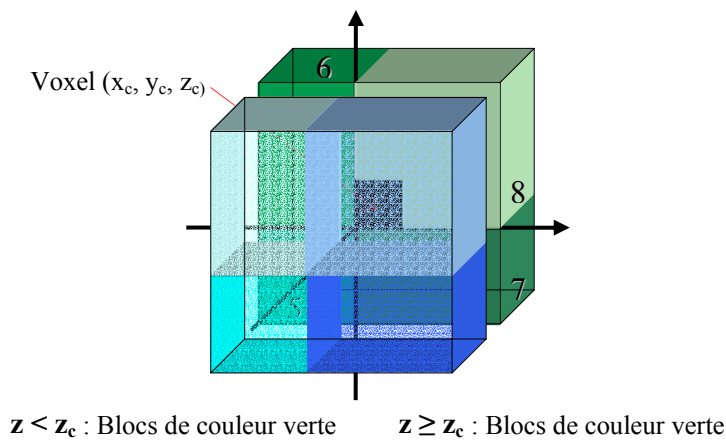
La complexité de cet algorithme est en  $O(n)$ , où  $n$  est le nombre de voxels du volume discret.

## 2.2. Construction du volume

Maintenant que nous disposons de la carte de densité, il faut construire le volume. Lors de la présentation de la méthode, nous l'avons présenté comme une expansion, i.e. un remplissage de voxels, vers les régions de plus fortes densités les plus proches. Cela revient en 3D à construire des chemins 26-connexes.

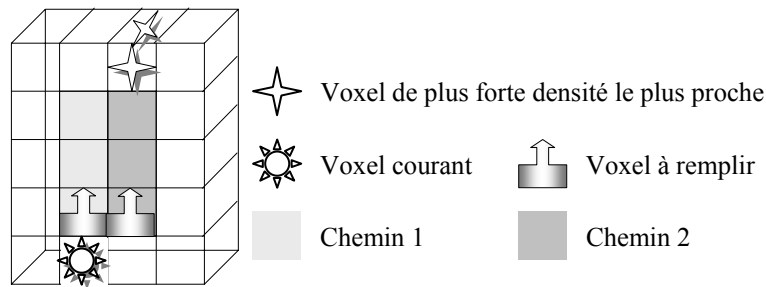
En fait, nous procédons un peu différemment. Avant d'expliquer plus en détail l'algorithme, nous introduisons la notion de sous-région autour d'un voxel. Autour d'un voxel  $(x_c, y_c)$ , huit sous-régions sont définies :

1. la région  $x < x_c$  et  $y < y_c$  et  $z < z_c$ ;
2. la région  $x < x_c$  et  $y \geq y_c$  et  $z < z_c$ ;
3. la région  $x \geq x_c$  et  $y < y_c$  et  $z < z_c$ ;
4. la région  $x \geq x_c$  et  $y \geq y_c$  et  $z < z_c$ ;
5. la région  $x < x_c$  et  $y < y_c$  et  $z \geq z_c$ ;
6. la région  $x < x_c$  et  $y \geq y_c$  et  $z \geq z_c$ ;
7. la région  $x \geq x_c$  et  $y < y_c$  et  $z \geq z_c$ ;
8. la région  $x \geq x_c$  et  $y \geq y_c$  et  $z \geq z_c$ ;



**Figure 68 : les 8 sous-régions autour d'un voxel**

L'expansion dans le volume s'effectue vers les régions de plus fortes densités les plus proches. Nous ne pouvons l'étendre dans une seule direction en occultant les autres, sous peine d'obtenir un volume non conforme à nos espérances. C'est pourquoi nous l'étendons dans chaque sous-région, dans une ou plusieurs directions données. Si l'on considère le cas de la région 4 nous pouvons remplir un voxel en  $(x_c+1, y, z)$ , en  $(x, y_c+1, z)$ , ou en  $(x_c+1, y_c+1, z)$ , en  $(x, y, z+1)$ , en  $(x+1, y, z+1)$ , en  $(x, y+1, z+1)$ , en  $(x+1, y+1, z+1)$  (cf. Figure 68). Le nombre de voxels à créer dépend de la (des) position(s) de la plus forte densité la plus proche (cf. Figure 69).



**Figure 69 : Construction vers les plus fortes densités les plus proches (coupe du volume)**

L'algorithme peut être synthétisé de la manière suivante :

**Algorithme** : Phase de construction du volume

```

POUR chaque point particules (variable point_courant) FAIRE
  voxel_courant = voxel qui contient le point_courant
  Detecte(voxel_courant)

/*----- Détecte(voxel_courant) -----*/
POUR chaque sous-region entourant le voxel_courant FAIRE
  POUR chacun des voxels de plus forte densité les plus proches
FAIRE
  Déterminer les voxels à remplir selon la position du (des)
  voxels de plus forte densité les plus proches
  POUR chaque nouveau_voxel déterminé à l'étape précédente FAIRE
    SI le nouveau_voxel est vide ALORS
      Le remplir
      Detecte(nouveau_voxel)

```

Comme pour le calcul de la carte de densité, cet algorithme est en  $O(n)$ , où  $n$  est le nombre de voxels du volume discret.

Les différentes variables utilisées par l'algorithme influencent la façon dont est construit le volume. Plus la taille de discrétisation est grande plus il sera possible de voir de détails. La taille du masque de convolution ainsi que la taille de la région doivent être en relation pour établir un volume satisfaisant et représentatif de la disposition des masses ponctuelles dans l'espace.

L'utilisateur ne donne pas la taille d'une région, mais la taille de la sous-région i.e. le nombre d'étages de voisins à prendre en compte.

## 2.3. Validation de la méthode

Rappelons notre problématique : habiller un ensemble de points de l'espace "sans structure prédéfinie".

A la vue de la Figure 70, et en les comparant à notre ensemble de particules initial, l'objectif semble atteint. Nous obtenons un volume non convexe englobant les particules du flux ne masquant pas les particularités de la pâte de dentifrice. Il ne faut pas s'étonner de la forme de la pâte en son sommet : si elle est carrée, c'est parce que l'entonnoir qui la projette dans le réceptacle a cette forme.

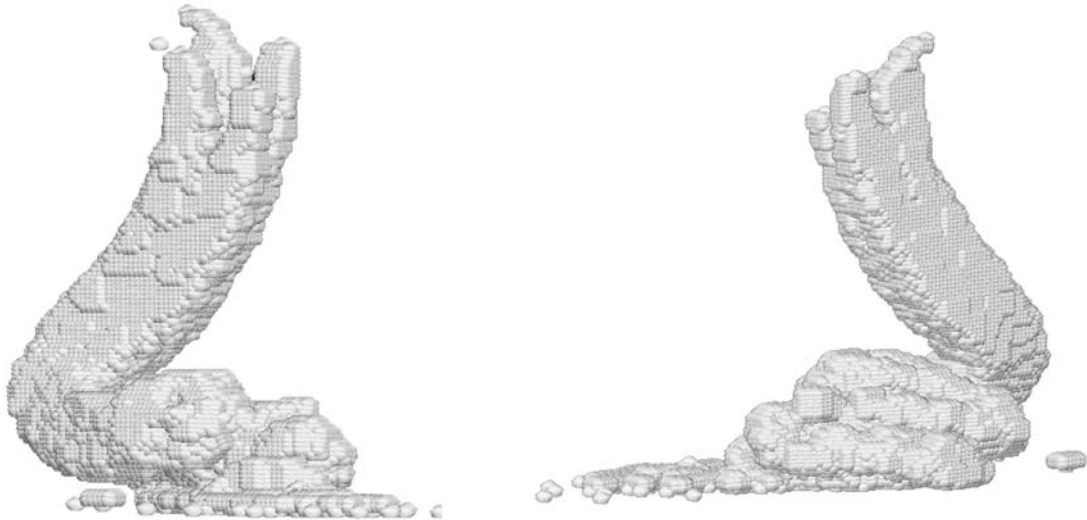


Figure 70 : volume construit (les voxels sont des sphères)

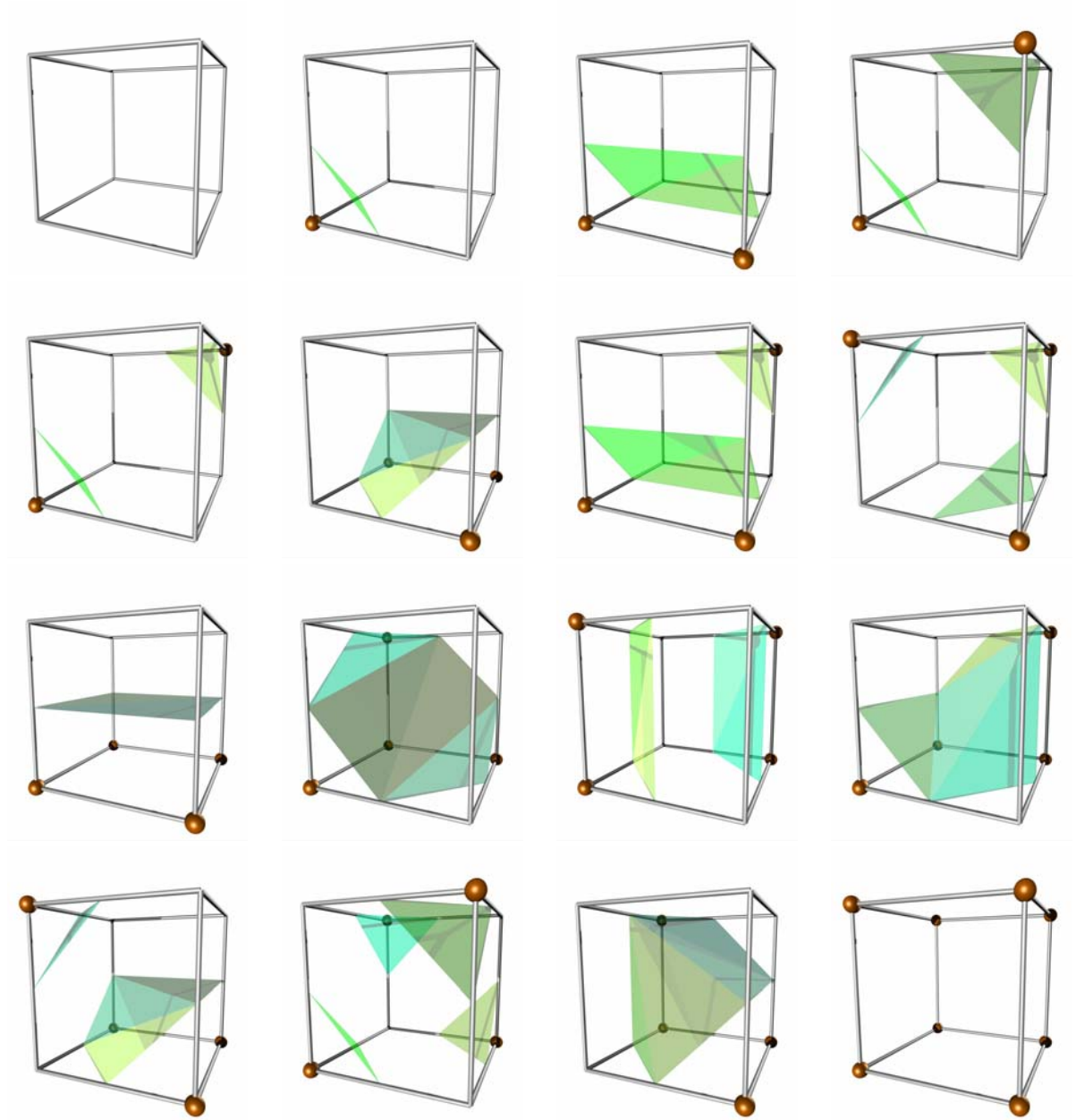
Du point de vue volumique, le but est atteint, mais la visualisation laisse à désirer. Une pâte n'est pas formée de voxels parallélépipédiques ou sphériques, elle a une surface lisse.

## 3. La visualisation par création d'un maillage

### 3.1. La méthode de polygonalisation

Cette méthode est utilisée pour visualiser les surfaces implicites. Il existe plusieurs méthodes de polygonalisation (pavage) d'isosurface. Nous utiliserons la plus utilisée, celle mise au point par Lorenson et Cline ([Lorenson87]) : les « Marching Cubes ».

Elle a pour principe de discrétiser l'espace entourant l'objet modélisé pour créer un ensemble de facettes qui approximent l'isosurface. Une facette est créée pour un voxel du volume suivant l'appartenance ou non des sommets. Il y a 8 sommets, deux états possibles pour chacun d'entre eux, donc  $2^8$  cas possibles. L'ensemble des cas peut être généré par rotation ou symétrie d'un des 16 cas de base (cf. Figure 71).



**Figure 71 : Les 16 cas de base**

L'algorithme de rendu de surfaces implicites se décompose en plusieurs phases. Dans un premier temps, on construit les 256 cas de création de facettes par application d'opérateurs géométriques sur les 16 cas de base. Ensuite pour chaque voxel de l'espace discrétisé, on établit l'appartenance des sommets à l'isosurface et on construit les facettes appropriées. On termine le rendu de la surface en calculant les normales en chaque sommet de facettes.

Nous avons adapté l'étape d'étiquetage des voxels à notre cas.

### **3.2. L'adaptation de la méthode à nos besoins**

Comme cela a été fait pour la phase de construction, nous avons commencé la conception de la visualisation en 2D. Cette simplification a été utile pour souligner les points sur lesquels il fallait approfondir les recherches.

Ne disposant pas d'une équation de surface, nous nous sommes basés sur le voisinage du voxel (Figure 66) pour l'étiquetage des sommets. Nous avons réalisé différentes études qui ont montré qu'un voxel en surface de l'objet n'est pas toujours plein. Cela est contraire à ce qui est communément admis lorsque l'on visualise un volume discret. Mais une telle approche est nécessaire pour éviter les discontinuités de surface. De plus la position des voisins pleins par rapport au voxel à étiqueter est importante : un voxel voisin par les faces a plus de poids qu'un voxel voisin par les arêtes, et donc qu'un voxel voisin par les sommets.

L'établissement d'une hiérarchie nous a permis d'établir le Tableau 5 pour connaître l'appartenance d'un sommet au volume. La colonne « Faces » indique le nombre de voisins par les faces pour qu'un voxel soit en bordure de surface. Il en est de même pour « Arêtes » et « Sommets ».

	Faces	Arêtes	Sommets
<b>Voxel plein</b>	1	$\geq 2$	-
	2	1	1
	2	$\geq 2$	-
	3	$\geq 1$	-
<b>Voxel vide</b>	0	3	1
	1	$\geq 3$	-
	2	1	1
	2	$\geq 2$	-
	3	$\geq 1$	-

**Tableau 5 : Conditions pour déduire si un sommet est en bordure de surface**

### 3.3. Les résultats

Les volumes de la Figure 72 ont été obtenus en appliquant la méthode de Lorensen et Cline modifiée une fois le volume construit.

Comme on peut le voir la surface n'est pas parfaitement lisse puisque les facettes sont positionnées à 45° (ou un multiple de 45°) les unes des autres. De plus une seule normale par facette est calculée, et non une par sommet.

Cette visualisation a permis de montrer la validité de la méthode, bien que la surface obtenue ne soit pas lisse. En regardant un des volumes construits, nous pouvons voir les différentes couches qui le composent. Pour être véritablement validée, la méthode de construction discrète de volume devra être utilisée pour visualiser des phénomènes autres que des pâtes, comme les fumées ou les sables par exemple.



**Figure 72 : Volumes construits par facettes**

## **4. Amélioration de la méthode CMR**

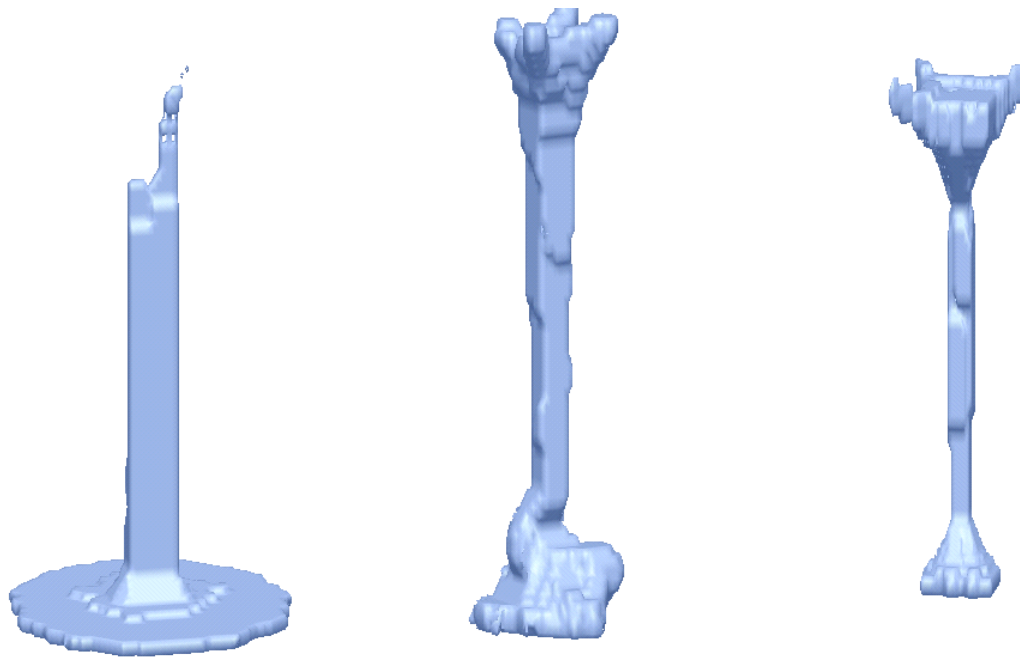
Cette méthode de construction massique récursive permet de construire un volume satisfaisant en manipulant trois paramètres principaux : la taille du volume, de la région dans la carte de densité, de la zone lors de la construction.

Lorsqu'on construit un ensemble de volumes pour chaque image d'une simulation nous sommes amenés à modifier ces paramètres car le volume englobant change à chaque image, il a toujours les mêmes dimensions discrètes (le nombre de voxels qui le compose) mais la taille du voxel est modifiée. L'animation ainsi obtenue engendre des à-coups dans la visualisation lors des changements des paramètres.

Pour remédier à cet inconvénient, nous calculons un volume englobant pour l'ensemble des particules pour toute la simulation. Ainsi la taille d'un voxel est fixe, et il n'y a plus besoin de faire varier les paramètres de la méthode (cf. Figure 73).

Nous avons essayé d'habiller nos modèles avec cette méthode mais certains s'y prêtent mal (le gel par exemple). Il faut que le flux soit suffisamment épais, et que la répartition soit suffisamment uniforme pour ne pas masquer de caractéristiques dynamiques. On remarque que cette méthode produit une discontinuité de surface, dans la mesure où les évolutions de forme ne sont pas lisses sur la surface du volume. Cet effet, peu esthétique, provient de l'angle constant séparant deux voxels habillés au niveau des discontinuités de surface.





**Figure 73 : Volumes construits avec la méthode modifiée**

## Chapitre 8      Visualisation par structuration du flux

---

Comme il a été exposé dans l'article [Luciani00], il existe cinq différentes configurations de liaisons inter-particulaires. Chacune d'entre elles est définie par au plus 4 paramètres, dont au moins un est une information de distance (la longueur du seuil de l'interaction). Pour les configurations C1, C2, C3 (cf. Chapitre 3 - Section 1), il n'y a qu'un seul seuil qui régit le mouvement de la matière, donc un seul influence les distances entre particules. Pour C4 et C5, il existe deux seuils et une relation de proportion les liant. Les deux distances jouent donc un rôle dans la répartition dans l'espace des particules. A elles deux, elles définissent une zone de non pénétrabilité autour de chaque point matériel.

Chercher à spécifier la répartition dans l'espace des particules à partir d'une distance, est un moyen de structurer la matière sans avoir recours à la topologie du modèle physique. Dans la phase d'observation (qui pourrait être similaire à l'observation de données provenant de mesures sur un phénomène réel), il est plus général de faire l'hypothèse que les informations de structure de la matière ne sont plus accessibles à l'observateur, mais qu'au contraire, une bonne représentation devrait lui permettre de les retrouver.

Nous avons voulu, pour cette deuxième méthode de visualisation, associer à chaque point matériel un encombrement spatial physique. Cet espace lui était attribué lors de la simulation et était représenté par les seuils élastiques et visqueux des liaisons inter-particulaires. Cet encombrement spatial physique détermine de manière implicite les positionnements relatifs dynamiques des points matériels les uns par rapport aux autres dans le mouvement réalisé. Une fois le mouvement réalisé, nous disposons uniquement de la cinématique de ce que nous avons appelé particules. L'idée a alors été de déduire directement de cette cinématique des « connexions » entre particules leur relation spatiale effective dans le mouvement réalisé.

Ces deux données, interactions physiques dans l'espace du modèle physique et connexions dans l'espace géométrique, en plus d'attribuer un espace propre à chaque élément (point matériel ou particule), produisent une structuration spatiale du flux, la première implicitement, la deuxième plus

explicitement et donc une idée de ce que pourrait être le contour de la matière simulée. Ainsi il sera possible d'obtenir un volume non-convexe enveloppant les particules.

Une technique de structuration permet de visualiser les modèles physiques et leurs caractéristiques. Cette approche d'habillage de particules a pour but de savoir si la réalisation d'une méthode aux paramètres indépendants du niveau de détails est possible.

Nous commençons par expliquer l'intérêt de la construction d'un graphe de voisinage, puis présentons la méthode de visualisation utilisée.

## 1. Le graphe de voisinage

Pour connaître la structuration du flux (interne et de surface) sans connaître les paramètres des interactions entre masses ponctuelles qui régissent le comportement dynamique du modèle, nous construisons un graphe de voisinage. Ces relations de voisinage sont calculées suivant une distance. Ainsi nous affectons implicitement à une particule un espace d'influence spatial observé. L'assemblage de ces espaces figure un volume englobant traduisant la forme globale suggérée par les particules. La structuration du flux décrit les caractéristiques dynamiques en termes de distances observées qu'elles soient internes ou non, au flux de matière.

Nous dirons qu'un graphe de voisinage est satisfaisant si sa structure ressemble à ce que l'œil détecte implicitement et s'il n'est pas composé d'un nombre trop élevé de connexions. Si l'on considère deux portions de matière formant un mélange inter-couches (caractéristique de pâte), ces dernières ne devront être connectées que lorsque la distance entre les deux sera inférieure à l'encombrement spatial physique des particules.

### 1.1. Choix de la distance

L'espace occupé par une masse ponctuelle de la simulation varie en fonction de la position de cette dernière dans le flux, et de l'instant de simulation choisi. L'encombrement spatial associé à chacune d'entre elle est fixe et définit une zone de non pénétration pour les autres masses ponctuelles.

Un graphe de voisinage s'établit à partir d'une relation de proximité entre particules. Si la distance euclidienne choisie pour construire le graphe n'est pas adaptée au nuage de particules à analyser, le graphe résultant ne sera pas significatif.

Les premières constructions de graphes ont été faites sur les modèles 2D. Nous nous sommes servis d'une distance qui nous semblait représentative de la répartition des particules dans l'espace à savoir la distance moyenne relative<sup>11</sup> entre particules. Cette distance peut être relativement grande par

---

<sup>11</sup> La distance moyenne relative est calculée pour chaque image de la simulation et représente la distance moyenne entre chaque particule du flux.

rapport au seuil de l'interaction physique inter-particulaires, mais nous ne pouvons en juger car *a priori* nous ignorons la structure du modèle physique.

Cette distance divisée par dix s'est avérée être significative pour certains modèles peu élastiques. Cependant elle était parfois trop grande ou trop petite. Nous l'avons modifiée en révisant la valeur du diviseur en fonction de la périodicité de la fonction représentant les vitesses des masses ponctuelles en fonction des numéros de ces mêmes masses. La périodicité correspond au nombre de masses ponctuelles constituant une couche de matière lors de leurs placements au début de la simulation. La distance ainsi obtenue est la base de la recherche d'un scalaire formant un graphe de voisinage satisfaisant.

Pour les modèles où l'élasticité prend le pas sur la viscosité, la distance ne peut être déterminée de manière aussi catégorique. En effet, les dynamiques sont plus brusques empêchant une continuité d'évolution des positions, donc de la vitesse, de l'accélération. De plus, pour certains de ces modèles, le placement des particules est se fait dans un demi entonnoir de forme triangulaire ou se fait via un injecteur. Il n'y a donc pas de périodicité de cette fonction vitesse. La détermination de la distance pour ces modèles s'est faite de manière empirique.

Pour construire les graphes de voisinage 3D, nous sommes partis des distances 2D et des distances moyennes relatives. Dans la plupart des cas, ces distances étaient trop grandes pour que le graphe soit satisfaisant. Nous voulions une structuration du flux mettant en avant ces caractéristiques avec un nombre raisonnable et suffisant de connexions. Il semble que les flux des modèles 3D découlant de modèles 2D, donc utilisant les mêmes paramètres, soient plus compacts, moins dispersés dans l'espace. Cette caractéristique de la matière est due à un nombre de particules plus élevé, induisant alors des forces plus cohésives.

## 1.2. Résultats

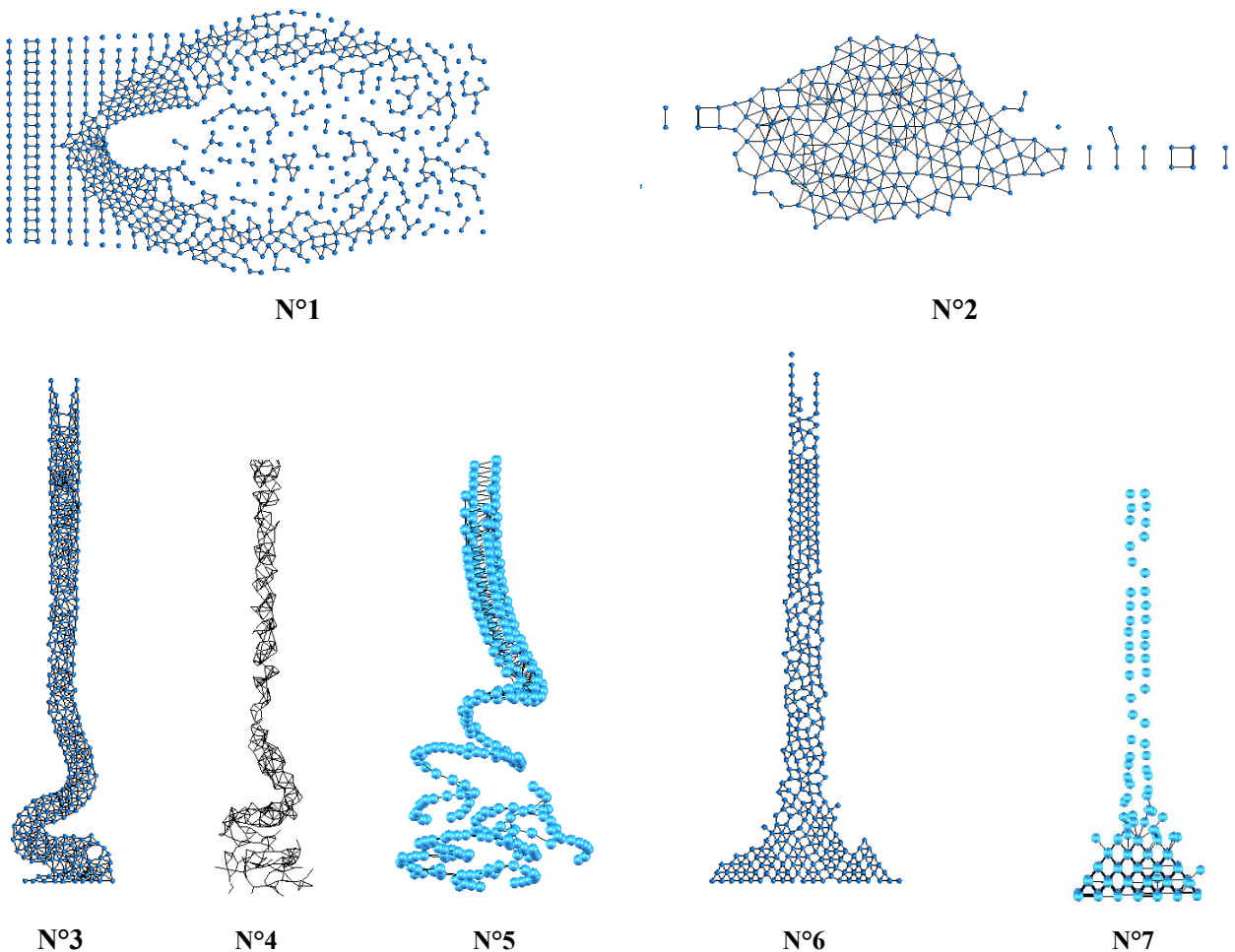
La Figure 74 présente quelques graphes de voisinage obtenus pour différentes simulations 2D et 3D.

Le graphe  $N^{\circ}3$  est pourvu de trop de connexions (environ 900 liens pour 300 particules). Le voisinage d'une particule est flou et ne nous permet pas d'analyser la structure du flux. Cette constatation faite ici pour l'image 375 n'est pas valable pour l'image 300, ou tout autre image antérieure à cet instant. Lorsque le flux de matière s'écoule, après le passage de l'entonnoir, celui-ci n'est pas contraint et occupe le maximum d'espace selon ses caractéristiques. La structure du flux est donc en perpétuelle évolution. Cependant il est possible d'établir une distance qui conduise à des graphes de voisinage satisfaisants pour la quasi-totalité des instants de simulations.

Les deux premiers graphes sont des modèles de fluides turbulents. On remarque que malgré leur caractère chaotique il existe une structure interne, qui se construit petit à petit pour le  $N^{\circ}2$ , et qui demeure dans le cas  $N^{\circ}1$  une fois l'écoulement établi autour de l'obstacle. Le graphe fait apparaître des formes courbes faisant penser à une propagation d'ondes (graphe  $N^{\circ}1$ ), et des remous entre les zones de propagation. Sur le cas  $N^{\circ}2$ , le phénomène d'enroulement du flux sur lui-même (tourbillon

de type Kelvin-Helmholtz) est visible de manière plus claire lorsque l'on s'intéresse aux particules en faisant abstraction de leur relation de voisinage.

La non connaissance de la topologie du modèle physique CORDIS-ANIMA pourrait apparaître comme un handicap. C'est le cas lorsque les masses ponctuelles issues d'un même injecteur sont plus éloignées les unes des autres que de celles issues d'injecteurs voisins. La connexion spatiale se fait alors entre masses de différents injecteurs et non entre masses d'un même injecteur comme le fait spontanément l'œil lors de la projection de la simulation. Dans le principe de cette thèse, le postulat de départ est de chercher à inférer le plus possible la structure du phénomène réalisé pour à terme mieux comprendre si une telle extraction est plus aisée à effectuer lorsqu'il s'agit de flux dynamique structuré par une loi physique ou un flux quelconque. Autrement dit, pourquoi l'œil détecte-t-il si bien les formes et les structures lorsqu'il s'agit de données de notre monde physique ? Bien que cette thèse ne prétende pas résoudre ou même approfondir cette question difficile, elle prend le parti de ne pas en biaiser l'avancée



**Figure 74 : Quelques graphes de voisinage 2D et 3D**

Le modèle de sable à sol lisse ( $N^{\circ}6$ ) fait apparaître une structure géométriquement ordonnée du tas en formation. Les lignes de force se traduisent par le fait que les particules participant à ce phénomène ne forment pas un triangle avec leurs voisins.

On peut remarquer que l'analyse des structures 3D se complique lorsque les particules ne sont pas aussi bien ordonnées ( $N^{\circ}4$  et  $N^{\circ}5$ ) que pour le sable lisse ( $N^{\circ}7$ ). Mais si l'on ne visualise pas les particules, la structuration apparaît plus clairement. Le modèle de gel ( $N^{\circ}5$ ) n'est pas formé d'un seul morceau de matière comme on aurait pu le croire. Par contre, sa portion de flux plus large, plus dense possède la forme de l'entonnoir, et toutes les particules sont réparties en son pourtour. Le graphe du modèle de pâte ( $N^{\circ}4$ ) est présenté ici sous sa forme brute (sans les particules). On distingue l'étalement des particules sur le sol, et une densité plus importante au dessus du premier extremum. L'ensemble des connexions donne, par endroits, l'impression d'une visualisation en fil de fer où seul le contour de la forme est représenté, alors que ce n'est pas le cas.

Les connexions du graphe de voisinage construisent une structure de la matière du phénomène simulé. Parfois cette structure est régulière avec peu, voire pas, de connexions qui s'intersectent. C'est le cas des sables ou du fluide expansif avant tassement. Pour les pâtes, ce n'est pas le cas, ce qui semble normal puisque la matière est plus cohésive.

Les données d'entrée de cette méthode, par rapport à la méthode de construction massique récursive, sont plus pertinentes car donner une distance a plus de sens que de donner un scalaire représentant un nombre de voxels.

### 1.3. Utilisation pour la visualisation

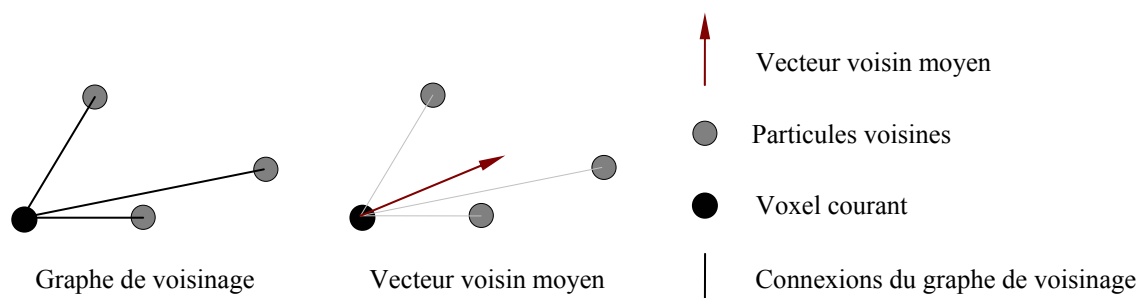
Nous avons construit des graphes de voisinage décrivant convenablement la matière du flux. Comment pouvons-nous utiliser ces informations pour visualiser le nuage de particules ? Un graphe de voisinage est une indication de structure spatiale interne et de contour qui caractérise un modèle en donnant un encombrement spatial à chaque particule du flux. La distinction des caractéristiques d'un modèle est entièrement basée sur la notion de distance. Cette conception en termes de distance du nuage de particules est donc un moyen de donner un volume à cet ensemble.

Nous nous sommes alors intéressées à la distance moyenne séparant une particule de ces voisines directes. La représentation de cette distance s'est faite dans un but précis : les particules des bords sont-elles plus éloignées de leurs voisines que celles que nous qualifions d'interne ? La visualisation de cette distance par une simple ligne horizontale n'a pas été pertinente. La représentation par une sphère plus ou moins grande l'a été davantage. Nous avons ainsi pu voir qu'il existait des disparités selon la place d'un point dans le flux.

La distance moyenne aux voisins est une information unidimensionnelle dans un environnement bi ou tridimensionnel. L'analyse des graphes de voisinage a mis en avant les différences de positionnement des voisins en fonction de la place de la particule dans le nuage. Se baser sur une information en adéquation avec la dimension de la scène dans laquelle évolue le nuage de particules nous permet de définir un ensemble de règles qui détermine la nature des particules (interne, aux bords). Pour cela nous étendons la notion de distance moyenne aux voisins en calculant la direction du vecteur voisin moyen<sup>12</sup> (cf. Figure 75).

---

<sup>12</sup> Par abus de langage nous serons amenés à utiliser le terme de « voisin moyen » ou de « vecteur moyen » pour désigner le vecteur voisin moyen.



**Figure 75 : Définition d'un vecteur voisin moyen**

Le graphe est une manière de représenter la densité de la matière. Calculer le vecteur voisin moyen revient à calculer le gradient<sup>13</sup> de la carte de densité. Ce vecteur calculé pour le premier niveau de voisins est pertinent. Nous l'avons aussi chiffré pour les voisins de niveaux supérieurs. Nous avons appelé voisins de niveau 2, les particules atteignables à partir d'une particule donnée par l'intermédiaire d'une particule. Les voisins de niveaux  $n$  sont ceux atteignables par visite de  $n-1$  particules.

La connaissance du graphe de voisinage ainsi que du vecteur voisin moyen est-elle suffisante pour visualiser un nuage de particules ? Il semble que oui, mais il serait intéressant de connaître la position d'une particule dans le volume.

Le vecteur moyen a servi à l'étiquetage des particules suivant leur nature dans le flux. De nombreux essais ont été faits en prenant en compte la taille et la direction du vecteur moyen de niveau 0, de niveau 1, etc. En 2D, il s'est avéré qu'il n'y avait pas besoin de calculer autant de données pour déterminer si une particule se trouvait en bordure de flux ou non.

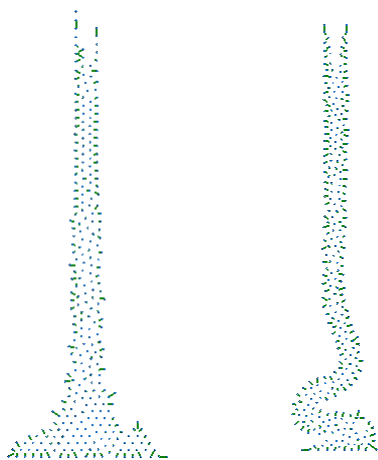
La Figure 76 représente la visualisation du vecteur voisin moyen pour le modèle de sable à sol lisse 2D et pour le modèle de pâte 2D. Les vecteurs voisins moyens des particules que nous jugeons, nous observateurs, sur le contour sont dirigés vers le centre du flux (comme le gradient) et leur longueur est plus grande que pour les particules internes.

Le modèle de pâte 2D fait apparaître des vecteurs plus grands ainsi qu'une disparité de longueur entre particules proches aux extrémités. Ces constatations se font sur pratiquement l'ensemble de nos modèles. La visualisation de la simulation par une représentation des vecteurs vers les voisins moyens a mis en avant la structure interne de la matière. Sur le modèle de sables, les lignes de force sont détectables par leur entourage proche où les vecteurs des particules en contact avec le sol ne sont pas verticaux mais dirigés vers ces lignes.

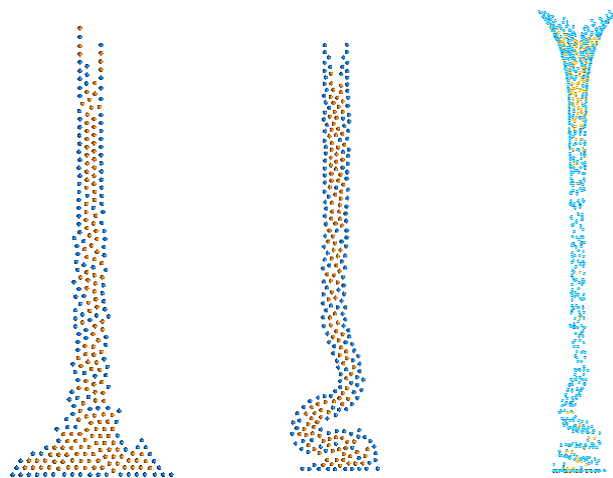
Cette représentation donne une notion d'intérieur-extérieur sur le flux. Nous avons donc cherché à établir des règles pour détecter les particules en bordure du nuage de particules. C'est par observation et analyse des graphes de voisinage et des vecteurs voisins moyens que cela a été réalisé. Nous avons travaillé sur des modèles 2D car l'étude était plus simple à réaliser.

<sup>13</sup> Le gradient de la carte de densité est la variation de la densité par unité de longueur (le voxel en 3D, le pixel en 2D). Il pointe vers les plus fortes densités, donc vers l'intérieur d'un objet.

De nombreux tests ont été effectués pour déterminer si une particule se trouve en bordure de volume. Nos recherches se sont d'abord basées sur la longueur du vecteur moyen par rapport au vecteur voisin maximal, mais cela n'a été suffisant. En effet, certaines particules des extrémités du flux ont un vecteur quasiment nul mais peu de voisins. Après analyse des résultats des différentes études, une particule est déclarée aux bords si la longueur du vecteur voisin est supérieure à un certain seuil du vecteur voisin maximal ou si le nombre de particules voisines est inférieur ou égal à 2. Une autre configuration donne des résultats satisfaisants. On reprend les deux critères de qualification et si par le dernier critère la particule a été marquée aux bords, alors elle ne le sera plus si le vecteur voisin moyen de deuxième génération (voisin des voisins) est inférieur à 0.5% du vecteur voisin de maximal. Les différences entre les deux méthodes au niveau des résultats visuels sont minimales.



**Figure 76 : Vecteurs voisins moyens**



**Figure 77 : Particules aux bords**

Dans la Figure 77, les particules aux bords sont représentées en bleu, et les autres en orange ou jaune. Les images du sable et de la pâte sont en partie conformes à nos attentes. Il arrive que certaines particules enfermées dans le flux soient étiquetées aux bords à cause d'une répartition non uniforme des particules dans l'espace. Pour la 3D, on constate que l'étiquetage n'a pas vraiment de sens sauf pour la portion de particules passant l'entonnoir.

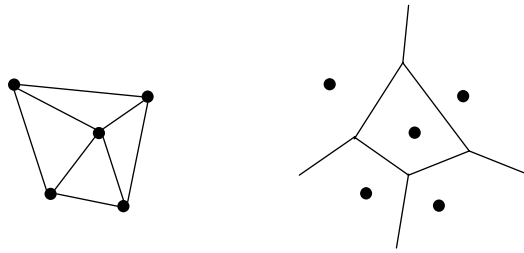
## 2. Visualisation de la surface

### 2.1. Choix de la méthode de visualisation

Nous disposons d'un contour défini par un graphe de voisinage. Ce graphe n'est pas une surface mais une charpente de celle-ci, une indication de l'endroit où elle doit se trouver à un espace près (les particules en bordure de surface n'ont pas d'encombrement spatial). Il donne aussi une information de la structuration interne du flux, donc des dynamiques caractéristiques internes.



La méthode de visualisation de ce graphe doit rendre les caractéristiques internes et de formes des modèles le mieux possible. Nous savons qu'il existe un graphe connectant les particules du modèle physique, donc une méthode de triangulation telle que le graphe de Voronoï ou de son dual Delaunay pourrait convenir (cf. Figure 78). Toutes deux sont des méthodes de construction de graphe géométrique qui produisent un pavage d'une surface à partir d'un ensemble de particules en fonction de leurs plus proches voisines. Le principal avantage dans la construction de ces graphes, est le fait qu'ils sont uniques pour un ensemble de particules données, mais leur utilisation dans le milieu de l'animation est dans la majorité des cas lourde, puisque les graphes doivent être recalculés à chaque image.



**Figure 78 : Diagrammes de Delaunay et de Voronoï**

Le diagramme de Voronoï est un graphe régulier car chaque sommet du diagramme possède un nombre constant de voisins, et ce nombre vaut trois sauf si il existe des germes cocirculaires. Tout sommet du graphe est le centre d'un cercle qui passe par trois germes et ne contient aucun autre germe. Le diagramme de Delaunay est l'ensemble des arêtes qui relient les particules appartenant à un même cercle et ne contenant aucun autre point.

Le graphe de Voronoï semble plus adapté puisqu'il fournit à chaque nœud, germe de son graphe, un espace qui lui est propre. Il partitionne l'espace en région d'influence. Chaque point à l'intérieur d'une zone est plus proche de son germe que n'importe quel autre germe du graphe. Nous savons qu'une représentation spatiale des particules fournit un ensemble d'informations permettant de concevoir la forme du contour de notre modèle. Avec cette méthode, chaque point du contour aurait alors son encombrement spatial mais pour cela il faut avoir détecté les particules aux bords pour pouvoir fermer le graphe. Mais comment clore ce graphe de manière adéquate pour ne pas faire scintiller le contour du volume ? La construction d'un tel graphe est lourde en calculs et à mesure que le nombre de particules, futurs germes du graphe, augmente, fort long à calculer. Son dual le graphe de Delaunay ne donnerait pas d'encombrement spatial à chaque particule aux bords, mais il a l'avantage de résoudre le problème de contour.

L'espace occupé par les particules définit le niveau de détail unique de la surface formée par les particules. L'obtention d'une surface lisse est problématique. La répartition non-uniforme des particules peut être source de perte de détails dans le contour. Il est possible que cela masque les caractéristiques dynamiques. Nous ne baserons pas la visualisation du graphe de voisinage sur ces méthodes.

L'habillage des particules par une surface composée d'un assemblage de primitives graphiques est une autre solution. Outre les méthodes de type CSG, il existe la modélisation par surfaces implicites à squelettes qui fusionnent des objets graphiques pour en former un seul au contour lisse. Les surfaces implicites sont utilisées pour modéliser des objets volumiques en prenant en compte la notion de densité et la proximité entre particules sources. De nombreuses méthodes de modélisation de phénomènes naturels les utilisent ([nishita97], [bittar95], [desbrun95], [triquet01], [chiba95], ...). Ce que l'œil analyse comme étant le contour de l'ensemble des masses ponctuelles, les surfaces implicites à squelettes le réalisent en construisant une surface. Cette méthode sera utilisée pour construire une surface à partir d'un graphe de voisinage.

Comment allons-nous juger de la qualité d'une surface construite par cette méthode ? Comme nous l'avons dit à plusieurs reprises toutes les caractéristiques dynamiques des phénomènes simulés doivent être représentées ou tout du moins une majorité conséquente. La surface doit être lisse et représenter au mieux le graphe de voisinage.

## 2.2. Les surfaces implicites

Les surfaces implicites sont des procédés de modélisation et de visualisation de surfaces par techniques implicites. Une technique implicite est soit une fonction algébrique, soit un ensemble d'éléments clés, soit une représentation par fonctions (F-Rep). Pour des explications détaillées sur le sujet entre ces différentes techniques, se reporter à [Menon96].

### 2.2.1 Définition

Une isosurface est une surface implicite. Les quadriques sont des isosurfaces mais toute isosurface n'est pas un quadrique. Une isosurface est la surface engendrée par l'ensemble des points

$$S = \left\{ M \in \mathbb{R}^3 \mid \sum_{i=0}^{i=n} q_i f_i(M) - T = 0 \right\}$$

ayant une même valeur de densité. Elle est définie par la fonction ci-dessus où  $f$  est une fonction de densité de la forme  $f(x, y, z) = ax^n + by^m + cz^p + d$  ou tout autre chose imaginable et  $T$  un scalaire inférieur ou égal à un,  $q$  est une densité qui peut être négative

Il existe plusieurs façons d'exprimer cette fonction potentielle. Les surfaces implicites à squelettes sont des isosurfaces auxquelles ont été ajoutés des éléments clés (centre d'une primitive graphique, caractéristique de la primitive). Les primitives peuvent être de la forme : sphère, cylindre, plan, ....

Blinn la définit par une fonction exponentielle ([Blinn82]), [nishimura85] et [wyvill86] utilisent une fonction polynomiale. Dans les équations ci-dessous,  $R$  est le rayon d'influence de la primitive graphique et  $r$  la distance entre l'isosurface et la primitive graphique.

1. [Blinn82] :  $f(x, y, z) = b_i e^{-a_i r_i^2}$

$$\begin{aligned}
2. \text{ [Nishimura85]} & : f(r) = \begin{cases} 1 - 3\left(\frac{r}{R}\right)^2 & \text{si } 0 \leq r \leq \frac{R}{3} \\ \frac{3}{2} \left[ 1 - \left(\frac{r}{R}\right)^2 \right]^2 & \text{si } \frac{R}{3} \leq r \leq R \end{cases} \\
3. \text{ [Murakami87]} & : f(r) = force * \left[ 1 - \left(\frac{r}{R}\right)^2 \right]^2 \\
4. \text{ [Wyvill86]} & : f(r) = 1 - \frac{4}{9} \frac{r^6}{R^6} + \frac{17}{9} \frac{r^4}{R^4} - \frac{22}{9} \frac{r^2}{R^2}
\end{aligned}$$

### 2.2.2 Rendu des surfaces implicites

Les surfaces implicites sont rendues soit par lancer de rayons, soit par polygonalisation de la surface. Il existe plusieurs méthodes de polygonalisation d'isosurface. La technique des « Marching Cubes » a été mise au point par Lorensen et Cline ([Lorensen87]). Elle a été présentée au chapitre précédent.

Elle a pour principe de discrétiser l'espace entourant l'objet modélisé pour créer un ensemble de facettes qui approximent l'isosurface. Cette méthode est expliquée en détail au Chapitre 7.

Cette méthode a fait l'objet de nombreuses adaptations pour le temps réel, pour l'application de textures sur les surfaces implicites ainsi définies, ... Elle est efficace et donne des résultats satisfaisants. Mais, pour certains des cas de base, il est possible de créer deux ensembles de facettes distinctes. On ne peut savoir à l'avance quel ensemble choisir car cela dépend de la configuration des voxels voisins. Cette indétermination génère des trous dans la surface rendue. C'est l'un des inconvénients majeurs de cette technique. De plus si on l'applique brutalement, la surface présentera des discontinuités de surface peu esthétique bien qu'il soit possible par déplacement des sommets des facettes d'éliminer cet effet.

### 2.3. Rendu par lancer de rayons

Nous utilisons un lancer de rayons du commerce capable de rendre des surfaces implicites. Nous ne pouvons choisir la fonction implicite de notre choix. Une surface implicite est un ensemble de primitives graphiques (sphère ou cylindre) auquel on attribue un centre (deux pour le cylindre), un rayon, et un ensemble de transformations géométriques de base (rotation, mise à l'échelle, translation). Toutes ces primitives s'associent selon la fonction de Murakami. La méthode utilisée est simple mais comporte la gestion d'un ensemble de paramètres pour obtenir un volume satisfaisant. A chaque particule du flux, nous associons une primitive graphique de la fonction implicite.

Cette fonction possède deux paramètres (la force et le rayon de la primitive). Il y a trois types de transformations géométriques<sup>14</sup>, soit neuf paramètres. La surface implicite est déterminée pour un

<sup>14</sup> Ces transformations ont bien sûr un ordre d'application pour obtenir l'effet désiré.

scalaire donné. Cela fait douze paramètres à caractériser, et un choix dans l'ordre des transformations géométriques si on les utilise.

Si l'on veut que la mise à l'échelle de la forme soit effective, il faut positionner la primitive au centre de la scène (0,0,0), appliquer la mise à l'échelle puis la déplacer. La translation effectuée avant la mise à l'échelle influence la mise à l'échelle. La rotation n'est pas une transformation utile dans notre cas. Tous ces paramètres sont interdépendants.

Les primitives de la surface implicite ont la forme de base suivante :

$$\begin{aligned} & \text{Sphère } \{ \langle 0,0,0 \rangle, \text{ rayon, force scale } \langle a,b,c \rangle \text{ translation } \langle x,y,z \rangle \} \\ & \text{Cylindre } \{ \langle 0,0,0 \rangle \langle u,v,t \rangle, \text{ rayon, force scale } \langle a,b,c \rangle \text{ translation } \langle x,y,z \rangle \} \end{aligned}$$

La fonction implicite est égale à 1 si la distance est nulle et 0 si la distance est le rayon de la primitive, mais seulement si la force vaut 1. Si l'on augmente ou diminue la force, la fonction n'a pas le même comportement. Cette caractéristique peut être forte intéressante. Cependant nous avons remarqué que la mise à l'échelle d'une primitive de force non unitaire est égale aux valeurs normales multipliées par la force. Un effet d'augmentation de taille par une mise à l'échelle peut être annulé par une force inférieure à 1, ou décuplé par une force supérieure à 1. Pour ne pas avoir à gérer des interdépendances complexes, la force sera égale à 1 pour la suite de nos tests.

Nous avons cependant réalisé des essais avec des forces différentes de 1, les résultats obtenus étaient plus ou moins satisfaisants mais les paramètres n'avaient pas de relation avec la répartition dans l'espace des particules. Nous avons cherché à ne pas attribuer des paramètres empiriquement, car selon les résultats que nous avons obtenus sur la structuration du flux, il semble tout à fait possible de définir les paramètres à partir de ces informations.

La surface implicite dépend des paramètres des primitives et du seuil de la surface. Si l'on veut une surface cohérente et homogène pour un nuage de particules à visualiser, nous devons fixer en premier la valeur de ce seuil. L'isosurface visualisée le sera pour un seuil de 0,5. Cela ne nous fait plus que quatre paramètres à définir (rayon et translation), ainsi qu'un choix de primitive à effectuer.

Nous avons réalisé un grand nombre de tests plus ou moins pertinents. Ils ont d'abord été faits en deux dimensions puis validés ou repensés pour les modèles tridimensionnels. Le premier a consisté à n'utiliser qu'une seule primitive graphique, la sphère, et d'utiliser comme rayon la distance utilisée pour construire le graphe de voisinage. La surface obtenue était dentelée aussi bien en bordure qu'à l'intérieur. Pour les modèles 2D, nous voulions réussir à visualiser l'intérieur et le contour de manière satisfaisante. Il fallait donc que l'intérieur soit rempli de manière lisse pour que les particularités internes puissent se voir. Une particularité interne est une particularité de distance du flux (particule externe à l'intérieur), ou une caractéristique dynamique. Pour que les deux soient visibles, nous nous sommes appuyées sur les informations préalablement calculées : le vecteur voisin moyen, la détermination des particules aux bords.

Nous avons à spécifier deux ensembles de paramètres, l'un pour les particules aux bords, l'autre pour les particules internes. Nous ne présenterons pas l'ensemble des tests effectués mais expliquerons le cheminement des recherches pour arriver à des paramètres qui, quel que soit le modèle 2D visualisé, donnent une surface satisfaisante.

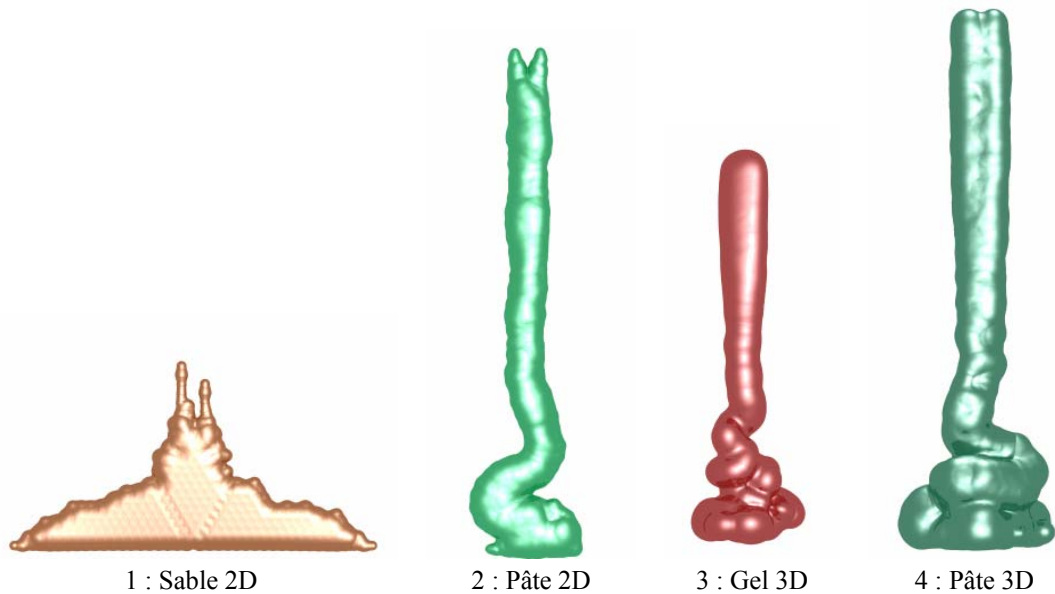
Pour les particules en bordure du nuage de particules nous nous servons du vecteur voisin moyen pour obtenir un contour collant au mieux au nuage de particules. Une sphère sur laquelle est appliquée une mise à l'échelle selon le vecteur voisin moyen n'a pas donné les résultats escomptés dans la mesure où les valeurs du vecteur étaient fortement inférieures à un. Une mise à l'échelle selon le vecteur auquel on ajoute un scalaire donne de meilleurs résultats. Le volume obtenu ne diffère que peu de celui pour lequel il n'y a pas de mise à l'échelle. Nous avons alors opté, pour les particules aux bords, pour un cylindre placé le long du vecteur voisin moyen.

Pour remplir l'espace interne du nuage, les sphères sont les plus adaptées car elles donnent à chaque particule un encombrement. Pour faire ressortir visuellement les particularités du flux, il fallait que la surface interne soit lisse.

Lors des multiples recherches sur la valeur des paramètres nous avons calculé une nouvelle information, la distance séparant une particule de sa voisine directe la plus éloignée. Avec une telle valeur pour rayon de la sphère, nous étions sûr de joindre tous les voisins d'une particule spécifiée dans le graphe de voisinage si l'isosurface en ce voisin lointain valait 0,5. Le rayon de la sphère modifié par un facteur obligeant l'isosurface à être égale à son seuil nous a permis d'obtenir l'effet voulu mais au détriment de gonflements de la surface au niveau du voisin le plus loin. Comme nous effectuions cette modification pour l'ensemble des particules internes, toute la surface interne semblait gonflée, mais nous arrivions ainsi à mettre en valeur les particularités internes (cf. Figure 79.1).

Forts de ces résultats en deux dimensions pour différents modèles, nous les avons utilisés pour les modèles 3D. Il s'est avéré qu'une distinction entre particules internes et particules aux bords n'était pas nécessaire puisque les particules internes sont enveloppées par les particules aux bords donc quasiment invisibles dans le volume final. Nous utilisons pour l'ensemble des particules une sphère de rayon égale à la distance maximale au voisin direct divisée par un certain facteur pour atteindre les particules voisines. Ce facteur peut être supprimé et remplacé par une mise à l'échelle mais cela fait une opération de plus, donc un temps de calcul plus grand. Nous n'utilisons pas de cylindre en 3D, car les temps de calcul deviennent prohibitifs. De plus la surface obtenue est quasiment identique à celle constituée uniquement de sphères.

Les images ci-dessus, ainsi que la planche placée en dernière page du présent chapitre, sont les résultats que l'on a pu obtenir sur différents modèles. Sur le modèle de sable ( $N^{\circ}1$ ) les lignes de force internes sont visibles, le contour n'est pas lisse laissant apparaître les avalanches de surfaces caractéristiques du sable. Le contour du modèle de pâte ( $N^{\circ}2$ ) ne cache pas les extrémités internes des caractéristiques et fait entrevoir que le flux n'est pas ordonné de manière symétrique. Les modèles 3D ( $N^{\circ}3$  et  $N^{\circ}4$ ) mettent en avant le mélange entre couches (tassement) de matière et rendent de manière satisfaisante les dynamiques des modèles.



**Figure 79 : Quelques surfaces implicites obtenues par lancer de rayons**

Le logiciel que nous utilisons a des inconvénients qui ne peuvent être résolus que par l'implémentation d'un autre algorithme. Il est impossible d'effectuer une fusion sélective, c'est-à-dire de ne faire fusionner que les squelettes reliés par le graphe de voisinage. De plus il est impossible de faire une visualisation temps réel ou temps interactif compte tenu du nombre de calculs (à moins de paralléliser les calculs). Nous avons donc choisi de réaliser une polygonalisation de surface implicite pour palier aux inconvénients du lancer de rayons. Cependant le plaquage de texture, (figures de la planche) ne sera pas implémenté.

## 2.4. Rendu par polygonalisation

Nous n'avons pas pour but de réaliser une méthode de conception complète et élaborée pour surfaces implicites. Notre but est de montrer qu'un rendu par isosurface à squelettes est à même de répondre à notre problématique.

Nous n'implémenterons qu'une primitive graphique, la sphère. Le rendu par lancer de rayons nous a montré que le cylindre était inutile en 3D, et que par le jeu de paramètres, il était remplaçable en 2D.

L'encombrement spatial d'une particule évolue avec le temps. Nous avons déjà effectué une visualisation prenant en compte cet aspect. Nous considérons maintenant un seul espace pour chaque particule. Cet espace correspond au volume donné par les caractéristiques des interactions entre masses ponctuelles.

Cette considération implique d'utiliser un rayon unique pour l'ensemble des particules. La conception de l'isosurface se fera à partir des connexions du graphe de voisinage. Il a donc été naturel d'utiliser la distance du graphe comme rayon des sphères.

La réalisation d'une polygonalisation de surface n'est pas triviale. De la structure de stockage des données et de la façon de parcourir l'espace, dépendra la vitesse de calcul de la surface. Nous avons choisi de ne pas stocker d'informations redondantes. Nous disposons d'une liste de facettes et d'un tableau des sommets pour stocker nos données.

Un sommet est une entité composée d'un numéro unique, d'une position et de pointeurs vers les facettes auxquelles il appartient ainsi que d'une normale initialement nulle. Une facette est formée de trois pointeurs vers les structures des sommets qui la définissent et d'une normale correspondant au produit vectoriel de deux vecteurs de base de la facette.

De telles structures sont optimales quant à la quantité de mémoire utilisée, mais plutôt lourdes pour un parcours. Nous ne les parcourons qu'en deux occasions : lors du calcul de la normale en chaque sommet, et lors de la création de la liste d'affichage (« display list »<sup>15</sup>).

Notre méthode de polygonalisation de surface se sert des « Marching Cubes ». Nous avons choisi une fonction implicite différente de celle utilisée pour le lancer de rayons car nous voulions que les joints entre primitives graphiques soient plus lisses sans être obligé d'augmenter les rayons des sphères. Nous utilisons la fonction de Wyvill et al. La fonction a les particularités suivantes :

$$F(0) = 1, F(R) = 0, F'(0) = 0, F'(R) = 1, F\left(\frac{R}{2}\right) = 1$$

Nous avons deux paramètres à régler : la valeur de l'isosurface et le rayon de chaque primitive. Nous n'utilisons qu'un seul type de squelette. Le rendu des modèles 3D par lancer de rayons ne fait pas de distinction entre particules en bordure de flux et à l'intérieur du flux. Nous avons choisi de garder ce critère pour les modèles en deux dimensions. En effet, la méthode implémentée en 2D est une application de la méthode de polygonalisation à une dimension inférieure. Nous ne voulons pas visualiser l'intérieur de la matière, mais juste le remplir, donc faire une distinction n'a pas vraiment de sens.

Le rayon des sphères est la distance utilisée pour le graphe de voisinage. L'isosurface, ainsi que la discrétisation de l'espace est spécifiée par l'utilisateur. Plus la discrétisation est élevée, plus le niveau de détail de la surface est grand. La forte discrétisation n'apporte pas des formes plus fines, mais est plus en accord avec le nuage de particules jusqu'à un certain point, où son augmentation n'apporte qu'un surcoût de calculs.

Le calcul de la surface se fait par parcours de l'espace discrétisé, en calculant pour chaque sommet des voxels la valeur de l'isosurface pour l'ensemble des voisins des particules contenues dans le graphe de voisinage. Il ne s'agit pas d'une fusion sélective à proprement parler mais d'une fusion effectuée selon la distance utilisée pour construire le graphe de voisinage.

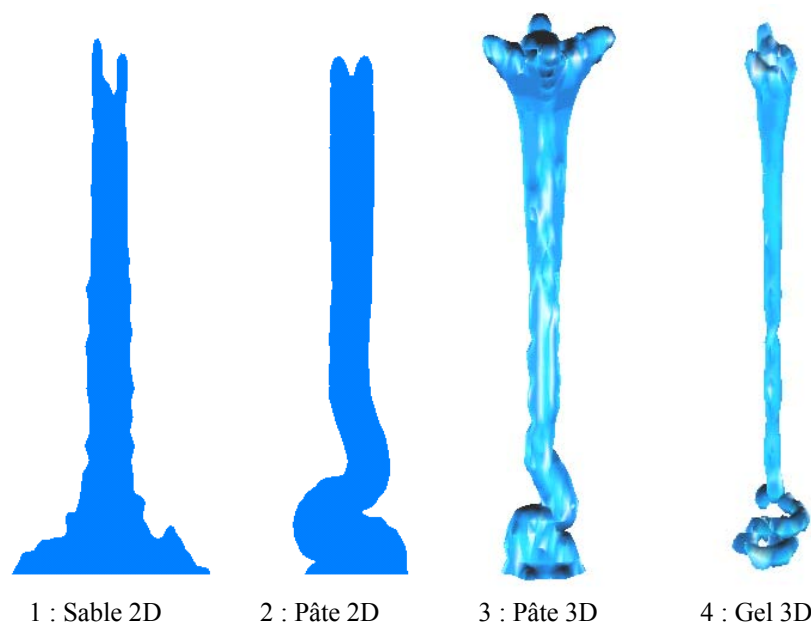
Les résultats obtenus avec des discrétisations moyennes sont satisfaisants. Elles sont en accord avec ce que nous construisons intuitivement en regardant les nuages de particules.

---

<sup>15</sup> Une *display list* est une structure OpenGL de stockage d'objets directement sur la carte graphique permettant ainsi un rafraîchissement plus rapide de l'image.

Certaines formes abruptes de la surface sont dues à un trop faible rayon des sphères mais si l'on augmente ce rayon alors on modifie le graphe de voisinage, donc le nombre de connexions, augmentant ainsi les temps de calcul. Une solution pour y remédier est de multiplier par deux le rayon pour être sûr que l'enveloppe du squelette atteigne tous les voisins, mais cela obligerait à mettre en place une fusion sélective entre squelettes de l'isosurface. Une autre solution plus simple est de diminuer le seuil de l'isosurface, ainsi on augmente de façon virtuelle le volume de chaque particule du modèle.

Comme nous l'avons remarqué pour le rendu par le lancer de rayons, un antialiasage de la surface donne de meilleurs résultats visuels, même si la surface ne change pas. Nous avons donc implémenté un antialiasage de la surface par superposition des tampons d'images pour lesquels les points de vue ont été très légèrement modifiés. Pour ce faire nous utilisons le tampon d'accumulation proposé dans l'implémentation d'OpenGL 1.2. Ce tampon matériel ne se trouve pas sur toutes les machines dont nous disposons. Les images sur fond blanc présentées ci-dessous n'ont pas été traitées par ce moyen. Pour une raison qui nous échappe, lorsque l'on effectue plusieurs passages sur le tampon d'accumulation le fond final de l'image est très légèrement grisé. Des tests sur des machines PC, avec le même algorithme d'antialiasage, n'ont pas eu cet effet indésirable. Nous pensons qu'il s'agit d'une imprécision numérique.



**Figure 80 : Quelques surfaces implicites obtenues par polygonalisation**

### 3. Conclusion

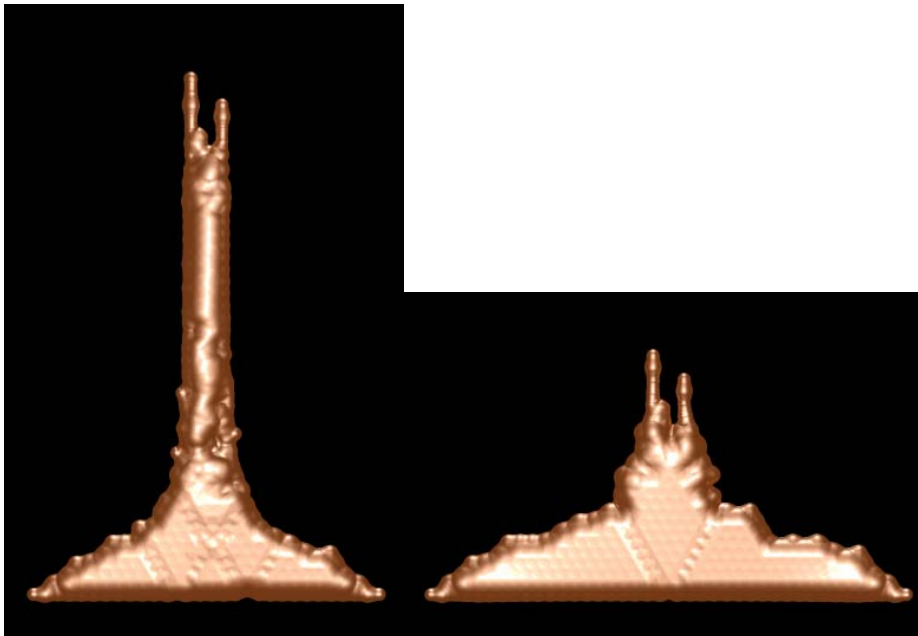
Les résultats obtenus sont satisfaisants puisqu'ils reproduisent les formes des phénomènes, et notamment leurs caractéristiques dynamiques de contour. Les surfaces sont lisses et continues. Le rendu par lancer de rayons a permis de visualiser les formes internes des flux de matière.



Nous avons proposé ici deux méthodes de visualisation par structuration du flux basées sur la construction d'un graphe de voisinage, de recherches d'informations sur la structure de la matière sans avoir recours à la topologie du modèle physique produit par la cinématique des masses ponctuelles.

Les graphes de voisinage sont une façon d'attribuer une structure à la matière visuelle définie par les particules sans attribuer d'espace pour les particules aux bords. Cependant lorsqu'on anime ces simulations par visualisation de ces graphes on remarque peu de changement de structure interne, sauf lorsque des dynamiques caractéristiques se mettent en place.

Ces méthodes conviennent à bon nombre de nos modèles. Les cellules, bien qu'étant des agglomérats, n'ont pas une structure interne composée de particules influençant sa forme. Les fluides turbulents sont plus difficiles à traiter du fait de leur constitution physique. L'utilisation de coulisses (cf. chapitre 5, section 4 : « Les fluides turbulents ») perturbe la construction des graphes et la visualisation par association d'une surface. De plus, comme nous l'avons dit auparavant, même si le graphe de voisinage structure la matière, elle ne le fait pas forcément dans la direction à laquelle on s'attendait.





# Chapitre 9 Optimisation des calculs d’habillage : vers un rendu temps réel

---

La méthode des « Marching Cubes » que nous avons utilisé pour le rendu des écoulements de pâtes et de gels, permet d’obtenir assez facilement une représentation des surfaces implicites générées. Toutefois, la simplicité de cet algorithme impose d’importantes limitations tant sur la qualité du rendu que sur la vitesse des calculs nécessaires.

Nous proposons un algorithme de rendu de surfaces implicites permettant de diminuer l’impact des trois principales limitations que nous avons identifiées, en tirant parti des spécificités des écoulements de pâtes et de gels.

Ces limitations sont essentiellement de trois types différents :

- Apparition d’artefacts visuels
- Apparition de discontinuités temporelles dans les animations
- Vitesse de calcul limitée

## 1. Artefacts visuels

La nature discrète de la méthode de Lorensen et Cline crée inévitablement des artefacts visuels : la transition entre l’extérieur et l’intérieur de l’écoulement de pâte, par exemple, est particulièrement abrupte et peut présenter des fréquences très élevées en regard de la fréquence d’échantillonnage de l’espace de voxels.

Ainsi, une discrétisation grossière pourra aller jusqu’à occulter un objet trop petit. D’une manière plus générale, une discrétisation insuffisante pour un objet de petite taille ne permettra d’en offrir qu’une représentation trop simplifiée.

On peut diminuer l'impact visuel de ces artefacts en augmentant la finesse de discrétisation, mais cette amélioration de l'image se fait au prix d'une multiplication du temps de calcul : la complexité du calcul est en effet en  $O(n^3)$  pour une finesse  $n$  donnée (i.e. une discrétisation 2 fois plus fine impose 8 fois plus de calculs). L'échantillonnage de l'espace dans cette méthode étant uniforme, la prise en compte des détails de petite taille passe donc inévitablement par une augmentation générale de la finesse de discrétisation.

Toutefois, on peut généralement limiter la fréquence d'échantillonnage de manière à s'assurer qu'à chaque pixel de l'écran correspondra la projection d'au minimum un voxel. (i.e. pour une image générée dans une résolution de qualité broadcast (768 x 576), un espace de 768 voxels de coté sera suffisant). Toutefois, afin de diminuer les effets d'aliasage, un sur-échantillonnage de l'espace sera préférable (en pratique sur nos images, un espace de 1024 voxels de coté donne visiblement de bons résultats).

## 2. Discontinuités temporelles

La discrétisation provoque également des artefacts disgracieux dans les animations : des discontinuités temporelles apparaissent quand le déplacement de l'isosurface d'un voxel à l'autre entraîne le déplacement brusque d'un élément de la surface.

De plus, des objets de petites tailles qui apparaissent sur une image peuvent disparaître sur la suivante au gré de leur déplacement.

Encore une fois, ces artefacts peuvent être diminués par une augmentation de la finesse de discrétisation ; mais on ne peut plus alors poser la résolution de l'image générée comme limite à la fréquence d'échantillonnage.

## 3. Limitation de la vitesse de calcul

Une évaluation théorique des temps nécessaires aux calculs du rendu de surfaces implicites par la méthode précédemment utilisée nous laisse espérer, moyennant une bonne optimisation de la quantité de calcul nécessaires à chaque image, un rendu en temps réel (au minimum à 25 images par seconde) de nos animations sur des microprocesseurs modernes.

### 3.1. Estimation des performances en calcul flottant de processeurs contemporains

Afin de produire une estimation des temps de calcul imposés par les « Marching Cubes » dans le cas de la visualisation d'écoulement, nous sommes partis du principe que la charge principale de calcul se trouvait dans le calcul de la fonction de densité générant la surface implicite :

$$f(r) = 1 - \frac{4}{9} \frac{r^6}{R^6} + \frac{17}{9} \frac{r^4}{R^4} - \frac{22}{9} \frac{r^2}{R^2}$$

avec  $r = (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2$

Dans l'algorithme de base, cette fonction est calculée pour chacun des sommets, pour chacun des voxels, pour chacune des particules génératrices du squelette de la surface.

Autrement dit, pour un écoulement de pâte composé de 900 masses ponctuelles rendu dans un espace discrétisé en 768 voxels de côté, on aura donc  $768^3 \times 8 \times 900$  de ces calculs, soit un total de  $3,26.10^{12}$  calculs de  $f(r)$ .

En pratique, une optimisation de l'algorithme nous permet de réduire grandement le nombre de ces calculs. En ne recalculant pas plusieurs fois les sommets communs aux voxels, on arrive à réduire la quantité de calcul à une évaluation par voxel (soit 8 fois moins de calcul).

En utilisant le graphe de voisinage (cf. Partie I - Chapitre 8) on réduit, pour chaque voxel, la contribution des particules génératrices à la valeur du potentiel, uniquement à celle des particules adjacentes. Ainsi au lieu de calculer à chaque fois les potentiels des 900 particules génératrices de la simulation de pâte, on ne calcule les contributions que de ceux figurant dans le graphe de voisinage : moins de 5 en moyenne, compte tenu du fait que de grandes parties de l'espace ne sont pas du tout décrites.

L'optimisation de l'algorithme nous permet donc en moyenne de faire 1500 fois moins de calculs, réduisant le nombre de calculs à environ 2 milliards par image, soit 50 milliards de calculs par seconde si on veut obtenir une animation à 25 images par seconde.

Cette évaluation grossière nous révèle que notre objectif d'obtenir une animation fluide de notre pâte dans une telle résolution n'est pas réaliste. En effet, les microprocesseurs actuels fonctionnent dans des fréquences de l'ordre du gigahertz, et il paraît peu probable que ces processeurs soient capables d'effectuer 50 calculs de notre fonction de densité à chaque cycle d'horloge.

Dès lors, nous allons nous contenter d'une discrétisation moins fine afin d'alléger les calculs. Une évaluation totalement subjective aimablement réalisée par un ensemble d'étudiants nous a révélé qu'une discrétisation de 250 voxels de côté présentait un bon compromis entre la réduction de la complexité et la qualité du rendu.

Dans ce cas, la quantité de calculs avec l'algorithme optimisé se réduit à  $250^3 \times 5 = 78125000$  de calculs par image, ou encore deux milliards par seconde.

Pour évaluer la vitesse limite d'un microprocesseur donné, nous allons estimer le nombre minimal de cycles d'horloge demandé à ce microprocesseur pour calculer une valeur de la fonction de densité.

### 3.1.1 Evaluation théorique de la vitesse maximale de calcul de la fonction de densité

Pour cette évaluation, nous partons des données fournies par les constructeurs concernant la latence<sup>16</sup> et le débit<sup>17</sup> des instructions des processeurs. Nous nous sommes intéressés à six microprocesseurs différents, microprocesseurs qui étaient utilisés par des ordinateurs du laboratoire :

- Mips R8000, utilisé par une SGI PowerChallenge hexaproscesseur
- Mips R10000, utilisé par les SGI Octane
- Intel Pentium III, utilisé par les compatibles PC
- Intel Pentium IV, utilisé par une station de travail de type PC
- AMD Athlon, utilisé par les compatibles PC
- Motorola PowerPC 7400, utilisé par les Apple Macintosh G4

	MIPS R10000		MIPS R8000		Intel Pentium III		Intel Pentium 4		AMD Athlon XP		Motorola PowerPC G4	
Lecture	3	1	1	1	3	1			2	1		
Ecriture	3	1	1	1	1	1			2	1		
Addition/Soustraction	2	1	4	1	3	1	5	1	4	1	4	1
Multiplication	2	1	4	1	5	0.5	7	2	4	1	4	1
Division	12	14	14	11	Stall(1/2)		23	2 3	16	1		
Multiply And Add	2	1	4	1	n/a		n/a		n/a		4	1

**Tableau 6 : Débit et latence d'instructions pour six microprocesseurs**

Nous évaluons ensuite le nombre de cycles d'horloges nécessaires au calcul d'une valeur de la fonction de densité (ou plutôt de sa traduction en assembleur).

Compte tenu des différences d'architecture entre les machines évaluées, le calcul doit être fait en fonction du code assembleur de chacune des architectures (MIPS, x86, PowerPC). Effectivement, des différences importantes existent dans la façon d'utiliser les instructions qui ne permettent pas de faire un programme assembleur symbolique équivalent pour tous les processeurs.

Par exemple, les architectures MIPS et PowerPC offrent une instruction permettant de faire une multiplication suivie d'une addition en une seule opération (l'architecture x86 offre une instruction similaire uniquement dans ses unités vectorielles, mais pas dans ses unités scalaires). De même, les x86, de part leur architecture CISC<sup>18</sup>, permettent de faire des calculs directement depuis la mémoire sans avoir à charger préalablement des registres comme avec les MIPS ou les PowerPC.

<sup>16</sup> Nombre de cycles d'horloge nécessaires pour que le résultat d'une instruction soit disponible (pour une autre instruction).

<sup>17</sup> Nombre de cycles d'horloge nécessaires entre le lancement de 2 instructions similaires.

<sup>18</sup> CISC : Complex Instructions Set Computer

$$f(r) = 1 - \frac{4}{9} \frac{r^6}{R^6} + \frac{17}{9} \frac{r^4}{R^4} - \frac{22}{9} \frac{r^2}{R^2}$$

$$d(r) = \frac{r^2}{R^2}$$

et  $r = (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2$

$$f(r) = 1 - \frac{4}{9} d^3(r) + \frac{17}{9} d^2(r) - \frac{22}{9} d(r)$$

$$= 1 - d(r) \left[ \frac{4}{9} d^2(r) - \frac{17}{9} d(r) + \frac{22}{9} \right]$$

$$= 1 - d(r) \left[ d(r) \left( \frac{4}{9} d(r) - \frac{17}{9} \right) + \frac{22}{9} \right]$$

### Equation 1 : Factorisation de la fonction d'isosurface

On traduit l'équation factorisée en assembleur (à partir de la version compilée d'un programme l'utilisant, en ayant activé toutes les optimisations pour la vitesse calcul), sans tenir compte des ordonnancements d'instructions (« scheduling ») nécessaires aux différentes implémentations des processeurs (autrement dit, pour les microprocesseurs partageant le même jeu d'instruction, nous n'avons pas organisé l'ordre des instructions en fonction du séquençement optimal demandé par le processeurs ; mais nous avons tout de même tenu compte de la disponibilité des différents unités d'exécution). Effectivement, nous ne cherchons pas à obtenir un temps exact de calcul mais plutôt le nombre de cycles demandé au minimum par le processeur.

Les tableaux suivants contiennent le code assembleur correspondant à l'équation pour chacune des architectures auxquelles nous nous sommes intéressés : MIPS IV et Intel x86 pour les processeurs MIPS R8000, MIPS R10000, Intel Pentium III Katmai et AMD Athlon XP.

En face de chacune des instructions, nous avons mis le nombre de cycles d'horloge nécessaire avant que l'instruction puisse débiter son exécution. Ces instructions s'exécutant sur des microprocesseurs super-scalaires, nous ne précisons le nombre de cycle d'horloge qu'au moment où il est calculable : c'est à dire au moment où une instruction est tenue d'attendre que les instructions qui la précèdent soient terminées (utilisation de leur résultat) ou bien que certaines instructions précédemment exécutées aient libéré des ressources (si par exemple un instruction monopolise longuement l'unité assurant le calcul des division, une autre instruction du même type ne pourra être lancée immédiatement).

Prenons par exemple, sur un processeur MIPS, les trois instructions suivantes :

```
lwc1   F1,x           // Charge la valeur de x dans le registre F1
lwc2   F2,y           // Charge la valeur de y dans le registre F2
sub.s  F1, F1, F2     // Place dans F1 le résultat de F1 - F2
```

Les deux premiers chargements sont indépendants et s'exécutent simultanément.

L'instruction « **sub.s** » qui les suit est par contre tenue d'attendre que les 2 chargements soient terminés avant de pouvoir utiliser leur valeur (F1 ou F2). La latence d'un chargement étant de 3 cycles,



le « **sub.s** » ne peut débiter qu'à l'issue de ces trois cycles : nous précisons donc +3 dans la colonne « cycles » du tableau.

Assembleur MIPS	Cycles R1000	Cycles R8000
// $r=(x-x1)^2 + (y-y1)^2 + (z-z1)^2$ lwc1 F1, x lwc1 F2, x1 sub.s F1, F1, F2 // $x-x1$ move F2, F1	+3 (loads) +2 (sub)	+1 +4
mul.s F1, F1, F2 // $(x-x1)^2$ lwc1 F3, y lwc1 F4, y1 sub.s F3, F3, F4 // $y-y1$	+1 (move) +3 (loads)	+1 +1
move F4, F3 madd.s F1, F1, F3, F4 // $(x-x1)^2 + (y-y1)^2$ lwc1 F5, z lwc1 F6, z1	+2 (sub) +1 (move)	+4 +1
sub.s F5, F5, F6 // $z-z1$ move F7, F5 madd.s F1, F1, F5, F6 // $(x-x1)^2 + (y-y1)^2 + (z-z1)^2$	+3 (loads) +2 (sub) +1 (move)	+1 +4 +1
// $d=(r*r)/(R2)$ lwc1 F2, r lwc1 F2, r mul.s F2, F2, F3	+3 (loads)	+1
lwc1 F4, INVR2 mul.s F2, F2, F4	+3 (load)	+1
// $f=1 - d * (d * (4/9 * d - 17/9) + 22/9)$ lwc1 F3, 1.000000f lwc1 F4, 4/9f lwc1 F5, -17/9f	+1 (2 load ports)	+1
madd.s F4, F4, F2, F5 // $4/9 * d - 17/9$ lwc1 F6, 22/9f madd.s F4, F4, F2, F6 // $d * (4/9 * d - 17/9) + 22/9$ nmsub.s F3, F4, F2, F3 // $1 - d * (d * (4/9 * d - 17/9) + 22/9)$	+3 (loads) +3 (load) +2 (madd)	+1 +1 +4
swe1 f, F3	+2 (nmsub) +3 (store)	+4 +1
Nombre total de cycles d'horloge nécessaires :	38	32

L'assembleur x86 est un peu particulier car son unité arithmétique est basée sur l'utilisation d'une pile de 8 niveaux (st(0) à st(7)) auxquels on peut accéder directement comme dans une architecture à registres classique. Les instructions sont appliquées sur les niveaux les plus hauts de la pile (st(0), st(1)) sauf si des paramètres ont été précisés. Les résultats sont placés en st(0).

fld : Load – chargement d'un flottant dans la pile  
fsub : Sub – soustraction

fadd : Add – Addition

fmul : Mul - Multiplication

fstp : Store And Pop – Ecriture en mémoire de st(0) puis Pop de st(0)

Assembleur x86	Cycles Pentium III	Cycles AMD Athlon XP
$// r=(x-x1)^2 + (y-y1)^2 + (z-z1)^2$ fld dword ptr [__real@x] fsub dword ptr [__real@x1] // fx = x-x1 fld dword ptr [__real@y]	+3 (load)	+2
fsub dword ptr [__real@y1] // fy = y-y1 fld dword ptr [__real@z] fsub dword ptr [__real@z1] // fz = z-z1	+3 (load) +3 (load)	+2 +2
$// r = fx*fx + fy*fy + fz*fz$ fld st(0) fmul st,st(1) fld st(2) fmul st,st(3)	+1 (pair) +1 (2 pipes) +1 (load)	+2 +4
faddp st(1),st fld st(3) fmul st,st(4) faddp st(1),st	+1 +1 (load) +1	+2 +2 +2
fstp st(3) fstp st(0) fstp st(0)	+1 +1	+1 +2
$// d=(r*r)/(R2)$ fld st(0) fmulp st(1),st fmul dword ptr [__real@(hash_1/R2)]	+1 (load) +5	+2 +4
$// f = 1 - d * ( d * (4/9*d + 17/9) - 22/9)$ fld dword ptr [__real@_4/9] fmul dword ptr [ebp-0Ch] fadd dword ptr [__real@_17/9] fmul dword ptr [ebp-0Ch] fsub dword ptr [__real@_22/9] fmul dword ptr [ebp-0Ch] fsubr dword ptr [__real@_1] fstp dword ptr [__real@(f)]	+5 +5 +5 +5 +5 +2	+4 +4 +4 +4 +3
Nombre total de cycles d’horloge nécessaires :	40	42

Dés lors, les performances théoriques maximales de calcul de notre fonction de densité sont :

Microprocesseur	Fréquence	Cycles/calcul	Calculs de la fonction de densité
MIPS R8000	75 MHz	32	2 340 375 calculs/seconde
MIPS R10000	175 MHz	38	4 605 263 calculs/seconde
Intel Pentium III	1000 MHz	40	25 000 000 calculs/seconde
AMD Athlon XP	1733 MHz	42	41 268 809 calculs/seconde

**Tableau 7 : Evaluation de la vitesse maximal théorique de calcul de la fonction de densité**

### 3.1.2 Mesure de la vitesse maximale de calcul de la fonction de densité

Nous avons également évalué le temps de calcul de notre fonction de densité sur les machines dont nous disposons :

- ❑ Le programme de test utilisé consiste en une boucle calculant deux cent vingt millions de fois une itération de la fonction de densité.
- ❑ Le programme est compilé avec les paramètres d'optimisation maximale, par le compilateur fourni par le constructeur du système testé.
- ❑ Pour éviter que les optimisations proposées par le compilateur évincent une partie des calculs (quand le compilateur constate par exemple que les résultats ne seront pas réutilisés), les résultats de chaque calcul ont à chaque fois été utilisés parmi les paramètres d'entrée du calcul suivant.
- ❑ Nous avons également veillé à ce que les résultats restent dans un domaine réaliste et ne provoque pas d'exception de l'unité arithmétique (un calcul aboutissant ou mettant en œuvre une donnée non valide de type NaN (« Not a Number ») peut selon le processeur aboutir en un temps plusieurs fois inférieur à la normal, ou peut au contraire ralentir drastiquement le processeur).

Système	Microprocesseur	Fréquence	Temps	Calculs de la fonction de densité
Silicon Graphics Octane	MIPS R10000	175 MHz	48 s	4 655 349 calculs/s
Silicon Graphics Power Challenge	MIPS R8000	75 MHz	124 s	1 802 070 calculs/s
Silicon Graphics O2	MIPS R5000	300 MHz	42 s	5 320 399 calculs/s
Apple PowerMacintosh G3	IBM PowerPC 750	500 MHz	21 s	10 640 799 calculs/s
Apple PowerMacintosh bi-G4	Motorola PowerPC 7400	450 MHz	24 s	9 310 700 calculs/s
Gateway Solo 9300	Intel Pentium III Mobile	500 MHz	19 s	11 740 883 calculs/s
Dell Dimension 4100	Intel Pentium III	1000 MHz	10 s	22 345 678 calculs/s
Dell Dimension 8100	Intel Pentium IV	1500 MHz	12 s	18 621 398 calculs/s
AMD	AMD Athlon	900 MHz	15 s	14 897 118 calculs/s

**Tableau 8 : Evaluation de la vitesse réelle de calcul de la fonction de densité**

Afin de compléter nos évaluations de vitesse de calcul des microprocesseurs du marché, nous nous sommes appuyés sur les résultats des outils d'évaluation Spec CPU 2000, ou plutôt sur la partie de ces outils qui se révèle proche de nos calculs.

### 3.1.3 Evaluation de la vitesse de calcul de la fonction de densité à partir des tests SpecFP 2000

Les outils Spec CPU 2000 (<http://www.spec.org>) sont un ensemble de logiciels de calculs destinés à évaluer les performances des microprocesseurs du commerce, distribués sous forme de code source C et Fortran. Ils sont utilisés par la quasi-totalité des fabricants de microprocesseurs ou d'ordinateurs et constituent donc une base fiable de comparaison de performances.

Ces tests sont constitués d'un ensemble de logiciels de calcul scientifique. Ces derniers ont été choisis pour tenter de mettre en œuvre l'ensemble des unités du microprocesseur, en évitant par exemple de faire des programmes qui tiendraient intégralement dans la mémoire cache des processeurs et qui révéleraient alors non pas les performances générales du système mais simplement les performances particulières liées à sa mémoire cache.

La sous-partie Spec FP 2000 du test Spec CPU 2000 est constituée d'un ensemble de programmes de calculs destinés à évaluer les performances en calculs flottants :

<i>168.wupwise</i>	<i>Fortran 77</i>	<i>Physics / Quantum Chromodynamics</i>
<i>171.swim</i>	<i>Fortran 77</i>	<i>Shallow Water Modeling</i>
<i>172.mgrid</i>	<i>Fortran 77</i>	<i>Multi-grid Solver: 3D Potential Field</i>
<i>173.applu</i>	<i>Fortran 77</i>	<i>Parabolic / Elliptic Partial Differential Equations</i>
<i>177.mesa</i>	<i>C</i>	<i>3-D Graphics Library</i>
<i>178.galgel</i>	<i>Fortran 90</i>	<i>Computational Fluid Dynamics</i>
<i>179.art</i>	<i>C</i>	<i>Image Recognition / Neural Networks</i>
<i>183.quake</i>	<i>C</i>	<i>Seismic Wave Propagation Simulation</i>
<i>187.facerec</i>	<i>Fortran 90</i>	<i>Image Processing: Face Recognition</i>
<i>188.ammmp</i>	<i>C</i>	<i>Computational Chemistry</i>
<i>189.lucas</i>	<i>Fortran 90</i>	<i>Number Theory / Primality Testing</i>
<i>191.fma3d</i>	<i>Fortran 90</i>	<i>Finite-element Crash Simulation</i>
<i>200.sixtrack</i>	<i>Fortran 77</i>	<i>High Energy Nuclear Physics Accelerator Design</i>
<i>301.apsi</i>	<i>Fortran 77</i>	<i>Meteorology: Pollutant Distribution</i>

Pour nos évaluations, nous avons pris en compte les performances mesurées des 20 microprocesseurs les plus rapides du marché (en Juillet 2002) dans les tests 171.swim et 172.mgrid<sup>19</sup>. Ces tests ont été choisis car ils présentent un accès répétitif aux données qui imposent une activité mémoire similaire à celle que nous connaissons lors de nos calculs de densité. De plus, pour ces deux tests, la proportion entre calculs et accès mémoire est comparable à celle de nos calculs.

Nous obtenons les résultats suivants présentés au Tableau 9.

Nous disposons maintenant de trois séries de résultats d'évaluation de performances dont nous allons nous servir pour établir un classement global des processeurs susceptibles de nous fournir une estimation de la complexité et de la vitesse d'animation qu'il est possible d'obtenir sur des microprocesseurs contemporains.

---

<sup>19</sup> Le détail de ces tests se trouve en Annexe

Microprocesseur	171.swim	172.mgrid	(SPECfp2000)	171+172
Intel Itanium 2 – 1000 MHz	3205	1720	1356	4925
IBM POWER4 – 1300 MHz	1915	782	1202	3117
Intel Pentium 4 – 2533 MHz	1425	845	878	2303
Intel Pentium 4 Xeon – 2400 MHz	1313	762	803	2116
API Alpha 21264C – 1000 MHz	1502	517	776	2278
Intel Itanium – 800 MHz	1369	749	703	2072
API Alpha 21264B – 833 MHz	1112	420	643	1755
Sun UltraSPARC-III – 1056 MHz	944	479	711	1655
API Alpha 21264A – 833 MHz	980	587	571	1551
AMD Athlon MP – 1733 MHz	895	564	584	1479
AMD Athlon XP – 1800 MHz	840	551	624	1464
HP PA-RISC – 750 MHz	779	529	526	1305
AMD Athlon – 1400 MHz	797	343	426	1223
Sun UltraSPARC-III – 900 MHz	752	377	427	1179
HP PA-RISC 8600 – 552 MHz	761	462	400	1161
MIPS R14000 – 600 MHz	529	379	499	1028
Sun SPARC64 GP – 810 MHz	536	390	483	1009
IBM POWER3-II – 450 MHz	402	330	426	828
Intel Pentium III – 1400 MHz	343	303	437	780
IBM POWER RS64-IV – 750 MHz	355	220	396	751

**Tableau 9 : Classement des 20 meilleurs processeurs aux tests 171.swim et 172.mgrid (du plus rapide au plus lent)**

Si les deux premières séries de résultats (évaluation théorique et mesure de la vitesse de calcul de la fonction de densité) semblent cohérents, certaines informations complémentaires (notamment les performances des systèmes de mémoire des machines testées) doivent être prises en compte avant de pouvoir les mettre en relation avec la troisième série de résultats.

### 3.2. Evaluation globale des performances en animation

Nos évaluations de la vitesse maximale de calcul de la fonction de densité sont un reflet des performances maximales de l'unité arithmétique des processeurs testés dans le cas du calcul de la fonction de densité. Par contre, ces résultats peuvent être très éloignés des résultats d'une application réelle (comme 171.swim et 172.grid du test SpecFP 2000) étant donné qu'ils ne prennent pas en compte d'autres paramètres limitants des microprocesseurs, que ceux liés à l'unité flottante.

Effectivement, en pratique, l'exécution des calculs de rendu de surfaces implicites par la méthode des « Marching Cubes » se déroule dans des temps très supérieurs aux temps théoriques. Une analyse rapide du déroulement de tels calculs nous révèle que dans certaines conditions les performances flottantes d'un processeur peuvent n'avoir qu'une influence mineure sur le temps de calcul global.

En fonction du type de microprocesseur, de la taille des mémoires caches mises en œuvre, de la taille de la mémoire centrale et de la finesse de discrétisation de l'espace nous avons ainsi pu constater que lors du calcul de rendu, le microprocesseur pouvait être pénalisé par les aléas d'accès aux données.

Autrement dit, le processus de calcul peut passer l'essentiel de son temps à attendre que des requêtes systèmes aboutissent (« temps système » au lieu de « temps utilisateur ») ; ou attendre que des données soit chargées depuis la mémoire centrale quand elles ne sont pas présentes dans la mémoire cache (la mémoire centrale peut-être jusqu'à une centaine de fois plus lente que la mémoire cache) plutôt que de calculer les fonctions de rendu. Une observation plus précise de l'activité mémoire révèle que la part importante passée par nos calculs en « temps système » est due aux défauts de page.

Ces défauts de page (« pagefaults ») surviennent quand une donnée de la mémoire n'est pas disponible pour le processus de calcul au moment où il en fait la demande. Le processus qui a fait la demande est alors mise en attente : ces défauts de page peuvent être d'une sévérité plus ou moins importante selon que la page mémoire demandée est présente dans la mémoire centrale ou dans le pire des cas sur le disque, en mémoire virtuelle.

Sur le même principe, des défauts de cache surviennent dans le microprocesseur lorsqu'une instruction demande une donnée qui n'est pas présente en mémoire cache, et que cette instruction est gelée en attendant de charger la donnée depuis un niveau de cache inférieur ; ou depuis la mémoire centrale.

Ainsi, en raison du volume important de données qu'impose la discrétisation de l'espace, les calculs de rendus peuvent être limités non pas par la puissance de calcul du processeur mais par les performances de son système de gestion de la mémoire.

Outre les besoins en puissance brute, il nous faut donc évaluer les besoins en espace mémoire et en bande passante de notre algorithme de rendu de surfaces implicites.

Lors de nos calculs, nous mettons en œuvre deux types de structures consommatrices d'espace mémoire.

Nous devons décrire d'une part l'espace discrétisé (les voxels et les données attenantes) ; et d'autre part, le maillage calculé à partir de l'espace discrétisé.

### *Espace discrétisé*

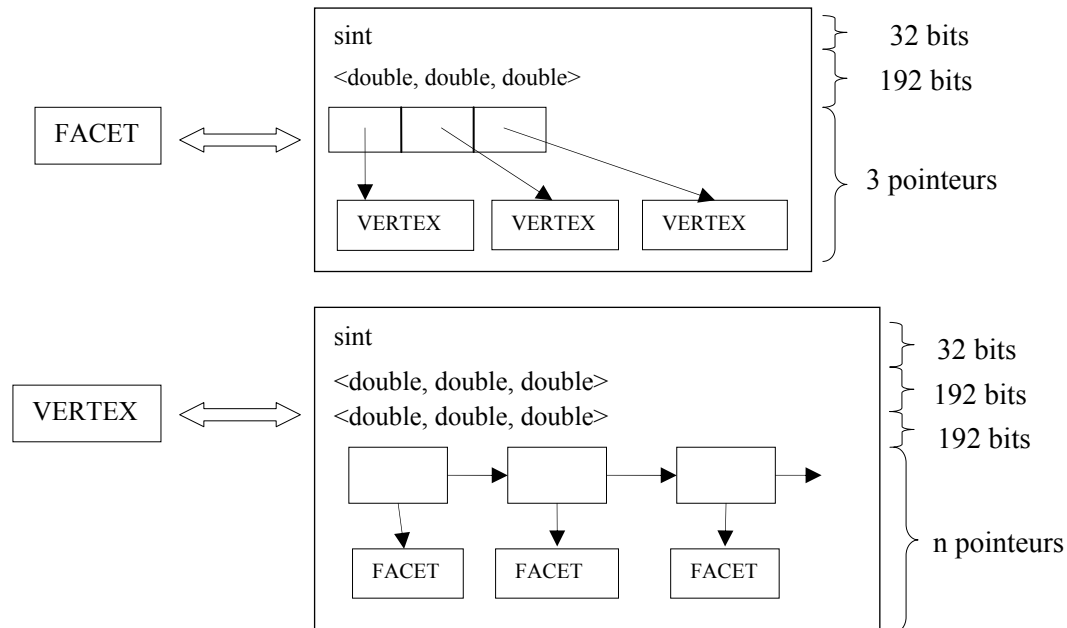
L'espace discrétisé est un tableau de voxels. Chaque voxel est composé de :

- Un octet indiquant si ses huit sommets sont à l'intérieur ou à l'extérieur du volume implicite
- Trois flottants donnant la normale à la surface implicite
- Une liste de pointeurs vers les facettes partageant ce sommet (2,5 pointeurs en moyenne)

## Maillage

Le maillage décrit l'ensemble des facettes qui composent la surface de l'objet rendu (cf. Figure 81).

Sur les stations de travail que nous avons utilisé pour le rendu, les images étaient typiquement produites avec une définition de type broadcast (768\*576 pixels). En pratique, à cette définition, une discrétisation comprenant 256 voxels de coté présente un aspect visuel correct pour l'animation.



**Figure 81 : Structure de stockage du maillage**

Pour un espace de 256 voxels de coté, notre structure d'espace discret requiert  $256 * 256 * 256 = 16777216$  voxels ;

- ❑ Avec un modèle mémoire 32 bits l'occupation mémoire est :  
1 voxel : 1 octet + 3 flottants 64 bits + 2,5 pointeurs 32 bits (en moyenne)  
= 35 octets  
soit pour 16777216 de voxels : 587 Mo par image
- ❑ Avec un modèle mémoire 64 bits,  
1 voxel : 1 octet + 3 flottants 64 bits + 2,5 pointeurs 64 bits  
= 45 octets  
soit 755 Mo par image

Ainsi, si on veut obtenir une animation en 25 images par seconde, la bande passante minimale requise pour parcourir l'espace discret sera donc de  $587 \text{ Mo} * 25 = 14,7 \text{ Go/s}$  pour un modèle mémoire 32 bits, étant donné que les 587 Mo décrivant l'espace seront intégralement parcourus et renouvelés à chaque image.

En 64 bits, la bande passante requise sera de 18,9 Go/sec

Pour une image similaire, le maillage sera composé en moyenne de 150 000 sommets et 700 000 facettes. Ce qui correspond pour un modèle 32bits à un espace mémoire de :

Sommets :  $150\,000 * (32 \text{ bits} + 192 \text{ bits} + 192 \text{ bits} + 3 * 32 \text{ bits}) = 9 \text{ Mo par image}$

Facettes :  $700\,000 * (32 \text{ bits} + 192 \text{ bits} + 3 * 32 \text{ bits}) = 26 \text{ Mo par image}$

Et pour un modèle mémoire 64 bits à :

Sommets :  $150\,000 * (32 \text{ bits} + 192 \text{ bits} + 192 \text{ bits} + 3 * 64 \text{ bits}) = 11 \text{ Mo par image}$

Facettes :  $700\,000 * (32 \text{ bits} + 192 \text{ bits} + 3 * 64 \text{ bits}) = 34 \text{ Mo par image}$

Ainsi, les besoins en bande passante de notre implémentation de l'algorithme de polygonalisation de la surface sont :

	Modèle mémoire 32 bits		Modèle mémoire 64 bits	
	Taille par image	Bande passante requise	Taille par image	Bande passante requise
Espace discrétisé	587 Mo	14,7 Go/sec	755 Mo	18,9 Go/sec
Maillage	35 Mo	0.875 Go/sec	45 Mo	1,1 Go/sec

Nous avons rassemblé ci-après les caractéristiques des systèmes de gestion de la mémoire des microprocesseurs comptant parmi les plus puissants du marché : le tableau donne, lorsque nous avons trouvé l'information, la taille des mémoires cache de différents niveaux (L1 étant la mémoire cache de niveau 1, soit la mémoire de plus bas niveau, généralement la plus rapide (très faible latence) ; et L3 étant, quand elle est présente, la mémoire cache la plus proche de la mémoire centrale (elle est d'ailleurs parfois de même nature (DDR-SDRAM) que la mémoire centrale)) et les débits des différentes mémoires – débits que nous pouvons comparer aux débits requis dans le tableau précédent.

Microprocesseur	Taille des caches (Ko)			Débits des caches (Go/sec)			Débit Mémoire
	L1	L2	L3	L1	L2	L3	
HP PA-8700 750 MHz	2304	-	-	36	-	-	1,9 Go/sec
IBM Power4 1300 MHz	96	1440	128000		100	11	11 Go/sec (2*128bits DDRSDRAM 200 MHz)
Sun Ultrasparc III Cu 1 GHz	96	8192		32	32		4,8 Go/sec (128 bits, SDRAM 150 MHz)
MIPS R14000 600 MHz	64	8192					2 Go/sec (2*64 bits, SDRAM 100 MHz)
Motorola PowerPC G4 1 GHz	64	256	2048	32	32	4	1 Go/sec (64 bits, SDRAM 133 MHz)
Intel Pentium IV 2533 MHz	32	512	0	81	81		3,2 Go/sec (2*16 bits, RDRAM 1066 MHz)
AMD Athlon XP 1600 MHz	128	256		51	51		2 Go/sec (64 bits, DDR-SDRAM 266 MHz)
Intel Itanium 800 MHz	32	96	4096				
Intel Itanium II 1000 MHz	32	256	3072	64			6,4 Go/sec (2*64 bits, DDR-SDRAM 200 MHz)
API Alpha 21264 1000 MHz		8192					8 Go/sec (2*256 bits, 125 MHz)

On voit donc que quel que soit le processeur utilisé, la bande passante requise par nos calculs dépasse largement la bande passante théorique du sous-système mémoire du processeur. Si les caches de niveau 1 offrent généralement une bande passante suffisante (ils débitent généralement de 64 à 384 bits par cycle d'horloge, généralement plusieurs dizaines de gigaoctets par seconde), la taille limitée de ces caches rend cet avantage inutile dans le cas des gros volumes de données que nous considérons. (Les structures parcourues pour le calcul d'une image sont plus de cent fois plus grandes que les mémoires caches de niveau 1 de ces processeurs).



Le but de nos évaluations était de déterminer quels processeurs étaient susceptibles de permettre un calcul d'habillage de nos écoulements de pâte en temps réel : nous pouvons dès à présent écarter la quasi-totalité des prétendants : seuls l'IBM Power 4, l'API Alpha 21264 et l'Intel Itanium II offrent des bandes passantes mémoire du même ordre que la bande passante requise (débit mémoire offert représentant de 30 à 60 % du débit requis).

Pour ces 3 processeurs, on évalue le nombre de calculs de la fonction de densité qu'ils sont susceptibles de faire chaque seconde, en interpolant les résultats des tests 171.swim et 172.mgrid.

Système	Débit mémoire	% débit requis	Calculs/seconde	% calculs requis
IBM Power 4 1300 MHz	11,1 Go/sec	58 %	120 millions	6 %
API Alpha 21264 1000 MHz	8 Go/sec	42 %	90 millions	4,5 %
Intel Itanium II 1000 MHz	6,4 Go/sec	30 %	190 millions	9,5 %

Il apparaît donc évident que l'habillage par surfaces implicites en temps réel ne peut être effectué qu'en utilisant un algorithme différent. Pour espérer obtenir du temps réel, cet algorithme devra réclamer une bande passante mémoire 3 à 4 fois plus faible et 10 à 20 fois moins de calculs, si l'on veut réaliser les calculs d'animation sur les processeurs les plus puissants disponibles.

Pour que ces animations soient calculables sur des microprocesseurs plus classiques, et avec une meilleure fluidité (50 images/seconde), l'idéal deviendrait alors de pouvoir diviser la bande passante par 20 pour la ramener en dessous d'1 Go/seconde.

Dans le cas idéal, la quantité de calcul devrait quant à elle être divisée par 100 afin qu'elle soit ramenée aux alentours de 20 millions de calculs par seconde : des performances que peuvent atteindre des processeurs courants (par exemple AIM Power PC G4 à 1GHz, Intel Pentium III 800 MHz, Mips R12000 à 400 MHz ...).

## 4. Algorithme de rendu de surfaces implicites pour le temps réel

### 4.1. Occupation mémoire : structure dynamique d'espace discret

Afin de réduire la consommation de mémoire de notre algorithme, nous avons dans un premier temps mis en place des structures permettant d'allouer et de parcourir uniquement les voxels utilisés par les simulations. En effet, un écoulement de pâte, par exemple, n'occupe qu'une part minoritaire de l'espace discrétisé : il est donc inutile de décrire l'ensemble du vide qui entoure la pâte.

Nous avons donc créé des objets C++ destinés à remplacer les tableaux à 3 dimensions dont nous nous servions jusqu'à présent.

Le principe repose sur la définition d'un objet **VoxelZone** destiné à représenter un ensemble de voxels contigus définis sur une dimension.

Un objet **VoxelPlane** permet d'ajouter une deuxième dimension en liant les différents **VoxelZone** qui appartiendraient au même plan. **VoxelPlane** offre des fonctions permettant d'ajouter

des **VoxelsZone** et de redéfinir si nécessaires ces **VoxelsZone** dans les cas où des recouvrements apparaissent.

Sur le même principe, **VoxelSpace** relie les **VoxelPlanes** entre eux. **VoxelPlane** permet alors d'allouer selon les besoins les zones précises occupées par l'écoulement de pâte.

Enfin, **VoxelSpace**, **VoxelPlane** et **VoxelZone** fournissent à l'algorithme de rendu des itérateurs sur l'espace discret qui permettent de le parcourir de façon apparemment linéaire, sans jamais passer par les zones vides.

Le surcoût imposé par la gestion de cette structure est aisément masqué par le gain de performance amené par la diminution de l'espace mémoire utilisé : effectivement, selon l'état de l'écoulement de pâte, cette structure nous permet d'occuper de 4 à 15 fois moins de mémoire qu'un tableau à 3 dimensions.

De plus, cette structure supporte la multi-résolution (on n'accède aux voxels non pas par leur indice, mais par des coordonnées symboliques) – il devient alors possible de diminuer la finesse de discrétisation sur les parties de l'objet à habiller qui ne requièrent qu'un faible niveau de détail, pour concentrer les calculs sur les parties complexes. Un tel traitement permet de diminuer de façon importante l'utilisation de la mémoire ; mais nous ne l'avons pas mis en œuvre car il est assez difficile de déterminer automatiquement qu'elles sont les zones qui peuvent se contenter d'un faible niveau de détail.

## **4.2. Quantité de calculs : réduire le nombre de calculs nécessaire en améliorant la précision des calculs effectués**

L'algorithme des « Marching Cubes » que nous avons implémenté nous impose pour chaque image, de parcourir l'espace discrétisé, et pour chacun de ses voxels appartenant à la surface implicite dont on calcule le rendu, de produire un ensemble de facettes. C'est le parcours de ce très grand espace mémoire qui constitue la principale limitation au calcul de rendu.

Pour s'approcher du rendu en temps réel, nous avons donc choisi un algorithme qui limite son recours à l'espace discrétisé :

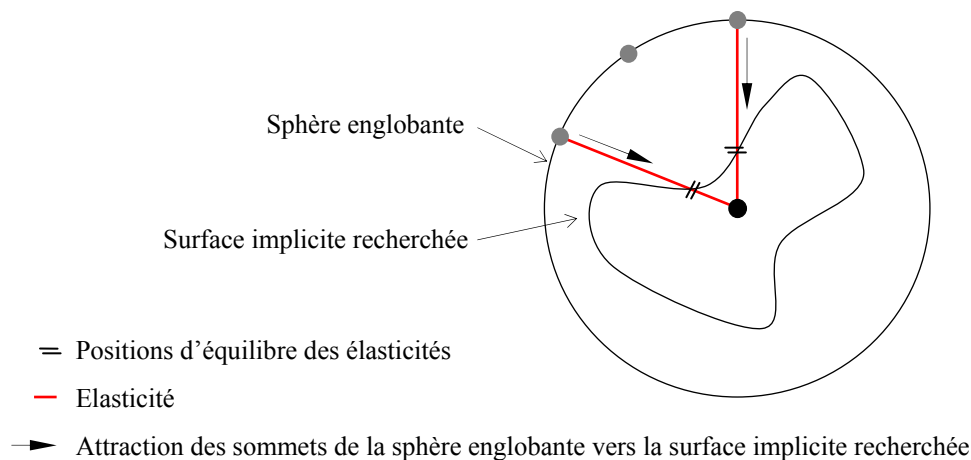
### **4.2.1 Première approximation de la surface**

Une première approximation suffit à fournir d'assez bons résultats pour produire une visualisation d'un objet implicite :

On part d'une sphère englobant l'objet implicite à visualiser. Cette sphère englobante est définie explicitement en tant que surface; c'est effectivement elle qui générera la surface finale de l'objet implicite.

On rapproche chacun des points de la sphère englobante, de la surface de l'objet implicite. La surface de l'objet implicite est définie par les positions de l'espace pour lesquelles le champ généré implicitement est égal à un certain seuil. On rapproche donc les points de la surface de la sphère jusqu'à cette position d'équilibre.

Dans cette approche, le déplacement des points se fait en utilisant des élasticités reliant les points de la surface de la sphère englobante au centre d'origine de cette dernière. Ces élasticités ont une position d'équilibre correspondant aux positions de l'espace où le champ est égal au seuil.



**Figure 82 : Première approximation de la surface**

S'il est possible d'obtenir rapidement une surface avec cette méthode, elle présente en revanche une sévère limitation : elle ne peut s'appliquer qu'aux surfaces convexes (ou proches des surfaces convexes). La sphère de départ ne peut pas en effet se déformer pour coller à n'importe quel type de formes.

Nous devons donc générer préalablement à l'étape de raffinement progressif, une surface approchée de l'objet implicite à habiller (i.e. une pâte ou un gel dans notre cas).

#### 4.2.2 Habillage précis à partir du raffinement progressif d'une surface approchée

1. Dans un premier temps, on génère une surface grossière et largement approchée de l'objet à habiller.
2. On rapproche ensuite cette surface de la surface réelle par une méthode similaire à celle utilisée dans l'approche présentée précédemment.

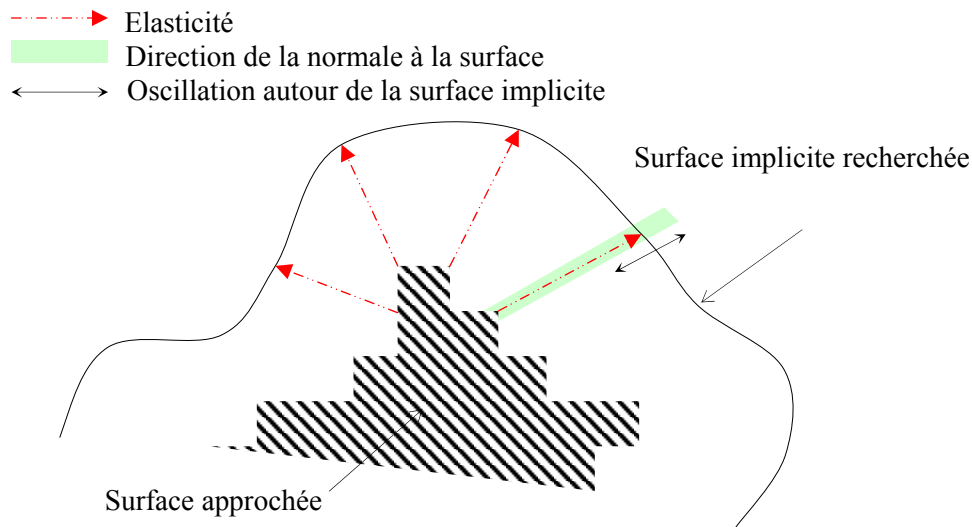
##### *Construction de la surface approchée*

1. On découpe l'espace de l'objet en voxels
2. Pour chaque voxel, on calcule la valeur du champ induit par l'objet
3. Pour chaque voxel dont le champ calculé est supérieur au seuil (=intérieur de l'objet)
4. Considérant que le voxel est un parallélépipède, on ajoute chaque face du voxel à la surface à générer si le voxel voisin correspondant à cette face à une valeur de champ inférieure au seuil (= extérieur de l'objet). On obtient ainsi une surface fermée approchée de l'objet.

##### *Affinement de la surface*

Pour chacun des points de la surface approchée que l'on vient de calculer :

1. Placer une élasticité dans la direction de la normale à la surface implicite en ce point (calculée d'après les équations implicites de l'objet). La direction de l'élasticité est fixe. Cette élasticité est attachée au point de la surface approchée, mais elle n'est pas réellement attachée à un point de l'espace à son autre extrémité. La position de cette autre extrémité est en fait une position implicite. Elle est en effet déterminée par les équations implicites de l'objet et elle varie au cours du temps (au cours du déplacement du point de la surface).
2. On laisse le système s'équilibrer sur la surface réelle de l'objet à habiller. Le point de la surface approchée se trouve alors à l'emplacement de l'autre extrémité de l'élasticité.



**Figure 83 : Affinement de la surface**

### *Ombrage de la surface*

L'ombrage de la surface ne nécessite pas de calculs supplémentaires, les valeurs des normales (nécessaires pour le calcul de l'éclairage) sont en effet connues pour chacun des points de la surface, ces normales ayant été calculées pour générer cette surface. De plus, les normales étant calculées analytiquement, on a la garantie de leur exactitude y compris pour les points suspects. Certains points induisent en effet des convergences (stabilisations) difficiles lorsque l'on utilise des opérations booléennes, notamment dans le cas de l'extrusion, entre plusieurs primitives implicites de l'objet à visualiser.

### *Coût des calculs*

En utilisant des liaisons dont la tension est directement proportionnelle à la différence entre le seuil et la valeur du champ au point d'attachement de la liaison, l'ensemble de la surface se stabilise en 5 itérations.

On peut améliorer cette vitesse de convergence en employant des fonctions de degrés plus élevés ou au travers de fonctions classiques de l'optimisation. Par exemple, la méthode d'optimisation *regula falsi* converge généralement en une seule itération dans les cas courants ; toutefois, celle-ci ne

converge plus pour des objets mettant en œuvre d'autres primitives que les points (sphères) et des fonctions booléennes.

De plus, on peut diminuer la quantité de mémoire accédée en limitant le nombre de voxels utilisés. On peut en effet subdiviser les facettes lors de leur génération afin d'obtenir autant de points de contrôle de la surface finale que nécessaire : le rapport entre nombre de voxels nécessaires et quantité de facettes étant fonction de la complexité de l'objet à habiller – le cas le plus simple est celui d'un cube qui pourra être approché par seulement 6 facettes ; un objet convexe plus complexe sera quant à lui approximable par aussi peu de voxels que pour le cube, mais nécessitera davantage de facettes (en subdivisant les faces de l'unique voxel contenant l'objet convexe).

Enfin, par rapport au « Marching Cubes », cette méthode permet de diviser de manière significative le nombre de calculs nécessaires : les points de la surface se trouvent, après équilibrage, placés exactement sur la surface implicite alors que la méthode de Lorensen et Cline n'offre qu'un placement approché : dès lors, une plus grande finesse de discrétisation est requise si on veut obtenir un effet similaire.

### **4.3. Occupation mémoire : supprimer le goulet d'étranglement en s'affranchissant de l'espace discrétisé**

Mais c'est lors du calcul du rendu de l'animation qu'il devient possible d'obtenir les meilleurs gains.

Sur le principe, l'algorithme que nous avons présenté repose sur le calcul d'une surface grossière qu'il suffit ensuite d'affiner : si on l'applique tel quel, il impose de parcourir l'espace discrétisé à chaque image et nécessite donc une importante bande passante.

Au lieu de générer à chaque image une nouvelle approximation grossière de la surface, on peut tirer parti du fait que dans une animation (et notamment dans un écoulement naturel) des images consécutives sont proches l'une de l'autre (une image peut être vue comme une déformation de l'image précédente). Ainsi, dans le cas de notre écoulement de pâte ; mais également dans les animations en général, la surface calculée des objets implicites est une excellente approximation grossière de la surface implicite de l'image suivante.

Nous pouvons donc nous affranchir quasi totalement de l'espace discrétisé en limitant son utilisation à la génération de la surface grossièrement approchée de la première image.

L'occupation mémoire est alors réduite à celle de la structure des points de contrôle de la surface rendue. Ces points de contrôles sont de même nature que les structures VERTEX que nous avons décrit (Figure 81).

Ainsi, pour une discrétisation de 250 voxels de côté, on aura en moyenne 50000 facettes générées (lors du rendu de la 1<sup>ère</sup> image), soit pour les autres images une occupation mémoire de :

$50000 * (32 \text{ bits} + 192 \text{ bits} + 3 * 32 \text{ bits}) = 1,9 \text{ Mo}$  par image (pour un modèle mémoire 32 bits)  
ou encore une bande passante de 47 Mo par seconde (25 images par seconde).

## 4.4. Vitesse de calcul : diminution de la quantité de calculs nécessaire

La méthode décrite, outre le fait qu'elle diminue la quantité de mémoire nécessaire au calcul du rendu de chaque image, permet aussi de diminuer le nombre de calculs.

Effectivement, hormis pour la première image d'une animation, il n'est plus nécessaire de calculer les valeurs de la fonction de densité pour chacun des voxels. Les seuls calculs de la fonction de densité qui sont nécessaires sont désormais ceux des points de contrôle de la surface rendue.

En début de chapitre nous avons évalué qu'en utilisant une version optimisée des « Marching Cubes » avec graphe de voisinage et une discrétisation de 250 voxels de côté, il était nécessaire de calculer la fonction de densité deux milliards de fois par seconde.

Pour une même discrétisation (250 voxels de côté, qui donnent en moyenne pour la pâte une surface comportant 50000 facettes) on aura besoin maintenant de cinq à dix millions de calculs de la fonction de densité par seconde (25 images par seconde, 50000 sommets à évaluer, 5 contributeurs par sommets et une à deux itérations de convergence par sommet).

Dès lors, cet algorithme permet de diviser le nombre de calculs nécessaires par un facteur compris entre 200 et 400.

## 5. Conclusion

S'il ne change pas la nature des calculs réalisés, l'algorithme que nous proposons permet néanmoins d'accélérer fortement les calculs de rendu et donne la possibilité d'obtenir un rendu d'animation en temps réel (à 25 images par seconde) en se concentrant sur la réduction de l'occupation mémoire.

Nous avons vu en effet que le principal facteur limitant à la vitesse de nos calculs de rendu n'était pas la puissance de calcul des processeurs, mais plutôt la bande passante de leur mémoire.

Nous étions partis, avec notre implémentation de l'algorithme de polygonalisation triangulaire de surface, de calculs de rendu qui imposaient, pour une animation en haute résolution (250 voxels de côté), une consommation de bande passante de l'ordre de 15 Giga-octet par seconde, la fonction de densité devant être calculée environ 2 milliards de fois par seconde.

Notre algorithme adapté au temps réel réduit la consommation de bande passante pendant l'animation.

	Modèle mémoire 32 bits		Modèle mémoire 64 bits	
	Fonction de densité	Bande passante requise	Fonction de densité	Bande passante requise
Marching Cubes	$2 \cdot 10^9$ calcs/sec	15 Go/sec	$2 \cdot 10^9$ calcs/sec	20 Go/sec
Algo temps réel	$7 \cdot 10^6$ calcs/sec	47 Mo/sec	$7 \cdot 10^6$ calcs/sec	62 Mo/sec
Gain	285	319	285	322

La bande passante requise par notre algorithme adapté au temps réel, n'est donc visiblement plus un facteur limitant pour la vitesse de rendu. Parmi les processeurs que nous avons évalués, les bandes passantes mémoire étaient effectivement systématiquement plus de 10 fois supérieures à la bande passante demandée.

La vitesse de calcul est donc désormais probablement limitée par les performances flottantes des processeurs considérés. En utilisant les évaluations faites pour les processeurs Intel Itanium II et IBM Power 4, et API Alpha 21264, il apparaît que ceux-ci présentent des capacités théoriques entre 5 et 10 fois supérieures à nos besoins.

En pratique, nous avons évalué notre algorithme sur un ordinateur à processeur Intel Pentium III Mobile à 600 MHz qui s'avère capable de calculer le rendu d'environ 30 images par seconde. La plupart des processeurs vendus il y a moins de 2 ans seront donc généralement capables d'assurer un rendu en temps réel.

Toutefois, l'implémentation que nous avons faite de cet algorithme est loin d'être fonctionnelle, en raison des divergences qui apparaissent au fur et à mesure de l'avance de l'animation. Généralement, les calculs de rendus sont corrects sur une vingtaine d'images consécutives ; mais en fonction de la forme de l'écoulement et surtout de sa vitesse, les points du maillage peuvent finir par se rassembler dans des zones proches, donnant à d'autres zones de l'écoulement une très faible résolution.

Pour que l'algorithme soit complet, il nous faudrait donc mettre en place des systèmes d'analyse des dynamiques d'écoulement, capables de décider à quel moment il est opportun de régénérer un maillage englobant de l'objet défini implicitement.

Enfin, il manque encore à notre algorithme des fonctions d'analyse de l'espace qui seraient capables de détecter quels sont les petits objets en mouvement susceptibles de créer des artefacts visuels d'animation. Une telle détection permettrait d'augmenter localement la définition de la discrétisation afin de supprimer ces artefacts.

## Chapitre 10 Visualisation par artefacts

---

Nous avons vu, jusqu'à présent, deux méthodes de visualisation de phénomènes naturels par construction d'un volume. Lors des recherches sur les travaux existants sur la modélisation et visualisation, nous avons constaté que pour des objets à contours flous, plusieurs études ne construisaient pas de volume mais travaillaient sur la nature de l'objet, sa relation avec la lumière, son environnement.

Habibi et al. ([habibi97]) utilisent une méthode qui s'apparente à la technique de modélisation par modèle physique appliquée à la visualisation. Les particules du modèle sont immergées dans un plan maillé de masses ponctuelles reliées par des interactions visco-élastiques. Ces dernières peuvent se mouvoir selon un seul axe, et leurs déplacements résultent de l'action du modèle sur eux-mêmes ou de la propagation d'action n'agissant pas directement sur eux. Cette différence de position est ensuite utilisée pour produire une couleur. La visualisation par écran d'épingles permet de rendre des modèles à contours flous ou au comportement turbulent. Son seul inconvénient est de ne prendre à l'heure actuelle que des modèles bidimensionnels.

La reproduction de fumées et de gaz est un enjeu important pour l'industrie du divertissement. Stam ([Stam01]) propose une méthode qui anime, en même temps qu'elle visualise, des fumées. Le modèle interagit en temps réel, et le rendu est conforme à ce que nous connaissons. Il se sert d'artefacts pour donner à ces particules l'aspect désiré. Ces artefacts sont des éléments plans en mouvement avec les particules dont les propriétés (couleur, transparence) sont établies par les paramètres de la simulation (vitesse, densité, etc.).

Au regard des simulations dont nous disposons (fluides turbulents compris), les masses ponctuelles des modèles restent sensiblement groupées et leurs voisines changent peu. Ces masses ponctuelles nous permettent de détecter et de former un contour visuellement sans même se préoccuper de la façon dont on le crée. Ce pourtour existe dans tous les cas de figure pour les modèles réalisés.

Les fluides turbulents modélisés ne sont pas visualisés car nous n'avons pas de méthode susceptible de les représenter d'une manière qui nous est familière. Une méthode telle que celle



utilisée par Stam est intéressante. L'adaptation à notre cas est susceptible de fournir des résultats satisfaisants. Nous voulons donner au flux un contour flou, qui semble se confondre avec l'environnement à mesure qu'on s'éloigne de lui. L'utilisation de la couleur comme unique paramètre de la méthode ne permettrait pas de visualiser des modèles tridimensionnels. Il faut introduire des artefacts visuels jouant sur la transparence de l'objet pour donner l'illusion d'un contour régulier.

Nous présentons une méthode de visualisation de phénomènes naturels par adaptation de la transparence d'objets graphiques élémentaires en se basant sur les données dont nous disposons : la position des particules au cours du temps. Cette méthode bien que mise en place pour les fluides turbulents servira à la visualisation d'autres phénomènes pour être validée.

## 1. Principe de base

Nous souhaitons visualiser un ensemble de particules sans structure par des éléments graphiques simples en modifiant leur taille, position et transparence. La méthode de « Construction Massique Réursive » (Chapitre 7) se basait sur une carte de densité construite autour des particules pour établir un volume. Nous avons remarqué alors que si l'on considérait les densités entourant le nuage de particules nous avons déjà une bonne approximation du volume voulu. Cette méthode donnait un volume carré aux angles abrupts, ainsi qu'une animation non fluide puisque l'on pouvait changer les paramètres au cours de la simulation. Ici nous n'allons pas chercher à construire mais simplement à visualiser de manière appropriée la carte de densité.

Cette dernière est la base de la visualisation, et doit être construite convenablement pour produire un contour satisfaisant. Pour ne pas retrouver les inconvénients de la méthode précédemment citée, nous basons nos calculs sur un unique volume englobant pour tous les instants de la simulation. La carte de densité aura les mêmes dimensions, et le même positionnement que le volume. Ainsi l'animation réalisée sera fluide, et les éléments rendus seront uniformes.

Il faut établir la manière dont nous allons calculer la carte de densité. Comme il n'y a qu'un seul volume pour toute la simulation, ce volume n'est jamais totalement rempli par les masses ponctuelles du modèle. Nous pouvons ainsi optimiser les calculs en ne s'intéressant qu'au sous-volume occupé par les particules à un instant donné.

Pour calculer la carte de densité, nous utilisons la distance reliant les particules entre elles selon un graphe de voisinage satisfaisant. Cette valeur a déjà été calculée pour la méthode de visualisation par structuration du flux (Chapitre 8).

Une fois la carte de densité conçue, nous attribuons à chaque voxel de l'espace discrétisé un cube aux dimensions en adéquation avec ce dernier et dont la transparence est le reflet de la densité qui s'y trouve. Ainsi un voxel de densité nulle ne sera pas représenté, un voxel de densité maximale sera opaque. Une telle représentation ne peut arriver, sans traitement, à représenter un espace au contour lisse avec une uniformité de la couleur.

## 2. Visualisation par artefacts visuels

Détaillons les différentes étapes de la visualisation par artefacts visuels à partir d'une simulation physique.

### 2.1. Initialisation

Dans un premier temps, nous recherchons le volume discret englobant au mieux notre nuage de particules pour tous les instants. Ce volume a une taille fixe. Cette variable définit le niveau de détail désiré. Elle influence bien évidemment le temps de calcul. Nous voulons un voxel uniforme c'est-à-dire de dimensions identiques selon les trois axes. La taille choisie par l'utilisateur est affectée à l'axe selon lequel les masses ponctuelles du modèle se déplacent le plus. Nous disposons d'un volume défini par les formules suivantes lorsque le flux s'écoule selon X (D représente la taille de discrétisation du volume) :

$$v = \frac{\Delta x}{D}$$
$$\begin{cases} X = D \\ Y = \left\lceil \frac{\Delta y}{v} \right\rceil \\ Z = \left\lceil \frac{\Delta z}{v} \right\rceil \text{ ou } Z = 1 \text{ si } \Delta z = 0 \end{cases}$$

A l'instant  $t$ , nous plaçons chacune des masses ponctuelles dans le volume englobant.

### 2.2. Calcul de la carte de densité

A partir du volume discret initialisé, nous calculons la densité pour l'ensemble des voxels contenu dans l'espace défini par les positions extrêmes des particules pour une région correspondant à la distance de voisinage dans l'espace ainsi discrétisé à l'instant  $t$ . Nous calculons en même temps la densité maximale  $D_{max}$ .

Pour calculer la densité en un point, une distance est passée en argument au programme. Ce scalaire est transformé en nombre de voxels,  $distV$ , suivant la discrétisation choisie. Ainsi la densité pour le voxel  $\langle i, j, k \rangle$  est égale à la somme du nombre de particules en  $\langle i', j', k' \rangle$  pondéré par la distance euclidienne entre  $\langle i, j, k \rangle$  et  $\langle i', j', k' \rangle$ , et cela pour la zone  $\langle i-distV, j-distV, z-distV \rangle$  à  $\langle i+distV, j+distV, z+distV \rangle$ .

Initialement, nous n'utilisons pas de distance pour calculer la densité. Une densité était juste une somme. Ainsi les voxels plus éloignés avaient autant de poids que les voxels les plus proches. Le

résultat fournissait une carte de densité aux formes carrées. Nous avons aussi essayé de nous servir d'une fonction non linéaire pour calculer la densité, mais nous avons abandonné cette technique compte tenu des temps de calcul qu'elle exigeait pour des résultats certes plus satisfaisants mais sur lesquels nous devons encore faire des traitements.

Les cartes de densités obtenues sont une répartition des influences de particules dans l'espace, et font apparaître des disparités dans le flux. Cependant, même si le niveau de discrétisation est suffisant pour qu'un voxel du volume soit assimilable à un pixel de l'écran, des carrés de densités uniformes sont mis en évidence. Il nous faut donc traiter cette carte de densités pour faire disparaître ces formes anguleuses.

### 2.3. Traitement de la carte de densité

L'élimination de forme anguleuse se fait par lissage de la carte de densité. Ce lissage a pour but d'homogénéiser les valeurs de la carte de densité sans pour autant gommer les particularités apparues lors de la construction de la carte. Ce traitement se fait par application de filtres sur la carte de densité.

Nous avons utilisé les filtres spatiaux habituellement utilisés en traitement d'images, pour faire ressortir les reliefs et éliminer des bruits. Il existe deux grandes familles de filtres : les filtres passe-bas et les passe-haut. Les filtres passe-bas lissent l'image en faisant une intégration des données, tandis que les filtres passe-haut mettent en évidence les contours par dérivation. Un filtre, ou masque de convolution est généralement symétrique et de dimension impaire.

$$\text{Fonction de convolution : } g(x, y) = \sum_{i=-\frac{Z-1}{2}}^{\frac{Z-1}{2}} \sum_{j=-\frac{Z-1}{2}}^{\frac{Z-1}{2}} f(x \oplus i, y \oplus j) \cdot \text{Filtre}(i, j)$$

$$i=0 \quad \begin{array}{|c|c|c|} \hline & j=0 & \\ \hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline \end{array}$$

**Figure 84 : Exemple de filtre de taille Z=3 (ici filtre moyenneur)**

Les filtres passe-bas éliminent le bruit en sommant les valeurs de pixels qui les entourent puis en divisant cette valeur par la somme des poids. Le filtre moyenneur associe à chaque portion du masque la même valeur, traitant ainsi chaque donnée de la même manière. Comme le filtre gaussien, il introduit un flou. Plus le masque est grand, plus le lissage devient important et plus le flou s'accroît.

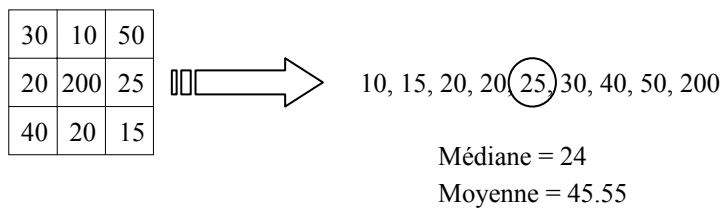
Il existe plusieurs masques pour le filtre gaussien, nous en utilisons deux suivant l'effet voulu : le premier de taille 3x3, le second de taille 5x5. Ces deux filtres favorisent plus ou moins les particules en bordure de masque.

1	2	1
2	4	2
1	2	1

0.0	0.6	1.7	0.6	0.0
0.6	13	36	13	0.6
1.7	36	100	36	1.7
0.6	13	36	13	0.6
0.0	0.6	1.7	0.6	0.0

**Figure 85 : Exemples de filtres gaussiens**

Le filtre médian est un filtre passe-bas non-linéaire. Il s'agit de prendre la médiane des valeurs de la zone traitée. Le filtre médian est utile pour éliminer les bruits poivre et sel.



**Figure 86 : Filtre médian**

Les filtres passe-bas génèrent du flou très visible sur les contours. Pour y remédier, on y applique un filtre sélectif adapté à la région à traiter. Il tient compte de la pente du contour en cet endroit. Le filtre est ainsi construit :

$$\begin{cases}
 F(i, j) = 0.5 \\
 F(k, l) = \frac{1}{\sum_{I, J} F(I, J)} \\
 d(i, j, k, l) = |Densité(i, j) - Densité(k, l)|
 \end{cases}$$

Tous ces traitements lissent bien les données, mais lorsqu'on les visualise, l'intervalle des valeurs est trop important pour que le rendu ne fasse pas apparaître des paliers malgré l'homogénéisation de la carte de densité.

Nous ré-échantillons ces données en leur appliquant une fonction non-linéaire qui amène la plus forte densité à 1.0 et la plus faible à 0. Partant de ce principe, nous en avons utilisé une, similaire, qui nous permet d'extraire les contours (une densité nulle se ré-échantillonne à 1.0, et une densité forte à 0.0).

Nous avons traité la carte de densité avec d'autres filtres (filtre multiplicatif, passe-haut, additionnel, différence de données), mais ils n'ont pas été retenus pour le traitement final des images.

Pour les modèles 2D nous appliquons un filtre gaussien 3x3, un ré-échantillonnage « interne » (de 0.0 à 1.0), un filtre gaussien 3x3, un filtre médian et un filtre sélectif pour diminuer les effets des filtres gaussiens. Pour extraire un contour, nous appliquons un ré-échantillonnage « externe » (de 1.0 à

0.0), un filtre gaussien, de moyenne, médian et enfin un filtre sélectif. Le détail des étapes du passage des filtres est présenté à la Figure 87.

Pour les modèles en 3D, il nous a fallu mettre en place des filtres n'agissant pas uniquement dans le plan [X, Y], mais aussi dans le plan [X, Z]. Ce traitement dans le plan [X, Z] s'est avéré utile pour mieux uniformiser la surface. Il n'est cependant pas obligatoire compte tenu de la façon dont nous visualisons les modèles 3D, à savoir par un ensemble de facettes dans le plan [X, Y].

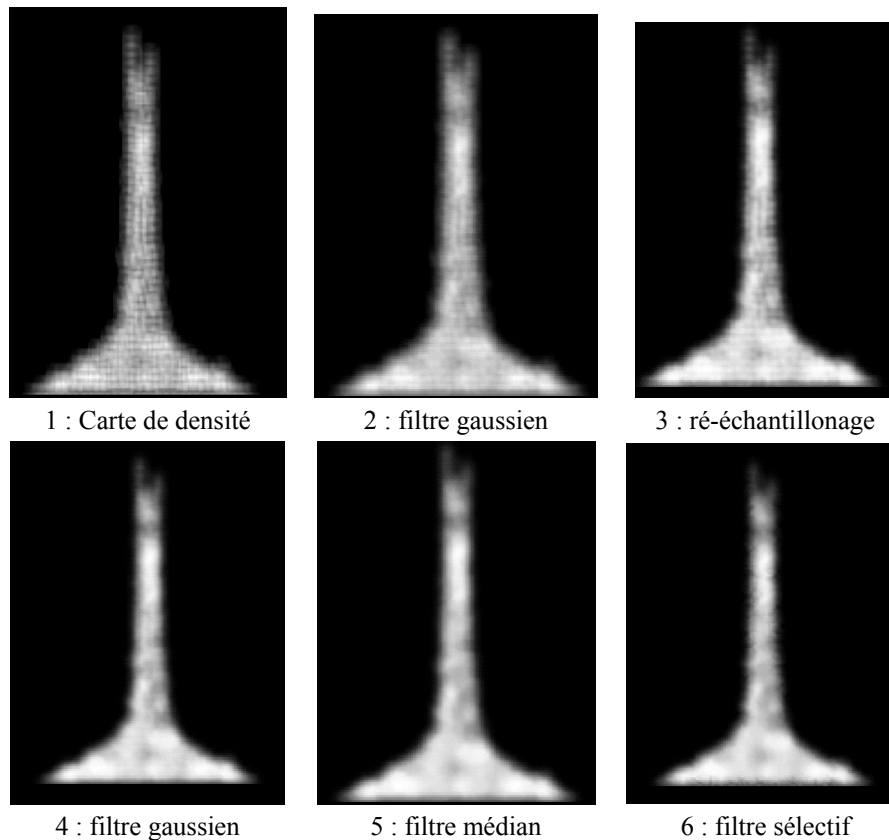


Figure 87 : traitement de la carte de densité

## 2.4. Visualisation du volume discret

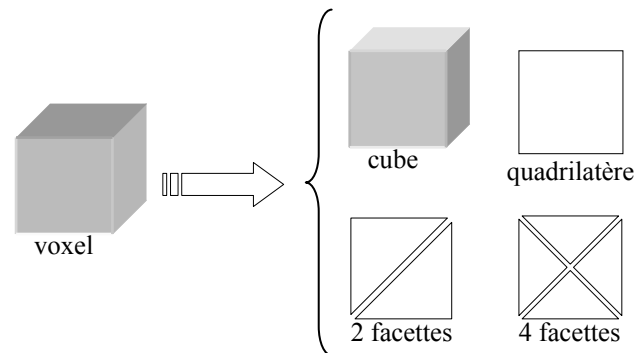
Il ne reste plus qu'à construire la visualisation en conséquence. Pour chaque voxel de densité  $d$  (non nulle), nous construisons un cube de taille  $\langle v, v, v \rangle$  de transparence  $v$ , où  $v = \frac{d}{D_{\max} - D_{\min}}$ .

Après l'application d'un ensemble de filtres sur la carte de densité calculée, nous obtenons un effet qui évoque l'apparence visuelle de la fumée.

Le problème de cette visualisation est le nombre de cubes utilisés. En effet chaque cube a 6 facettes, et même si on ne représente que 30% des voxels du volume englobant cela fait 112 500 cubes soit 675 000 facettes si le volume a pour taille  $100*75*50$ . Nous avons essayé de diminuer ce nombre de facettes lourdes à gérer par nos ordinateurs. Au lieu de remplacer chaque voxel par un cube nous

allons le remplacer par deux ou quatre facettes. Nous aurons alors 225 000 ou 450 000 facettes suivant les cas, ce qui serait un gain non négligeable.

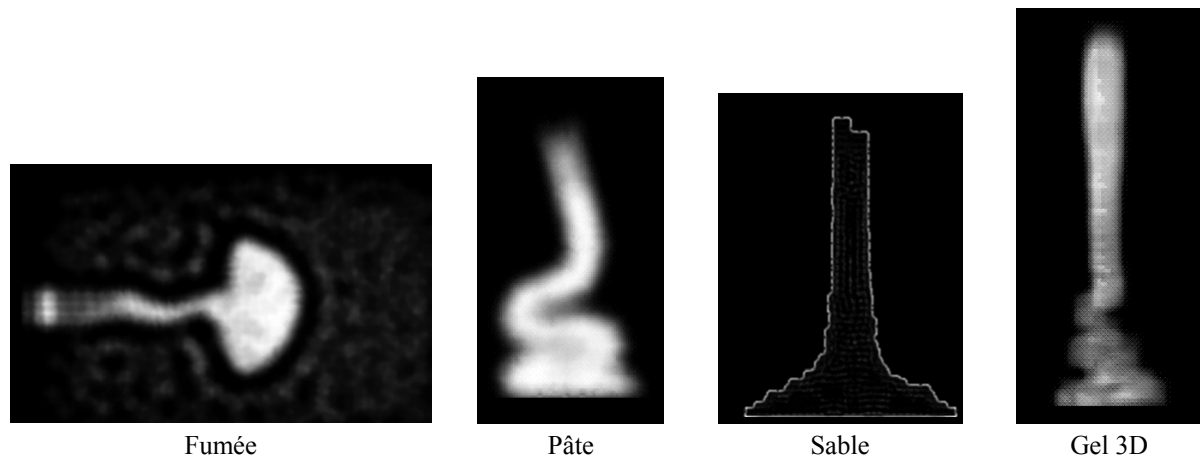
Pour contrer le besoin d'une discrétisation importante, nous utilisons une visualisation par facettes planes dont nous recalculons la densité en chacun des sommets en prenant en compte



**Figure 88 : Décomposition d'un voxel en élément graphique**

uniquement les voisins directs sur la même tranche (même z) que le voxel examiné (soit quatre voisins). La densité est alors la somme des densités des voisins directs divisées par quatre.

Les contours que nous obtenons sont tels que nous les imaginions en observant le film fait de masses ponctuelles.



**Figure 89 : Quelques exemples de visualisation par artefacts**

### 3. Conclusion

Cette technique est simple d'utilisation, d'implémentation et, si le taux de discrétisation n'est pas exorbitant, assez rapide. La densité d'un espace caractérise cet espace en sous-espace plus ou moins dense. Si la densité est concentrée alors elle représente une forme, un contour. Ce que nous visualisons ici est assimilable à une isosurface de seuil 0 où la fonction est remplacée par la densité qui,

si on peut dire, apporte en quelque sorte le même type d'information dans la mesure où elle caractérise un objet représenté par un ensemble de particules.

Nous cherchions à reconnaître des phénomènes à partir de caractéristiques dynamiques sans que la visualisation n'influence notre jugement. Comme la méthode se sert uniquement de positions de particules et d'une distance qui caractérise le flux, nous avons répondu à nos attentes. Cependant le recalcul de densité en chaque sommet de voxel n'est peut-être pas approprié, mais cette technique nous a semblé être une manière simple de procéder à moindre coût. Cependant cette phase est inutile lorsqu'un voxel du volume discret est assimilable à un pixel de l'écran.

La méthode proposée ici joue sur la perception et l'interprétation d'une succession d'objets plus ou moins opaques. Si la définition du volume englobant est suffisamment fine, le contour est parfaitement visible même si il est trouble, flou. Cette méthode a aussi l'avantage de donner l'illusion d'un volume.

Pour les modèles 3D, nous devons spécifier une normale pour que les facettes ne soient pas toutes éclairées de la même façon et donc masquer les caractéristiques des « tranches » de facettes. La normale est dirigée vers l'utilisateur (axes Z) et à l'opposé de la densité maximale. Ainsi nous dirigeons la normale vers l'extérieur du volume.

**Partie IV - CONCLUSION ET  
REFERENCES BIBLIOGRAPHIQUES**





# Chapitre 11 Conclusion

---

Cette thèse porte sur la modélisation et la visualisation de phénomènes naturels. Un phénomène naturel est une dynamique, plus ou moins complexe, aux formes diverses et à topologie variable, qui déplace de la matière. Cette animation est induite soit par une intervention extérieure, soit par un état interne qui nécessite une réorganisation de matière (un écoulement par exemple).

Notre cerveau est capable de reconnaître un phénomène naturel reproduit pas un ordinateur si celui-ci fait preuve de réalisme dynamique. Nous l'identifions par ses dynamiques, ses caractéristiques de formes. Elles nous permettent de dire que la matière en mouvement est visqueuse ou élastique uniquement à partir de spécificités perceptibles sur une forme, un mouvement. Nous sommes capables de caractériser par un simple regard les dynamiques : notre cerveau est un outil d'analyse précis et performant suite aux différents apprentissages auxquels il a été confronté.

## 1. Bilan

La reproduction par ordinateur de phénomènes naturels est un vaste sujet. Les méthodes de modélisation ne sont pas toujours basées sur la physique du phénomène à simuler. Ainsi plusieurs recherches ont démontré qu'à l'aide de méthode physique, il est possible de réaliser un modéleur unique permettant de reproduire l'ensemble des phénomènes naturels, des objets physiques. Le laboratoire ICA et l'ACROE ont développé un système générique de simulation par physique particulière pour modèles représentés à un niveau mésoscopique. Ce modéleur-simulateur s'appelle CORDIS-ANIMA et représente un objet est comme un réseau de masses ponctuelles interconnectées qui évoluent les unes par rapport aux autres selon des lois physiques.

Durant cette thèse, nous avons utilisé CORDIS-ANIMA pour concevoir des phénomènes naturels complexes et variés en spécifiant peu de paramètres (quatre paramètres pour représenter les interactions - viscosité, élasticité, seuil de la viscosité, seuil de l'élasticité -). En effet, nous avons réalisé des modèles de pâte, de sable, de gel (2D et 3D), de fluide turbulent ainsi que de croissance de

cellules végétales (2D). Cependant la matière du phénomène naturel simulé ne peut se déplacer sans un environnement physique adéquat, source de mouvement.

Ces phénomènes avaient déjà été pour la plupart modélisés en 2D. Nous avons remanié ces travaux pour valider les modèles 2D existants, pour créer une homogénéité entre modèles, pour obtenir une connaissance en matière de modélisation de dynamiques complexes (savoir gérer les paramètres, l'environnement physique du modèle), pour pouvoir mettre au point des modèles 3D robustes et fiables sans avoir à réviser les champs paramétriques et physiques du modèle.

Rien ne garantissait que les principes de modélisation des modèles bidimensionnels resteraient valides pour des modèles tridimensionnels. Avant cette thèse, il n'existait aucun modèle 3D puisque aucune méthode de visualisation n'existait ; l'évaluation du phénomène simulé dépendait fortement de son observabilité. De plus, nous n'étions pas sûrs que le nombre de points nécessaires pour obtenir de bons effets dynamiques en 3D ne serait pas prohibitif durant la phase de conception (pour des questions de temps et coût de calcul. De plus en 2D le nombre de points nécessaires s'est avéré bien plus faible que ce que nous pensions a priori. Les modèles 3D nous ont permis de valider nos modèles 2D. En effet, en 3D les phénomènes présentaient les mêmes caractéristiques dynamiques qu'en 2D sans être obligé d'augmenter de manière importante le nombre de masses ponctuelles du modèle de matière (le modèle de gel 3D contient autant de masses ponctuelles que le modèle de pâte 2D, à savoir 300). En outre, ces modèles 3D nous ont permis de confirmer un des résultats essentiels obtenus dans nos modèles 2D : les caractéristiques environnementales sont indispensables à la formation de certaines dynamiques (un empilement ne peut avoir lieu que sur un sol rugueux). Toutefois certains modèles 2D n'ont pas été reproduits à une dimension supérieure. C'est le cas des modèles de fluides turbulents : nous n'avons pas disposé d'assez de temps pour définir les conditions minimales d'apparition de la turbulence en 3D. Néanmoins les autres modèles 3D ont apporté entière satisfaction.

Toutes ces recherches sur la modélisation de phénomènes naturels ont validé notre démarche de simulation, ont prouvé la généralité du modéleur-simulateur physique particulière CORDIS-ANIMA. L'aboutissement de ces travaux a montré l'inutilité de la conception de modèles comportant un trop grand nombre de masses ponctuelles : même avec peu d'entre elles, les dynamiques pertinentes se produisent.

La phase de modélisation effectuée, il nous fallait visualiser ces phénomènes naturels. Une simulation se présente sous la forme d'un ensemble de particules (appelées précédemment masses ponctuelles) pour lesquelles nous connaissons leurs coordonnées pour chaque instant de la simulation. Disposer uniquement de ces points pour visualiser les phénomènes est problématique. Cela est *a priori* peu pour construire une visualisation qui permette de discerner aussi bien des formes, que des dynamiques du phénomène reproduit. Nous avons cherché à élaborer un volume qui englobe au mieux ces particules dispersées dans l'espace sans structure prédéfinie. Pour n'omettre aucun axe de prospection, il a été nécessaire d'avoir à l'esprit un ensemble de solutions envisageables en procédant à un travail de réflexion et de recherche sur les méthodes existantes. Nous voulions avoir un regard exhaustif de ce qui existait en matière de simulation-visualisation par ordinateur.

Au cours de ce travail de prospection, nous n'avons trouvé dans la littérature aucune méthode de visualisation à partir de données telles que les nôtres qui ne contiennent aucune information de positionnements les unes par rapport aux autres, et sans savoir où se situe la surface. La plupart des

méthodes de modélisation traitent en parallèle simulation et rendu, ce qui n'est pas notre cas. Elles ont donc à leur disposition des données physiques sur lesquelles baser la visualisation. La connaissance de telles méthodes de modélisation et de visualisation nous a permis de trouver des axes de recherches susceptibles de répondre à notre problématique et d'en éliminer d'autres.

Lorsque nous observons un ensemble de masses ponctuelles, nous déterminons une surface par un regroupement implicite des particules les plus proches. Notre première méthode cherche à reproduire notre démarche de construction visuelle d'une forme en créant un volume dans un espace discret par ajout de morceaux de matière visuelle. L'expansion se fait sur des notions de proximité et de densité. Cette méthode induit une certaine liberté dans les règles de construction, mais son utilisation dans un processus d'animation nous a montré qu'il est difficile d'obtenir un volume satisfaisant, régulier au cours du temps. De plus certains phénomènes naturels au caractère élancé présentant une faible densité ne peuvent être visualisés par cette méthode.

Définir une relation de voisinage plus forte et plus précise nous a semblé être une deuxième solution. L'idée est de relier les masses ponctuelles d'un modèle en établissant un réseau de connexions, dont le nombre nécessaire et suffisant, suffit à caractériser ce modèle. Etablir cet ensemble est un moyen d'attribuer à chaque particule un encombrement spatial qu'elle possédait lors de la simulation. La distance utilisée pour construire ce graphe est choisie de façon adéquate de manière à caractériser le volume occupé par le nuage de particules. Lors de l'étude des paramètres nous avons constaté qu'il était possible d'identifier de manière assez précise les particules en bordure de surface de celles à l'intérieur, tout du moins en 2D.

Nous avons donc décidé de réaliser une interprétation de ce graphe à l'aide de surfaces implicites. La définition de ces surfaces présente certains avantages. L'association à chaque particule d'une sphère implicite symbolise l'espace occupé par les connexions la reliant à ses voisines dans le graphe de voisinage. Les interstices dus à une répartition non uniforme des particules dans la scène sont comblés par la réunion de plusieurs portions de sphères lorsque la valeur de la surface sur cette zone est inférieure à l'isosurface choisie. Trouver les valeurs des paramètres des primitives graphiques (squelettes) n'a pas été simple. Cela s'est effectué par études successives sur des critères subjectifs (surface obtenue), et objectifs (valeur du rayon utilisé, distance entre particules, direction du vecteur voisin moyen). Les résultats obtenus ont été visualisés à l'aide d'un logiciel de lancer de rayons du commerce. Il nous a permis d'appliquer un ensemble de texture sur la surface pour se rapprocher de l'aspect du phénomène tel que nous le connaissons ou laisser libre cours à notre imagination.

Les fluides turbulents n'avaient pas de méthode mettant en avant leurs volutes, leurs caractéristiques visuelles. Nous nous sommes inspirés des travaux de Stam et al. pour la visualisation de phénomènes gazeux. Chaque portion de matière est rendue par une facette dont la transparence traduit la densité de la zone qu'elle représente.

Nous avons conçu un modèle de croissance de cellules végétales. Pour ce type de phénomène, nous avons opté pour une visualisation que nous avons qualifiée de « non photo réaliste ». La visualisation est paramétrable pour rendre la cellule non conforme à sa représentation réelle. Cependant elle doit rester identifiable par quiconque la regarde, soit par sa forme, soit par ses composants internes (méthode présentée Annexe A).

## 2. Perspectives

Les simulations obtenues sont satisfaisantes du point de vue dynamique de leurs dynamiques visuelles. Nos animations sont reconnaissables, cependant il est difficile de fixer un ensemble de paramètres interdépendants pour produire des dynamiques complexes. Il serait donc intéressant d'inférer le modèle physique à partir de données issues du monde réel. Notamment peut-on à partir des données phénoménologiques telles que la vitesse et l'accélération trouver les paramètres physiques d'un modèle de matière ainsi que son environnement.

Les résultats des simulations et visualisations sont globalement satisfaisants mais les temps de calcul peuvent devenir rédhibitoires à mesure que le nombre de particules du modèle physique sous-jacent augmente. Une parallélisation des algorithmes permettrait de diminuer fortement les temps de conception.

La méthode de « Construction Massique Réursive » n'est pas adaptée à des modèles pour lesquels les flux sont fins ou les formes complexes, mais une approche incluant une subdivision de l'espace par « octree » avant toute phase de construction et une ré-étude des critères de construction pourrait régler les problèmes que nous avons rencontrés. La méthode de visualisation par artefacts pourrait être améliorée si un voxel de l'espace discrétisé correspondait à un pixel de l'écran. Cela pourrait être un avantage pour l'obtention d'une surface lisse.

Il serait intéressant d'analyser avec précision la possibilité d'intégrer dans la visualisation les données de vitesse et d'accélération issues de calculs post-modélisation. Comme cela a été expliqué au Chapitre 6, ce sont des données phénoménologiques, des indications de structures internes à la plupart des modèles. On pourrait s'en servir pour améliorer la méthode optimisée de visualisation de surfaces implicites. Un brusque changement des vitesses et des accélérations pourrait correspondre à une indication pour recalculer une surface approchée, et ainsi ne plus avoir de discontinuité temporelle dans l'animation.

Explorer le domaine de la visualisation par génération automatique d'un volume par algorithme génétique permettrait une approche différente de la problématique. Ce principe est une manière de reproduire les actions de notre cerveau lors de l'analyse. En effet il détermine par un simple regard sur un nuage de points un volume englobant, *i.e.* un contour. Il faudrait procéder à une étude sur la perception visuelle, mettant en avant les mécanismes d'association de forme à un nuage de points. On pourrait ainsi en ressortir des règles de base, source de l'algorithme génétique, qui par apprentissage et analyse qualitative des résultats s'étofferaient.

## 3. Conclusion

CORDIS-ANIMA est un système de modélisation et de simulation numérique d'objets physiques visibles, audibles et manipulables. Il permet par la gestion de quelques paramètres de

réaliser toutes sortes de phénomènes sans connaître la physique sous-jacente au phénomène. La visualisation de ces modèles n'est pas simple mais cela permet de laisser libre cours à son imagination et de faire évoluer toutes sortes de formes dans une harmonie de couleur et de mouvement.



## Chapitre 12 Bibliographie

---

- [adzhiev99] V. Adzhiev, R. Cartwright, E. Fausett, A. Ossipov, A. Pasko, V. Savchenko. **HyperFun project: a framework for collaborative multidimensional F-rep modelling**. *Eurographics '99, Workshop on Implicit Surfaces*. 1999. Bordeaux, France.
- [artzy81] E. Artzy, G. Frieder, G.T. Herman. **The theory, design implementation and evaluation of three-dimensional surface detection algorithm**. *Computer Graphics and Image Processing*. 15(1):1-24. 1981.
- [aurenhammer91] F.Aurenhammer. **Voronoi diagrams: a survey of fundamental geometric data structure**. *ACM Computing Surveys*. 33(3):345-405. 1991.
- [barrero00] D. Barrero, M. Paulin, R. Caubet. **Simulating turbulent combustion**. *Proceedings of GraphiCon*, 10:181-188. August 2000. Moscow, Russia.
- [bentley80] L. Bentley. **Multidimensional divide and conquer**. *Communications of ACM*. 23(4):214-229. 1980.
- [berger90] M.O. Berger. **Snake growing**. *Rapport de recherche CRIN 90-R-071*, LORIA, Nancy. 1990.
- [bernardini99] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin. **The ball-pivoting algorithm for surface reconstruction**. *RC 21463, IBM Research Center - Yorktown Heights*. April 1999.
- [bertin94] E. Bertin. **Diagramme de voronoï 2D et 3D : application en analyse d'images**. *Thèse de l'université Joseph Fourier – Grenoble 1, laboratoire TIMC*. Janvier 1994.



- [bittar95] E. Bittar, N.Tsingos, M.P. Gascuel. **Automatic reconstruction of unstrutred 3D data: combining a medial axis and implicit surfaces.** *Eurographics'95*. September 1995.
- [bittar98] E. Bittar. **Modèles déformables, surfaciques, implicites et volumiques, pour l'imagerie médicale.** *Thèse de l'Institut National Polytechnique de Grenoble, laboratoire TIMC*. 1998.
- [blinn82] J. Blinn. **A generalization of algebraic surfaces drawing.** *ACM Transactions on Graphics*. pp235-256. July 1982.
- [cadoz90] C. Cadoz, A. Luciani, J.L. Florens. **CORDIS-ANIMA : système de modélisation et de simulation d'instruments et d'objets physiques pour la création musicale et l'image animée.** *Article présenté au colloque « Modèle Physique, Création Musicale et Ordinateur »*. pp585-628. 1990. Grenoble.
- [cadoz93] C. Cadoz, A. Luciani, J.L. Florens. **CORDIS-ANIMA a modelling and simulation system for sound and image synthesis - The general formalism.** *Computer Music Journal*. 17(4), 1993.
- [chiba95] N. Chiba, S. Sanakanishi, K. Yokoyama, I. Ootawara. **Visual simulation of the water currents using a particle based behavioural model.** *The Journal of Visualization and Computer Animation*. 6:155-171. 1995.
- [dadalto97] L. Da Dalto. **Modèles pour la simulation de phénomènes naturels en image de synthèse.** *Thèse de l'université de Toulouse 3*, 1997. collection : ANRT - Grenoble.
- [debunne99] G. Debunne, M. Desbrun, M.P. Cani. **Interactive multiresolution animation of deformable models.** *Eurographics'99. Workshop on Computer Animation and Simulation*. September 1999. Milano, Italia.
- [debunne00] G. Debunne, M. Desbrun, M. P. Cani, A. Barr. **Adaptative simulation of soft bodies in real-Time.** *Computer Animation 2000*. May 2000. Philadelphia.
- [desbrun95] M. Desbrun M.P. Gascuel. **Animating soft substances with implicit surfaces.** *Computer Graphics, Proceedings of SIGGRAPH'95*. pp287-290. 1995.
- [desbrun96] M. Desbrun M.P. Gascuel. **Smoothed particles: A new paradigm for animating highly deformable bodies.** *Eurographics'96, Computer Animation and Simulatio*. pp61-76. September 1996. Poitiers, France.
- [desbrun97] M. Desbrun. **Modélisation et animation de matériaux déformables en synthèse d'images.** *Thèse de l'institut National Polytechnique de Grenoble, laboratoire GRAVIR/IMAGIS*. Décembre 1997.

- [dobashi00] Y. Dobashi, K. Kaneda, H. Yamshita, T. Okita, T. Nishita. **A simple, efficient method for realistic animation of clouds.** *Computer Graphics, Proceedings of SIGGRAPH'00*. pp19-28. August 2000.
- [falzon92] F. Falzon, G. Giraudon, M. Berthod. **Scale-space 3D surface representation: a model study.** *Rapport de recherche n°1639 – Programme 4 Robotique, Image et Vision. INRIA Sophia Antipolis*. Janvier 1992
- [favardin93] C. Favardin. **Détermination automatique de structures géométriques destinées à la reconstruction de courbes et de surfaces à partir de données ponctuelles.** *Thèse de l'Université Paul Sabatier à Toulouse*. 1993.
- [fearing00] P. Fearing. **Computer modelling of fallen snow.** *Computer Graphics, Proceedings of SIGGRAPH'00*. pp37-46. August 2000.
- [fedkiw01] R. Fedkiw, J. Stam, H. W. Jensen. **Visual simulation of smoke.** *Computer Graphics, Proceedings of SIGGRAPH'01*. pp 23-30. August 2001.
- [ferley96] E. Ferley, M.P. Gascuel, D. Attali. **Skeletal reconstruction of branching shapes.** *Second International Workshop on Implicit surfaces - Implicit Surfaces'96*. October 1996.
- [flickner94] M. Flickner, H. Sawhney, D. Pryor, J. Lotspiech. **Intelligent interactive image outlying using spline snakes.** *RJ 9918, IBM Research Division, Almaden Research Center*. November 1994.
- [foster96] N. Foster D. Metaxas. **Realistic animation of liquids.** *Graphical Models and Image Processing*. 58(5):471-483. 1996.
- [foster97] N. Foster D. Metaxas. **Modeling the motion of a hot turbulent gas.** *Computer Graphics, Proceedings of SIGGRAPH'97*. pp181-188. 1997.
- [foster01] N. Foster, R. Fedkiw. **Practical animation of liquids.** *Computer Graphics, Proceedings of SIGGRAPH'2001*. pp15-22. August 2001.
- [france95] L. France. **Modèles de morphogenèse : croissance et fractures végétales par modèle physique.** *Rapport de stage de DEA Signal, Image, Parole. ACROE*. Juin 1995.
- [galin97] E. Galin. **Métamorphose et visualisation de blobs à squelettes.** *Thèse à l'université de Lyon 1*. 1997. collection : ANRT - Grenoble.
- [gareau97] A. Gareau. **Utilisation des systèmes de particules pour la simulation de phénomènes naturels. Présentation d'une architecture permettant l'intégration de systèmes animés hétérogènes.** *Thèse à l'université de Lyon 1*. 1997. collection : ANRT - Grenoble.

- [gascuel\_jd98] J.D. Gascuel, M.P. Cani-Gascuel, M. Desbrun. **Simulating landslides for natural disaster prevention.** *Eurographics '98, Workshop on Computer Animation and Simulation.* September 1998.
- [gascuel\_mp88] M.P. Gascuel-Cani. **Recollement et pincement de surfaces splines : Nouvelles méthodes pour la création interactive d'objets complexes et l'habillage automatique de squelettes.** *Rapport de recherche LIENS-88-12, ENS (France).* 1988.
- [gascuel\_mp97] M.P. Gascuel M. Desbrun. **Deformable models using implicit surfaces.** *IEEE Transaction on Visualization and Computer Graphics.* 3(1). March 1997.
- [guo97] B. Guo, J.P. Menon, B. Willette. **Surface reconstruction using alpha shapes.** *RC 20689, IBM Research Center - Yorktown Heights.* January 1997.
- [guilbaud00] C. Guilbaud, A. Luciani, A. Leclerc. **Volume construction from moving unorganised points.** *Proceedings of GraphiCon.* 10:276-281. August 2000. Moscow, Russia.
- [greenspan73] D. Greenspan. **Discrete Models.** Editor Addison-Wesley. reading in applied mathematics. 1973.
- [habibi97] A. Habibi. **Visualisation d'objets très déformables : relation mouvement – forme – image.** *Thèse de l'Institut National Polytechnique de Grenoble. Laboratoire ICA.* Janvier 1997.
- [hitschfeld97] N. Hitschfeld. **Generalization of modified octrees for geometric modelling.** *Proceedings of the Conference on Theory and Practice of Geometric Modelling.* pp260-272. 1997.
- [hoppe94] Hugues Hoppe. **Surface reconstruction from unorganized points.** *PhD thesis, University of Washington (USA).* 1994.
- [huriet97] C. Huriet. **Les enjeux techniques et éthiques des images de synthèse.** *Rapport du Sénat 169, Office Parlementaire d'évaluation des choix scientifiques et technologiques.* Octobre 1997.
- [jimenez93] S. Jimenez. **Modélisation et simulation physique d'objets volumiques déformables complexes.** *Thèse de l'Institut National Polytechnique de Grenoble, laboratoire ICA.* 1993.
- [joe91] B. Joe. **Construction of three-dimensional Delaunay triangulations using local transformations.** *Computer Aided Design.* 8: 2, pp123-142. May 1991.

- [juneidi97] Z. Juneidi. **Du sable au fluide : recherche d'un modèle rendant compte des états intermédiaires de la matière - Modèle physique particulière générique de pâtes.** *Rapport de stage de DEA IVR, ACROE.* Juin 1997.
- [kass90] M. Kass G. Miller. **Rapid stable fluid dynamics for computer graphics.** *Computer Graphics, Proceedings of SIGGRAPH'90.* 24(4):49-57. 1990.
- [lachaud98] J.O. Lachaud. **Extraction de surfaces à partir d'images tridimensionnelles : Approche discrète et approche par modèle déformable.** *Thèse de l'institut National Polytechnique de Grenoble, laboratoire TIMC.* 1998.
- [logan94] L.P. Logan, D.P.M. Wills, N.J. Avis. **Deformable objects in virtual environments.** *2nd UK VR-SIG Conference.* Silicon Graphics Reality Center, Theale (UK). December 1994.
- [lorensen87] W.E. Lorensen, H.E. Cline. **Marching Cubes: a high resolution 3D surface construction algorithm.** *Computer Graphics, Proceedings of SIGGRAPH'87.* 21(4):163-169. July 1987.
- [luciani86] A Luciani C. Cadoz. **Utilisation de modèles mécaniques et géométriques pour la synthèse sonore et le contrôle d'images animées.** *Deuxième colloque image - CESTA,* 1986. Nice France.
- [luciani89] A. Luciani. **Modèles physiques pour l'animation.** *Rapport interne de Recherche* 89(14), ACROE. 1989.
- [luciani91] A. Luciani, S. Jimenez, J.L. Florens, C. Cadoz, O. Raoult. **Computational physics: a modeller simulator for animated physical objects.** *Eurographics'91, Computer Animation and Simulation.* September 1991. Vienna, Austria.
- [luciani91b] A. Luciani, S. Jimenez, O. Raoult, C. Cadoz, J.L. Florens. **An unified view of multiple behaviour, flexibility, plasticity and fractures: balls, bubbles and agglomerates.** *Modeling in Computer Graphics, SpringerVerlag Ed.* pp54-74. 1991.
- [luciani95] A. Luciani, A. Habibi, E. Manzotti. **A multi-scale physical model of granular materials.** *Proceedings of Graphics Interface'95.* pp136-146. Mai 1995. Québec, Canada.
- [luciani95b] A. Luciani, A. Habibi, A. Vapillon, Y. Duroc. **A physical model of turbulent fluids.** *6th Eurographics Workshop on Animation and Simulation, Springer Verlag Ed.* pp16-29. 1995. Maastricht (The Netherlands).

- [luciani00] A. Luciani. **From granular avalanches to fluid turbulences through oozing pastes - A mesoscopic physically-based particle model.** *Proceedings of GraphiCon.* (10):282-289. August 2000. Moscow, Russia.
- [luding96] S. Luding, J. Duran, E. Clément, J. Rajbenbach. **Simulations of dense granular flow: dynamic arches and spin organisation.** *Journal of Physics.* (6):823-836. 1996.
- [mcinerney95] T. McInerney D. Terzopoulos. **Topologically adaptable snakes.** *5th International Conference on Computer Vision - ICCV'95.* June 1995. Cambridge, MA (USA).
- [manzotti95] E. Manzotti. **Modèle physique pour la modélisation de fluides en régime turbulent.** *Rapport de stage de DEA Signal, Image, Parole.* ACROE. Juin 1995.
- [menon96] J. Menon. **Introduction to implicit techniques.** *Courses of SIGGRAPH'96.*
- [miller89] G. Miller A. Pearce. **Globular Dynamics: a connected particle system for animating viscous fluids.** *Computers and Graphics,* 13(3):305-309. 1989. also in SIGGRAPH'89 Course notes number 30.
- [miller91] J.V. Miller, D.E. Breen, W.E. Lorensen, R.M. O'Bara, M.J. Wozny. **Geometrically deformed models: A method for extracting closed geometric models from volume data.** *Computer Graphics, Proceedings of SIGGRAPH'91.* 25(4):217-226. July 1991.
- [monaghan92] J.J. Monaghan. **Smooth Particles Hydrodynamics.** *Annual revue of Astronomic Astrophysic.* 30:543. 1992.
- [murakami87] S. Murakami, H. Ichibara. **On a 3D display method by metaball technique.** *Journal of papers given by at the Electronics Communication.* Volume 570-D, n°8, pp 1607-1615. 1987. en japonais.
- [muraki91] S. Muraki. **Volumetric shape description of range data using blobby model.** *Computer Graphics, Proceedings of SIGGRAPH'9.* 25(4):227-235. July 1991.
- [nishimura85] H. Nishimura, M. Hirai, T. Kawai, T. Kawaka, I. Shirakawa, K. Omura. **Object modelling by distribution function and a method of image generation.** *Journal of papers given by at the Electronics Communication.* Volume 568-D, n°8, pp 718-725. 1985. en japonais.
- [nishita97] T. Nishita, H. Iwasaki, Y. Dobashi, E. Nakamae. **A modelling and rendering method for snow using meatballs.** *Eurographics'97.* 16(3). 1997.

- [nullans98] S. Nullans. **Reconstruction géométrique de formes – Application à la géologie**. *Thèse de l'université de Sophia-Antipolis préparée à l'INRIA de Sophia-Antipolis*. Décembre 1998.
- [obrien99] J. F. O'Brien, J. K. Hodgins. **Graphical modeling and animation of brittle fracture**. *Computer Graphics, Proceedings of SIGGRAPH'99*. pp 137-146. August 1999.
- [pasko00] A. Pasko, V. Adzhiev, V.Savchenko. **Volumetric and multidimensional modeling using HyperFun**. *Proceedings of GraphiCon*. 10:28. August 2000. Moscow, Russia.
- [reeves83] W. T. Reeves. **Particle systems: A technique for modeling a class of fuzzy Objects**. *Computers Graphics, Proceedings of SIGGRAPH'83*. 17(3): 359-376. July 1983.
- [rigaudiere00] D. Rigaudière, G. Gesquière, D. Faudot. **Shape modelling with skeleton based implicit primitives**. *Proceedings of GraphiCon*. 10:174-178. August 2000. Moscow, Russia.
- [sims90] K. Sims. **Particle animation and rendering using data parallel computation**. *Computer Graphics, Proceedings of SIGGRAPH'90*. 24(4):376-405. August 1990.
- [stora98] D. Stora, P.O. Agliati, M.P. Cani, F. Neyret. **Animation de coulée de lave**. *6ème journées du groupe de travail Animation et Simulation*. Octobre 1998.
- [szeliski92] R. Szeliski D. Tonnesen. **Surface modelling with oriented particle systems**. *Computer Graphics, Proceedings of SIGGRAPH'92*. 26(2):185-199. 1992. Chicago, USA.
- [stam93] J. Stam, E. Fiume. **Turbulent wind fields for gaseous phenomena**. *Computer Graphics, Proceedings of SIGGRAPH'93*. pp121-128. August 1993.
- [stam95] J. Stam, E. Fiume. **Depicting fire and other gaseous phenomena using diffusion processes**. *Computer Graphics, Proceedings of SIGGRAPH'95*. pp129-136. August 1995. Los Angeles.
- [stam97] J. Stam. **A general animation framework for gaseous phenomena**. *ERCIM Research Reports ERCIM-01/97-R047*, ERCIM, VTT. January 1997.
- [stam99] J. Stam. **Stable fluids**. *Computer Graphics, Proceedings of SIGGRAPH'99*. pp121-128. 1999.

- [terzopoulos87] D. Terzopoulos, J. Platt, A. Barr, K. Fleisher. **Elastically deformable models**. *Computer Graphics, Proceedings of SIGGRAPH'87*. 21(4):205-214. 1987.
- [terzopoulos88] D. Terzopoulos A. Witkin. **Physically based models with rigid and deformable components**. *IEEE Computer Graphics and Applications*. pp41-46. November 1988.
- [terzopoulos89] D. Terzopoulos, J. Platt, K. Fleisher. **Heating and melting deformable models (from goop to glop)**. *Graphics Interface'89*. pp219-226. June 1989. London, Ontario.
- [tonnesen91] D. Tonnesen. **Modelling liquids and solids using thermal particles**. *Graphics Interface'91*. pp255-262. June 1991. Calgary, AL.
- [triquet01] F. Triquet, P. Meseure, C. Chaillou. **Fast polygonization of implicit surfaces**. *Journal of WSCG*, 9 : 283-290. Feb. 2001, Plzen, Czech Republic.
- [vapillon95] A. Vapillon. **Modèle physique pour la modélisation de fluides en régime turbulent**. *Rapport de DEA Signal, Image, Parole*. ACROE. Juin 1995.
- [veltkamp90] R.C. Veltkamp. **The  $\gamma$  neighborhood graph**. *CWI CS-R9034*, Stinching Mathematisch Centrum - CWI Amsterdam (The Netherlands). July 1990.
- [veltkamp92\_article] R.C. Veltkamp. **Closed G1-continuous cubic Bézier surfaces**. *CWI CS-R9226*, Stinching Mathematisch Centrum - CWI Amsterdam (The Netherlands). July 1992.
- [veltkamp92\_thesis] R.C. Veltkamp. **Closed object boundaries from scattered points**. *PhD dissertation*, CWI Amsterdam (The Netherlands). 1992. in Lecture Notes in Computer Science 885.
- [weimer99] H. Weimer. J. Warren. **Subdivision schemes for fluid flow**. *Computer Graphics, Proceedings of SIGGRAPH'99*. pp 111-120. August 1999.
- [witting99] P Witting. **Computational fluid dynamics in traditional animation environment**. *Computer Graphics, Proceedings of SIGGRAPH*. pp129-136. August 1999.
- [wyvill86] G. Wyvill, C. McPheeters, B. Wyvill. **Data structures for soft objects**. *The Visual Computer*. 2(4):227-234. August 1986.
- [xin93] J. Xin Li M. Moshell. **Modelling soil: real-time dynamic model for soil slippage and manipulation**. *Computer Graphics, Proceedings of SIGGRAPH'93*. 1993.

- [xu98] C. Xu J.L. Prince. **Snakes, shapes and gradient vector flow.** *IEEE Transactions on Image Processing*, pp359-369. March 1998.
- [yagel92] R. Yagel, D. Cohen, A. Kaufman. **Discrete ray tracing.** *IEEE Computers Graphics and Applications*. 1992.
- [yngve00] G. D. Yngve, J. F. O'Brien, J. K. Hodgins. **Animating explosions.** *Computer Graphics, Proceedings of SIGGRAPH'00*. pp29-36. 2000.
- [young96] W. Young S. McNamara. **Dynamics of a freely evolving two dimensional granular medium.** *Physical Review*. 53(5). 1996
- [yu96] J. Yu W. Pattreson. **A fire model for 2D computer animation.** *Eurographics'96, Workshop on Computer Animation and Simulation*. 1996.
- [zhao01] H. K. Zhao, S. Osher, R. Fedkiw. **Fast Reconstruction using the level set method.** *First IEEE Workshop and Level Set Methods, in conjunction with the 8th International Conference on Computer Vision (ICCV)*. pp194-202. 2001. Vancouver, Canada.





# **ANNEXES**



# Annexe A Croissance de cellules végétales

---

Le modèle physique que nous avons réalisé est une coupe de cellule végétale, la visualisation du modèle est bidimensionnelle.

Les végétaux sont constitués de cellules qui leur assurent leur structure, leur forme et contrôlent leurs fonctions vitales. Les cellules sont constituées de différentes parties dont certaines sont propres à l'espèce de la plante. Ainsi certaines possèdent des chloroplastes et des vacuoles tandis que d'autres n'en n'ont pas. Mais, toute cellule végétale a une **paroi cellulaire**, un **noyau** et un **cytoplasme**.

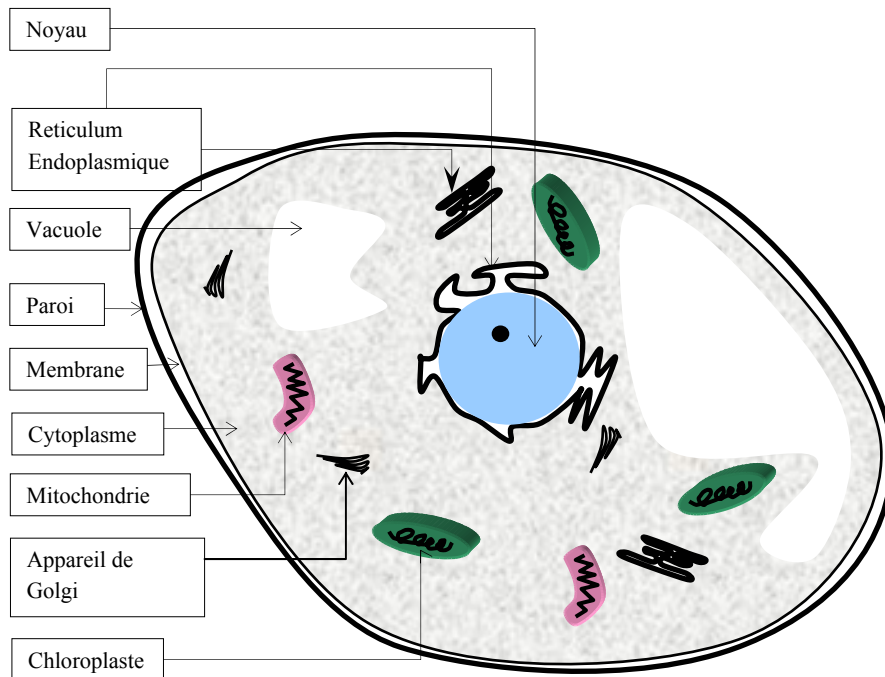
- ❑ Le **noyau**, de forme sphérique ou ovale, situé près du centre de la cellule, contrôle les processus exécutés par la cellule.
- ❑ Le **cytoplasme**, granuleux mais clair, se déplace pour assurer la vie de la cellule.
- ❑ La **paroi**, composée de cellulose, supporte et protège la cellule. Son épaisseur varie en fonction de la structure végétale qu'il représente : plus la structure végétale est solide (épaisseur), plus cette paroi est épaisse.

Nous disposons d'un modèle de croissance de cellule végétale ([France95]). Le modèle a été réalisé pour analyser l'accrochage entre cellules à la jonction entre la feuille et la tige de plante. Les cellules que nous allons visualiser auront une structure propre aux cellules de plantes vertes.

## 1. Quelques notions de biologie végétale

La cellule est entourée d'une **paroi cellulaire** enfermant une **membrane plasmique** contenant le **cytoplasme**. Le cytoplasme, dans lequel baignent les organites et molécules de la cellule, est en perpétuel mouvement pour assurer la vie de la cellule. Sa matière est granuleuse tout en étant claire. A proximité du centre de la cellule se trouve le contrôleur de la cellule, le **noyau** de forme sphérique ou

ovale. Le **réticulum endoplasmique**, n'entoure pas seulement la cellule, mais se trouve répartie dans le cytoplasme, tout comme les **appareils de Golgi**, les **mitochondries** et les **chloroplastes**, qui



**Figure 90 : Coupe d'une cellule végétale**

donnent sa couleur verte à la plante. Le dernier élément important constituant la cellule végétale est la **vacuole**, une poche remplie de liquide. Une cellule peut contenir plusieurs vacuoles qui grandissent en même temps que la cellule.

Le cytoplasme est composé d'un cytosquelette source de la cyclose (courant à l'intérieur du cytoplasme). Le cytosquelette est un réseau de fibres qui parcourt le cytoplasme. Il apporte un soutien mécanique à la cellule, lui conserve sa forme, lui permet de changer de forme. Ces fibres constituent l'ossature de la cellule mais aussi sa musculature.

## 2. Définition des courbes splines cardinales

La paroi, la membrane, le réticulum endoplasmique, l'appareil de Golgi, la vacuole ont une forme qui se prête à l'utilisation de courbes. Nous utiliserons des splines pour représenter ces composants. Ces courbes sont des splines pour lesquelles la direction du vecteur tangent en un point  $P_k$  est définie comme étant celle du segment  $[P_{k-1}P_{k+1}]$ . Il n'y a donc qu'un seul degré de liberté : la longueur du vecteur tangent. Les équations suivantes sont celles des courbes splines cardinales.

$$\begin{aligned}
C_k(t) &= D_0(t)P_{k-1} + D_1(t)P_k + D_2(t)P_{k+1} + D_3(t)P_{k+2} \\
\begin{cases} D_0(t) = -rH_0(t) \\ D_1(t) = H_1(t) - D_3(t) \\ D_2(t) = H_2(t) - D_0(t) \\ D_3(t) = r_{k+1}H_3(t) \end{cases} \\
\begin{cases} H_0(t) = t - 2t^2 + t^3 \\ H_1(t) = 1 - 3t^2 + 2t^3 \\ H_2(t) = 3t^2 - 2t^3 \\ H_3(t) = -t^2 + t^3 \end{cases}
\end{aligned}$$

Les fonctions  $H_i(t)$  ont les fonctions utilisées pour calculer les courbes splines hermitiennes de niveau 1. Lorsque tous les  $r_k$  sont égaux à 0.5 alors on parle de courbes cardinales basiques, courbe fréquemment utilisée en modélisation géométrique.

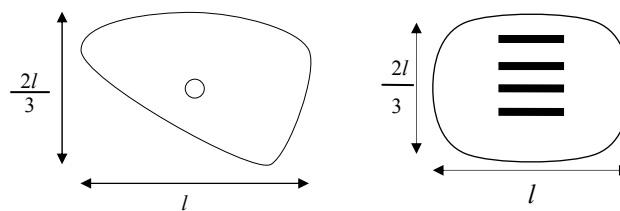
$$C_{0.5}(t) = \frac{1}{2} \left[ (-t + 2t^2 - t^3)P_{k-1} + (2 - 5t^2 + 3t^3)P_k + (t + 4t^2 - 3t^3)P_{k+1} + (-t^2 + t^3)P_{k+2} \right]$$

Nous avons choisi d'utiliser les courbes cardinales basiques pour visualiser les différents composants de la cellule.

### 3. Modélisation des composants

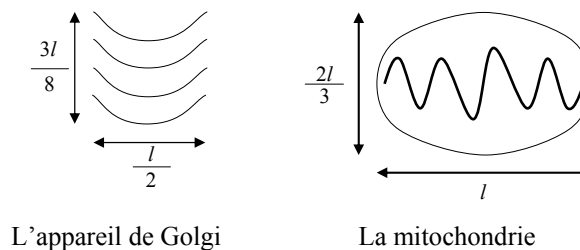
La visualisation des composants de la cellule se fait par des courbes cardinales basiques fermées. La membrane délimite le cytoplasme, il est alors naturel de représenter les deux entités par une seule et même courbe pour laquelle on visualise l'espace qu'elle délimite ainsi que son contour. La paroi est une délimitation extérieure du volume de la cellule, une courbe de contour suffit à la symboliser. Pour ces deux corps, on utilise les masses ponctuelles comme points de contrôle des courbes.

Chaque composant interne de la cellule a une forme de base (cf. Figure 91). Il a une taille définie par la dimension de la cellule à maturité. La longueur  $l$  indiquée sur les schémas correspond à la distance entre le centre de la cellule et son pourtour. Ces différents composants croissent en même temps que la cellule. Nous expliquerons à la section 4, la manière dont nous spécifierons ce phénomène.



Le noyau

Le chloroplaste



**Figure 91 : Forme de base des différents composants internes**

Le noyau est représenté par une courbe fermée où le contour et l'intérieur sont visualisés. Nous avons choisi de représenter son germe pour mieux le distinguer. Pour les chloroplastes, mitochondries et appareil de Golgi nous avons choisi une schématisation classique utilisée en biologie végétale.

La croissance du cytoplasme est liée à la croissance de la membrane et au cytosquelette. Ce cytosquelette est modélisé par les liaisons internes de la cellule. Nous avons choisi d'ignorer la structure du modèle physique donc les interactions entre masses ponctuelles ne peuvent servir à la visualisation de la cyclose<sup>20</sup>. Une manière simple de représenter la granularité du cytoplasme ainsi que ces mouvements est d'appliquée une texture, régénérée de façon aléatoire à chaque instant de la simulation, sur son « intérieur ».

L'intérieur de la vacuole est blanc. Sa taille est proportionnelle à la taille de la cellule. Il est possible que deux vacuoles fusionnent au cours de la croissance. Les vacuoles sont définies par des points de contrôle liés aux points de contrôle de la membrane et aux emplacements des autres composants internes.

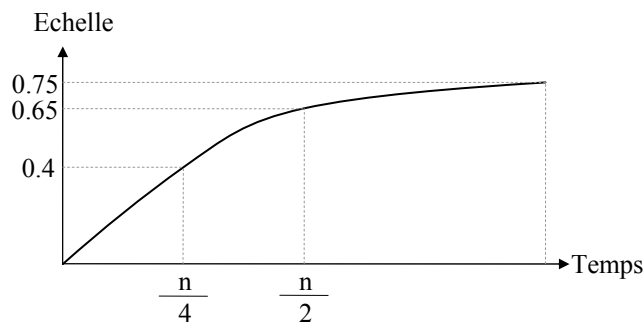
Le réticulum endoplasmique possède une forme complexe qui nécessite un grand nombre de points de contrôle ainsi qu'une recherche sur la manière de faire coller ce composant au noyau de la cellule. Sa représentation n'est pas indispensable pour reconnaître une cellule végétale, nous ne le modéliserons donc pas.

## 4. Croissance

Chaque module de la cellule croît. La paroi, la membrane et le cytoplasme suivent les mouvements des masses ponctuelles du modèle physique.

Nous nous servons d'une fonction cubique pour spécifier la courbe de croissance des composants. Nous l'avons choisi de façon à ce qu'elle évolue de manière similaire à la cellule (cf. Figure 92). Chaque composant a une taille finale proportionnelle à la taille de la cellule à maturité. Pour simuler leur croissance, chaque module est mis à l'échelle suivant la fonction d'évolution pour un instant donné.

<sup>20</sup> La cyclose est le courant cytoplasmique qui anime le cytoplasme et par là-même les composants s'y trouvant.



**Figure 92 : Fonction d'évolution de la croissance des composants internes**

## 5. Déplacement

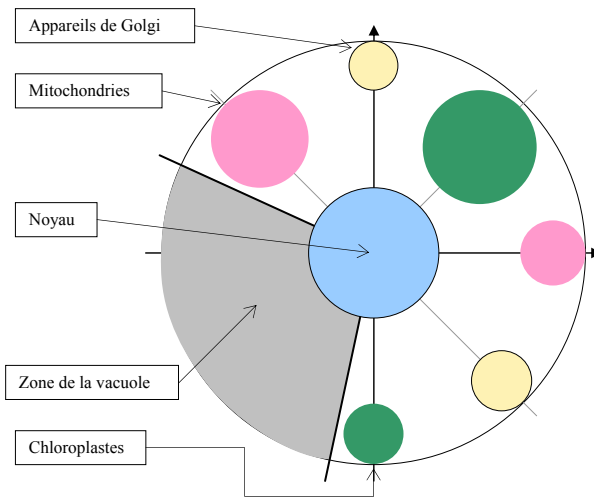
La cyclose déplace les composants internes se trouvant dans le cytoplasme. Il faut donc les animer en fonction des déplacements des positions des points de contrôle servant à la modélisation du cytoplasme. Chaque module se déplace en fonction d'une direction qui lui est affectée. Ce mouvement pour qu'il semble naturel est perturbé par des scalaires aléatoires.

Comme pour toute animation faisant intervenir plusieurs composants, il faut s'assurer que ces modules ne sortent pas du cadre dans lesquels ils évoluent et n'entrent pas en collision. Le second problème n'en est pas un pour nous. En effet nous visualisons une coupe d'un objet tridimensionnel, donc les composants peuvent se superposer sans que cela ne soit un inconvénient. En ce qui concerne la restriction de mouvement, nous approximons chaque entité par un cercle circonscrit. La cellule est, quant à elle, approximée par son cercle inscrit. Il est donc simple de tester si un composant sort de son champ d'action. La Figure 93 est une représentation de la cellule à maturité si les déplacements n'étaient pas aléatoires.

## 6. Conclusion

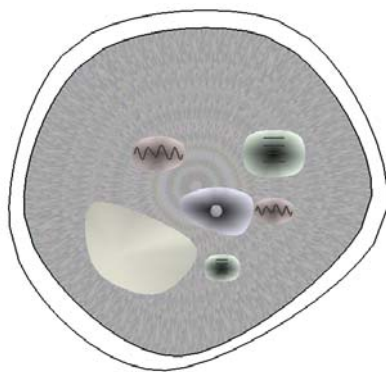
Les visualisations obtenues avec cette méthode sont intéressantes car la cellule prend ainsi vie. Nous utilisons une schématisation de la cellule pratiquée dans le domaine de la biologie végétale. Une personne ayant des notions en ce domaine doit être capable d'identifier notre modèle ainsi visualiser.



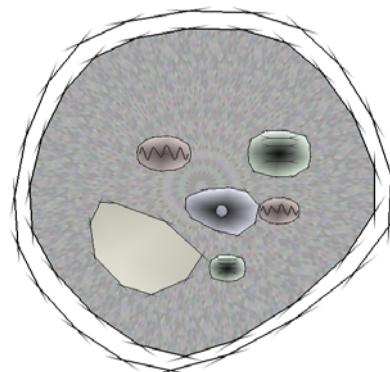


**Figure 93 : Structuration de la cellule avec ces composants**

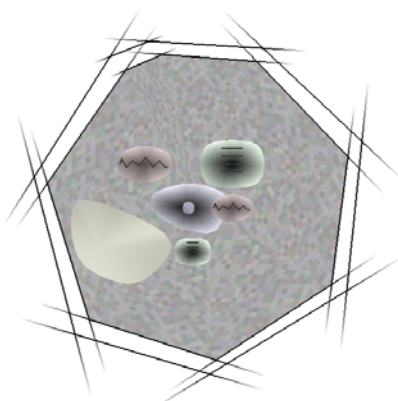
Pour que l'utilisateur puisse représenter ces modèles de différentes façons nous avons mis en place un procédé qui s'apparente à une méthode non photo-réaliste (NPR). Chaque spline visualisée est composée d'un ensemble de lignes mis bout à bout. Ce nombre de ligne peut être modifié par l'utilisateur pour rendre un contour plus ou moins grossier pour les composants internes ainsi que pour les composants de pourtour. Nous obtenons ainsi une variété de visualisation possible (Figure 94).



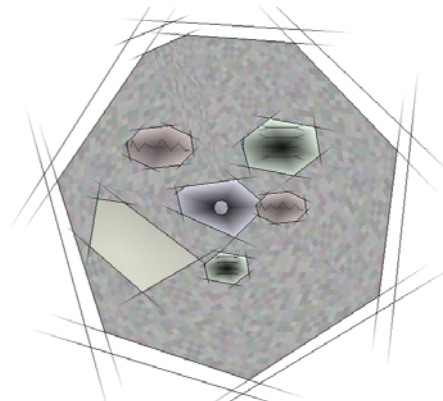
Cellule hexagonale – image 335



Cellule hexagonale – image 335



Cellule heptagonale – image 257



Cellule heptagonale – image 378

**Figure 94 : Différentes visualisations possibles pour différentes formes de cellule**

## Annexe B Spécifications CFP2000

---

Voici les différents tests CFP2000 du CPU 2000 sur les flottants. Seuls les tests sur lesquels nos recherches se sont basées (171.swim et 172.mgrid) sont explicités. Pour connaître la nature des autres tests se reporter à [www.spec.org/osg/cpu2000/CFP200/](http://www.spec.org/osg/cpu2000/CFP200/).

Benchmark	Language	Category
168.wupwise	Fortran 77	Physics / Quantum Chromodynamics
171.swim	Fortran 77	Shallow Water Modeling
172.mgrid	Fortran 77	Multi-grid Solver: 3D Potential Field
173.applu	Fortran 77	Parabolic / Elliptic Partial Differential Equations
177.mesa	C	3-D Graphics Library
178.galgel	Fortran 90	Computational Fluid Dynamics
179.art	C	Image Recognition / Neural Networks
183.quake	C	Seismic Wave Propagation Simulation
187.facerec	Fortran 90	Image Processing: Face Recognition
188.amp	C	Computational Chemistry
189.lucas	Fortran 90	Number Theory / Primality Testing
191.fma3d	Fortran 90	Finite-element Crash Simulation
200.sixtrack	Fortran 77	High Energy Nuclear Physics Accelerator Design
301.apsi	Fortran 77	Meteorology: Pollutant Distribution

### 1. 171.swim

**Benchmark Author**

Paul N. Swarztrauber

**Benchmark Program General Category**

Meteorology: Shallow Water Modeling

**Benchmark Description**

Benchmark weather prediction program for comparing the performance of current supercomputers. The model is based on the paper, "The Dynamics of Finite-Difference Models of the Shallow-Water Equations", by Robert Sadourny, J. ATM. SCIENCES, VOL 32, NO 4, APRIL 1975.

Adapted by SPEC for use in the SPEC CPU Suites as an example of a compute intensive floating point program that was once relegated only to "supercomputers" but can now be done on current computer systems.

**Input Description**

It does a 1335x1335 area array of data and iterates over 512 timesteps.

**Output Description**

It prints the diagonal elements of the velocity field.

**Programming Language**

Fortran 77

**Known portability issues**

None

**Reference**

"The Dynamics of Finite-Difference Models of the Shallow-Water Equations", by Robert Sadourny, J. ATM. SCIENCES, VOL 32, NO 4, APRIL 1975.

And

G.-R. Hoffmann, P. N. Swarztrauber, and R. A. Sweet, Aspects of using multiprocessors for meteorological modeling, In: Multiprocessing in Meteorological Models, G.-R. Hoffman and D.F. Snelling, eds., Springer-Verlag, 1988, pp. 125-196.

Last updated: 16 August 2000

## 2. 172.mgrid

**Benchmark Name**

172.mgrid

**Benchmark Author**

Eric Barszcz, Paul O. Frederickson

**Benchmark Program General Category**

Multi-grid Solver: 3D Potential Field

**Benchmark Description**

172.mgrid demonstrates the capabilities of a very simple multigrid solver in computing a three dimensional potential field.

Adapted by SPEC from the NAS Parallel Benchmarks with modifications for portability and a different workload.

### **Input Description**

1. It solves only a constant coefficient equation, and only on a uniform cubical grid.
2. It solves only a single equation, representing a scalar field rather than a vector field.

### **Output Description**

In addition to echoing some of the inputs, the main part of the output is from the smoothed approximate inverse. Only a small fraction of the smoothed output is printed, just enough to assure that all work is being done and at to check intermediate results for accuracy. In addition the L2 norm and Inf norms are used as a checksum of the output.

### **Programming Language**

Fortran 77

### **Known portability issues**

None

### **Reference**

<http://www.nas.nasa.gov/Pubs/TechReports/NASreports/NAS-95-020>

Last update: 16 August 2000

---

## RESUME

Un phénomène naturel se manifeste par des dynamiques, plus ou moins complexes, aux formes diverses et à topologie variable. Ce mouvement est induit soit par une intervention extérieure, soit par un état interne, qui induit une réorganisation de matière (une agitation ou un écoulement par exemple).

CORDIS-ANIMA, système de modélisation et de simulation numérique d'objets physiques, permet par la gestion de quelques paramètres de réaliser toutes sortes de phénomènes sans connaître l'expression formelle de la physique sous-jacente à celui-ci. A l'aide de ce formalisme, nous avons réalisé des modèles de pâtes, de sable, de gel (2D et 3D), de fluides turbulents (2D) et de croissance végétales.

Les calculs de simulations produisent un nuage de points matériels qui font partie intégrante de la matière simulée. Ces points n'étant pas disposés sur la surface de l'objet, il n'est dès lors pas possible de réaliser directement un rendu de la surface de cet objet. Nous avons donc mis au point des méthodes de construction volumique qui à partir des informations produites par la simulation, génèrent les indications volumiques manquantes.

En étudiant les paramètres physiques des phénomènes simulés, en caractérisant les dynamiques internes révélées par les mouvements des points matériels calculés, ou encore en tentant de reproduire simplement les comportements cognitifs qui permettent à un observateur de "deviner" la forme d'un objet à partir d'une information de forme incomplète (i.e. un nuage de point issu de la simulation), nous avons réalisé un ensemble de techniques de construction et de rendu adaptées aux types de phénomènes simulés.

---

## Modelling and Visualisation of natural phenomena simulated with a physically-based particle system

---

### ABSTRACT

A natural phenomenon is a set of complex dynamics with various shapes and fickle topology. The movement is due to an intervention, or an internal state, which underlie matter reorganization.

CORDIS-ANIMA, a generic modeller and simulator for physical objects, is able to synthesize, with only few parameters, a large variety of phenomena, without a formal knowledge of their physics. With this formalism, we achieve sand to turbulent fluid models.

A simulation provides a set of scattered and unorganized points describing the matter flow. Since these points do not lay on the object's surface, the surface rendering is not a trivial operation. We developed construction methods, which generate volumes from partial information coming from the simulation by looking for the lacking information.

The study of physical parameters, the description of internal dynamics revealed by the movements of material points, the reproduction of cognitive behaviours that let an observer identify the object form from unorganised points, allowed the development of a set of built-in and rendering methods suited to the types of simulated phenomena.

---

**Spécialité :** Imagerie, Vision, Robotique

---

**Mots clés :** Synthèse d'images, animation par ordinateur, modèle physique particulière, visualisation d'un nuage de points, animation de phénomènes naturels.

---

Laboratoire ICA  
Institut National Polytechnique de Grenoble  
46 avenue Félix Viallet  
38031 GRENOBLE CEDEX