

## TABLE DES MATIÈRES

	Page
INTRODUCTION .....	1
CHAPITRE 1 ÉTAT DE L'ART .....	5
1.1 Simulations de fluides .....	5
1.2 Simulations eulériennes .....	7
1.3 Simulations lagrangiennes.....	8
1.3.1 Mouvement .....	11
1.3.2 Noyaux .....	13
1.3.3 Stabilité .....	14
1.3.3.1 <i>Pas de temps</i> adaptatif .....	15
1.3.3.2 Viscosité .....	16
1.3.3.3 XSPH .....	18
1.3.3.4 Incompressibilité .....	18
1.3.4 Simulations hybrides.....	19
1.4 Simulations multiphases Eulériennes .....	19
1.5 Simulation multiphases lagrangiennes.....	20
1.6 Contributions.....	23
CHAPITRE 2 FONTE ET SOLIDIFICATION .....	25
2.1 Simulation liquide .....	25
2.1.1 <i>Smoothed-particle hydrodynamics</i> .....	25
2.1.2 Détection de collision .....	26
2.2 Changement de phase.....	27
2.2.1 Erreur de déformation.....	28
2.2.2 Correction de la déformation .....	31
2.2.3 Critère d'arrêt .....	34
2.2.4 Propagation des contraintes.....	35
CHAPITRE 3 STABILITÉ DE LA SIMULATION .....	39
3.1 <i>Pas de temps</i> adaptatif .....	39
3.2 Diffusion de chaleur implicite.....	41
3.3 Ajustement de la masse volumique initiale.....	46
CHAPITRE 4 IMPLÉMENTATION .....	49
4.1 Description du simulateur .....	49
4.1.1 Initialisation .....	49
4.1.2 Exécution de la simulation.....	50
4.2 Parcours des voisins .....	51
4.3 Détection de collision.....	53
4.4 Paramètres.....	53

CHAPITRE 5	RÉSULTATS .....	55
5.1	Présentation des résultats .....	55
5.2	Comparaison avec les travaux antérieurs .....	59
CHAPITRE 6	DISCUSSION .....	63
6.1	Avantages .....	63
6.2	Limitations .....	64
CHAPITRE 7	TRAVAUX FUTURS .....	65
CONCLUSION	.....	67
BIBLIOGRAPHIE	.....	68

## LISTE DES TABLEAUX

	Page
Tableau 4.1	Valeurs des paramètres physiques ..... 54
Tableau 5.1	Statistiques des exemples ..... 60
Tableau 5.2	Comparaison avec la viscosité traditionnelle ..... 60



## LISTE DES FIGURES

	Page
Figure 1.1	Cellule d'une grille échelonnée en 2D. .... 7
Figure 1.2	Simulation de fumée à l'aide d'une simulation semi-lagrangienne. .... 8
Figure 1.3	Simulation liquide eulérienne ..... 9
Figure 1.4	Simulation lagrangienne de type SPH ..... 10
Figure 1.5	Coupe transversale des noyaux $W_{poly6}$ , $W_{spiky}$ et $W_{viscosité}$ ..... 14
Figure 1.6	Fonte d'un lapin à l'aide d'une simulation eulérienne. .... 19
Figure 1.7	Fonte et solidification à l'aide d'une simulation solide, élastique et rigide. . 22
Figure 2.1	Calcul de la position cible ..... 31
Figure 2.2	Ajustement des forces rigides ..... 32
Figure 2.3	Venus de Milo avec et sans propagation des contraintes ..... 36
Figure 2.4	« h » avec et sans la seconde condition. .... 38
Figure 3.1	Comparaison d'une simulation avec et sans <i>pas de temps</i> adaptatif. .... 39
Figure 3.2	Instabilité de la diffusion de chaleur explicite ..... 42
Figure 3.3	Diffusion implicite de la chaleur ..... 43
Figure 3.4	Masse volumique problématique aux frontières du fluide ..... 46
Figure 4.1	Fonctionnement du simulateur ..... 50
Figure 4.2	Recherche de voisins dans une grille ..... 52
Figure 5.1	Fonte de deux blocs avec des rigidités $s_i$ différentes ..... 56
Figure 5.2	Fonte des bras de l'Armadillo de Stanford ..... 57
Figure 5.3	Fonte du lapin de Stanford ..... 58
Figure 5.4	Exemples du « h ». .... 59
Figure 5.5	Comparaison entre la viscosité traditionnelle et notre approche. .... 61



## LISTE DES ALGORITHMES

	Page
Algorithme 2.1      Simulation de fluides de type SPH .....	26
Algorithme 2.2      Simulation de fluides avec changement de phases .....	29





## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

SPH	Smoothed-particle hydrodynamics
CFL	Courant–Friedrichs–Lewy
<i>Pas de temps</i>	Laps de temps simulé lors d’une itération d’un simulateur ( <i>time step</i> )
LBM	Lattice Boltzmann methods
Problème raide	<i>Stiff problem</i>
Forces d’amortissement	<i>Damping forces</i>
Saute-mouton	<i>Leapfrog</i>
Grille échelonnée	<i>staggered grid</i>



## LISTE DES SYMBOLES ET UNITÉS DE MESURE

$\vec{v}_i$	Vitesse de la particule $i$
$\vec{x}_i$	Position de la particule $i$
$m_i$	Masse de la particule $i$
$\mu_i$	Viscosité de la particule $i$
$p_i$	Pression de la particule $i$
$\rho_i$	Masse volumique de la particule $i$
$\rho_0$	Masse volumique du fluide au repos
$h$	Rayon d'influence
$\vec{x}_{ij}$	$\vec{x}_i - \vec{x}_j$
$W_{ij}$	$W(\vec{x}_{ij}, h)$ , Noyau
$c$	Vitesse de propagation d'une onde dans le fluide
$\alpha_v$	Coefficient de viscosité de volume et de cisaillement
$\xi$	Coefficient de viscosité des ondes ayant un grand nombre de Mach
$\vec{x}_{ij}^0$	Position initiale relative de $i$ par rapport à $j$
$\alpha_{ij}$	Coefficient du lien rigide entre les particules $i$ et $j$
$s_i$	Rigidité de la particule $i$
$T_i$	Température de la particule $i$
$T_{solid}$	Température où le fluide devient complètement rigide
$T_{liquid}$	Température où le fluide devient complètement liquide
$dist_0$	Distance initiale entre les particules
$N$	Nombre de voisins
$\nabla f$	Gradient de la fonction $f$
$\nabla \cdot \vec{u}$	Divergence du champ de vecteurs $\vec{u}$
$\nabla^2$	Laplacien d'un champ de vecteurs ou de scalaires
$\ \vec{x}\ $	Norme du vecteur $\vec{x}$



# INTRODUCTION

## Contexte

Au cinéma, comme à la télévision, de nombreux effets visuels normalement complexes à reproduire sont réalisés par ordinateur. L'utilisation de l'ordinateur pour générer ces effets permet de réduire les coûts de production et parfois même les dangers auxquels les acteurs et cascadeurs doivent faire face. De plus, l'ordinateur permet d'avoir un meilleur contrôle sur ces effets visuels et permet aussi de créer des effets visuels ne pouvant normalement pas être reproduits. Parmi ces effets, l'on retrouve l'animation de fluides : fumée, liquide, fonte d'un solide, etc. Par exemple, il peut être coûteux et dangereux d'inonder une ville dans le cadre d'une scène. Par ailleurs, si le résultat obtenu n'est pas satisfaisant, chaque tentative supplémentaire engendrera des coûts énormes. Un autre exemple est la fonte d'un objet : si le résultat initial ne donne pas le résultat escompté, il faudra en faire fondre un nouveau. De plus, contrôler exactement comment ce dernier fond est très complexe et nécessitera potentiellement de nombreux essais avant d'obtenir un résultat satisfaisant. Ces risques et ces coûts peuvent être grandement diminués en créant ces effets par ordinateur. Toutefois, ces effets demeurent difficiles à reproduire correctement avec les techniques d'animation par ordinateur traditionnelles. Ceci est dû à la complexité du comportement des fluides, ce qui rend difficile la production d'une animation visuellement plausible, fluide et temporellement cohérente. C'est pour cela qu'une simulation physique est normalement utilisée pour réaliser ce type d'animation. Puisqu'il n'est pas nécessaire que l'animation soit physiquement exacte, mais plutôt visuellement plausible, les simulateurs peuvent se permettre de ne pas être physiquement exacts, ce qui permet d'améliorer le contrôle de l'artiste sur le résultat tout en élargissant l'éventail d'effets pouvant être produits.

## Problématique et hypothèse

De nombreux travaux portent sur les simulations de fluides et couvrent différents aspects liés à celles-ci : simulation de liquide, simulation de gaz, coexistence des phases, couplage avec le solide, contrôle, incompressibilité, etc. Toutefois, la simulation des phénomènes de fonte et de

solidification représente toujours un défi avec les approches par particules. La nature des équations utilisées par celles-ci requiert un *pas de temps* très petit afin d'assurer la stabilité de la simulation. Cela engendre de grands temps de calcul puisqu'un très grand nombre d'itérations de simulation doivent être exécutées pour chaque image de l'animation. De plus, les déformations qui affectent globalement une grande partie d'un liquide mettent du temps à se propager car chaque élément du fluide n'influence que ses voisins immédiats. Ceci est particulièrement vrai lorsqu'un fluide est extrêmement visqueux, voire presque rigide. Un élément du liquide doit avoir un comportement cohérent avec l'ensemble du liquide, ce qui s'obtient au fur et à mesure que les éléments de liquide s'influencent mutuellement de voisin en voisin. Ainsi, afin de propager l'effet global d'une déformation avant que les déformations locales ne soient trop grandes et potentiellement incohérentes avec la déformation globale, il est nécessaire d'utiliser un petit *pas de temps*. Ceci contribue à augmenter les temps de calcul, ce qui fait de la simulation de fonte et solidification un problème très ardu. D'autre part, la rigidité d'un fluide est normalement assurée à l'aide de forces de viscosité ou de contrainte visqueuse. La grandeur de celles-ci est contrôlée à l'aide d'un paramètre variant entre 0 (liquide) et  $+\infty$  (complètement rigide) : il est donc difficile de déterminer quelle valeur ce paramètre doit avoir pour obtenir le niveau de rigidité désiré. Étant donné les grands temps de calcul, il peut être très long et difficile d'ajuster ce paramètre par essai-erreur.

Dans le cadre de ce mémoire, l'hypothèse suivante est émise : il est possible d'aborder le problème sous une forme permettant d'améliorer la stabilité de la simulation et d'utiliser un *pas de temps* plus élevé. De plus, ceci permettra l'utilisation d'un paramètre variant entre 0 (liquide) et 1 (solide) offrant une variation plutôt linéaire de la rigidité au sein de cet intervalle. Finalement, cette approche sera en mesure de bien gérer la propagation de la déformation, permettant ainsi la simulation de fluides presque complètement rigides.

## Organisation du document

Le chapitre 1 présente une revue de la littérature sur les simulations de fluide en infographie et les simulations multiphases. Dans le chapitre 2, la technique développée dans le cadre de ce projet est détaillée. Dans un premier temps, les bases du simulateur permettant de simuler

des liquides sont expliquées. Par la suite, les modifications apportées au simulateur afin de permettre la simulation de phénomènes de fonte et solidification sont traitées. Les détails sur l'implémentation de l'approche sont fournis au chapitre 3. Les résultats obtenus à l'aide de cette technique sont présentés dans le chapitre 4. Le chapitre 5 offre un retour critique sur le projet et ses résultats sous la forme de discussion. Finalement, des travaux futurs sont proposés dans le chapitre 5.





## CHAPITRE 1

### ÉTAT DE L'ART

L'animation de fluides à l'aide de simulations physiques ne date pas d'hier. Ce chapitre introduira les concepts liés à la simulation de fluide en infographie, ainsi que les approches populaires utilisées dans ce domaine. Par la suite, les techniques récentes utilisées pour simuler des fluides multi-phases et la transition entre ces phases seront détaillées.

#### 1.1 Simulations de fluides

Le comportement d'un fluide, à l'échelle macroscopique, peut être décrit selon les équations de Navier-Stokes. Ce système d'équations comporte deux équations dont l'objectif est de décrire le mouvement d'un fluide et la conservation de sa masse. Pour une explication plus détaillée des équations de Navier-Stokes dans le contexte de l'infographie, veuillez vous référer au livre de Bridson (2008).

La première équation, l'équation de la quantité de mouvement, prend la forme suivante pour les fluides Newtoniens incompressibles :

$$\rho \left( \frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} \right) = -\nabla p + \mu \nabla^2 \vec{v} + \vec{f} \quad (1.1)$$

Celle-ci est directement dérivée de la seconde loi de Newton ( $F = ma$ ). Les différentes variables et opérateurs contenus dans cette équation sont expliqués dans la « liste des symboles et unités de mesure » au début de ce document. L'expression contenue dans la parenthèse, à la gauche de l'égalité, correspond à la dérivée matérielle de la vitesse du fluide. Cette dérivée matérielle indique que la vitesse du fluide évolue dans le temps et dans l'espace. Les termes à la droite de l'égalité représentent les forces appliquées au sein du fluide. Le premier terme décrit les forces de pression agissant sur le fluide. Celles-ci sont calculées en utilisant le gradient de la pression. Ceci indique donc que le fluide a tendance à se déplacer vers les régions où la pression est plus faible. Le second terme décrit les forces de viscosité. Celles-ci sont calculées

en utilisant la divergence du gradient (ou le Laplacien) de la vitesse. Celles-ci génèrent donc des forces basées sur le flux moyen du fluide, résultant en un effet de lissage dont la grandeur est influencée par la viscosité  $\mu$  du fluide. Le dernier terme inclut toute force externe appliquée au sein du fluide (telle que la gravité).

La seconde équation est l'équation de continuité (ou conservation de la masse). Celle-ci indique que la masse du fluide doit demeurer constante dans le temps :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \quad (1.2)$$

Il existe trois catégories d'approches typiques pour résoudre ces équations : les approches eulériennes, lagrangiennes et hybrides. La première catégorie discrétise le fluide dans l'espace sous forme de cellules. Chaque cellule possède une position fixe et les propriétés du fluide (vitesse, masse volumique, etc.) sont évaluées au centre de ces cellules. C'est la variation de ces propriétés qui est calculée selon les équations de Navier-Stokes et qui permettra de déterminer le comportement du fluide. La seconde catégorie discrétise le fluide sous forme de particules, c'est-à-dire des points ayant une position variant dans le temps. Le mouvement de ces particules décrit le comportement du fluide et la variation des propriétés du fluide est évaluée aux points dans l'espace où elles se situent. La position de ces points est mise à jour en appliquant des forces, déterminées en fonction des équations de Navier-Stokes, sur ces particules. Finalement, la troisième catégorie utilise une approche eulérienne qu'elle couple avec une approche lagrangienne afin d'éliminer des lacunes propres aux simulations eulériennes, telle que la perte de masse et la difficulté à représenter les détails plus fins tel que des éclaboussures.

Les méthodes les plus communes pour chacune de ces trois approches, ainsi que les techniques utilisées pour simuler les phénomènes de changement de phase à partir de celles-ci, seront détaillées dans les prochaines sections de ce chapitre.

## 1.2 Simulations eulériennes

Les approches eulériennes actuelles découlent souvent des travaux de Harlow et Welsh (1965) qui introduisirent l'approche *Marker-And-Cell* (MAC) afin de simuler des liquides en 2D. Dans cette approche, le fluide est simulé dans une grille échelonnée où les propriétés du fluide (telle que la pression) sont évaluées au centre des cellules. Quant aux vitesses, elles sont divisées selon leurs composantes cartésiennes et évaluées aux frontières entre les cellules (voir figure 1.1). L'évaluation des vitesses aux frontières des cellules permet une meilleure précision que l'éva-

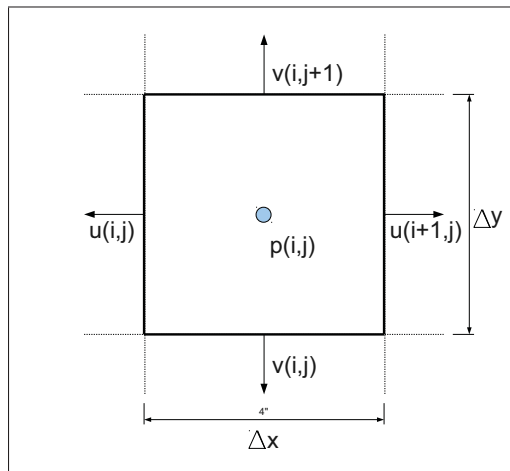


Figure 1.1 Cellule d'une grille échelonnée en 2D.

Les propriétés du fluide sont évaluées au centre des cellules alors que les vitesses sont divisées selon leurs composantes cartésiennes ( $u,v$ ) et évaluées aux frontières avec les cellules adjacentes.

luation de celles-ci au centre (Bridson, 2008). Toutefois, certains auteurs, Stam (1999) par exemple, ont obtenus de bon résultats en évaluant la vitesse au centre des cellules. Ceci leur permet de simplifier l'implémentation de la technique. Pendant la simulation, les cellules de la grille sont mises à jour en appliquant la méthode des différences finies aux équations de Navier-Stokes (équations 1.1 et 1.2). Celles-ci sont intégrées à l'aide d'un intégrateur itératif afin de mettre à jour la pression et les vitesses au sein du volume de simulation, c'est-à-dire la grille. De nombreuses années plus tard, Foster et Metaxas (1996) adaptèrent cette technique au 3D et y ajoutèrent le support pour la collision avec des géométries statiques. Ils adaptèrent ensuite leurs travaux aux phénomènes gazeux (Foster et Metaxas, 1997). Bien qu'elle donne

un comportement visuellement plausible, cette approche nécessite un *pas de temps* petit afin de conserver la stabilité de la simulation. Par conséquent, cette dernière engendre de grands temps de calculs.

Afin de minimiser ce problème, Stam (1999) proposa l'approche semi-lagrangienne qui permet l'utilisation d'un *pas de temps* beaucoup plus grand. Dans cette approche, l'intégration des termes responsables de l'advection des propriétés au sein du fluide est remplacé par un traitement inspiré des approches lagrangiennes : Les propriétés sont mises à jour en effectuant le parcours inverse de l'élément de fluide se trouvant au centre de la cellule. La figure 1.2 montre une simulation de fumée exécutée à l'aide du simulateur semi-lagrangien du logiciel *Houdini* de *Side Effects Software*. Stam appliqua cette technique uniquement aux gaz, toutefois elle fut

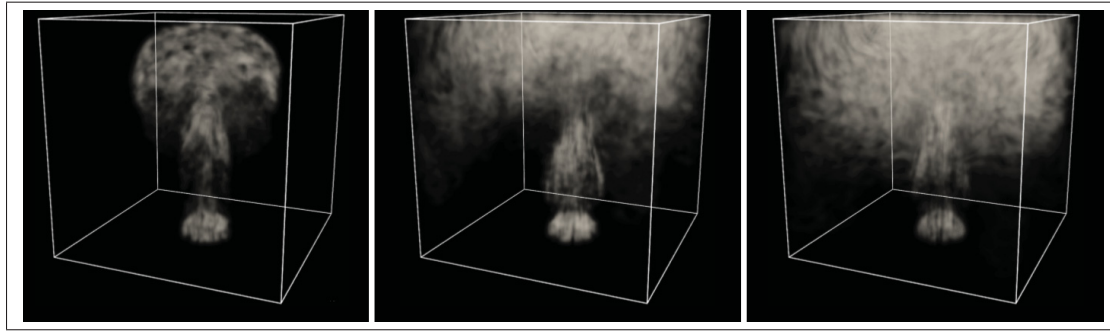


Figure 1.2 Simulation de fumée à l'aide d'une simulation semi-lagrangienne. Simulation et rendu effectués à l'aide du logiciel Houdini.

ensuite appliquée avec succès aux liquides par Foster et Fedkiw (2001). La figure 1.3 montre une simulation eulérienne d'un liquide interagissant avec une géométrie statique exécutée à l'aide du simulateur d'Houdini.

### 1.3 Simulations lagrangiennes

Les *Smoothed-particle hydrodynamics* (SPH) furent introduites en infographie par Desbrun et Cani (1996). Ces derniers s'inspirèrent des travaux de Monaghan (1992), en astrophysique, décrivant une approche permettant de calculer une propriété dans un continuum à partir d'un ensemble de particules. Ces particules se déplacent dans l'espace et possèdent des propriétés

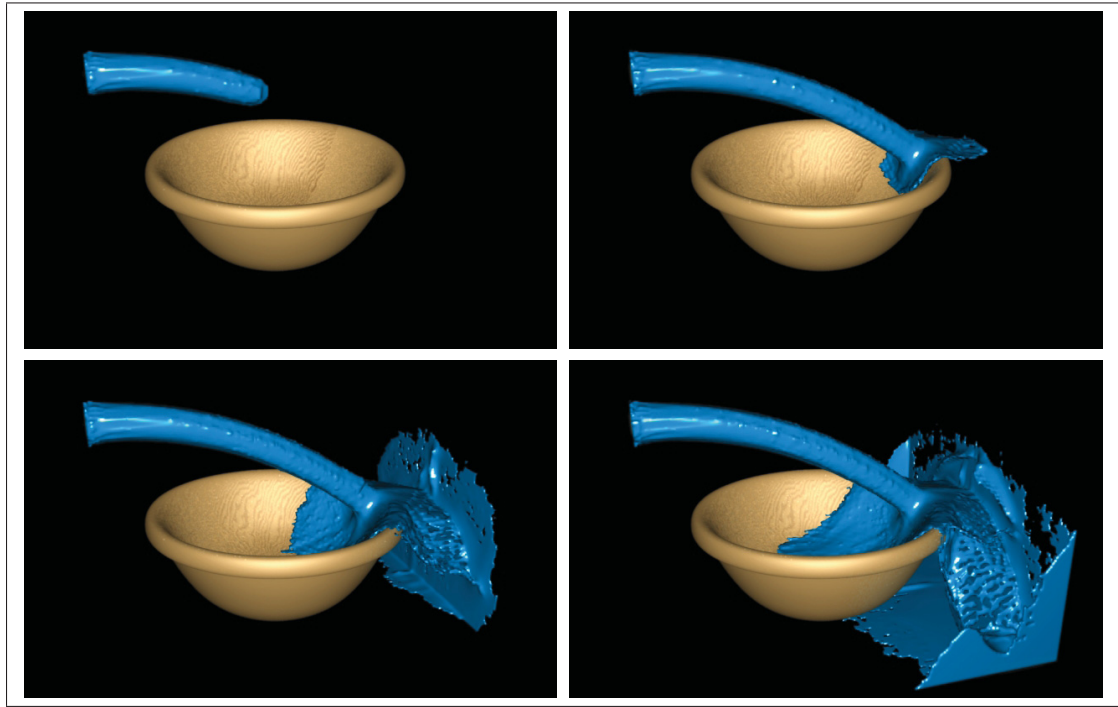


Figure 1.3 Simulation liquide eulérienne  
Simulation et rendu effectués à l'aide du logiciel Houdini.

qui évoluent dans le temps. Une masse  $m_i$  est associée à chaque particule, quoiqu'il soit courant que toutes les particules aient la même masse. La figure 1.4 montre une simulation par particules de type SPH ainsi que la surface reconstruite à partir de celles-ci.

Selon le principe des SPH, il est possible de calculer une propriété  $A$  à la position  $\vec{x}$  dans un continuum en utilisant la formule suivante :

$$A(\vec{x}) = \sum_j m_j \frac{A_j}{\rho_j} W(\|\vec{x} - \vec{x}_j\|, h) \quad (1.3)$$

La sommation est effectuée en utilisant toutes les particules  $j$  voisines au point  $\vec{x}$ . Une particule  $j$  est voisine du point  $\vec{x}$  lorsque la distance les séparant est inférieure au rayon d'influence  $h$  des particules. Le noyau  $W$  est une fonction permettant d'accorder plus ou moins d'importance aux particules en fonction de leur distance. Plus cette distance est proche du rayon d'influence  $h$ , moins la particule aura d'influence sur le calcul de la propriété. Lorsque la distance est supérieure à  $h$ , les particules n'influencent plus l'équation 1.3.

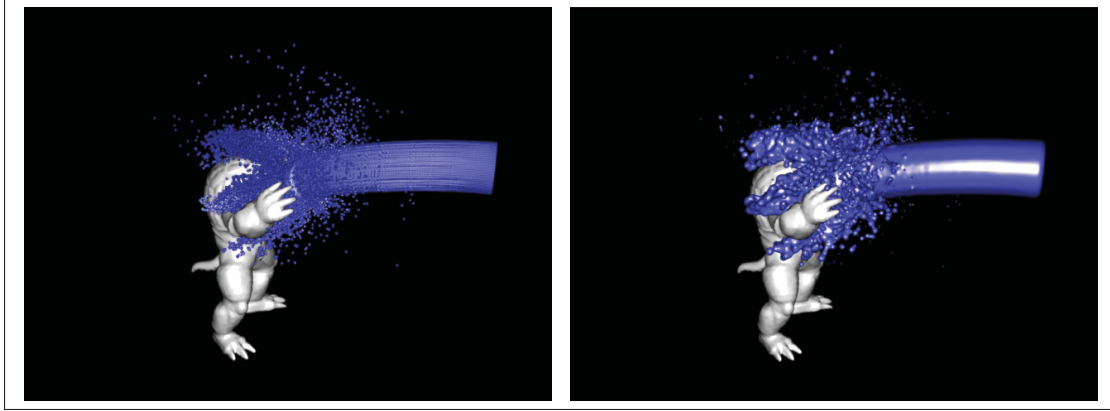


Figure 1.4 Simulation lagrangienne de type SPH

À gauche : visualisation des particules. À droite : une surface a été reconstruite à l'aide des particules. Simulation et rendu effectués à l'aide du logiciel Houdini

Le calcul d'une propriété dans le continuum dépend de la masse volumique  $\rho_i$  des particules. Celle-ci peut être calculée à l'aide de l'équation 1.3, ce qui donne :

$$\begin{aligned}
 \rho_i &= \rho(\vec{x}_i) \\
 &= \sum_j m_j \frac{\rho_j}{\rho_j} W(\|\vec{x}_i - \vec{x}_j\|, h) \\
 &= \sum_j m_j W(\|\vec{x}_i - \vec{x}_j\|, h) \\
 &= \sum_j m_j W_{ij}
 \end{aligned} \tag{1.4}$$

Pour calculer le gradient d'une propriété, il suffit de remplacer le noyau  $W$  par son gradient dans l'équation 1.3. Le même principe s'applique au calcul du Laplacien :

$$\nabla A(\vec{x}_i) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W_{ij} \tag{1.5}$$

$$\nabla^2 A(\vec{x}_i) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W_{ij} \tag{1.6}$$

À l'aide de ces concepts, il sera possible d'estimer les propriétés des particules en fonction de leurs voisins et de déterminer les forces qui régissent leur comportement.

### 1.3.1 Mouvement

En appliquant les principes décrits ci-dessus aux équations de Navier-Stokes, Desbrun et Cani (1996) obtinrent une formulation lagrangienne de la simulation de fluides en infographie. La masse d'une particule étant invariable, l'équation de conservation de la masse peut être ignorée. Il ne reste donc que l'équation de conservation de la quantité de mouvement. Puisque les particules se déplacent dans l'espace suivant donc le mouvement du fluide, la dérivée matérielle du côté gauche de l'équation de la quantité de mouvement (équation 1.1) peut être simplifiée :

$$\begin{aligned}\rho \left( \frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} \right) &= -\nabla p + \mu \nabla^2 \vec{v} + \vec{f} \\ \rho \frac{\partial \vec{v}}{\partial t} &= -\nabla p + \mu \nabla^2 \vec{v} + \vec{f}\end{aligned}$$

De plus, chaque terme du côté droit représente une force agissant sur les particules :

$$\begin{aligned}\rho \frac{\partial \vec{v}}{\partial t} &= -\nabla p + \mu \nabla^2 \vec{v} + \vec{f} \\ &= \vec{f}^{pression} + \vec{f}^{viscosité} + \vec{f}^{externe}\end{aligned}\tag{1.7}$$

Une particule  $i$  est donc influencée par les forces (pression, viscosité et externe) agissant sur cette dernière :

$$\rho_i \frac{d\vec{v}}{dt} = \vec{f}_i^{pression} + \vec{f}_i^{viscosité} + \vec{f}_i^{externe}\tag{1.8}$$

Cette équation est utilisée pour mettre à jour la vitesse des particules à chaque itération de la simulation. Les forces de pression peuvent être calculées directement en utilisant l'équation 1.5 pour calculer le gradient de la pression :

$$\begin{aligned}\vec{f}_i^{pression} &= -\nabla p(\vec{x}_i) \\ &= -\sum_j m_j \frac{\rho_j}{\rho_j} \nabla W_{ij}\end{aligned}\tag{1.9}$$

La pression  $p_i$  d'une particule est calculée en fonction de l'écart entre la masse volumique  $\rho_i$  d'une particule et la masse volumique au repos  $\rho_0$  du fluide :

$$p_i = k(\rho_i - \rho_0) \quad (1.10)$$

Le coefficient  $k$  permet d'ajuster la grandeur des forces de pression. La grandeur de ce paramètre modifie la compressibilité du fluide. Plus le coefficient  $k$  est élevé, moins le fluide sera compressible. Toutefois, ceci engendrera de grandes forces nécessitant un petit *pas de temps* afin qu'elle reste stable.

Selon Monaghan (1992), l'équation 1.9 ne conserve pas le principe d'action-réaction puisque les forces ne sont pas symétriques entre les particules. Il proposa donc cette forme qui préserve la symétrie des forces :

$$\vec{f}_i^{pressure} = -\rho_i \sum_j m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W_{ij} \quad (1.11)$$

Un peu plus tard, Müller *et al.* (2003) proposèrent cette forme :

$$\vec{f}_i^{pressure} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W_{ij} \quad (1.12)$$

Tout en préservant la symétrie des forces, celle-ci est plus simple et répond mieux à leurs exigences en termes de stabilité et rapidité de calculs dans le cadre de la simulation interactive de fluides.

Pour le calcul des forces de viscosité, Desbrun et Cani (1996) utilisèrent les équations de viscosité artificielle (voir section 1.3.3) normalement utilisées en astrophysique. Müller *et al.* (2003) proposèrent par la suite une formulation basée sur le terme de viscosité de l'équation de la quantité de mouvement (equation 1.1) qu'ils modifièrent afin de préserver la symétrie des forces :

$$\vec{f}_i^{viscosité} = \mu \sum_j \frac{m_j}{\rho_j} (\vec{v}_j - \vec{v}_i) \nabla^2 W_{ij}$$



Ceux-ci modifièrent ensuite cette équation afin de permettre la simulation de fluides ayant une viscosité variable (Müller *et al.*, 2005) :

$$\vec{f}_i^{viscosité} = \sum_j \frac{\mu_j + \mu_i}{2} \frac{m_j}{\rho_j} (\vec{v}_j - \vec{v}_i) \nabla^2 W_{ij} \quad (1.13)$$

Les équations présentées dans cette section sont à la base des implémentations actuelles des SPH en infographie et sont celles utilisées dans le cadre de ce projet (voir section 2.1.1).

### 1.3.2 Noyaux

Le choix du noyau  $W$  utilisé est important car il influencera le comportement de la simulation et la stabilité de celle-ci. D'ailleurs, les noyaux utilisés normalement en astrophysique (Monaghan, 1992) ont tendance à créer des agglomérations de particules. Ceci est dû au gradient qui s'atténue lorsque deux particules deviennent très rapprochées. Ce problème est atténué par la nature des équations utilisées en astrophysique, mais est présent dans le contexte des équations de Desbrun et Cani (1996). Afin de remédier à ce problème, les auteurs proposèrent un nouveau noyau  $W_{spiky}$  mieux adapté à la simulation de fluides en infographie :

$$W_{spiky}(r, h) = \frac{15}{h^6 \pi} \begin{cases} (h - r)^3 & 0 \leq r \leq h \\ 0 & \text{autrement} \end{cases}$$

Par la suite, Müller *et al.* (2003) proposèrent un autre noyau  $W_{poly6}$  nécessitant moins de temps de calculs :

$$W_{poly6}(r, h) = \frac{315}{64h^9 \pi} \begin{cases} (h^2 - r^2)^3 & 0 \leq r \leq h \\ 0 & \text{autrement} \end{cases}$$

Contrairement à  $r$ ,  $r^2$  peut être calculé sans utiliser de racine carrée, allégeant ainsi les calculs. Les auteurs utilisèrent ce noyau dans le cadre de la simulation interactive de fluides. Cependant, ce noyau souffre du même problème d'agglomération décrit précédemment. Pour cette raison, ils utilisèrent le noyau  $W_{spiky}$  de Desbrun et Cani (1996) afin de calculer les forces de pression qui nécessitent le gradient du noyau. De plus, l'utilisation de ces noyaux dans le calcul de la viscosité (équation 1.13) peut causer des instabilités dans la simulation. Ceci est dû au fait que

leur Laplacien change de signe lorsque deux particules sont plus rapprochées. Müller *et al.* (2003) créèrent donc un troisième noyau  $W_{viscosité}$  dont le Laplacien ne change pas de signe :

$$W_{viscosité}(r, h) = \frac{15}{2h^3\pi} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1 & 0 \leq r \leq h \\ 0 & \text{autrement} \end{cases}$$

Ces derniers utilisèrent ce noyau lors du calcul de la viscosité, assurant ainsi la stabilité des calculs. La figure 1.5 montre une coupe transversale des trois noyaux décrits dans cette section.

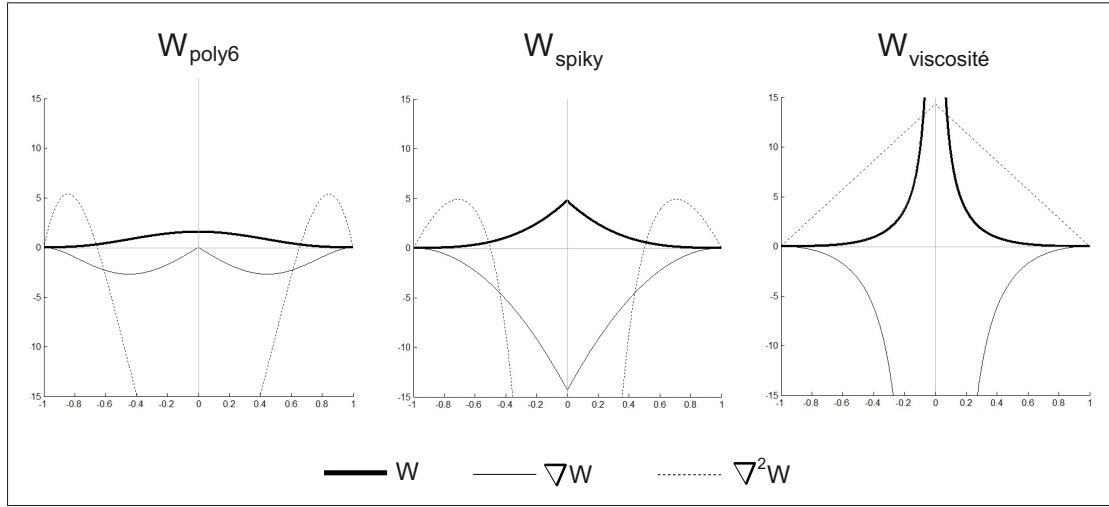


Figure 1.5 Coupe transversale des noyaux  $W_{poly6}$ ,  $W_{spiky}$  et  $W_{viscosité}$   
Le rayon d'influence  $h = 1$ .

Dans cette figure, le gradient  $\nabla W$  (en direction du centre) et le Laplacien  $\nabla^2 W$  de ces noyaux sont aussi représentés.

### 1.3.3 Stabilité

La simulation par particules de type SPH, telle que décrite dans la section précédente, est sujette à des instabilités. Ceci est dû à la discrétisation temporelle de l'animation et au fait que la mise à jour de la vitesse et de la position des particules est posée sous forme d'un problème raide. Par exemple, de très grandes forces de pression sont générées si des particules sont trop rapprochées. Appliquer ces grandes forces avec un *pas de temps* trop grand ajoute de l'énergie dans la simulation et peut résulter en un comportement instable. L'utilisation d'un *pas de temps*

adéquat permet d'éliminer en grande partie ces instabilités. Toutefois, ceci peut engendrer de longs temps de calcul et n'est pas suffisant pour assurer la stabilité de la simulation. Cette section décrit différentes techniques permettant d'améliorer la stabilité d'une simulation de SPH.

### 1.3.3.1 *Pas de temps adaptatif*

Lors d'une simulation, la grandeur optimale du *pas de temps* assurant la stabilité avec le moins de temps de calculs possible varie en fonction de divers facteurs. Celle-ci est difficile à déterminer puisqu'elle varie d'une simulation à l'autre. De plus, un petit *pas de temps* peut être nécessaire uniquement pour quelques itérations de la simulation, alors qu'une valeur beaucoup plus grande peut être utilisée pour le reste de la simulation. Utiliser une valeur fixe est donc non triviale (recours à l'essai-erreur) et n'est pas optimale. L'utilisation d'un *pas de temps* adaptatif règle ces deux problèmes. Cette technique se base généralement sur la condition Courant–Friedrichs–Lewy (CFL) :

$$\Delta x_{max} \leq \beta \cdot h$$

Cette condition est nécessaire pour qu'un algorithme d'intégration explicite puisse converger. Dans le cadre d'une simulation par particules, celle-ci implique que le déplacement maximal  $\Delta x_{max}$  d'une particule ne peut dépasser une certaine distance. Dans le cadre des SPH, cette distance est une fraction  $\beta$  du rayon d'influence  $h$ . La condition peut être réécrite en fonction de la vitesse maximale :

$$v_{max} \Delta t \leq \beta \cdot h \quad (1.14)$$

Donc, en connaissant la vitesse maximale  $v_{max}$  d'une particule il est possible d'obtenir le *pas de temps* à utiliser :

$$\begin{aligned} v_{max} \Delta t &= \beta \cdot h \\ \Delta t &= \beta \frac{h}{v_{max}} \end{aligned} \quad (1.15)$$

En astrophysique, Monaghan (1989) utilisa deux critères dérivés de l'équation 1.15 pour calculer le *pas de temps*  $\Delta t$  :

$$\Delta t = 0.25 \min \left( \sqrt{\frac{h}{f_{\max}}}, \min_i \frac{h}{c + 0.6(\alpha_v c + \xi \max_j (\mu_{ij}))} \right)$$

Le premier critère tient compte de la force maximale appliquée sur les particules, alors que le second critère tient compte de la vitesse de propagation d'une onde dans le fluide et des forces de viscosité artificielles (voir section 1.3.3.2). Paiva *et al.* (2006) adaptèrent les critères de Monaghan (1989) à leur modèle viscoplastique et obtinrent les deux critères suivants :

$$\Delta t = 0.1 \min \left( \frac{h}{v_{\max} + c}, \frac{h^2}{6\eta_{\max}} \right)$$

Le second critère est lié aux forces découlant de leur modèle viscoplastique. Le premier critère est intéressant car il combine la vitesse maximale  $v_{\max}$  des particules au début de l'itération à la vitesse de propagation d'une onde dans le fluide  $c$  afin d'approximer la vitesse maximale théorique d'une particule. Ces approches permettent de calculer un *pas de temps* assurant la stabilité de la simulation. Toutefois, étant donné le fait que la vitesse maximale  $v_{\max}$  est approximée, les calculs ne sont pas exacts. Par exemple, il est possible que la variation de vitesse excède la valeur de  $c$  si de nombreuses particules se retrouvent comprimées, générant ainsi de très grandes forces de pression. De plus, bien que l'utilisation d'un *pas de temps* dynamique permette d'empêcher les problèmes d'instabilité liés à l'utilisation d'un *pas de temps* trop élevé, ce dernier n'empêche pas l'ajout d'énergie dans la simulation par les forces de pression.

### 1.3.3.2 Viscosité

Les forces de pression génèrent une attraction/répulsion entre les particules en fonction de la masse volumique du fluide dans ces régions. Ces forces ont tendance à générer des mouvements oscillatoires et ajouter de l'énergie dans la simulation, engendrant ainsi des instabilités. Il est donc nécessaire d'ajouter des forces d'amortissement dans la simulation afin de contrer ces effets. Pour cette raison, une viscosité assez grande pour contrer ces problèmes doit être utilisée, et ce, même pour les fluides dont la viscosité est négligeable.

En plus de la viscosité introduite par Müller *et al.* (2003), une autre formulation du terme de viscosité, nommée viscosité artificielle, est utilisée dans la littérature scientifique. Cette méthode fut introduite par Monaghan (1989) dans le cadre de ses travaux en astrophysique et fut ensuite réutilisée en infographie par Desbrun et Cani (1996). Dans cette approche, les forces de viscosité  $f_i^{viscosité}$  sont calculées ainsi :

$$\vec{f}_i^{viscosité} = -\rho_i \sum_j m_j \Pi_{ij} \nabla W_{ij} \quad (1.16)$$

$$\Pi_{ij} = \begin{cases} \frac{2(-\alpha_v \mu_{ij} c + \xi \mu_{ij}^2)}{\rho_i + \rho_j}, & (\vec{v}_i - \vec{v}_j) \cdot (\vec{x}_i - \vec{x}_j) < 0 \\ 0, & (\vec{v}_i - \vec{v}_j) \cdot (\vec{x}_i - \vec{x}_j) \geq 0 \end{cases}$$

$$\mu_{ij} = \frac{h (\vec{v}_i - \vec{v}_j) \cdot (\vec{x}_i - \vec{x}_j)}{\|\vec{x}_i - \vec{x}_j\|^2 + 0.01 h^2}$$

Cette force agit en fonction du mouvement relatif des particules entres-elles. Les mouvement de rapprochement des particules seront atténués par cette force, venant ainsi contrer les mouvements oscillatoires des particules. Les coefficients  $\alpha_v$  et  $\xi$  permettent de contrôler la grandeur des forces générées. Le premier terme de  $\Pi_{ij}$  correspond à la viscosité de volume et la viscosité de cisaillement alors que le second terme est semblable à la viscosité artificielle de Von Neumann-Richtmyer et permet de prévenir l'interpénétration des particules lorsque leur vitesse est élevée. Dans ces deux travaux, les auteurs utilisent une valeur de  $\alpha_v = 1$  et  $\xi = 2$ . Cette équation fut reprise par Paiva *et al.* (2006) qui ne conservèrent que le premier terme de  $\Pi_{ij}$  :

$$\Pi_{ij} = \begin{cases} -\frac{2\alpha_v \mu_{ij} c}{\rho_i + \rho_j}, & (\vec{v}_i - \vec{v}_j) \cdot (\vec{x}_i - \vec{x}_j) < 0 \\ 0, & (\vec{v}_i - \vec{v}_j) \cdot (\vec{x}_i - \vec{x}_j) \geq 0 \end{cases}$$

Cette approche contrôle la grandeur des forces à l'aide d'un seul paramètre et donne de bons résultats.

### 1.3.3.3 XSPH

Tel que mentionné précédemment, des particules trop rapprochées peuvent engendrer des forces de pression très grandes dues à la compression du fluide. Monaghan (1989) introduisit une correction de la vitesse, nommée XSPH, permettant d'atténuer ce problème :

$$\vec{v}_i \leftarrow \vec{v}_i + \varepsilon \sum_j \frac{m_j}{0.5(\rho_i + \rho_j)} (\vec{v}_j - \vec{v}_i) W_{ij}$$

Cette correction est appliquée aux vitesses une fois qu'elles ont été mises à jour à partir des forces. Le facteur  $\varepsilon$ , variant entre 0 et 1, permet d'ajuster la grandeur de cette correction. Cette correction ajuste la vitesse d'une particule selon le champ de vitesses moyen du fluide. Ceci permet de diminuer les problèmes de pénétration entre les particules, empêchant le fluide de trop se comprimer.

### 1.3.3.4 Incompressibilité

Selon les équations des forces de pression, le fluide doit se comprimer afin que les particules puissent se repousser. Il n'est donc pas possible de simuler un fluide totalement incompressible. Toutefois, en augmentant la valeur du coefficient de pression  $k$ , il est possible de diminuer la compression du fluide afin qu'il devienne presque incompressible. Ceci requiert néanmoins l'utilisation d'un *pas de temps* extrêmement petit afin de ne pas compromettre la stabilité de la simulation. Certains auteurs (Cummins et Rudman, 1999; Losasso *et al.*, 2008; Shao, 2006) se sont inspirés des approches eulériennes pour obtenir un champ de vitesses sans divergence. Par la suite, Solenthaler et Pajarola (2009) utilisèrent un intégrateur de type prédiction-correction afin d'optimiser la masse volumique du fluide en corrigeant les forces de pression itérativement.

Cette approche leur permit d'utiliser un grand *pas de temps* et de diminuer les temps de calculs par rapport aux approches précédentes puisqu'un faible nombre d'itérations est requis pour converger vers une solution. Ceux-ci démontrèrent ainsi qu'il est possible d'optimiser une propriété du fluide en corrigeant des forces à l'aide d'un intégrateur de type prédiction-correction.

De plus, le résultat obtenu est stable et visuellement plausible. L'approche présentée dans ce mémoire s'inspire grandement de ce concept.

### 1.3.4 Simulations hybrides

Une première technique combinant une approche eulérienne et une approche lagrangienne fut présentée par Harlow (1963). Ceux-ci introduisirent la technique *Particle-In-Cell* (PIC) utilisant une grille de simulation eulérienne afin de simuler des fluides compressibles. Toutefois, l'advection du fluide se fait à l'aide de particules et le résultat est ensuite retransféré sur la grille de simulation. Cette approche souffre toutefois de dissipation numérique, donnant un aspect plus visqueux. Plus tard, Brackbill et Ruppel (1986) améliorèrent cette approche afin de résoudre ce problème, donnant ainsi naissance à la technique *Fluid Implicit-Particle* (FLIP). Ces techniques furent finalement adaptées aux fluides incompressibles par Zhu et Bridson (2005). Ceux-ci utilisèrent ces approches pour simuler des matériaux granulaires. Ces approches utilisent des particules pour corriger la grille de simulation, toutefois ces particules n'interagissent pas entre elles. Un peu plus tard, Losasso *et al.* (2008) innovèrent en couplant une simulation semi-lagrangienne à une simulation de SPH. Ceux-ci parviennent à utiliser les forces des deux approches afin de simuler adéquatement les mouvements denses d'eau, à l'aide d'une simulation eulérienne, et les mouvements plus fins, tel que des éclabousses, à l'aide des SPH.

## 1.4 Simulations multiphases Eulériennes

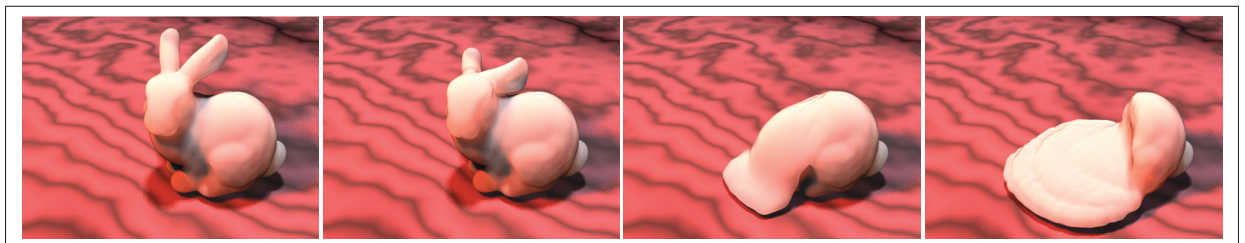


Figure 1.6 Fonte d'un lapin à l'aide d'une simulation eulérienne.  
Adapté de Carlson *et al.* (2002). ©2002 Association for Computing Machinery, Inc.  
Reprinted by permission. <http://doi.acm.org/10.1145/545261.545289>

Carlson *et al.* (2002) proposèrent une approche permettant d'effectuer des simulations liquide-solide offrant une transition fluide entre les états. Ceux-ci utilisèrent une simulation de type MAC (voir section 1.2) où la variation du paramètre de viscosité permet de contrôler la « rigidité » du fluide. Lorsque la viscosité du fluide tend vers l'infini, le comportement du fluide s'apparente d'avantage à celui d'un solide. Cependant, une valeur élevée du paramètre de viscosité nécessite un *pas de temps* très petit afin de conserver la stabilité de la simulation. Les auteurs proposèrent donc une formulation implicite des équations nécessaires à la résolution du terme de viscosité. Ceci leur permit d'exécuter des simulations très visqueuses en utilisant un *pas de temps* très grand. Toutefois, leur approche gère mal les fluides en *vol libre*, donnant un effet de ralentissement. Un traitement spécial doit être effectué pour réduire l'impact de ce problème. La figure 1.6 montre un lapin partiellement en fonte simulée à l'aide de cette approche. La viscosité varie au sein du même fluide en fonction de la température, donnant un effet réaliste de fonte. Par la suite, Fält et Roble (2003) améliorèrent cette technique afin d'éliminer le problème d'atténuation des vitesses au sein d'un fluide en *vol libre*. Plus tard, Rasmussen *et al.* (2004) améliorèrent l'approche de Carlson *et al.* (2002) en utilisant une approche semi-implicite. Cette approche gère mieux la viscosité variant au sein d'un même fluide et ne souffre pas des problèmes de ralentissement du fluide en vol libre. Plus tard, Batty et Bridson (2008) proposèrent une approche entièrement implicite permettant de bien gérer la viscosité variable, mais aussi de gérer les effets de la tension de surface. Celle-ci permet la simulation d'effets rotationnels dans les fluides, donnant ainsi des effets beaucoup plus réalistes. Par exemple, une barre de fluide visqueux fixée par une extrémité ne tombera pas en ligne droite, ce qui est le cas avec les approches précédentes. Celle-ci courbera sous l'effet de la gravité et du stress subi en surface.

## 1.5 Simulation multiphases lagrangiennes

En premier lieu, Monaghan et Kocharyan (1995) utilisèrent des SPH pour simuler de la poussière (solide) couplée à un gaz. Par la suite, Monaghan (1997) proposa une approche implicite-itérative couplant de la poussière et un gaz. Cette dernière traite individuellement chaque lien entre les particules de manière implicite permettant une simulation stable des deux phases,



même lorsque la viscosité est plus élevée. Toutefois, ces approches ne gèrent pas la transition entre les états (solidification, fonte, sublimation, etc.). Plus tard, Müller *et al.* (2005) proposèrent une approche multiphase où le passage à l'état gazeux ou liquide s'effectue en modifiant les propriétés des particules. Néanmoins, cette approche ne gère pas la phase solide.

Paiva *et al.* (2006) utilisèrent ensuite un modèle viscoplastique où le traitement de la viscosité est remplacé par l'utilisation d'un tenseur de stress. Ce modèle est mieux adapté aux fluides viscoplastiques et permet d'obtenir une viscosité variable en fonction de la grandeur des forces appliquées sur les particules. La rigidité est contrôlée par un paramètre unique appelé *jump number*. Ce dernier est calculé en fonction de la température des particules qui est contrôlée par l'utilisateur et l'équation de diffusion de la chaleur. Par contre, à l'instar des approches précédentes, obtenir un fluide presque rigide demande l'utilisation d'un très petit *pas de temps*. Afin de permettre la simulation de fluides rigides et d'obtenir une meilleure transition entre l'état rigide et l'état liquide, Solenthaler *et al.* (2007) présentèrent une approche utilisant trois types de simulations différentes pour chaque état de la matière. Celle-ci utilise une simulation de corps rigides pour simuler les particules complètement rigides, une simulation de fluide pour les particules liquides et une simulation élastique pour la phase de transition entre les deux états. Un modèle basé sur la différence entre la position relative des particules voisines et leur position relative initiale est utilisé afin de générer les forces élastiques. Lorsque les particules deviennent liquide ou se solidifient, des liens sont détruits ou créés. De plus, lorsque deux particules subissent un stress trop élevé en fonction de leur température, leurs positions relatives sont mises à jour afin de permettre aux objets de se déformer. Comme pour les travaux précédents, la température des particules est utilisée pour ajuster la rigidité du fluide définie par la grandeur des forces d'élasticité. Toutefois, ceci donne un comportement élastique dans la phase de transition du fluide, ce qui n'est pas toujours désirable. La figure 1.7 montre le résultat d'une simulation obtenue à l'aide de cette technique. Dans cet exemple, la base de trois cubes a été fondue et resolidifiée, fusionnant les trois objets sous la forme d'un nouveau solide.

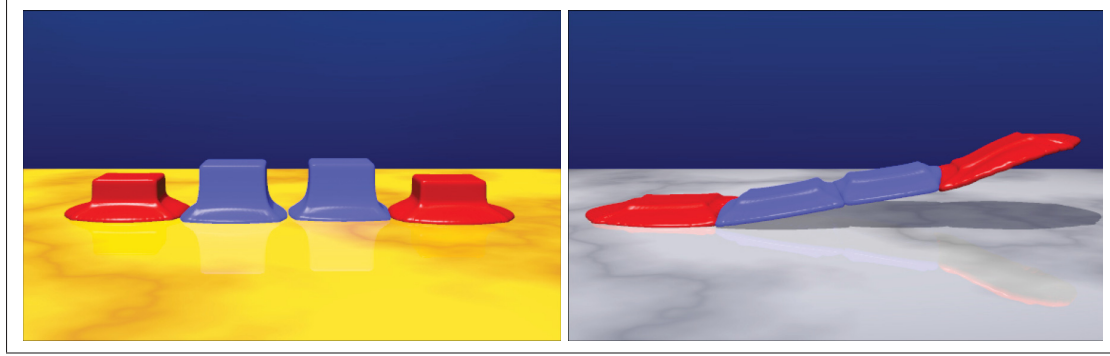


Figure 1.7 Fonte et solidification à l'aide d'une simulation solide, élastique et rigide.  
Adapté de Solenthaler Solenthaler *et al.* (2007). ©2007 John Wiley & Sons Ltd.  
<http://dx.doi.org/10.1002/cav.162>

La diffusion de la chaleur permet d'ajouter du réalisme aux simulations en reproduisant l'échange de chaleur entre les différents éléments du fluide. Stora *et al.* (1999) furent les premiers à exprimer une formulation de l'équation générale de diffusion de la chaleur dans le contexte des SPH :

$$\begin{aligned}\frac{dT_i}{dt} &= k \nabla^2 T_i \\ &= k \sum_j \frac{m_j}{\rho_j} \nabla T_j \nabla W_{ij}\end{aligned}$$

Où le gradient de la température  $\nabla T_i$  d'une particule est calculé ainsi :

$$\nabla T_i = \sum_j \frac{m_j}{\rho_j} (T_j - T_i) \nabla W_{ij}$$

L'approche mentionnée ne calcule pas le Laplacien directement. Selon les auteurs, l'effet de lissage du kernel ne permet pas de calculer facilement le Laplacien de la température  $\nabla^2 T_i$ . Toutefois, Solenthaler *et al.* (2007) et Chang *et al.* (2009) utilisèrent directement le Laplacien de la température et obtinrent de bons résultats :

$$\frac{dT_i}{dt} = k \sum_j \frac{m_j}{\rho_j} (T_j - T_i) \nabla^2 W_{ij}$$

Paiva *et al.* (2006, 2009) stipulèrent que cette approche peut donner des résultats incohérents dans leur cas puisque le Laplacien de leur noyau peut changer de signe. De plus, cette approximation du Laplacien peut engendrer des pertes thermiques. Ceux-ci proposèrent donc une formulation du Laplacien utilisant le gradient du noyau :

$$\frac{dT_i}{dt} = k \sum_j \frac{m_j}{\rho_j} \left( \frac{4\rho_i}{\rho_i + \rho_j} \right) (T_j - T_i) \frac{\vec{x}_{ij} \cdot \nabla W_{ij}}{\|\vec{x}_{ij}\|^2 + 0.01h^2}$$

Bien que ces différentes approches donnent des résultats visuellement plausibles, celles-ci sont toutes posées sous la forme de problème raide, donc elles sont sujettes à des instabilités lorsque la valeur du *pas de temps* ou du coefficient de diffusion de la chaleur  $k$  est élevée (voir section 3.2).

## 1.6 Contributions

Dans le domaine des simulations eulériennes, les travaux actuels permettent d'effectuer des simulation ultra-visqueuses. Il est donc possible de simuler, dans des temps raisonnables, la transition entre l'état solide et liquide afin de créer des effets de fonte et solidification. Du côté des simulations lagrangiennes, la transition entre l'état solide et liquide demeure difficile à bien simuler, surtout lorsque le fluide est presque rigide. Ceci est dû à la nature des équations, posées sous un problème raide, qui nécessitent un *pas de temps* extrêmement petit lorsque la viscosité du fluide augmente. Ce problème, toutefois, est similaire à celui de l'incompressibilité du fluide dans les simulation lagrangiennes. Ce dernier fut résolu par Solenthaler et Pajarola (2009) qui utilisèrent un intégrateur de type prédiction-correction afin de déterminer les forces permettant de minimiser la compression du fluide. Il devrait être possible d'appliquer une solution similaire au problème des fluides très visqueux.

L'objectif de ce projet de recherche est de poser sous la forme d'un problème non raide le calcul d'un fluide hautement visqueux dans le cadre d'une simulation lagrangienne. Pour réaliser cet objectif, il est nécessaire d'atteindre les objectifs secondaires suivants :

- Déterminer un modèle permettant de mesurer la déformation du fluide et de calculer l'erreur entre celle-ci et la déformation désirée.
- Créer un mécanisme permettant d'influencer le fluide afin de réduire cet écart.
- Assurer la stabilité de la simulation en tout temps.

Les contributions du projet présenté dans ce mémoire sont les suivantes :

- Un modèle lagrangien permettant la simulation de fluides de différentes viscosités incluant les cas de viscosité extrême.
- Simulation de la transition entre les états solides et liquides contrôlée à l'aide d'un unique paramètre et pour lequel le comportement du liquide varie de façon relativement linéaire entre 0 et 1.
- Réduction des temps de calculs nécessaires à la simulation de fluides extrêmement visqueux à l'aide d'une simulation lagrangienne.
- Simulation inconditionnellement stable de la diffusion de la chaleur entre les particules à l'aide d'une approche implicite.
- Meilleur calcul du *pas de temps* nécessaire au maintien de la stabilité de la simulation

## CHAPITRE 2

### FONTE ET SOLIDIFICATION

Dans ce chapitre, la technique développée dans le cadre de ce projet de recherche est détaillée. Tout d'abord, la section 2.1 décrit le simulateur de fluide à la base de notre approche. Par la suite, la section 2.2 explique les modifications, donc les contributions de ce mémoire, à cette simulation afin de permettre la simulation de phénomènes de fonte et solidification.

#### 2.1 Simulation liquide

À l'instar des travaux précédents sur la simulation de phénomènes de fonte et solidification (voir chapitre 1), une simulation de fluides a été utilisée afin de créer l'animation de l'état liquide et de la transition entre cet état et l'état solide. L'approche présentée dans ce mémoire se base sur une simulation lagrangienne de type SPH, donc le fluide est discrétisé sous forme de particules évoluant selon l'équation des SPH (équation 1.8).

##### 2.1.1 *Smoothed-particle hydrodynamics*

Les forces de pression  $\vec{f}_i^{pression}$  sont calculées à l'aide de l'équation 1.12. La seule force externe  $\vec{f}_i^{externe}$  agissant sur les particules est la gravité :

$$\vec{f}_i^{externe} = \rho_i \cdot \vec{g} \quad (2.1)$$

Afin d'assurer la stabilité de la simulation une force de viscosité  $\vec{f}_i^{viscosité}$  est calculée à l'aide de l'équation de la viscosité artificielle (équation 1.16). Ceci permet d'ajouter des forces d'amortissement dans la simulation et ainsi éviter que de l'énergie soit ajoutée dans cette dernière. De plus, cette force permet de prévenir un mouvement d'oscillation des particules dû aux forces de pression.

Une fois les forces calculées, la vitesse et la position des particules sont mises à jour à l'aide de la méthode d'intégration saute-mouton (*leapfrog*). Celle-ci a l'avantage d'avoir une précision

de deuxième ordre, donc elle est plus précise et stable que la méthode d'Euler sans nécessiter beaucoup de calculs et de mémoire supplémentaire. Cette méthode consiste en l'utilisation de la vitesse au temps  $t + \Delta t/2$  pour calculer la nouvelle position au temps  $t + \Delta t$  à l'aide de l'accélération  $a_i$  calculée en utilisant l'équation des SPH (équation 1.8) :

$$\begin{aligned}\vec{x}_i(t + \Delta t) &= \vec{x}_i(t) + \vec{v}_i(t + \Delta t/2)\Delta t \\ \vec{v}_i(t + \Delta t/2) &= \vec{v}_i(t - \Delta t/2) + \vec{a}_i(t)\Delta t\end{aligned}$$

La vitesse  $\vec{v}_i$  au temps  $t$ , utilisée lors du calcul de la viscosité, est ensuite obtenue en calculant la moyenne des vitesses aux temps  $t - \Delta t/2$  et  $t + \Delta t/2$  :

$$\vec{v}_i(t) = \frac{\vec{v}_i(t - \Delta t/2) + \vec{v}_i(t + \Delta t/2)}{2}$$

À chaque itération de la simulation de fluide, les forces sont tout d'abord calculées. Par la suite, la vitesse et la position des particules sont mises à jour selon cette méthode (voir algorithme 1).

#### Algorithme 2.1 Simulation de fluides de type SPH

1	<b>tant que</b> Temps simulation < Temps fin simulation <b>faire</b>
2	// Phase 1 : Calcul des forces (Section 2.1.1)
3	Calculer la masse volumique $\rho_i$ et la pression $p_i$ de chaque particule
4	Calculer les forces $\vec{f}_i^{pression}$ , $\vec{f}_i^{viscosité}$ et $\vec{f}_i^{externe}$ pour chaque particule
5	// Phase 2 : Intégration
6	Calculer $\Delta t$
7	Mettre à jour la vitesse $\vec{v}_i$ et la position $\vec{x}_i$ pour chaque particule
8	Temps simulation += $\Delta t$

##### 2.1.2 Détection de collision

La détection de collision est gérée lorsque la position des particules est mise à jour. Lorsqu'une particule traverse un plan de collision, la particule est repositionnée au point d'impact. La composante parallèle  $\vec{v}_i^{\parallel}$  et la composante perpendiculaire  $\vec{v}_i^{\perp}$  de sa vitesse, par rapport au

plan de collision, sont alors corrigées ainsi :

$$\begin{aligned}\vec{v}_i^{\parallel} &\leftarrow \omega \cdot \vec{v}_i^{\parallel} \\ \vec{v}_i^{\perp} &\leftarrow (0,0,0) \\ \vec{v}_i &= \vec{v}_i^{\parallel} + \vec{v}_i^{\perp}\end{aligned}$$

Ceci permet d'empêcher les particules de pénétrer les plans de collision tout en permettant aux particules de glisser sur ce dernier. La valeur du coefficient  $\omega$  permet de contrôler la friction entre les particules et le plan de collision. Une valeur de  $\omega = 0$  empêchera tout glissement, alors qu'une valeur de  $\omega = 1$  n'appliquera aucune friction sur les particules.

## 2.2 Changement de phase

Afin de permettre le contrôle de la fonte et la solidification du fluide, une température est assignée à chaque particule. Celle-ci est initialement spécifiée par l'artiste et évolue selon l'équation de diffusion de la chaleur (voir section 3.2). Lorsque la température d'une particule est sous le point de solidification  $T_{solide}$ , celle-ci aura un comportement rigide alors qu'une température au dessus du point de liquéfaction  $T_{liquide}$  résultera en un comportement complètement liquide. La technique présentée dans cette section permet de traiter la transition entre ces deux états, donc de déterminer le comportement des particules dont la température se trouve entre le point de solidification et le point de liquéfaction.

Tout d'abord, une rigidité  $s_i$  est assignée à chaque particule. Cette rigidité permettra de déterminer le degré de rigidité de la particule  $i$ . Une valeur de  $s_i = 0$  correspond à l'état complètement liquide, alors qu'une valeur de  $s_i = 1$  correspond à l'état complètement solide. Celle-ci est calculée en fonction de la température  $T_i$  de la particule :

$$s_i = \begin{cases} 1, & T_i \leq T_{solide} \\ \frac{T_{liquide} - T_i}{T_{liquide} - T_{solide}}, & T_{solide} < T_i < T_{liquide} \\ 0, & T_{liquide} \leq T_i \end{cases}$$

Afin de contrôler le comportement rigide d'une particule adéquatement en fonction de la rigidité  $s_i$  des particules, une nouvelle force de rigidité  $\vec{f}_i^{rigide}$  est ajoutée à l'équation des SPH (équation 1.8).

$$\frac{d\vec{v}_i}{dt} = \frac{1}{\rho_i} \left( \vec{f}_i^{pression} + \vec{f}_i^{viscosite} + \vec{f}_i^{externe} + \vec{f}_i^{rigide} \right) \quad (2.2)$$

L'objectif est donc de trouver les forces  $\vec{f}_i^{rigide}$  permettant d'obtenir le niveau de rigidité désiré, c'est-à-dire le niveau de déformation désiré. Pour ce faire, un modèle permettant de déterminer la déformation subie par une particule a tout d'abord été élaboré. Une métrique nommée « erreur de déformation » ( $\varphi$ ) permet ensuite de quantifier l'écart entre la déformation d'une particule et sa déformation cible. L'approche présentée dans ce mémoire minimise cet écart en ajustant des forces de rigidité  $\vec{f}^{rigide}$  afin de préserver la rigidité désirée au sein du fluide :

$$\min_{\vec{f}^{rigide}} \varphi(particules, \vec{f}^{rigide})$$

Cette métrique est minimisée à l'aide d'un intégrateur itératif de type « prédiction-correction ». Celui-ci s'intègre aisément dans l'algorithme des SPH en ajoutant l'optimisation des forces  $\vec{f}^{rigide}$  tout juste avant l'intégration des particules (voir algorithme 2.2, lignes 6 à 10). Bien que le processus itératif demande beaucoup de temps de calcul, l'utilisation d'un *pas de temps* beaucoup plus grand rend cette approche plus viable lors du calcul de fluides extrêmement visqueux.

L'erreur de déformation et le modèle permettant de mesurer la déformation d'une particule sont décrits dans la section 2.2.1. Le processus itératif permettant d'ajuster les forces afin de minimiser l'erreur de déformation est décrit dans la section 2.2.2.

### 2.2.1 Erreur de déformation

Avant même de pouvoir contrôler la déformation d'un fluide, il est primordiale de définir un modèle permettant de mesurer cette dernière. Le modèle utilisé dans le cadre de l'approche présentée dans ce mémoire s'inspire du modèle élastique de Solenthaler *et al.* (2007). Ces derniers utilisent la position initiale d'une particule, c'est-à-dire au début de la simulation, par rapport



### Algorithme 2.2 Simulation de fluides avec changement de phases

```

1 tant que Temps simulation < Temps fin simulation faire
2   // Phase 1 : Calcul des forces (Section 2.1.1)
3   Calculer la masse volumique  $\rho_i$  et la pression  $p_i$  de chaque particule
4   Calculer les forces  $\vec{f}_i^{pression}$ ,  $\vec{f}_i^{viscosité}$  et  $\vec{f}_i^{externe}$ 
5   Calculer  $\Delta t$  (Section 3.1)
6   // Phase 2 : Optimisation des forces rigides (Section 2.2)
7   tant que Critère d'arrêt non rencontré faire
8     Prédire la position des particules (Section 2.2.2)
9     Calculer l'erreur de déformation  $\phi_i$  des particules (Section 2.2.1)
10    Ajuster  $f_i^{rigide}$  (Section 2.2.2)
11  // Phase 3 : Intégration
12  Mettre à jour la vitesse  $\vec{v}_i$  et la position  $\vec{x}_i$  des particules
13  Diffuser la température (Section 3.2)
14  Temps simulation +=  $\Delta t$ 

```

à chaque voisin comme position cible et des forces sont générées pour chaque paire d'interaction afin de faire tendre la position de la particule vers cette position cible. Cette position relative est mise à jour lorsque le fluide subit un stress trop élevé, ce qui lui permet de s'étirer. Ceci permet d'ajouter une résistance à la déformation du fluide contrôlée par la grandeur des forces de déformation. Cependant, cette approche produit un comportement élastique lorsque le fluide devient plus rigide, limitant cette approche à ce type de comportement. Toutefois, en s'inspirant de leur modèle élastique, nous pouvons émettre l'hypothèse qu'une particule n'a subi aucune déformation lors d'une itération de la simulation si sa position relative par rapport à tous ses voisins demeure inchangée. Une première version du modèle permettant de mesurer la déformation pourrait se baser sur l'écart entre la position relative de la particule par rapport à ses voisins avant ( $\vec{x}_{ij}^{rigide}$ ) et après ( $\vec{x}_{ij}$ ) l'application des forces :

$$\vec{D}_i = \sum_j \left( \vec{x}_{ij}^{rigide} - \vec{x}_{ij} \right) W_{ij} \quad (2.3)$$

La position relative  $\vec{x}_{ij}^{rigide}$  est calculée au début de l'itération courante puisque l'objectif est de calculer la déformation subie par la particule au cours de l'itération de simulation. L'utilisation du noyau  $W_{ij}$  permet de donner plus d'importance aux voisins plus rapprochés. Ce modèle

permet de déterminer si la particule  $i$  est déformée par rapport à ses voisins et d'attribuer une certaine grandeur et direction à cette déformation. Minimiser  $D_i$  pour toutes les particules permet d'obtenir un fluide sans déformation. Cependant, l'objectif est de contrôler la rigidité du fluide et de fournir une transition fluide entre l'état solide et liquide. Il est donc nécessaire de permettre un certain degré de déformation et d'être en mesure de contrôler ce dernier à l'aide d'un paramètre variant dans l'intervalle  $[0..1]$ . Pour ce faire, une métrique nommée « erreur de déformation » a été conçue. Celle-ci mesure l'écart entre la position relative  $\vec{x}_{ij}$  d'une particule et une position relative cible  $\vec{x}_{ij}^{cible}$  offrant la déformation désirée pour chaque voisin :

$$\varphi_i = \sum_j \alpha_{ij} \left( \vec{x}_{ij}^{cible} - \vec{x}_{ij} \right) W_{ij} \quad (2.4)$$

Afin d'assurer une symétrie entre les particules voisines, une rigidité  $s_{ij}$  commune est calculée pour chaque paire d'interactions entre particules voisines. Ainsi, les deux particules traiteront leur lien de la même façon. Le minimum des rigidités des deux particules est utilisé dans le but de limiter l'effet des particules plus rigides sur les particules moins rigides. Par exemple, une particule complètement liquide ne devra pas être influencée par une particule complètement rigide :

$$s_{ij} = \min(s_i, s_j)$$

Le coefficient  $\alpha_{ij}$ , utilisé lors du calcul de l'erreur de déformation (équation 2.4), permet d'accorder plus d'importance aux particules rigides lors de la minimisation de l'erreur de déformation. Il est calculé ainsi :

$$\begin{aligned} \alpha_{ij} &= s_{ij}^b \\ b &= 20 \end{aligned} \quad (2.5)$$

La valeur de  $b$  a été fixée empiriquement. En calculant l'erreur de déformation présentée dans cette section il est maintenant possible de quantifier l'écart entre la rigidité du fluide et la rigidité désirée.

La position relative cible  $\vec{x}_{ij}^{cible}$  est obtenue en interpolant les positions relatives correspondant à la déformation maximale et minimale entre les deux particules :

$$\vec{x}_{ij}^{cible} = s_{ij} \cdot \vec{x}_{ij}^{deformation\_minimale} + (1 - s_{ij}) \cdot \vec{x}_{ij}^{deformation\_maximale} \quad (2.6)$$

Selon l'équation 2.3, la position relative  $\vec{x}_{ij}^{rigide}$  correspond à l'absence de déformation, donc à la déformation minimale. De plus, l'on peut aussi supposer que la position relative de la particule  $i$  lorsqu'elle est complètement liquide  $\vec{x}_{ij}^{liquide}$  correspond à la déformation maximale (figure 2.1). L'équation 2.6 devient alors :

$$\vec{x}_{ij}^{cible} = s_{ij} \cdot \vec{x}_{ij}^{rigide} + (1 - s_{ij}) \cdot \vec{x}_{ij}^{liquide}$$

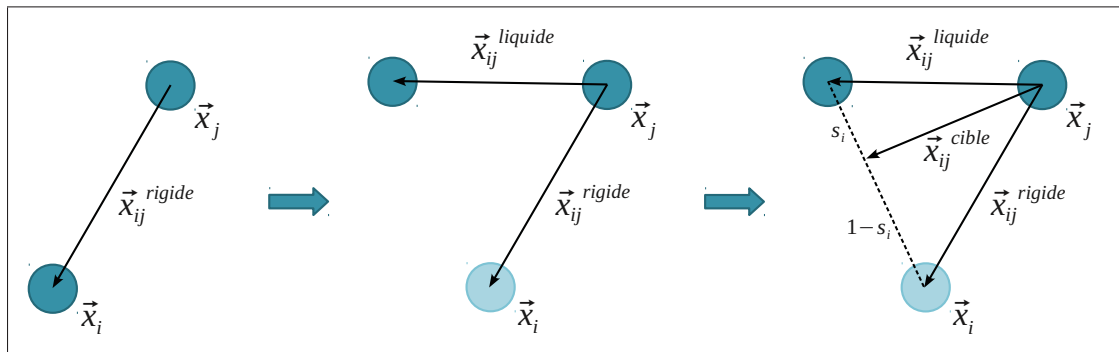


Figure 2.1 Calcul de la position cible

À gauche : La position initiale de  $x_i$  correspond à sa position rigide.

Au milieu :  $x_i$  après l'application des forces SPH correspond à sa position liquide.

À droite : La position cible est une interpolation entre ces deux positions.

## 2.2.2 Correction de la déformation

Un modèle permettant de mesurer l'erreur de déformation  $\varphi_i$  d'une particule a été présenté dans la section précédente. Ce dernier mesure l'écart entre la position d'une particule et la position cible permettant d'obtenir le niveau de déformation désiré. En diminuant cet écart pour chaque particule, le comportement du fluide tendra vers un comportement visqueux adéquat. Dans cette section, le calcul des forces rigides  $\vec{f}_i^{rigide}$  minimisant  $\varphi_i$  sera détaillé.

L'approche utilisée pour minimiser  $\varphi_i$  s'inspire de celle utilisée par Solenthaler et Pajarola (2009) afin d'assurer l'incompressibilité d'un fluide (voir la section 1.3.3.4). À l'instar de ces travaux, un intégrateur itératif de type « prédiction-correction » est utilisé afin de déterminer les forces de rigidité  $\vec{f}_i^{rigide}$  minimisant l'erreur de déformation  $\varphi_i$ . Cependant, la fonction à minimiser ici est celle définie à la section 2.2.1 et les forces calculées sont des forces de rigidité plutôt que des forces de pression. Ces forces sont initialement nulles et sont ensuite ajustées itérativement, convergeant vers la solution  $\varphi = 0$ . À chaque itération du processus d'ajustement des forces, une prédiction de l'erreur de déformation  $\varphi_i$  est calculée pour chaque particule. Cette prédiction est calculée en utilisant la position  $x_i$  des particules calculée à partir des forces SPH précédemment calculées (voir phase 1 de l'algorithme 2.2) et la valeur actuelle des forces de rigidité. Par la suite, les forces rigides  $\vec{f}_i^{rigide}$  sont ajustées en fonction de cette prédiction. Étant donné les interrelations entre les particules, il est peu probable que les forces générées donnent le minimum de  $\varphi$  à la fin d'une seule itération. Ce processus doit donc être répété jusqu'à ce que le critère d'arrêt soit respecté (voir section 2.2.3). La figure 2.2 décrit ce processus qui correspond, en fait, aux lignes 6 à 10 de l'algorithme 2.2

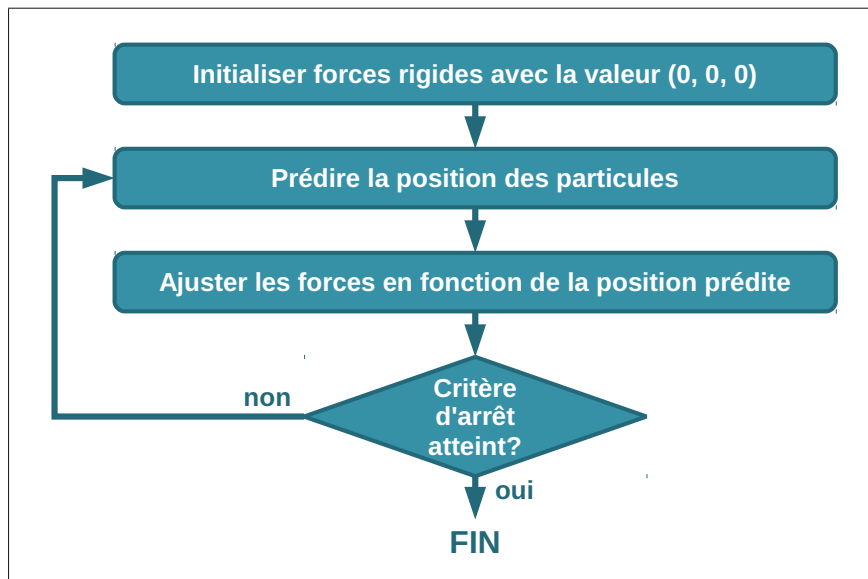


Figure 2.2 Ajustement des forces rigides  
Diagramme d'états décrivant le processus d'ajustement des forces rigides

Pour corriger ces forces, la relation entre la force rigide  $\vec{f}_i^{rigide}$  appliquée sur une particule  $i$  et l'erreur de déformation doit tout d'abord être déterminée. Entre le temps  $t$  et  $t + \Delta t$ , l'erreur de déformation  $\varphi_i$  d'une particule  $i$  évolue ainsi :

$$\begin{aligned}
 \varphi_i(t + \Delta t) &= \sum_j \alpha_{ij} \left[ x_{ij}^{cible} - x_{ij}(t + \Delta t) \right] W_{ij} \\
 &= \sum_j \alpha_{ij} \left[ x_{ij}^{cible} - (x_{ij}(t) + \Delta x_{ij}(t)) \right] W_{ij} \\
 &= \sum_j \alpha_{ij} \left( x_{ij}^{cible} - x_{ij}(t) \right) W_{ij} \\
 &\quad - \sum_j \alpha_{ij} \Delta x_{ij}(t) W_{ij} \\
 &= \varphi_i(t) - \sum_j \alpha_{ij} \Delta x_{ij}(t) W_{ij} \\
 &= \varphi_i(t) + \Delta \varphi_i(t)
 \end{aligned}$$

Ces calculs supposent que  $W_{ij}$  demeure constant entre les temps  $t$  et  $t + \Delta t$ . En supposant que la variation de position  $\Delta x_{ij}(t)$  n'est influencée que par une force correctrice  $\vec{f}_i^{corr}$  :

$$\begin{aligned}
 \vec{f} &= m \vec{a} \\
 \vec{f} &= m \frac{\Delta \vec{x}}{\Delta t^2} \\
 \Delta \vec{x} &= \frac{\vec{f} \Delta t^2}{m} \\
 \Delta \vec{x}_{ij}(t) &= \frac{\vec{f}_i^{corr} \Delta t^2}{m_i}
 \end{aligned}$$

L'incorporation de ce résultat dans  $\Delta \varphi_i(t)$  donne :

$$\begin{aligned}
 \Delta \varphi_i(t) &= - \sum_j \alpha_{ij} \Delta x_{ij}(t) W_{ij} \\
 &= - \sum_j \alpha_{ij} \left( \frac{\vec{f}_i^{corr} \Delta t^2}{m_i} \right) W_{ij} \\
 &= - \frac{\vec{f}_i^{corr} \Delta t^2}{m_i} \sum_j \alpha_{ij} W_{ij}
 \end{aligned}$$

À partir de ces équations, la force nécessaire pour produire un changement dans l'erreur de déformation d'une particule peut être calculée ainsi :

$$\vec{f}_i^{corr} = -\Delta\varphi_i(t) \frac{m_i}{\Delta t^2 \sum_j \alpha_{ij} W_{ij}}$$

Puisque l'objectif est d'obtenir  $\varphi_i(t + \Delta t) \approx 0$ ,  $\Delta\varphi_i$  est remplacé par  $-\varphi_i$  dans l'équation précédente :

$$\vec{f}_i^{corr} = \varphi_i(t) \frac{m_i}{\Delta t^2 \sum_j \alpha_{ij} W_{ij}} \quad (2.7)$$

L'équation 2.7 permet d'obtenir la force nécessaire à appliquer à la particule  $i$  afin qu'elle ait la déformation désirée, sans tenir compte de l'impact des forces appliquées sur les autres particules. Cette force correctrice est ensuite utilisée afin de corriger la force rigide  $\vec{f}_i^{rigide}$  :

$$\vec{f}_i^{rigide} \leftarrow \vec{f}_i^{rigide} + \vec{f}_i^{corr} \quad (2.8)$$

### 2.2.3 Critère d'arrêt

Tel que mentionné dans la section précédente, le calcul des forces rigides s'exécute de manière itérative jusqu'à ce qu'il ait convergé vers une solution, c'est-à-dire lorsque le critère d'arrêt a été respecté. Ce critère d'arrêt doit permettre à l'algorithme d'exécuter assez d'itérations pour fournir une solution acceptable tout en évitant les longs temps de calculs engendrés par l'exécution d'un trop grand nombre d'itérations. Contrairement à l'approche de Solenthaler et Pajarola (2009) qui arrête l'optimisation des forces lorsque l'erreur (le niveau de compression du fluide, dans le cas de cette approche) est sous un seuil prédéterminé, le critère utilisé par l'algorithme présenté dans ce mémoire se base sur la variation de l'erreur de déformation totale du fluide  $\varphi$ , c'est-à-dire la somme des erreurs de déformation  $\varphi_i$ , entre l'itération courante et l'itération précédente :

$$\varphi^{précédent} - \varphi^{courant} < 0.01\varphi^{précédent}$$

Ceci permet à l'algorithme de s'exécuter suffisamment longtemps pour converger vers une solution et de s'arrêter lorsque que l'erreur de déformation se stabilise. Toutefois, du bruit a été

observé dans la variation de  $\varphi$  lors de certaines expérimentations. Ceci a pour effet d'entraîner un arrêt hâtif de l'algorithme, ne permettant pas à ce dernier de donner une solution assez proche de l'optimale. Afin de réduire la sensibilité aux variations irrégulières de  $\varphi$ , une fenêtre de cinq itérations (fixée empiriquement) est utilisée pour évaluer la variation de  $\varphi$  à l'itération  $p$  :

$$\frac{1}{5} \sum_{l=p-9}^{p-5} \varphi^{itération_l} - \frac{1}{5} \sum_{l=p-4}^p \varphi^{itération_l} \leq 0.01 \cdot \frac{1}{5} \sum_{l=p-9}^{p-5} \varphi^{itération_l}$$

Ce nouveau critère équivaut à comparer la variation de l'erreur de déformation en utilisant la moyenne des 5 dernières itérations et la moyenne des 5 itérations les précédentes. Ceci permet d'atténuer l'effet du bruit dans la variation de l'erreur de déformation. Ce critère est donc assez souple pour laisser l'algorithme converger vers une solution tout en évitant de calculer inutilement des itérations qui influencent peu le résultat.

#### 2.2.4 Propagation des contraintes

L'optimisation des forces rigides présentée dans les sections précédentes donne de bons résultats au sein du fluide, cependant la technique présentée gère moins bien l'effet des contraintes externes telles que les collisions et les particules animées à l'extérieur du simulateur. Ceci est dû au fait que les particules ne sont affectées que par leurs voisins immédiats. Ces contraintes nécessitent donc plusieurs itérations afin de propager leurs effets adéquatement. Puisqu'une particule est affectée par plusieurs voisins, l'effet d'une contrainte sur cette dernière est moindre si la proportion de voisins affectés par cette contrainte est faible. Une particule nécessite donc plusieurs itérations pour être correctement affectée par une contrainte « globale » et l'effet s'atténue au fur et à mesure que l'on s'éloigne des particules correctement affectées par cette contrainte. La figure 2.3 montre un cas typique où l'on peut observer ce problème. Dans cette figure, un fluide, ayant la forme de la « Vénus » de Milo et dont la solidité des particules  $s_i = 1$ , est affecté par la collision avec le sol. Étant donné le nombre élevé de particules entre le sol et le sommet du fluide, l'effet de la collision avec le sol se propage moins bien sur les particules plus éloignées. Cette figure montre que le mécanisme de propagation présenté dans cette section permet de remédier à ce problème.

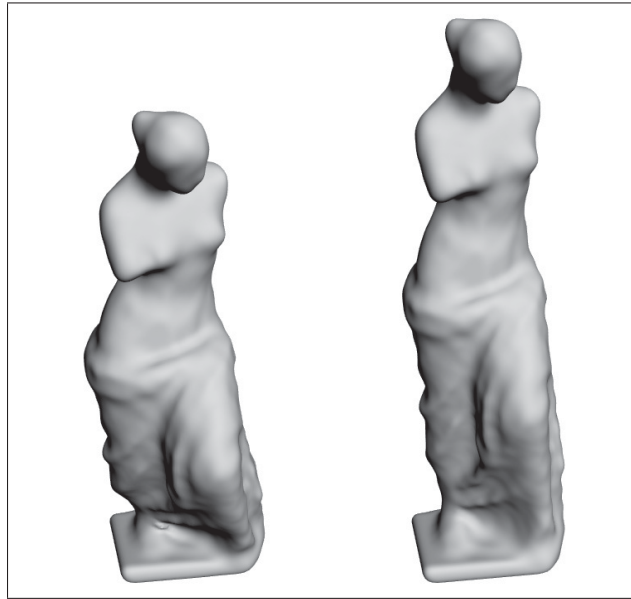


Figure 2.3 Venus de Milo avec et sans propagation des contraintes

À gauche : Sans propagation des contraintes. À droite : Avec propagation des contraintes

Ce mécanisme de propagation des contraintes s'intègre dans le processus d'optimisation des forces rigides et est basé sur la priorisation des particules selon leur état : « stabilisée » ou « non stabilisée ». Une particule est dite « stabilisée » si elle est correctement affectée par une contrainte. Tout juste avant l'exécution du processus d'optimisation des forces rigides (voir phase 2 de l'algorithme 2.2), les particules affectées directement par une contrainte, par exemple une collision avec le sol, sont identifiées comme étant « stabilisées ». Les autres particules sont considérées comme étant « non stabilisées ». Lorsqu'une particule possède un ou plusieurs voisins « stabilisés », seuls ces voisins sont utilisés lors du calcul de l'erreur de déformation et de la correction des forces rigides. Ainsi, les voisins des particules correctement affectées par les contraintes seront ajustés uniquement en fonction de celles-ci. Après avoir calculé la déformation d'une particule ayant au moins un voisin « stabilisé », les deux conditions suivantes sont testées pour cette dernière :



- La moyenne des longueurs  $\|(\vec{x}_j + \vec{x}_{ij}^{cible}) - \vec{x}_i\|$ , où  $j$  est une particule voisine « stabilisée », est sous un certain seuil :

$$\frac{\sum_j \|(\vec{x}_j + \vec{x}_{ij}^{cible}) - \vec{x}_i\|}{N} \leq 0.01 \cdot dist_0$$

- Parmi toutes les particules ayant au moins une particule « stabilisée » comme voisin, elle est parmi celles possédant un voisin ayant la plus grande rigidité  $s_i$ , si l'on ignore les voisins « non stabilisés ».

Si ces deux conditions sont respectées, la particule devient alors « stabilisée ». La première condition vérifie si la particule est stabilisée par rapport à ses voisins stabilisés. Ceci s'effectue en calculant l'écart moyen entre la position cible de la particule par rapport à ses voisins et la position prédite de la particule. Cet écart moyen est ensuite comparé à un seuil prédéfini. Dans les exemples présentés dans ce mémoire, ce seuil correspond à 1% de la distance initiale  $dist_0$  entre les particules. La seconde condition prévient des effets indésirables causés par des contraintes se propageant à travers des particules ayant des rigidités  $s_i$  différentes. Ainsi, les particules ayant une plus grande rigidité sont affectées en premier par les contraintes, prévenant ainsi le transfert de déformations moins réalistes par les régions moins rigides du fluide vers les régions plus rigides. La figure 2.4 met en évidence l'effet de cette condition. Dans cette figure, un fluide en forme de « h » dont les particules sont rigides, à l'exception d'une portion de la colonne de droite, est simulé. La figure montre que, sans cette condition, les particules du haut de la colonne de droite suivent le mouvement des particules du bas, alors qu'elles devraient rester fixes avec les autres particules du « h ». Dans le cas où une particule respecte uniquement la seconde condition, mais ne réussit pas à respecter la première condition après un certain nombre d'itérations (5 dans les exemples présentés dans ce mémoire), cette dernière est malgré tout considérée comme étant « stabilisée » afin d'éviter un blocage dans le mécanisme de propagation.

Ce processus se poursuit jusqu'à ce que les contraintes aient été entièrement propagées au sein du fluide, et ce, même si le critère d'arrêt (voir section 2.2.3) a été atteint. Lorsqu'il n'y a plus

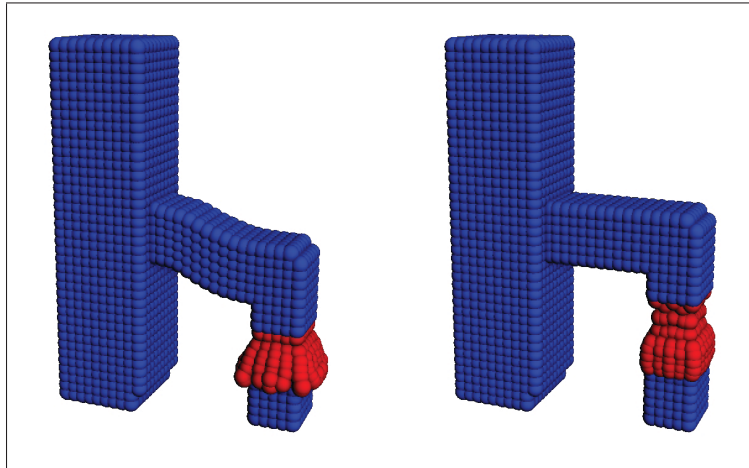


Figure 2.4 « h » avec et sans la seconde condition.

À gauche : Sans la seconde condition. À droite : Avec la seconde condition.

de particules « voisines d'une particule stabilisée », les itérations de l'algorithme d'optimisation des forces rigides continuent normalement jusqu'à ce que le critère d'arrêt habituel soit atteint. Ceci permet d'assurer la propagation des effets des contraintes à toutes les particules impliquées. Grâce à ce processus, un résultat plus satisfaisant peut être atteint avec un nombre beaucoup plus petit d'itérations tout en respectant davantage les contraintes.

## CHAPITRE 3

### STABILITÉ DE LA SIMULATION

Plusieurs facteurs peuvent influencer la stabilité d'une simulation de fluide. L'approche présentée dans le chapitre précédent permet d'éliminer l'impact de la viscosité (ou rigidité) du fluide sur la stabilité de la simulation puisque les équations ne sont plus posées sous forme de problème raide. Toutefois, d'autres sources d'instabilité existent dans la simulation. Ce chapitre présente des approches permettant de gérer ces différentes sources d'instabilité.

#### 3.1 *Pas de temps adaptatif*

La taille du *pas de temps* affecte grandement la stabilité et les temps de calcul. L'utilisation d'un *pas de temps* trop grand pour les forces appliquées au sein du fluide risque d'ajouter de l'énergie dans la simulation physique et donner un comportement erroné, voir chaotique. La figure 3.1 montre une même simulation exécutée avec un *pas de temps* fixe trop grand, la simulation est devenue chaotique alors qu'avec un *pas de temps* adaptatif elle demeure stable. D'autre part, l'utilisation d'un *pas de temps* trop petit assurera la stabilité de la simulation, mais demandera beaucoup plus d'itérations pour simuler une seconde de simulation, engendrant ainsi de grands temps de calcul.

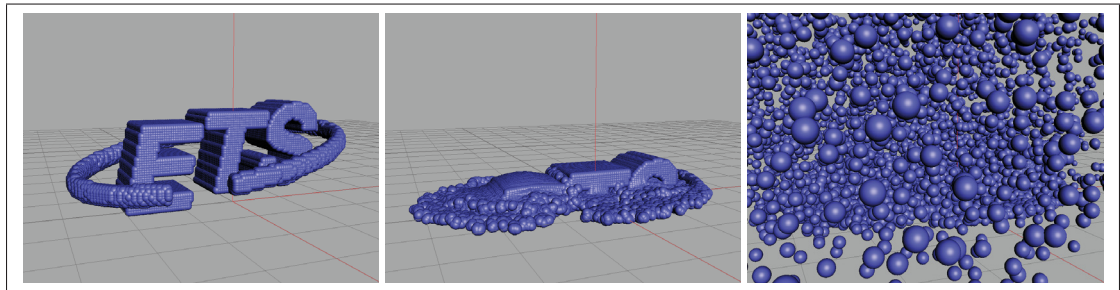


Figure 3.1 Comparaison d'une simulation avec et sans *pas de temps* adaptatif  
À gauche : simulation liquide au temps  $t = 0s$ . Au centre : avec *pas de temps* adaptatif, au temps  $t = 0.37s$ . À droite : avec *pas de temps* fixe trop grand ( $\Delta t = 0.0015$ ) au temps  $t = 0.37s$ .

Dans le projet présenté dans ce mémoire, la rigidité (ou viscosité) ne génère pas des forces sujettes à l'instabilité puisque le problème n'est pas posé sous une forme raide. En conservant uniquement le terme basé sur la vitesse de propagation d'une onde dans le fluide  $c$ , l'équation permettant de calculer le *pas de temps* de Paiva *et al.* (2006) devient :

$$\Delta t = \beta \cdot \frac{h}{\|v_{max}\| + c} \quad (3.1)$$

Ces équations dépendent de la vitesse maximale précédente et du coefficient de pression  $c$  qui correspond à la vitesse de propagation d'une onde dans le fluide. Ceci ne tient donc pas compte des forces réelles subies par le fluide. Il est possible que les forces de pression soient faibles (lorsque le fluide est au repos, par exemple). Dans ce cas, le pas de temps calculé sera plus petit que nécessaire. D'un autre côté, une particule peut avoir plusieurs voisins rapprochés dû au fait que les SPH ne sont pas incompressibles. Ceci engendrera de très grandes forces de pression nécessitant un *pas de temps* plus petit que celui calculé afin d'assurer la stabilité.

Selon la condition de CFL (équation 1.14), une particule ne peut avoir un déplacement plus grand qu'une certaine fraction  $\beta$  du rayon des particules. Les approches précédentes (Monaghan, 1989; Paiva *et al.*, 2006) utilisent une série de contraintes permettant d'approximer ce déplacement maximal en utilisant soit la force maximale, la vitesse de propagation (correspondant à une vitesse maximale théorique) ou une combinaison de la vitesse de propagation maximale et de la vitesse maximale à l'itération précédente. Toutefois, aucune de ces approches ne combine la vitesse des particules et les forces exercées sur celles-ci en une même condition. En utilisant une combinaison de la vitesse des particules et des forces appliquées sur celles-ci, il est possible d'obtenir une meilleure approximation du *pas de temps* idéal. Il n'est pas possible de calculer la vitesse maximale au temps  $t + \Delta t$  d'une particule sans préalablement calculer le *pas de temps*, puisque ce dernier est nécessaire pour connaître la variation de vitesse due à l'accélération de la particule. Cependant, il est possible de supposer que la vitesse maximale sera une combinaison de l'accélération maximale et la vitesse maximale au temps  $t$ .

$$\tilde{v}_{max}(t + \Delta t) = \|\vec{v}_{max}(t)\| + \|\vec{a}_{max}(t)\Delta t\|$$

Cette approximation donne toujours une vitesse plus grande ou égale à la vitesse maximale réelle. En utilisant cette formulation de  $\tilde{v}_{max}$  dans l'équation 1.14, l'on obtient :

$$\begin{aligned} \Delta t &= \beta \frac{h}{\|\vec{v}_{max}(t)\| + \|\vec{a}_{max}(t)\|\Delta t} \\ \|\vec{a}_{max}(t)\|\Delta t^2 + \|\vec{v}_{max}(t)\|\Delta t - \beta h &= 0 \end{aligned}$$

Résoudre cette équation en fonction de  $\Delta t$  donne :

$$\Delta t = -\frac{\|\vec{v}_{max}(t)\| \pm \sqrt{\|\vec{v}_{max}(t)\|^2 + 4\|\vec{a}_{max}(t)\|\beta h}}{2\|\vec{a}_{max}(t)\|} \quad (3.2)$$

Puisque la racine présente dans l'équation 3.2 est toujours plus grande que  $\|\vec{v}_{max}(t)\|$ , seule la partie négative de l'opérateur  $\pm$  donne une valeur positive de  $\Delta t$ . En ne conservant que le signe négatif, l'équation devient :

$$\Delta t = -\frac{\|\vec{v}_{max}(t)\| - \sqrt{\|\vec{v}_{max}(t)\|^2 + 4\|\vec{a}_{max}(t)\|\beta h}}{2\|\vec{a}_{max}(t)\|} \quad (3.3)$$

À chaque itération de la simulation, l'équation 3.3 est utilisée pour calculer le *pas de temps* idéal (voir algorithme 2.2) après avoir calculé les forces agissant sur les particules. Ceci permet le calcul d'un *pas de temps* plus grand tout en assurant la stabilité de la simulation.

### 3.2 Diffusion de chaleur implicite

La diffusion de la chaleur est une étape importante dans la simulation de phénomènes de fonte et de solidification qui permet d'ajouter du réalisme au résultat. Les approches typiques utilisant la diffusion de chaleur (Stora *et al.*, 1999; Paiva *et al.*, 2006; Solenthaler *et al.*, 2007; Chang *et al.*, 2009) sont posées sous la forme d'un problème raide et sont donc sujettes à devenir instables lorsque le *pas de temps* ou le coefficient de diffusion de la chaleur ont une grande valeur. La figure 3.2 montre cette instabilité dans une simulation de fonte du « lapin de Stanford ». Le *pas de temps* élevé lors du calcul de la première image de l'animation engendre une correction trop élevée de la température à la jonction entre les particules chaudes et les

particules froides. Ceci explique la transition discontinue de la température après une itération de la simulation (image de droite).

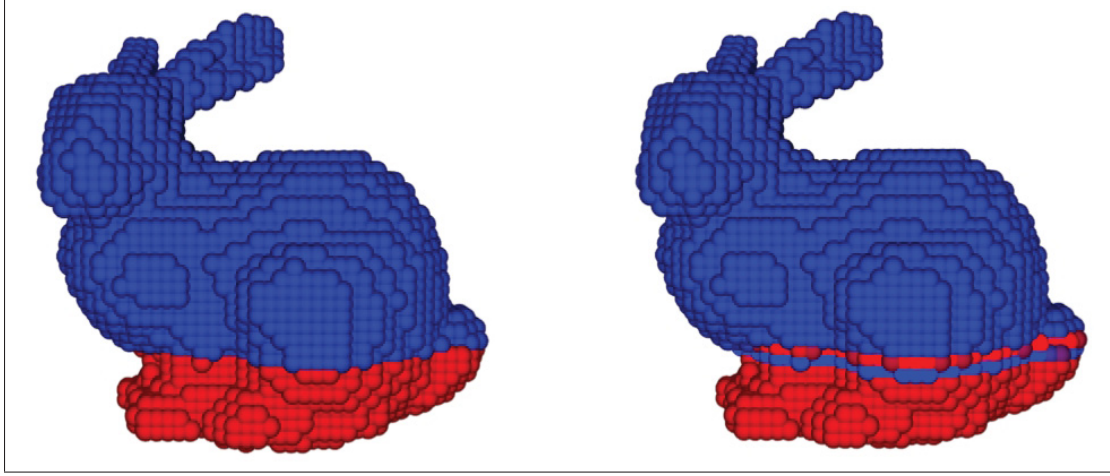


Figure 3.2 Instabilité de la diffusion de chaleur explicite

Les particules en bleu sont plus froides et les particules en rouge sont plus chaudes. À gauche : État initial d'une simulation. À droite : État après 0.041s.

Prenons, par exemple, la forme utilisant directement le Laplacien du noyau (Solenthaler *et al.*, 2007; Chang *et al.*, 2009) :

$$\begin{aligned} T_i(t + \Delta t) &= T_i(t) + k \sum_j \frac{m_j}{\rho_j} (T_{ji}(t)) \nabla^2 W_{ij} \Delta t \\ T_{ji}(t) &= T_j(t) - T_i(t) \end{aligned} \quad (3.4)$$

Il est évident que lorsque le *pas de temps*  $\Delta t$  ou le coefficient de diffusion de la chaleur  $k$  sont trop grands, l'équation 3.4 devient instable. Un simple calcul des limites le confirme :

$$\begin{aligned} \lim_{k \rightarrow +\infty} T_i(t + \Delta t) &= \begin{cases} \pm\infty & T_{ji} \neq 0 \\ 0 & T_{ji} = 0 \end{cases} \\ \lim_{\Delta t \rightarrow +\infty} T_i(t + \Delta t) &= \begin{cases} \pm\infty & T_{ji} \neq 0 \\ 0 & T_{ji} = 0 \end{cases} \end{aligned}$$

Pour solutionner ce problème, une approche implicite a été utilisée. Celle-ci se base sur la démarche utilisée par Monaghan (1997) afin d'exprimer l'application des forces de viscosité sous une forme implicite. Cette formulation traite individuellement chaque paire de particules voisines, permettant de poser le problème sous une forme implicite et de manipuler les équations afin d'éliminer les variables inconnues au temps  $t$ . En appliquant cette formulation successivement pour chaque paire de particules voisines, Monaghan a pu obtenir une viscosité inconditionnellement stable. La solution proposée dans cette section applique avec succès cette démarche au calcul de la diffusion de la chaleur. La figure 3.3 montre le processus de diffusion de chaleur pour une particule  $i$ . Pour chaque voisin de cette particule, l'échange de température

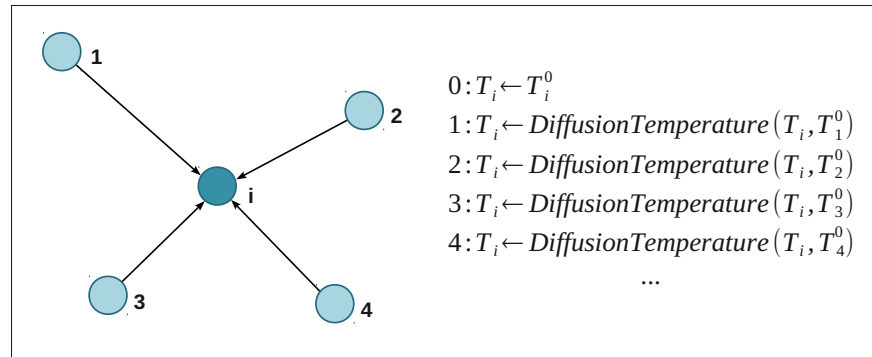


Figure 3.3 Diffusion implicite de la chaleur

est calculé à l'aide d'une formulation implicite ne tenant compte que de l'interaction entre ces deux particules. La température  $T_i$  de la particule  $i$  est ensuite mise à jour et c'est cette nouvelle température qui sera utilisée dans le calcul de diffusion de chaleur avec le prochain voisin. Ce processus se poursuit jusqu'à ce que tous les voisins aient été traités. Afin que le résultat ne soit pas influencé par l'ordre de parcours des particules, la température initiale  $T_j^0$ , avant l'application du processus de diffusion de chaleur, des particules voisines est utilisée. La suite de cette section décrit la formulation implicite de l'échange de chaleur entre deux particules voisines.

La forme implicite de l'équation 3.4 permettant de calculer la variation de température peut être obtenue en utilisant la différence de température  $T_{ji}$  au temps  $t + \Delta t$ , plutôt qu'au temps  $t$  :

$$T_i(t + \Delta t) = T_i(t) + k \sum_j \frac{m_j}{\rho_j} (T_{ji}(t + \Delta t)) \nabla^2 W_{ij} \Delta t \quad (3.5)$$

À l'aide de quelques simplifications et hypothèses, il est possible d'obtenir, à partir de cette équation, une expression inconditionnellement stable de la température  $T_i$  au temps  $t + \Delta t$ .

Tout d'abord, en considérant individuellement chaque paire de particules en interaction, il est possible de déterminer comment une particule  $i$  et une particule  $j$  s'influencent à l'aide de l'équation 3.5 :

$$T_i(t + \Delta t) = T_i(t) + k \frac{m_j}{\rho_j} (T_{ji}(t + \Delta t)) \nabla^2 W_{ij} \Delta t \quad (3.6)$$

$$\begin{aligned} T_j(t + \Delta t) &= T_j(t) + k \frac{m_i}{\rho_i} (T_{ij}(t + \Delta t)) \nabla^2 W_{ij} \Delta t \\ &= T_j(t) - k \frac{m_i}{\rho_i} (T_{ji}(t + \Delta t)) \nabla^2 W_{ij} \Delta t \end{aligned} \quad (3.7)$$

En supposant que la masse volumique ( $\rho_i$  and  $\rho_j$ ) et que le Laplacien du noyau  $\nabla^2 W_{ij}$  sont constantes entre le temps  $t$  et le temps  $t + \Delta t$ , il est possible de trouver une formulation permettant de calculer  $T_i(t + \Delta t)$  à partir de  $T_{ji}(t)$ . Bien qu'il est fort probable que ces propriétés varient entre le temps  $t$  et le temps  $t + \Delta t$ , ces suppositions sont utiles pour effectuer les simplifications nécessaires pour exprimer l'équation 3.6 uniquement en fonction de variables au temps  $t$ . Il y a donc compromis entre l'exactitude des résultats et la stabilité. Toutefois, les expérimentations ont démontré que les résultats obtenus demeurent visuellement plausibles. À l'aide des équations 3.6 et 3.7, il est possible d'exprimer  $T_{ji}(t + \Delta t)$  en fonction de  $T_{ji}(t)$ ,



éliminant ainsi toute dépendance à des variables au temps  $t + \Delta t$  :

$$\begin{aligned}
 T_{ji}(t + \Delta t) &= T_j(t + \Delta t) - T_i(t + \Delta t) \\
 &= \left[ T_j(t) - k \frac{m_i}{\rho_i} (T_{ji}(t + \Delta t)) \nabla^2 W_{ij} \Delta t \right] - \left[ T_i(t) + k \frac{m_j}{\rho_j} (T_{ji}(t + \Delta t)) \nabla^2 W_{ij} \Delta t \right] \\
 &= T_j(t) - T_i(t) - k \left( \frac{m_i}{\rho_i} + \frac{m_j}{\rho_j} \right) (T_{ji}(t + \Delta t)) \nabla^2 W_{ij} \Delta t \\
 &= T_{ji}(t) - k \left( \frac{m_i}{\rho_i} + \frac{m_j}{\rho_j} \right) (T_{ji}(t + \Delta t)) \nabla^2 W_{ij} \Delta t
 \end{aligned}$$

En déplaçant vers le côté gauche de l'équation les termes contenant  $T_{ji}(t + \Delta t)$ , l'on obtient :

$$(T_{ji}(t + \Delta t)) \left( 1 + k \left( \frac{m_i}{\rho_i} + \frac{m_j}{\rho_j} \right) \nabla^2 W_{ij} \Delta t \right) = T_{ji}(t)$$

En ne conservant que  $T_{ji}(t + \Delta t)$  du côté gauche de l'équation, l'on obtient :

$$T_{ji}(t + \Delta t) = \frac{T_{ji}(t)}{1 + k \left( \frac{m_i}{\rho_i} + \frac{m_j}{\rho_j} \right) \nabla^2 W_{ij} \Delta t} \quad (3.8)$$

Avec cette formulation de  $T_{ji}(t + \Delta t)$ , il est maintenant possible de calculer  $T_i(t + \Delta t)$  en utilisant uniquement des variables au temps  $t$  à l'aide de l'équation 3.6. Il suffit d'appliquer l'opération suivante successivement pour chaque voisin d'une particule  $i$  pour trouver sa nouvelle température  $T_i$  au temps  $t + \Delta t$  :

$$T_i \leftarrow T_i + k \frac{m_j}{\rho_j} \left( \frac{T_j(t) - T_i}{1 + k \left( \frac{m_i}{\rho_i} + \frac{m_j}{\rho_j} \right) \nabla^2 W_{ij} \Delta t} \right) \nabla^2 W_{ij} \Delta t$$

Dans le cas où un voisin a déjà été traité, c'est toujours la température  $T_j(t)$  au temps  $t$  qui est utilisée et non la température déjà mise à jour par cette opération.

Cette approche est inconditionnellement stable, permettant donc l'utilisation d'un *pas de temps* et d'un coefficient de diffusion plus grands. Ceci est démontrable en calculant les limites de

l'équation 3.6, avec l'équation 3.8, par rapport à  $\Delta t$  et  $k$  :

$$\lim_{k \rightarrow +\infty} T_i(t + \Delta t) = T_i(t) + \frac{m_j \rho_i (T_{ji}(t))}{m_i \rho_j + m_j \rho_i}$$

$$\lim_{\Delta t \rightarrow +\infty} T_i(t + \Delta t) = T_i(t) + \frac{m_j \rho_i (T_{ji}(t))}{m_i \rho_j + m_j \rho_i}$$

### 3.3 Ajustement de la masse volumique initiale

Lors de l'initialisation, les particules sont positionnées de façon régulière au sein du fluide. Cela permet d'ajuster facilement la résolution de la simulation en contrôlant l'espacement entre les particules, toutefois les particules à la surface du fluide ont moins de voisins que celles à l'intérieur du fluide (figure 3.4). Ceci résulte en une masse volumique inférieure à la masse

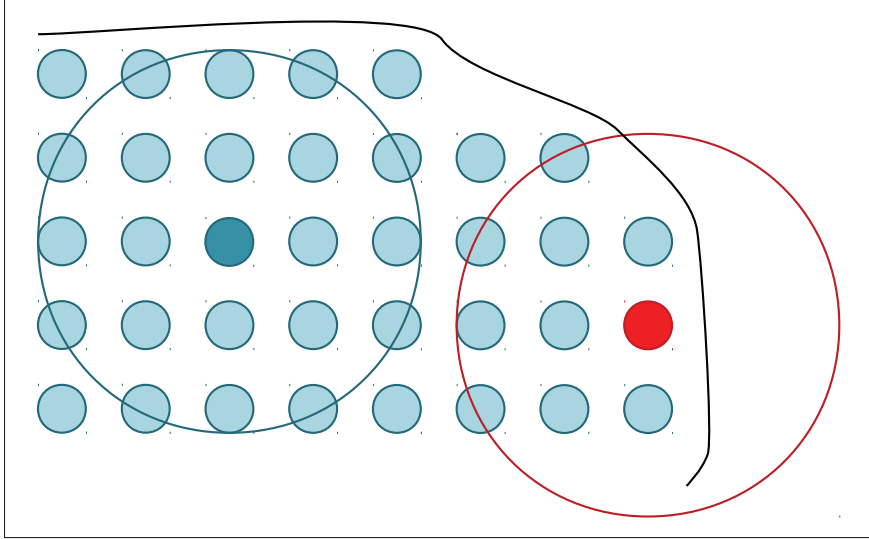


Figure 3.4 Masse volumique problématique aux frontières du fluide  
Au début de la simulation, une particule au sein du fluide (en bleu foncé) possède plus de voisins qu'une particule à la frontière du fluide (en rouge).

volumique au repos  $\rho_0$ , engendrant des forces de pression non désirées en surface. Un facteur  $\gamma_i$  multipliant la masse volumique est ajouté à l'équation 1.4 afin d'assurer que le fluide soit

initialement au repos, donc stable :

$$\rho_i = \gamma_i \sum_j m_j W_{ij} \quad (3.9)$$

$$\gamma_i = \frac{\rho_0}{\rho_i^{init}}$$

La masse volumique initiale  $\rho_i^{init}$  est calculée à l'aide de l'équation 1.4 en utilisant la configuration initiale des particules.

Cette approche a donné de bons résultats et a été utilisée pour tous les cas présentés dans ce document. Des approches alternatives pourraient toutefois être considérées. Une autre approche possible consiste en l'initialisation des particules avec une masse volumique au repos et, par la suite, la mise-à-jour de celles-ci en fonction du mouvement relatif entre les particules. Cette masse volumique pourrait être mise-à-jour selon l'équation décrite par Monaghan (1989) :

$$\frac{d\rho_i}{dt} = \sum_j m_j (\vec{v}_i - \vec{v}_j) \cdot \nabla W_{ij}$$



## CHAPITRE 4

### IMPLÉMENTATION

Ce chapitre fourni des détails sur l'implémentation de la technique décrite dans ce mémoire. Cette dernière a été utilisée afin de valider l'approche et générer les résultats de la section 5. Le fonctionnement global du simulateur est tout d'abord expliqué à la section 4.1. Par la suite, la section 4.2 présentera l'approche utilisée afin d'accélérer le parcours des voisins d'une particule. La section 4.3 explique ensuite les détails au niveau de la collision entre le fluide et les frontières de la simulation. Finalement, la section 4.4 traite des paramètres physiques utilisés dans la simulation et de leur valeur.

#### 4.1 Description du simulateur

La figure 4.1 montre le fonctionnement du simulateur. Ce dernier effectue une série d'opérations afin de mettre à jour l'état du fluide et du simulateur. Dans un premier temps, le simulateur et le fluide sont initialisés. Par la suite, un cycle d'opérations permet de faire évoluer le fluide dans le temps, par incrément de temps  $\Delta t$ , jusqu'à ce que la simulation soit terminée.

L'affichage n'est pas géré par le simulateur. Ce dernier n'a pour tâche que la mise-à-jour de l'état des particules. Cet état est ensuite enregistré sur le disque pour chaque image de l'animation. Par exemple, pour une animation contenant 24 images par secondes, l'état des particules est stocké sur le disque à chaque 0.0416s de simulation. Les particules sont ensuite chargées dans un logiciel externe, par exemple *Houdini* de *Side Effects Software* où elles sont visualisées. Il n'est donc pas nécessaire de traiter la visualisation des particules et la construction de la surface dans le simulateur.

##### 4.1.1 Initialisation

Les particules sont créées et initialisées dans un logiciel externe et, par la suite, leur position, leur vitesse et leur température sont stockées sur le disque. Lors de l'étape d'initialisation du simulateur, cet état initial de la simulation est chargé en mémoire. Lors de cette étape, les

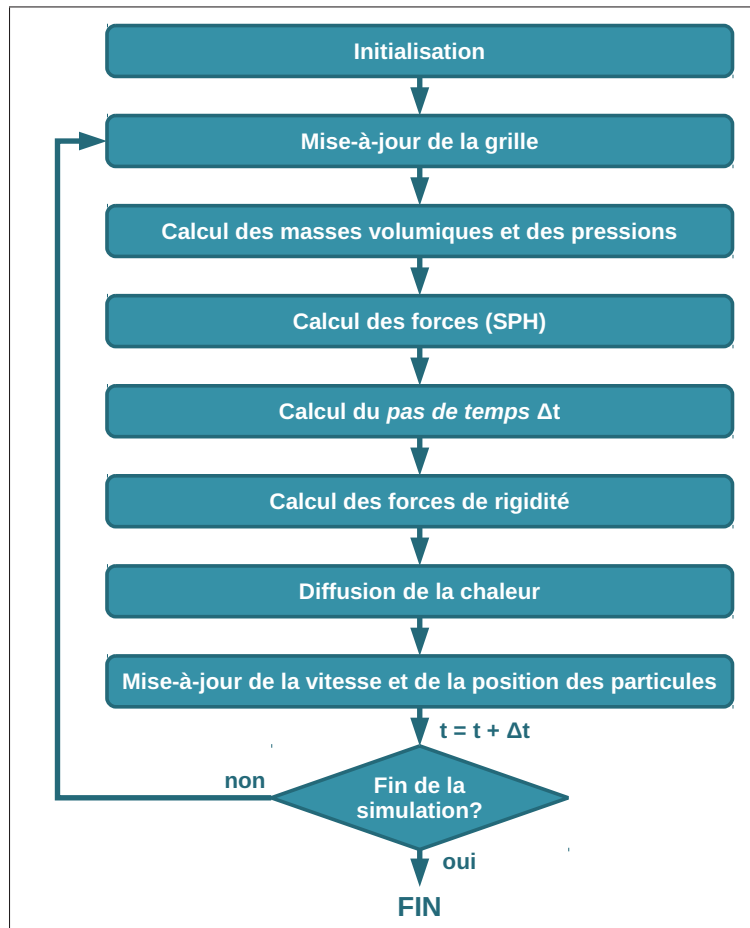


Figure 4.1 Fonctionnement du simulateur

paramètres physiques (gravité, rayon d'influence  $h$ , etc.) sont aussi chargés en mémoire à partir d'un fichier de configuration. La masse des particules est ensuite initialisée en fonction de la résolution de la simulation (voir section 4.4).

À la fin de cette opération, les structures en mémoire sont initialisées, l'état initial du fluide est chargé en mémoire et la simulation est prête à être exécutée.

#### 4.1.2 Exécution de la simulation

Une fois que la simulation est initialisée elle peut être exécutée. L'exécution de la simulation consiste en une série d'opérations exécutées en boucle jusqu'à ce que la simulation soit terminée. Au début de chaque itération de la simulation, la grille utilisée pour identifier les voisins

d'une particule est initialisée (voir section 4.2). Par la suite, la masse volumique  $\rho_i$  et la pression  $p_i$  des particules sont mises à jour à l'aide des équations 3.9 et 1.10. Ces nouvelles valeurs sont ensuite utilisées afin de calculer les forces de pression  $f_i^{pression}$ , viscosité  $f_i^{viscosité}$  et externes  $f_i^{externe}$  (équations 1.12, 1.16 et 2.1). Le *pas de temps* est ensuite calculé en fonction des forces précédemment calculées (équation 3.3). Les opérations décrites précédemment sont propres aux approches typiques des SPH.

Par la suite, les forces de rigidité  $f_i^{rigide}$  sont calculées pour chaque particule. Les informations calculées précédemment permettent de calculer la position liquide des particules et prédire leur position à chaque itération de la simulation. La position prédite des particules est calculée de la même manière que leur position finale alors que l'erreur de déformation  $\phi_i$  d'une particule est calculée à l'aide de l'équation 2.4. À partir de cette dernière, la correction à apporter aux forces rigides est finalement calculée (équations 2.7).

Une fois les forces de rigidité calculées, la température des particules est mise à jour par le processus de diffusion de chaleur (équation 3.9). Finalement, la vitesse et la position des particules sont mises à jour à l'aide de la méthode d'intégration saute-mouton (voir section 2.1.1).

## 4.2 Parcours des voisins

Déterminer les voisins d'une particule est une opération très coûteuse. Une manière simple de déterminer les voisins d'une particule est de vérifier la distance entre cette dernière et toutes les autres particules de la simulation. Cette approche, quoique simple à implémenter, est peu efficace et a une complexité de  $O(n^2)$ . Pour cette raison, il est courant d'avoir recours à un partitionnage spatial afin de diminuer le nombre de tests de distance devant être effectués. Une approche simple, mais beaucoup plus efficace, consiste en la subdivision de l'espace en une grille (figure 4.2). Chaque particule est ensuite associée à la cellule qui la contient. Lors du calcul des voisins d'une particule, il suffit donc de vérifier les particules contenues dans les cellules se trouvant dans son rayon d'influence. Ceci réduit considérablement le nombre de tests devant être effectués. L'insertion des particules dans la grille a une complexité  $O(n)$  et parcourir les voisins de toutes les particules a une complexité  $O(nc)$ , où  $c$  est le nombre

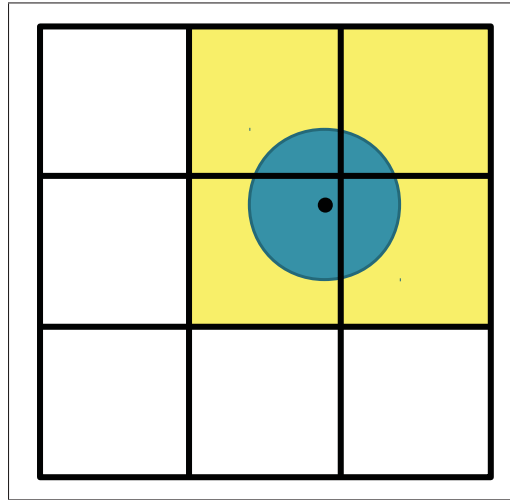


Figure 4.2 Recherche de voisins dans une grille

Seuls les voisins dans les cellules (en jaune) contenant le rayon d'influence de la particule sont traités.

moyen de particules contenues dans les cellules voisines. Les gains de performance augmentent considérablement lorsque le nombre de particules augmente.

Dans l'implémentation de la technique, la taille des cellules a été fixée au double de la longueur du rayon d'influence  $h$  des particules. Ceci offre un bon compromis entre l'espace mémoire nécessaire pour contenir l'information des cellules et les temps de calculs. Cette approche a toutefois le désavantage de limiter l'espace de simulation, le domaine de simulation étant limité par les dimensions de la grille, et d'utiliser inutilement de la mémoire lorsqu'un grand nombre de cellules sont vides. Afin d'éliminer ces limitations, une grille avec hachage spatial peut être utilisée (Teschner *et al.*, 2003), toutefois, l'approche décrite dans cette section donne de bons résultats dans le cas de simulation de fonte et solidification.

Puisque le parcours des voisins d'une particule est effectué plusieurs fois par itération de la simulation et que la position des particules ne change pas au cours d'une même itération, une liste des voisins et de leur distance est conservée pour chaque particule. Ces données sont calculées lors du premier parcours des voisins, c'est-à-dire lors du calcul de la masse volumique des particules, et sont ensuite utilisées lors des prochaines opérations, accélérant ainsi les temps de calculs.



### 4.3 Détection de collision

L'implémentation actuelle du simulateur ne supporte pas la détection de collision avec des géométries complexes. Seules les collisions avec les limites du volume de simulation (prédéterminées par l'utilisateur) sont gérées. Ces limites consistent en 6 plans alignés sur les trois axes, formant une boîte contenant le fluide. Lors de la prédiction de la position des particules et lors du calcul de la nouvelle position des particules, un test est effectué pour chaque particule afin de s'assurer qu'elle ne traverse pas ces plans. La position et la vitesse des particules traversant ces plans sont corrigés tel que décrit dans la section 2.1.2.

Étant donné la simplicité actuelle de la détection de collision, les temps de calcul supplémentaires engendrés par ces vérifications sont négligeables. Pour cette raison, aucun mécanisme de division spatiale n'a été utilisé afin d'accélérer ces calculs. Cependant, cela permettrait d'améliorer les temps de calcul liés à la détection de collisions dans le cas où l'on désirerait supporter les collisions avec des géométries complexes.

### 4.4 Paramètres

Dans les exemples présentés à la section 5, la longueur, la hauteur et la largeur du volume de simulation varient entre 1m et 5m. Les particules sont normalement disposées de manière régulière (sous forme de grille) au sein du fluide. Une distance fixe sépare donc les particules le long des axes principaux. Cette distance peut être contrôlée et détermine la résolution de la simulation. Le rayon d'influence  $h$  des particules est égal au double de cette distance. Les autres paramètres physiques, à l'exception de la masse des particules, demeurent fixes indépendamment de la résolution de la simulation. La valeur de ces paramètres est donnée dans le tableau 4.1.

Toutes les particules ont la même masse  $m_i$ . Celle-ci est initialisée de sorte à ce qu'une particule typique, selon la configuration initiale sous forme de grille, ait une masse volumique  $\rho_i$  au repos

Tableau 4.1 Valeurs des paramètres physiques

Paramètre	Valeur
$k$	100.00
$c$	$\sqrt{k} = 10.00$
$\alpha_v$	0.1
$T_{solide}$	0.00
$T_{liquide}$	100.00
$\rho_0$	998.23
$\beta$	0.2

lorsque  $\gamma_i = 1$ . La masse  $m$  des particules peut donc être calculée à l'aide de l'équation 1.4 :

$$\begin{aligned}\rho_0 &= \sum_j m W_{ij} \\ \rho_0 &= m \sum_j W_{ij} \\ m &= \frac{\rho_0}{\sum_j W_{ij}}\end{aligned}$$

## CHAPITRE 5

### RÉSULTATS

L'approche proposée a été validée à l'aide de plusieurs scènes ayant différentes valeurs de  $s_i$  dans un contexte de fonte et solidification. Ce chapitre présente les résultats obtenus lors de la simulation de ces différentes scènes. Dans un premier temps, la configuration des exemples est expliquée. Dans un second temps, les résultats obtenus en utilisant la technique présentée dans ce mémoire sont expliqués. Finalement, ces résultats sont comparés à ceux des travaux précédents.

#### 5.1 Présentation des résultats

La figure 5.1 montre deux volumes d'eau tombant et se frappant ensuite (lorsqu'ils sont liquides). La même simulation a été effectuée avec différents niveaux de solidité  $s_i$  qui sont les mêmes pour toutes les particules d'une même simulation. Cette figure montre que la technique permet de simuler adéquatement des fluides ayant différentes rigidités. Au fur et à mesure que la valeur de  $s_i$  augmente, le liquide prend un aspect plus rigide, ou plus visqueux. Lorsque la rigidité est de 1, les deux volumes de fluide conservent leur forme et ne vont donc jamais entrer en collision, ce qui est le comportement attendu. Les statistiques de ces simulations, ainsi que des autres simulations présentées dans cette section, sont présentées dans le tableau 5.1. En général, le calcul des forces rigides  $f_i^{rigide}$  est l'opération la plus coûteuse en termes de temps de calcul lors d'une itération de la simulation (85% et plus des temps de calcul). Toutefois, ce rapport diminue considérablement lorsque le fluide est complètement liquide. Ceci est dû au fait que les particules complètement liquides sont ignorées lors de certains calculs des forces rigides  $f_i^{rigide}$ . Par ailleurs, le cas contraire où les particules sont toutes solides donne de meilleurs temps de calculs par image de l'animation. Ceci est dû au grand *pas de temps* utilisé puisque la vitesse des particules et les forces de pression agissant entre elles sont faibles. Par conséquent, ceci indique que les performances de l'approche sont aussi influencées par les particules moins rigides qui peuvent nécessiter un *pas de temps* plus petit lorsque leur vitesse ou les forces agissant sur ces dernières deviennent plus grandes.

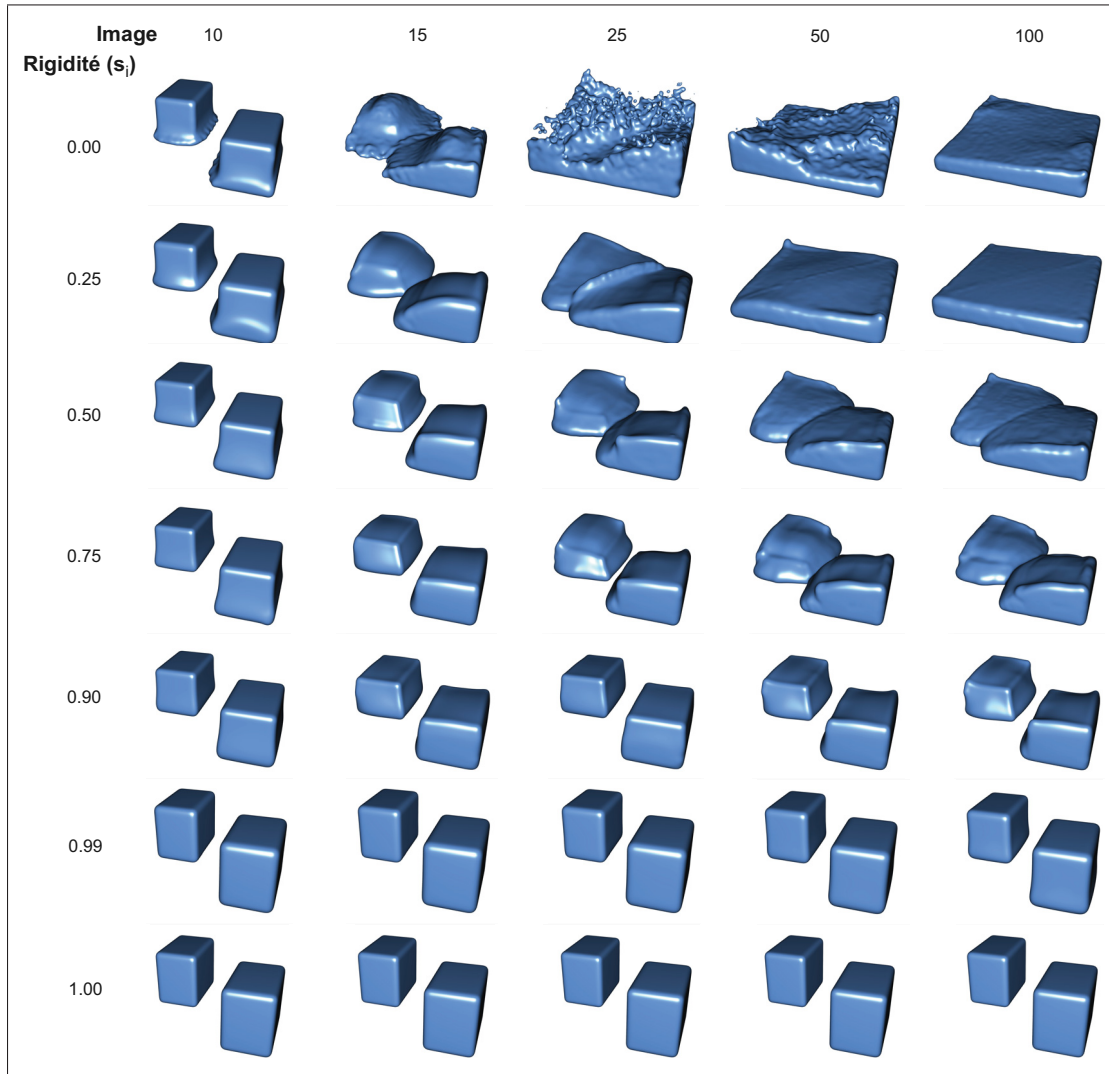


Figure 5.1 Fonte de deux blocs avec des rigidités  $s_i$  différentes

Les bras de l'Armadillo de Stanford sont liquéfiés dans la figure 5.2. Au début de la simulation, la température des particules aux épaules est initialisée à 0 degrés (solide) alors que la température au bout des bras est initialisée à 10 degrés (très visqueux). Entre ces deux points, la température varie linéairement. Afin de limiter le nombre de particules simulées, seules les particules à partir des épaules sont simulées, les particules du corps, de la tête et des jambes restent fixes. Cet exemple montre tout d'abord que le comportement visqueux est plausible et cohérent dans un contexte où la rigidité  $s_i$  des particules varie au sein d'un même fluide. De plus, il vient montrer qu'il est possible de coupler cette approche avec des particules non

contrôlées par la simulation. Les particules à la jonction entre les épaules (simulées par l'approche présentée dans ce mémoire) et le reste du corps (fixes) sont correctement influencées par ces dernières. Cependant, le réalisme de l'animation pourrait être amélioré en traitant les mouvements rotationnels lorsque le fluide devient plus rigide. L'exemple montre que les particules visqueuses tombent directement au sol en ligne droite au lieu de se rapprocher du corps sous l'effet des forces entre les particules.

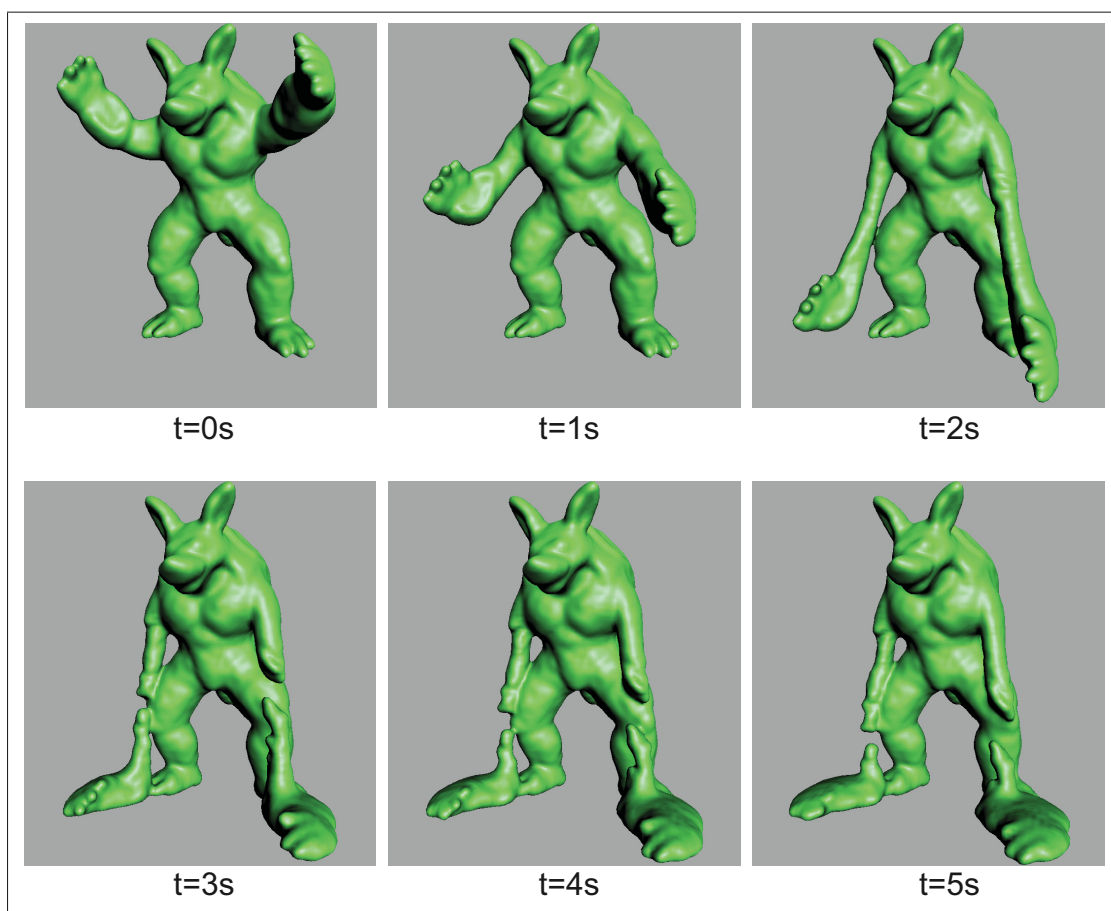


Figure 5.2 Fonte des bras de l'Armadillo de Stanford

Un autre exemple, cette fois avec le lapin de Stanford, montre la fonte d'un fluide à partir du bas (figure 5.3). La température des particules du bas est initialisée à 20 degrés, alors qu'elle est initialisée à 0 degrés (donc complètement rigide) pour les autres particules. Pendant l'animation, les particules du dessus, dont la température  $T_i = 0$ , conservent la même configuration

et suivent le mouvement des particules en fonte sous celles-ci. La diffusion de la température permet d'obtenir plus de réalisme en adoucissant la transition entre les régions plus chaude et plus froide. La fonte s'exécute lentement sur plusieurs secondes, montrant l'efficacité de la technique pour simuler un fluide très visqueux.

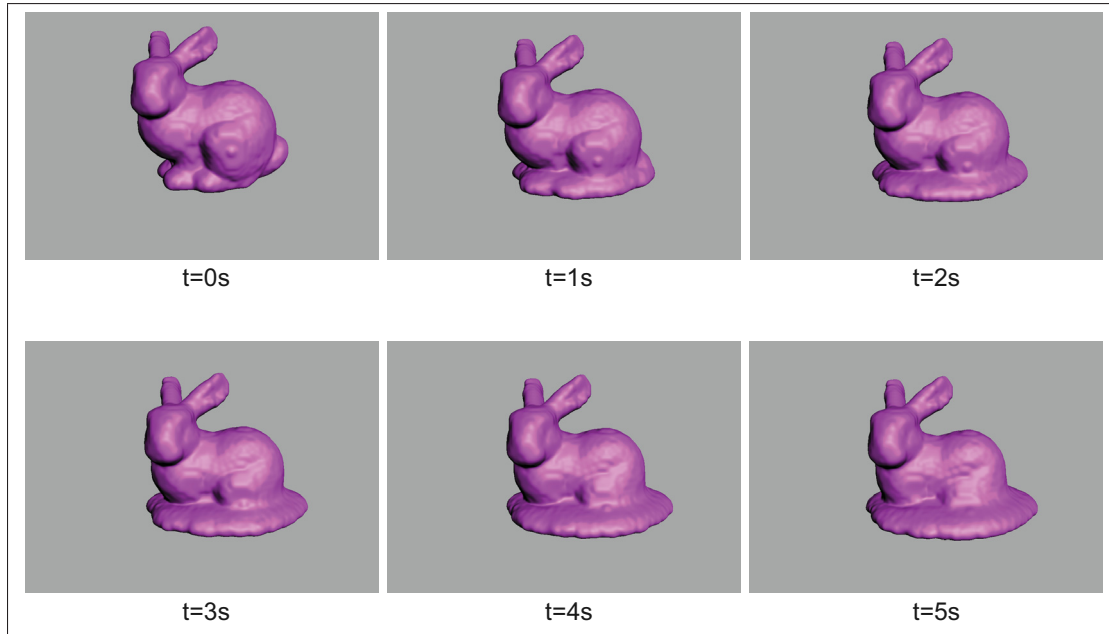


Figure 5.3 Fonte du lapin de Stanford

La figure 5.4 montre un test effectué sur un volume ayant la forme d'un « h ». Ce test a pour but de valider la propagation des contraintes dans différents cas limites, notamment lorsque deux contraintes ayant traversé un fluide dont la rigidité varie se rencontrent. Dans le premier cas, une portion de la colonne de droite est initialisée avec une température de 20 degrés, alors que le reste du fluide est initialisé avec une température de 0 degrés. Le second cas est semblable au premier, toutefois une portion de la colonne de gauche a été initialisée avec une température de 10 degrés. Les résultats montrent que la propagation donne les résultats escomptés. Dans l'image de gauche, la section fondue s'est bel et bien détachée du reste du volume et le reste est resté rigide tel que désiré. Dans l'image de droite, le haut du volume suit la fonte du côté le plus rigide (colonne de gauche) et le fluide fondant plus rapidement (colonne de droite) se détache. Ces résultats montrent que la propagation s'adapte bien aux fluides ayant une rigidité

variable. Pour ces deux simulations, le rapport entre le temps de calcul des forces rigides et le temps total (tableau 5.1) est beaucoup plus grand. Ceci s'explique par la propagation des contraintes qui doit franchir un plus grand nombre de particules avant de se terminer dans ces deux cas. La distance, en nombre de particules, devant être parcourue par une contrainte est donc un facteur important influençant les temps de calcul.

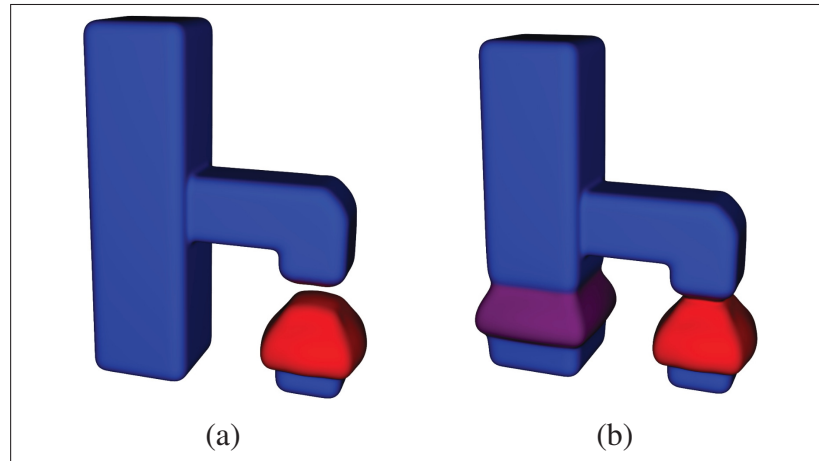


Figure 5.4 Exemples du « h ».

À gauche : Une section de la colonne de droite est fondue ( $T_i = 20$ ). À droite : Une section supplémentaire est fondue ( $T_i = 10$ ) plus lentement sur la colonne de gauche

Ces résultats montrent que l'objectif principal a été atteint. Le traitement de la rigidité du fluide utilise des équations non raides, éliminant ainsi la restriction sur le *pas de temps*. Bien qu'une itération du simulateur soit beaucoup plus coûteuse en termes de temps de calcul, ceci permet de simuler des fluides extrêmement visqueux dans des temps raisonnables puisqu'un nombre considérablement plus faible d'itérations est nécessaire pour exécuter la simulation. De plus, la stabilité du simulateur a été améliorée par une nouvelle formulation implicite de la diffusion de la chaleur et une nouvelle manière de calculer le *pas de temps* adaptatif.

## 5.2 Comparaison avec les travaux antérieurs

Le tableau 5.2 compare les temps de calcul entre l'approche présentée dans ce document et l'approche traditionnelle utilisant l'équation 1.13 (Müller *et al.*, 2005). Une même simulation a été exécutée plusieurs fois avec l'approche traditionnelle en variant la viscosité  $\mu_i$  des parti-

Tableau 5.1 Statistiques des exemples

Statistiques des exemples : Nombre de particules, temps moyen par image, temps moyen par *pas de temps*, nombre moyen de *pas de temps* par image, ratio entre le temps du calcul des forces  $f_i^{rigide}$  et le temps total de la simulation

	Figure	Nb. part.	Temps moy. image	Temps moy. p. d. t.	Nb. p. d. t. image	Ratio t rigide t total
(5.1)	Blocs $s_i = 0.00$	52.7k	17.0s	1.0s	16.2	0.33
(5.1)	Blocs $s_i = 0.25$	52.7k	88.1s	9.0s	9.7	0.88
(5.1)	Blocs $s_i = 0.50$	52.7k	90.2s	9.9s	9.0	0.89
(5.1)	Blocs $s_i = 0.75$	52.7k	56.8s	7.4s	7.6	0.91
(5.1)	Blocs $s_i = 0.90$	52.7k	94.5s	14.5s	6.4	0.92
(5.1)	Blocs $s_i = 0.99$	52.7k	65.5s	17.1s	3.8	0.94
(5.1)	Blocs $s_i = 1.00$	52.7k	23.5s	21.4s	1.1	0.97
(5.3)	Lapin de Stanford	52.4k	480.1s	50.3s	9.5	0.97
(5.2)	Armadillo de Stanford	17.1k	165.2	14.1	11.6	0.92
(5.4a)	« h » (a)	55.7k	619.7	49.3	12.5	0.97
(5.4b)	« h » (b)	55.7k	848.7	53.1	15.9	0.98

Tableau 5.2 Comparaison avec la viscosité traditionnelle

Viscosité traditionnelle			Notre approche		
$\mu_i$	$\Delta t$	Temps total	$s_i$	$\Delta t$ moy.	Temps total
1 000	$6.1 \times 10^{-4}$ s	47.80 min	0.75	$4.05 \times 10^{-3}$ s	85.03 min
10 000	$6.1 \times 10^{-5}$ s	484.81 min	0.92	$4.80 \times 10^{-3}$ s	103.70 min
100 000	$5.9 \times 10^{-6}$ s	4474.26 min	0.98	$6.36 \times 10^{-3}$ s	161.65 min

cules à chaque fois. Dans le cadre de cette simulation, le fluide est initialisé de sorte à prendre la forme des lettres « ETS » et toutes les particules sont initialisées avec la même viscosité  $\mu_i$ . Le fluide est ensuite simulé jusqu'au temps  $t = 4.0s$ . Afin d'obtenir un résultat correct lorsque la viscosité est élevée, le *pas de temps* doit être ajusté en conséquence. En effet, la condition CFL (voir sections 1.3.3.1 et 3.1) permet d'assurer la stabilité de la simulation, mais lorsque la viscosité est très élevée, un *pas de temps* plus petit doit être utilisé afin que les grandes forces de viscosité générées n'apportent pas une correction trop grande sur la vitesse des particules. Pour ces simulations, un *pas de temps* fixe a été utilisé. Ce dernier a été déterminé par « essai-erreur », avec une précision de deux chiffres significatifs. Ceci signifie, par exemple,



que l'utilisation d'un *pas de temps* de  $6.2 \times 10^{-4}$  donne un résultat erroné dans la simulation où  $\mu_i = 1000$ , alors qu'un *pas de temps* de  $6.1 \times 10^{-4}$  donne un bon résultat. Pour chaque viscosité  $\mu_i$ , une simulation avec une rigidité  $s_i$  donnant un résultat semblable a été exécutée avec notre approche afin de comparer les temps de calcul. La figure 5.5 montre le résultat des simulations au temps  $t = 4.0s$ .

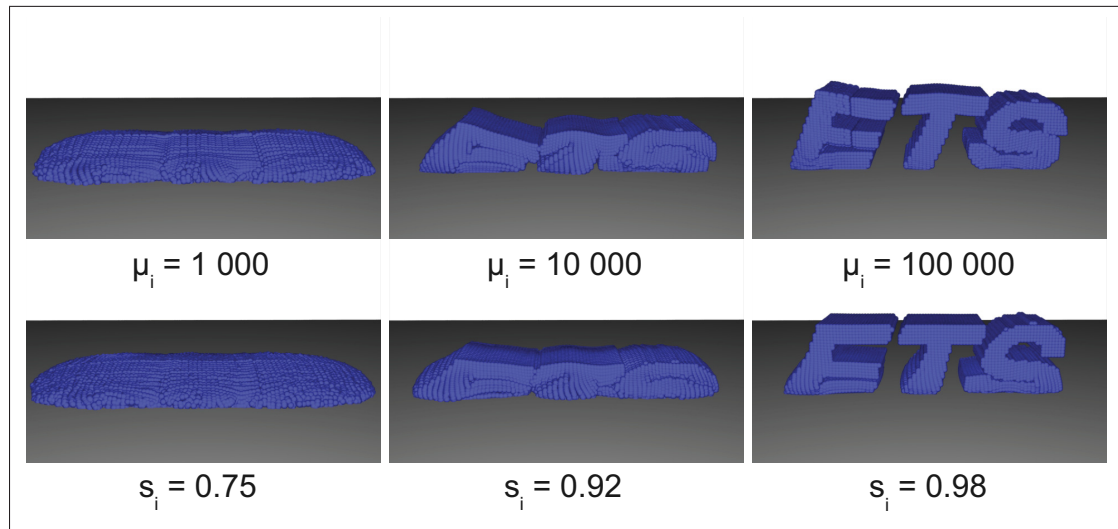


Figure 5.5 Comparaison entre la viscosité traditionnelle et notre approche.  
En haut : viscosité traditionnelle. En bas : Notre approche.

Les résultats montrent que l'approche traditionnelle est plus rapide lorsque la viscosité du fluide est faible. Toutefois, les temps de calcul varient linéairement en fonction de la viscosité  $\mu_i$  des particules, alors que le comportement rigide perçu augmente beaucoup plus lentement. Obtenir un viscosité extrême exige donc une très grande valeur de  $\mu_i$  et, par conséquent, engendre de très long temps de calculs. Notre approche est donc plus rapide lorsque la viscosité (ou rigidité) du fluide est très grande ( $s_i \geq 0.92$  et  $\mu_i \geq 10000$ ).

Les approches de Paiva *et al.* (2006, 2009) et Chang *et al.* (2009) permettent aussi de simuler des fluides hautement visqueux, toutefois le même problème persiste quant à la grandeur du *pas de temps* lorsqu'ils tentent de simuler des fluides extrêmement visqueux. Ceci rend ces approches non viables lorsque le comportement du fluide s'approche de celui d'un rigide. Cependant, elles permettent d'obtenir un comportement plus réaliste puisqu'elles sont basées

sur les équations des fluides Non-Newtoniens. Dans le premier cas ceci leur permet d'obtenir des fluides viscoplastiques, alors que dans le second cas ceci leur permet d'obtenir des fluides viscoélastiques. Ces approches conservent la vitesse angulaire des particules, même lorsque le fluide devient presque rigide, ce qui n'est pas le cas avec notre approche. Ce problème est visible dans la figure 5.2, où les bras de l'armadillo tombent en ligne droite au lieu de se refermer sur son corps.

Du côté des approches eulériennes, les travaux de Carlson *et al.* (2002) permettent la simulation de fluides extrêmement visqueux de manière stable. L'approche présentée dans ce document permet d'obtenir des comportements extrêmement visqueux semblables à ceux obtenus dans ces travaux. Toutefois, ceux-ci nécessitent moins de temps de calcul. De plus, les travaux de Batty et Bridson (2008) ont permis d'élargir l'éventail de comportement visqueux pouvant être simulés en incorporant le support de mouvements rotationnels dans les approches eulériennes. Les approches eulériennes demeurent donc plus rapides et donnent un comportement plus plausible. Afin de concurrencer les approches eulériennes, il est donc nécessaire de diminuer les temps de calcul et d'ajouter le support des mouvement rotationnels au modèle présenté. Néanmoins, les travaux décrits dans ce document apportent une amélioration considérable aux approches lagrangiennes en matière de simulation de fluides aux comportements extrêmement visqueux.

## CHAPITRE 6

### DISCUSSION

Ce chapitre présente une revue de la technique décrite dans ce document sous forme de discussion traitant des avantages et des limitations de la technique développée.

#### 6.1 Avantages

L'approche présentée dans ce document amène plusieurs avantages. Tout d'abord, il est possible de simuler des comportements extrêmement visqueux qui étaient jusqu'alors pratiquement impossibles à simuler dans des temps raisonnables à l'aide d'une approche lagrangienne. Cette dernière n'est pas basée sur la physique, mais les résultats sont visuellement plausibles. Bien que l'approche décrite nécessite beaucoup de temps de calcul afin de calculer les forces rigides, celle-ci permet l'utilisation d'un *pas de temps* beaucoup plus grand, surpassant ainsi les approches traditionnelles lorsque le fluide devient presque complètement rigide puisqu'un nombre beaucoup plus faibles d'itérations est requis à ce moment. Le contrôle de la rigidité (ou viscosité) de la simulation s'effectue à l'aide d'un paramètre intuitif variant dans l'intervalle 0 (liquide) et 1 (solide), il est donc plus facile de déterminer quelle valeur donnera la bonne rigidité au fluide. De plus, cette approche, utilisant un intégrateur de type prédiction-correction, est stable lorsqu'un très grand *pas de temps* est utilisé. Le *pas de temps* est donc uniquement limité par la grandeurs des autres forces appliquées sur la simulation. L'utilisation d'un plus grand pas de temps rend toutefois l'équation de diffusion de chaleur davantage sujette à être instable. L'approche implicite présentée dans ce mémoire permet d'éliminer cette source d'instabilité tout en fournissant des résultats plausibles. Par ailleurs, une nouvelle manière de calculer le *pas de temps* adaptatif permet d'assurer la stabilité de la simulation avec des meilleurs temps de calculs. Tout cela permet la simulation stable de fluides extrêmement visqueux à l'aide d'une approche lagrangienne. Il est à noter que l'approche de diffusion et le calcul de pas de temps adaptatif peuvent être utilisés dans d'autres approches, courantes ou futures, lagrangiennes ou eulériennes.

## 6.2 Limitations

La plus grande limitation de cette technique est l'absence des mouvements de rotation lorsque le fluide est plus rigide. Ceci est dû au modèle utilisé pour mesurer la déformation qui utilise le positionnement relatif des particules. Ceci ne permet donc que des mouvements de translation entre les particules.

Une seconde limitation existe au niveau des temps de calcul. Tout d'abord, notre approche permet l'utilisation d'un *pas de temps* considérablement plus grand qu'avec les approches traditionnelles lorsque le fluide est presque rigide. Cet écart compense pour les longs temps de calcul des forces rigides, ce qui rend notre approche plus rapide dans ce cas. Toutefois, lorsque le fluide devient moins rigide, l'écart entre les *pas de temps* est plus faible et ne compense plus pour ces longs temps de calcul, rendant ainsi plus rapide les approches traditionnelles. Par ailleurs, les temps de calcul augmentent considérablement lorsque le nombre de particules augmente, notamment à cause du *pas de temps* plus petit nécessaire et du nombre de particules à traiter, ce qui est également le cas dans les approches de type SPH typiques. Cependant l'augmentation des temps de calculs est aussi due au plus grand nombre de particules à parcourir lors de la propagation des contraintes, augmentant le nombre d'itérations nécessaires au calcul des forces rigides. Malheureusement, dans un contexte de production, beaucoup de particules sont nécessaires afin de pouvoir bien représenter la surface des objets, ce qui rend les temps de calculs très longs.

## CHAPITRE 7

### TRAVAUX FUTURS

Tel que mentionné dans la section précédente, il existe deux limitations principales à l'approche présentée dans ce mémoire : Les mouvements de rotation qui ne sont pas pris en compte par le modèle et les temps de calcul.

Dans le premier cas, le modèle peut être amélioré afin de tenir compte de ces mouvements de rotation. Il serait intéressant de tester un modèle ne tenant compte que de la distance entre les particules. Cette distance ne tient pas compte de l'orientation des particules, mais permettrait de conserver la configuration d'un regroupement de particules. Une autre approche possible consisterait en la création d'une heuristique permettant d'approximer le mouvement de rotation subi par les voisins et ensuite de transformer la position relative des voisins conformément à ce mouvement. Cela permettrait d'ajouter du réalisme au comportement des particules extrêmement visqueuses.

Dans le second cas, les temps de calcul rendent cette approche difficilement utilisable dans un contexte de production où un très grand nombre de particules est requis pour représenter la surface qui est normalement construite à partir des particules. Une approche multirésolution pourrait être utilisée afin de réduire le nombre de particules tout en conservant un bon niveau de détails en surface. La technique présentée dans ce document pourrait aussi être jumelée avec une autre représentation de la surface qui serait mise-à-jour en fonction du mouvement des particules. Ceci permettrait d'obtenir et conserver des détails au niveau de la surface avec un minimum de particules. Outre cela, des améliorations pourraient être apportées à la technique afin d'améliorer les temps de calcul sur un grand nombre de particules. Tout d'abord, le processus de propagation des contraintes demande beaucoup d'itérations d'optimisation. Il serait intéressant de tenter de développer une heuristique permettant de propager l'impact d'une contrainte, sous forme de forces ajoutées aux particules, avant même de calculer les forces de rigidité. Ainsi, ces forces seraient utilisées uniquement afin de corriger les déformations locales du fluide.

Un autre problème lié aux temps de calcul est la dépendance qu'à le *pas de temps* sur les forces de pression agissant sur le fluide. Coupler l'approche présentée dans ce mémoire à une autre approche n'ayant pas de restriction au niveau du *pas de temps* en fonction des forces de pression permettrait l'utilisation d'un *pas de temps* plus grand. L'approche des PCISPH serait un bon candidat, tout comme l'utilisation d'un intégrateur de type « prédiction-relaxation » (inspiré de Clavet *et al.* (2005)). Par ailleurs, la parallélisation de l'algorithme sur le GPU permettrait d'étendre la technique à un très grand nombre de particules.

D'autre part, des techniques permettant d'améliorer la stabilité de la simulation ont été présentées dans ce document. Bien que les résultats obtenus permettent de constater cette stabilité, il serait nécessaire d'effectuer des tests plus complets comparant ces dernières aux autres techniques se trouvant dans la littérature.

Il serait intéressant de mener des tests avec des sujets humains afin de comparer la rigidité perçue par les utilisateurs par rapport à la rigidité spécifiée dans le système. À partir de ces données, un modèle serait construit afin de rendre plus intuitif le contrôle de la rigidité des particules.

Finalement, le résultat de la diffusion implicite de la température dépend de l'ordre de parcours des voisins des particules. Il serait nécessaire d'effectuer davantage de tests afin de vérifier l'impact que cela a sur le résultat.

## CONCLUSION

Dans le cadre du projet décrit dans ce mémoire, une approche permettant de simuler des fluides presque rigides à l'aide d'une simulation lagrangienne a été développée. Celle-ci offre une transition fluide entre l'état liquide et solide à l'aide d'un seul paramètre facile à contrôler variant entre 0 (liquide) et 1 (solide). Afin d'obtenir ce résultat, un modèle permettant de mesurer la déformation d'un fluide a été établi. Par la suite, une métrique mesurant l'écart entre la déformation du fluide et la déformation désirée a été conçue. Cette métrique est ensuite minimisée en ajustant des forces correctrices appliquées au sein du fluide à l'aide d'un intégrateur de type prédiction-correction. Cette méthodologie permet d'obtenir des forces stables, peu importe la grandeur du *pas de temps*, donnant la rigidité désirée du fluide. Cependant, la stabilité de la simulation est aussi affectée par d'autres éléments de la simulation. À cet effet, une formulation implicite de la diffusion de la chaleur a été élaborée et une nouvelle manière de calculer le *pas de temps* a été développée. Tout cela permet la simulation de phénomènes de fonte et solidification. Toutefois, le support des mouvements rotationnels au sein du fluide rigide devra être ajouté et les temps de calculs devront être améliorés afin de rendre cette méthode utilisable dans un contexte de production cinématographique et ainsi compétitionner avec les approches eulériennes existantes.





## BIBLIOGRAPHIE

- Batty, C. et R. Bridson. 2008. « Accurate viscous free surfaces for buckling, coiling, and rotating liquids ». In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. (Aire-la-Ville, Switzerland, Switzerland 2008), p. 219–228. Eurographics Association.
- Brackbill, J. et H.M. Ruppel. 1986. « FLIP : A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions ». *Journal of Computational Physics*, vol. 65, n° 2, p. pages 314 - 343.
- Bridson, R., 2008. *Fluid Simulation*. Natick, MA, USA : A. K. Peters, Ltd.
- Carlson, M., P. J. Mucha, R. B. Van Horn, III, et G. Turk. 2002. « Melting and flowing ». In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. (New York, NY, USA 2002), p. 167–174. ACM.
- Chang, Y., K. Bao, Y. Liu, J. Zhu, et E. Wu. 2009. « A particle-based method for viscoelastic fluids animation ». In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*. (New York, NY, USA 2009), p. 111–117. ACM.
- Clavet, S., P. Beaudoin, et P. Poulin. 2005. « Particle-based viscoelastic fluid simulation ». In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. (New York, NY, USA 2005), p. 219–228. ACM.
- Cummins, S. J. et M. Rudman. 1999. « An SPH Projection Method ». *Journal of Computational Physics*, vol. 152, n° 2, p. pages 584–607.
- Desbrun, M. et M.-P. Cani. Aug 1996. « Smoothed Particles : A new paradigm for animating highly deformable bodies ». In *Eurographics Workshop on Computer Animation and Simulation (EGCAS)*. p. 61-76. Springer-Verlag.
- Fält, H. et D. Roble. 2003. « Fluids with extreme viscosity ». In *ACM SIGGRAPH 2003 Sketches & Applications*. (New York, NY, USA 2003), p. 1. ACM.
- Foster, N. et R. Fedkiw. 2001. « Practical animation of liquids ». In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. (New York, NY, USA 2001), p. 23–30. ACM.
- Foster, N. et D. Metaxas. 1996. « Realistic Animation of Liquids ». *Graphical Models and Image Processing*, vol. 58, n° 5, p. pages 471–483.
- Foster, N. et D. Metaxas. 1997. « Modeling the motion of a hot, turbulent gas ». In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. (New York, NY, USA 1997), p. 181–188. ACM Press/Addison-Wesley Publishing Co.
- Harlow, F. H. 1963. « The particle-in-cell method for numerical solution of problems in fluid dynamics ». *Experimental arithmetic, high-speed computations and mathematics*.

- Harlow, F. H. et J. E. Welsh. 1965. « Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface ». *The Physics of Fluids*, vol. 8, N. 10, p. pages 2182–2189.
- Losasso, F., J.O. Talton, N. Kwatra, et R. Fedkiw. juillet-août 2008. « Two-Way Coupled SPH and Particle Level Set Fluid Simulation ». *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, n° 4, p. pages 797–804.
- Monaghan, J. J. May 1989. « On the problem of penetration in particle methods ». *J. Comput. Phys.*, vol. 82, p. pages 1–15.
- Monaghan, J. J. 1992. « Smoothed Particle Hydrodynamics ». *Ann. Rev. of Astronomy and Astrophysics*, vol. 30, p. pages 543–574.
- Monaghan, J. J. Dec. 1997. « Implicit SPH drag and dusty gas dynamics ». *J. Comput. Phys.*, vol. 138, p. pages 801–820.
- Monaghan, J. J. et A. Kocharyan. 1995. « SPH simulation of multi-phase flow ». *Computer Physics Communications*, vol. 87, n° 1-2, p. pages 225–235.
- Müller, M., D. Charypar, et M. Gross. 2003. « Particle-based fluid simulation for interactive applications ». In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. (Aire-la-Ville, Switzerland, Switzerland 2003), p. 154–159. Eurographics Association.
- Müller, M., B. Solenthaler, R. Keiser, et M. Gross. 2005. « Particle-based fluid-fluid interaction ». In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. (New York, NY, USA 2005), p. 237–244. ACM.
- Paiva, A., F. Petronetto, T. Lewiner, et G. Tavares. 2006. « Particle-based non-Newtonian fluid animation for melting objects ». In *Processing, 2006. SIBGRAPI '06. 19th Brazilian*. p. 78-85. SIBGRAPI.
- Paiva, A., F. Petronetto, T. Lewiner, et G. Tavares. April 2009. « Particle-based viscoplastic fluid/solid simulation ». *Comput. Aided Des.*, vol. 41, p. pages 306–314.
- Rasmussen, N., D. Enright, D. Nguyen, S. Marino, N. Sumner, W. Geiger, S. Hoon, et R. Fedkiw. 2004. « Directable photorealistic liquids ». In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. (Aire-la-Ville, Switzerland, Switzerland 2004), p. 193–202. Eurographics Association.
- Shao, S. 2006. « Incompressible SPH simulation of wave breaking and overtopping with turbulence modelling ». *International Journal for Numerical Methods in Fluids*, vol. 50, n° 5, p. pages 597–621.
- Solenthaler, B. et R. Pajarola. July 2009. « Predictive-corrective incompressible SPH ». *ACM Trans. Graph.*, vol. 28, p. pages 40 :1–40 :6.

- Solenthaler, B., J. Schläfli, et R. Pajarola. 2007. « A unified particle model for fluid–solid interactions ». *Computer Animation and Virtual Worlds*, vol. 18, n° 1, p. pages 69–82.
- Stam, J. 1999. « Stable fluids ». In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. (New York, NY, USA 1999), p. 121–128. ACM Press/Addison-Wesley Publishing Co.
- Stora, D., P.-O. Agliati, M.-P. Cani, F. Neyret, et J.-D. Gascuel. 1999. « Animating lava flows ». In *Proceedings of the 1999 conference on Graphics interface '99*. (San Francisco, CA, USA 1999), p. 203–210. Morgan Kaufmann Publishers Inc.
- Teschner, M., B. Heidelberger, M. Mueller, D. Pomeranets, et M. Gross. 2003. « Optimized spatial hashing for collision detection of deformable objects ». *Proceedings of Vision, Modeling, Visualization (VMV 2003)*, p. 47–54.
- Zhu, Y. et R. Bridson. July 2005. « Animating sand as a fluid ». *ACM Trans. Graph.*, vol. 24, p. pages 965–972.