



Université de Montréal

# Animation de fluides viscoélastiques à base de particules

par

Simon Clavet

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la faculté des études supérieures

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.)

en informatique

août 2005

© Simon Clavet, 2005

Université de Montréal  
Faculté des études supérieures

Ce mémoire de maîtrise intitulé

Animation de fluides viscoélastiques à base de particules

présenté par  
Simon Clavet

a été évalué par un jury composé des personnes suivantes :

Président: Neil F. Stewart

Directeur de recherche: Pierre Poulin

Membre: Balázs Kégl

# Sommaire

Ce mémoire introduit une méthode de simulation de fluides viscoélastiques permettant de générer des animations réalistes de substances telles que la peinture ou la boue alors qu'elles éclaboussent sur des objets en mouvement. La simulation est basée sur un système de particules évoluant dans le temps en tenant compte de l'incompressibilité, de la tension de surface, de la viscosité, de l'élasticité et de la plasticité. L'incompressibilité et la tension de surface sont simultanément obtenues à l'aide d'une nouvelle procédure appelée *relaxation de double densité*, qui fait évoluer les particules selon deux forces de pressions opposées. Cette technique minimise les changements de volume tout en empêchant le regroupement des particules en agrégats. Des effets de tension de surface tels que la formation de gouttes et de filaments émergent naturellement de ce processus.

L'élasticité et la plasticité sont obtenues en plaçant des ressorts de longueur de repos variable entre les particules voisines. Des effets de plasticité non-linéaire (permettant de générer des substances telles que la plasticine) sont simulés en modifiant les longueurs de repos des ressorts selon des règles simples.

Une extension permettant les interactions entre le fluide et des objets dynamiques est aussi proposée. Nous présentons divers scénarios de simulations, incluant la simulation de pluie, de fontaines, d'argile, de tourbillons de miel, de fluides adhérant sur des objets, et d'objets flottants. La méthode est robuste et stable, et peut animer interactivement des éclaboussures détaillées.

## Mots clefs :

Phénomènes naturels, animation basée sur la physique, dynamique des fluides, fluides viscoélastiques, systèmes de particules, *smoothed particle hydrodynamics*, infographie.

# Abstract

We present a new particle-based method for viscoelastic fluid simulation. We achieve realistic small-scale behavior of substances such as paint or mud as they splash on arbitrary geometries. Incompressibility and particle anti-clustering are enforced with a new procedure called *double density relaxation*, which updates particle positions according to two opposing pressure terms. From this process surface tension effects emerge, enabling drop and filament formation.

Elastic and plastic effects are obtained by adding springs with varying rest length between particles. We achieve non-linear plastic behavior by means of simple rest length update rules.

We also extend the technique to handle interaction between fluid and dynamic objects. Various simulation scenarios are presented including rain drops, fountains, clay manipulation, swirling honey, stickiness of fluids on surfaces, and floating objects. The method is robust and stable, and can animate complex splashing behavior at interactive framerates.

## **Keywords :**

Natural phenomena, physically-based animation, computational fluid dynamics, viscoelastic fluids, surface tension, particle systems, smoothed particle hydrodynamics, computer graphics.

# Table des matières

<b>Remerciements</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Simulation de fluides . . . . .	2
1.2 Contributions . . . . .	3
<b>2 État de l’art</b>	<b>5</b>
2.1 Dynamique des fluides . . . . .	5
2.2 Navier-Stokes . . . . .	7
2.3 Méthodes eulériennes . . . . .	8
2.4 Méthodes lagrangiennes . . . . .	11
2.5 Smoothed particle hydrodynamics . . . . .	12
2.6 Navier-Stokes sous forme lagrangienne . . . . .	14
2.7 Grilles vs. Particules . . . . .	15
<b>3 Schéma d’intégration</b>	<b>17</b>
3.1 Schéma explicite . . . . .	18
3.2 Schéma de prédiction-relaxation . . . . .	19
3.3 Relaxation séquentielle . . . . .	21
3.4 Pas de simulation . . . . .	22
<b>4 Incompressibilité et tension de surface</b>	<b>24</b>
4.1 Densité et pression . . . . .	25
4.2 Algorithme de relaxation . . . . .	27
4.3 Anti-agrégation . . . . .	29
4.4 Tension de surface et courbure . . . . .	32

---

4.5	Effets de tension de surface par forces locales . . . . .	36
<b>5</b>	<b>Viscoélasticité</b>	<b>39</b>
5.1	Viscosité . . . . .	39
5.2	Elasticité . . . . .	41
5.3	Plasticité . . . . .	42
<b>6</b>	<b>Interactions avec des objets</b>	<b>44</b>
6.1	Calcul du champ de distance . . . . .	45
6.2	Contact et friction . . . . .	46
6.3	Adhérence . . . . .	49
<b>7</b>	<b>Implémentation et résultats</b>	<b>51</b>
7.1	Hachage spatial . . . . .	51
7.2	Extraction de surface . . . . .	52
7.3	Résultats . . . . .	53
<b>8</b>	<b>Conclusion</b>	<b>59</b>
8.0.1	Travaux Futurs . . . . .	60
	<b>Bibliographie</b>	<b>64</b>

# Table des figures

1.1	Exemple de résultat obtenu avec notre méthode. On observe la formation de filaments et de gouttes tombant sous l'effet de la gravité. . . . .	1
1.2	Représentations de la vitesse selon les deux paradigmes de simulation. . . . .	2
1.3	Jet de boue sur le représentant d'une puissance militaire non-identifiée. . . . .	4
2.1	Advection semi-lagrangienne d'un champ scalaire selon un champ de vitesse. . . . .	9
2.2	Champ de distance. . . . .	10
2.3	Le verre d'eau de Enright <i>et al.</i> [EMF02]. . . . .	10
2.4	Le fluide viscoélastique de Goktekin <i>et al.</i> [GBO04]. . . . .	11
2.5	Force de Lennard-Jones. . . . .	12
2.6	Le verre d'eau de Müller <i>et al.</i> [MCG03]. . . . .	13
2.7	Simulation de Goktekin <i>et al.</i> [GBO04], où un filament trop mince reste en équilibre dans l'air. . . . .	16
3.1	Schéma explicite sans viscosité pour la contrainte $x = 0$ . . . . .	18
3.2	Schéma de prédiction-relaxation. . . . .	19
3.3	Schéma de prédiction-relaxation pour la contrainte $x = 0$ . . . . .	20
4.1	Région d'influence autour de la particule $i$ . . . . .	24
4.2	Trois choix possibles pour le noyau de densité. . . . .	25
4.3	Si la pression est positive la particule $i$ <i>pousse</i> ses voisines. . . . .	26
4.4	Si la pression est négative la particule $i$ <i>attire</i> ses voisines. . . . .	26
4.5	Accumulation des déplacements sur la particule centrale. . . . .	29
4.6	Agrégation de particules sous l'effet de la relaxation de simple densité (sans gravité). . . . .	30

---

4.7	Artefact d'empilement se produisant si on utilise une simple force de répulsion pour empêcher l'agrégation. . . . .	30
4.8	Noyaux de densité. . . . .	31
4.9	Formation de structures régulières sous l'effet de la relaxation de double densité. . . . .	32
4.10	Forces d'attraction causant la tension de surface. . . . .	34
4.11	Liquide sous l'effet de la gravité dans un contenant. À droite, le liquide subit la relaxation de proximité. On remarque la courbure caractéristique de la surface. . . . .	36
4.12	Goutte oscillante. . . . .	36
4.13	Filaments se formant naturellement avec la relaxation de double densité. . . . .	37
5.1	Impulsion de viscosité entre une paire de particules (gauche). Effet lissant de la viscosité (droite). . . . .	39
5.2	Calcul de l'impulsion de viscosité. . . . .	40
5.3	Moulage d'un personnage élastique. . . . .	42
5.4	Changement de longueur de repos d'un ressort. . . . .	42
5.5	Changement de longueur de repos en fonction de la longueur courante. . . . .	43
5.6	Déformation permanente d'une boule de matériel viscoélastique. . . . .	43
6.1	Interactions entre particules et objets. . . . .	47
6.2	Particules adhérant sous une sphère avant de former une goutte. . . . .	50
6.3	Effet de l'adhérence sur la forme de la surface. On remarque la formation du ménisque caractéristique à droite. . . . .	50
7.1	Structure de recherche de voisins en 2D. Les cellules nécessaires sont créés dynamiquement et placées dans une table de hachage. . . . .	52
7.2	Liquide tombant sur le <i>Bunny</i> de Stanford. Les particules sont contraintes par la géométrie du bec de la théière avant de couler sur le lapin en formant des gouttes et des filaments. . . . .	54
7.3	Une vague se replie sur elle-même alors qu'on remplit un contenant rectangulaire d'un liquide visqueux. . . . .	54

---

7.4	Pluie visqueuse sur un personnage. Chaque goutte de pluie est formée d'environ 20-40 particules. Les gouttes se divisent et s'unissent alors qu'elles coulent sur la surface du personnage. . . . .	55
7.5	Liquide remplissant un bac contenant un lapin relativement lourd. Cet exemple illustre le couplage bidirectionnel liquide/objets : le liquide réagit à la présence du lapin en le contournant, et le lapin réagit à la présence du liquide en se déplaçant. . . . .	55
7.6	Le même lapin, mais beaucoup plus léger, flotte sur le liquide. . . . .	56
7.7	Le <i>buddha</i> sous une douche d'or liquide. Le liquide se sépare en de longs filaments visqueux. . . . .	56
7.8	Tourbillon de sirop s'étendant sur un tabouret. L'émergence du tourbillon est causée par l'élasticité de la substance alors qu'elle se replie sur elle-même. Les plis caractéristiques se dissipent rapidement sous l'effet de la plasticité. . . . .	56
7.9	Éclaboussure détaillée sur un personnage en rotation. Ce scénario de collision extrême, où un liquide interagit violemment avec un objet en mouvement puis se sépare en structures complexes se déplaçant à grande vitesse, est particulièrement difficile à traiter autrement qu'avec un système de particules de grande stabilité. . . . .	57
7.10	Jet de liquide éclaboussant le général. Le jet est suffisamment puissant pour faire basculer le personnage dans l'abîme. . . . .	57
7.11	Modification interactive des paramètres. . . . .	58

# Remerciements

Cette maîtrise n'aurait pas été possible sans le support des membres du LIGUM. Je me souviendrai avec nostalgie de ces belles années passées dans le temple du triangle pixelisé du deuxième étage, à discuter de science et de jeux vidéos avec Philippe Beaudoin, François Duranleau, Jean-François Dufort, Fabrice Rousselle, Luc Leblanc, Yannick Simard, Nicolas Bergeron, Jean-François St-Amour, Martin Granger-Piché, Emric Epstein et Lawrence Hum.

Mon professeur, Pierre Poulin, est probablement la personne qui mérite le plus ma reconnaissance. C'est avec un enthousiasme contagieux qu'il m'a montré comment faire apparaître de belles images sur un écran d'ordinateur. Pierre m'a gratifié de sa confiance complète et aveugle, me laissant faire à peu près ce que je voulais tout en me donnant le feed-back, le soutien et la pression nécessaire pour produire un résultat scientifiquement significatif et joli à regarder.

Je remercie Neil F. Stewart et Balázs Kégl d'avoir bien voulu reviser ce mémoire. Je remercie aussi mes sources de financement - et d'endettement - , c'est-à-dire le fond de recherche NSERC, l'aide financière de Québec, et mon père (les taux d'intérêt de ce dernier sont imbattables).

Finalement, je profite de cette page pour mentionner mes amis (en particulier Maxime Bélanger, Simon Pilette, Marie Ouimet, Maryse Boisvert, Philippe Sicard et tous les membres du Groundhog Day Squad), mon frère Julien, et mes parents Lise et Jacques. Sans la présence de ces gens, ma vie serait considérablement plus plate.

# Chapitre 1

## Introduction

Les comportements complexes de la fumée et des liquides sont très difficiles à reproduire avec les techniques d'animation traditionnelles. On peut difficilement imaginer un artiste 3D modélisant la surface de l'eau d'une fontaine, et encore moins l'animer en décidant à la main où seront les gouttes à chaque trentième de seconde. Même un animateur chevronné mettra des jours à animer l'éclaboussure la plus bénigne.

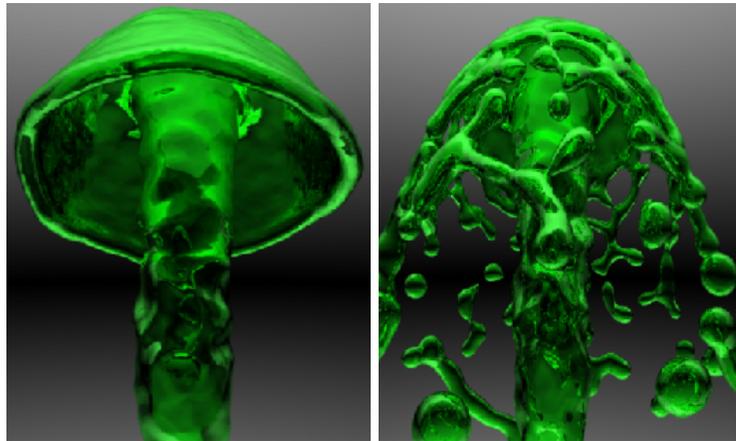


FIG. 1.1 – Exemple de résultat obtenu avec notre méthode. On observe la formation de filaments et de gouttes tombant sous l'effet de la gravité.

Pour obtenir une animation véritablement complexe, on n'a d'autre choix que d'abandonner nos valeureux artistes et de se tourner vers l'inquiétant monde de la simulation physique. Simuler numériquement un phénomène naturel consiste à représenter son état dans l'ordinateur, puis à modifier cet état à chaque pas de temps selon les lois de la physique. Une animation physique complexe est plus difficile à contrôler qu'une animation

traditionnelle, mais elle est incomparablement plus réaliste.

La mécanique des fluides a été extensivement étudiée par les physiciens et les numériciens, ce qui a mené à l'élaboration d'algorithmes permettant de simuler le comportement complexe d'un fluide en mouvement. Ces algorithmes ont été utilisés avec succès en ingénierie, par exemple pour le design de turbines ou d'imprimantes à jet d'encre. Ces techniques ont récemment été introduites dans le contexte de la synthèse d'images, créant par le fait même un nouveau champ de recherche. Ici, la science la plus aride est au service de l'art et du divertissement : l'hydrodynamique computationnelle entre aujourd'hui dans le monde trépidant des jeux vidéo.

## 1.1 Simulation de fluides

Deux grandes catégories de simulation existent : les grilles eulériennes et les particules lagrangiennes (Figure 1.1). La stratégie eulérienne est présentement la plus utilisée, et elle a permis d'obtenir les simulations les plus réalistes à ce jour. Elle consiste à discrétiser l'espace en une grille fixe, et à suivre l'évolution des propriétés du fluide à chaque point de la grille. Ainsi, la vitesse du fluide est représentée par un ensemble de vecteurs positionnés en des endroits fixes de l'espace. Pour une simulation de liquide, on a aussi pour chaque point d'espace un certain marqueur indiquant s'il y a ou non du liquide à cet endroit. Ce type de simulation discrétise directement les équations de la dynamique des fluides, ce qui permet de reproduire précisément les comportements variés et complexes des substances viscoélastiques.

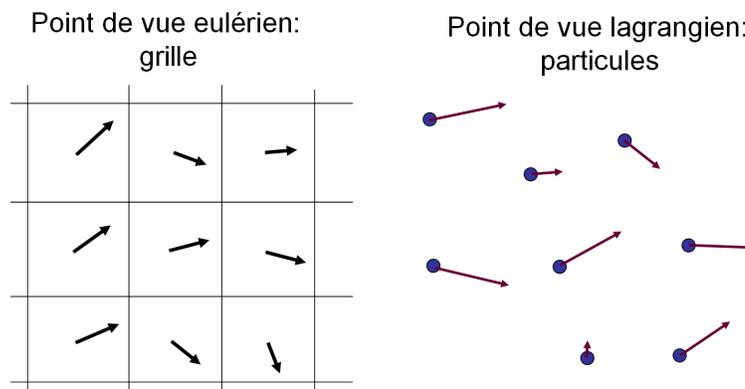


FIG. 1.2 – Représentations de la vitesse selon les deux paradigmes de simulation.

L'autre stratégie, la simulation lagrangienne, consiste à discrétiser la masse du fluide

en un ensemble de particules. Le mouvement du fluide suit l'évolution des particules dans l'espace, modifiant leur vitesse selon des forces existant entre elles.

Euler regarde les points de l'espace et se demande comment se déplace la masse s'y trouvant. Lagrange regarde les points de masse et se demande comment ils se déplacent dans l'espace.

Dans une approche eulérienne, la distance pouvant être parcourue par le fluide pendant un pas de simulation est limitée par la résolution de la grille. Avec les particules, cette limitation disparaît. Pour simuler un liquide détaillé se déplaçant à grande vitesse, suivre les changements alors qu'ils se produisent à un endroit fixe semble moins naturel que de suivre les changements se produisant le long de la trajectoire d'une particule.

Dans ce travail, nous choisissons résolument la représentation à base de particules. Nous tentons de réduire les problèmes qui ont souvent découragé son utilisation, et proposons des modifications qui renforcent sa qualité et sa robustesse. Nous présentons aussi des extensions augmentant sa généralité et sa versatilité. Notre but est de proposer un modèle à la fois puissant et intuitif. Les paramètres de la substance devraient être facile à ajuster, permettant d'obtenir de l'eau, de la boue ou de la plasticine dans un cadre unifié. Plutôt que de se concentrer sur la précision physique, nous tentons de rendre la simulation de substances viscoélastiques assez simple, rapide et versatile pour être utilisée dans un monde virtuel interactif.

## 1.2 Contributions

Notre travail apporte quatre contributions spécifiques : une simulation stable pour de longs pas de temps, une nouvelle façon d'obtenir la tension de surface, un système de ressorts à longueur dynamique pour la viscoélasticité, et finalement le couplage à deux directions avec un système de corps rigides. Après un rapide survol de l'état de l'art au Chapitre 2, ce mémoire présente en détail ces contributions.

Notre schéma d'intégration, détaillé au Chapitre 3, est moins précis que ceux utilisés habituellement, mais il permet d'obtenir des simulations stables et visuellement correctes dans un contexte de temps de calcul limité. Nous utilisons un schéma de type prédiction-relaxation : nous avançons d'abord les particules à une position prédite, puis nous modifions (*relaxons*) directement ces positions pour qu'elles respectent certaines contraintes.

La contrainte principale régissant le mouvement de nos particules est une *relaxation de double densité*, qui consiste à modifier les positions des particules en fonction de deux différentes mesures de la distribution des particules voisines. En plus de minimiser les changements de volume, cette méthode permet d’obtenir un autre comportement important : la tension de surface. La relaxation de double densité est détaillée au Chapitre 4.

Dans notre système, la viscoélasticité est obtenue en ajoutant et retirant dynamiquement des ressorts entre paires de particules voisines (Chapitre 5). Les longueurs de repos des ressorts sont modifiées à chaque pas de temps, ce qui permet d’obtenir les comportements caractéristiques de substances semi-rigides telles que la plasticine ou la boue (Figure 1.3).

Notre dernière contribution est l’intégration du fluide dans un système de corps rigides (Chapitre 6). Le couplage est à deux directions, c’est-à-dire que non seulement les objets peuvent pousser le fluide, mais le fluide peut également pousser les objets. Ainsi, plusieurs scénarios de simulation intéressants peuvent être obtenus, tel qu’un jet de liquide faisant tomber un personnage ou encore un objet flottant dans un liquide en mouvement.

Dans les deux derniers chapitres, nous présentons certains détails de l’implémentation, décrivons les résultats des tests que nous avons effectués pour valider notre méthode, et finalement nous proposons quelques avenues de recherche future.



FIG. 1.3 – Jet de boue sur le représentant d’une puissance militaire non-identifiée.

# Chapitre 2

## État de l'art

Dans ce chapitre, nous décrivons d'abord brièvement les principes de base du comportement des fluides, puis nous traitons des différentes façons de les simuler numériquement selon les méthodes les plus représentatives de celles utilisées de nos jours.

### 2.1 Dynamique des fluides

Contrairement à un solide, un fluide (gaz ou liquide) n'a pas de forme précise. Les forces agissant sur lui ne sont pas contrebalancées par des forces internes sévères comme dans le cas des solides. Toutefois, les fluides, comme les solides, doivent respecter deux principes universels : la conservation de la masse et la conservation de la quantité de mouvement.

La conservation de la masse stipule que rien ne se perd et rien ne se crée. Dans le contexte d'une simulation de liquide, cette contrainte prend une signification très concrète et souvent moins triviale qu'il n'y paraît : il ne faut pas que les gouttes disparaissent.

La conservation de la quantité de mouvement stipule quant à elle qu'un morceau de matière ne subissant aucune force doit conserver sa vitesse. Les forces externes telles que la gravité peuvent toutefois changer sa quantité de mouvement. À l'intérieur du fluide, des forces internes existent entre les morceaux de matière : ils se poussent et se tirent. Pour que ces forces conservent la quantité de mouvement, elles doivent suivre le principe d'action-réaction : si je te pousse, je dois nécessairement accélérer dans la direction opposée.

Ces deux principes, appliqués à un continuum où la vitesse peut être différente à chaque point, mènent à un système d'équations différentielles décrivant le mouvement. Étant donnée une situation initiale, on peut résoudre ces équations pour prédire l'évolution du système.

Chaque substance a une façon particulière de réagir aux déformations. C'est ce qui fait la différence entre une brique, du miel, de l'eau, ou de la plasticine. Regardons un point particulier de la substance. Étant donné une certaine déformation locale, le point de matière pousse et tire les points autour de lui. Dans les équations, la déformation est représentée par une matrice. Cette matrice encode de quelle façon le bout de matière est en train de s'étirer, de se compresser ou de se tordre. Le besoin de tirer et pousser les points voisins est appelé *stress*, et est aussi représenté par une matrice. C'est en fait une force, mais plutôt que de simplement tirer ou pousser selon une direction, le stress veut tordre le bout de matière d'une façon arbitraire. Le stress répond à la déformation. Dans une brique, il répond très énergiquement : aucune déformation n'est tolérée. Le stress tend à compenser exactement la déformation potentielle, et ainsi elle n'a même pas le temps de se produire. Dans l'eau, le stress répond d'une façon un peu plus subtile à la déformation : il l'accepte sans réagir, en autant que celle-ci ne change pas le volume du liquide.

La science des substances (la rhéologie) est dédiée à l'étude de la fonction prenant en argument la déformation et retournant le stress que cette déformation induit dans le matériel. Prenons un point dans un élastique. Dans ce cas, les matrices de déformation et de stress se réduisent à des scalaires. La déformation décrit comment l'élastique est étiré ; le stress indique comment un point pousse ses voisins. Pour un élastique linéaire, une déformation  $\mathbf{x}$  engendre une force  $\mathbf{F}$  selon la formule  $\mathbf{F} = -k\mathbf{x}$ , capturant le fait que plus on est étiré, plus on tire (pousse négativement). C'est la relation stress-déformation la plus simple qu'on puisse imaginer. En général,  $\mathbf{F}$  et  $\mathbf{x}$  sont des matrices (plus spécifiquement, des tenseurs), et la fonction qui les relie peut être arbitrairement complexe.

Dans le cas des fluides, il importe de préciser ce qu'on entend par "déformation". Un fluide ne peut être déformé par rapport à une forme de repos inexistante. Dans ce cas, le stress ne dépend pas d'une déformation particulière (*e.g.* être trop à droite par rapport à une position de repos parmi ses voisins), mais plutôt du taux de déformation (*e.g.*

aller à droite par rapport à ses voisins). Mathématiquement, le taux de déformation correspond aux dérivées spatiales du champ de vitesse. Pour trouver les valeurs de vitesse au prochain pas de temps, on cherche sa variation dans le temps, c'est-à-dire sa dérivée temporelle. Les dérivées spatiales et temporelles du champ de vitesse d'un fluide sont contraintes par un système d'équations aux dérivées partielles : les équations de Navier-Stokes.

## 2.2 Navier-Stokes

Les contraintes présentes dans un fluide apparaissent sous forme de deux équations de conservation : une pour la masse et une pour la quantité de mouvement. La contrainte de conservation de masse met en relation le champ de densité  $\rho$  avec le champ de vitesse  $\mathbf{v}$  selon *l'équation de continuité* :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (2.1)$$

signifiant que le flot de matière entrant dans un volume infinitésimal est égal au changement de sa densité. Dans le cas incompressible, la densité ne change pas, et l'équation se réduit à la *contrainte de divergence zéro* :  $\nabla \cdot \mathbf{v} = 0$ . La divergence du champ de vitesse est un scalaire indiquant à quel point le champ se dilate. Une divergence négative en un point signifie que le champ se comprime en ce point (les flèches convergent). Le fait que la conservation de la masse se traduise en une contrainte de divergence zéro est la base des simulateurs eulériens, et on verra plus loin que c'est à la fois leur force et leur faiblesse.

La deuxième équation de Navier-Stokes exprime la conservation de la quantité de mouvement. Dans le cas des fluides visqueux, à densité constante et sans élasticité, elle s'exprime sous la forme de l'équation vectorielle

$$\rho(\mathbf{v}_t + \mathbf{v} \cdot \nabla \mathbf{v}) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v} \quad (2.2)$$

où  $\mathbf{g}$  est la gravité,  $\mu$  est le coefficient de viscosité et  $p$  est le champ de pression. Pour connaître l'évolution du champ  $\mathbf{v}$ , il suffit d'isoler  $\mathbf{v}_t$  ( $= \frac{\partial \mathbf{v}}{\partial t}$ ). Cette équation est l'équivalent continu de  $m\mathbf{a} = \mathbf{F}$ , avec  $\rho$  à la place de  $m$  et une force en trois termes. Étant donné que nous n'utilisons pas directement cette équation dans ce travail, nous nous limiterons à une brève discussion de ces composantes.

Le premier terme de droite indique que la matière tend à aller de la haute pression vers la basse pression. Dans le cas d'un fluide compressible, la pression dépend de la densité selon l'équation d'état de la substance. Pour les gaz parfaits, la relation est linéaire. Pour les liquides, la relation entre la pression et la densité est plus rigide, c'est-à-dire qu'une petite augmentation de densité peut correspondre à une grande augmentation de pression, selon un exposant pouvant être élevé. Dans le cas d'un fluide idéal incompressible, la pression s'ajuste de façon à ce que la soustraction de son gradient rende le champ de vitesse sans divergence (calculer cette pression revient à solutionner un système linéaire). Les autres termes composant la force sont la gravité et la viscosité. Cette dernière est proportionnelle au laplacien du champ de vitesse. On peut l'interpréter comme une force de lissage, qui tend à égaliser le champ de vitesse.

La partie gauche de l'équation correspond à la dérivée matérielle du fluide, c'est-à-dire la variation de vitesse *alors qu'on se déplace avec la substance*. Le terme d'*advection*  $\mathbf{v} \cdot \nabla \mathbf{v}$  sert à exprimer l'équation en format eulérien, permettant de considérer le changement de vitesse à un point fixe de l'espace plutôt qu'à un point fixe de masse. On verra que la simulation par particules simplifie grandement la formulation habituelle du système, en revenant en quelque sorte à ses racines.

À partir d'une situation initiale et de conditions aux frontières, les équations de Navier-Stokes déterminent complètement le champ de vitesse pour tous les temps futurs. Malheureusement, elles sont fortement couplées et non-linéaires (à cause du terme  $\mathbf{v} \cdot \nabla \mathbf{v}$ , qui résulte en des composantes quadratiques). Même pour une situation simple, les équations n'admettent aucune solution analytique. On n'a d'autre choix que de discrétiser le problème et de le solutionner numériquement.

Deux grands paradigmes de discrétisation existent : les grilles eulériennes et les particules lagrangiennes. Les grilles discrétisent l'espace et les particules discrétisent la masse.

## 2.3 Méthodes eulériennes

Dans le cadre d'une simulation eulérienne, l'état d'un système est représenté par un ensemble de champs définis sur une grille. En particulier, la vitesse est échantillonnée à chaque point de la grille.

Foster et Metaxas [FM97] furent les premiers en infographie à discrétiser directement

les équations de la dynamique des fluides sur une grille 3D, de façon à obtenir un comportement réaliste de liquide. Chaque opérateur différentiel ayant sa contrepartie discrète, il est possible d'évoluer le champ de vitesse dans le temps selon les équations de Navier-Stokes. La région de la grille occupée par le liquide est représentée à l'aide de particules sans masse qui suivent le champ de vitesse. Si une cellule de la grille contient au moins une particule, alors elle est marquée "liquide".

Stam [Sta99] a amélioré la simulation eulérienne en introduisant la méthode d'advection semi-lagrangienne. Cette méthode stabilise et simplifie le transport des quantités sur la grille, et permet par exemple une simulation rapide de fumée.

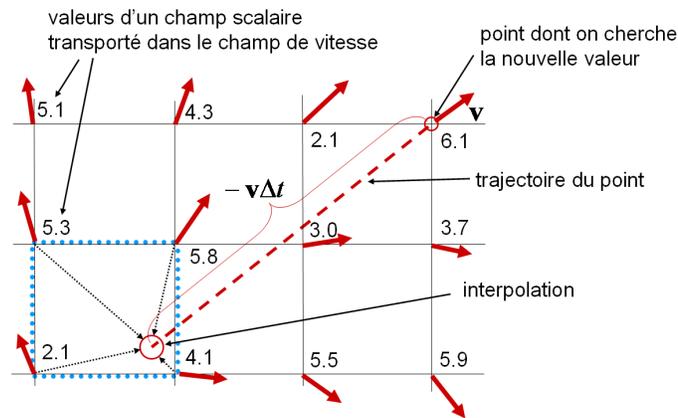


FIG. 2.1 – Advection semi-lagrangienne d'un champ scalaire selon un champ de vitesse.

Plutôt que d'avancer le champ selon ses dérivées spatiales (en discrétisant directement l'équation d'advection), on trouve les nouvelles valeurs en projetant la position des points de grille au pas de temps précédent et en interpolant le champ à cet endroit. Puisque l'advection n'est plus calculée localement, la vitesse peut augmenter jusqu'à plusieurs cellules par pas de temps sans causer d'instabilités. La Figure 2.1 illustre la méthode semi-lagrangienne pour l'advection d'un champ scalaire. Dans l'équation de Navier-Stokes, le terme non-linéaire d'advection  $\mathbf{v} \cdot \nabla \mathbf{v}$  transporte le champ de vitesse lui-même (voir [Sta99] pour plus de détails).

Foster et Fedkiw [FF01] ont appliqué la méthode semi-lagrangienne aux liquides. La surface est représentée par un *level-set* (une isosurface d'une fonction distance définie sur la grille) advecté selon le champ de vitesse (Figure 2.2). Cette représentation permet de conserver une surface lisse et bien définie, mais elle souffre de pertes de masse dans les régions à haute courbure. Pour contourner ce problème, ils corrigent la fonction de

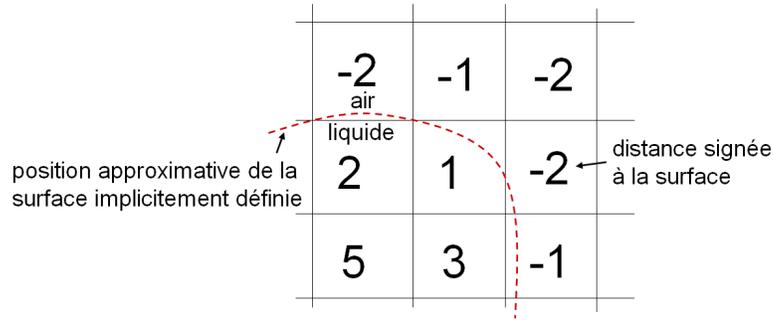
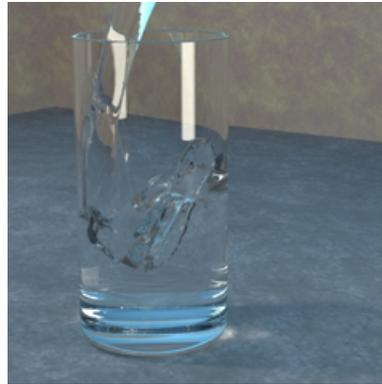


FIG. 2.2 – Champ de distance.

distance aux endroits critiques en plaçant un ensemble de particules dans le liquide, près de la surface. Enright *et al.* [EMF02] placent aussi des particules de correction à l'extérieur du liquide puis ils y extrapolent les vitesses pour obtenir un mouvement de surface de plus haute qualité (Figure 2.3).

FIG. 2.3 – Le verre d'eau de Enright *et al.* [EMF02].

Losasso *et al.* [LGF04] ont étendu la simulation eulérienne au paradigme de la multi-résolution en simulant des liquides et de la fumée sur un octree, produisant des résultats de haute définition en des temps de calcul raisonnables.

Les grilles peuvent aussi être utilisées pour simuler les comportements viscoélastiques. Carlson *et al.* [CMHT02] simulent des fluides très visqueux dont les paramètres changent au cours du temps. Ceci leur permet d'animer des objets qui fondent. Récemment, Goktekin *et al.* [GBO04] ont introduit à la communauté graphique une méthode eulérienne pour simuler non seulement la viscosité, mais aussi l'élasticité et la plasticité (Figure 2.4). Les informations de déformation sont discrétisées sur la grille, et elles sont advectées selon le champ de vitesse. Ainsi, ils peuvent à la limite simuler un bloc parfaitement

élastique de façon eulérienne.

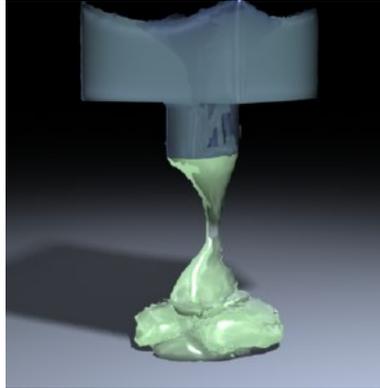


FIG. 2.4 – Le fluide viscoélastique de Goktekin *et al.* [GBO04].

## 2.4 Méthodes lagrangiennes

Reeves [Ree83] fut le premier à utiliser des particules dans un contexte graphique. Elles sont rapidement devenues un outil incontournable dans la simulation d'objets fous, tels que du feu, de la fumée, ou une chute d'eau. À la base ces particules sont simplement des objets graphiques, sans composante physique. Elles apparaissent à une position source, se déplacent selon des trajectoires prédéfinies et disparaissent lorsque leur durée de vie est écoulée. Une particule est rendue dans l'image sous forme d'un point ou plus généralement d'un polygone texturé pouvant être semi-transparent. On peut ainsi facilement produire un jet de poussière derrière les roues d'un véhicule sur un chemin de terre.

Un traitement plus physique du mouvement multiplie les possibilités offertes par les particules. Elles se déplacent selon un champ de vitesse (ou d'accélération) défini dans l'espace ; elles subissent des collisions avec les objets de l'environnement ; et elles interagissent avec les particules environnantes.

Par exemple, Miller et Pearce [MP89] obtiennent une simulation de base de liquides et d'objets qui fondent. Ils s'inspirent de la mécanique globulaire en plaçant des forces de type Lennard-Jones entre les particules. La force est orientée selon l'axe entre les particules, et sa magnitude dépend de leur distance (Figure 2.5). Elles se poussent si elles sont rapprochées, s'attirent si elles sont à distance moyenne, et n'interagissent plus si elles sont trop distantes. Terzopoulos *et al.* [TPF89] ont aussi utilisé ce type de force

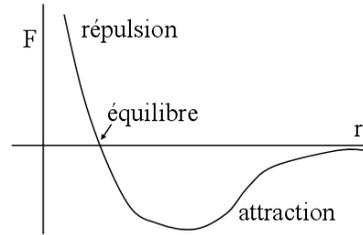


FIG. 2.5 – Force de Lennard-Jones.

entre les particules, et ont de plus ajouté des ressorts entre les particules. Par contre, leur modèle de ressort est trop simple pour pouvoir adéquatement simuler généralement la plasticité. Contrairement à ce que nous faisons dans notre travail, les longueurs de repos de leurs ressorts ne changent pas : seules les constantes de rigidité changent. Les ressorts disparaissent à mesure que le solide fond.

La prochaine étape dans le développement des méthodes de simulation à base de particules est d'ajouter des propriétés plus fondamentales aux particules. En fait, il faut voir les particules comme un véritable échantillon de tout ce qui se passe autour de sa position. Plutôt que de considérer la particule comme un point de masse en interaction avec ses voisins, on peut la considérer comme une représentante d'un champ continu. Par exemple, un champ scalaire tel que la température peut être évalué à tout point de l'espace en moyennant les valeurs de températures échantillonnées sur les particules environnantes. Selon la version forte ce paradigme, une particule représente véritablement une région étendue et diffuse de l'espace. Un ensemble de particules peut ainsi représenter un champs scalaire ou vectoriel quelconque. Les équations de l'hydrodynamique s'appliquant aux continus, il est souhaitable de les discrétiser directement sur les particules. Pour ce faire, le formalisme SPH a été introduit.

## 2.5 Smoothed particle hydrodynamics

La théorie SPH a été développée initialement par Lucy [Luc77] et par Gingold et Monaghan [GM77] pour traiter certains problèmes d'astrophysique. Puisqu'elles représentent adaptativement l'espace en se plaçant aux endroits critiques et en abandonnant les régions plus calmes, les particules SPH sont idéales pour simuler efficacement des phénomènes tels que la formation des galaxies.

Desbrun et Gascuel [DG96] ont appliqué la théorie SPH à la simulation de corps

déformables. Müller *et al.* [MCG03] l'ont utilisée pour simuler interactivement des liquides (Figure 2.6).



FIG. 2.6 – Le verre d'eau de Müller *et al.* [MCG03].

Une particule SPH représente un échantillon *macroscopique* de matière, avec toutes les propriétés mesurables qu'on peut attendre d'un véritable morceau de substance. Une particule SPH d'eau n'est pas une molécule d'eau. C'est plutôt une estimation de ce qui se passe dans une région étendue autour d'elle.

Imaginons qu'un champ scalaire  $A$  est échantillonné sur un ensemble de particules. L'estimation de  $A$  en un point  $\mathbf{r}$  quelconque est la moyenne pondérée des valeurs de  $A$  définies sur les particules environnantes. La contribution d'une particule dépend de sa distance et de son volume estimé. L'équation de base est donc

$$A(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} W_h(\mathbf{r} - \mathbf{r}_j) \quad (2.3)$$

où  $m_j$ ,  $\rho_j$  et  $\mathbf{r}_j$  sont la masse, la densité et la position de la particule  $j$ , et  $A_j$  est la valeur du champ  $A$  échantillonnée sur cette particule. On remarque que  $m_j/\rho_j$  correspond au volume estimé de la particule ( $\rho = m/V$ ).  $W_h$  est une fonction radiale et décroissante, par exemple une cloche gaussienne de largeur  $h$ , ou encore un simple cône. Le paramètre  $h$  est le rayon d'interaction des particules. La valeur du champ est d'autant plus lissée que le nombre de particules contenues dans une sphère de rayon  $h$  est grand.

Pour connaître la densité de la particule  $i$ , on réécrit l'équation 2.3 avec  $A_j = \rho_j$  et  $\mathbf{r} = \mathbf{r}_i$  :

$$\rho_i = \rho(\mathbf{r}_i) = \sum_j \rho_j \frac{m_j}{\rho_j} W_h(\mathbf{r}_i - \mathbf{r}_j) = \sum_j m_j W_h(\mathbf{r}_i - \mathbf{r}_j). \quad (2.4)$$

La densité est donc simplement une somme pondérée des masses voisines.

Pour avoir un modèle utile, il reste à introduire les opérateurs différentiels. Par exemple, pour le gradient, on obtient

$$\nabla A(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla W_h(\mathbf{r} - \mathbf{r}_j) \quad (2.5)$$

où l'opérateur est simplement commuté avec la sommation et les constantes. On a une expression analytique simple pour le gradient de  $W_h$ , et donc le calcul de  $\nabla A$  est aussi facile que celui de  $A$ .

La méthode SPH est flexible et puissante. Tout système d'équations différentielles pouvant s'exprimer sous forme lagrangienne peut être résolu avec cette méthode. Une équation en forme lagrangienne donne le taux de changement d'une quantité *alors qu'on se déplace avec la matière*. Par exemple, la dérivée de la température donnera le taux de changement de la température d'un morceau de matière en mouvement plutôt que celui d'un point fixe de l'espace.

## 2.6 Navier-Stokes sous forme lagrangienne

Les équations de Navier-Stokes peuvent s'exprimer sous forme lagrangienne, et la formulation résultante s'en trouve grandement simplifiée. En fait, la principale difficulté rencontrée lorsqu'on dérive la formulation habituelle est de la mettre sous forme eulérienne.

Premièrement, étant donné que par hypothèse les particules ont des masses constantes et que le nombre de particules ne change pas, l'équation de continuité 2.1 est automatiquement satisfaite. Il est certain que nous ne perdrons pas de masse. Deuxièmement, on remarque que la partie gauche de l'équation 2.2 contient la dérivée matérielle ( $\mathbf{v}_t + \mathbf{v} \cdot \nabla \mathbf{v}$ ). C'est le taux de changement du vecteur vitesse alors qu'on suit le morceau de matière. Cette forme eulérienne de l'équation de Navier-Stokes a justement été obtenue à partir de la forme lagrangienne, en développant la dérivée totale  $d\mathbf{v}/dt$  en ses dérivées partielles. Dans le cadre lagrangien, l'évolution de la vitesse  $\mathbf{v}_i$  d'une particule se calculera directement avec cette dérivée, selon le système simplifié

$$\frac{d\mathbf{v}_i}{dt} = \frac{\mathbf{f}_i}{\rho_i} \quad \mathbf{f}_i = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}_i. \quad (2.6)$$

Le terme d'advection  $\mathbf{v} \cdot \nabla \mathbf{v}$ , qui ne servait qu'à porter le mouvement sur la grille statique, disparaît.

La densité  $\rho_i$  d'une particule est calculée en additionnant les contributions des particules voisines selon l'équation 2.4. Le gradient de la pression et le laplacien de la vitesse peuvent aussi être directement intégrés dans le formalisme SPH.

## 2.7 Grilles vs. Particules

Sur une grille, le champ de vitesse est contraint à l'incompressibilité en s'assurant que le flot net pénétrant chaque cellule est nul : ce qui y entre est égal à ce qui en sort. Puisque ce qui sort d'une cellule est directement relié à ce qui entre dans la cellule voisine, le système d'équations résultant est fortement couplé. L'incompressibilité est fondamentalement non-locale : solutionner les problèmes de divergence non-nulle dans une cellule donnée implique potentiellement la modification de la vitesse de cellules arbitrairement éloignées. On peut exprimer le problème comme un système linéaire  $Ax = b$  à résoudre. La matrice étant *creuse* (la plupart de ses éléments sont nuls), le système est facile à solutionner. Ceci est la force principale des grilles : il existe des méthodes très efficaces pour rendre le champ de vitesse incompressible. Notons que ce calcul doit être très précis, à défaut de quoi on peut perdre ou gagner de la masse. En effet, si une cellule a une divergence négative, le fluide disparaîtra. Si la divergence est positive, on assistera à un non-moins perturbant phénomène de génération spontanée.

Le problème réside dans le fait que le champ de vitesse défini sur la grille a une résolution fixe, et que celle-ci ne correspond pas nécessairement à la résolution des petits détails de la surface. Si un détail du liquide (une petite goutte par exemple) est plus petit qu'une cellule, rien ne dit que le champ de vitesse fera évoluer le détail adéquatement. Malheureusement pour la simulation sur grille, la formation de petits détails est inévitable. L'évolution d'un mince filament de liquide amène nécessairement à des détails plus petits qu'une cellule. Dès que la cellule contient plus d'air que de liquide, le simulateur ne peut plus donner une vitesse adéquate au petit détail du liquide. La goutte prend alors la vitesse de l'air environnant : elle reste là sans tomber.

Ce désagréable artefact, où un détail trop petit a soudainement un comportement insensé, est selon nous extrêmement difficile à éviter dans le cadre d'une simulation eulérienne. La source théorique de la difficulté est que rien ne peut empêcher la simulation de devenir sous-résolue, et qu'une simulation sous-résolue se comporte très mal. On voit très clairement qu'il y a un problème lorsque le liquide se sépare en petites gouttes

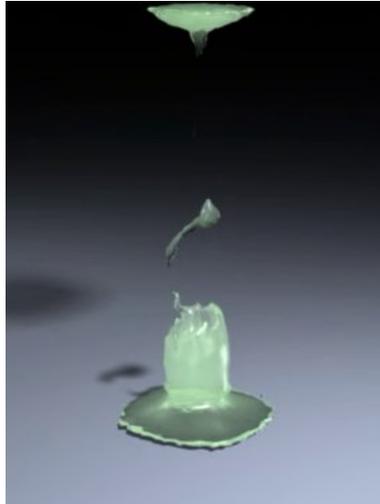


FIG. 2.7 – Simulation de Goktekin *et al.* [GBO04], où un filament trop mince reste en équilibre dans l'air.

qui se fixent dans l'air, oscillent bizarrement ou disparaissent sans plus d'explications.

Les partisans des simulations eulériennes répondront à cet argument numérico-philosophique en avançant qu'on ne peut rien faire si la résolution n'est pas suffisante, que la simulation soit eulérienne ou lagrangienne. Notre réponse est qu'une simulation lagrangienne sous-résolue est moins perturbante qu'une simulation eulérienne sous-résolue.

La sous-résolution, dans le cas d'une simulation à base de particules, se produit lorsque le nombre de particules n'est pas suffisant pour représenter adéquatement le phénomène simulé. Une goutte formée d'une seule particule n'aura pas un comportement très intéressant : elle ne se déformera pas, elle ne se séparera pas, elle restera sphérique et tombera en respectant la loi de la gravité. Avec la représentation consistante fournie par les particules, la qualité d'une simulation sous-résolue se dégrade plus élégamment qu'avec une grille.

Cette résistance à la sous-résolution s'ajoute aux autres arguments en faveur des particules, en l'occurrence l'indépendance par rapport à un système de coordonnées particulier et la conservation automatique de la masse. Si on ajoute à ces arguments leur désarmante simplicité d'implémentation, leur généralité, leur versatilité et leur puissance éprouvée quotidiennement dans un contexte de production, on peut présager avec confiance que nos bonnes vieilles particules ont encore de beaux jours devant elles...

## Chapitre 3

# Schéma d'intégration

La simulation physique consiste à prédire le futur. Deux objectifs contradictoires peuvent être choisis : la précision, et la possibilité de prédire un futur éloigné. Les algorithmes utilisés par les physiciens atteignent des niveaux de précision élevés, mais ils ne fonctionnent que pour de courts pas de temps : ils doivent typiquement en faire plusieurs milliers par seconde. Si on utilise les mêmes méthodes avec de longs pas de temps, on obtient des résultats insensés. Plutôt que devenir graduellement imprécises, ces méthodes cessent simplement de fonctionner. On assiste alors ce qu'on appelle une *instabilité numérique*, ou pire encore, une *explosion numérique*, un événement algorithmique intrigant qui consiste en une véritable explosion de la substance simulée.

Dans un contexte de simulation interactive, on ne peut se permettre de simuler beaucoup plus de 30 pas de temps par seconde, et il faut éviter à tout prix les explosions numériques. C'est pourquoi on est prêt à sacrifier la précision en échange de la stabilité.

Notre simulation est constituée d'un ensemble de particules se déplaçant dans l'espace sous l'influence de forces et de contraintes. La procédure principale du programme est le *pas d'intégration*, dans lequel on considère la position et la vitesse courantes des particules, et on prédit la position et la vitesse qu'elles devront avoir un trentième de seconde plus tard. Avant de détailler les contraintes et les forces que nous appliquons, nous discutons ici du choix de schéma d'intégration utilisé. Notre but est de proposer un mécanisme de satisfaction de contraintes robuste et stable pour de longs pas de temps. Il s'avère que la technique résultante est intuitive et facile à implémenter.

### 3.1 Schéma explicite

Habituellement, un pas de simulation physique est composé des étapes suivantes. Premièrement, diverses forces sont calculées et accumulées pour chaque objet. Ces forces modifient alors les vitesses, qui sont finalement utilisées pour mettre à jour les positions des objets. Selon ce paradigme, on traite les contraintes en les convertissant d'abord en forces, avant de s'en servir pour modifier les vitesses des objets. Pour visualiser ce qui se passe, prenons un exemple simple : une particule en une dimension. Soit une contrainte spécifiant qu'elle doit rester à la position  $x = 0$ . Supposons que la particule est initialement à la position  $x = 5$ , avec une vitesse  $v = 0$ , et que le pas de temps est 1 ( $\Delta t = 1$ ).

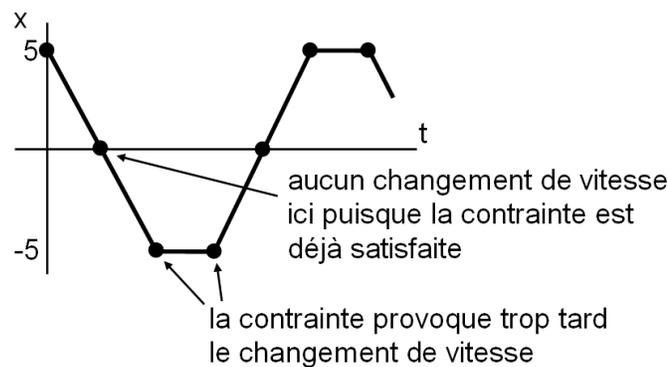


FIG. 3.1 – Schéma explicite sans viscosité pour la contrainte  $x = 0$ .

Comment la contrainte  $x = 0$  se traduit-elle en changement de vitesse? On voit intuitivement que le mieux qu'elle peut faire est de diminuer la vitesse de 5 (elle sera donc de  $-5$ ). En utilisant cette vitesse modifiée pour avancer la position, on arrive à  $x = 0$ . Lors du prochain pas de temps, la contrainte sera déjà satisfaite au début, et donc aucune force ne sera produite. N'oublions pas que selon ce schéma, la seule chose que la contrainte peut faire est d'appliquer une accélération (une force). La vitesse restera donc  $v = -5$ , et la position à la fin de ce pas de temps sera  $x = -5$ . Au prochain pas de temps, la contrainte verra la position  $x = -5$ , et donc elle augmentera la vitesse de 5. La vitesse remontera à 0 (elle était de  $-5$ ), et la position restera  $x = -5$ . Continuons avec un dernier pas :  $x = -5$ , donc la vitesse augmente encore de 5. À la fin de ce pas de temps, la particule est revenue à 0 et a une vitesse positive (voir la Figure 3.1).

On voit que la traduction d'une contrainte en changement de vitesse se produisant au début du pas de temps mène à une oscillation autour de la position de repos. Ceci

se produit puisque la contrainte ne dépend que de la position au début du pas et elle ne peut que modifier la vitesse utilisée au cours du pas. Limiter l'effet d'une contrainte à produire une accélération dépendante de la position n'est pas suffisant pour obtenir l'effet souhaité. Pour éviter les oscillations autour de la contrainte, on ajoute habituellement un amortissement limitant artificiellement le mouvement. Nous verrons qu'une autre solution existe : il est possible d'éviter complètement les oscillations en permettant à la contrainte de modifier directement la position plutôt que de tout faire par l'intermédiaire d'une force.

La procédure que nous venons de décrire correspond à un schéma d'intégration de Verlet. Mathématiquement, on peut l'écrire de la façon suivante :

$$\mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_{n-\frac{1}{2}} + \Delta t \mathbf{F}(\mathbf{x}_n)/m \quad (3.1)$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \mathbf{v}_{n+\frac{1}{2}}. \quad (3.2)$$

La vitesse est définie au milieu du pas de temps (tel qu'indiqué par l'indice fractionnaire  $n+\frac{1}{2}$ ), ce qui est cohérent puisqu'elle est proportionnelle à la différence entre les positions au début et à la fin du pas, et donc elle "existe" nécessairement quelque part entre les deux. Remarquons que puisque les contraintes doivent être exprimées sous forme de forces, on doit tenir compte de la masse ( $\mathbf{F} = m\mathbf{a}$ ).

### 3.2 Schéma de prédiction-relaxation

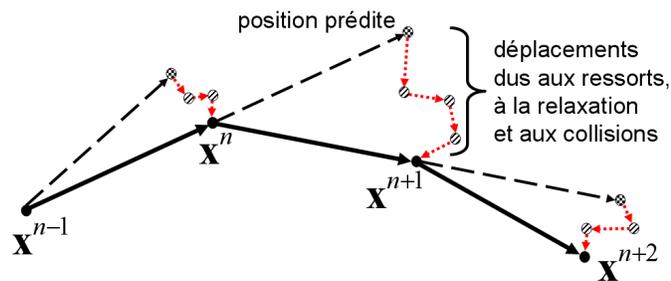


FIG. 3.2 – Schéma de prédiction-relaxation.

Dans notre cas, nous préférons avancer la simulation en utilisant un schéma de prédiction-relaxation (Figure 3.2). On avance les particules à leurs positions prédites selon leurs vitesses, puis on modifie ces positions de façon à satisfaire les contraintes. À la fin du pas de temps, les vitesses sont recalculées en soustrayant la position précédente

de la nouvelle position. Recalculer ainsi la vitesse est la clé de ce schéma : la vitesse utilisée au prochain pas sera automatiquement cohérente avec la vitesse résultante du pas courant. Le danger d'explosion numérique est limité parce que le schéma prédit les configurations délicates et réagit avant qu'elles ne se produisent. Appliquer les contraintes en modifiant directement les positions plutôt qu'en les transformant d'abord en forces simplifie et augmente la stabilité du schéma d'intégration.

Regardons ce qui se passe avec la situation décrite précédemment : la contrainte  $x = 0$  avec position et vitesse initiales  $x = 5$  et  $v = 0$ . Puisque la vitesse est nulle, la position prédite reste  $x = 5$ . On modifie cette position selon la contrainte : maintenant  $x = 0$ . La vitesse est recalculée en soustrayant la position finale de la position initiale :  $v = 0 - 5 = -5$ . Après ce premier pas, on se trouve dans la même situation qu'avec le schéma de Verlet.

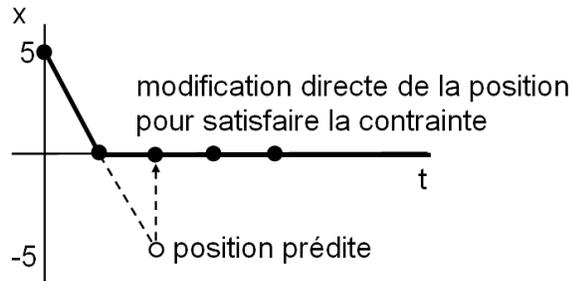


FIG. 3.3 – Schéma de prédiction-relaxation pour la contrainte  $x = 0$ .

Pour le deuxième pas, la position prédite est  $x = -5$ , puis on la modifie ensuite selon la contrainte : la position est ramenée à  $x = 0$ . On calcule la prochaine vitesse en soustrayant la position initiale de la position finale :  $v = 0 - 0 = 0$ . La vitesse à la fin du deuxième pas est nulle, et la contrainte est satisfaite. Aucune oscillation autour de la contrainte ne se produit (Figure 3.3).

Pour une seule force, le schéma de prédiction-relaxation s'écrit de la façon suivante :

$$\mathbf{x}_* = \mathbf{x}_n + \Delta t \mathbf{v}_{n-\frac{1}{2}} \quad (3.3)$$

$$\mathbf{x}_{n+1} \leftarrow \mathbf{x}_* + \Delta t^2 \mathbf{F}(\mathbf{x}_*)/m \quad (3.4)$$

$$\mathbf{v}_{n+\frac{1}{2}} = (\mathbf{x}_{n+1} - \mathbf{x}_n)/\Delta t. \quad (3.5)$$

La position prédite  $x_*$  est directement modifiée par la force due à la contrainte. La vitesse est obtenue en utilisant la position au début du pas.

Les équations précédentes peuvent être simplifiées, ce qui mène au schéma

$$\begin{aligned}\mathbf{v}_{n+\frac{1}{2}} &= \mathbf{v}_{n-\frac{1}{2}} + \Delta t \mathbf{F} \left( \mathbf{x}_n + \Delta t \mathbf{v}_{n-\frac{1}{2}} \right) / m \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \Delta t \mathbf{v}_{n+\frac{1}{2}}\end{aligned}$$

qui est similaire à l'équation 3.1, mais avec la force calculée à la position prédite.

Pour évaluer l'impact de ce changement de la position du calcul de la force, testons le schéma avec la force  $\mathbf{F} = -k\mathbf{x}$ . On obtient

$$\mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_{n-\frac{1}{2}} + \Delta t \left( -k \mathbf{x}_n - k \Delta t \mathbf{v}_{n-\frac{1}{2}} \right) / m. \quad (3.6)$$

On voit que la force effectivement calculée n'est pas  $-k\mathbf{x}$ , mais plutôt  $-k\mathbf{x} - k \Delta t \mathbf{v}$ . Le terme supplémentaire est proportionnel à la vitesse, dans le sens opposé. On peut donc le voir comme une viscosité numérique (une force proportionnelle à la vitesse, dans le sens contraire du mouvement). Étant proportionnelle à  $k$ , elle s'ajuste automatiquement à la rigidité du système. Finalement, le facteur  $\Delta t$  indique que la viscosité intrinsèque du schéma numérique tend vers 0 lorsque le pas de temps tend vers 0, ce qui est essentiel à la validité d'une méthode d'intégration.

### 3.3 Relaxation séquentielle

Avec plusieurs contraintes, notre schéma d'intégration peut s'écrire de la façon suivante :

$$\begin{aligned}\mathbf{x}_* &= \mathbf{x}_n + \Delta t \mathbf{v}_{n-\frac{1}{2}} \\ \mathbf{x}_* &\leftarrow \mathbf{x}_* + \Delta t^2 \mathbf{F}^1(\mathbf{x}_*) / m \\ \mathbf{x}_* &\leftarrow \mathbf{x}_* + \Delta t^2 \mathbf{F}^2(\mathbf{x}_*) / m \\ \mathbf{x}_* &\leftarrow \mathbf{x}_* + \Delta t^2 \mathbf{F}^3(\mathbf{x}_*) / m \\ &\dots \\ \mathbf{x}_{n+1} &= \mathbf{x}_* \\ \mathbf{v}_{n+\frac{1}{2}} &= (\mathbf{x}_{n+1} - \mathbf{x}_n) / \Delta t.\end{aligned}$$

La force est décomposée en ses différents composants  $\mathbf{F}^1$ ,  $\mathbf{F}^2$ ,  $\mathbf{F}^3$ , etc., et chacun dépend des positions déjà modifiées par les composants précédents. Cette application

séquentielle des composants de la force est analogue à ce qu'on fait souvent en simulation eulérienne, lorsque les opérateurs de viscosité, d'advection, d'incompressibilité, etc., modifient séquentiellement le champ de vitesse.

Une contrainte positionnelle modifie les positions des particules de façon à ce que le fluide ait certaines propriétés. Dans notre cas, les contraintes sont au nombre de quatre : incompressibilité, anti-agrégation, rigidité des ressorts et anti-pénétration des particules dans les objets. Ces contraintes sont souvent opposées et contradictoires. Par exemple, en écartant les particules, la contrainte d'anti-agrégation diminue la densité et donc empêche la contrainte d'incompressibilité d'atteindre sa densité constante. Aussi, respecter complètement les contraintes de ressorts rigides peut contredire les autres contraintes.

Nous avons observé qu'appliquer séquentiellement les contraintes sur les positions prédites stabilise le système. Ceci permet d'augmenter la *rigidité* des contraintes, et donc d'atteindre plus facilement les objectifs d'incompressibilité et d'élasticité. On peut intuitivement visualiser qu'en permettant à une contrainte de "savoir" ce que les autres contraintes ont fait, on lui permet de produire une réponse plus adaptée à la situation.

Imaginons deux contraintes identiques forçant une particule à être à la position  $x = 0$ , et supposons une fois de plus que la particule initialement à  $x = 5$ . Si on applique en parallèle les contraintes, la première contrainte dira "la particule doit descendre de 5", puis la deuxième contrainte dira la même chose. Le résultat combiné sera que la particule descendra de 10, et la position finale sera  $-5$ . En appliquant séquentiellement les contraintes, la deuxième ne dira rien puisqu'elle sera déjà satisfaite : la première contrainte a déjà fait tout le travail. Ainsi, la position finale sera 0, ce qu'on attendait. Concrètement, l'application séquentielle des contraintes mène à une diminution notable des vibrations apparaissant lorsque plusieurs contraintes rigides existent simultanément.

### 3.4 Pas de simulation

L'Algorithme 1 présente le pseudo-code de la procédure principale de notre programme. Premièrement, on modifie les vitesses des particules selon les forces qui ne peuvent être considérées comme des contraintes : la gravité et la viscosité. Après avoir sauvegardé les positions initiales, on avance ensuite les particules à leurs positions prédites. On traite alors la plasticité en ajoutant et retirant des ressorts, et en mo-

difiant leurs longueurs de repos. Le reste de la procédure modifie les positions prédites selon les ressorts, la double relaxation, et les collisions avec les corps rigides. Finalement, les nouvelles vitesses sont calculées en soustrayant les positions initiales des positions finales.

**Algorithme 1** : Pas de simulation. \_\_\_\_\_

1. pour chaque particule  $i$
  2.     // applique la gravité
  3.      $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t \mathbf{g}$
  4.     // modifie les vitesses selon la viscosité
  5.     appliqueViscosité
  6. pour chaque particule  $i$
  7.     // sauvegarde les positions précédentes
  8.      $\mathbf{x}_i^{\text{prev}} \leftarrow \mathbf{x}_i$
  9.     // avance aux positions prédites
  10.      $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$
  11.     // ajoute et enlève les ressorts, change leurs longueurs de repos
  12.     ajusteRessorts
  13.     // applique les déplacements dus aux ressorts,
  14.     // à la relaxation de double densité et aux collisions
  15.     appliqueRessorts
  16.     relaxeDoubleDensité
  17.     traiteCollisions
  18. pour chaque particule  $i$
  19.     // retrouve la vitesse en utilisant les positions précédentes
  20.      $\mathbf{v}_i \leftarrow (\mathbf{x}_i - \mathbf{x}_i^{\text{prev}}) / \Delta t$
-

## Chapitre 4

# Incompressibilité et tension de surface

Les particules représentant un liquide incompressible doivent idéalement occuper un volume constant. Une solution consiste à appliquer une force de répulsion entre chaque paire de particules. Une force qui dépend seulement de la distance entre les particules mène toutefois à certains problèmes : le liquide ressemble à un groupe de particules qui rebondissent les unes sur les autres. Elles s'empilent et forment des structures rigides, ce qui empêche le liquide de s'écouler naturellement. La solution que nous proposons est d'éviter les forces qui dépendent exclusivement de la distance entre deux particules, et de tenir compte du contexte. La force de repulsion dépend donc non seulement de la distance entre les particules, mais aussi de la densité de particules dans la région environnante.

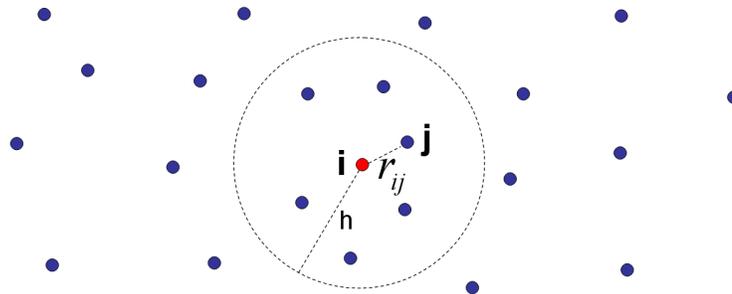


FIG. 4.1 – Région d'influence autour de la particule  $i$ .

## 4.1 Densité et pression

On évalue la densité d'une particule en additionnant les contributions de chacune de ses voisines. On considère une particule comme voisine lorsque sa distance est inférieure au rayon d'interaction  $h$ . La contribution d'une voisine est pondérée avec une fonction qu'on appelle le *noyau de densité*.

Plusieurs choix de fonctions s'offrent à nous (Figure 4.2). Un choix populaire est la cloche, représentée par une gaussienne ou un polynôme tel que  $(1 - (r/h)^2)^3$ . La cloche est lisse, ce qui lui donne de bonnes propriétés numériques, mais dans notre cas sa pente nulle à l'origine cause un problème important. En effet, on verra plus loin que le noyau de la force doit être proportionnel à la pente du noyau de densité. Si la force s'annule à l'origine, des particules superposées ne s'échangeront plus de force et ne pourront pas se séparer.

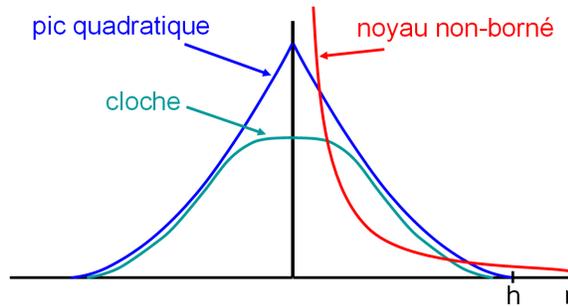


FIG. 4.2 – Trois choix possibles pour le noyau de densité.

Une autre possibilité est de prendre un noyau non-borné tel que  $(r/h)^{-3}$ . L'avantage théorique de ce noyau est que sa dimension est bien celle d'une densité, *i.e.* l'inverse d'un volume (on ne tient pas compte de la masse, qui devrait apparaître si on calculait une véritable densité). Par contre, le fait qu'il tende vers l'infini à l'origine est numériquement très désagréable. Idéalement, le noyau devrait aussi avoir un support fini (nul pour  $r > h$ ), de façon à ce que les interactions restent locales.

Notre choix de noyau de densité est le pic quadratique  $(1 - r/h)^2$ . Cette fonction et sa dérivée atteignent zéro lorsque  $r = h$ , ce qui assure que la contribution d'une voisine qui s'approche augmente gentiment. Cette fonction est bornée, et sa pente reste non-nulle près de l'origine.

Nous définissons donc la densité  $\rho$  de la particule  $i$  selon la sommation suivante :

$$\rho_i = \sum_{j \in N(i)} (1 - r_{ij}/h)^2 \quad (4.1)$$

où  $r_{ij} = |\mathbf{r}_{ij}|$ ,  $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ , et  $N(i)$  est l'ensemble des voisines de la particule  $i$  telles que  $r_{ij} < h$ .

À partir de la densité, on définit une pression. Cette pression n'est une véritable pression physique, avec des unités de force par distance carrée, mais plutôt une pseudo-pression ajustée selon nos besoins. Notre pression est simplement un indicateur de ce qu'une particule doit faire pour atteindre sa densité de repos. Une pression positive indique une densité trop élevée, et donc la particule doit *pousser* ses voisines (Figure 4.3). Une pression négative indique une densité trop faible, et donc la particule doit *attirer* ses voisines (Figure 4.4).

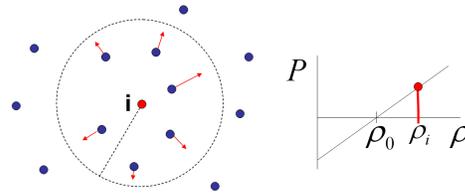


FIG. 4.3 – Si la pression est positive la particule  $i$  *pousse* ses voisines.

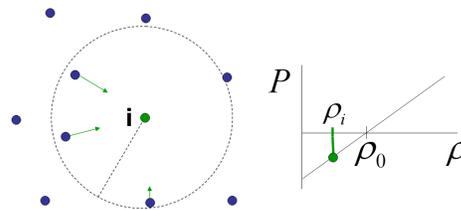


FIG. 4.4 – Si la pression est négative la particule  $i$  *attire* ses voisines.

La pression  $P_i$  est proportionnelle à la différence entre la densité de la particule  $i$  et une densité de repos  $\rho_0$  :

$$P_i = k(\rho_i - \rho_0). \quad (4.2)$$

Le paramètre  $k$  permet de spécifier la *rigidité* de la contrainte de conservation de volume. Cette formulation, malgré son aspect heuristique et intuitif, donne de bons résultats.

## 4.2 Algorithme de relaxation

La procédure de relaxation (Algorithme 2) boucle sur chaque particule  $i$ . Une itération de cette boucle est composée de deux passes sur ses voisines  $j$ . La première passe évalue la densité; la deuxième passe déplace la particule  $i$  et ses voisines en fonction de cette densité.

**Algorithme 2 :** Relaxation de densité. \_\_\_\_\_

1. pour chaque particule  $i$
2.      $\rho_i \leftarrow 0$
3.     // calcule la densité
4.     pour chaque particule  $j \in \text{voisines}(i)$
5.          $q \leftarrow r_{ij}/h$
6.         if  $q < 1$
7.              $\rho \leftarrow \rho_i + (1 - q)^2$
8.     // calcule la pression
9.      $P_i \leftarrow k(\rho_i - \rho_0)$
10.      $\mathbf{dx} \leftarrow \mathbf{0}$
11.     pour chaque particule  $j \in \text{voisines}(i)$
12.          $q \leftarrow r_{ij}/h$
13.         if  $q < 1$
14.             // applique les déplacements
15.              $\mathbf{D}_{ij} \leftarrow \Delta t^2 P_i (1 - q) \hat{\mathbf{r}}_{ij}$
16.              $\mathbf{x}_j \leftarrow \mathbf{x}_j + \mathbf{D}_{ij}$
17.              $\mathbf{dx} \leftarrow \mathbf{dx} - \mathbf{D}_{ij}$
18.      $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{dx}$

L'impulsion de pression est appliquée sous forme d'un déplacement orienté selon l'axe entre chaque paire de particules. Elle est proportionnelle à la pression, calculée selon l'équation 4.2. Ainsi, si la pression est négative (la densité est plus petite que la densité de repos), alors le déplacement sera négatif, et donc les particules se rapprocheront.

Les particules plus éloignées interagissent moins avec la particule, et ceci doit trans-

paraître dans l'équation du déplacement. Nous choisissons simplement le noyau linéaire  $(1 - r/h)$ . Le déplacement appliqué sur la particule  $j$  dans la deuxième passe de la particule  $i$  s'écrit donc :

$$\mathbf{D}_{ij} = \Delta t^2 P_i (1 - r_{ij}/h) \hat{\mathbf{r}}_{ij} \quad (4.3)$$

où  $\hat{\mathbf{r}}_{ij}$  est le vecteur unitaire pointant de la particule  $i$  à la particule  $j$ . Le noyau de déplacement  $(1 - r/h)$  fonctionne bien puisqu'il est proportionnel à la dérivée du noyau de densité  $(1 - r/h)^2$ . Le choix de ces fonctions a été déterminé en grande partie par essais et erreurs, en tentant de trouver des noyaux qui donnent les comportements les plus *gentils* (qui forcent énergiquement l'incompressibilité, mais sans introduire d'instabilités).

Il est toutefois possible de voir intuitivement pourquoi un noyau de déplacement proportionnel à la dérivée du noyau de densité donne un bon résultat. Un premier indice consiste à interpréter la densité comme un potentiel : on sait que la dérivée d'un potentiel (fonction de coût) correspond habituellement à une force (réaction au potentiel). Pensons à une particule dont la densité est trop élevée. Regardons une voisine en particulier, et sa contribution à la fonction de densité. Supposons que la pente de la fonction de densité est abrupte à cette distance. Ceci signifie que pousser cette voisine aidera beaucoup à diminuer la densité. Supposons au contraire que la pente de la densité est faible à cet endroit. Dans ce cas ça ne sert à rien de trop pousser la particule : ça n'aidera pas vraiment à diminuer la densité. La force avec laquelle cette particule poussera la voisine sera proportionnelle à la pente de la fonction de contribution à la densité. Plus le fait de pousser la particule aide à diminuer la densité, plus on la poussera. On peut voir cette idée comme une version du principe de moindre action : la particule veut atteindre son objectif en faisant le moins d'effort. Le principe intuitif consistant à "pousser d'autant plus que pousser aide" se traduit en "force proportionnelle à la dérivée du potentiel".

Les déplacements subis par les particules doivent respecter certaines conditions pour favoriser la préservation des moments linéaires et angulaires du système. Le moment linéaire (quantité de mouvement) doit absolument être préservé, à défaut de quoi des forces internes pourraient changer la vitesse globale du système. La façon la plus simple de favoriser la conservation de la quantité de mouvement est de toujours appliquer des impulsions égales et opposées entre paires de particules. Puisque nos impulsions sont

appliquées sous forme de déplacements, et puisque les particules ont la même masse, il suffit de toujours déplacer les particules selon des déplacements égaux et opposés. Pour ce qui est du moment angulaire, la stratégie la plus sécuritaire est de toujours appliquer les impulsions selon l'axe entre les particules. On peut montrer mathématiquement (et on peut visualiser intuitivement) qu'une impulsion radiale ne peut changer le moment angulaire total de la paire de particules.

La deuxième boucle sur les voisines  $j$  de la particule  $i$  déplace immédiatement la particule  $j$ , mais les déplacements appliqués sur la particule  $i$  sont accumulés puis appliqués seulement à la fin de la boucle. Ainsi, le biais dû à l'ordre de traitement des voisines est évité. La Figure 4.5 illustre ce processus.

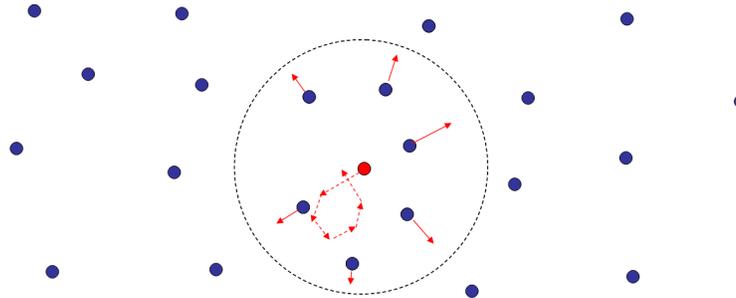


FIG. 4.5 – Accumulation des déplacements sur la particule centrale.

### 4.3 Anti-agrégation

La procédure de relaxation décrite précédemment permet de garder un volume global approximativement constant. Par contre, la distribution locale des particules n'est pas nécessairement régulière. Une particule peut atteindre sa densité de repos en attirant un petit nombre de particules près d'elle. Les particules se séparent alors en un ensemble d'agrégats, tel qu'illustré à la Figure 4.6. Lors d'une éclaboussure, les particules se séparent en groupes incohérents. Même si le liquide est globalement incompressible, il n'adopte pas la forme et le comportement naturel d'un véritable liquide.

La stratégie naïve pour empêcher l'agrégation est d'ajouter une force de répulsion entre les particules qui sont trop rapprochées. Cette solution mène toutefois à un comportement bizarre où les particules s'empilent de façon rigide (Figure 4.7). Un empilement de balles rigides ne peut ressembler à un liquide dynamique et non-visqueux (qui

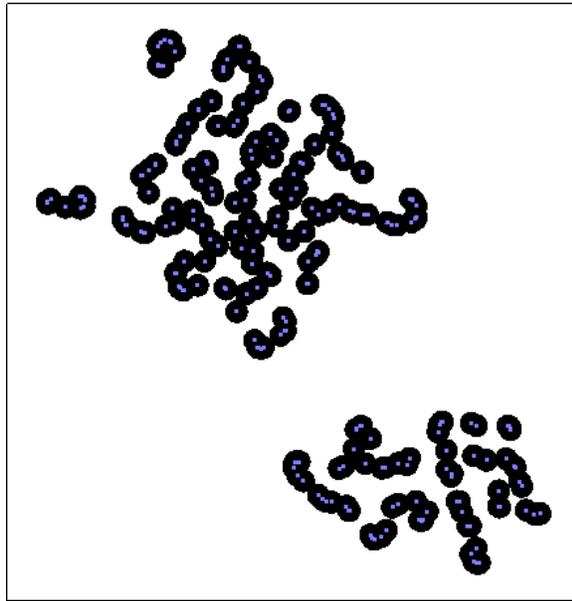


FIG. 4.6 – Agrégation de particules sous l’effet de la relaxation de simple densité (sans gravité).

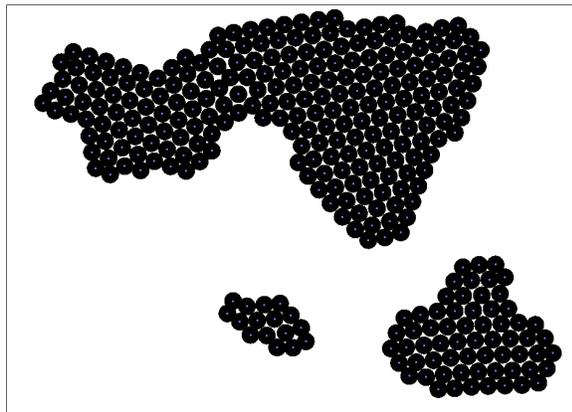


FIG. 4.7 – Artefact d’empilement se produisant si on utilise une simple force de répulsion pour empêcher l’agrégation.

ne résiste pas aux déformations autres qu’aux changements de volume). Il faut s’efforcer d’empêcher l’agrégation en appliquant une force plus douce qu’une simple répulsion dépendante de la distance.

Notre solution à ce problème est l’une des contributions principales de notre travail. L’agrégation des particules se produisant lors de la relaxation de densité est combattue à l’aide d’une seconde relaxation de densité. Cette seconde relaxation est effectuée en même temps que la première, en calculant une seconde densité et en ajoutant un terme à l’expression du déplacement (équation 4.3).

Cette nouvelle densité quantifie le nombre de voisines rapprochées de la particule. On calcule cette densité de proximité similairement à la densité ordinaire, mais avec un noyau de densité plus pointu. Notre définition de la densité de proximité est

$$\rho_i^{\text{prox}} = \sum_{j \in N(i)} (1 - r_{ij}/h)^3. \quad (4.4)$$

La seule différence avec la densité ordinaire est que nous utilisons une fonction cubique à la place d'une quadratique pour pondérer les contributions des voisines. La Figure 4.8 compare les deux noyaux de densité.

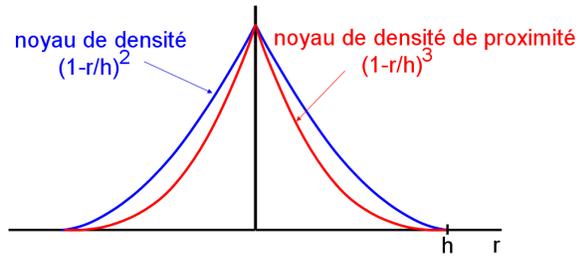


FIG. 4.8 – Noyaux de densité.

La densité de proximité contrôle la force de répulsion par l'intermédiaire d'une pression de proximité. Cette pression est définie similairement à la pression ordinaire (équation 4.2), mais avec une densité de repos nulle. On a donc l'équation

$$P_i^{\text{prox}} = k^{\text{prox}} \rho_i^{\text{prox}} \quad (4.5)$$

où  $k^{\text{prox}}$  spécifie l'importance globale de la répulsion de proximité.

La force de répulsion ajoute un terme au déplacement appliqué lors de la deuxième passe sur les voisines. L'équation 4.3 est augmentée d'un nouveau terme, ce qui mène à

$$\mathbf{D}_{ij} = \Delta t^2 \left( P_i (1 - r_{ij}/h) + P_i^{\text{prox}} (1 - r_{ij}/h)^2 \right) \hat{\mathbf{r}}_{ij}. \quad (4.6)$$

La répulsion de proximité entre deux particules est proportionnelle à la pression de proximité  $P_i^{\text{prox}}$ . Elle est aussi proportionnelle au noyau  $(1 - r_{ij}/h)^2$ , qui pondère la magnitude de la répulsion en fonction de la distance entre les particules. Ce noyau est proportionnel à la dérivée du noyau de la densité de proximité. L'Algorithme 3 détaille chacune des étapes de notre relaxation de double densité.

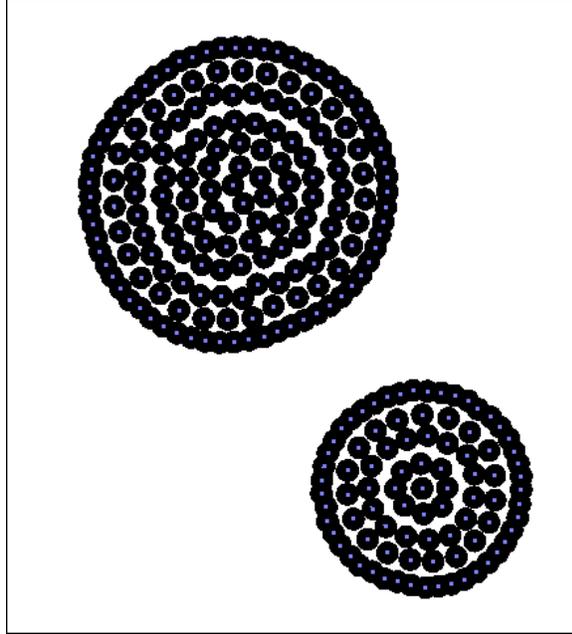


FIG. 4.9 – Formation de structures régulières sous l’effet de la relaxation de double densité.

#### 4.4 Tension de surface et courbure

La tension de surface est causée par une force d’attraction à grande distance existant entre les molécules. À l’intérieur du liquide, ces forces existent mais la somme des attractions avec les molécules voisines est nulle. Près de la surface, l’asymétrie des voisines cause une force nette vers le liquide. Plus la surface est courbée, plus l’asymétrie est importante, et plus la force résultante augmente.

Habituellement, on traite donc la tension de surface comme une force externe s’appliquant près de la surface. La force est proportionnelle à sa courbure, et dirigée selon sa normale :

$$\mathbf{F}(\mathbf{x}) = \sigma\gamma(\mathbf{x})\kappa(\mathbf{x})\hat{\mathbf{n}}(\mathbf{x}). \quad (4.7)$$

Dans cette équation,  $\sigma$  est une constante de tension de surface,  $\gamma$  est une fonction indiquant où la force s’applique (significative seulement près de la surface),  $\kappa$  est la courbure de la surface, et  $\hat{\mathbf{n}}$  est la normale. Dans le contexte d’une simulation eulérienne, il est facile d’évaluer chacun de ces composants en se basant sur les dérivées de la fonction de distance à la surface. La normale est le gradient normalisé de la distance, et la courbure est la divergence de la normale. La fonction indicatrice  $\gamma$  est simplement évaluée en utilisant une fonction cloche centrée à l’origine avec la distance à la surface

**Algorithme 3 :** Relaxation de double densité. \_\_\_\_\_

1. pour chaque particule  $i$
2.      $\rho \leftarrow 0$
3.      $\rho^{\text{prox}} \leftarrow 0$
4.     // calcule la densité et la densité de proximité
5.     pour chaque particule  $j \in \text{voisines}(i)$
6.          $q \leftarrow r_{ij}/h$
7.         if  $q < 1$
8.              $\rho \leftarrow \rho + (1 - q)^2$
9.              $\rho^{\text{prox}} \leftarrow \rho^{\text{prox}} + (1 - q)^3$
10.     // calcule la pression et la pression de proximité
11.      $P \leftarrow k(\rho - \rho_0)$
12.      $P^{\text{prox}} \leftarrow k^{\text{prox}}\rho^{\text{prox}}$
13.      $\mathbf{dx} \leftarrow 0$
14.     pour chaque particule  $j \in \text{voisines}(i)$
15.          $q \leftarrow r_{ij}/h$
16.         if  $q < 1$
17.             // applique les déplacements
18.              $\mathbf{D}_{ij} \leftarrow \Delta t^2(P(1 - q) + P^{\text{prox}}(1 - q)^2)\hat{\mathbf{r}}_{ij}$
19.              $\mathbf{x}_j \leftarrow \mathbf{x}_j + \mathbf{D}_{ij}$
20.              $\mathbf{dx} \leftarrow \mathbf{dx} - \mathbf{D}_{ij}$
21.      $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{dx}$

comme argument.

Pour une simulation à base de particules, cette même approche peut être envisagée, et c'est ce que font Müller *et al.* [MCG03] par exemple. Ils approximent la fonction de distance à la surface en se basant sur la distribution des voisines. L'idée est de considérer qu'une particule ayant moins de voisines se trouve nécessairement près de la surface. Chaque particule évalue sa densité, puis on considère que cette valeur est approximativement proportionnelle au champ de distance près de la surface. On peut donc appliquer l'opérateur gradient sur ce champ défini sur les particules, pour avoir la

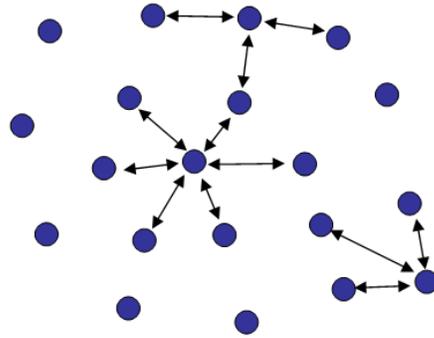


FIG. 4.10 – Forces d’attraction causant la tension de surface.

direction de la normale. Ensuite on peut dériver de nouveau ce champ de normale, ce qui mène à une approximation de la courbure.

Cette méthode de calcul de courbure pour une surface définie par des particules est problématique, en partie puisque l’hypothèse de base – les particules ont une plus faible densité près de la surface – est souvent fautive. L’algorithme d’incompressibilité tend justement à égaliser les densités des particules. Des variations plus complexes de la méthode SPH tiennent compte de la situation particulière des particules de surface et considèrent qu’elles ne doivent pas nécessairement obtenir la densité élevée des particules de l’intérieur. Pour ce faire, on peut par exemple initialiser les particules de surface différemment des autres, en leur donnant une densité de repos plus petite que les autres. Cette façon de faire nous semble difficilement applicable pour une situation où des changements de topologie extrêmes surviennent continuellement.

Même si on a une façon robuste de déterminer la distance à la surface pour chaque particule, il n’est pas évident que le calcul de tension de surface fonctionnera bien pour n’importe quelle situation. Un détail de surface formé d’un petit nombre de particules définit un champ de distance extrêmement sous-résolu, et chaque application d’opérateurs différentiels multiplie les erreurs. La dérivée d’une fonction bruitée est toujours encore plus instable.

Imaginons une amas désordonné comportant une vingtaine de particules. Même à la main, le problème consistant à assigner une normale à chaque particule est mal défini. Si on parvient néanmoins à proposer une solution, les valeurs de normales sont habituellement peu robustes, c’est-à-dire qu’elles peuvent changer dramatiquement si on change légèrement la position d’une particule. Puisque la magnitude de la force de

tension dépend des dérivées de ce champ instable et bruité, la valeur obtenue à la fin du calcul est souvent insensée.

Une façon de réduire les instabilités consiste à lisser les champs de manière à abstraire les détails définis par un trop petit nombre de particules. Les normales calculées en prenant le gradient du champ de densité sont lissées en réduisant l'écart entre une normale et la moyenne de celles des particules voisines. La valeur de la courbure peut être lissée de la même façon. Cette façon artificielle de masquer les problèmes a pour effet de produire une force qui n'a pas la même résolution que les particules. Donc un petit détail de surface (une simple bosse par exemple) ne subira pas la tension de surface nécessaire à le faire disparaître. Il serait nettement préférable de calculer une force ayant la même résolution que la représentation de base : valable mais différente pour chaque particule.

Une autre façon de réduire les problèmes de la méthode consiste à détecter les cas où le calcul de normale est mal défini. Par exemple dans le cas extrême d'une particule seule, aucune normale satisfaisante ne peut être assignée. Aussi, loin à l'intérieur du liquide, l'effet du manque de particules à la surface n'est plus visible, et les gradients de densité n'indiquent plus des directions normales valables. Ces tests de validité impliquent toutefois la spécification de seuils, ce qui réduit encore plus la robustesse de la méthode : lorsqu'une normale devient valide, la courbure change. Une fois de plus, il semble impossible de trouver une valeur stable et précise pour une force de tension de surface applicable sur des particules.

Un dernier problème pouvant gêner le calcul de tension par courbure est l'impossibilité de garantir directement la conservation de la quantité de mouvement. Puisque la tension est traitée comme une force externe s'appliquant sur les particules de la surface, on en est réduit à espérer que les forces s'appliquant d'un côté d'une goutte soient exactement compensées par celles s'appliquant de l'autre côté. Si ce n'est pas le cas, la goutte accélèrera dans une direction improbable comme si un esprit malin lui avait insufflé une volonté incongrue. On peut toutefois montrer que si le calcul de courbure est exact, et que la force est appliquée sur une épaisseur égale partout, alors les forces de tension s'égalisent et la quantité de mouvement est conservée. Croire que la somme de forces obtenues à la suite d'une série d'approximations grossières sera exactement nulle relève d'un acte de foi que certains sont prêts à honorer. Nous considérons ici

que le monde rassurant des forces locales, radiales, égales et opposées entre paires de particules est trop confortable pour être abandonné au profit d'un calcul global tentant de quantifier la deuxième dérivée d'une hypothétique *surface*.

#### 4.5 Effets de tension de surface par forces locales

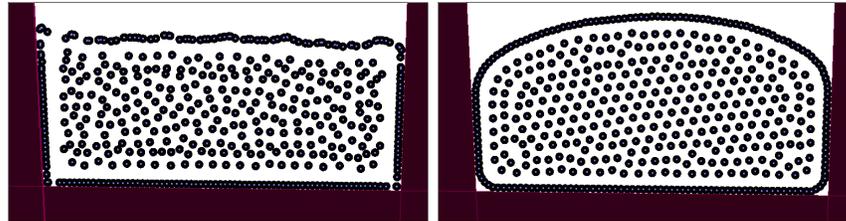


FIG. 4.11 – Liquide sous l'effet de la gravité dans un contenant. À droite, le liquide subit la relaxation de proximité. On remarque la courbure caractéristique de la surface.

Certains scientifiques considèreront que nous avons été *serendipitous* en développant notre méthode de tension de surface. Être *serendipitous* (mot anglais sans traduction) consiste à être chanceux en faisant une découverte fortuite. En effet, nous avons observé qu'un liquide subissant la relaxation de double densité adopte un comportement ressemblant étrangement à un liquide sous l'influence d'une tension de surface (Figure 4.11). La surface tend à diminuer sa courbure ; les gouttes deviennent sphériques ; deux gouttes se touchant se fusionnent rapidement et oscillent quelques instants ; un filament de liquide s'étire graduellement puis se sépare soudainement en une série de petites gouttes bien rondes. La Figure 4.12 montre une goutte oscillante simulée avec notre système. Ces comportements relèvent directement de la tension de surface, toutefois aucun calcul de courbure n'apparaît dans notre code.

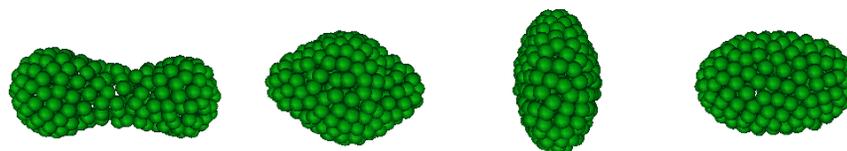


FIG. 4.12 – Goutte oscillante.

Un effet particulièrement subtil de la tension de surface est la formation des membranes composant les bulles et la mousse. La structure particulière d'un complexe formé

de plusieurs bulles de savon est habituellement étudiée en tant que problème d'optimisation. En effet, on peut montrer que pour de minces films, la tension de surface tend à minimiser l'aire totale de la membrane composant la mousse. En 2D, une membrane correspond à un filament. La Figure 4.13 montre la configuration de repos d'un ensemble de particules ne subissant aucune gravité. Elles collent à quatre objets selon la force d'adhérence discutée à la Section 6.3. L'angle formé à l'intersection de trois filaments est d'environ 120 degrés, et la longueur totale des filaments semble minimale, ce qui correspond à la théorie des films liquides. À notre connaissance, c'est la première fois que ce type de comportement est obtenu avec une simulation par particules.

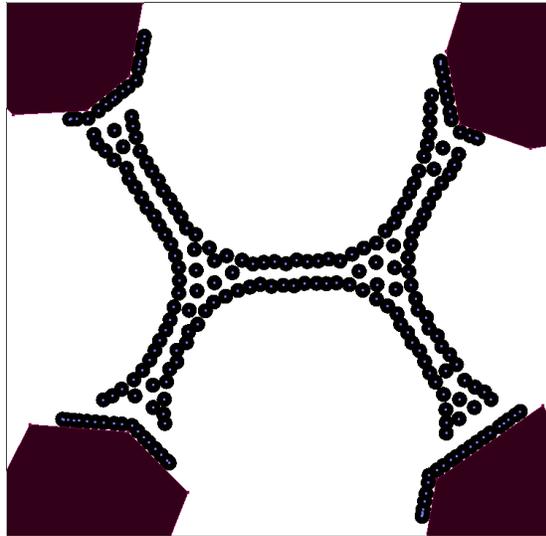


FIG. 4.13 – Filaments se formant naturellement avec la relaxation de double densité.

Il semble difficile de prouver que la relaxation de double densité mène bien à une force proportionnelle à la courbure. Toutefois, sachant que la tension de surface est ultimement causée par une attraction à grande distance, on peut se demander si notre méthode cause effectivement une attraction entre les particules. C'est le cas : une force d'attraction indirecte émerge effectivement du processus de double relaxation.

Pour visualiser cette attraction, il faut se rappeler ce que fait la double relaxation : elle pousse ou tire les voisins selon le noyau linéaire pour atteindre la densité de repos, puis pousse les voisins selon le noyau quadratique pour diminuer la densité de proximité. Supposons que notre particule a une densité trop faible. Les voisins seront alors tirés *linéairement* pour augmenter la densité, et simultanément poussés *quadratiquement* pour diminuer la densité de proximité. Puisque la répulsion quadratique n'a que peu d'ef-

fet sur les particules éloignées, elles seront plus attirées que les particules rapprochées. Même si l'attraction linéaire attire plus les particules rapprochées, la répulsion quadratique l'annule. L'effet net est une attraction des particules éloignées et une répulsion des particules rapprochées.

Il est intéressant de comparer notre méthode avec l'application d'une simple force dépendante de la distance, par exemple la force de Lennard-Jones. Cette force est répulsive à courte distance, attractive à distance moyenne, et nulle à grande distance. Le parallèle avec notre méthode est flagrant : c'est exactement ce qu'on prétend obtenir. Pourtant, nous avançons que notre méthode est fondamentalement différente de l'application de forces qui dépendent exclusivement de la distance. Deux particules subissant une force de Lennard-Jones semblent avoir un bâton entre elles. Les particules s'accrochent et s'empilent comme des balles rigides. Dans notre cas, la force est plus *gentille*, puisqu'elle dépend du contexte. Elle s'adapte à la situation pour atteindre son but de façon indirecte.

# Chapitre 5

## Viscoélasticité

### 5.1 Viscosité

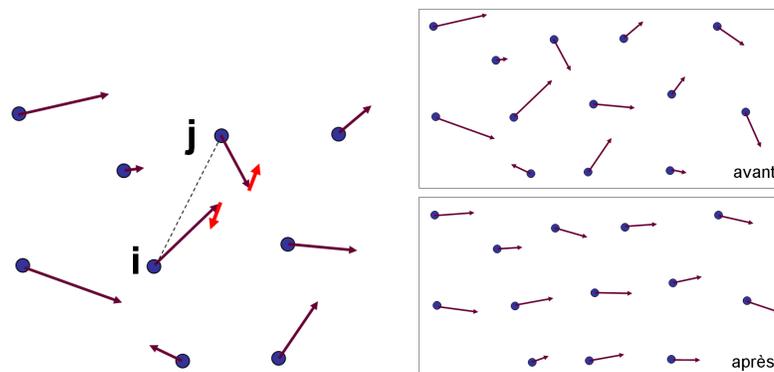


FIG. 5.1 – Impulsion de viscosité entre une paire de particules (gauche). Effet lissant de la viscosité (droite).

La viscosité a pour effet de lisser le champ de vitesse. Elle s'applique sous forme d'impulsions radiales entre paires de particules (Figure 5.1). Ces impulsions modifient les vitesses au début du pas de temps, avant que les particules n'avancent à leurs positions prédites.

L'Algorithme 4 montre comment la viscosité affecte les vitesses. On calcule la vitesse de rapprochement radiale en projetant la composante de différence de vitesses sur l'axe entre les deux particules (ligne 5 et Figure 5.2), puis on applique l'impulsion si les particules avancent l'une vers l'autre. On ne met pas de viscosité entre deux particules qui s'éloignent, puisque le but est surtout d'empêcher que des particules passent au

Algorithme 4 : Impulsions de viscosité.

1. pour chaque paire de voisines  $ij$
2.      $q \leftarrow r_{ij}/h$
3.     si  $q < 1$
4.         // vitesse de rapprochement
5.          $u \leftarrow (\mathbf{v}_i - \mathbf{v}_j) \cdot \hat{\mathbf{r}}_{ij}$
6.         si  $u > 0$
7.             // impulsion linéaire et quadratique
8.              $\mathbf{I} \leftarrow \Delta t(1 - q)(\sigma u + \beta u^2)\hat{\mathbf{r}}_{ij}$
9.              $\mathbf{v}_i \leftarrow \mathbf{v}_i - \mathbf{I}$
10.             $\mathbf{v}_j \leftarrow \mathbf{v}_j + \mathbf{I}$

travers l'une de l'autre. Pour une viscosité élevée, on pourrait aussi appliquer une viscosité entre deux particules s'éloignant, mais ça peut être dangereux puisqu'on rendrait la contrainte d'anti-agrégation plus difficile à satisfaire (la répulsion serait en quelque sorte contredite par la viscosité).

La dépendance de l'impulsion de viscosité par rapport à la distance entre les particules est capturée par un noyau de distance. On choisit simplement le noyau linéaire  $(1 - r/h)$ . La dépendance par rapport à la vitesse est plus complexe : on a un terme linéaire et un terme quadratique. Les paramètres  $\sigma$  et  $\beta$  ajustent les contributions des deux termes. Cette formulation non-linéaire de la viscosité s'inspire d'une formulation commune en SPH (voir par exemple [DG96]).

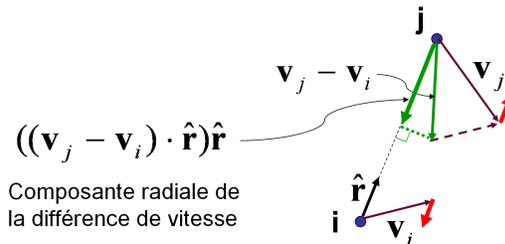


FIG. 5.2 – Calcul de l'impulsion de viscosité.

Si une viscosité élevée est désirée, on prend une valeur de  $\sigma$  élevée. Ainsi, les vitesses

relatives radiales diminuent drastiquement. Si un fluide à basse viscosité est désiré, on pourrait tout simplement ne pas traiter de viscosité. Par contre, sans viscosité, les particules peuvent passer au travers l'une de l'autre sans s'influencer. C'est pourquoi, même si on veut le liquide le moins visqueux possible, on doit ajouter une force empêchant les interpénétrations. La viscosité quadratique donne une impulsion proportionnelle à la vitesse au carré. Ceci permet aux petites vitesses de ne pas être trop affectées, mais les grandes vitesses de rapprochement sont fortement réduites. Pour un scénario tel qu'une sphère de liquide non-visqueux tombant dans une piscine, seulement le coefficient de viscosité quadratique  $\beta$  est donc fixé à une valeur non-nulle.

## 5.2 Elasticité

**Algorithme 5** : Déplacements dus aux ressorts. 

---

1. pour chaque ressort  $ij$
  2.      $\mathbf{D} \leftarrow \Delta t^2 k^{\text{ressort}} (L_{ij} - r_{ij})(1 - L_{ij}/h) \hat{\mathbf{r}}_{ij}$
  3.      $\mathbf{x}_i \leftarrow \mathbf{x}_i - \mathbf{D}$
  4.      $\mathbf{x}_j \leftarrow \mathbf{x}_j + \mathbf{D}$
- 

Nous simulons l'élasticité en ajoutant dynamiquement des ressorts entre les particules. Ces ressorts appliquent une impulsion radiale égale et opposée aux deux particules attachées. Selon notre modèle, un ressort de longueur de repos  $L$  et de longueur courante  $r$  génère une impulsion proportionnelle à  $(L - r)(1 - L/h)$ . Le premier facteur correspond à la compression du ressort, indiquant que la force dépend linéairement de la déformation (modèle de *Hooke*). Le deuxième facteur diminue l'effet des longs ressorts. Un ressort dont la longueur de repos correspond au rayon d'interaction n'exerce aucune force. La force augmente linéairement à mesure que la longueur au repos diminue. Ceci assure que l'interaction reste locale, et qu'elle diminue graduellement avec la distance. Les ressorts appliquent leurs forces sur les particules en déplaçant séquentiellement leurs positions prédites (Algorithme 5).

Malgré son étonnante simplicité, ce simple système de ressorts rend possible le scénario de simulation illustré à la Figure 5.3, où un moule est rempli de particules,

on ajoute les ressorts, puis on retire le moule. L'objet élastique résultant peut être manipulé et déformé : il reviendra à sa forme de repos.



FIG. 5.3 – Moulage d'un personnage élastique.

### 5.3 Plasticité

Les forces internes d'une substance élastique combattent les forces externes telles la gravité dans l'espoir de retrouver sa forme de repos. Si aucune plasticité n'est présente, la forme de repos ne change pas. La plasticité peut être définie comme la faculté qu'ont les matériaux de subir des déformations permanentes.

Cette vision intuitive de la plasticité suggère de la traiter en changeant les longueurs de repos des ressorts. À chaque pas de temps, ces dernières sont modifiées en fonction des longueurs courantes (Figure 5.4). Les ressorts sont ajoutés et enlevés dynamiquement à mesure que la simulation progresse dans le temps.

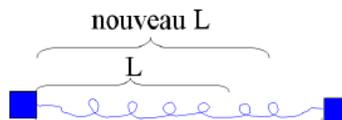


FIG. 5.4 – Changement de longueur de repos d'un ressort.

Le taux de changement de la longueur d'un ressort à plasticité linéaire est

$$\Delta L = \Delta t \alpha (r - L) \tag{5.1}$$

où  $\alpha$  est le coefficient de plasticité linéaire. Lorsque  $\alpha$  est égal à  $1/\Delta t$ , la longueur de repos d'un ressort devient instantanément la longueur courante.

Une substance à plasticité linéaire coule lentement jusqu'à ce que toutes les forces atteignent l'équilibre. Pour modéliser une substance comme l'argile, qui se déforme sous la pression des doigts mais qui résiste aux plus petites forces telles que la gravité, on a besoin d'un modèle de plasticité non-linéaire.

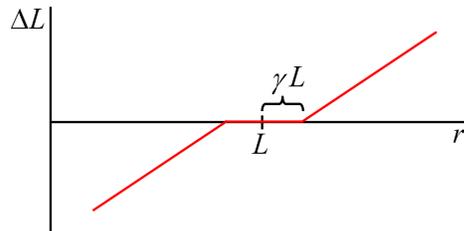


FIG. 5.5 – Changement de longueur de repos en fonction de la longueur courante.

Notre modèle de plasticité est inspiré du critère de *von Mises*, qui stipule intuitivement qu'une déformation devient permanente seulement si elle est assez grande. Traduite dans le système de ressorts, la condition se réduit à dire qu'un changement de longueur de repos survient si le ressort est allongé ou compressé de plus d'un certain pourcentage de sa longueur. On appelle ce pourcentage le seuil de plasticité  $\gamma$ . La modification de longueur peut alors s'écrire

$$\Delta L = \Delta t \alpha \text{sign}(r - L) \max(0, |r - L| - \gamma L) \quad (5.2)$$

qui devient identique à l'équation 5.1 lorsque  $\gamma = 0$ . Ce type de plasticité permet de simuler les détails persistants de substances telles que la boue. À la Figure 5.6, on voit un objet imprimant une déformation permanente à une boule de matériel viscoélastique. Ce comportement ne serait pas possible avec une plasticité exclusivement linéaire.

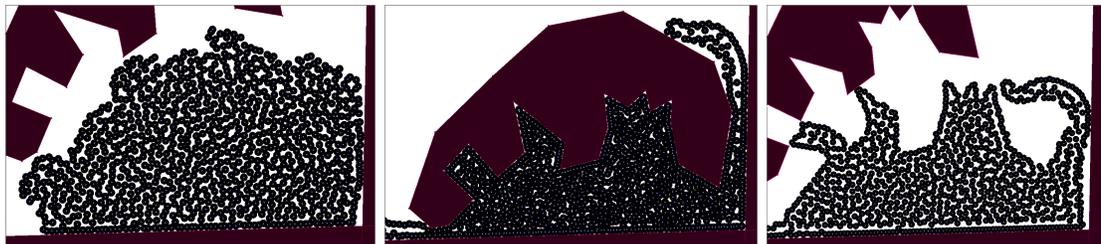


FIG. 5.6 – Déformation permanente d'une boule de matériel viscoélastique.

## Chapitre 6

# Interactions avec des objets

Pouvoir simuler l'interaction entre un fluide et des objets en mouvement permet une multitude de scénarios d'animation. Dans ce travail, nous traitons non seulement les effets du mouvement des objets sur le mouvement du fluide, mais aussi l'inverse, c'est-à-dire l'effet du fluide sur les objets. Ce couplage à deux directions permet d'animer par exemple un jet de liquide poussant un objet ou un objet flottant.

Contrôler adéquatement les conditions aux frontières d'une simulation de liquide est reconnu comme un problème difficile. Dans le cas d'une simulation eulérienne, même un simple mur immobile complexifie l'implémentation dramatiquement. Souvent, la majeure partie du code ne sert qu'à contraindre le champ de vitesse de façon à ce que le liquide ne passe pas à travers les parois d'un récipient.

Spécifier les conditions aux frontières consiste souvent à s'assurer que la composante normale de la vitesse est nulle le long d'un mur. Ainsi, le liquide ne le pénètre pas (théoriquement). Si on veut empêcher le liquide de glisser, on annule aussi la composante tangentielle, et la vitesse s'annule complètement au mur. Le problème avec les conditions aux frontières d'une simulation par grille est que les vitesses sont discrétisées à des endroits réguliers de l'espace, et qu'un mur n'est en général pas aligné sur la grille. Simplement contraindre les vitesses autour du mur à respecter la condition donne de très mauvais résultats, puisque ceci revient à faire une erreur équivalente à la taille d'une cellule dans la position du mur observée par le fluide. Un jet de liquide interagissant avec un objet irrégulier en mouvement est une situation très difficile à gérer si tout ce qu'on peut faire est contrôler la vitesse aux points fixes de la grille. Si on veut en plus que le fluide influence le mouvement des objets, la situation devient extrêmement complexe.

Empêcher les pertes de masse et les artefacts de grille autour d'un objet irrégulier en mouvement est hautement non-trivial.

Étant données les complications introduites par les conditions aux frontières dans le cadre de la simulation eulérienne, vient maintenant le temps d'avancer une conjecture : avec des particules c'est plus facile.

Le traitement des interactions fluide-objets considère que le fluide est simplement un ensemble de sphères rigides entrant en collision avec les objets. Les collisions sont détectées en utilisant un champ de distance pour chaque objet. En précalcul, ce champ de distance est échantillonné sur une grille régulière positionnée dans le système de coordonnées de l'objet. Pour chaque particule, on obtient sa distance signée (négative à l'intérieur) en interpolant trilinéairement entre les valeurs trouvées aux huit coins du cube où se trouve la particule. La normale à la surface est obtenue à partir du gradient du champ de distance, calculé par différences finies.

## 6.1 Calcul du champ de distance

Échantillonner un champ de distance pour un objet polygonal n'est pas aussi simple qu'il n'y paraît, surtout si le maillage est *non-manifold* ou contient des polygones inutiles à l'intérieur. Ceci n'est pas une contribution importante de ce travail, mais puisque la méthode que nous utilisons est nouvelle (à notre connaissance), nous la décrivons brièvement.

Il est important que le champ de distance soit consistant à l'intérieur de l'objet, puisque les particules s'en serviront pour en sortir. Les polygones inutiles se trouvant à l'intérieur de l'objet (par exemple dans les articulations pour les objets composés) compromettent la validité du champ de distance.

Notre méthode ignore les polygones intérieurs en identifiant de façon robuste la région extérieure, puis en marchant vers l'intérieur à partir des cellules valides. Ainsi, il suffit que la surface extérieure du modèle géométrique soit bien formée et sans trous pour que l'algorithme donne un bon résultat. Notre échantillonnage comporte quatre étapes :

1. obtenir la distance sur les points de grille près de l'objet,
2. balayer le reste du domaine à partir de ces points,

3. identifier les points extérieurs en balayant vers l'intérieur à partir d'un point qu'on sait extérieur, et finalement,
4. effacer ce qui n'est pas identifié comme extérieur et le recalculer en balayant à partir des cellules extérieures.

L'important ici n'est pas la précision mais bien la robustesse. Ce qui importe vraiment est de pouvoir trouver une distance et une direction permettant aux particules qui intersectent l'objet d'en sortir.

## 6.2 Contact et friction

Nous avons intégré notre simulateur de fluide avec un simulateur de corps rigides, de façon à pouvoir les coupler ensemble sans restrictions. Notre simulateur de corps rigides est basé sur les impulsions, comme dans le travail de Guendelman *et al.* [GBF03]. Dans ce type de système, les interpénétrations sont résolues séquentiellement entre chaque paire d'objets en leur appliquant des impulsions. Étant donnée une paire d'objets qui seront interpénétrés à la fin du pas de temps si on ne fait rien, on calcule et applique une impulsion qui annule exactement leur vitesse de pénétration au point de contact. Ensuite, on passe à la paire suivante. Les traitements subséquents peuvent défaire ce qu'on vient de contraindre, et donc chaque paire doit possiblement être traitée plusieurs fois.

Au départ, nous avons tenté d'appliquer la même stratégie pour les contacts particule-objet, mais cette méthode ne fonctionne pas très bien. La raison pour laquelle l'application séquentielle d'impulsions entre particules et objets fonctionne mal réside dans le fait qu'une particule a typiquement une masse beaucoup plus petite qu'un corps rigide. Dans le cas d'un objet flottant sur une grande quantité de particules, l'ordre de traitement des particules sous l'objet introduit un biais trop important. La particule traitée en premier reçoit une impulsion beaucoup plus grande que celle reçue par la dernière particule traitée. L'artefact observé est une vibration des objets flottants et on voit clairement certaines particules sous l'objet se faire éjecter par de trop grandes impulsions.

Cette situation, où un objet est supporté par des dizaines de particules, est difficile puisque le mouvement final de l'objet dépend de la somme des influences des particules. Relaxer chaque paire particule-objet individuellement ne fonctionne pas puisque

la première particule traitée ne peut prévoir que d'autres particules l'aideront à pousser l'objet. Alors elle pense qu'elle ne pourra pas résister au mouvement de l'objet et elle se tasse rapidement. Ensuite, les autres paires particule-objet sont traitées et l'objet finit par ralentir. Mais il est déjà trop tard, la première particule s'étant déplacée beaucoup plus que nécessaire.

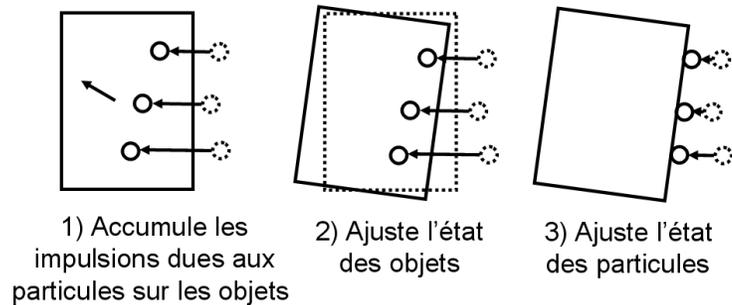


FIG. 6.1 – Interactions entre particules et objets.

Notre solution à ce problème est de procéder par étapes globales plutôt que de faire une boucle séquentielle d'impulsions locales. On sépare le traitement des corps rigides en trois étapes (Figure 6.1) :

1. Trouver l'effet que les particules ont sur les objets.
2. Modifier les vitesses et les positions finales des objets en tenant compte de cet effet. Traiter les collisions entre les objets.
3. Modifier les vitesses et positions finales des particules.

L'Algorithme 7 détaille le processus. Premièrement, on avance les objets à leurs positions prédites et on trouve l'influence globale qu'ont les particules sur eux. Pour ce faire, les particules ajoutent des impulsions de collision dans les tampons de force et de torque des objets. On ne bouge pas les particules ni les objets à cette étape ; le but est uniquement d'estimer l'effet global des particules sur les objets.

Ensuite, on replace les objets à leur position de départ, puis on les avance de nouveau mais cette fois en tenant compte des impulsions produites par les particules. Pour ce faire, on ajoute aux vitesses linéaires et angulaires les forces linéaires et angulaires accumulées lors de l'étape précédente. Maintenant que les objets sont à une position finale tenant compte de l'effet des particules, on traite les collisions entre les objets. Après cette étape, les objets ont trouvé leur position finale, et ils ne bougeront plus.

La troisième et dernière étape bouge les particules. Chaque particule pénétrant un objet subit une impulsion comme si l'objet avait une vitesse fixe. La particule prend toute l'impulsion nécessaire pour annuler sa vitesse de pénétration.

**Algorithme 7 :** Interactions particules-objets. \_\_\_\_\_

1. pour chaque *objet*
  2.     *sauvegarde la position et l'orientation originales de l'objet*
  3.     *avance l'objet en utilisant  $\mathbf{V}$  et  $\omega$*
  4.     *initialise les tampons de force et de torque*
  5.     pour chaque *particule à l'intérieur d'un objet*
  6.         *calcule l'impulsion de collision  $\mathbf{I}$*
  7.         *ajoute  $\mathbf{I}$  à la force et au torque*
  8. pour chaque *objet*
  9.     *modifie  $\mathbf{V}$  avec la force et  $\omega$  avec le torque*
  10.    *avance à partir de la position originale en utilisant les nouveaux  $\mathbf{V}$  et  $\omega$*
  11. *traite les collisions et contacts entre les objets*
  12. pour chaque *particule dans un objet*
  13.     *calcule l'impulsion de collision  $\mathbf{I}$*
  14.     *applique  $\mathbf{I}$  à la particule*
  15.     *extraie la particule de l'objet si elle est encore à l'intérieur*
- 

Les impulsions échangées entre les particules et les objets sont basées sur un modèle de collisions sans restitution. Une particule ne rebondit pas sur un objet. L'impulsion comporte aussi une composante tangentielle, de façon à ralentir le glissement de la particule sur l'objet.

Lors du traitement des collisions, les positions des particules ont déjà été relaxées, et donc la vitesse courante  $\mathbf{v}_i$  doit être recalculée en soustrayant la position précédente de la position courante. Ensuite, on obtient la vitesse relative de la particule  $\bar{\mathbf{v}} = \mathbf{v}_i - \mathbf{v}_p$ , où  $\mathbf{v}_p$  est la vitesse de l'objet au point de contact. Cette vitesse est séparée en ses composants normal et tangentiel :

$$\bar{\mathbf{v}}^{\text{normal}} = (\bar{\mathbf{v}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} \quad (6.1)$$

$$\bar{\mathbf{v}}^{\text{tangent}} = \bar{\mathbf{v}} - \bar{\mathbf{v}}^{\text{normal}}. \quad (6.2)$$

L'impulsion calculée aux lignes 6 et 13 annule complètement la vitesse normale et enlève une partie de la vitesse tangentielle :

$$\mathbf{I} = \bar{\mathbf{v}}^{\text{normal}} - \mu \bar{\mathbf{v}}^{\text{tangent}} \quad (6.3)$$

où  $\mu$  est un paramètre de friction permettant une condition aux frontières avec glissement ( $\mu = 0$ ) ou sans glissement ( $\mu = 1$ ).

Dans certaines situations difficiles, comme par exemple lorsqu'une particule est dans un objet au début du pas de temps, annuler la vitesse de pénétration n'est pas suffisant. Nous réglons simplement ce problème en sortant directement les particules qui sont toujours dans l'objet à la fin du pas de temps. Il est facile de trouver le déplacement à appliquer à la particule en évaluant la distance et la normale à la position de la particule.

Le traitement de collision par champ de distance a une limitation importante. On ne teste pas vraiment les intersections des objets avec les trajectoires des particules, mais seulement avec les positions à la fin du pas de temps. Il est donc possible de manquer des collisions si les particules vont trop vite par rapport à l'épaisseur des objets. Dans nos exemples, les objets sont assez épais, et les fuites de particules sont rares.

### 6.3 Adhérence

L'interaction entre un liquide réel et une surface est plus complexe qu'un simple contact avec friction. Le liquide peut aussi coller sur l'objet, parfois même en dessous d'une surface horizontale. La physique des interactions fluide-solide est très complexe, et ici nous nous limitons à un traitement naïf de l'adhérence. Néanmoins, nous parvenons à faire coller les particules sur les objets (Figure 6.2).

L'idée est simplement d'ajouter une force d'attraction entre les objets et les particules environnantes. Soit  $d_i$  la distance entre l'objet et la surface de la particule (de rayon  $R$ ). Une impulsion d'attraction est échangée lorsque  $d_i$  est inférieur à une certaine distance  $d^{\text{adhérence}}$ . Pour éviter les artefacts visibles tels qu'une particule qui *saute* sur l'objet,  $d^{\text{adhérence}}$  doit rester petite comparée au rayon d'interaction  $h$ .

La force d'attraction que nous utilisons a la forme d'une parabole inversée commençant à zéro à la surface de l'objet, atteignant son maximum à  $d^{\text{adhérence}}/2$ , puis

retournant à zéro lorsque  $d_i = d^{\text{adhérence}}$ . Voici l'équation :

$$\mathbf{I}^{\text{adhérence}} = -\Delta t k^{\text{adhérence}} d_i \left(1 - \frac{d_i}{d^{\text{adhérence}}}\right) \hat{\mathbf{n}}. \quad (6.4)$$

Cette force est calculée et appliquée en même temps que l'impulsion de collision, aux lignes 6 et 13 de l'Algorithme 7. Cet algorithme doit être modifié pour tenir compte des particules qui sont dans le rayon d'adhérence.

Le couplage de l'adhérence avec la tension de surface permet d'obtenir la forme caractéristique de la surface de l'eau dans un verre (Figure 6.3).

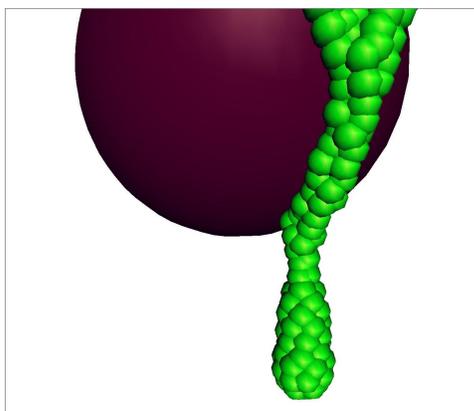


FIG. 6.2 – Particules adhérant sous une sphère avant de former une goutte.

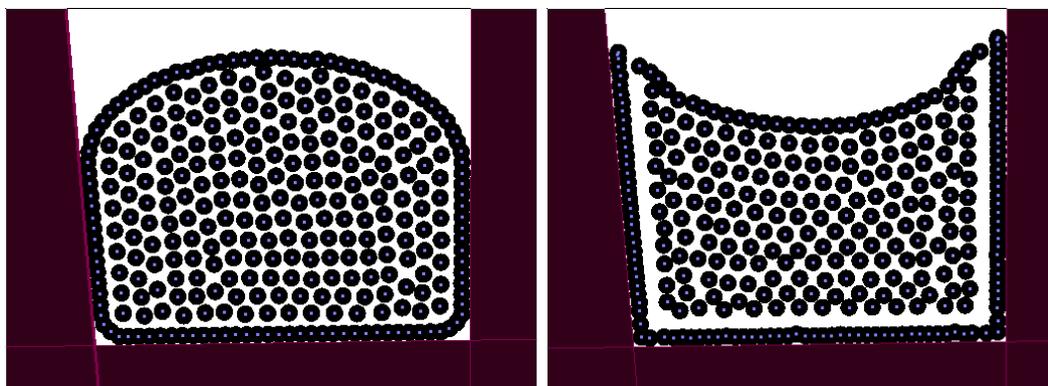


FIG. 6.3 – Effet de l'adhérence sur la forme de la surface. On remarque la formation du ménisque caractéristique à droite.

# Chapitre 7

## Implémentation et résultats

Dans ce chapitre nous traitons d'abord de deux éléments importants de l'implémentation, le hachage spatial et l'extraction de surface, puis nous discutons des résultats obtenus avec notre méthode.

### 7.1 Hachage spatial

Au cours de la simulation, on a souvent besoin d'accéder aux voisines d'une particule. Il est donc primordial que la recherche de ces voisines se fasse rapidement. L'algorithme naïf obtient la liste des paires de voisines en parcourant complètement la liste pour chaque particule. On garde seulement les voisines qui sont à une distance inférieure au rayon d'interaction  $h$ . Cette méthode est de complexité  $O(n^2)$ , ce qui est beaucoup plus long que nécessaire.

Puisque le rayon d'interaction est constant et identique pour chaque particule, il est possible d'utiliser une grille de recherche spatiale. Chaque cube de la grille stocke la liste de particules qu'il contient. La taille d'un cube est égale à  $h$ . Ainsi, les voisines d'une particule se trouvent exclusivement dans le cube qui la contient et dans les 26 cubes voisins.

Nous évitons d'utiliser une grille fixe en plaçant les cubes dans une table de hachage (Figure 7.1). On instancie seulement les cubes qui contiennent des particules. Lorsqu'un cube se vide de ses particules, on le retire de la table et on l'efface. Le code de hachage d'un cube est calculé à partir de son indice 3D, comme dans Teschner *et al.* [THM<sup>+</sup>03]. Ainsi, le scénario de simulation existe dans un monde virtuellement infini, et on n'a pas

besoin de connaître les positions futures du fluide au début de la simulation.

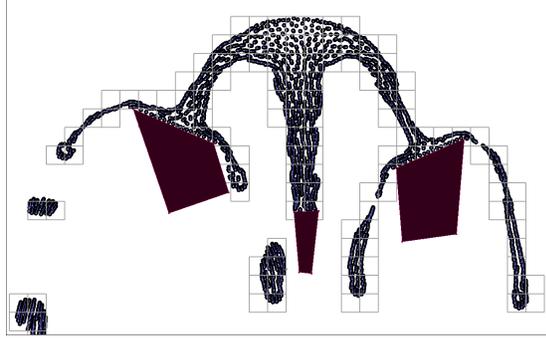


FIG. 7.1 – Structure de recherche de voisins en 2D. Les cellules nécessaires sont créés dynamiquement et placées dans une table de hachage.

## 7.2 Extraction de surface

La surface du liquide est construite en polygonisant une isosurface d’une fonction implicite définie par les particules. La fonction que nous utilisons est

$$\phi(\mathbf{x}) = \left( \sum_{N(i)} (1 - r/h)^2 \right)^{\frac{1}{2}} \quad (7.1)$$

où  $r = |\mathbf{x} - \mathbf{x}_j|$ . La racine carrée assure que la fonction se comporte comme une fonction de distance, c’est-à-dire qu’elle a une pente constante autour de l’isosurface extraite. Ceci est important pour les isosurfaces en mouvement puisque sa position dans un cube est déterminée linéairement entre les valeurs aux coins. Si la fonction n’est pas approximativement régulière autour de l’isosurface, des artefacts temporels apparaissent alors que la surface bouge. Utiliser une simple somme de cones  $(1 - r/h)$  atteindrait ce même objectif, mais la surface obtenue ne serait pas assez lisse.

Nous utilisons une variante de l’algorithme du *marching cube* [LC87] pour polygoniser la surface. Puisque la simulation ne se produit pas dans un domaine fixe et connu d’avance, il est impossible d’utiliser une grille de reconstruction fixe. Similairement à ce que nous faisons pour la recherche de voisines, nous plaçons les cubes de reconstruction dans une table de hachage, et seulement les cubes dont au moins un coin a une valeur non-nulle existent.

Les particules accumulent leurs contributions à la fonction aux coins des cubes environnants, on prend la racine carrée, et finalement on triangularise l’isosurface dans

les cubes touchés. La grandeur d'un cube de *marching cube* est indépendante du rayon d'interaction  $h$ , et est typiquement plus petite.

### 7.3 Résultats

Nous avons implémenté notre technique en 2D et en 3D, avec le langage C++ et l'API graphique *OpenGL*. La version 2D permet de créer interactivement des objets polygonaux et de les déplacer avec la souris. On ajoute des particules directement avec la souris ou par l'intermédiaire d'un objet source. Cette source projette les particules avec une certaine vitesse, ce qui permet de faire une fontaine.

En 3D, on peut afficher les particules sous forme de sphères polygonisées pour une visualisation rapide. On peut aussi afficher les triangles de la surface extraite directement avec OpenGL. Pour un résultat de haute qualité, on sauvegarde le maillage de la surface pour chaque image, puis on le rend avec le logiciel de lancer de rayons *Pixie* [PIX].

Les temps de calcul sont grossièrement linéairement proportionnels au nombre de particules simulées. Une simulation interactive à 10 images par seconde traite environ 1000 particules, ce qui est suffisant pour créer des effets relativement complexes. Les résultats présentés dans les séquences vidéo [VID] utilisent typiquement 20000 particules et demandent environ 2 secondes de calcul par image (le film joue à 30 images par seconde). Ces temps de calcul incluent l'extraction de surface, qui peut prendre jusqu'à 50% du temps. Il est à noter que notre implémentation pourrait être considérablement optimisée, par exemple en améliorant la cohérence de la mémoire.

Notre méthode est spécifiquement adaptée aux longs pas de temps, ce qui permet de faire un seul pas de simulation par image dans la majorité des cas. Pourtant, pour des situations extrêmes, où par exemple un jet de liquide très rapide entre en collision avec un objet, il peut être nécessaire de calculer plusieurs pas de temps par image (jusqu'à 10 dans nos exemples).

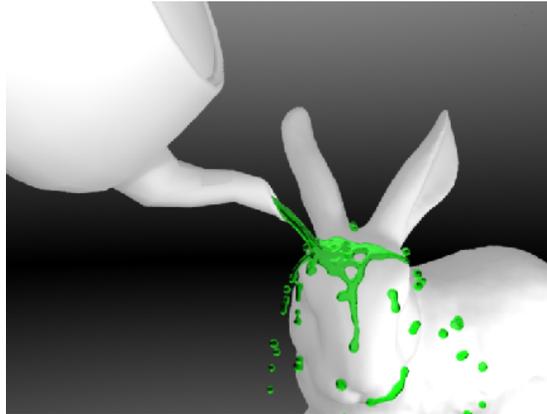


FIG. 7.2 – Liquide tombant sur le *Bunny* de Stanford. Les particules sont contraintes par la géométrie du bec de la théière avant de couler sur le lapin en formant des gouttes et des filaments.

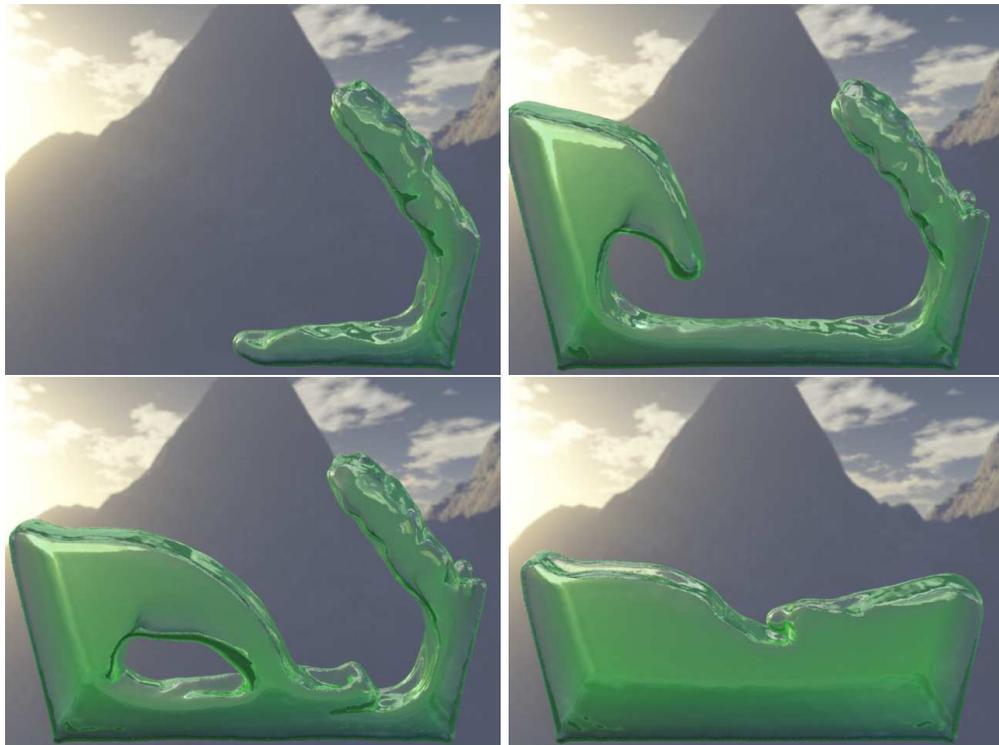


FIG. 7.3 – Une vague se replie sur elle-même alors qu'on remplit un contenant rectangulaire d'un liquide visqueux.

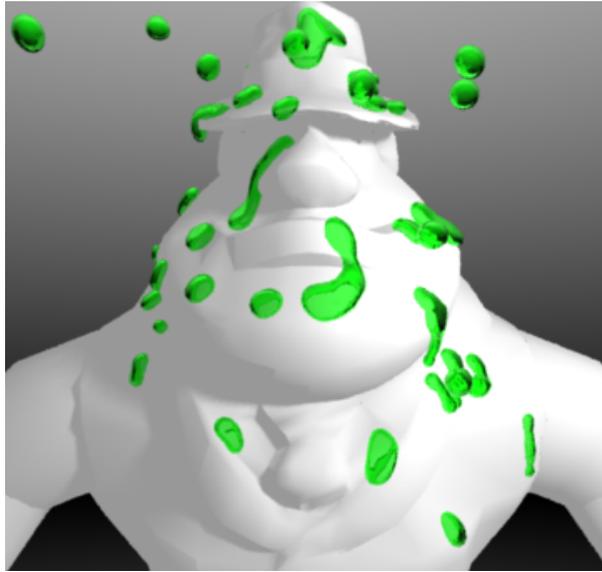


FIG. 7.4 – Pluie visqueuse sur un personnage. Chaque goutte de pluie est formée d'environ 20-40 particules. Les gouttes se divisent et s'unissent alors qu'elles coulent sur la surface du personnage.

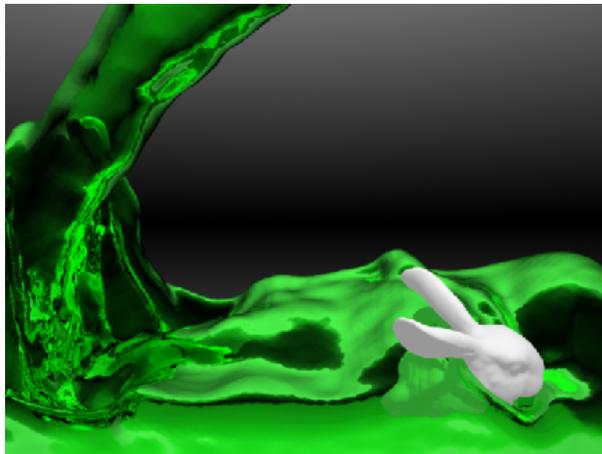


FIG. 7.5 – Liquide remplissant un bac contenant un lapin relativement lourd. Cet exemple illustre le couplage bidirectionnel liquide/objets : le liquide réagit à la présence du lapin en le contournant, et le lapin réagit à la présence du liquide en se déplaçant.

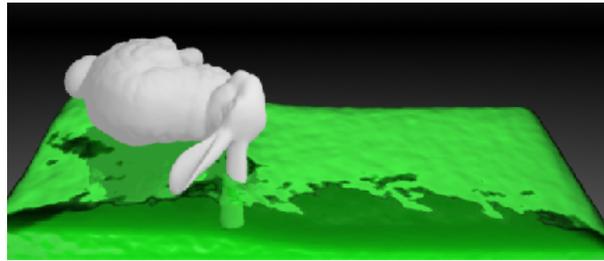


FIG. 7.6 – Le même lapin, mais beaucoup plus léger, flotte sur le liquide.



FIG. 7.7 – Le *buddha* sous une douche d'or liquide. Le liquide se sépare en de longs filaments visqueux.

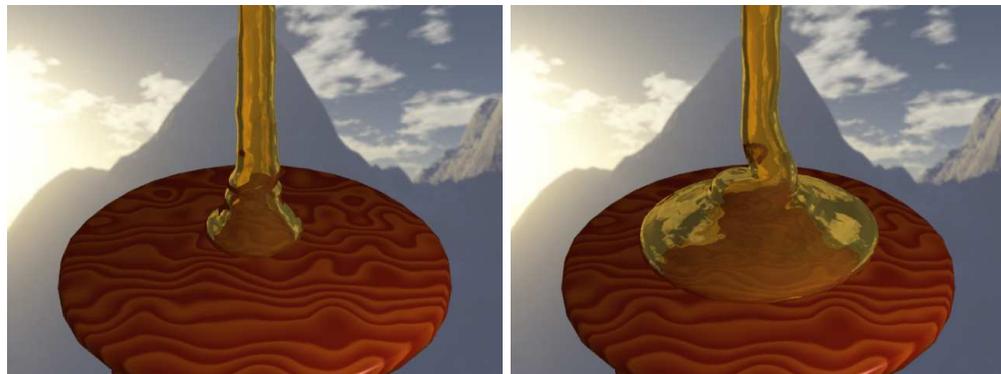


FIG. 7.8 – Tourbillon de sirop s'étendant sur un tabouret. L'émergence du tourbillon est causée par l'élasticité de la substance alors qu'elle se replie sur elle-même. Les plis caractéristiques se dissipent rapidement sous l'effet de la plasticité.

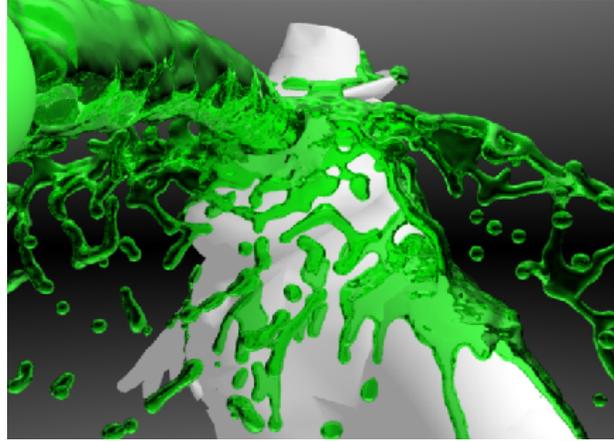


FIG. 7.9 – Éclaboussure détaillée sur un personnage en rotation. Ce scénario de collision extrême, où un liquide interagit violemment avec un objet en mouvement puis se sépare en structures complexes se déplaçant à grande vitesse, est particulièrement difficile à traiter autrement qu’avec un système de particules de grande stabilité.

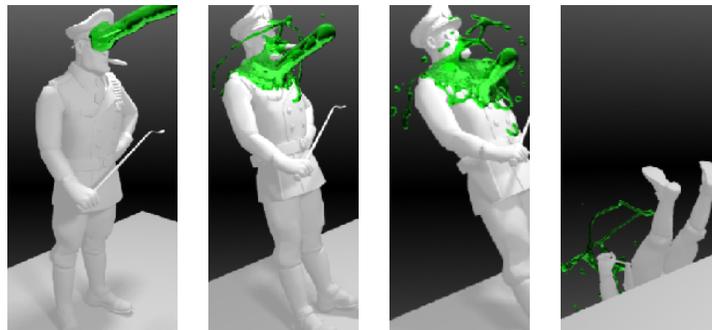
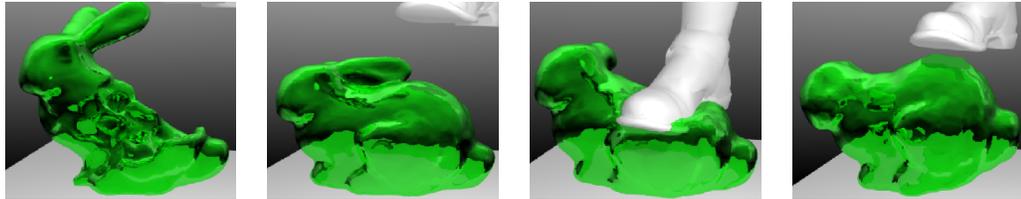
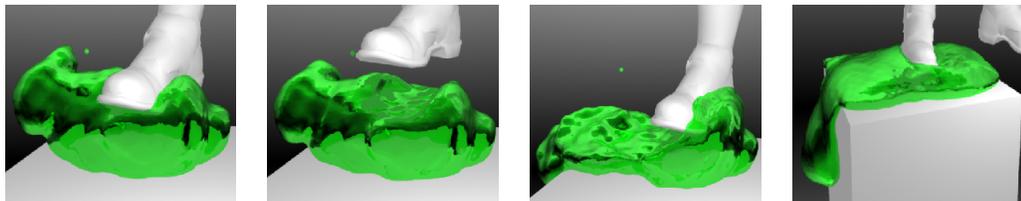


FIG. 7.10 – Jet de liquide éclaboussant le général. Le jet est suffisamment puissant pour faire basculer le personnage dans l’abîme.



Un moule invisible est rempli de liquide. On place des ressorts entre les particules, puis on enlève le moule. On écrase le lapin élastique résultant, mais il reprend sa forme.



Ensuite, on augmente la plasticité, et on l'écrase de nouveau. Cette fois, le pied laisse sa trace. Finalement, on enlève les ressorts et le lapin s'écoule.

FIG. 7.11 – Modification interactive des paramètres.

# Chapitre 8

## Conclusion

Ce mémoire présente notre travail sur la simulation de fluides viscoélastiques à base de particules. Nos contributions incluent un schéma d'intégration stable pour de longs pas de temps, un traitement robuste de la tension de surface sans calcul de courbure, des ressorts à longueurs de repos dynamiques pour les comportements viscoélastiques, et finalement un couplage bidirectionnel avec des corps rigides. Ces contributions ont fait l'objet d'une publication au *Symposium on Computer Animation 2005* [CBP05].

Le point clé de notre approche est le principe de *prédiction-relaxation*, couplé avec des mesures de distribution de particules, permettant de faire évoluer une particule en fonction du contexte dans lequel elle se trouve. Notre pas de temps consiste à avancer les particules à leurs *positions prédites* selon leurs vitesses puis de s'assurer que la configuration résultante respecte certaines contraintes en appliquant des déplacements radiaux entre paires de particules. Au pas de temps suivant, les particules continuent d'avancer selon la vitesse correspondant à la différence entre la position relaxée et la position initiale, ce qui force les particules à conserver une vitesse conforme aux différents événements qui se produisent au cours du pas de temps.

La relaxation des positions prédites tient compte de plusieurs objectifs : 1) une particule doit avoir un "*nombre*" constant de voisines, 2) une particule doit avoir peu de voisines rapprochées, 3) les particules reliées par un ressort doivent se placer à une distance correspondant à la longueur de repos du ressort, et 4) les particules ne doivent pas pénétrer dans les objets. Des comportements divers tels que la tension de surface, la viscoélasticité et la flottaison émergent de la confrontation entre ces objectifs souvent contradictoires.

Nos expérimentations avec la forme spécifique des équations régissant le mouvement des particules ont mené à une formulation étonnamment simple et intuitive, puis nous avons montré que la méthode résultante est rapide, versatile et visuellement correcte. Sa validité pourra subir le test définitif prochainement, alors que des développeurs imaginatifs tenteront de l'intégrer dans le contexte fébrile d'un environnement virtuel interactif.

### 8.0.1 Travaux Futurs

Voici quelques spéculations sur les directions que notre recherche sur la simulation à base de particules pourrait prendre.

#### Fluides multi-phases

En définissant les propriétés du fluide indépendamment pour chaque type de particule, on pourra simuler par exemple l'interaction eau/huile ou eau/air. Concernant la définition de la densité, certains choix devront être faits : est-ce que les particules d'huile devraient participer à la densité d'une particule d'eau ? Comment interagissent deux particules de types différents ?

L'intuition acquise lors de tests préliminaires nous incite à favoriser l'approche suivante : traiter indépendamment chaque type de fluide, sans compter les particules d'eau dans les mesures de densité de l'huile, puis ajouter une simple répulsion entre les particules de différents types. De cette façon, la relaxation de double densité continuera de causer des effets de tension de surface. Si on considère toutes les particules environnantes dans le calcul de densité, sans égard à leur type, on perd la tension de surface. Physiquement, traiter indépendamment chaque type de fluide semble correct. En effet une particule d'eau n'a rien à faire d'autre avec une particule d'huile que simplement la repousser.

Une goutte d'air dans un verre d'eau monte parce qu'elle est plus légère que le liquide environnant. Toutefois, la gravité affecte l'air et l'eau de la même façon, l'air tombe autant que l'eau. On pourrait tricher et changer la gravité dépendamment de la masse de la particule (force de buoyance), mais on verrait des artefacts assez dérangeants : une goutte d'eau aurait de la difficulté à tomber à cause de l'inertie de l'air. On voit qu'on n'a pas le choix de diminuer l'inertie des particules légères, on ne peut pas seulement changer la gravité. Ceci nous encourage à essayer de faire monter une bulle d'air en

considérant seulement le fait qu'elle a moins d'inertie que l'eau.

Lors d'un échange de forces entre une particule d'eau et une particule d'air, la particule d'air bouge plus que la particule d'eau. La bulle d'air monte simplement parce que chaque échange d'impulsion avec l'eau la défavorise : l'eau va où elle veut et l'air se tasse docilement. Le rapport de masse entre l'eau et l'air est de 1000 pour 1. On arrive au problème suivant : une bulle d'air composée d'une dizaine de particules se fait littéralement écraser par le liquide environnant. Ce problème est très difficile. La bulle d'air a besoin de conserver son volume, mais la seule façon qu'elle a de réclamer son espace est de pousser l'eau environnante. Chaque fois qu'elle essaie de pousser une particule d'eau, elle doit reculer 1000 fois plus. Alors elle pousse la particule d'eau se trouvant de l'autre côté, sans obtenir plus de succès. On peut imaginer la particule d'air oscillant d'un côté et de l'autre en tentant de convaincre l'eau environnante de se tasser un peu.

L'eau gagne le combat contre l'air pour prendre la place, mais en même temps l'air a le droit indéniable d'avoir de l'espace. Finalement, l'air devrait bouger vers le haut, après une bataille interminable contre les particules d'eau voisines. Simuler une bulle montante à base de particules sans introduire de force artificielle de buoyance est un défi beaucoup plus ardu qu'il n'y paraît, mais quelqu'un y parviendra sûrement un jour.

### **Particules rotatives et articulations généralisées**

Pour obtenir un matériel tel la plasticine, il est important que le comportement élastique soit le plus rigide possible. Idéalement, aucune oscillation ne devrait se produire lorsqu'on laisse tomber une boule de plasticine sur le sol. Notre façon de traiter la viscosité et l'élasticité est adéquate pour ce type de matériel (la prédiction-relaxation sert bien les systèmes rigides et visqueux), mais notre modèle de ressorts pourrait être amélioré. Une façon d'augmenter la puissance du modèle serait de permettre aux ressorts d'avoir un effet angulaire. Ainsi, les oreilles du lapin pourraient tenir en l'air même si elles ne sont composées que de quelques particules. Présentement, les ressorts radiaux doivent être nombreux et coopérer ensemble pour parvenir à solidifier le matériel. Une chaîne de particules liées par des ressorts se comporte comme une corde, mais on aimerait qu'elle se comporte comme une tige rigide. Il sera très intéressant de lier les particules avec des liens intrinsèquement rigides.

On généralisera nos particules en leur donnant une orientation et une vitesse de rotation. On considère donc une particule comme un corps rigide sphérique. Un ressort angulaire peut alors être exprimé comme un ensemble d'articulations entre deux particules. On peut contraindre complètement le mouvement relatif en plaçant plusieurs articulations entre deux particules. L'articulation est définie par sa position relative dans la particule 1 et sa position relative dans la particule 2. On applique des impulsions aux deux particules jointes de façon à ce que l'articulation ne se sépare pas. En 2D, deux articulations par paire de particules suffisent à contraindre le mouvement relatif, et en 3D, on a besoin de trois articulations.

À partir de ce modèle augmenté de particules rotatives et d'articulations généralisées, on pourra traiter la plasticité de façon unifiée. On modifiera les positions relatives des articulations en fonction de leur déformation courante. Ce type de modèle a déjà été investigué récemment au cours du projet de stage de Lawrence Hum [Hum05].

### Multirésolution

Dans le but d'augmenter l'efficacité du modèle, il faudra éventuellement le généraliser en permettant aux particules d'avoir des tailles différentes. De grosses particules à l'intérieur et de petites particules près de la surface. Les équations devront être réécrites de façon à ce que deux particules puissent avoir des masses différentes. Deux questions se posent alors. Premièrement, quand doit-on séparer ou fusionner des particules ? On peut imaginer qu'un critère de subdivision et de fusion pourrait simplement dépendre de la distance estimée à la surface. Deuxièmement, comment faire ces opérations sans subir de *popping* ?

Lors d'une fusion entre deux particules, il faut éviter de simplement les remplacer par une grosse particule, ce qui cause invariablement une discontinuité dans le comportement (causant le *popping*). Pour solutionner ce problème, on pourrait développer un système de transfert de masse entre particules. Une des deux particules transférerait graduellement sa masse à l'autre, avant de disparaître. Pour ce qui est de la subdivision, plutôt que de séparer directement une particule, on pourrait faire apparaître des particules sans masse dans son voisinage. On transférerait alors graduellement la masse de la particule trop grosse vers les particules plus petites. Chaque transfert de masse devra s'accompagner d'un changement de vitesse et de position des particules impliquées,

de façon à conserver le même centre de masse et la même quantité de mouvement. Ainsi, on aurait un système dont la résolution est véritablement adaptative, et dont les changements de résolution sont continus.

# Bibliographie

- [CBP05] Simon Clavet, Philippe Beaudoin et Pierre Poulin. « Particle-based Viscoelastic Fluid Simulation ». Dans *SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 219–228, 2005.
- [CMHT02] Mark Carlson, Peter J. Mucha, R. Brooks Van Horn et Greg Turk. « Melting and Flowing ». Dans *SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 167–174, 2002.
- [DG96] Mathieu Desbrun et Marie-Paule Gascuel. « Smoothed Particles : A New Paradigm for Animating Highly Deformable Bodies ». Dans *Computer Animation and Simulation '96*, pages 61–76, 1996.
- [EMF02] D. Enright, S. Marschner et R. Fedkiw. « Animation and Rendering of Complex Water Surfaces ». Dans *SIGGRAPH*, pages 736–744, 2002.
- [FF01] Nick Foster et Ronald Fedkiw. « Practical Animations of Liquids ». Dans *SIGGRAPH*, pages 23–30, 2001.
- [FM97] Nick Foster et Dimitris Metaxas. « Modeling the Motion of a Hot, Turbulent Gas ». Dans *SIGGRAPH*, pages 181–188, 1997.
- [GBF03] Eran Guendelman, Robert Bridson et Ronald Fedkiw. « Nonconvex rigid bodies with stacking ». Dans *SIGGRAPH*, pages 871–878, 2003.
- [GBO04] Tolga G. Goktekin, Adam W. Bargteil et James F. O'Brien. « A Method for Animating Viscoelastic Fluids ». Dans *SIGGRAPH*, pages 463–468, 2004.
- [GM77] R.A. Gingold et J.J. Monaghan. « Smoothed Particle Hydrodynamics – Theory and Application to non-spherical stars ». *Monthly Notices of the Royal Astronomical Society*, volume 181, page 375, 1977.
- [Hum05] Lawrence Hum. « Projet de Baccalauréat, IFT3051 », 2005.

- [LC87] William E. Lorensen et Harvey E. Cline. « Marching Cubes : A High Resolution 3D Surface Construction Algorithm ». Dans *SIGGRAPH*, pages 163–169, 1987.
- [LGF04] F. Losasso, F. Gibou et R. Fedkiw. « Smoke, Water & Goop : Simulating Water and Smoke with an Octree Data Structure ». Dans *SIGGRAPH*, pages 457–462, 2004.
- [Luc77] Leon Lucy. « A Numerical Approach to the Testing of the Fission Hypothesis ». *Astronomical Journal*, volume 82, page 1013, 1977.
- [MCG03] Matthias Müller, David Charypar et Markus Gross. « Particle-based Fluid Simulation for Interactive Applications ». Dans *SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 154–159, 2003.
- [MP89] G. Miller et A. Pearce. « Globular Dynamics : A Connected Particle System for Animating Viscous Fluids ». *Computers & Graphics*, volume 13, numéro 3, pages 305–309, 1989.
- [PIX] « [sourceforge.net/projects/pixie](http://sourceforge.net/projects/pixie) ».
- [Ree83] W. T. Reeves. « Particle Systems – A Technique for Modelling a Class of Fuzzy Objects ». Dans *SIGGRAPH*, pages 359–376, 1983.
- [Sta99] Jos Stam. « Stable Fluids ». Dans *SIGGRAPH*, pages 121–128, 1999.
- [THM<sup>+</sup>03] M. Teschner, B. Heidelberger, M. Mueller, D. Pomeranets et M. Gross. « Optimized Spatial Hashing for Collision Detection of Deformable Objects ». Dans *Vision, Modeling, and Visualization*, pages 47–54, 2003.
- [TPF89] Demetri Terzopoulos, John Platt et Kurt Fleischer. « Heating and Melting Deformable Models (from goop to glob) ». Dans *Graphics Interface*, pages 219–226, 1989.
- [VID] « [www.iro.umontreal.ca/labs/infographie/papers/Clavet-2005-PVFS/](http://www.iro.umontreal.ca/labs/infographie/papers/Clavet-2005-PVFS/) ».