



**HAL**  
open science

# Représentation et rendu de l'océan en synthèse d'images réalistes

Emmanuelle Darles

► **To cite this version:**

Emmanuelle Darles. Représentation et rendu de l'océan en synthèse d'images réalistes. Synthèse d'image et réalité virtuelle [cs.GR]. Université de Limoges, 2008. Français. NNT: . tel-00914626

**HAL Id: tel-00914626**

**<https://theses.hal.science/tel-00914626>**

Submitted on 5 Dec 2013

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

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

**UNIVERSITÉ DE LIMOGES**  
ÉCOLE DOCTORALE « Science, Technologie et Santé »  
FACULTÉ DES SCIENCES ET TECHNIQUES

Thèse N° 56-2008

**THÈSE**  
pour obtenir le grade de  
**DOCTEUR DE L'UNIVERSITÉ DE LIMOGES**  
Discipline / Spécialité : **Informatique**  
présentée et soutenue par  
**Emmanuelle Darles**  
le 24 Octobre 2008

**Représentation et rendu de l'océan en  
synthèse d'images réalistes**

*Thèse dirigée par Djamchid Ghazanfarpour,  
Co-encadrée par Benoît Crespin*

**JURY**

Rapporteurs :

M. Christophe SCHLICK

Professeur à l'Université de Bordeaux 1

Mme Véronique GAILDRAT

Maître de conférence (HDR) à l'Université de Toulouse 3

Examineurs :

M. Djamchid GHAZANFARPOUR

Professeur à l'Université de Limoges

M. Benoît CRESPIN

Maître de conférences à l'Université de Limoges

M. Xavier PUEYO

Professeur à l'Universitat de Girona (Espagne)



# Remerciements

Je souhaite, dans un premier temps, exprimer toute ma gratitude envers mon directeur de thèse, Djamchid Ghazanfarpour, et mon encadrant, Benoît Crespin, pour toute la confiance et les encouragements qu'ils m'ont prodigués pendant ces trois années. Ces travaux n'auraient jamais été menés à bien sans eux et sans leurs précieux conseils.

Je voudrais également remercier tous les membres permanents du département Mathématique / Informatique du laboratoire XLIM et notamment Suzy, Dimitri Plemenos, Julien Cartigny, et enfin Olivier Terraz qui m'a encadré durant mon stage de Master Recherche et qui m'a initié à la recherche avec toute la passion qu'il porte à ce domaine.

Mes remerciements s'adressent également à tous mes collègues doctorants, dans le désordre : Céline, Stan, Boussad, Amadou, Alex, Mike, Cyril, Nadir, Salman et Jeff.

Je tiens à témoigner ma reconnaissance envers toutes mes amies et spécialement à Stephanie, Sandrine, Céline, Sandra et Isa. Merci pour m'avoir souvent remonté le morale, d'avoir toujours veillé sur moi, d'avoir été tout simplement là pendant toutes ces années.

Ma pensée se dirige également vers mon compagnon, Laurent, qui m'a aidé pendant ces longs moments de travail, m'a conseillé et a supporté ma mauvaise humeur et mes états de stress.

Je désire, bien sûr, remercier toute ma famille et plus particulièrement ma mère, mes deux frères Lionel et Daniel et mes cousines Pato, Vivi et Claire. On dit souvent qu'on ne choisit pas sa famille, mais si j'avais dû le faire un jour, c'est vous que j'aurais choisi. Malgré les moments difficiles que nous avons traversés, nous sommes toujours restés soudés. Vous m'avez toujours soutenu, et sans vous, tout ceci n'aurait aucun intérêt.

Je souhaite enfin dédier cette thèse à mon père. Il y a plus de 8 ans, je t'ai promis d'aller le plus loin possible dans mes études, de ne jamais baisser les bras et de toujours me battre dans la vie quoi qu'il arrive et quoi qu'il m'en coûte. J'espère que la haut, tu es fier de moi.

# Résumé

De nos jours, les images de synthèse sont omniprésentes dans notre quotidien. Le réalisme de ces images est grandissant, surprenant, et il n'est souvent pas aisé de distinguer la réalité de la virtualité, cette réalité faite et enrichie par toute la complexité des phénomènes naturels qui nous entourent. L'eau est un de ces phénomènes dont la variété et la richesse dynamique rend la représentation complexe. Nous nous intéressons dans cette thèse à sa forme la plus étendue, celle des océans, qui font partie intégrante de nos paysages.

Dans un premier temps, nous étudions les méthodes permettant la simulation et le rendu de l'océan à la fois dans le domaine physique mais aussi dans le domaine de la synthèse d'images réalistes. Dans le second chapitre, nous proposons une nouvelle méthode de rendu unifiée permettant une visualisation plus rapide de l'océan au large et permettant d'approximer les échanges lumineux surfaciques et sous-surfaciques, l'écume et les phénomènes d'éblouissements. Dans le chapitre 3, nous nous intéressons au déferlement des vagues en proposant une nouvelle approche adaptative basée physique permettant de reproduire ce phénomène et de réduire les temps de calculs imposés par la résolution des équations de la mécanique des fluides en 3D. Dans le quatrième chapitre, nous étendons ce modèle en proposant une approche hiérarchique permettant une plus forte accélération du processus de résolution et d'obtenir une simulation proche de l'interactivité.

**Mots-clés :** synthèse d'images, phénomènes naturels, rendu réaliste, scènes océaniques, simulation de vagues déferlantes, SPH.



# Abstract

Nowadays, computer generated images are very present in our everyday life. Realism of these images is increasing, surprising and it isn't often easy to distinguish the reality of the virtuality, this reality made by all the complexity of the natural phenomena which surround us. Water is one of these phenomena among which the variety and the dynamic comportment produce complex representation. We are interested in this thesis in its most area shape, that of the oceans, which are a part integral of our landscapes.

At first, we study the methods allowing the simulation and rendering of the ocean in the physical domain and in computer graphics. In the second chapter, we propose a new rendering method unified allowing a faster visualization of ocean surface and allowing to approximate scattering and sub-scattering exchanges of light, foam and glare effects. In the chapter 3, we are interested to breaking waves by proposing a new physical based adaptive approach allowing to reproduce this phenomenon and to reduce time of calculations imposed by the resolution of equations of the fluid mechanics in 3D. In the fourth chapter, we extend this model by proposing a hierarchical approach allowing a stronger acceleration of the process of resolution and to obtain a simulation close to interactivity.

**Keywords :** computer graphics, natural phenomena, realistic rendering, oceanic scenes, breaking waves simulation, SPH.





# Table des matières

|  |           |
|--|-----------|
| <b>Introduction</b>  | <b>1</b>  |
| <b>1 Simulation et rendu de l’océan en synthèse d’images réalistes</b>       | <b>7</b>  |
| 1.1 Qu’est-ce qu’une vague? . . . . .  | 7         |
| 1.1.1 Définitions . . . . .  | 7         |
| 1.1.2 La vie d’une vague . . . . .   | 8         |
| 1.2 Physique des vagues . . . . .  | 10        |
| 1.2.1 Représentation paramétrique . . . . .                                  | 11        |
| 1.2.2 Représentation spectrale . . . . .                                     | 13        |
| 1.2.3 Représentation basée sur la mécanique des fluides . . . . .            | 15        |
| 1.2.4 Bilan . . . . .  | 16        |
| 1.3 Simulation de la surface de l’océan en synthèse d’images réalistes . . . | 17        |
| 1.3.1 Approches approximatives . . . . .                                     | 17        |
| 1.3.2 Approches physiques . . . . .  | 24        |
| 1.3.3 Bilan . . . . .  | 32        |
| 1.4 Rendu de la surface de l’océan . . . . .                                 | 32        |
| 1.4.1 Visualisation de la surface libre . . . . .                            | 32        |
| 1.4.2 Rendu de l’écume et des embruns marins . . . . .                       | 34        |
| 1.4.3 Interactions surfaciques et sous-surfaciques . . . . .                 | 36        |
| 1.4.4 Bilan . . . . .  | 41        |
| 1.5 Conclusion . . . . .   | 41        |
| <b>2 Représentation de l’océan en eau profonde</b>                           | <b>43</b> |
| 2.1 Visualisation de la surface par Sphere-Tracing . . . . .                 | 43        |
| 2.1.1 Approche Générale . . . . .  | 43        |
| 2.1.2 Anti-aliassage . . . . .   | 48        |
| 2.1.3 Cohérence spatiale et cohérence temporelle . . . . .                   | 49        |
| 2.2 Rendu de l’écume . . . . .   | 50        |
| 2.3 Rendu des interactions surfaciques et sous-surfaciques . . . . .         | 53        |
| 2.3.1 Atténuation surfacique . . . . .                                       | 54        |

|          |  |            |
|----------|--|------------|
| 2.3.2    | Diffusion sous-surfacique . . . . .                      | 56         |
| 2.4      | Eblouissement . . . . .                                  | 59         |
| 2.5      | Performances . . . . .                                   | 63         |
| 2.6      | Conclusion . . . . .                                     | 63         |
| <b>3</b> | <b>Représentation de l’océan en eau peu profonde</b>     | <b>67</b>  |
| 3.1      | Approche SPH . . . . .                                   | 67         |
| 3.2      | Simulation de vagues déferlantes . . . . .               | 70         |
| 3.2.1    | Méthode d’impulsion finie . . . . .                      | 72         |
| 3.2.2    | Déferlement plongeant et déferlement glissant . . . . .  | 76         |
| 3.2.3    | Combinaison de train de vagues . . . . .                 | 79         |
| 3.2.4    | Simulation des embruns marins . . . . .                  | 79         |
| 3.2.5    | Discussion . . . . .                                     | 82         |
| 3.3      | Approche SPH multirésolution . . . . .                   | 82         |
| 3.3.1    | Identification des particules . . . . .                  | 83         |
| 3.3.2    | Transfert de masse progressif . . . . .                  | 84         |
| 3.3.3    | Conservation de la symétrie des forces . . . . .         | 85         |
| 3.3.4    | Algorithme général . . . . .                             | 85         |
| 3.3.5    | Performances . . . . .                                   | 85         |
| 3.4      | Définition de la surface libre . . . . .                 | 91         |
| 3.4.1    | Extraction par l’algorithme des Marching Cubes . . . . . | 91         |
| 3.4.2    | Surfaces variationnelles . . . . .                       | 92         |
| 3.5      | Conclusion . . . . .                                     | 93         |
| <b>4</b> | <b>Contrôle de fluide</b>                                | <b>95</b>  |
| 4.1      | Simulation SPH hiérarchique . . . . .                    | 95         |
| 4.1.1    | Fusion . . . . .   | 96         |
| 4.1.2    | Calcul de la vitesse . . . . .                           | 96         |
| 4.1.3    | Évaluation de la densité . . . . .                       | 98         |
| 4.1.4    | Changement de mère . . . . .                             | 98         |
| 4.1.5    | Algorithme Général . . . . .                             | 99         |
| 4.2      | Résultats . . . . .                                      | 99         |
| 4.3      | Conclusions . . . . .                                    | 102        |
|          | <b>Bibliographie</b>                                     | <b>115</b> |

# Table des figures

|      |   |    |
|------|---|----|
| 1    | Photographies de la surface de l'océan en eaux profondes (à droite) et en eaux peu profondes (à gauche) . . . . .   | 2  |
| 1.1  | Représentation théorique d'une vague sous forme ondulatoire - Le plan défini par $y = 0$ représente la surface de l'eau au repos et $\eta(x, t)$ désigne l'expression de la surface de libre à un instant $t$ . . . . . | 8  |
| 1.2  | Vie d'une vague de vent - schéma inspiré de [Tho01] . . . . .   | 9  |
| 1.3  | Spectre directionnel de Pierson-Moskowitz pour une vitesse de vent $U_{10}$ de $8km/h$ . . . . .  | 15 |
| 1.4  | Représentations de la surface de l'océan extraites de [FR86] (à gauche) et [Pea86] (à droite) . . . . .   | 18 |
| 1.5  | Simulation de l'océan en temps-réel (extrait de [Kry05]) . . . . .  | 20 |
| 1.6  | Simulation de la surface de l'océan par la méthode de [Tes01] . . . . .   | 22 |
| 1.7  | Simulation de la surface de l'océan extraite de [TDG00] . . . . .   | 23 |
| 1.8  | Simulation eulérienne (de gauche à droite et de haut en bas) de [FF01], [EMF02], [LSSF06], [EMF02], [MMS04] et de [TSS+07] . . . . .  | 24 |
| 1.9  | Simulation <i>SPH</i> (de gauche à droite et de haut en bas) de [MCG03], [BYM05], [KW06], [MKN+04], [KAG+05] et de [CBP05] . . . . .  | 28 |
| 1.10 | Simulation hybride (de gauche à droite et de haut en bas) de [GH04], [TRS06], [ZYP06], [TRS06], [KCC+06] et de [LTKF08] . . . . .   | 30 |
| 1.11 | Rendu de l'écume par la méthode extraite de [JG01] (à droite) . . . . .   | 35 |
| 1.12 | Rendu des embruns marins par la méthode extraite de [WZC+06] (à gauche) et celle extraite de [CC06] (à droite) . . . . .  | 36 |
| 1.13 | Illustration des lois de réflexion et de réfraction de <i>Snell-Descartes</i> . . . . .   | 37 |
| 1.14 | Spectres d'absorption (à gauche) et de diffusion (à droite) de l'eau selon la concentration en phytoplancton (en jaune $C = 10mg/m^3$ , en bleu $C = 1mg/m^3$ et magenta $C = 0.01mg/m^3$ ) . . . . .                   | 39 |
| 1.15 | Rendu de la surface de l'océan pour une eau profonde (à gauche) et eau claire (à droite) par la méthode présentée dans [PA01] . . . . .   | 41 |
| 2.1  | Principe du <i>Sphere-Tracing</i> de [Har96] . . . . .  | 44 |

|      |   |    |
|------|---|----|
| 2.2  | Illustration de notre méthode avec une surface très perturbée - Les rayons des sphères représentent les pas consécutifs sur le rayon. . . . .   | 49 |
| 2.3  | Surface de l'eau sans aliassages obtenue avec notre méthode . . . . .   | 50 |
| 2.4  | Cohérence spatiale - En pointillé, les sphères précalculées d'un rayon voisin. En plein, celles calculées pour le rayon courant. . . . .  | 51 |
| 2.5  | Cohérence temporelle - En pointillé, les sphères précalculées à l'instant $t$ . En plein, celles calculées à l'instant $t + 1$ . . . . .  | 51 |
| 2.6  | Rendu de l'écume avec une vitesse de vent ( $U_{10}$ ) de 7 et 10km/h . . . . .   | 54 |
| 2.7  | Ensemble des sphères le long d'un rayon affichées par niveaux de gris, correspondant à la progression du coefficient d'extinction (le plus clair désignant la plus faible valeur). . . . .                  | 56 |
| 2.8  | Rendu de l'atténuation atmosphérique avec des vitesses de vent de 7 et 10km/h. . . . .  | 57 |
| 2.9  | Ensemble des sphères le long d'un rayon réfracté affichées par niveaux de gris, correspondant à la progression de la quantité de lumière diffusée (le plus clair désignant la plus faible valeur) . . . . . | 58 |
| 2.10 | Rendu de la diffusion sous-surfacique avec une eau claire et trouble . . . . .  | 59 |
| 2.11 | rendu de la diffusion sous-surfacique avec (de gauche à droite et de haut en bas) une concentration en chlorophylle $C$ de 0.001, 0.01 0.05 et 1.0mg/m <sup>3</sup> . . . . .                               | 60 |
| 2.12 | Texture d'éblouissement utilisée dans notre méthode (à gauche) et le buffer de réflexion spéculaire (à droite) . . . . .  | 62 |
| 2.13 | Rendu de l'éblouissement par notre méthode . . . . .  | 62 |
| 3.1  | Simulation SPH d'un liquide visqueux (à gauche) et de l'eau (à droite) . . . . .  | 72 |
| 3.2  | Magnitude de la vitesse horizontale (à gauche) et verticale (à droite) en fonction du temps . . . . .   | 73 |
| 3.3  | Les différentes phases du déferlement d'une vague en 2D calculé avec notre méthode . . . . .  | 74 |
| 3.4  | Les différentes phases du déferlement d'une vague en 3D calculées avec notre méthode . . . . .  | 75 |
| 3.5  | En haut, variation de la hauteur avec $H = 0.2$ et $H = 0.35$ . En bas, variation de l'angle de déviation avec $\theta = 0$ et $\theta = 0.5$ (en radians) . . . . .  | 75 |
| 3.6  | Moyenne de la magnitude des forces d'impulsions du système sur une animation de 500 images . . . . .  | 76 |
| 3.7  | Profil de la vitesse verticale en fonction du temps (de gauche à droite et de haut en bas) pour $c = 1.0$ , $c = 0.5$ , $c = 0.2$ et $c = 0.1$ . . . . .  | 78 |
| 3.8  | Une vague glissante (à gauche) et plongeante (à droite) au même instant de simulation . . . . .   | 78 |

|      |  |     |
|------|--|-----|
| 3.9  | Combinaison de deux vagues déferlantes . . . . .   | 80  |
| 3.10 | Simulation des embruns marins dû au déferlement . . . . .  | 81  |
| 3.11 | Transfert de masse progressif pendant les opérations de fusion et de division entre une particule mère (au centre en vert) et ses filles (en bleues). . . . .  | 83  |
| 3.12 | Comparaison de la simulation obtenue sans (à gauche) et avec (à droite) multirésolution avec 100K particules initiales et un soliton de paramètre (0.35, 0.1, 0, 0). Le rayon des particule correspond au rayon d'interaction qui a été agrandi pour une meilleur visualisation. . . . . | 86  |
| 3.13 | Comparaison des surfaces extraites des simulations de la figure 3.12. . . . .  | 87  |
| 3.14 | Gain amené par la multirésolution en pourcentage de particules désactivées (à gauche) et évolution du temps de calcul moyen en secondes par image (à droite) entre une simulation standard et multirésolution en fonction du nombre de particules initial. . . . .                       | 88  |
| 3.15 | Comparaison de l'évolution de la moyenne des densités par image sur l'ensemble des particules actives de la simulation (à gauche) et estimation de l'erreur relative commise correspondante avec et sans multirésolution . . . . .   | 89  |
| 3.16 | Comparaison de l'évolution de la moyenne de la magnitude des forces par image sur l'ensemble des particules actives de la simulation (à gauche) et estimation de l'erreur relative commise correspondante avec et sans multirésolution . . . . .   | 89  |
| 3.17 | Simulation avec la méthode de [APKG07] (les images de la colonne de droite permettent de montrer les particules fusionnées en vert) . . . . .  | 90  |
| 3.18 | Localisation de la surface en fonction de la position des centres et de leurs valeurs. . . . .   | 92  |
| 4.1  | Processus de fusion hiérarchique . . . . .   | 97  |
| 4.2  | Processus de changements de mère - Les flèches indiquent les liaisons entre les particules filles et les particules mères. . . . .   | 99  |
| 4.3  | Exemple de simulation SPH hiérarchique sur 3 niveaux - Les particules en bleu sont celles du niveau 1, en rouge, du niveau 2 et en vert du niveau 3. . . . .   | 101 |
| 4.4  | Comparaison de notre approche avec une approche classique - La première colonne représente la simulation SPH standard (avec 1 seul niveau). Les deuxième et troisième colonnes représentent respectivement une simulation SPH hiérarchique avec 2 et 3 niveaux de simulation             | 103 |

---

|      |  |     |
|------|--|-----|
| 4.5  | Comparaison du nombre de Reynolds avec une approche classique (en noir) pour différentes simulations hiérarchiques avec 2 niveaux (en rouge), 3 niveaux (en vert) et 5 niveaux (en bleu) . . . . .   | 104 |
| 4.6  | Simulation de vagues déferlantes avec notre système hiérarchique - La première colonne représente les particules du niveau le plus bas qui sont contrôlées hiérarchiquement par les particules du niveau le plus haut, affichées sur la seconde colonne. . . . . | 105 |
| 4.7  | Coefficient d'atténuation de l'eau pure - données extraites de [GM83]  | 109 |
| 4.8  | Représentation graphique de la fonction $W_{poly6}(\mathbf{r}, h)$ (en trait plein), $\nabla W_{poly6}(\mathbf{r}, h)$ (en trait fin) et $\nabla W_{poly6}(\mathbf{r}, h)$ (en pointillé) pour $h = 1.0$   | 112 |
| 4.9  | Représentation graphique de la fonction $W_{spiky}(\mathbf{r}, h)$ (en trait plein), $\nabla W_{spiky}(\mathbf{r}, h)$ (en trait fin) et $\nabla W_{spiky}(\mathbf{r}, h)$ (en pointillé) pour $h = 1.0$   | 113 |
| 4.10 | Représentation graphique de la fonction $W_{visc}(\mathbf{r}, h)$ (en trait plein), $\nabla W_{visc}(\mathbf{r}, h)$ (en trait fin) et $\nabla W_{visc}(\mathbf{r}, h)$ (en pointillé) pour $h = 1.0$ .  | 113 |

# Liste des tableaux

|     |  |     |
|-----|--|-----|
| 2.1 | Gain en temps de calcul apporté par notre méthode comparé à un ray-marching et à un sphere-tracing non optimisé en fonction du nombre de composantes . . . . . | 64  |
| 2.2 | Apport de la cohérence spatiale et temporelle en fonction du nombre de composantes . . . . .   | 64  |
| 2.3 | Coût mémoire de notre structure de données pour différentes résolutions d'images et avec une surface représentée par 30 composantes . .                        | 64  |
| 2.4 | Répartition des calculs pour le rendu d'une image . . . . .  | 64  |
| 3.1 | Comparaison des pourcentages des temps de calculs par étape avec et sans (entre crochets) multirésolution avec 50K, 100K et 200K particules initiales. . . . . | 89  |
| 4.1 | Temps de calculs moyen par images (en secondes) avec différents niveaux hiérarchiques pour 10K, 20K et 50K particules initiales. . . .                         | 103 |





# Introduction

## Cadre de la thèse

De nos jours, les images de synthèse sont omniprésentes dans notre quotidien. Que ce soit de la simple affiche publicitaire au film d'animation, ces images s'insinuent sur nos écrans, dans nos rues et sur nos murs. Pourtant l'informatique graphique est une science moderne dont les débuts datent de moins de 40 ans. L'avènement de nouveaux médias tels que des jeux vidéos ou d'Internet a permis de développer ce domaine dont le but, aujourd'hui, est d'obtenir des images photoréalistes afin d'immerger sensoriellement un spectateur dans un monde virtuel. Le réalisme obtenu est grandissant, surprenant, et il n'est pas souvent aisé de distinguer la réalité de la virtualité, cette réalité faite et enrichie par toute la complexité des phénomènes naturels qui nous entourent.

L'eau est un de ces phénomènes dont la variété et la richesse dynamique rendent la représentation complexe. Du flocon de neige aux nuages que nous pouvons contempler, de la goutte de pluie aux immenses glaciers en Antarctique, on la retrouve sous des formes diverses et dans des situations variées. Nous nous intéressons dans cette thèse à sa forme la plus étendue, celle des océans, qui recouvrent plus de 70% de notre planète et qui font partie intégrante de nos paysages.

En océanographie, les physiciens distinguent principalement trois zones : les eaux dites *profondes*, situées très loin du rivage, celle dites *intermédiaires* et enfin celles dites *peu profondes*, situées près des côtes. Cette distinction permet une caractérisation différente du mouvement de la surface selon la zone étudiée. Par exemple, en eaux profondes, on observe généralement des mouvements oscillatoires de la surface libre, c'est-à-dire la surface définie par l'interface entre l'eau et l'air. Dans les eaux peu profondes, c'est à dire, à l'approche du rivage, on assiste au déferlement des vagues, phénomène complexe et s'accompagnant de vives interactions à l'interface air/eau (voir figure 1). Du fait de cette complexité dynamique et spatiale à diverses échelles, les océans demeurent un sujet de recherche en constante activité.



FIG. 1 – Photographies de la surface de l’océan en eaux profondes (à droite) et en eaux peu profondes (à gauche)

Pour modéliser et simuler ces échanges, les océanographes tentent de caractériser le comportement de la surface de l’océan soit à l’aide d’équations paramétriques soit à l’aide de modèles issus du domaine de la mécanique des fluides. Pourtant, le formalisme de ces approches reste approximatif et ne peut représenter l’intégralité de ce phénomène. En s’appuyant sur ces modèles physiques, de nombreux outils ont été proposés en synthèse d’images afin de reproduire des scènes océaniques, tel que le plugin *Psunami* qui a été notamment utilisé dans des films comme *Titanic*, *Le prince d’Egypte* ou *WaterWorld*. Cependant, bien que les images obtenues soient très réalistes, les effets sont limités avec des coûts de calculs souvent prohibitifs, allant de plusieurs heures par images. D’un autre côté, la simulation des vagues déferlantes reste un sujet problématique en synthèse d’images et comme nous le verrons, peu de méthodes ont été proposées afin de simuler ce phénomène. Aucune d’entre elles ne permet sa représentation dans sa diversité et sa complexité visuelle.

C’est dans ce cadre que se place cette thèse : simuler et représenter la surface de l’océan de façon visuellement réaliste, dans sa richesse et sa diversité.

## Problématiques

Comme cela est mentionné dans [Tho01], la simulation de la surface océanique est liée à un double problème : d’une part sa représentation et d’autre part son rendu. En effet, le mouvement dynamique de la surface de l’océan est le plus souvent influencé par de fortes perturbations à grandes échelles, et il n’est pas aisé de représenter ces échanges. D’autre part, l’eau possède des propriétés optiques com-

plexes. Au niveau des océans, ces propriétés sont liées à divers paramètres comme par exemple la concentration en phytoplancton, ce qui permet d'expliquer en partie pourquoi nous pouvons observer des eaux teintées vertes ou alors très bleues.

Comme nous le verrons dans le chapitre 1, de nombreux travaux ont tenté d'approximer cette variété dynamique et visuelle, caractéristique des scènes océaniques. D'une part au niveau de la visualisation de la surface, des méthodes se sont intéressées au rendu des échanges surfaciques, d'autres aux échanges sous-surfaciques ou d'autres à l'écume. Il manque donc une méthode de rendu unifiée capable de simuler l'ensemble de ces phénomènes, sans pour autant trop augmenter les temps de calcul. D'autre part au niveau de la simulation des vagues déferlantes, où il existe peu d'articles en informatique graphique s'intéressant à ce phénomène. Comme nous le verrons, les approches proposées ne permettent pas de le représenter à la fois dans son intégralité et sa diversité. Il manque donc une méthode permettant de pallier à ces limitations.

## Apports de cette thèse

Dans un premier temps, nous nous sommes intéressés à la simulation et à la visualisation de la surface océanique en eaux profondes. Nous avons axé nos recherches sur l'unification d'une méthode de rendu capable de prendre en compte divers phénomènes tels que l'écume, l'éblouissement et les échanges lumineux surfaciques et sous-surfaciques. Dans ce contexte, nous avons proposé un algorithme complet permettant d'approximer et d'accélérer le calcul de ces effets. Pour parvenir à ce but, nous avons proposé une nouvelle méthode du calcul de l'intersection entre un rayon et la surface de l'océan en se basant sur une méthode précédemment employée dans le cadre des *hypertextures*. Nous verrons comment cette approche peut s'adapter au modèle de représentation utilisé pour la surface de l'océan en réduisant considérablement les temps de calculs mais aussi les artefacts visuels tels que les effets de moirés à l'horizon. Contrairement aux méthodes existantes, notre approche permet de reproduire la complexité visuelle de la surface océanique dans son intégralité sans engendrer des temps de calcul excessifs.

Dans un second temps, nous nous sommes intéressés au domaine situé en eaux peu profondes et, plus spécifiquement, à la simulation des vagues déferlantes, en proposant une nouvelle méthode basée sur la résolution des équations de la mécanique des fluides. Afin de pallier aux limites des méthodes existantes, nous avons proposé un nouveau modèle physique permettant de représenter ce phénomène à la fois dans sa

diversité mais aussi en prenant en compte des sous-processus intervenants durant ce phénomène, tels que les embruns marins. Ce modèle est capable de simuler à la fois différents types de vagues déferlantes mais permet aussi à un utilisateur de contrôler les caractéristiques géométriques de chaque vague. Afin de réduire les temps de calculs, nous avons recherché à approximer les équations de la mécanique des fluides à travers une méthode adaptative.

Enfin, nous nous sommes également intéressés à proposer un nouveau modèle basé physique permettant d'accélérer le processus de résolution des équations de la mécanique des fluides en 3D à travers l'emploi d'un système hiérarchique. Comme nous le verrons, cette approche permet également d'ajouter un haut niveau de contrôle et peut s'adapter dans le cadre de la simulation de vagues déferlantes.

## Organisation du mémoire

Dans un premier temps, nous étudierons les méthodes permettant la simulation et le rendu de l'océan à la fois dans le domaine physique et dans le domaine de la synthèse d'images réalistes. Nous introduirons les principaux concepts physiques permettant la représentation de la surface de l'océan et notamment ceux utilisés en mécanique des fluides pouvant représenter le déferlement des vagues. Nous nous attacherons à étudier les approches existantes en informatique graphique permettant de simuler et de visualiser la surface océanique.

Dans un second temps, nous étudierons une nouvelle méthode de rendu unifiée permettant une visualisation de la surface de l'océan en eaux profondes, plus rapide et permettant d'approximer les échanges lumineux surfaciques et sous-surfaciques. Nous montrerons comment notre algorithme est capable d'accélérer les calculs de ces phénomènes en s'appuyant sur des modèles physiques simplifiés. Nous verrons également une nouvelle méthode permettant de simuler l'écume et les effets d'éblouissements à la surface et de pallier aux limites des techniques existantes.

Dans le chapitre 3, nous nous intéresserons au déferlement des vagues. Nous verrons une nouvelle méthode, basée sur la mécanique des fluides, permettant de reproduire ce phénomène et d'approximer les sous-processus intervenant durant ce phénomène tel que les embruns marins, permettant d'enrichir le rendu des images finales et d'obtenir une simulation plus réaliste. Cette méthode permet de reproduire différents types de vagues déferlantes tels que des vagues glissantes ou plongeantes. Comme nous le verrons, l'approche que nous avons proposée est également adaptative, per-

mettant de réduire les temps de calculs imposés par la résolution des équations de la mécanique des fluides en 3D.

Dans le quatrième chapitre, nous verrons comment cette méthode peut être également étendue au contrôle de fluide, afin de permettre à un utilisateur d'obtenir un aperçu de l'écoulement d'un fluide et d'agir sur cet écoulement. Nous introduirons une méthode hiérarchique capable d'approximer les équations de la mécanique des fluides et d'obtenir une très forte accélération du processus de résolution. Comme nous le verrons, notre approche permet un fort contrôle utilisateur et peut s'adapter à la représentation de vagues déferlantes à l'aide de la méthode décrite dans le chapitre 3.

Enfin, nous concluons sur l'ensemble de ces travaux en examinant les différentes limites et perspectives possibles. Nous verrons comment ces travaux apportent une solution efficace à la représentation de la surface de l'océan et comment ils peuvent être étendus.

Deux annexes sont placées en fin de ce mémoire. L'annexe A fournit des données employées dans la méthode de rendu que nous proposons dans le chapitre 2. L'annexe B détail les fonctions mathématiques utilisées pour la résolution des équations de la mécanique des fluides dans les chapitres 3 et 4.



# Chapitre 1

## Simulation et rendu de l'océan en synthèse d'images réalistes

### 1.1 Qu'est-ce qu'une vague ?

Qu'est-ce qu'une vague ? Comment se forme-t-elle et se propage-t-elle ? Derrière ces questions, pour le moins triviales en apparence, se cache un processus physique complexe longtemps étudié par bon nombre de physiciens, mathématiciens ou océanographes. La compréhension du mécanisme dynamique de ce phénomène reste pourtant encore aujourd'hui un sujet de recherche en constante activité en raison de la variété spatio-temporelle des échanges mis en jeu. Cependant, en approximant ce comportement et ces échanges, on peut représenter une vague sous la forme d'une onde périodique avec une durée de vie limitée qui s'étend de sa naissance jusqu'à son déferlement, ce que nous tentons d'expliquer dans cette partie.

#### 1.1.1 Définitions

Une vague est une déformation périodique de la surface de l'eau qui correspond à un mouvement oscillatoire de l'interface entre l'air et l'eau maintenu par un échange entre énergie cinétique et potentielle gravitationnelle. Elle peut donc être représentée sous la forme d'une onde périodique caractérisée par plusieurs éléments (voir la figure 1.1) :

- une hauteur  $H$ , représentant la distance verticale entre la crête et le creux ;
- une amplitude  $A$ , représentant la moitié de la hauteur ;
- une longueur d'onde  $\lambda$  représentant la distance horizontale entre deux crêtes successives ;
- une période  $T$ , représentant le temps en secondes entre deux crêtes successives ;
- une fréquence  $f$ , représentant l'inverse de la période  $T$  ;



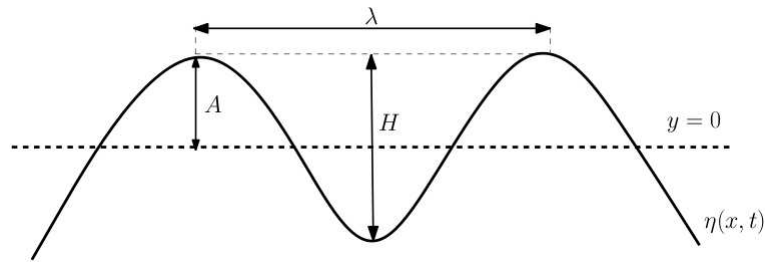


FIG. 1.1 – Représentation théorique d'une vague sous forme ondulatoire - Le plan défini par  $y = 0$  représente la surface de l'eau au repos et  $\eta(x, t)$  désigne l'expression de la surface de libre à un instant  $t$

- une célérité  $c$  représentant sa vitesse de propagation ;
- une pulsation  $\omega$ , qui peut être calculée en fonction de sa fréquence  $\omega = 2\pi f$  ;
- un nombre d'onde  $k$ , qui peut être calculé en fonction de sa longueur d'onde par  $k = 2\pi/\lambda$ ,
- une cambrure  $\gamma$  définie par  $\gamma = H/\lambda$ .

Il existe différents types de vagues qui peuvent être classés selon leurs facteurs de génération :

- **les vagues de marée** qui sont des vagues de très grandes longueurs d'onde (qui peuvent aller de 12 à 24 heures) et sont dues à l'attraction de la lune sur les océans ;
- **les vagues de vent** qui sont dues à l'action mécanique du souffle du vent et caractérisées par une très faible longueur d'onde (environ de 1 à 20 secondes). En se regroupant, elles forment ce que l'on nomme le plus souvent *la houle* ;
- **les vagues d'envergure exceptionnelles** telles que les tsunamis ou les vagues scélérates qui sont principalement provoquées par les tremblements de terre. Elles constituent actuellement un sujet de recherche pour les physiciens et océanographes, qui tentent encore aujourd'hui d'expliquer leurs processus de formation et de propagation.

Nous nous intéressons dans cette thèse aux vagues que l'on observe le plus souvent, c'est à dire celles générées par le vent. Ainsi, dans tous le reste de ce mémoire, nous désignerons par le terme "*vague*" ce type de vague.

### 1.1.2 La vie d'une vague

La vie d'une vague est organisée à travers 4 étapes : sa genèse, sa propagation, sa transformation, puis enfin son déferlement (voir la figure 1.2).

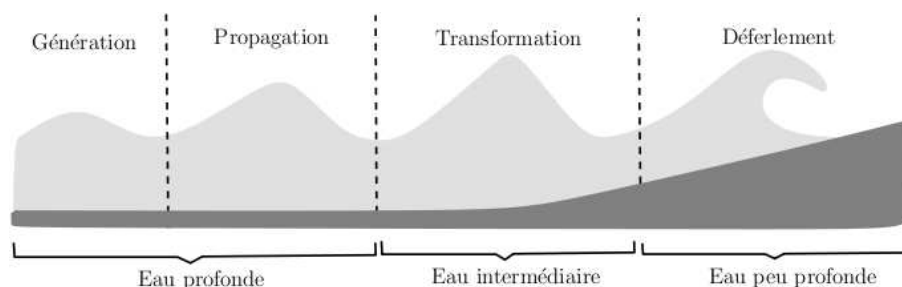


FIG. 1.2 – Vie d'une vague de vent - schéma inspiré de [Tho01]

**Génération :** La genèse d'une vague de vent est due, comme son nom l'indique, à l'effet du vent sur la surface de l'eau qui fait osciller celle-ci autour de sa position de repos. Au cours du temps, on assiste à une amplification des vaguelettes générées qui augmentent en hauteur, en longueur d'onde et en période. L'ensemble de ces vagues constituent ce que l'on appelle la *mer de vent* dont la force dépend directement de celle du souffle du vent, mais aussi de sa durée et de sa distance appelée *fetch*.

**Propagation :** En dehors de leurs zones génératrices, ces vagues se propagent en direction de la côte pour se grouper et former des vagues plus régulières : c'est ce que l'on appelle communément la houle. Durant leurs propagations, plusieurs trains de houles de longueur d'onde et de direction différentes peuvent alors se superposer pour former ce que l'on appelle une *mer croisée*.

**Transformation :** En approchant près de la côte, cet ensemble de vagues subit alors l'influence du fond marin : la trajectoire des particules d'eau passe d'un mouvement circulaire à un mouvement elliptique pour tendre vers un mouvement horizontal. Ceci a pour effet d'augmenter la hauteur et la cambrure des vagues et de diminuer leurs longueurs d'onde et leurs célérités.

**Déferlement :** Lorsque la cambrure d'une vague dépasse une certaine limite, elle déferle. Ce phénomène est complexe et s'accompagne le plus souvent de vives interactions entre les interfaces de l'eau et de l'air et d'une forte dissipation de l'énergie. Plusieurs travaux ont permis de déterminer l'origine du déferlement :

- Mitchell a montré qu'une vague déferlait si sa cambrure  $\gamma$  était supérieure à  $1/7$ ;
- Stokes a montré que si l'angle formé entre la crête et le front descendant d'une vague était supérieur à  $120^\circ$  alors elle pouvait déferler.

Le type de déferlement dépend aussi de la configuration du relief sous-marin :

- **le déferlement glissant** survient pour une pente de plage très faible. La vague s'écroule en glissant sur le rivage ;
- **le déferlement plongeant** s'observe avec une pente de plage moyenne. La crête de la vague se détache en un jet pour se retourner sur elle-même et plonger ; la surface de l'eau devient alors un mélange d'air et d'eau ;
- **le déferlement frontal** est caractéristique des pentes très abruptes. La hauteur de la vague devient alors très importante puis elle s'écroule brutalement.

Galvin en 1968 a proposé de classer ces différents types de déferlement en utilisant un nombre adimensionnel  $K$ , prenant en compte la hauteur  $H$  d'une vague, sa période  $T$  et l'angle de la pente de plage  $\alpha$ . Ce nombre peut être défini par :

$$K = \frac{H \tan^{-1} \alpha}{gT}$$

avec  $g$  la valeur du champs gravitationnel terrestre ( $9.81m.s^{-2}$ ). Une valeur inférieure à 0.003 caractérise un déferlement glissant, supérieure à 0.0068, un déferlement frontal et comprise entre ces deux valeurs, un déferlement plongeant.

Similairement, *le nombre d'Irribarren*  $I$  [Bat74] permet aussi de prédire le type de déferlement en fonction de la pente de la plage :

$$I = \frac{\tan \alpha}{\sqrt{H/\lambda}} \quad (1.1)$$

avec :

- $I \leq 0.4$  pour un déferlement glissant ;
- $0.4 < I \leq 2.0$  pour un déferlement plongeant ;
- $I \geq 2.0$  pour un déferlement frontal.

## 1.2 Physique des vagues

En physique, il existe différents moyens de représenter la dynamique de l'océan, de façon paramétrique, spectrale ou en utilisant directement les équations de Navier-Stokes issues de la mécanique des fluides. La première catégorie d'approches permet de calculer la trajectoire des particules d'eau et de la surface libre au moyen d'équations paramétriques basée sur des observations. La seconde tente d'approximer directement l'état de la mer en calculant les caractéristiques des vagues selon leurs amplitudes et leurs fréquences. Enfin, les équations de Navier-Stokes permettent de simuler la dynamique d'un fluide de manière générale et peuvent être ainsi utilisées pour représenter le comportement de la surface de l'océan de manière physiquement réaliste.

### 1.2.1 Représentation paramétrique

Les modèles paramétriques de houle permettent de modéliser la dynamique de la surface en s'appuyant sur la représentation ondulatoire des vagues. Ils s'appuient sur des hypothèses communes qui consistent à considérer le fluide comme étant parfait et incompressible, c'est à dire à le supposer non visqueux et de volume constant. Leurs domaines d'application sont variables, des eaux profondes jusqu'aux eaux peu profondes, ce qui permet une large étendue de modélisation. Nous présentons ici les modèles principaux selon leur ordre d'apparition, coïncidant avec leur complexité.

#### Modèle de Gerstner

Le modèle proposé par Gerstner en 1802 est le premier modèle de représentation de houle. Ce modèle suppose que le fond est situé à une distance infinie de la surface de l'eau, négligeant ainsi les effets de frottement sur la trajectoire des particules d'eau. Il considère alors que chaque particule de fluide suit une trajectoire circulaire autour d'une position de repos, de rayon égal à l'amplitude de la vague. La position d'une particule d'eau de coordonnées 2D  $x$  et  $z$  peut alors être décrite à l'aide des équations suivantes :

$$\begin{cases} x = x_0 - Ae^{ky_0} \sin(kx_0 - wt) \\ y = y_0 - Ae^{ky_0} \cos(kx_0 - wt) \end{cases} \quad (1.2)$$

La forme de la courbe définie par le déplacement des particules dépend alors directement du produit entre l'amplitude  $A$  et le nombre d'onde  $k$  de la vague. Si  $kA < 1$ , cette courbe est alors une trochoïde. Si  $kA = 1$ , elle forme une cycloïde. Dans le dernier cas ( $kA > 1$ ), cette trajectoire ne peut plus représenter un mouvement réaliste.

Ce modèle simple n'étant valable qu'à de très grandes profondeurs, les modèles suivants permettent d'étendre leur domaines d'application afin de couvrir les eaux intermédiaires et peu profondes.

#### Modèle de Stokes

Le modèle de Stokes permet de modéliser la houle à la fois dans le domaine des eaux profondes mais aussi dans la zone dite des eaux intermédiaires (entre les eaux profondes et les eaux peu profondes) en décrivant la trajectoire des particules sous la forme d'une ellipse dont le rayon croît exponentiellement en fonction de la profondeur. Ce modèle s'appuie sur la résolution de l'équation de Bernoulli issue de la mécanique des fluides, décrivant la conservation d'énergie d'un fluide parfait et incompressible. Le long d'un conduit de hauteur  $y$ , la somme des pressions et des

énergies mécaniques par unité de volume du fluide est considérée comme constante :

$$\frac{1}{2}\vec{U}^2 + \frac{p}{\rho} + gy = cste \quad (1.3)$$

avec  $\vec{U}$  le vecteur vitesse du fluide,  $p$  sa pression,  $\rho$  sa densité et  $g$  la valeur du champ gravitationnel terrestre ( $9.81m/s^{-2}$ )

Stokes suppose le fluide comme étant irrotationnel, ce qui implique que le rotationnel du vecteur vitesse d'un volume élémentaire de fluide est nul :  $rot(\vec{U}) = \vec{0}$ . Ceci est aussi équivalent à dire que le vecteur vitesse du fluide ne dépend que d'une fonction scalaire  $\phi$  appelée potentiel de vitesse, soit :  $\vec{U} = \nabla(\phi)$ , avec  $\nabla(\phi)$  étant l'opérateur gradient s'exprimant par le vecteur de coordonnées  $(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z})$ . L'équation 1.3 devient alors :

$$\frac{1}{2}\nabla(\phi)^2 + \frac{p}{\rho} + gy = cste \quad (1.4)$$

À l'ordre 1, la solution proposée par Stokes est d'exprimer le potentiel de vitesse et le profil de la surface libre  $\eta_1$  sous la forme suivante :

$$\begin{cases} \phi_1 = A \frac{\cosh(k(y+d))}{\sinh(kd)} \cos(kx - wt) \\ \eta_1 = -A \sin(kx - wt) \end{cases} \quad (1.5)$$

Aux ordres supérieurs, ce modèle permet de prendre en compte des termes non linéaires, s'approchant ainsi d'une solution numériquement plus exacte. Par exemple, à l'ordre 2, les expressions du potentiel et de la surface libre sont données respectivement par :

$$\begin{cases} \phi_2 = \phi_1 + \frac{3\pi H^2}{16T} \frac{\cosh(2k(y+d))}{\sinh^4(kh)} \sin(kw - wt) \\ \eta_2 = \eta_1 + \frac{\pi H}{4L} \frac{3 - \tan^2(kh)}{\tan^3(kh)} \cos(kx - wt) \end{cases}$$

avec  $L = \frac{gT^2}{2\pi} \tanh(\frac{2\pi h}{\lambda})$ ,  $\phi_1$  et  $\eta_1$  étant définis par le système d'équations 1.5.

Bien que ce modèle soit capable de modéliser la houle dans le domaine des eaux intermédiaires, il néglige l'influence des effets de frottement du fond sur la trajectoire des particules, se traduisant par une augmentation de l'amplitude et de la cambrure de la vague et conduisant à son déferlement. Le modèle de Biesel permet de pallier à cette limitation.

### Modèle de Biesel

La théorie de Biesel [Bie52] permet de représenter le cas de houle déferlante. Dans ce modèle, la position horizontale  $x$  et verticale  $y$  d'une particule d'eau s'exprime sous la forme suivante :

$$\begin{cases} x = x_0 + A_x \sin(\int_0^{x_0} k dx - wt) - B_x \cos(\int_0^{x_0} k dx - wt) \\ y = y_0 + A_y \cos(\int_0^{x_0} k dx - wt) - B_y \sin(\int_0^{x_0} k dx - wt) \end{cases}$$

avec les termes  $A_x$ ,  $A_y$ ,  $B_x$  et  $B_y$  définis par :

$$\left\{ \begin{array}{l} A_x = \frac{\Theta}{\tanh(kh)} \\ A_y = \Theta \\ B_x = \Theta \tan(\alpha) \left( \frac{1 + kh \tanh(kh)}{\Psi^2 \tanh(kh)^2} + \frac{2kh \tanh(kh) + (kh)^2}{2\Psi \sinh(kh) \cosh(kh) \tanh(kh)} - \frac{kh}{\tanh(kh)} - 1 \right) \\ B_y = \Theta \tan(\alpha) \left( \frac{kh + \tanh(kh)}{\Psi^2 \tanh(kh)} + \frac{2kh + (kh)^2 \tanh(kh)}{2\Psi \sinh(kh) \cosh(kh) \tanh(kh)} - \frac{1}{\tanh(kh)} - kh \right) \\ \Psi = \left( 1 + \frac{kh}{\sinh(kh) \cosh(kh) \tanh(kh)} \right) \\ \Theta = \frac{A}{\sqrt{\Psi \tanh(kh)}} \end{array} \right.$$

Ce modèle est capable de représenter une vague déferlante mais partiellement, c'est à dire, jusqu'à son point de déferlement. Pour représenter ce phénomène dans son intégralité, il faut faire appel aux équations de la mécanique des fluides que nous présentons dans la section 1.2.3.

## 1.2.2 Représentation spectrale

La représentation spectrale permet de caractériser la distribution des vagues selon leurs fréquence et leurs amplitudes en calculant des spectres de vagues  $S(f)$  représentant de manière statistique la distribution de l'énergie  $E$  (proportionnelle à l'amplitude) des vagues selon leurs fréquences  $f$ . La prise en compte de la direction de propagation  $\theta$  permet de définir un spectre 3D appelé spectre directionnel  $S(f, \theta)$  qui peut être exprimé de façon générale sous la forme suivante :

$$S(f, \theta) = E(f)D(f, \theta) \quad (1.6)$$

avec  $E(f)$  la distribution statistique de l'énergie des vagues selon leurs fréquences et  $D(f, \theta)$  leurs répartitions fréquentielles selon leurs directions de propagation. Nous présentons ici les spectres directionnels les plus utilisés en synthèse d'images, Pierson-Moskowitz et JONSWAP.

### Spectre de Pierson-Moskowitz

En se basant sur des données expérimentales, Pierson et Moskowitz [PM64] ont proposé de calculer la distribution de l'énergie avec la formule suivante :

$$E_{PM}(f) = \frac{\alpha g^2}{16\pi^4 f^5} \exp\left(-\frac{5}{4}\left(\frac{f_m}{f}\right)^4\right) \quad (1.7)$$

avec  $\alpha = 0.0081$  la constante de Phillips,  $g$  la constante gravitationnelle, et  $f_m$  le pic de fréquence défini par :

$$f_m = \frac{0.13g}{U_{10}} \quad (1.8)$$

avec  $U_{10}$  la vitesse du vent mesuré à 10 mètres au dessus de la surface.

La distribution directionnelle peut être calculée par :

$$D_{PM}(f, \theta) = N_S^{-1} \cos^{2S}\left(\frac{\theta}{2}\right) \quad (1.9)$$

avec :

$$S = 9.77\left(\frac{f}{f_m}\right)^\mu$$

$$\mu = \begin{cases} 4.06 & \text{si } f \leq f_m \\ -2.34 & \text{si } f > f_m \end{cases}$$

$$N_S = \frac{2^{1-2S}\pi\Gamma(2S+1)}{\Gamma^2(S+1)}$$

et  $\Gamma$  la fonction gamma d'Euler.

La figure 1.3 montre un exemple de spectre directionnel en 3D généré pour une vitesse de vent de  $8km/h$ .

Comme nous le verrons ce spectre a été largement utilisé en synthèse d'images afin de représenter la surface de l'océan en raison de sa simplicité. En effet, seule la vitesse du vent  $U_{10}$  est nécessaire pour connaître la distribution des vagues.

### Spectre de JONSWAP

Le spectre de JONSWAP (**J**oint **N**orth **S**ea **W**ave **P**roject) [Ha73] est une modification du spectre précédent élaborée à l'aide d'observations océanographique. Ce spectre permet d'augmenter le pic de fréquence, ce qui a pour effet de diminuer les amplitudes des vagues dont les fréquences sont proches de ce pic et d'obtenir ainsi une description plus détaillée de la surface. Dans cette approche, la distribution de l'énergie spectrale est calculée de la façon suivante :

$$E_J(f) = E_{PM}(f) \exp[\ln(\gamma) \exp\left(\frac{-(f-f_m)^2}{2\sigma^2 f_m^2}\right)] \quad (1.10)$$

avec :

$$\sigma = \begin{cases} 0.07 & \text{si } f \leq f_m \\ 0.09 & \text{si } f > f_m \end{cases}$$

La distribution directionnelle utilisée dans ce spectre est la même que celle définie par le spectre directionnel de Pierson et Moskowitz (équation 1.9). Ce spectre permet d'obtenir une meilleure caractérisation des vagues et ainsi une meilleure représentation de l'état de la surface de l'océan.

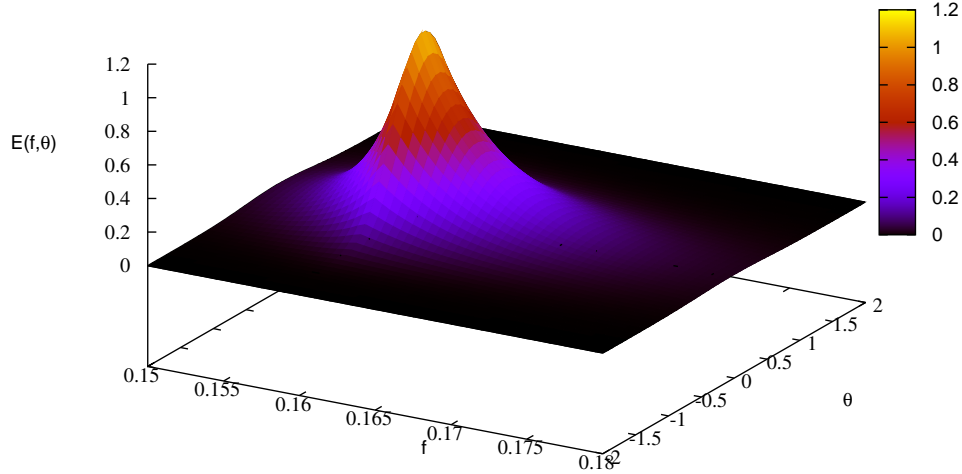


FIG. 1.3 – Spectre directionnel de Pierson-Moskowitz pour une vitesse de vent  $U_{10}$  de  $8\text{km/h}$ .

### 1.2.3 Représentation basée sur la mécanique des fluides

La mécanique des fluides permet de simuler le comportement dynamique d'un fluide de manière physiquement réaliste. Nous présentons ici les équations de Navier-Stokes et leurs approximations dans le cas de faibles profondeurs appelées équations de Saint-Venant.

#### Equations de Navier-Stokes

Dans un repère cartésien, les équations de Navier-Stokes peuvent se formuler sous la forme d'un système de deux équations dont la première représente la conservation de la masse, et la seconde la conservation du moment :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{U} = 0 \quad (1.11)$$

$$\frac{\partial \vec{U}}{\partial t} + \vec{U} \nabla \vec{U} + \frac{\nabla p}{\rho} - \mu \frac{\nabla^2 \vec{U}}{\rho} - \vec{g} = 0 \quad (1.12)$$

avec  $\vec{U} = (u, v, w)$  le vecteur vitesse du fluide,  $\mu$  sa viscosité,  $p$  sa pression,  $\rho$  sa densité, et  $\vec{g}$  le vecteur gravitationnel terrestre de coordonnées  $(0, 9.81, 0)$ . Les opérateurs  $\nabla \vec{U}$  et  $\nabla^2 \vec{U}$  représentent respectivement le gradient et le laplacien du vecteur  $\vec{U}$  et s'expriment par  $\nabla \vec{U} = \left( \frac{\partial u}{\partial t}, \frac{\partial v}{\partial t}, \frac{\partial w}{\partial t} \right)$  et  $\nabla^2 \vec{U} = \frac{\partial^2 u}{\partial t^2} + \frac{\partial^2 v}{\partial t^2} + \frac{\partial^2 w}{\partial t^2}$ .



La première équation garantit la conservation de la masse : l'évolution temporelle de la densité de fluide est constante. La seconde équation garantit la conservation du moment : l'accélération du fluide ( $\frac{\partial \vec{U}}{\partial t}$ ) est égale à la somme des forces auquel il est soumis pondérée par sa densité.

Posons  $f = \vec{U}\nabla\vec{U} + \frac{\nabla p}{\rho} - \mu\frac{\nabla^2\vec{U}}{\rho} - \vec{g}$  : le premier terme ( $\vec{U}\nabla\vec{U}$ ) représente les effets d'advection, c'est à dire le transport du fluide par sa vitesse ; le second terme ( $\frac{\nabla p}{\rho}$ ) celui des forces de pression, le troisième ( $-\mu\frac{\nabla^2\vec{U}}{\rho}$ ) celui des forces de viscosité (plus un fluide sera considéré comme visqueux plus sa vitesse sera ralentie), et enfin le dernier ( $\vec{g}$ ) désigne les forces extérieures.

### Cas de faibles profondeurs : les équations de Saint-Venant

L'approximation de Saint-Venant considère que la dynamique d'un fluide parfait (sans viscosité), incompressible (de volume constant) et irrotationnel dérive des équations de Navier-Stokes dans le cas de faibles profondeurs.

Dans ce contexte, plusieurs approximations peuvent être établies : la première considère que la vitesse verticale du fluide est très faible et constante ; la seconde que la pression verticale est proche de la pression hydrostatique (le champ de pression  $p$  est tel  $p = \rho gh + cste$  avec  $h$  la hauteur du volume d'eau).

Sous ces conditions et par décomposition des équations de Navier-Stokes sous leurs formes cartésiennes on obtient les équations de Saint-Venant :

$$\begin{cases} \frac{\partial h}{\partial t} = -\frac{\partial hu}{\partial x} - \frac{\partial hw}{\partial z} \\ \frac{\partial u}{\partial t} = -\vec{U}\nabla u + \vec{g}\frac{\partial h}{\partial x} \\ \frac{\partial w}{\partial t} = -\vec{U}\nabla w + \vec{g}\frac{\partial h}{\partial z} \end{cases} \quad (1.13)$$

avec  $u$  et  $w$  les vitesses horizontales du fluide (selon l'axe des  $x$  et  $z$ ),  $\vec{U}$  le vecteur vitesse du fluide et  $h$  la hauteur de la colonne d'eau.

### 1.2.4 Bilan

Comme nous venons de le voir, il existe différentes façon de représenter la surface de l'océan. D'une part, les approches paramétriques permettent de représenter la dynamique de la surface en s'appuyant sur des modèles théoriques. Leurs domaines de représentation s'étendent des eaux profondes jusqu'aux eaux peu profondes. D'un autre côté les modèles spectraux permettent d'approximer l'état de la surface de l'océan en s'appuyant sur des observations effectuées en pleine mer. Comme nous

allons le voir dans la section suivante, ces deux approches peuvent être combinées, donnant lieu à des approches hybrides. Enfin, les équations issues de la mécanique des fluides offrent la capacité de représenter la dynamique d'un fluide de façon générale et peuvent être utilisées pour une simulation complète du déferlement.

## 1.3 Simulation de la surface de l'océan en synthèse d'images réalistes

Nous nous intéressons ici aux méthodes de la simulation de la surface de l'océan en informatique graphique. Dans ce contexte, les approches existantes peuvent être classées en 2 catégories : d'un côté les approches dites approximatives, issues des modèles théoriques ou spectraux, et d'un autre celles directement déduites de l'utilisation des équations de la mécanique des fluides. La différence principale entre ces deux catégories est la zone de simulation : en eaux profondes, où l'on observe le plus souvent des mouvements de houle, l'emploi des modèles théoriques ou spectraux suffit pour représenter efficacement ce phénomène. En eaux peu profondes, ces modèles ne suffisent plus, et on doit utiliser les équations de Navier-Stokes, qui elles seules sont capables de simuler le comportement de la surface des vagues près des côtes et leur déferlement. Nous invitons le lecteur à consulter également un survol des méthodes de représentation de la surface de l'eau en synthèse d'images présenté dans [Igl04].

### 1.3.1 Approches approximatives

Les approches approximatives s'appuient sur des modèles théoriques ou des observations expérimentales afin de décrire la surface de l'eau. On peut subdiviser cette famille de méthodes en trois catégories : d'une part, celles représentant la surface directement dans le domaine spatial, celles utilisant le domaine spectral, et enfin celles combinant ces deux approches. Comme nous allons le voir, les approches spatiales permettent de représenter directement la surface avec une carte de hauteur obtenue à l'aide d'une superposition linéaire de fonctions périodiques qui peuvent être facilement animées par simple déphasage. Les approches spectrales utilisent des spectres de vagues pour décrire la surface. Enfin les approches hybrides proposent de combiner ces deux méthodes.

#### Approches spatiales

Le but des approches spatiales est de représenter géométriquement le comportement de la surface de l'eau en l'approximant sous la forme d'une superposition de



FIG. 1.4 – Représentations de la surface de l’océan extraites de [FR86] (à gauche) et [Pea86] (à droite)

fonctions périodiques qui évoluent temporellement par déphasage.

Ce principe est utilisé dès 1981 par [Max81] en combinant une série de fonctions sinusoïdales de hautes et faibles amplitudes. Les hautes amplitudes représentent alors la forme générale des vagues tandis que les faibles amplitudes permettent d’ajouter du détail en représentant les ridelettes à la surface. En utilisant le même principe, [Per85] a proposé d’utiliser une technique basée sur le *bump mapping* [BN76], en perturbant les normales d’un plan à l’aide d’une superposition linéaire de fonctions cycloïdales.

Les auteurs de [Pea86] ont proposé une méthode permettant la simulation de vagues déferlantes en tenant compte du phénomène de réfraction des vagues, à l’aide d’une superposition de fonctions sinusoïdales et cycloïdales représentée sous la forme d’une carte de hauteur.

Parallèlement, Fournier et Reeves [FR86] ont proposé de modifier le modèle de Gerstner décrit en section 1.2.1 afin de simuler des vagues déferlantes. Le principe de cette méthode est la prise en compte de la transformation de la trajectoire des particules d’eau selon les changements topologiques du relief sous-marin afin de représenter la transformation de la trajectoire circulaire de ces particules en un mouvement elliptique, comme cela est le cas dans la réalité. A l’aide de divers paramètres, cette méthode permet aussi de contrôler la géométrie des vagues déferlantes et d’obtenir des crêtes plus ou moins aiguës, ce qui offre un résultat plus réaliste.

Ts’o et Barsky dans [TB87] ont proposé de représenter la réfraction des vagues en se basant sur le même principe que celui de la réfraction de la lumière formulé

par la loi de Descartes (voir section 1.4.3). Leur approche, baptisée *wave tracing*, consiste à générer une surface *spline* en lançant des rayons depuis l'horizon dans une grille 2D uniforme, en progressant à l'aide de l'algorithme de Bresenham. La déviation d'un rayon est calculée en fonction de la différence de profondeur d'une cellule à une autre. Un inconvénient majeur de cette méthode est le manque de détails de la surface générée lorsque les rayons divergent trop. En effet, dans ce cas, peu de rayons définissent la surface du fait de leurs importantes déviations. L'approche proposée par [GS00] permet d'améliorer cette technique en générant de nouveaux rayons, et offre ainsi une meilleure représentation de la surface. Cette approche a été utilisée dans [GM02] afin d'extraire une carte de phases dont les valeurs sont utilisées dans un modèle paramétrique, qui peut alors être facilement animé.

La représentation de la surface sous la forme d'une série de fonctions périodiques s'adapte particulièrement à une implémentation sous carte graphique. De nombreuses méthodes en temps-réel ont donc aussi été proposées.

En 2001, Schneider [SW01] a proposé d'employer une carte de déplacement calculée à l'aide d'un bruit de Perlin [Per85], laquelle est évaluée dans un *vertex shader* sur GPU afin d'obtenir un résultat interactif.

Les auteurs de [IVB02] ont aussi proposé une méthode temps-réel en perturbant un maillage précalculé à l'aide de 4 fonctions sinusoïdales basses fréquences évaluées dans un *vertex shader*. Les détails fins sont obtenus à l'aide d'une technique de *bump mapping* en combinant plusieurs textures. Avec un résultat interactif, cette technique permet de représenter de façon réaliste la surface (voir la figure 1.5) et a permis également de montrer que finalement peu de fonctions suffisent pour parvenir à un résultat visuellement correct.

Les auteurs de [Fin04] ont proposé un raffinement adaptatif d'un maillage pré-calculé et animé avec le modèle de Gerstner. Le principe de leur méthode consiste à adapter le niveau de discrétisation du maillage de la surface de l'océan en fonction de la distance avec l'observateur, en le discrétisant grossièrement dans la zone située loin de l'observateur et plus finement sinon. Ceci a pour effet d'éliminer les hautes fréquences dans les parties discrétisées grossièrement, afin de limiter l'aliassage dû aux effets de moirés à l'horizon et de n'évaluer que certaines fonctions, permettant ainsi un gain en temps de calcul et un résultat interactif.

Dans [Kry05], les auteurs ont proposé de bénéficier des capacités techniques des cartes graphiques nouvelle génération. La nouveauté de leur approche réside dans l'utilisation de *vertex textures* de résolution adaptative en évaluant des cartes de hauteurs représentant respectivement le mouvement des vagues et les détails fins, et



FIG. 1.5 – Simulation de l’océan en temps-réel (extrait de [Kry05])

en augmentant leurs résolutions selon la distance entre la caméra et le plan définissant la surface de l’eau, ce qui permet d’adapter les calculs nécessaires selon le point de vue.

**Avantages et inconvénients :** Bien que ce type d’approche permette de représenter une variété d’effets, tels que le déferlement ou la réfraction des vagues, l’animation de la surface semble peu réaliste car elle ne s’appuie pas sur des données physiques ou proche de la réalité. On observe la plupart du temps des motifs répétitifs, ce qui limite cette catégorie de méthodes.

### Approches spectrales

Ce type de représentation consiste à décrire la surface de l’océan à l’aide d’une distribution spectrale de vagues issue de données théoriques ou expérimentales. Le passage au domaine spatial est effectué en utilisant une transformée de Fourier inverse, ce qui permet d’obtenir une caractérisation géométrique de la surface sous la forme d’une carte de hauteur, en corrélant la hauteur en un point avec son niveau de gris correspondant. L’utilisation d’un spectre de vagues permet un résultat plus réaliste car il décrit mieux les caractéristiques de la surface de l’océan.

Ce principe a été utilisé dans [MWM87] afin de synthétiser une image de bruit blanc à l’aide du spectre de Pierson et Moskowitz (voir section 1.2.2) qui est ensuite affichée sous la forme d’une carte de hauteur après passage dans le domaine spatial.

Bien que les résultats observés semblent réalistes pour une vue éloignée, la forme des vagues semble grossière pour une vue de près car peu de détails apparaissent en raison de la résolution de l'image utilisée. Pour obtenir une meilleure précision dans ce cas, il faudrait alors générer une nouvelle image, ce qui limite considérablement cette approche.

Pour pallier à cet inconvénient, Tessendorf [Tes01] a proposé alors une nouvelle méthode qui fut notamment utilisée dans la production des films *Waterworld* ou *Titanic*. Le principe de cette approche suit l'idée précédente en calculant les composantes spectrales de manière pseudo-aléatoire, en combinant un spectre de *Phillips* et un générateur aléatoire gaussien, ce qui se justifie en raison de la distribution gaussienne des vagues souvent observée en pleine mer par les océanographes. Afin d'obtenir des détails fins à la surface, chaque composante est perturbée à l'aide d'une fonction de bruit, permettant d'accentuer les crêtes des vagues et produisant un effet plus réaliste, quelque soit la résolution de l'image de bruit blanc synthétisée.

Plus récemment, des approches interactives ont aussi été proposées.

Dans [HVT<sup>+</sup>06], les auteurs ont proposé une adaptation par niveau de détails de l'approche précédente en représentant la surface de l'océan sous la forme de deux surfaces. La première, discrétisée finement, est animée à l'aide d'une carte de déplacement évaluée dans un *vertex shader*, tandis que la seconde est discrétisée de façon adaptative et est traitée sous la forme d'un *bump mapping* à l'aide d'un *pixel shader*. Cette méthode permet de représenter une vaste surface en n'animant que la partie située aux alentours de l'observateur et permet ainsi un résultat interactif (autour de 100 images par secondes calculée sur une carte graphique GeForce 3).

Dans [Mit05], l'auteur a aussi proposé une implémentation GPU en n'animant que les fréquences générant une perturbation significative de la surface. En suivant le même principe que l'approche de [Tes01], cette méthode consiste à synthétiser une image de bruit blanc à l'aide du spectre de *Phillips* et à séparer les fréquences en deux groupes. Les basses fréquences, permettant de définir le profil général des vagues, sont stockées dans une carte de déplacement et évaluées dans un *vertex shader*. Les hautes fréquences, représentant les détails fins, sont stockées dans une carte de normales utilisée au moment du rendu. Cette technique permet d'animer une seule partie du spectre et d'obtenir un gain en temps de calcul.

Les auteurs de [CC06] ont étendu cette méthode en proposant une facétisation par niveaux de détails en projetant une grille plane adaptative selon la distance avec le plan de vue sur un plan horizontal définissant la surface de l'eau. Les régions éloignées sont alors automatiquement facétisées en basse résolution tandis que les



FIG. 1.6 – Simulation de la surface de l’océan par la méthode de [Tes01]

régions près de l’observateur sont discrétisées plus finement.

**Avantages et inconvénients :** En s’appuyant sur des données physiques, ce type d’approche permet d’obtenir une représentation de la surface proche de la réalité observée. Cependant, elle nécessite d’utiliser une transformée de Fourier inverse pour passer dans le domaine spatial, ce qui peut être assez coûteux.

### Approches hybrides

Le principe des approches hybrides est de combiner les approches spatiales et spectrales, ce qui permet d’obtenir une représentation géométrique de la surface tout en ayant une description proche de la réalité des composantes définissant les trains de vagues.

En partant sur ce principe, l’approche proposée par [TDG00] combine ainsi une superposition linéaire de trochoïdes dont les caractéristiques sont synthétisées à l’aide du spectre de Pierson et Moskowitz. Afin d’obtenir des détails fins, une fonction de turbulence 3D est utilisée. Cette technique permet d’obtenir un résultat très réaliste de la surface de l’océan.

Les auteurs de [PA01] s’appuient sur l’idée précédente en calculant la hauteur en un point de la surface à l’aide d’une superposition de fonctions gaussiennes dont les caractéristiques sont obtenues dans le domaine fréquentiel à l’aide du spectre de JONSWAP. Cet article présente aussi une nouvelle méthode de rendu de la surface



FIG. 1.7 – Simulation de la surface de l'océan extraite de [TDG00]

de l'océan permettant la simulation de divers phénomènes tels que l'écume ou la diffusion sous-surfacique. Nous verrons cette approche plus en détail dans les sections 1.4.2 et 1.4.3.

Dans [HNC02], les auteurs proposent un schéma par niveaux de détails de la technique de [TDG00] en adaptant le niveau de discrétisation du maillage de la surface selon sa distance à l'observateur et en réduisant le nombre de composantes trochoïdales selon cette distance. En n'évaluant qu'une partie de ces composantes au cours de l'animation, cette approche permet de réduire les calculs nécessaires à la simulation et d'obtenir un résultat proche de l'interactivité (environ 20 à 30 images par secondes sur une carte graphique GeForce 2).

**Avantages et inconvénients :** Ces méthodes possèdent l'avantage d'être facilement contrôlables puisqu'elles permettent d'obtenir à la fois une description géométrique de la surface dont les caractéristiques sont issues de données physiques, produisant ainsi une animation simplifiée et réaliste. L'un des désavantages provient du fait qu'elles nécessitent un grand nombre de fonctions périodiques afin de générer une surface proche de la réalité, et donc un coût mémoire et en temps de calcul importants durant l'animation.



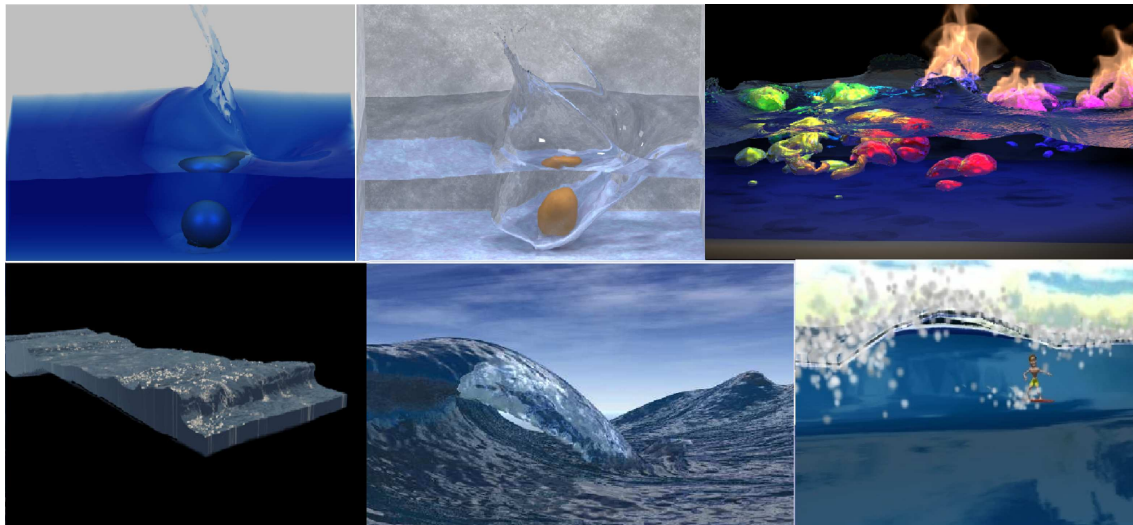


FIG. 1.8 – Simulation eulérienne (de gauche à droite et de haut en bas) de [FF01], [EMF02], [LSSF06], [EMF02], [MMS04] et de [TSS<sup>+</sup>07]

### 1.3.2 Approches physiques

Les approches physiques permettent de capturer la complexité dynamique de l'écoulement d'un fluide en résolvant les équations de Navier-Stokes vues dans la section 1.2.3. Pour résoudre ces équations il existe principalement deux méthodes de discrétisation : d'une part à l'aide de grilles 2D ou 3D, c'est ce que l'on appelle les approches eulériennes, et d'autre part les systèmes de particules qui forment les approches lagrangiennes. Les approches hybrides combinent ces deux techniques et, comme nous le verrons, permettent d'ajouter des détails aux approches eulériennes tels que des bulles ou des embruns. Un excellent tutorial sur la simulation des fluides en synthèse d'images est présentée dans [BMFG06]. En raison de la grande diversité des phénomènes que ces approches peuvent simuler, nous nous concentrerons principalement sur la simulation de liquide, et plus particulièrement sur l'océan.

#### Approches Eulériennes

Dès 1990, les auteurs de [KM90] ont proposé d'utiliser une approche physiquement réaliste afin de simuler la surface de l'eau. Le principe de leur méthode consistait à résoudre les équations de Saint-Venant en 2D en obtenant une carte de hauteur. Cette méthode a été étendue par les auteurs de [CL95] en utilisant les équations de Navier-Stokes en 2D, permettant ainsi de prendre en compte les effets de pressions et d'obtenir une modulation de la surface plus réaliste.

Il faudra attendre 1996 pour obtenir une résolution pleinement 3D des équations

de Navier-Stokes. Foster et Fedkiw [FM96] proposent alors d'introduire la méthode de [HW65] issue de la physique en informatique graphique et dont le principe consiste à voxeliser l'espace de façon uniforme. Les vitesses sont alors définies au centre des huit faces de chaque cellule et la pression est définie au centre de chaque cellule. Le gradient et le laplacien de la vitesse d'une face sont alors calculés en utilisant le théorème des différences finies entre les faces des cellules adjacentes, ce qui permet de résoudre l'équation 1.12. Le divergent de chaque cellule est calculé en fonction des vitesses mises à jour de chaque face. Afin de garantir l'incompressibilité du fluide, le champ de pression d'une cellule est calculé en fonction de ce divergent. Cette approche a été étendue ensuite dans [FM97] afin d'ajouter un contrôle utilisateur et de simuler une plus grande variété de phénomènes tels que des explosions.

Dans la méthode présentée par [FM96], des particules virtuelles sont placées dans la grille afin d'extraire la surface du fluide, c'est à dire de la localiser spatialement. Cependant, la surface obtenue ne semble que peu réaliste. Ce problème a été partiellement résolu par Foster et Fedkiw [FF01] qui ont proposé de représenter la surface du fluide sous la forme d'une fonction implicite ("*level-set*") évoluant selon la vitesse du fluide [OS88] et pouvant ainsi être visualisée. Cependant, on constate dans les résultats une compressibilité du volume de fluide, se traduisant principalement par une perte visible du volume de fluide au cours de l'animation.

Pour résoudre ce problème, la méthode proposée par [EMF02] consiste alors à placer des particules au-dessus et au-dessous de la surface, lesquelles sont advectées en fonction de la vitesse de la cellule à laquelle elles appartiennent. Une surface implicite est alors calculée en fonction de la position de ces particules et la surface implicite finale représentant la surface du fluide est obtenue par interpolation entre ces deux fonctions implicites. Du fait de sa capacité à s'adapter à tous types de fluides (et pas seulement aux liquides), cette approche, baptisée "*particles level-set method*", connaît un très grand succès et a été utilisée à des fins diverses, notamment par exemple dans la représentation des interactions fluides/solides [CMT04, WMT05]. Elle a été aussi récemment étendue afin de représenter les interactions entre plusieurs liquides [LSSF06]. Cependant, le principal désavantage de cette méthode reste son coût de calcul. Dans ce contexte, les auteurs de [LGF04] ont proposé d'utiliser une structure de données adaptative de type *Octree* en raffinant les cellules proches de l'interface. Ce type d'approche permet d'obtenir un comportement dynamique très réaliste de la surface de l'eau et elle peut être aussi utilisée dans le cadre de scènes océaniques.

L'article présenté par [EMF02] propose également de simuler les vagues déferlantes

en conditionnant les vitesses du système à l'aide d'équations paramétriques [RO98]. Bien que les résultats semblent réalistes, cette technique permet de représenter principalement des vagues glissantes (voir la figure 1.8 en bas à gauche).

Les auteurs de [MMS04] ont proposé de simuler des vagues déferlantes en 2D en conditionnant les vitesses de chaque cellule à l'aide de l'approche paramétrique de [TDG00]. Une librairie de profils de vagues déferlantes est alors calculée, permettant à un utilisateur de choisir la vague désirée. La simulation 3D de ce phénomène est alors obtenue par extension du profil choisi selon l'axe perpendiculaire à la plage.

Les auteurs de [TSS<sup>+</sup>07] ont proposé une simulation de vagues déferlantes en temps réel obtenue en résolvant les équations de Saint-Venant en 2D. Une vague déferlante est obtenue en détectant la région la plus anguleuse de la carte de hauteur représentant le front de la vague et en apposant un maillage représentant le front plongeant. Chaque sommet du maillage est alors projeté sur la grille puis advecté selon la vitesse de la cellule correspondante.

**Avantages et inconvénients :** Le principal avantage des approches eulériennes réside dans leur capacité à simuler une grande variété de phénomènes, et elles profitent également du fait qu'elles ont été longtemps étudiées en synthèse d'images. Cependant, pour obtenir des détails fins tels que des embruns ou des bulles, elles requièrent un niveau de discrétisation très fin (une grille très échantillonnée), ce qui peut engendrer un coût de calcul considérable.

### Approches lagrangiennes

Le principe des approches lagrangiennes est de discrétiser un fluide sous la forme d'un système de particules soumises à des lois dynamiques, permettant ainsi d'obtenir un résultat physiquement réaliste avec un niveau de détail très fin. Ce type de méthode a été introduit en informatique graphique par Reeves dès 1983 dans [Ree83] pour la simulation dynamique de phénomènes naturels tels que l'eau, le feu, les nuages ou la fumée.

Cette catégorie de méthode a connu plusieurs extensions dont la première fut introduite par les travaux de [MP89] afin de simuler les interactions entre les particules, en les connectant entre elles à l'aide de ressorts afin de représenter les forces d'attraction ou de répulsion entre deux particules voisines. Ceci a permis d'obtenir une simulation de l'écoulement d'un liquide visqueux pour la première fois avec un système de particules. Cette approche a été étendue ensuite par [TPF89] afin de simuler

le passage progressif d'un fluide du domaine solide au domaine liquide en changeant les constantes de raideur de chaque ressort. Les auteurs de [Ton91] ont proposé la simulation de ces mêmes effets en résolvant l'équation de diffusion de chaleur sur un système de particules classique.

Cependant, bien que les résultats semblent réalistes, ce type de méthode ne permet de simuler la complexité dynamique d'un fluide, que seule la résolution des équations de Navier-Stokes permet de capturer. C'est dans ce contexte que les auteurs de [SF95] introduisent la première fois la notion de SPH en informatique graphique. Les *Smoothed Particle Hydrodynamics* sont une approche issue de l'astrophysique par [LL77] utilisée dans l'étude de la dynamique de nébuleuses gazeuses dans l'espace, et dont le formalisme fut étendu par Monaghan dans [Mon92] au cas des fluides en général. Le principe cette approche consiste alors à représenter les particules sous la forme de surfaces potentielles (plus connues sous le noms de *blobs*) influençant chacune ses voisines et en calculant les forces exercées sur une particule en fonction de son voisinage. Cette approche fut étendue par les travaux de [DC96] et [SAC<sup>+</sup>99] afin de représenter respectivement les déformations d'objets élastiques et l'écoulement de la lave, en modélisant le transfert de chaleur à l'intérieur de la lave et en calculant la viscosité du fluide en fonction de la température.

Plus récemment, [MCG03] a adapté cette méthode au cas des liquides peu compressibles comme l'eau. En proposant entre autres de nouvelles fonctions de pondération basées sur les travaux en physique de [GM82], une définition de la tension de surface physiquement réaliste selon la méthode de [Mor99], l'apport de leurs travaux dans ce domaine est considérable et a permis de démocratiser cette approche, très utilisée depuis dans la simulation de phénomènes divers tels que les cheveux [HMT01], la représentation de matériaux granuleux [BYM05], la simulation de l'écoulement du flux sanguin en simulation chirurgicale [MST04], de rivière [KW06], ou même les déformations d'objets élastiques ou plastiques [MKN<sup>+</sup>04] [KAG<sup>+</sup>05] [SSP06] (voir la figure 1.9).

Concernant la simulation de liquides, cette méthode a été étendue dans [CBP05] afin de représenter la dynamique de fluides visqueux en la couplant avec une approche de type masses-ressorts en suivant le principe de [MP89], en soumettant chaque particule à la fois aux lois de Navier-Stokes mais aussi à la dynamique du ressort auquel elle appartient, ce qui permet de réduire la distance entre les particules voisines et de ralentir le mouvement du fluide en fonction de la raideur des ressorts. L'approche de [MCG03] a été également étendue dans [MSKG05] afin de représenter les inter-

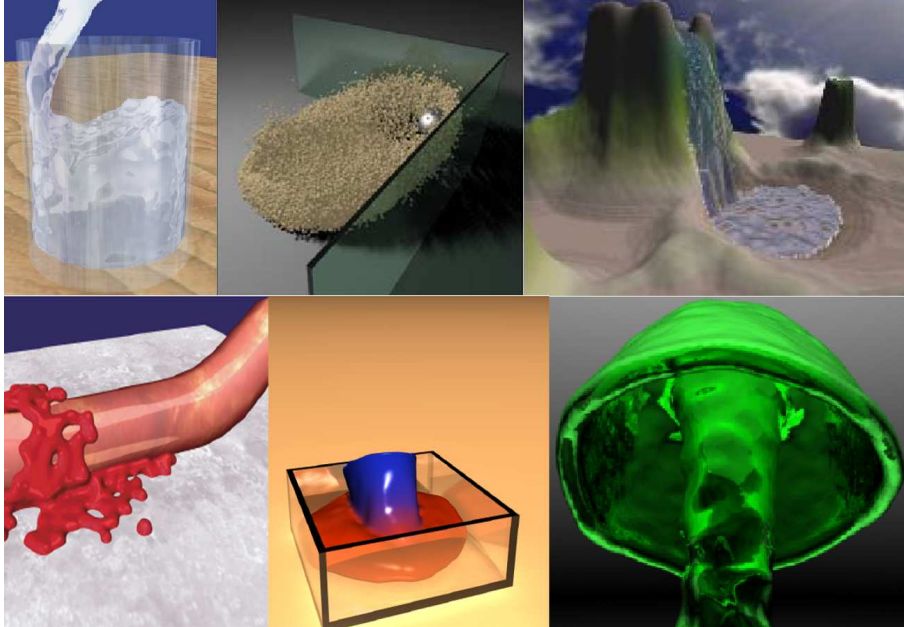


FIG. 1.9 – Simulation *SPH* (de gauche à droite et de haut en bas) de [MCG03], [BYM05], [KW06], [MKN<sup>+</sup>04], [KAG<sup>+</sup>05] et de [CBP05]

actions entre différents fluides de différentes phases. Le principe de leur technique consiste à modifier les forces de viscosité entre deux particules différentes afin de tenir compte des natures des fluides mis en jeu. Les auteurs proposent aussi de simuler les bulles générées par le bouillonnement de l'eau en faisant varier la densité au repos des particules selon la température du fluide afin de représenter leur gazéification.

Cependant, l'approche des particules SPH possède deux inconvénients. D'une part, on observe le plus souvent des effets de compressibilité, c'est à dire un changement du volume total du fluide au cours du temps, ce qui limite le réalisme des simulations. D'autre part, pour simuler une grande étendue d'eau, elle requiert un très grand nombre de particules donc une charge importante de calcul et d'occupation mémoire.

Pour pallier au premier problème, plusieurs méthodes ont été proposées.

Les auteurs de [PTB<sup>+</sup>03] ont introduit en informatique graphique la méthode des MPS (*Moving Particle Semi-implicit*) issue de la physique [KO96]. Cette approche est une extension de la méthode des SPH et consiste à modifier la densité d'une particule et à relaxer sa pression en résolvant une équation de Poisson permettant de garantir l'incompressibilité du fluide. Cette technique a été notamment utilisée

en 2D pour la simulation de vagues déferlantes dans [WZC<sup>+</sup>06] en calculant une série de profils et les juxtaposant afin d'obtenir une surface 3D. Cette méthode présente aussi une technique de génération des embruns marins dûs au déferlement en ajoutant un système de particules classique et en les advectant selon la vitesse des particules de la simulation.

Les auteurs de [BT07] ont proposé une nouvelle méthode afin de garantir l'incompressibilité dans une approche SPH dont le principe consiste à modifier la pression d'une particule en tenant compte de sa variation de densité par rapport à sa densité au repos. En effet, le fait de relaxer la pression permet d'agir directement sur les forces qui permettent de représenter l'attraction ou la répulsion entre deux particules. En augmentant ou en diminuant ces effets les auteurs parviennent ainsi à conserver une distance acceptable entre les particules, et donc une conservation du volume total de fluide quasi-constante.

Des schémas adaptatifs ont aussi été proposés afin de réduire le nombre de particules et donc de limiter les coûts de calcul et l'occupation mémoire que peut nécessiter cette approche pour la simulation d'importants volumes de fluide.

Dans [DC99], les auteurs proposent de fusionner ou diviser des particules selon leurs densités. La masse totale du système est conservée en recalculant la masse d'une particule chaque fois qu'elle subit l'une de ces deux opérations. Afin de conserver la symétrie des forces entre deux particules de masses différentes, une méthode de *shooting / gathering* est utilisée qui consiste à redistribuer les forces entre les particules de façon à ce qu'une particule reçoive et exerce la même force sur ses voisines. Bien que cette approche permette d'avoir un gain en temps de calcul par rapport à une approche standard, des artefacts visuels sont visibles durant l'animation en raison de la fusion ou division instantanée d'une particule à la surface du fluide.

Les auteurs de [HHK07] proposent aussi une simulation adaptative en faisant varier la taille et le nombre des particules selon leurs distances à des couches horizontales placées arbitrairement au préalable. Tout comme la technique de [DC99], la masse est recalculée dès lors qu'une particule est fusionnée ou divisée. Bien que cette méthode soit efficace dans les exemples illustrés, elle nécessite que l'utilisateur sache où placer les couches et ne peut fonctionner que dans ces cas bien précis.

Les auteurs de [APKG07] étendent et améliorent considérablement la méthode proposée par [DC99] en fusionnant ou divisant des particules selon leurs distances à

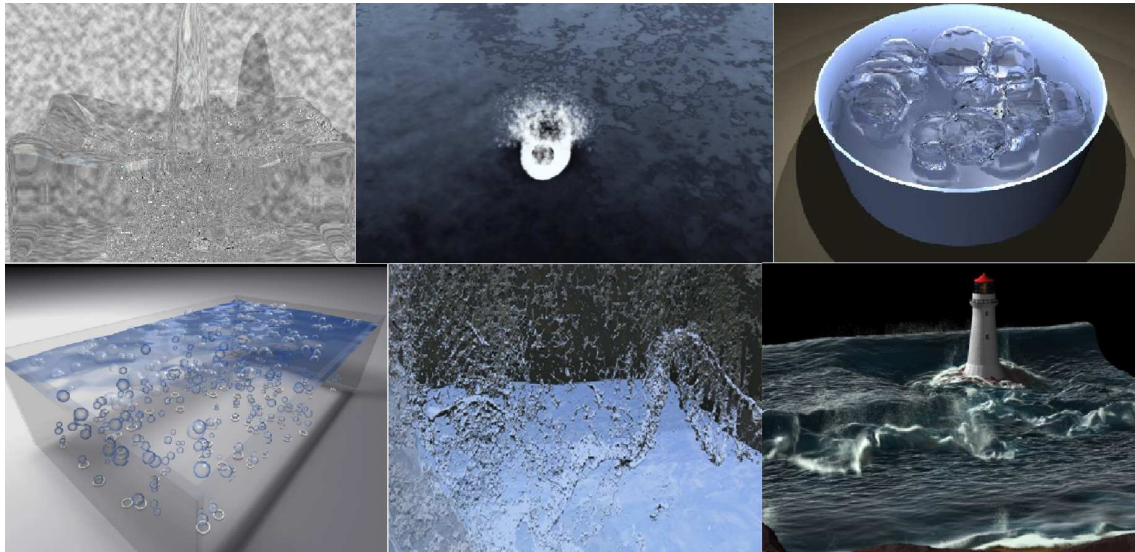


FIG. 1.10 – Simulation hybride (de gauche à droite et de haut en bas) de [GH04], [TRS06], [ZYP06], [TRS06], [KCC<sup>+</sup>06] et de [LTKF08]

l’observateur et à l’interface calculée préalablement à l’aide d’une méthode par *Marching Cubes* [LC87]. L’utilisation de ces critères permet de rassembler ou diviser les particules situées à l’intérieur du fluide et donc d’éviter les artefacts visuels observés dans la méthode de [DC99]. Afin de garantir le respect de la symétrie des forces, la méthode de *shooting / gathering* est adaptée au cas des forces de pression et de viscosité formulées par [MCG03].

**Avantages et inconvénients :** Nous avons vu que les méthodes lagrangiennes permettent de simuler de manière réaliste une grande variété de phénomènes. La force incontestable de ces approches réside dans leur capacité à simuler des détails fins, non représentables à l’aide d’approches eulériennes. Cependant, elles nécessitent la plupart du temps de prendre en compte un très grand nombre de particules. Bien que des schémas adaptatifs ont été proposés, ils nécessitent des calculs supplémentaires mais permettent de réduire les coûts de calculs pour la simulation d’un large système de particules.

### Approches hybrides

Le principe des approches hybrides est de simuler du fluide à l’aide d’une approche eulérienne et les détails fins, tels que l’écume, les embruns ou les bulles, à l’aide d’une approche lagrangienne. Ceci permet de profiter des avantages des catégories de méthodes étudiées précédemment et d’obtenir des résultats très réalistes (voir la figure 1.10).

En partant de ce contexte, les auteurs de [OH95] ont proposé de coupler une approche eulérienne avec un système de particules afin de représenter des embruns. Le principe de cette approche consiste à résoudre les équations de Navier-Stokes en 2D et à générer des embruns marins en fonction de la magnitude de la vitesse verticale de chaque cellule, en les advectant en fonction de la vitesse de la cellule à laquelle ils appartiennent.

Les auteurs de [TFK<sup>+</sup>03] proposent de coupler une simulation eulérienne avec un système de particules générées dans les régions de haute courbure de la surface et en les advectant de manière indépendante.

Les auteurs de [GH04] proposent de coupler la méthode des *Particles Level-Set* [EMF02] avec un système de particules générées dans les régions de haute courbure de la surface et en les advectant en fonction de la vitesse de la cellule à laquelle ils appartiennent et de la force de friction exercée par le fluide, puis en faisant interagir chaque particule avec ses voisines.

L'approche de [ZYP06] étend cette méthode en déformant chaque bulle en fonction de la tension de surface du fluide afin d'éviter d'avoir des bulles parfaitement sphériques.

Les auteurs de [KCC<sup>+</sup>06] proposent d'utiliser la méthode des *Particles Level-Set* adaptative proposée par [LGF04] en générant des particules selon la variation temporelle du volume de fluide dans chaque cellule. Le principe de leur technique consiste à utiliser des particules de marquage situées sous la surface et de les transformer en particules d'eau au moment du rendu. Le nombre de particules transformées est calculé en fonction du volume d'eau perdu dans une cellule, ce qui permet de caractériser la turbulence de la cellule considérée et de générer ainsi des particules dans les régions très agitées.

Dans [TRS06], les auteurs proposent de coupler une simulation 2D des équations de Saint-Venant (voir section 1.2.3) et un système de particules générées en fonction de la vitesse de chaque cellule et de la tension de surface présente à l'interface. Cette méthode a été étendue dans [TSS<sup>+</sup>07] en utilisant un système de particules SPH afin de représenter les interactions entre les différentes particules.

Dans le même esprit, la technique proposée par les auteurs de [LTKF08] consiste à combiner la méthode de [EMF02] avec un système de particules SPH dont le nombre est calculé en fonction du divergent du fluide, permettant ainsi de simuler des bulles ou des embruns marins générés pendant le déferlement.



**Avantages et inconvénients :** En ajoutant des détails aux approches eulériennes à l'aide des techniques lagrangiennes, cette catégorie de méthodes permet d'obtenir des résultats très réalistes et tente de se développer en se basant sur les progrès effectués dans le cadre des simulations SPH.

### 1.3.3 Bilan

Comme nous l'avons vu, il existe principalement deux grandes techniques pour représenter la surface de l'eau. D'un côté les approches approximatives permettent la plupart du temps de simuler la surface de l'océan en eaux profondes, soit en la caractérisant géométriquement, soit en utilisant des spectres de vagues ou soit en combinant ces deux méthodes. D'autre part, les méthodes physiquement réalistes s'appuient sur la résolution des équations de la mécanique des fluides et permettent de représenter la dynamique de la surface de l'eau de façon très réaliste, et peuvent être donc employées pour la simulation des vagues déferlantes. Cependant, aucune méthode proposée à ce jour ne permet de représenter ce phénomène dans son intégralité et sa diversité, au sens des différents sous-processus l'accompagnant (comme les embruns marins), et les différents types de déferlement que nous avons vus en section 1.1.2.

## 1.4 Rendu de la surface de l'océan

Nous nous intéressons ici aux méthodes de rendu réaliste de la surface de l'océan en synthèse d'images. Dans un premier temps, nous verrons les méthodes permettant de visualiser la surface. Puis, afin d'enrichir le rendu, nous verrons les approches permettant de simuler les phénomènes nécessaires, tels que les échanges lumineux, l'écume et les embruns marins.

### 1.4.1 Visualisation de la surface libre

Afin de pouvoir rendre la surface de l'eau, nous devons dans un premier temps la visualiser. Comme nous allons le voir, il existe principalement deux types de représentation : la représentation explicite, pouvant être directement traitée par le pipeline graphique et la représentation implicite consistant à décrire la surface sous la forme d'un champ scalaire. Nous proposons ici d'étudier ces deux catégories dans le cadre de la représentation de scènes océaniques.

### Surface explicite

La représentation explicite consiste à décrire la surface sous la forme d'un ensemble de primitives géométriques pouvant être directement affiché à l'écran, comme par exemple à l'aide d'un ensemble de polygones [IVB02] [Fin04] [HVT<sup>+</sup>06] [CC06] [HNC02], d'une surface NURBS [SW01] ou de *Bêta-Splines* [TB87]. Ce type de représentation permet d'obtenir un rendu quasi-immédiat car elle peut être directement traitée par le pipeline graphique. Cependant, dans le cas de vastes surfaces, elle nécessite souvent d'utiliser un grand nombre de primitives, ce qui a conduit de nombreuses approches à proposer des schémas adaptatifs, tels que ceux présentés en section 1.3.1. .

### Surface implicite

Le représentation implicite consiste à décrire la surface sous la forme d'un champ scalaire, autrement dit sous la forme d'une fonction définie de  $\mathbb{R}^3$  dans  $\mathbb{R}$  qui permet d'associer en tout point de l'espace 3D une valeur réelle. Usuellement, une valeur positive permet de représenter l'espace situé au-dessus de la surface, négative, en dessous, et proche de 0, aux alentours de celle-ci. Pour visualiser cette surface, on peut soit la rendre directement à l'aide d'une technique de lancer de rayons, soit recourir à une approche de reconstruction vers une surface explicite. C'est ce que nous présentons dans les paragraphes suivants.

**Lancer de rayons :** La technique du lancer de rayons, développée dans [Whi80] permet de visualiser des surfaces en prenant en compte les effets de réflexions, réfractions et d'ombres portées, permettant ainsi de produire des images photoréalistes. Le principe de cette méthode consiste à envoyer des rayons dans chaque pixel de l'image depuis un observateur puis de procéder de façon récursive. Dans le cas de surfaces implicites, cette approche permet de visualiser la surface en calculant les points d'intersections avec chaque rayon. Cependant, un calcul analytique de l'intersection avec les rayons est souvent difficile. Il existe notamment deux approches permettant de remédier à ce problème : d'une part la recherche incrémentale [KH84] et d'autre part, le *sphere-tracing* [Har96].

La recherche incrémentale consiste à parcourir le rayon d'un pas constant puis à procéder par dichotomie jusqu'à ce que la fonction implicite soit proche de 0. Cette technique a été notamment utilisée dans le cadre du rendu de scènes océaniques par [Tho01] et [Gon99]. Bien que cette méthode soit efficace, elle peut s'avérer très lente si le pas choisi est très petit. Dans le cas inverse, pour un pas de départ très grand, on risque de ne pas trouver l'intersection.

Une méthode permettant de pallier à ces problèmes est celle proposée par le *Sphere-Tracing* développé par Hart [Har96] afin de visualiser des surfaces complexes représentées sous forme implicite et dont le but consiste à calculer une distance minimale de parcours sur le rayon garantissant de ne pas traverser la surface, ce qui permet de ne pas procéder par dichotomie comme dans le cas précédent. Elle est donc plus rapide mais le calcul de la distance de parcours est souvent délicat. Nous proposons d'étendre cette approche dans le cadre de la visualisation de la surface océanique dans le chapitre suivant.

**Extraction de maillage :** Une technique souvent employée afin de visualiser une surface implicite est d'en extraire une surface explicite pouvant alors être directement affichée à l'écran. C'est ce que propose l'algorithme du *marching-cube* proposé par [LC87] qui consiste à extraire un maillage 3D structuré et uniforme d'une iso-surface. Le principe de cette méthode est de diviser l'espace 3D en cubes élémentaires et de parcourir chaque cube en calculant la valeur de la fonction implicite pour chaque sommet du cube. Selon le nombre de valeurs au-dessus d'un seuil définissant la proximité de la surface (généralement proche de 0), une configuration d'un ensemble de triangles est alors retenue, permettant de représenter explicitement la surface. Cette technique est très utilisée en synthèse d'images, notamment dans le cadre des approches lagrangiennes afin de visualiser la surface définie par les systèmes de particules.

### 1.4.2 Rendu de l'écume et des embruns marins

Nous nous intéressons ici aux méthodes de rendu de l'écume et des embruns marins. Comme nous l'avons vu en section 1.3.2, les approches physiques de type lagrangiennes ou hybrides permettent une simulation de ces phénomènes. Nous nous concentrerons donc sur le cas des scènes océaniques étudiées en section 1.3.1. Parmi ces méthodes, nous pouvons distinguer deux catégories d'approches, d'une part les approches phénoménologiques qui définissent la quantité d'écume en un point de la surface, et d'autre part les systèmes de particules qui peuvent être utilisés pour générer les embruns marins.

#### Approches phénoménologiques

Le but de ce type d'approche est de définir une quantité d'écume en se basant sur des critères empiriques.

Les auteurs de [JG01] proposent de décrire cette quantité en fonction de la hauteur



FIG. 1.11 – Rendu de l'écume par la méthode extraite de [JG01] (à droite)

du point considéré. Le principe de cette méthode consiste à calculer la différence de hauteur entre ce point et ces voisins et de générer une quantité d'écume si cette différence est supérieure à un certain seuil, rendue en plaquant une texture d'écume dont la transparence dépend de la quantité d'écume au point considéré. Cette méthode a été étendue dans [JBS03] en calculant cette quantité selon l'amplitude des vagues au point considéré.

Le problème principal de ces méthodes provient de l'animation de l'écume, celle-ci ne se déplaçant pas en fonction de la surface. Un autre inconvénient se situe au niveau des heuristiques employées. En se basant sur de simples critères de hauteur, elle ne tiennent pas compte des conditions atmosphériques permettant de justifier physiquement l'apparition de ce phénomène. Des observations océanographiques ont permis par exemple de montrer que l'écume ne se forme qu'à partir d'une vitesse de vent supérieur à  $13\text{km/h}$  [Mun47].

Dans [PA01] les auteurs se basent sur des observations faites par [MM86] afin de calculer la quantité d'écume en prenant en compte la vitesse du vent et la différence de température entre l'eau et l'air. En induisant des paramètres physiques, cette formulation permet de tenir compte des conditions atmosphériques nécessaires à l'apparition de ce phénomène et peut permettre ainsi à un utilisateur de moduler cette quantité en jouant avec les conditions présentes dans l'environnement virtuel. Cependant, comme le montre la figure 1.11, les résultats semblent limités. De plus, comme dans les méthodes précédentes, l'animation reste problématique.

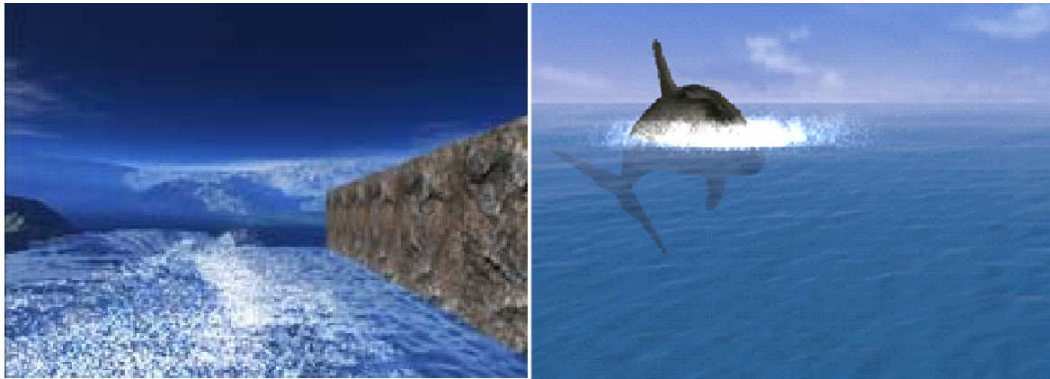


FIG. 1.12 – Rendu des embruns marins par la méthode extraite de [WZC<sup>+</sup>06] (à gauche) et celle extraite de [CC06] (à droite)

### Système de particules

Dans [Pea86], les auteurs utilisent un système de particules afin de représenter les embruns marins générés lors du déferlement, rendus sous la forme d'un nuage de points.

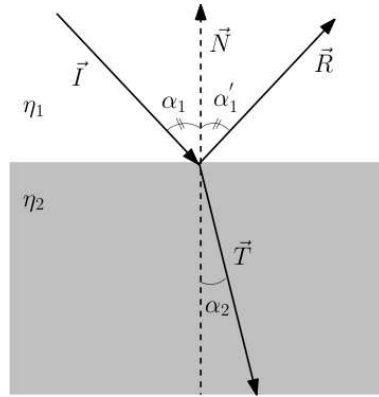
Les auteurs de [WZC<sup>+</sup>06] utilisent également une approche similaire. En représentant des vagues déferlantes à l'aide d'une approche lagrangienne, les auteurs génèrent alors un autre système de particules dont le nombre est directement déduit de la vitesse des particules de simulation.

Les auteurs de [HW04] proposent de générer les particules d'écume en fonction de l'amplitude des vagues au point considéré et de les rendre également sous la forme d'un nuage de points. Les auteurs de [JG01][JBS03][CC06] utilisent aussi une approche similaire afin de générer des embruns marins en fonction de la variation temporelle de hauteur en point.

Bien que ces méthodes soient efficaces (voir la figure 1.12), elles nécessitent de considérer et contrôler une très grande quantité de particules, et peuvent s'avérer très coûteuses pour de grandes étendues maritimes.

### 1.4.3 Interactions surfaciques et sous-surfaciques

Nous nous intéressons ici aux méthodes de représentation des interactions surfaciques et sous-surfaciques en synthèse d'images. Après quelques considérations physiques sur les échanges entre la lumière et l'eau, nous présentons les approches permettant de simuler et/ou d'approximer ces échanges.

FIG. 1.13 – Illustration des lois de réflexion et de réfraction de *Snell-Descartes*

### Un peu de physique...

Les échanges lumineux à la surface de l'eau sont caractérisés par de multiples réflexions et réfractions. En effet, lorsqu'un rayon heurte la surface de l'eau celui-ci peut être à la fois réfléchi, réfracté et absorbé. Le rayon réfléchi peut lui-même être réfléchi et transmis en heurtant la surface ou d'autres objets. La partie transmise peut elle-même être réfléchie ou réfractée et ainsi de suite. Il existe deux manières de représenter ces échanges : d'une part, en faisant appel aux lois de l'optique géométrique et d'autre part, en décrivant la lumière sous sa forme ondulatoire.

**Représentation géométrique** Les directions de réflexion et de réfractions peuvent être calculées en utilisant les lois de *Snell-Descartes*. La première loi indique que l'angle de réflexion  $\alpha'_1$  est égale à l'angle d'incidence  $\alpha_1$  (voir la figure 1.13). La direction d'un rayon réfléchi  $\vec{R}$  peut être calculé alors par :

$$\vec{R} = \vec{I} - 2\vec{N}(\vec{N} \cdot \vec{I}) \quad (1.14)$$

Concernant la réfraction, son angle de déviation peut être calculé en fonction des indices de réfraction  $\eta_1$  et  $\eta_2$  des milieux considérés :

$$\eta_1 \sin \alpha_1 = \eta_2 \sin \alpha_2$$

Les indices de réfraction pour l'air et l'eau sont respectivement de 1.0 et 1.334.

La direction du vecteur réfracté  $\vec{T}$  peut être alors calculée par :

$$\vec{T} = \eta_r(\vec{N} \cdot \vec{I}) - \sqrt{1 - \eta_r^2(1 - (\vec{N} \cdot \vec{I})^2)}\vec{N} - \eta_r\vec{I} \quad (1.15)$$

avec  $\eta_r = \eta_1/\eta_2$ .

En considérant la lumière comme non polarisée, les coefficients de *Fresnel* permettent de déterminer la quantité de lumière réfléchie  $K_r$  et transmise  $K_t$  :

$$K_r = \frac{1}{2} \left( \frac{\sin(\alpha_2 - \alpha_1)}{\sin(\alpha_2 + \alpha_1)} \right)^2 + \frac{1}{2} \left( \frac{\tan(\alpha_2 - \alpha_1)}{\tan(\alpha_2 + \alpha_1)} \right)^2 \quad (1.16)$$

$$K_t = 1 - K_r \quad (1.17)$$

Dans [Sch94], l'auteur propose une approximation de ces coefficients permettant de réduire les calculs. Le coefficient de réflexion  $K_r$  est alors formulé par :

$$K_r = R_0 + (1 - R_0)(1 - \cos \alpha_1)^5 \quad (1.18)$$

avec  $R_0 = \frac{1 - \eta_r^2}{1 + \eta_r^2}$ . Le coefficient  $K_t$  est calculé de la même façon que précédemment, en utilisant l'équation 1.17.

**Représentation ondulatoire** Lorsqu'un rayon traverse la surface de l'eau, une partie de ce rayon est diffusée, absorbée et réémise à cause de la présence de particules en suspension. Pour représenter cette complexité et déterminer la quantité de lumière en un point dans le volume d'eau considéré, il faut faire appel à la représentation corpusculaire de la lumière qui consiste à décrire un rayon lumineux sous la forme d'une onde caractérisée par une longueur d'onde. L'équation de transfert radiatif formulée ci-après permet alors de déterminer la quantité de lumière de longueur d'onde  $\lambda$  en un point de l'espace de coordonnées sphériques  $(z, \theta, \phi)$  :

$$\begin{aligned} dL_\lambda(z, \theta, \phi) = & -a_\lambda L_\lambda(z, \theta, \phi) + a_\lambda(z) L_{\lambda emis}(z, \theta, \phi) \\ & -b_\lambda(z) L_\lambda(z, \theta, \phi) + \frac{b_\lambda(z)}{4\pi} \int \int L_\lambda(z, \theta, \phi) \Phi_\lambda(z, \theta, \phi) d\theta d\phi \end{aligned}$$

avec  $a_\lambda$  et  $b_\lambda$ , respectivement le coefficient d'atténuation et de diffusion à la longueur d'onde  $\lambda$ ,  $L_{\lambda emis}(z, \theta, \phi)$  la quantité de lumière réémise et  $\Phi_\lambda(z, \theta, \phi)$ , la fonction de phase décrivant la distribution angulaire de l'énergie diffusée. Cette équation peut être séparée en quatre termes :

- le premier terme désigne la quantité de lumière absorbée et dépend directement du coefficient d'atténuation  $a(\lambda)$  ;
- le second terme désigne la quantité de lumière réémise ;
- le troisième terme désigne la diffusion sortante, c'est à dire la quantité de lumière diffusée en direction de la surface ;
- le dernier terme représente la diffusion entrante, c'est à dire la quantité de lumière diffusée en direction inverse de la surface.

Les coefficients d'atténuation  $a(\lambda)$  et de diffusion  $b(\lambda)$  peuvent être déterminés en utilisant les lois formulées respectivement par [Mor91] et [GM83] qui permettent

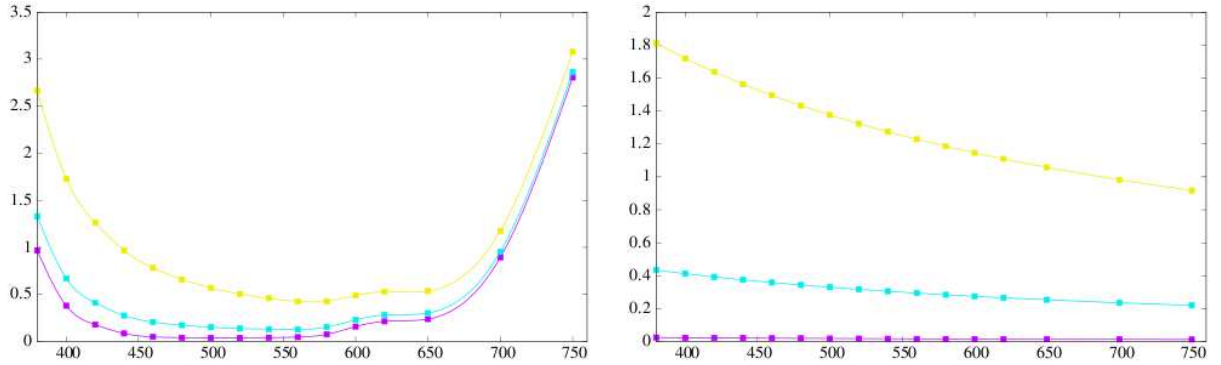


FIG. 1.14 – Spectres d'absorption (à gauche) et de diffusion (à droite) de l'eau selon la concentration en phytoplancton (en jaune  $C = 10\text{mg}/\text{m}^3$ , en bleu  $C = 1\text{mg}/\text{m}^3$  et magenta  $C = 0.01\text{mg}/\text{m}^3$ )

d'exprimer ces coefficients en fonction de la concentration en phytoplancton  $C$  contenue dans l'eau :

$$a(\lambda) = (a_w(\lambda) + 0.06C^{0.65})(1 + 0.02e^{-0.014(\lambda-380)}) \quad (1.19)$$

$$b(\lambda) = \frac{550}{\lambda}0.30C^{0.32} \quad (1.20)$$

avec  $a_w(\lambda)$  le coefficient d'atténuation de l'eau pure pour une longueur d'onde  $\lambda$  (voir Annexe A).

Ces équations permettent de montrer l'influence de la concentration en phytoplancton sur les spectres d'atténuation et de diffusion de l'eau (voir la figure 1.14). Ainsi, plus une eau sera chargée en particules de phytoplancton, plus elle apparaîtra turbide, c'est à dire trouble.

### Méthodes de rendu des interactions surfaciques et sous-surfaciques en synthèse d'images

Comme nous allons le voir il existe différentes manières de simuler les échanges lumineux à la surface de l'océan en synthèse d'images. Nous étudierons ici les approches dédiées aux scènes océaniques.

Une première approche est d'utiliser directement un lancer de rayons en appliquant les lois de l'optique géométrique et en utilisant les coefficients de Fresnel ou l'approximation de Schlick [Sch94], puis en en atténuant l'intensité  $I$  d'un rayon lumineux incident à l'aide de la loi de Beer-Lambert de manière exponentielle en fonction de sa distance parcourue  $d$  et d'un coefficient d'atténuation  $a$  :

$$I = I(0)e^{-ad} \quad (1.21)$$



avec  $I(0)$  étant l'intensité d'origine du rayon. Dans le cas de la réflexion ce rayon peut être atténué en fonction de l'épaisseur optique constituée par les particules en suspension dans l'air. Dans le cas de la réfraction il est atténué en fonction du coefficient d'atténuation de l'eau [Tes01].

Afin de réduire les calculs engendrés par la technique du lancer de rayons ou d'obtenir des applications interactives, de nombreuses méthodes préfèrent recourir à des techniques plus simples, facilement implémentables sous GPU.

L'utilisation de placage d'environnement [BN76] permet d'approximer les effets de réflexion et de réfraction. Cette approche consiste à projeter un environnement virtuel soit directement sur une sphère, soit en prenant six vues différentes de la scène et en les plaquant sur un cube. Dans les deux cas, on suppose que les objets et l'observateur sont infiniment lointains, auquel cas on peut considérer que les rayons incidents sont tous parallèles et donc que les rayons réfléchis ne dépendent que de la normale à la surface. Cette technique est très utilisée en synthèse d'images, notamment pour le rendu surface de l'océan [SW01][JG01] et peut être utilisée pour la réfraction [BD06] en utilisant une texture représentant le fond sous-marin.

Une autre alternative consiste à plaquer une texture (dans notre cas le ciel) directement sur la surface, en la considérant comme un réflecteur parfait, considérée comme un miroir [HVT<sup>+</sup>06][Bel03].

Il existe aussi des approches permettant de calculer la diffusion sous-surfacique dans le cadre de scènes océaniques. Comme nous l'avons vu ces échanges sont complexes car ils ne peuvent être calculés qu'en résolvant l'équation de transfert radiatif.

Dans ce contexte, les auteurs de [IDN02] tentent d'approximer cette équation en discrétisant le volume d'eau à l'aide de *volumes d'illumination* formés par les rayons réfractés et le fond de l'océan, lesquels sont ensuite découpés en sous-volumes. La quantité de lumière réémise vers la surface est alors calculée en accumulant celle calculée dans chaque sous-volume d'illumination. Les auteurs ont étendu cette méthode dans [IDN03] en discrétisant le volume d'eau à l'aide de couches planes. La lumière réémise à la surface est calculée en chaque point d'intersection entre le rayon réfracté et une couche, la quantité de lumière totale réémise vers la surface étant alors obtenue en accumulant ces intensités le long du rayon réfracté.

Les auteurs de [PA01] et [CS04] proposent de calculer la diffusion sous-surfacique en prenant en compte divers paramètres bio-optiques tels que la concentration en chlorophylle et le type de l'eau classé par [Jer76] selon sa turbidité, afin de représenter

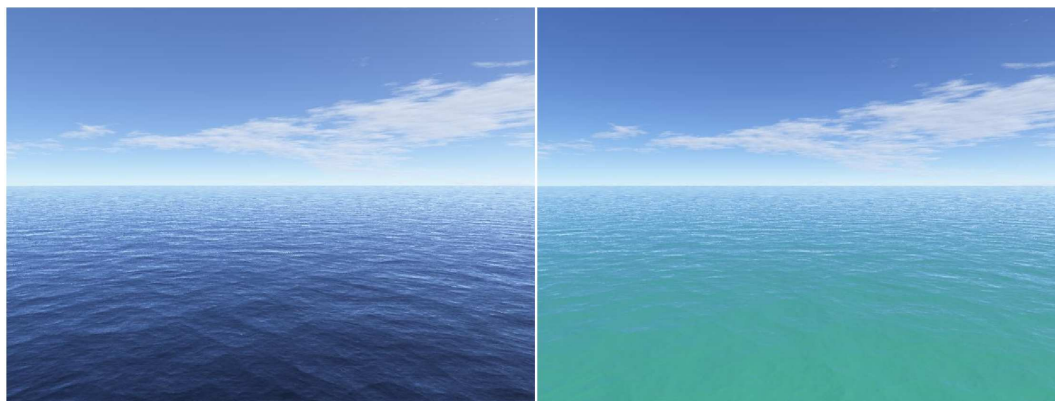


FIG. 1.15 – Rendu de la surface de l’océan pour une eau profonde (à gauche) et eau claire (à droite) par la méthode présentée dans [PA01]

des océans plus ou moins chargés en particule en suspension. Comme le montre la figure 1.15, les résultats produits avec cette approche sont très réalistes.

Il existe aussi dans la littérature d’autres méthodes permettant de représenter la diffusion sous-surfacique en considérant l’eau comme un milieu participant. Nous invitons le lecteur à consulter [PCPS97] afin de voir un aperçu des méthodes existantes.

#### 1.4.4 Bilan

Comme nous venons de le voir il existe dans la littérature de nombreuses méthodes permettant de prendre en compte des effets divers, tels que la diffusion sous-surfacique, l’écume et les embruns marins. Bien que les résultats obtenus apparaissent très réalistes, il n’existe cependant aucune approche proposant une méthode de rendu unifiée représentant à la fois l’ensemble de ces phénomènes. Nous proposons dans le chapitre qui suit une alternative à ce problème, autorisant à la fois une approximation de l’ensemble de effets, tout en permettant d’accélérer les temps de calcul et de réduire l’occupation mémoire nécessaire à la simulation de cet ensemble de phénomènes.

## 1.5 Conclusion

Nous avons présenté dans ce chapitre un état de l’art de l’ensemble des méthodes capables de simuler et de rendre la surface de l’océan.

Dans un premier temps, nous avons vu qu'il existait principalement deux types d'approches pour reproduire le comportement dynamique de la surface de l'océan : d'une part les approches approximatives et, d'autre part, celles utilisant les équations de la mécanique des fluides. La différence principale entre ces deux types de méthode réside dans la zone d'étude. Comme nous l'avons vu, en eau profondes, les modèles approximatifs suffisent pour caractériser le mouvement de la surface. En eaux peu profondes, ces modèles ne conviennent plus et nous devons utiliser des modèles physiques. Nous avons vu également qu'il existait plusieurs techniques pour résoudre ces équations, dont l'approche des *SPH* permet le mieux d'obtenir des détails fins. Cependant le problème principal est qu'elle nécessite un important coût de calcul qui peut être réduit à l'aide d'une approche adaptative.

Concernant la visualisation, nous avons vu qu'il existait de nombreuses méthodes pouvant simuler divers phénomènes, tels que l'écume, les embruns marins, les échanges lumineux surfaciques et sous-surfaciques. Nous avons vu également que la recherche du point d'intersection durant le processus de lancer de rayons était problématique avec une définition complexe de la surface, comme cela est souvent le cas pour les approches spatiales et hybrides. Nous proposons dans le chapitre qui suit une approche alternative permettant une accélération de ce processus et une méthode de rendu unifiée permettant d'approximer et d'accélérer le calcul de l'ensemble de ces phénomènes.

## Chapitre 2

# Représentation de l’océan en eau profonde

Nous présentons dans ce chapitre une nouvelle méthode de rendu unifiée de scènes océaniques en eau profonde. Ces travaux ont donné lieu à une communication internationale [DCG07a].

Dans un premier temps, nous étudierons la méthode proposée afin de permettre une visualisation rapide de la surface de l’océan, particulièrement adaptée au modèle de représentation paramétrique que nous avons choisi. Nous verrons successivement les approches que nous avons proposées afin de simuler l’écume, l’atténuation lumineuse à la surface due aux particules en suspension dans l’air, la diffusion sous-surfacique et pour finir, l’éblouissement. Nous nous attacherons à montrer comment notre algorithme de rendu permet de réduire les temps de calculs et l’occupation mémoire dûs à la représentation de ces phénomènes pour une large étendue d’eau. Enfin, nous discuterons des performances et des limites qu’offre notre méthode et nous conclurons en examinant les différentes perspectives.

## 2.1 Visualisation de la surface par Sphere-Tracing

### 2.1.1 Approche Générale

Comme nous l’avons vu dans le chapitre précédent, une méthode commune pour la visualisation de la surface libre est celle employée par la recherche incrémentale qui consiste à avancer le long du rayon d’un pas constant puis à procéder par dichotomie afin de trouver le point d’intersection entre un rayon et la surface, ce qui peut se révéler très lent.

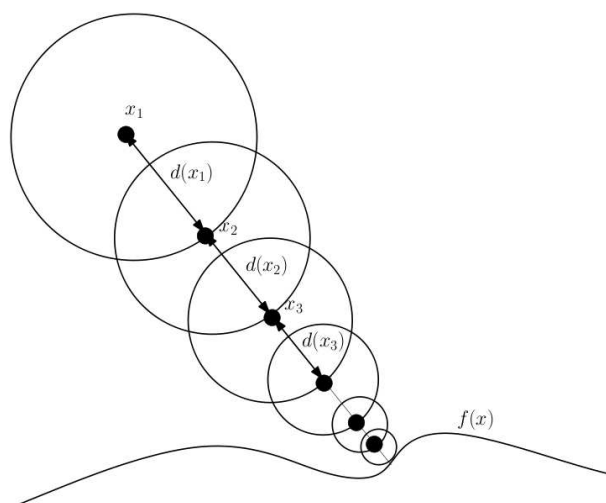


FIG. 2.1 – Principe du *Sphere-Tracing* de [Har96]

Une approche permettant de remédier à ce problème évoqué est celle proposée par le Sphere-Tracing [Har96] que nous avons vu au chapitre précédent et qui consiste à trouver une distance  $d$  de parcours le long du rayon sans traverser la surface.

**Théorème 1.** *Une fonction  $f$  est dite  $\lambda$ -lipschitzienne s'il existe un réel  $\lambda$  (appelée borne lipschitzienne) tel que pour deux points quelconques  $x$  et  $y$  de l'espace on a :*

$$|f(x) - f(y)| \leq \lambda|x - y|$$

*En utilisant cette propriété, on peut calculer une fonction de distance  $d$  en tout points  $x$  de l'espace telle que :*

$$d(x) = |f(x)|/\lambda$$

En supposant la fonction implicite comme étant lipschitzienne, l'approche du Sphere-Tracing consiste à avancer sur le rayon d'une distance  $d$ , garantissant de ne pas traverser la surface, et en procédant de façon récursive tant que l'on n'a pas intersecté la surface ou que la distance parcourue sur le rayon soit inférieure à une distance maximale de parcours (voir figure 2.1).

Une optimisation de cette méthode est celle présentée dans [HW96] pour le cas des hypertextures et pouvant s'adapter à tout types de surfaces définies à l'aide d'une somme de fonctions, comme c'est le cas pour les surfaces représentées par les approches spatiales ou hybrides.

**Théorème 2.** *La borne lipschitzienne  $\lambda$  d'une somme de fonctions est égale à la somme des constantes lipschitziennes  $\lambda_i$  de chacune des fonctions.*

$$\lambda = \sum_{i=1}^N \lambda_i$$

**Théorème 3.** *La borne lipschitzienne d'une fonction constante est nulle.*

Si la fonction  $f$  est définie sous la forme d'une somme de fonctions  $n_i$ , on peut calculer une fonction de distance  $d$  en tout point  $x$  de l'espace en utilisant le théorème 2 :

$$d(x) = \sum_{i=1}^N \frac{n_i(x)}{\lambda_i} \quad (2.1)$$

avec  $\lambda_i$  la borne lipschitzienne de la composante  $n_i$ .

En négligeant certaines constantes lipschitzienne  $\lambda_i$ , on peut alors maximiser la fonction de distance  $d$ . En partant de cette idée, les auteurs de [HW96] proposent de négliger certaines constantes selon la contribution de chaque primitive qui constitue la surface. Pour cela, on compare chaque  $n_i$  avec son minimum global. Si la différence entre ces deux valeurs est proche de 0, alors cette primitive peut être approchée à l'aide de son minimum. Étant donné que la constante lipschitzienne d'une fonction constante est nulle (théorème 3), la borne lipschitzienne résultante est plus petite dans ce cas et on peut ainsi maximiser le pas sur le rayon.

D'autres optimisations sont aussi possibles en s'appuyant sur la cohérence spatiale et/ou temporelle, c'est à dire en avançant sur le rayon de la même façon qu'un rayon voisin déjà traité (cohérence spatiale) ou sur celui calculé à l'instant précédent (cohérence temporelle).

Nous proposons ici d'étendre cette méthode dans le cas de la surface de l'océan en maximisant le pas sur le rayon afin d'accélérer le processus du calcul d'intersection.

### Modélisation de la surface

Pour modéliser la surface, nous avons choisi d'utiliser le modèle proposé par Thon [TDG00] afin de représenter la surface de l'eau. Comme nous l'avons vu, cette approche hybride offre un compromis entre réalisme physique et contrôlabilité utilisateur, puisqu'il permet d'injecter des données spectrales issues de l'océanographie dans une approche paramétrique. Dans ce modèle <sup>1</sup>, la hauteur  $y$  d'un point de la

---

<sup>1</sup>Nous avons utilisé le spectre de Pierson-Moskowitz afin de synthétiser les caractéristiques de chaque train de vagues.

surface de coordonnées horizontales  $x$  et  $z$  est définie par :

$$y = \sum_{i=1}^N A_i \cos(k_i(x \cos \theta_i + z \sin \theta_i)w_i t + \phi_i) \quad (2.2)$$

avec :

- $A_i \geq 0$ , l'amplitude de la vague,
- $k_i$ , son nombre d'onde, qui peut être calculé en fonction de sa longueur d'onde par  $k_i = 2\pi/\lambda_i$ ,
- $\theta_i$ , son angle de déviation,
- $\omega_i$ , sa pulsation, qui peut être calculée en fonction de sa fréquence  $\omega_i = 2\pi f_i$  ;
- $\phi_i$ , son déphasage.

### Calcul de la fonction implicite

En utilisant la formule 2.2, nous pouvons définir une fonction  $f(p)$  en tout point  $p$  de coordonnées  $(x, y, z)$  par :

$$f(p) = y - \sum_{i=1}^N A_i \cos(k_i(x \cos \theta_i + z \sin \theta_i) - w_i t + \phi_i) \quad (2.3)$$

avec :

- $f(p) > 0$  si  $p$  est au-dessus de la surface ;
- $f(p) = 0$  si  $p$  est situé à la surface ;
- $f(p) < 0$  si  $p$  est sous la surface.

Montrons que cette fonction est lipschitzienne.

**Théorème 4.** *une fonction est dite lipschitzienne si et seulement si sa dérivée est bornée*

La fonction  $f$  étant définie de  $\mathbb{R}^3$  dans  $\mathbb{R}$ , montrons que son vecteur gradient est borné. Le gradient de la fonction  $f$  peut être calculé par :

$$\nabla f(p) = \begin{pmatrix} \sum_{i=1}^N A_i k_i \cos \theta_i \sin(k_i(x \cos \theta_i + z \sin \theta_i) - w_i t + \phi_i) \\ 1 \\ \sum_{i=1}^N A_i k_i \sin \theta_i \sin(k_i(x \cos \theta_i + z \sin \theta_i) - w_i t + \phi_i) \end{pmatrix}$$

Or,  $\forall X \in \mathbb{R}$  nous avons :  $-1 \leq \cos(X) \leq 1$  et  $-\frac{\pi}{2} \leq \sin(X) \leq \frac{\pi}{2}$ . Le gradient de la fonction  $f$  peut être borné par :

$$\begin{pmatrix} -\sum_{i=1}^N \frac{\pi}{2} A_i k_i \cos \theta_i \\ 1 \\ -\sum_{i=1}^N \frac{\pi}{2} A_i k_i \sin \theta_i \end{pmatrix} \leq \nabla f(p) \leq \begin{pmatrix} -\sum_{i=1}^N \frac{\pi}{2} A_i k_i \cos \theta_i \\ 1 \\ -\sum_{i=1}^N \frac{\pi}{2} A_i k_i \sin \theta_i \end{pmatrix} \quad (2.4)$$

soit :

$$\begin{pmatrix} -\sum_{i=1}^N \frac{\pi}{2} A_i k_i \\ 1 \\ -\sum_{i=1}^N \frac{\pi^2}{4} A_i k_i \end{pmatrix} \leq \nabla f(p) \leq \begin{pmatrix} -\sum_{i=1}^N \frac{\pi}{2} A_i k_i \\ 1 \\ -\sum_{i=1}^N \frac{\pi^2}{4} A_i k_i \end{pmatrix}$$

Le vecteur gradient de la fonction  $f$  étant bornée, on peut donc conclure que cette fonction est bien lipschitzienne.

### Calcul du minimum global

En reformulant l'équation 2.3, on obtient :

$$f(p) = \sum_1^N \left( \frac{y}{N} - A_i \cos(k_i(x \cos \theta_i + z \sin \theta_i) - w_i t + \phi_i) \right)$$

soit :

$$f(p) = \sum_1^N n_i(p)$$

avec :

$$n_i(p) = \frac{y}{N} - A_i \cos(k_i(x \cos \theta_i + z \sin \theta_i) - w_i t + \phi_i) \quad (2.5)$$

Or, nous avons :

$$n_i(p) \geq \frac{y}{N} - A_i$$

Or, pour tout point situé au-dessus de la surface, on a :  $y \geq \sum_{j=1}^N A_j \cos(k_j(x \cos \theta_j + z \sin \theta_j) - w_j t + \phi_j) \geq -\sum_{j=1}^N A_j$ . On a donc :

$$n_i(p) \geq -A_i - \left( \sum_{j=1}^N A_j \right) / N$$

Le minimum global  $m_i$  de chaque fonction  $n_i$  peut alors être calculé en posant :

$$m_i = -A_i - \left( \sum_{j=1}^N A_j \right) / N \quad (2.6)$$

### Calcul de la borne lipschitzienne

Comme cela est mentionnée dans[Har96], nous pouvons calculer la borne lipschitzienne  $\lambda$  d'une fonction  $f$  en utilisant la relation suivante :

$$\lambda \geq \|\nabla f(p)\|$$

avec  $p$  un point quelconque de l'espace. En utilisant le théorème 2, nous avons :

$$\lambda = \sum_{i=1}^N \lambda_i \geq \sum_{i=1}^N \|\nabla n_i(p)\|$$



avec  $\lambda_i$  la borne lipschitzienne de chaque fonction  $n_i$ . En utilisant l'équation 2.4, nous posons  $\lambda_i$  telle que :

$$\lambda_i = \sqrt{\frac{1}{N^2} + \left(\frac{\pi}{2} A_i k_i \cos \theta_i\right)^2 + \left(\frac{\pi}{2} A_i k_i \sin \theta_i\right)^2}$$

soit :

$$\lambda_i = \sqrt{\frac{1}{N^2} + \left(\frac{\pi}{2} A_i k_i\right)^2} \quad (2.7)$$

### Maximisation du pas sur le rayon

Au départ, nous calculons un plan horizontal d'équation  $y = \sum_{i=1}^N A_i$  situé au-dessus de la surface. Nous calculons alors l'intersection du rayon avec ce plan de façon analytique. Si une intersection est trouvée, nous recherchons le point d'intersection à la surface en maximisant le pas. Pour cela, nous procédons de façon similaire à la méthode proposée par [HW96]. Nous comparons la différence entre la valeur de chaque fonction  $n_i$  (équation 2.5) et son minimum global  $m_i$  (équation 2.6). Si cette différence est supérieure à un certain seuil (choisi par expérimentation à  $10^{-1}$ ) alors cela signifie que la hauteur à la surface engendrée par la présence de cette primitive est non négligeable et sa borne lipschitzienne est calculée (équation 2.7). Dans le cas contraire, elle n'est pas retenue et sa borne est négligée, ce qui permet de maximiser le pas sur le rayon.

La figure 2.2 montre une capture de ce processus sur une surface très perturbée avec 30 composantes initiales. Dans cet exemple, seules 7 composantes contribuent au calcul du pas, les 23 autres n'étant pas significatives.

Comme nous le verrons dans la partie 2.5 cette méthode permet d'accélérer considérablement le calcul des points d'intersection par rapport à une recherche incrémentale usuelle.

#### 2.1.2 Anti-aliasage

Afin de pallier aux problèmes d'aliasage nous proposons d'utiliser une méthode basée sur un filtrage fréquentiel injecté directement dans le calcul du point d'intersection [TDG00]. Pour cela, nous réduisons l'amplitude d'une composante contributive au calcul du pas en fonction de la distance avec l'observateur, en la négligeant si son amplitude est proche d'un certain seuil. Nous utilisons un filtrage de type linéaire. L'amplitude  $A'$  de chaque fonction en un point  $p$  est alors calculée par la formule suivante :

$$A'(p) = \frac{15A}{D(O, p)} \quad (2.8)$$

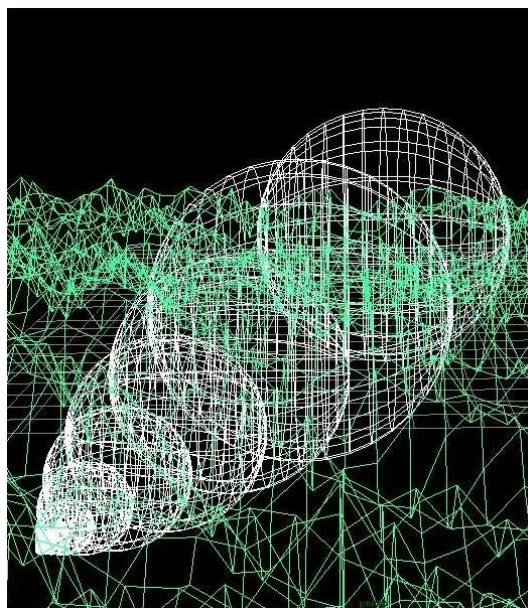


FIG. 2.2 – Illustration de notre méthode avec une surface très perturbée - Les rayons des sphères représentent les pas consécutifs sur le rayon.

avec  $D(O, p)$  la distance entre le point considéré et l'observateur. Nous négligeons une composante contributive si son amplitude filtrée passe en dessous d'un certain seuil, fixé à  $10^{-3}$  et choisi expérimentalement.

La figure 2.3 illustre un résultat de notre méthode, dans lequel les effets d'aliasage sont réduits. Nous pouvons noter que notre approche permet de limiter efficacement les phénomènes de moirés apparents à l'horizon, caractéristiques des scènes océaniques sans anti-aliasage (voir figure 1.7). En négligeant certaines composantes, elle permet aussi de réduire les temps de calculs nécessaires à la recherche du point d'intersection entre un rayon et la surface (voir la section 2.5).

### 2.1.3 Cohérence spatiale et cohérence temporelle

Afin d'accélérer les calculs nous employons la cohérence spatiale et la cohérence temporelle [HW96] décrite précédemment ; les buts respectifs de ces deux approches étant d'approximer le point d'intersection d'un rayon en s'appuyant sur celui calculé pour son voisin (cohérence spatiale) ou en fonction de celui calculé à l'instant précédent (cohérence temporelle).

Pour la cohérence spatiale, nous avançons le long du rayon de pas égaux au rayons des sphères du rayon voisin tant que la fonction implicite est positive. Dans le cas où elle devient négative, nous reculons et nous procédons de la même manière que



FIG. 2.3 – Surface de l'eau sans aliassages obtenue avec notre méthode

précédemment. Ce principe est illustré sur le schéma 2.4.

Pour la cohérence temporelle, nous avançons le long du rayon jusqu'au point d'intersection à l'instant précédent. Si la fonction implicite est positive, nous avançons alors sur le rayon à l'aide de la méthode générale. Dans le cas où la fonction est négative, nous procédons à un nouveau calcul du point d'intersection. Ce principe est illustré sur le schéma 2.5.

Durant l'animation, nous utilisons d'abord la cohérence temporelle pour le calcul du point d'intersection du premier rayon. Pour les rayons voisins nous choisissons entre la cohérence spatiale et temporelle selon la distance parcourue le long du rayon et le signe de la fonction implicite : si la fonction implicite est positive dans les deux cas, nous parcourons alors la distance la plus grande entre celles calculées par la cohérence spatiale et la cohérence temporelle. Si dans les deux cas nous obtenons une fonction implicite négative, nous procédons de façon générale.

## 2.2 Rendu de l'écume

Comme nous l'avons vu dans le chapitre précédent, il existe deux types de méthodes afin de simuler l'écume à la surface de l'océan : d'une part les approches phénoménologiques qui calculent une quantité d'écume en chacun des points de la surface et d'autre part les systèmes de particules. Le problème majeur de cette deuxième catégorie de méthode est qu'elle requiert une très grande occupation mé-

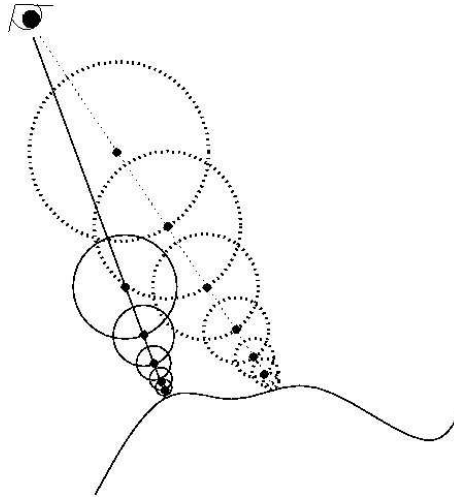


FIG. 2.4 – Cohérence spatiale - En pointillé, les sphères précalculées d'un rayon voisin. En plein, celles calculées pour le rayon courant.

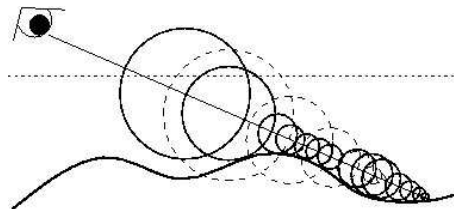


FIG. 2.5 – Cohérence temporelle - En pointillé, les sphères précalculées à l'instant  $t$ . En plein, celles calculées à l'instant  $t + 1$ .

moire dans le cadre de scènes océaniques. Nous proposons ici d'étendre l'approche proposée par [PA01] en améliorant la génération de l'écume dans les régions de la surface soumises aux plus fortes variations et en proposant une méthode d'animation en accord avec celle de la surface.

Dans cette approche, la quantité d'écume  $s$  en un point de la surface est calculée en fonction de divers paramètres météorologiques, tels que la vitesse du vent  $U_{10}$  et la différence de température entre l'eau  $T_w$  et l'air  $T_a$  :

$$s = 1.59 \cdot 10^{-5} U_{10}^{2.55} e^{0.086 \cdot (T_w - T_a)} \quad (2.9)$$

Comme nous l'avons évoqué précédemment, les résultats fournis avec cette approche ne semblent pas réalistes car elle ne favorise pas la simulation de ce phénomène selon les caractéristiques de la surface de l'eau, le peu de nappes observées (après un grand zoom) étant dispersées de façon purement aléatoire. Or, comme nous pouvons l'observer dans la réalité certaines régions de la surface les plus exposées au vent sont favorisées par l'apparition de cet effet, comme le haut des crêtes des vagues. De plus les régions les plus anguleuses soumises à de plus grandes variations de la surface dues à la présence de turbulence sous-surfacique peuvent également faire apparaître ce phénomène.

En partant de ces observations, nous proposons d'adapter la formule afin de tenir compte de ces critères. La quantité d'écume  $s'$  en un point de hauteur  $y$  est calculée en fonction de sa hauteur par :

$$s' = s \left( \frac{y - y_{min}}{y_{max} - y_{min}} \right)^n \|\vec{N}\| \quad (2.10)$$

avec  $\vec{N}$  le vecteur normal au point considéré calculé par  $\vec{N} = \nabla f(p)$  et  $y_{max} = \sum_{i=1}^N A_i$  et  $y_{min} = -\sum_{i=1}^N A_i$ , la hauteur maximale et minimale de la surface. Le paramètre  $n$  nous permet de contrôler l'étalement de l'écume sur une crête. Après plusieurs expérimentations, nous avons trouvé que le meilleur résultat était obtenu en choisissant  $n$  entre 6 et 9. Afin d'obtenir un résultat non uniforme, nous ajoutons un bruit de Perlin filtré selon la distance avec l'observateur, de la même façon que le filtrage fréquentiel utilisé afin de réduire l'aliassage.

Chaque point possédant une quantité d'écume est appelé "point d'écume" caractérisé aussi par un âge, initialement mis à 1 et décrétementé de 0.01 à chaque étape de l'animation. Pour rendre ce point, nous modifions sa luminance  $L_{foam}$  en fonction de son âge, comme cela est suggéré dans [Koe84] :

$$L_{foam} = s' f_{ef} R_e \quad (2.11)$$

où  $f_{ef}$  représente l'âge de ce "point d'écume", et  $R_e$  la réflectance de l'écume pure.

La luminance de ce point est finalement calculée par :

$$L(0) = L_{sky} + L_{sun} \cos \theta_{sun} + L_{foam} \quad (2.12)$$

avec  $L_{sky}$  (resp.  $L_{sun}$ ) la luminance du ciel (resp. du soleil).

Pour pallier aux problèmes d'animation de l'écume évoqués dans le chapitre précédent, nous proposons de stocker la position et l'âge des "point d'écume" à la surface dans une texture 2D ayant la même taille que l'image calculée. Pendant l'animation, nous déplaçons les points stockés dans la texture en utilisant les formules de la houle de Gestner étendues au cas 3D [Tho01] selon les axes x,y et z :

$$\begin{cases} x_{t+1} = x_t + \sum_{i=0}^N A_i \sin(k_i(x_0 \cos \theta_i + z_0 \sin \theta_i) - \omega_i t + \phi_i) \\ y_{t+1} = y_0 - \sum_{i=0}^N A_i \cos(k_i(x_0 \cos \theta_i + z_0 \sin \theta_i) - \omega_i t + \phi_i) \\ z_{t+1} = z_t + \sum_{i=0}^N A_i \sin(k_i(x_0 \cos \theta_i + z_0 \sin \theta_i) - \omega_i t + \phi_i) \end{cases}$$

A chaque étape de l'animation, nous créons alors une nouvelle texture d'écume. Pour cela nous comparons la distance entre le point à la surface et les points de la texture d'écume au voisinage du pixel traité. Nous avons alors deux cas :

- si un point de la texture pour lequel cette distance est inférieure à un certain seuil (correspondant à  $10^{-1}$ ) est trouvé nous ajoutons dans la nouvelle texture les caractéristiques du texel trouvé ;
- si aucun texel n'est trouvé nous créons un nouveau point d'écume selon les critères du point à la surface.

Au final, la nouvelle texture correspond à la texture d'écume de l'étape courante.

La figure 2.6 montre le rendu de l'écume pour des vitesses de vent de 7 et 10km/h. La température de l'air  $T_a$  utilisée est de 20°C, celle de l'eau, de 12°C. Comme nous pouvons l'observer, la quantité d'écume générée dépend directement de la vitesse du vent et des caractéristiques de la surface, ce qui nous permet d'obtenir un résultat plus réaliste que la méthode présentée par [PA01].

## 2.3 Rendu des interactions surfaciques et sous-surfaciques

Comme nous l'avons vu dans le chapitre précédent, les échanges lumineux à la surface de l'océan sont caractérisés par de multiples réflexions et réfractions et représentent un phénomène complexe. Nous présentons dans cette section une nouvelle



FIG. 2.6 – Rendu de l'écume avec une vitesse de vent ( $U_{10}$ ) de 7 et 10km/h

méthode afin de rendre ces interactions surfaciques et sous-surfaciques. Nous nous attacherons à montrer comment ces effets peuvent être approximés et comment notre algorithme permet d'accélérer le calcul de la simulation de ces phénomènes.

### 2.3.1 Atténuation surfacique

Lorsqu'un rayon traverse la surface de l'eau, une partie de ce rayon est absorbée selon la loi de Beer-Lambert qui dépend directement de la concentration en particules en suspension dans l'eau. Ce principe est aussi valable durant la propagation du rayon dans l'air qui peut être plus moins chargé en embruns marins. Nous proposons ici une approche permettant de représenter ce phénomène en atténuant un rayon lumineux arrivant à la surface selon les conditions météorologiques de l'environnement de simulation.

Comme nous l'avons vu dans le chapitre précédent, l'intensité d'un faisceau lumineux d'intensité à la surface  $I(0)$  est atténué selon sa distance parcourue  $d$  et un coefficient d'atténuation  $a$  caractéristique du milieu dans lequel il se propage :

$$I' = I(0)e^{-ad} \quad (2.13)$$

Dans le cas océanique, cette loi peut être adaptée en utilisant l'épaisseur optique  $\tau_{aod}$  des embruns marins qui représente le coefficient d'atténuation volumique intégré verticalement dû à la présence de embruns marins :

$$\tau_{aod} = \int_0^{y_{max}} k_e(z) dz \quad (2.14)$$

avec  $y_{max}$  la hauteur maximale au-dessus de laquelle le phénomène devient négligeable, et  $k_e$  le coefficient d'extinction des embruns marins défini par [Aub02] :

$$k_e = \int_{r_{min}}^{r_{max}} \frac{dF(r)}{dr} 2\pi r^2 dr \quad (2.15)$$

avec  $\frac{dF(r)}{dr}$  la distribution en taille des particules en suspension dans l'air de rayon  $r$  qui peut être calculée par [SF79] :

$$\frac{dF(r)}{dr} = \frac{dF(r, 0)}{dr} e^{-y/y'} \quad (2.16)$$

avec  $y$  la hauteur du point courant,  $y'$  la hauteur de sa projection à la surface et  $dF(r, 0)$  la distribution en taille des particules de embruns marins à la surface de l'eau. Cette distribution est donnée par [MM86] :

$$\frac{dF(r, 0)}{dr} = 1.373 U_{10}^{3.41} r^{-3} (1 + 0.0057 r^{1.05}) 10^{1.19} e^{-B^2}$$

avec  $B = \frac{0.380 \log(r)}{0.65}$ . Nous modifions cette équation afin de tenir compte de la quantité d'écume  $s'$  du point à la surface :

$$\frac{dF(r, 0)'}{dr} = \frac{dF(r, 0)}{dr} s' \quad (2.17)$$

Durant le calcul du point d'intersection entre un rayon et la surface, à chaque pas, nous projetons le point sur le rayon à la surface. Nous calculons alors la quantité d'écume présente en ce point (équation 2.10) et nous calculons pour chaque sphère le coefficient d'extinction correspondant (2.15) en utilisant  $r_{min} = 10^{-4}$  et  $r_{max} = 10^{-3}$  et un pas d'intégration de  $2 \cdot 10^{-4}$  (voir figure 2.7). Nous approximons alors l'épaisseur optique  $\tau_{aod}$  des particules d'embruns marins en intégrant l'équation 2.14 le long du rayon :

$$\tau_{aod} = \sum_{i=1}^N k_{e_i} r_i \quad (2.18)$$

avec  $N$  le nombre de sphères le long du rayon,  $r_i$  le rayon de la sphère  $i$  et  $k_{e_i}$  son coefficient d'extinction.

La quantité de lumière totale  $L_0$  en un point de la surface est obtenue à l'aide de la formule suivante :

$$L(0) = (L_{sky} + L_{sun} \cos \theta_{sun}) e^{-\tau_{aod}} + L_{foam} \quad (2.19)$$

Afin de réduire les calculs, nous utilisons la cohérence spatiale afin de ne pas recalculer le coefficient d'extinction pour deux même sphères issues de rayons voisins.



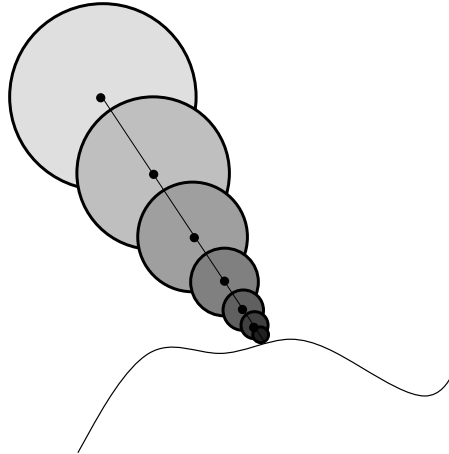


FIG. 2.7 – Ensemble des sphères le long d'un rayon affichées par niveaux de gris, correspondant à la progression du coefficient d'extinction (le plus clair désignant la plus faible valeur).

La figure 2.8 montre le rendu de l'atténuation surfacique pour des vitesses de vent  $U_{10}$  de 7 et 10  $km/h$ . Nous pouvons constater que cette méthode permet de simuler le comportement de la lumière dû à la présence des embruns marins en fonction de paramètres physique, ce qui se traduit globalement par des images qui s'assombrissent en fonction de la vitesse du vent et de la quantité d'écume.

### 2.3.2 Diffusion sous-surfacique

Comme nous l'avons mentionné dans le chapitre précédent, la diffusion sous-surfacique est un processus physique complexe. Les travaux proposés par [PA01] permettent d'approximer ces échanges en tenant compte des propriétés bio-optiques de l'eau de façon simple. Nous avons choisi d'étendre ces travaux, et en montrant comment notre méthode de visualisation permet d'accélérer le calcul de ce phénomène.

Selon [PA01], la luminance en point situé à la surface de coordonnées sphérique  $(0, \theta, \phi)$  peut être approximée par :

$$L(0, \theta, \phi) = L(y_{fond}, \theta, \phi)e^{-cR} + L_{df}(y) \quad (2.20)$$

avec  $R = \frac{y}{\cos\theta}$ ,  $L(y, \theta, \phi)$  la luminance au fond de l'océan et  $L_{df}(y)$  la luminance totale diffusée définie par :

$$L_{df}(y) = L_{df}(0)(1 - e^{(-c - K_a \cos\theta)R}) \quad (2.21)$$



FIG. 2.8 – Rendu de l'atténuation atmosphérique avec des vitesses de vent de 7 et 10km/h.

avec  $c$  le coefficient d'extinction de l'eau,  $K_d$  le coefficient de diffusion sortante dépendant du type d'eau (claire ou trouble) et  $L_{df}(0)$  représentant la radiance diffuse à la surface qui peut être approximée par :

$$L_{df}(0) = \frac{0.33b_b}{a} \left( \frac{E_d(0)}{\pi} \right) \quad (2.22)$$

avec  $a$  le coefficient d'atténuation qui peut être calculé en utilisant l'équation 1.19 vue au chapitre précédent,  $E_d(0)$  l'éclairement à la surface de l'eau selon les caractéristiques du ciel et de l'azimut du soleil et  $b_b$  défini par :

$$b_b(\lambda) = \frac{550}{\lambda} 0.3C^{0.62} (0.002 + 0.02(0.5 - 0.25 \log(C)))$$

Nous simplifions les équation 2.21 et 2.22 respectivement par :

$$L_{df}(y) = L_{df}(0)(1 - e^{(-c-K_d)y'}) \quad (2.23)$$

$$L_{df}(0) = \frac{0.33b_b}{a} L(0) \quad (2.24)$$

avec  $L(0)$  la luminance à la surface, prenant en compte les embruns marins en suspension dans l'air et l'écume, calculée à l'aide de la formule 2.19, et  $y'$  la différence de hauteur entre le point à la surface et le centre de la sphère.

Nous modélisons le fond de l'océan à l'aide d'un ensemble de composantes de la même façon que la surface. Nous appliquons l'approche décrite en section 2.1.1 afin de calculer l'intersection entre un rayon réfracté et le fond marin. Durant ce processus, nous calculons pour chaque sphère la quantité de lumière diffusée en utilisant les équations 2.23 et 2.24 (voir figure 2.9). La quantité totale de lumière diffusée le

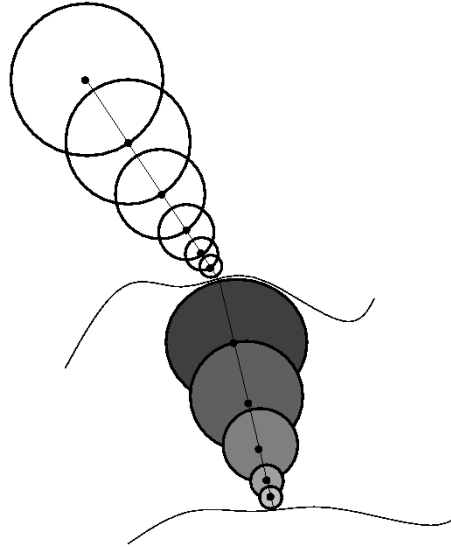


FIG. 2.9 – Ensemble des sphères le long d'un rayon réfracté affichées par niveaux de gris, correspondant à la progression de la quantité de lumière diffusée (le plus clair désignant la plus faible valeur)

long du rayon réfracté est alors obtenue par :

$$L_{df}(y) = \sum_{i=0}^N L_{df_i}(y)r_i \quad (2.25)$$

avec  $N$  le nombre de sphères présentes sur le rayon réfracté,  $r_i$  le rayon de la sphère  $i$  et  $L_{df_i}$  la quantité de lumière diffusée dans cette sphère définie par l'équation 2.24.

Nous effectuons ce calcul dans le domaine  $RGB$ , aux longueurs d'ondes 700, 546 et 435nm.

En utilisant l'équation 2.20, la lumière en un point de la surface est finalement calculée par :

$$L(0) = L_{fond}e^{-cL} + L_{df}(y) \quad (2.26)$$

avec  $L_{fond}$  la luminance du fond sous-marin et  $L$  la longueur du rayon réfracté.

De façon similaire à l'approche précédente, nous utilisons la cohérence spatiale afin d'éviter de recalculer la quantité de lumière diffusée dans chaque sphère pour deux rayons voisins. Ceci nous permet de réduire considérablement les calculs (voir section 2.5).

Les figures 2.10 et 2.11 montrent respectivement une comparaison de la diffusion sous-surfacique pour une eau trouble et une eau claire avec différentes concentrations

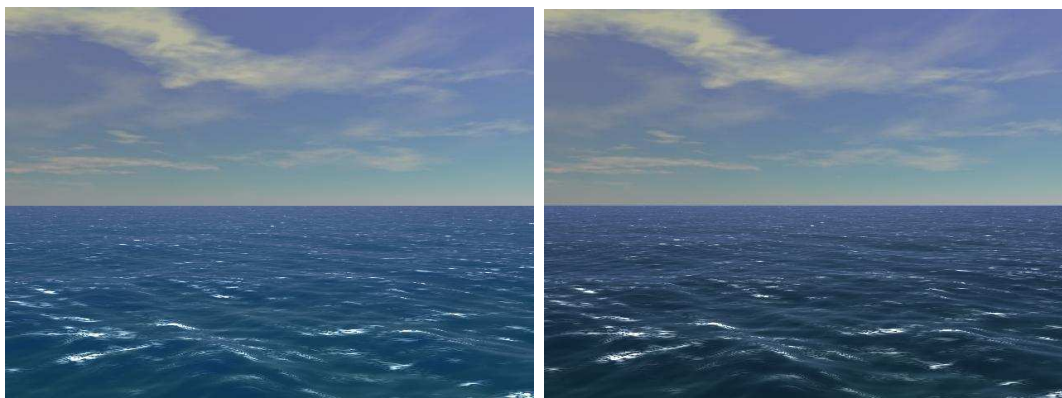


FIG. 2.10 – Rendu de la diffusion sous-surfacique avec une eau claire et trouble

en plancton. Comme nous pouvons le constater, cette approche permet de représenter ce phénomène de façon réaliste. En s'appuyant sur des paramètres physiques, il est alors possible de représenter différentes zones océaniques.

## 2.4 Eblouissement

La simulation de l'éblouissement en synthèse d'images a fait l'objet de nombreuses recherches. La première approche a été introduite par Shinya [SST89] en entourant les points les plus brillants d'une image d'un motif lumineux en forme d'étoile en post-traitement. Les auteurs de [NKON90] ont représenté l'éblouissement dû à la lumière directe de phares de voitures et à leurs réflexions sur une chaussée mouillée, en élaborant un modèle d'oeil simplifié, prenant en compte la diffraction de la lumière par la pupille et par les cils. L'approche proposée dans [SSZG95] est un modèle complet d'oeil humain qui permet de gérer différentes géométries d'éblouissement selon la nature de l'oeil considéré. Dans [Rok93], les auteurs ont également proposé également un algorithme se basant sur des observations scientifiques. Plus récemment, les auteurs de [MKH05] ont présenté une méthode de simulation de l'éblouissement, à l'aide d'une décomposition spectrale de la lumière, en proposant une fonction permettant de caractériser la distribution des points d'éblouissement selon le type de la pupille.

Dans sa thèse, Thon [Tho01] a proposé une méthode efficace limitant le coût de calcul et s'appliquant à des scènes océaniques. Le principe de cette méthode est de récupérer un buffer de réflexion spéculaire et de texturer les points les plus spéculaires à l'aide d'une texture d'éblouissement identique à celle perçue par un oeil humain (voir figure 2.12). Cependant, le problème de cette approche est double :

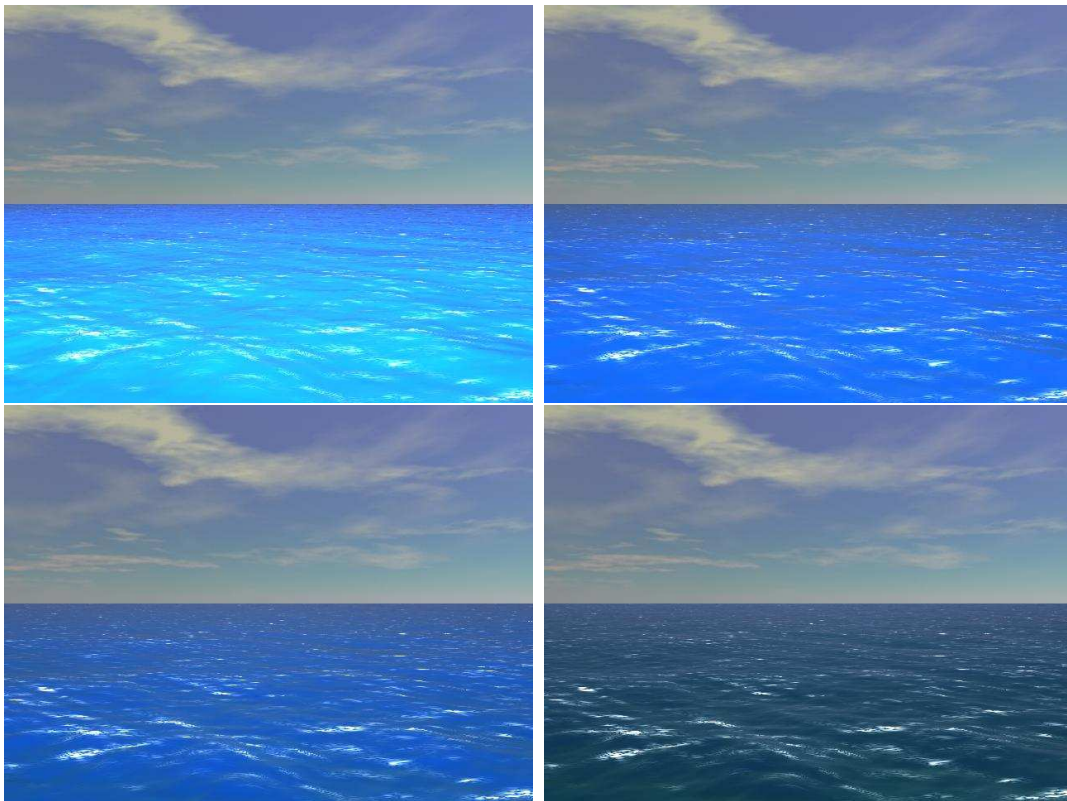


FIG. 2.11 – rendu de la diffusion sous-surfacique avec (de gauche à droite et de haut en bas) une concentration en chlorophylle  $C$  de 0.001, 0.01 0.05 et  $1.0\text{mg}/\text{m}^3$

d'une part, la surface de l'océan étant fortement spéculaire, une concentration de points d'éblouissement dans une même région peut former des zones d'éblouissement trop intenses, réduisant le réalisme des résultats. D'autre part un effet d'aliasage temporel peut aussi être observé : en considérant à chaque image de nouveaux points d'éblouissements, ceux-ci peuvent alors se mettre à clignoter. Cependant, cette méthode est très rapide car elle ne nécessite quasiment aucun calculs. Nous proposons ici d'étendre ces travaux.

Au départ de l'animation, nous définissons un seuil de spécularité permettant de sélectionner certains points de la surface générant un effet d'éblouissement. Afin de pallier au premier problème, nous forçons une distance minimale entre deux points d'éblouissement, pouvant être choisie par l'utilisateur. Nous stockons ces points dans une texture (voir la figure 2.12). Afin d'éviter des phénomènes d'aliasage temporel, nous mettons à jour cette texture durant l'animation. Pour cela, nous désactivons un point d'éblouissement si sa spécularité devient inférieure à un seuil donné et nous l'activons dans le cas contraire. A chaque instant, l'intensité d'un point d'éblouissement  $I_{ebloui}$  est calculée par :

$$I_{ebloui} = \frac{I_s - I_{min}}{I_{max} - I_{min}} \quad (2.27)$$

avec  $I_s$  l'intensité spéculaire du point,  $I_{min}$  le seuil de spécularité défini par l'utilisateur et  $I_{max}$  l'intensité spéculaire la plus élevée dans l'image traitée. Cette méthode nous permet alors de réduire considérablement l'aliasage temporel : en effet, comme l'intensité d'un point d'éblouissement est calculée en fonction de sa spécularité, aucun point ne sera désactivé lorsque son intensité sera maximale ; celle-ci diminuera progressivement jusqu'à ce qu'il soit désactivé.

La figure 2.13 montre le rendu de l'éblouissement avec notre approche, avec un seuil minimal de 0.95 et une distance minimale entre deux points d'éblouissement de 5.0.

Bien que cette méthode ne tienne pas compte des propriétés physiques du phénomène d'éblouissement, elle permet de capturer cet effet de façon réaliste, enrichissant le rendu des scènes synthétisées. Cependant, la gestion du seuil spéculaire peut s'avérer délicate : si un seuil trop petit est choisi par l'utilisateur, trop de points d'éblouissement peuvent apparaître, induisant une perte de réalisme. Dans le cas contraire, trop peu ou aucun point d'éblouissement ne sont générés. En revanche, cette approche est très rapide, et peut être facilement implémentée sur GPU pour une application en temps réel.

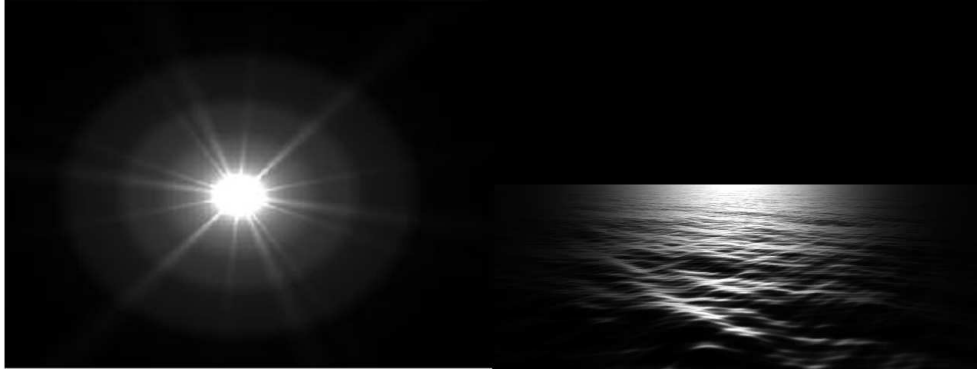


FIG. 2.12 – Texture d'éblouissement utilisée dans notre méthode (à gauche) et le buffer de réflexion spéculaire (à droite)



FIG. 2.13 – Rendu de l'éblouissement par notre méthode

## 2.5 Performances

Nous avons testé les performances de notre algorithme avec différentes configurations de la surface, c'est à dire avec 30, 40 et 50 composantes. Comme nous pouvons le voir sur la figure 2.1, notre algorithme permet d'obtenir un gain en temps de calcul d'environ 45% par rapport à une recherche incrémentale et 26% à un sphere-tracing non optimisé [HW96]. Nous avons remarqué que plus le nombre de primitives est élevé, plus notre approche converge rapidement. En effet, le temps moyen de la génération d'une image est environ de 245, 218 et 170 secondes avec respectivement 30, 40 et 50 composantes. Ceci s'explique par le fait que notre approche permet de réduire le nombre de composantes contribuant au calcul de l'intersection entre un rayon et la surface.

L'apport de la cohérence spatiale et temporelle permet aussi de réduire considérablement les temps de calcul car nous obtenons un gain respectif de 21% et 24% d'accélération en utilisant ces deux techniques (voir figure 2.2).

Comme nous pouvons le voir sur la figure 2.3, notre structure n'engendre pas un coût mémoire excessif car pour le rendu d'une image de 640x480 nous obtenons un coût mémoire de 18.20Mo, ce qui reste raisonnable.

La figure 2.4 montre la répartition des temps de calculs moyens par image pour une animation avec 30 composantes. L'étape la plus coûteuse reste celle de la recherche du point d'intersection qui représente environ 62.5% du temps de calcul total pour le rendu d'une image, soit environ 153 secondes. Le rendu de l'atténuation surfacique et de la diffusion sous-surfacique représentent respectivement 18.3% et 12.4% du temps de calcul total (soit 45 et 30 secondes), ce qui nous permet de noter que notre méthode n'engendre pas de calculs excessifs. Le rendu de l'éblouissement et de l'écume sont très rapides car ils représentent respectivement 3.7% et 3.1% (soit 9 et 7.5 secondes). Ces deux méthodes, basées textures, peuvent être implémentées sur carte graphique (GPU), ce qui permettrait d'obtenir la simulation de ces effets en temps-réel.

## 2.6 Conclusion

Nous avons présenté dans ce chapitre une nouvelle méthode de rendu unifiée de scènes océaniques en eaux profondes. Une nouvelle approche de visualisation de la surface a été proposée basée sur l'emploi du Sphère-Tracing [HW96], permettant



|    | Recherche incrémentale | Sphere-tracing non optimisé |
|----|------------------------|-----------------------------|
| 30 | 45%                    | 26%                         |
| 40 | 49%                    | 32%                         |
| 50 | 50%                    | 34%                         |

TAB. 2.1 – Gain en temps de calcul apporté par notre méthode comparé à un ray-marching et à un sphere-tracing non optimisé en fonction du nombre de composantes

|    | Cohérence spatiale | Cohérence temporelle |
|----|--------------------|----------------------|
| 30 | 21%                | 24%                  |
| 40 | 23%                | 30%                  |
| 50 | 27%                | 39%                  |

TAB. 2.2 – Apport de la cohérence spatiale et temporelle en fonction du nombre de composantes

|         | Nombre de sphères | Taille mémoire |
|---------|-------------------|----------------|
| 320x280 | 1187549           | 4.53Mo         |
| 480x360 | 2680632           | 10.22Mo        |
| 640x480 | 4772078           | 18.20Mo        |

TAB. 2.3 – Coût mémoire de notre structure de données pour différentes résolutions d'images et avec une surface représentée par 30 composantes

| Étape                                 | Pourcentage |
|---------------------------------------|-------------|
| recherche des points d'intersection   | 62.5%       |
| rendu de l'atténuation surfacique     | 18.3%       |
| rendu de la diffusion sous-surfacique | 12.4%       |
| rendu de l'éblouissement              | 3.7%        |
| rendu de l'écume                      | 3.1%        |

TAB. 2.4 – Répartition des calculs pour le rendu d'une image

d'obtenir une accélération considérable en temps de calcul vis à vis des méthodes existantes, en prenant en compte à la fois la cohérence spatiale et la cohérence temporelle au cours de l'animation. Les caractéristiques de la structure de données proposée permettent également la simulation de phénomènes divers tels que l'atténuation surfacique due à la présence de particules en suspension dans l'air et de la diffusion sous-surfacique. La représentation de l'écume et de l'éblouissement à l'aide de méthodes basées-texture nous a permis d'obtenir ces effets à moindre coût.

Cependant, notre algorithme ne permet pas une réelle représentation des embruns marins, ceux-ci étant représentés de manière implicite. Pour parvenir à ce but, il aurait fallu utiliser un système de particules, comme la plupart des approches existantes. Afin de limiter le coût mémoire, une méthode intuitive aurait été de ne les générer que si le point d'intersection était un "point d'écume" et en les rendant à l'aide d'un ensemble de *blobs* d'opacités différentes, en fonction de la quantité d'embruns présente dans la sphère à laquelle ils appartiennent. Le déplacement de ces particules aurait pu être contrôlé en fonction du "point d'écume" générateur et en rajoutant un bruit aléatoire afin de simuler le vent. C'est une piste que nous n'avons malheureusement pas eu le temps d'explorer mais qui peut permettre d'obtenir un résultat plus réaliste que celui présenté sur la figure 2.8.



## Chapitre 3

# Représentation de l'océan en eau peu profonde

La dynamique de la surface océanique en eaux peu profondes est plus complexe qu'en eaux profondes, où les modèles théoriques ne suffisent plus pour représenter le comportement de l'interface. Cette zone est le siège de vives interactions dont un processus révélateur est le déferlement. Nous proposons dans cette partie une nouvelle méthode basée physique, permettant de reproduire de manière physiquement réaliste le comportement des vagues déferlantes et des différents processus accompagnant ce phénomène, tels que les embruns marins. Ces travaux ont donné lieu à une communication nationale ayant reçu le prix du meilleur papier à la conférence AFIG07 [DCG07b] et est actuellement en cours de soumission dans une revue internationale [DCG08].

Dans une première partie, nous étudierons le modèle lagrangien SPH que nous avons utilisé afin de résoudre équations de Navier-Stokes. Puis nous introduirons le modèle *d'impulsion finie* que nous avons proposé afin de représenter les vagues déferlantes et les embruns marins générés lors du déferlement. Dans une troisième section, nous étudierons comment ce modèle lagrangien peut être approximé afin de réduire la complexité calculatoire et l'occupation mémoire. Enfin nous concluons en examinant les perspectives possibles qu'offre notre méthode.

### 3.1 Approche SPH

Nous présentons dans cette section l'approche SPH issue des travaux de [MCG03] que nous avons utilisé pour résoudre les équations de Navier-Stokes. Nous présentons ici les concepts de cette approche et détaillons les calculs afin de parvenir à la solution de ces équations.

### Résolution des équations de Navier-Stokes

Comme nous l'avons vu dans le chapitre 1, l'approche SPH est une méthode lagrangienne qui consiste à discrétiser le fluide à l'aide de particules possédant leurs propres masses, densités, pressions, vitesses et en résolvant les équations de Navier-Stokes en approximant les forces pour une particule donnée en fonction de celles exercées par ses voisines.

Dans ce type d'approche, les équations de Navier-Stokes sont simplifiées et linéarisées. En effet, en considérant chaque particule comme ayant sa propre masse, l'équation de conservation de la masse (équation 1.11) peut être omise : on est certain qu'au cours du temps la masse totale du fluide n'évoluera pas. De plus, dans la seconde équation (équation 1.12) le terme d'advection non linéaire, représentant le transport du fluide par ses vitesse dans le cas eulérien peut être lui aussi omis car il n'a pas ici de représentation. On aboutit donc à une seule équation de type linéaire exprimée par :

$$\frac{\partial \vec{U}}{\partial t} = \frac{1}{\rho} (-\nabla p + \mu \nabla^2 \vec{U} + \rho \vec{g}) \quad (3.1)$$

où  $\vec{U}$  désigne la vitesse du fluide,  $\rho$  sa densité,  $p$  sa pression,  $\mu$  sa viscosité et  $\vec{g}$  le vecteur du champ gravitationnel terrestre. Notons  $\vec{f} = -\nabla p + \mu \nabla^2 \vec{U} + \rho \vec{g}$  l'ensemble des forces agissant sur une particule. Le terme  $\nabla p$  représente les forces de pression,  $\mu \nabla^2 \vec{U}$  les forces de viscosité et  $\rho \vec{g}$  les forces extérieures. L'accélération d'une particule de fluide peut alors être calculée par :

$$a = \frac{\partial \vec{U}}{\partial t} = \frac{1}{\rho} (\vec{f}^{press} + \vec{f}^{visc} + \vec{f}^{ext}) \quad (3.2)$$

où  $\vec{f}^{press}$  (respectivement  $\vec{f}^{visc}$  et  $\vec{f}^{ext}$ ) représente les forces de pression (respectivement de viscosité et extérieures).

### Formalisme SPH

Afin de calculer les différents termes de l'équation 3.1, la méthode SPH consiste à évaluer la densité puis les forces de pression et de viscosité en approximant ces quantités selon le voisinage d'une particule. Dans cette approche, une particule ne représente pas une molécule de fluide, mais plutôt une estimation du fluide dans une région donnée. Formellement, toute quantité scalaire  $A$  peut être calculée par :

$$A_i = \sum_{j=1}^N m_j \frac{A_j}{\rho_j} W(r, h) \quad (3.3)$$

avec  $N$  le nombre de particules présentes dans le système,  $m_j$  (resp.  $A_j$ ) la masse (resp. le champ scalaire) de la particule  $j$ ,  $r$  la distance entre les deux particules  $i$

et  $j$ ,  $h$  la distance au-dessus de laquelle les interactions entre deux particules sont négligeables et  $W$  une fonction d'interpolation symétrique et normalisée :

$$\begin{cases} W(-r, h) = W(r, h) \\ \int W(r) dr = 1 \end{cases}$$

Le gradient et le laplacien de la quantité  $A$  peuvent être respectivement calculés par :

$$\nabla A_i = \sum_{j=1}^N m_j \frac{A_j}{\rho_j} \nabla W(r, h) \quad (3.4)$$

$$\nabla^2 A_i = \sum_{j=1}^N m_j \frac{A_j}{\rho_j} \nabla^2 W(r, h) \quad (3.5)$$

Les forces de pression et de viscosité peuvent être calculées à l'aide des équations 3.4 et 3.5. Cependant, ces formules ne permettent pas directement de conserver la symétrie des forces. Comme nous l'avons vu dans le chapitre 1, la méthode proposée par [MCG03] permet de remédier à ce problème tout en conservant la stabilité du système par l'emploi de fonctions d'interpolations appropriées. Nous avons utilisé cette méthode et nous détaillons ci-après les calculs.

**Calcul du champ de densité et de pression :** La densité  $\rho_i$  et la pression  $p_i$  d'une particule  $i$  peuvent être calculées par :

$$\rho_i = \sum_{j=1}^N m_j W_{poly6}(r, h) \quad (3.6)$$

$$p_i = k(\rho_i - \rho_0) \quad (3.7)$$

avec  $k$  étant une constante dépendant du fluide considéré et  $\rho_0$  sa densité au repos.

**Calcul des forces :** Les forces de pression et de viscosité exercée sur une particule  $i$  sont calculées respectivement par :

$$\vec{f}_i^{press} = - \sum_{j=1}^N m_j \frac{p_i + p_j}{2\rho_j} \nabla W_{spiky}(r, h) \quad (3.8)$$

$$\vec{f}_i^{visc} = \mu \sum_{j=1}^N m_j \frac{\vec{U}_j - \vec{U}_i}{\rho_j} \nabla^2 W_{visc}(r, h) \quad (3.9)$$

avec  $\mu$  la viscosité du fluide considérée,  $\vec{U}_i$  (resp.  $\vec{U}_j$ ) la vitesse de la particule  $i$  (resp.  $j$ ) et  $p_i$  la pression de la particule  $i$ . Ces forces sont évaluées si les deux particules  $i$

et  $j$  sont différentes.

La force de tension de surface exercée sur une particule  $i$  est calculée par :

$$\vec{f}_i^s = -\sigma \nabla^2 c_s \frac{\vec{n}}{|\vec{n}|} \quad (3.10)$$

avec :

$$c_s = \sum_{j=1}^N \frac{m_j}{\rho_j} W_{poly6}(r, h)$$

et  $\vec{n} = \nabla c_s$ . Le gradient et le laplacien de la fonction  $c_s$  sont calculés en utilisant les équation 3.4 et 3.5. Cette force est évaluée si  $|\vec{n}|$  est supérieure à un certain seuil  $l$ .

Les fonctions d'interpolation  $W_{poly6}$ ,  $W_{spiky}$  et  $W_{visc}$  sont définies dans l'annexe B.

**Algorithme général :** Le processus de simulation consiste alors à calculer la densité et la pression de chaque particule à l'aide des équations 3.6 et 3.7, puis les forces internes par les équations 3.8 et 3.9, et externes telles que la tension de surface (équation 3.10) et la gravité. Nous utilisons la méthode d'intégration Euler explicite afin de calculer la nouvelle position de chaque particule. Cette mise à jour doit s'effectuer après l'ensemble des calculs des forces car leurs expressions dépendent des caractéristiques des particules à l'instant de simulation et non à l'instant suivant. L'ensemble de ces étapes est décrit dans l'algorithme 1.

Nous avons utilisé les paramètres présentés dans [Kel06]. La distance maximale d'interaction  $h_i$  d'une particule  $i$  est calculée par :

$$h_i = \sqrt{\frac{3m_i x}{4\pi\rho_0}} \quad (3.11)$$

avec  $x$  le taux maximal de particules pouvant interagir avec une particule quelconque et dépendant du fluide considéré. Pour l'eau, les différents paramètres sont :  $m = 0.02$ ,  $x = 20$ ,  $\rho_0 = 998.29$ ,  $k = 3.0$ ,  $\mu = 3.5$ ,  $\sigma = 0.0728$  et  $l = 7.065$ . Pour un liquide visqueux, comme le mucus, ces paramètres sont :  $m = 0.04$ ,  $x = 40$ ,  $\rho_0 = 1000$ ,  $x = 40$ ,  $k = 5.0$ ,  $\mu = 40.0$ ,  $\sigma = 6.0$ ,  $l = 5.0$ . La figure 3.1 présente une capture des simulations que nous avons obtenu avec un pas de temps  $\Delta t = 0.001$ .

## 3.2 Simulation de vagues déferlantes

Le déferlement d'une vague est un phénomène complexe qui a été longtemps étudié. Pour une vague plongeante, il peut être résumé en plusieurs phases. Dans un

---

**Algorithme 1** Algorithme de simulation sph

---

**Algorithme SPH****Variables**

|  $m, x, \rho_0, \mu, k, \sigma, l, \Delta t$  : flottants

**Début**

**Pour**  $i$  **variantDe** 1 à  $N$  **Faire**

| Réinitialisation de  $\rho_i$ ,  $p_i$  et  $F_i$

| **Pour**  $j$  **variantDe** 1 à  $N$  **Faire**

| |  $\rho_i \leftarrow \rho_i + \text{Densite}(i, j)$

| **FinPour**

|  $p_i \leftarrow \text{Pression}(i)$

| **FinPour**

**Pour**  $i$  **variantDe** 1 à  $N$  **Faire**

| **Pour**  $j$  **variantDe** 1 à  $N$  **Faire**

| | **Si**  $i \neq j$  **Alors**

| | |  $\vec{f}_i \leftarrow \vec{f}_i + \text{ForcePression}(i, j)$

| | |  $\vec{f}_i \leftarrow \vec{f}_i + \text{ForceViscosité}(i, j)$

| | **FinSi**

| |  $\vec{f}_i \leftarrow \vec{f}_i + \text{TensionSurface}(i, j)$

| **FinPour**

|  $\vec{f}_i \leftarrow \vec{f}_i + \rho_i G$

| **FinPour**

**Pour**  $i$  **variantDe** 1 à  $N$  **Faire**

|  $\vec{U}_{i_{t+1}} \leftarrow \vec{U}_{i_t} + \frac{\vec{f}_i}{\rho_i} \Delta t$

|  $\vec{X}_{i_{t+1}} \leftarrow \vec{X}_{i_t} + \vec{U}_{i_t} \Delta t$

| **FinPour**

**Fin**

---





FIG. 3.1 – Simulation SPH d'un liquide visqueux (à gauche) et de l'eau (à droite)

premier temps, on observe une levée et une propagation d'une onde solitaire dont la hauteur augmente au fur et à mesure. Arrivée à une certaine hauteur, la vague plonge, ce qui génère de nombreux embruns marins et de l'écume à la surface. Enfin la surface de l'océan se stabilise.

Afin de simuler les vagues déferlantes, nous avons vu que les méthodes existantes reposent sur un conditionnement du fluide à l'aide de modèles paramétriques tels que celui proposé par [TDG00]. Le problème de ces méthodes réside dans la limitation temporelle : pour reproduire un nouveau déferlement, un conditionnement des vitesses de chaque composante doit être alors effectué. Ceci peut induire une divergence numérique du système global, obligeant à un utilisateur à procéder par essais/erreurs afin de maintenir la stabilité du système. Pour pallier à cette limitation, nous proposons ici une nouvelle approche en représentant le déferlement sous la forme d'une force extérieure. Cette force, que nous avons nommé la *force d'impulsion finie* permet à un utilisateur de contrôler la géométrie de la vague recherchée mais aussi le type de déferlement (plongeant, glissant ou frontal). Une mise à jour de cette force est alors effectuée afin de permettre la naissance d'une nouvelle vague déferlante et d'obtenir un résultat non limité dans le temps de ce phénomène. De plus, comme nous le verrons, on peut combiner ce modèle à l'aide d'une superposition linéaire afin de représenter plusieurs vagues.

### 3.2.1 Méthode d'impulsion finie

Notre modèle s'appuie sur une extension d'un modèle physique 2D océanographique que nous avons représenté sous la forme d'une force extérieure évaluée dans l'équation 3.2. Nous détaillons ci-après l'extension du modèle 2D au modèle 3D avec l'ajout de différents paramètres permettant de contrôler la géométrie de la vague

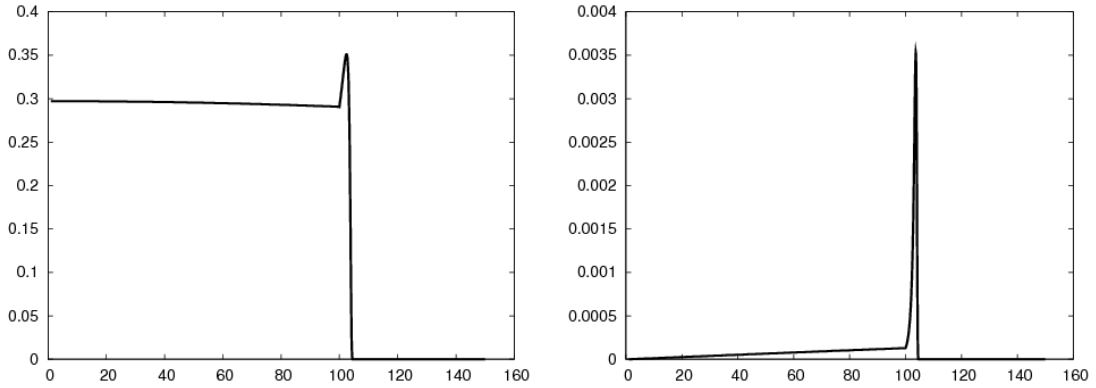


FIG. 3.2 – Magnitude de la vitesse horizontale (à gauche) et verticale (à droite) en fonction du temps

et le processus de réinitialisation afin de représenter plusieurs vagues déferlantes successives.

### Modèle 2D

Le modèle du soliton [RO98] est un modèle physique qui consiste à décrire une vague déferlante comme une onde solitaire. Dans ce modèle, la vitesse initiale en 2D d'un élément de fluide de coordonnées  $(x, y)$  s'exprime par :

$$\begin{cases} u = \sqrt{gd} \frac{H}{d} \operatorname{sech}^2(B(x - c\bar{t})) \\ v = \sqrt{3gd} \left(\frac{H}{d}\right)^{3/2} \frac{y}{d} \operatorname{sech}^2(B(x - c\bar{t})) \tanh(B(x - c\bar{t})) \end{cases}$$

avec  $\operatorname{sech} = 1/\cosh$ ,  $H$  la hauteur du soliton,  $c$  sa célérité,  $d$  la profondeur de l'élément de fluide par rapport au fond,  $B = \sqrt{\frac{3H}{4d^3}}$  et  $\bar{t}$  étant son temps fictif, c'est-à-dire le temps propre du soliton indépendant du temps de la simulation.

Ce modèle permet de prendre en compte les variations de profondeur et de simuler une vague déferlante. En effet, la diminution de la profondeur  $d$  provoque une augmentation puis une diminution brutale des vitesses horizontale et verticale d'un élément de fluide, ce qui permet de faire déferler la vague. La figure 3.2 illustre ce principe pour un soliton de paramètres  $H = 0.3$  et  $c = 0.1$  dont les vitesses horizontale et verticale sont calculées pour un élément de coordonnées 2D  $(0.1, 0.1)$  avec une profondeur initiale  $d = 10$  qui diminue de 0.1 à chaque pas de temps à partir de  $t = 100$ .

Le principe de notre méthode consiste à appliquer ce modèle sous la forme d'une force extérieure notée nommée *force d'impulsion* et notée  $\vec{f}^{\text{soliton}}$  sur chaque parti-

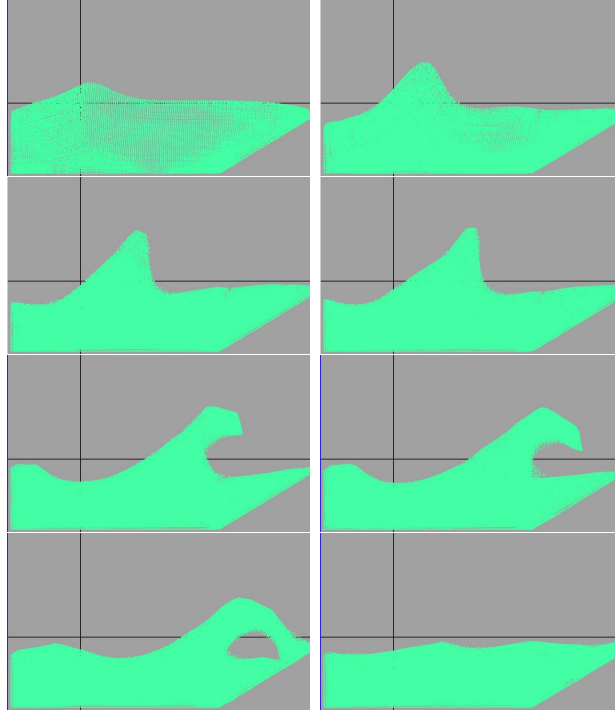


FIG. 3.3 – Les différentes phases du déferlement d'une vague en 2D calculé avec notre méthode

culé SPH. Les différentes phases d'une vague déferlante obtenue avec cette méthode sont illustrées sur la figure 3.3. Dans un premier temps, du fait que cette force augmente, le fluide se lève en créant une vague solitaire qui se propage au cours du temps. Quand la profondeur diminue, cette force décroît rapidement, provoquant le plongement de la vague générée. Comme la magnitude de la force d'impulsion est proche de 0, le fluide n'est plus soumis qu'à la gravité et finit par se stabiliser.

### Modèle 3D

Nous proposons d'étendre ce modèle au cas 3D en incluant différents paramètres tels qu'un paramètre de déphasage  $\phi$ , permettant d'avoir une solution non centrée sur 0, ainsi qu'un angle  $\theta$  de déviation par rapport à l'axe de déferlement et le plan perpendiculaire à celui-ci, permettant d'obtenir une asymétrie de la vague déferlante. Les composantes de la force d'impulsion appliquée à chaque particule du système sont :

$$\begin{cases} u' = \sqrt{gd\frac{H}{d}} \operatorname{sech}^2(B(x - c\bar{t} + \phi)) \\ v' = \sqrt{3gd}\left(\frac{H}{d}\right)^{3/2} \frac{y}{d} \operatorname{sech}^2(B(x - c\bar{t} + \phi)) \tanh(B(x - c\bar{t} + \phi)) \\ w' = \sqrt{gd\frac{H}{d}} \operatorname{sech}^2(B(z \cos \theta - c\bar{t} + \phi)) \end{cases} \quad (3.12)$$

Les figures 3.4 et 3.5 montrent respectivement les différentes phases en 3D du

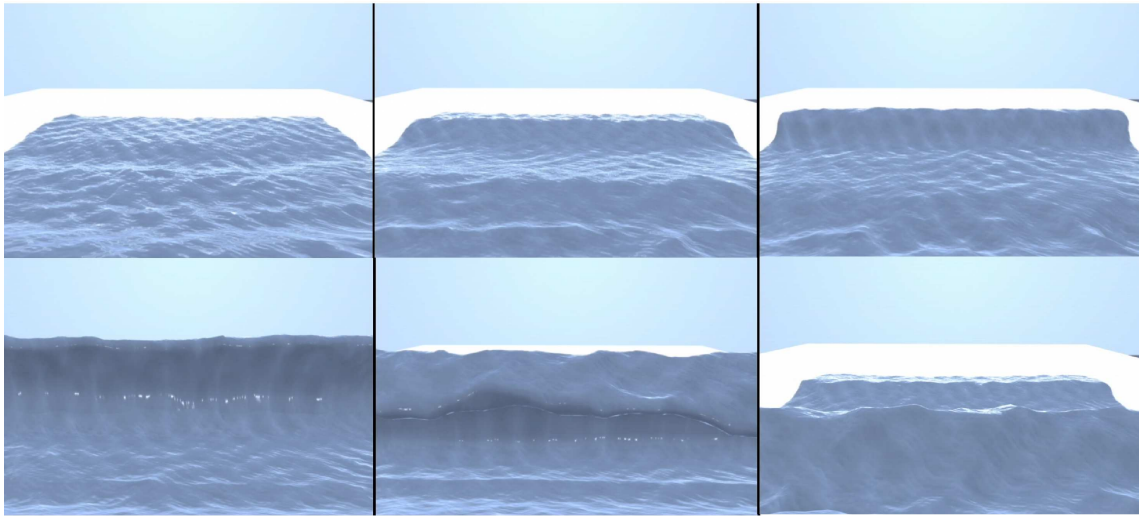


FIG. 3.4 – Les différentes phases du déferlement d’une vague en 3D calculées avec notre méthode

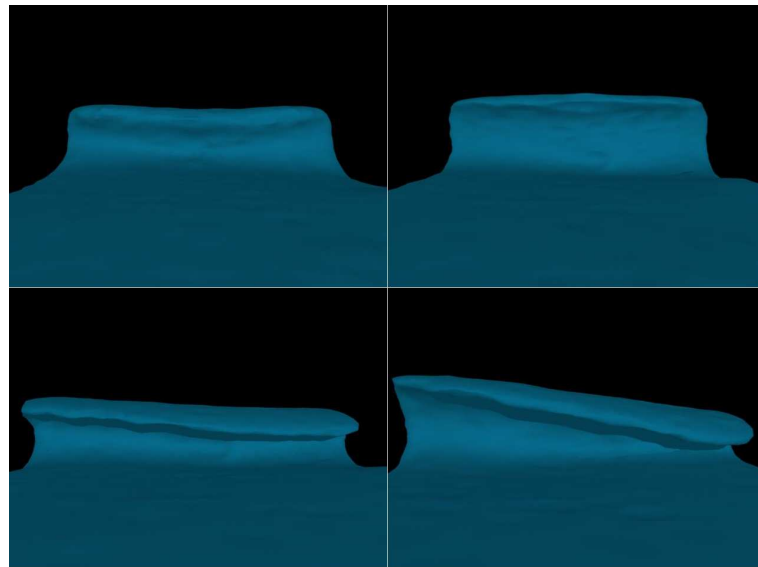


FIG. 3.5 – En haut, variation de la hauteur avec  $H = 0.2$  et  $H = 0.35$ . En bas, variation de l’angle de déviation avec  $\theta = 0$  et  $\theta = 0.5$  (en radians)

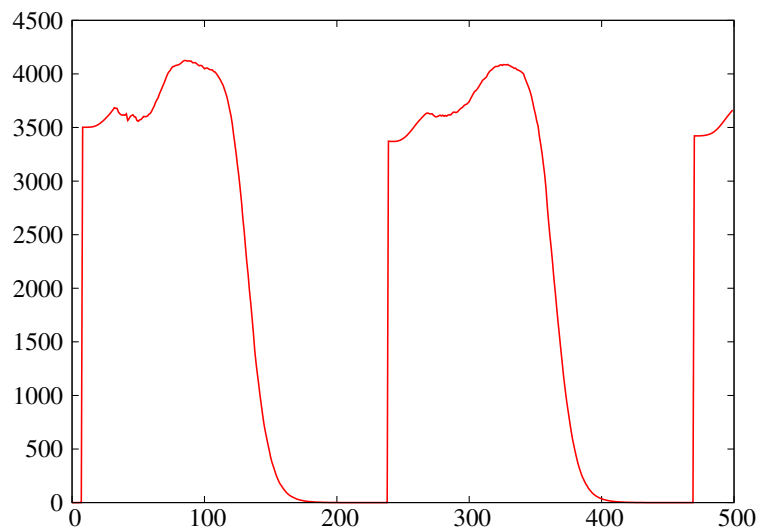


FIG. 3.6 – Moyenne de la magnitude des forces d’impulsions du système sur une animation de 500 images

déferlement avec notre méthode et l’influence de la variation de la hauteur et de l’angle de déviation.

### Processus de réinitialisation

Afin d’éviter une limitation temporelle, le temps fictif  $\bar{t}$  est réinitialisé quand la moyenne des forces d’impulsions de l’ensemble des particules est proche de 0. En effet dans ce cas le fluide n’a plus assez de force pour déferler. La ré-initialisation nous permet alors de relancer le système et de procéder ainsi à un nouveau déferlement. Cet effet est illustré dans la figure 3.6 : une impulsion est donnée au temps  $t = 10$  de la simulation, la force d’impulsion des particules augmentant pour finalement décroître rapidement. Quand la moyenne des forces d’impulsion est proche de 0, le temps fictif  $\bar{t}$  du soliton est alors réinitialisé et elle recommence à augmenter. Cette force représente en moyenne 10% de l’ensemble des forces auxquelles est soumise une particule.

### 3.2.2 Déferlement plongeant et déferlement glissant

Comme nous l’avons vu dans le chapitre 1, il existe différents types de vagues déferlantes (glissante, plongeante et frontale) selon la configuration du fond marin. Cette diversité peut être représentée par ce modèle en utilisant des paramètres adéquats. En effet, en variant seulement la célérité  $c$  du soliton, on peut voir que le profil vertical de la vitesse d’un élément 2D s’accroît si  $c$  diminue (voir figure 3.7). Sous forme de forces, cette propriété permet de représenter différents types de vagues dé-

ferlantes : plus la vitesse verticale des particules diminue brusquement, plus la vague g n r e plonge et r ciproquement. En d'autres termes, plus la c l rit  d'un soliton est importante, moins la force verticale d'impulsion associ e   ce soliton d cro t rapidement, ne permettant pas   l'ensemble des particules de plonger et r ciproquement.

Afin de calculer la c l rit  d'un soliton en fonction du type de d ferlement souhait , nous avons propos  d'employer le nombre d'Iribarren  $I$  d fini dans l' quation 1.1 et qui permet de pr dire le type de la vague d ferlante en fonction de la pente de la plage et de la hauteur de la vague. La c l rit   $c$  d'un soliton peut  tre calcul e par :

$$c = \sqrt{\frac{g\lambda}{2\pi}}$$

avec  $\lambda$   tant la longueur d'onde du soliton. Or, comme nous l'avons vu le nombre d'Iribarren  $I$  s'exprime par :

$$I = \frac{\tan \alpha}{\sqrt{H/\lambda}}$$

Nous calculons donc la c l rit  d'un soliton par :

$$c = \frac{I}{\tan \alpha} \sqrt{\frac{gH}{2\pi}} \quad (3.13)$$

La figure 3.8 montre deux captures d'une vague plongeante et glissante prises au m me temps de simulation (au temps  $t = 100$  avec une lev e commen ant au temps  $t = 10$ ). Le nombre d'Iribarren utilis  pour chacun de ces solitons est respectivement 0.5 et 0.3 avec une m me pente de plage  $\alpha = 0.14$  et une hauteur  $H = 0.3$

Bien que ce param tre soit relativement simple, omettant de nombreuses variables affectant la forme de la vague telles que vent par exemple, il permet n anmoins de contr ler le type de vague d ferlante souhait  par un utilisateur.

Cependant, il ne permet pas directement de reproduire des vagues frontales. Pour repr senter ce ph nom ne, il faudrait que la force d'impulsion soit quasi-constante afin de permettre de maintenir le volume d'eau presque   la verticale et puis fortement d croissante afin de lui permettre de s' craser litt ralement. Or, la force d'impulsion associ e   un soliton d cro t toujours en fonction du temps propre du soliton (voir figure 3.7). Une solution pour reproduire ce ph nom ne serait d'utiliser une force constante (en dehors des  quations du soliton et ne prenant pas en compte la variable temporelle) et d'arr ter celle-ci   partir d'une certaine limite. Ceci aurait pour effet de reproduire la prise de hauteur de la vague (comme dans le cas plongeant) et de proc der   son effondrement pour simuler son caract re frontal.

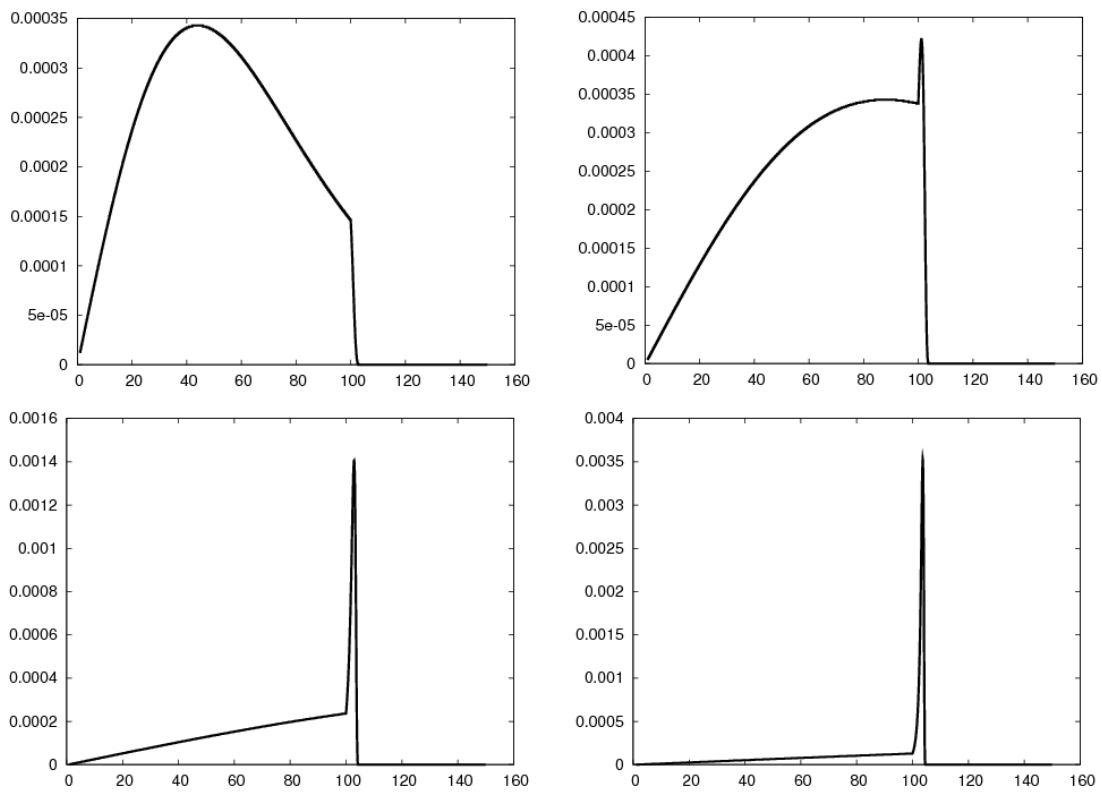


FIG. 3.7 – Profil de la vitesse verticale en fonction du temps (de gauche à droite et de haut en bas) pour  $c = 1.0$ ,  $c = 0.5$ ,  $c = 0.2$  et  $c = 0.1$

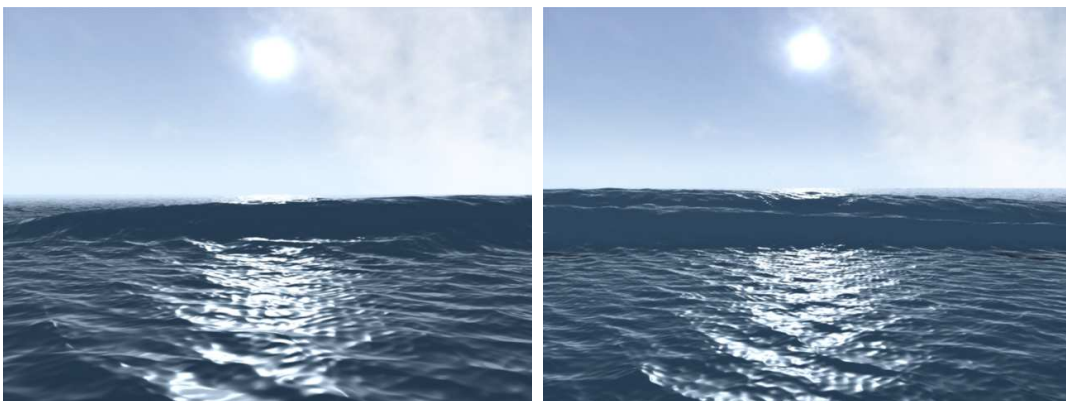


FIG. 3.8 – Une vague glissante (à gauche) et plongeante (à droite) au même instant de simulation

### 3.2.3 Combinaison de train de vagues

Nous avons également étendu ce modèle afin de simuler plusieurs vagues déferlantes en décrivant la force d'impulsion appliquée à une particule sous la forme d'une superposition linéaire de plusieurs solitons :

$$\vec{f}^{imp} = \sum_i^N \vec{f}_i^{soliton} \quad (3.14)$$

où  $N$  est le nombre de solitons présents dans la scène et  $\vec{f}_i^{soliton}$  désigne la force d'impulsion associée à un soliton  $i$ .

Un problème peut avoir lieu lorsque deux phases de solitons sont trop proches : la magnitude de la force d'impulsion résultante agissant sur les particules peut alors être trop importante, ce qui conduit à une très forte levée de fluide, induisant un résultat non réaliste. Pour résoudre ce problème, nous avons choisi de moyenner la force d'impulsion résultante.

Pour reproduire le déferlement de plusieurs vagues, nous créons alors plusieurs solitons en réinitialisant chaque soliton si la moyenne des forces d'impulsions qui lui est associée est proche de 0, comme cela est expliqué dans la section précédente. Afin d'obtenir un résultat non uniforme temporellement, nous générons un quadruplet aléatoire  $(H_i, I_i, \phi_i, \theta_i)$  à chaque ré-initialisation.

La figure 3.9 montre un aperçu de la simulation avec deux vagues déferlantes de paramètres respectifs  $(0.3, 0.3, 0.05, 0.5)$  et  $(0.3, 0.3, 0.15, 0.3)$ . Comme nous pouvons le voir, cette approche permet de combiner plusieurs trains de vagues déferlantes dont le nombre peut être fixé par l'utilisateur.

### 3.2.4 Simulation des embruns marins

Nous présentons ici notre méthode de génération automatique des embruns marins issus du déferlement en générant un sous-système de particules contrôlé dynamiquement par les particules de simulation afin de réduire les temps de calculs nécessaires à la représentation de ce phénomène au cas des scènes océaniques. Notre processus de génération s'appuie sur notre modèle physique de déferlement afin de calculer le nombre d'embruns marins à chaque instant de la simulation.

En effet, comme nous l'avons déjà vu, le moment où une particule déferle se traduit par une augmentation puis une décroissance de sa hauteur mais aussi de sa



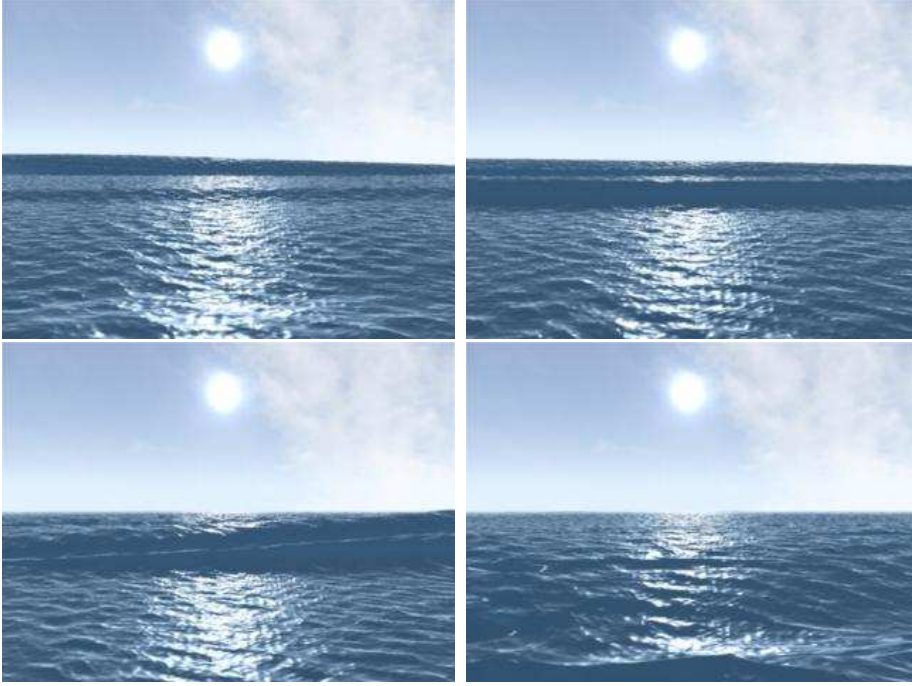


FIG. 3.9 – Combinaison de deux vagues déferlantes

force d'impulsion. En partant de ce constat, nous identifions une particule comme génératrice d'embruns si sa hauteur et la magnitude de la force d'impulsion à laquelle elle est soumise croissent. Nous marquons ces particules que nous nommons *particules de déferlement*. Le nombre  $N$  de particules générées est alors calculé par la formule suivante :

$$N_{embrunsmarins} = n_{embrunsmarins} \left( \frac{\Delta H_{t+1}}{H_{t+1}} + \frac{\Delta f_{t+1}^{imp}}{f_{t+1}^{imp}} \right) \quad (3.15)$$

avec  $\Delta H_{t+1} = H_{t+1} - H_t$  et  $\Delta f_{t+1}^{imp} = |f_{t+1}^{imp}| - |f_t^{imp}|$  avec  $H_t$  (resp.  $H_{t+1}$ ) correspondant à la hauteur de la particule à l'instant  $t$  (resp.  $t + 1$ ),  $f_t^{imp}$  (resp.  $f_{t+1}^{imp}$ ) correspondant à la magnitude de la force d'impulsion à l'instant  $t$  (resp.  $t + 1$ ), et  $n_{embrunsmarins}$  contrôlant la quantité d'embruns marins générés. Par expérimentation, nous avons choisi d'utiliser  $n = 100$

A chaque pas de la simulation, nous identifions si une particule est une particule de déferlement. Si c'est le cas, nous calculons alors le nombre d'embruns marins qu'elle génère et nous les déplaçons en fonction du mouvement de la particule génératrice. Afin d'obtenir un résultat non uniforme, nous utilisons également un déplacement aléatoire simulant le vent. La position de chaque particule du sous-système est alors calculée par :

$$\vec{X}_{t+1} = \vec{X}_t + (\vec{U}_t + \vec{V}_t) \Delta t \quad (3.16)$$

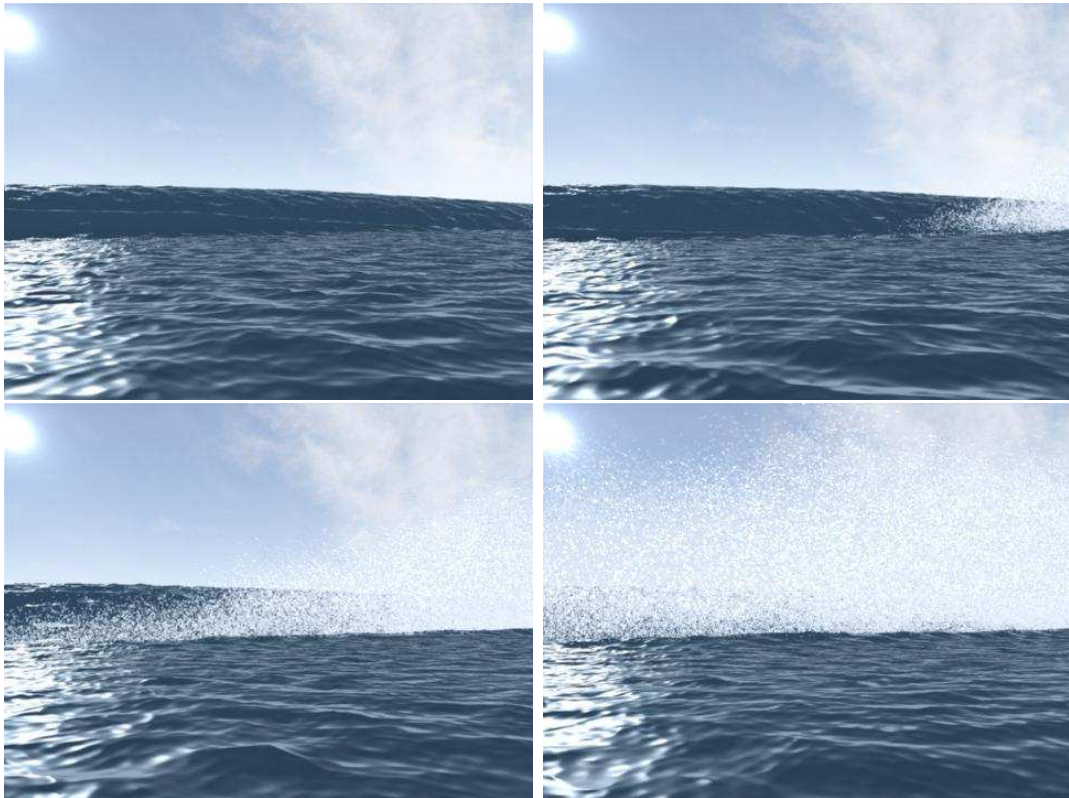


FIG. 3.10 – Simulation des embruns marins dû au déferlement

avec  $\vec{U}_t$  étant le vecteur vitesse de la particule génératrice et  $\vec{V}_t$  un vecteur aléatoire dont les coordonnées sont comprises entre 0 et 1. Comme nous pouvons le noter aucune interaction entre les particules d’embruns marins n’est calculée : celles-ci sont conduites dynamiquement par leur particule génératrice et un vecteur aléatoire.

Chaque particule d’embruns possède les attributs d’un système de particules classique [Ree83], tels qu’une durée de vie, mais aussi un rayon et une opacité qui sont calculés de manière aléatoire et variant respectivement entre 0 et 1 et entre  $10^{-3}$  (valeur limite au-dessus de laquelle l’embrun apparaît trop grossier) et  $10^{-4}$  (valeur limite en dessous de laquelle la particule n’est plus affichée). Si nous analysons le graphique 3.6, nous pouvons voir qu’il faut au minimum environ 200 images afin d’obtenir un nouveau déferlement. Nous fixons cette durée de vie à chaque particule d’embruns que nous décrétons à chaque instant. Finalement, ces particules sont rendues sous la forme d’un nuage de points, traité lors du rendu avec le logiciel *Autodesk Maya* (voir section 3.4).

En s’appuyant sur le modèle de déferlement proposé, cette approche permet une génération automatique des embruns marins lors du déferlement et d’enrichir le rendu

(voir la figure 3.10).

### 3.2.5 Discussion

La méthode présentée permet de simuler des vagues déferlantes tout en évitant une limitation temporelle et en permettant un contrôle des caractéristiques de chaque train de vagues déferlantes. L'utilisateur peut alors définir la hauteur, la phase d'une vague, son angle de déviation, tout en contrôlant le nombre de trains de vagues. Cette approche permet aussi de représenter différents types de vagues déferlantes, tel que plongeante ou glissante. Le cas du déferlement frontal peut être obtenu en combinant un déferlement plongeant et un arrêt ponctuel de la force. Elle permet également la génération automatique des embruns marins issus du déferlement à partir du modèle physique de représentation de ce phénomène.

Malheureusement, afin de simuler une très vaste étendue de fluide, elle nécessite d'utiliser un très grand nombre de particules, ce qui peut se révéler très coûteux. Afin de remédier à cette limitation, nous proposons dans la section suivante une approche multirésolution permettant de réduire le nombre de particules nécessaires.

## 3.3 Approche SPH multirésolution

Comme nous l'avons vu dans le chapitre 1, la multirésolution lagrangienne apporte une solution au problème du contrôle du coût en mémoire et en temps de calcul en permettant une représentation hiérarchique du système global à travers deux opérations élémentaires appelées "fusion" et "division" qui consistent respectivement à rattacher ou détacher des particules au sein d'une même particule. Cependant, les approches existantes [APKG07] nécessitent des calculs coûteux de champs de distance vis à vis de la surface ainsi qu'un redistancement des particules. Nous proposons donc une simplification spécifique au déferlement des vagues afin de limiter les calculs.

Notre schéma adaptatif s'effectue en différentes étapes. D'une part, les particules devant être fusionnées ou divisées sont identifiées au moyen d'heuristiques simples, ne nécessitant quasiment aucun calcul. Ensuite, afin de conserver la masse totale du système, la masse des particules filles est transmise progressivement vers la mère (voir figure 3.11). Enfin, afin de garantir la symétrie des forces entre des particules de tailles différentes, nous employons la méthode de *shooting-gathering* de [DC99] étendue par [APKG07].

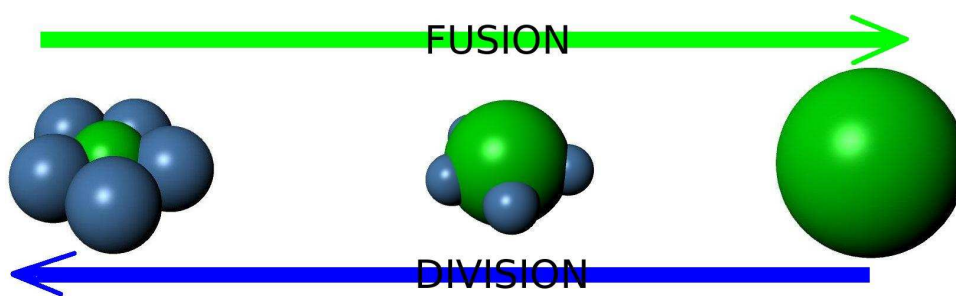


FIG. 3.11 – Transfert de masse progressif pendant les opérations de fusion et de division entre une particule mère (au centre en vert) et ses filles (en bleues).

### 3.3.1 Identification des particules

Les particules du système pouvant être affectées par une opération de fusion ou de division sont identifiées en tenant compte de différents critères tels que la profondeur de la particule, sa force d'impulsion et sa distance par rapport à l'observateur. En effet, nous supposons que plus une particule est profonde, plus nous pouvons approximer le calcul des forces car, notamment, les forces de tensions de surface exercées sur celles-ci seront minimales ou quasi-nulles. En contrepartie, plus la force d'impulsion de déferlement à laquelle est soumise une particule est importante, moins nous pouvons approximer ses échanges. Enfin, comme chaque particule possède un impact visuel sur le rendu final de la simulation, plus une particule se trouve éloignée de l'observateur, plus nous pouvons approximer son comportement dynamique.

Au départ de la simulation, les particules de fluide sont initialisées à l'aide d'une boîte englobante de hauteur  $H_{max}$  et  $H_{min}$ , paramètres globaux fixés au départ de la simulation, que l'utilisateur peut contrôler. Notons  $H_i$  la profondeur d'une particule  $i$  par rapport à la boîte englobante. Notre critère de profondeur est alors :

$$C_{H_i} = \left| \frac{H_i - H_{min}}{H_{max} - H_{min}} \right| \quad (3.17)$$

Les critères de distance à la caméra  $C_{D_i}$  et de force d'impulsion  $C_{\vec{f}_i^{imp}}$  sont définis de façon similaire (resp. à partir de la distance  $D_i$  et de la force d'impulsion verticale  $\vec{f}_i^{imp}$  à laquelle est soumise la particule). Une différence notable est que la force d'impulsion verticale maximale n'est pas un seuil défini arbitrairement mais le maximum de la somme des forces d'impulsions pouvant s'appliquer sur une particule.

Notre critère de choix est simplement défini par :

$$C_i = C_{H_i} + C_{D_i} + C_{\vec{f}_i^{imp}} \quad (3.18)$$

On considérera qu'une particule  $i$  peut "attacher" d'autres particules si  $C_i > 1$ . Si c'est le cas, les particules qui lui sont attachées sont celles se trouvant à une distance inférieure à  $h_i$  et qui respectent elles aussi le critère de choix. Par contre, une particule ne pourra plus représenter dynamiquement ses filles si elle a subi une opération de fusion et si son critère de choix  $C_i \leq 1$ . Dans ce cas là, elle sera divisée.

### 3.3.2 Transfert de masse progressif

Après cette identification, la seconde étape du processus consiste à transférer une partie de la masse des particules filles vers la particule mère. Dans le cas de la fusion, la masse perdue par une particule est toujours gagnée par une autre et réciproquement durant le processus de division.

Durant l'opération de fusion, la masse de chaque particule mère est augmentée progressivement alors que celle de ses filles est réduite : à chaque pas de temps, chaque particule fille perd une faible quantité de masse rétribuée à sa mère. Ce processus s'arrête dès que la particule mère a atteint sa masse maximale, calculée comme la somme de sa masse initiale et des masses initiales des particules filles. Une fois atteint ce critère, nous **désactivons** les particules filles au sein du système : elles n'interviendront notamment plus dans les calculs de voisinage représentant l'essentiel du coût de la simulation. Durant l'opération de division, nous transférons une faible quantité de masse de la particule vers chacune de ses filles, et nous arrêtons le processus lorsque la particule mère est revenue à sa masse initiale.

Nous choisissons de transférer 10% de la masse initiale, ce qui nous garantit de limiter les artefacts visuels pendant le processus de fusion ou de division. Notre méthode de transfert linéaire de masse entre particules filles et mères nous garantit le respect de la conservation de la masse durant le processus et ainsi la stabilité du système global. De plus, l'utilisateur peut contrôler à travers le nombre initial de particules la mémoire maximale nécessaire pour la simulation, puisqu'aucune nouvelle particule n'est créée par notre méthode.

Cette méthode permet ainsi d'obtenir un nombre réduit de particules actives nécessaires à une simulation de vagues déferlantes : les zones d'importance sont échantillonnées par des particules de taille minimale, alors que dans les zones moins significatives cette taille croît linéairement pendant la simulation, comme le montrent les images prises à différentes étapes de la simulation montrant les tailles adaptatives des particules (Figure 3.12 et 3.13).

### 3.3.3 Conservation de la symétrie des forces

Comme notre système de simulation présente des particules de taille variable, il est nécessaire de garantir la symétrie des forces internes. En effet, étant donné que les particules ont des rayons d'interactions différents, rien ne garantit le fait qu'une petite particule puisse interagir avec une plus grosse particule de façon réciproque. Ce problème a été étudié dans [DC99], dont la solution est celle du *shooting-gathering* qui consiste à forcer la symétrie des forces entre deux particules de tailles différentes et qui a été étendue par [APKG07] à l'expression des forces de pression et de viscosité proposée par [MCG03] :

$$\begin{cases} \vec{f}_i^{press} = -V_i V_j (p_i + p_j) (\nabla W(r, h_i) + \nabla W(r, h_j)) / 2 \\ \vec{f}_i^{viscosity} = \mu V_i V_j (\vec{U}_j - \vec{U}_i) (\nabla^2 W(r, h_i) + \nabla^2 W(r, h_j)) / 2 \end{cases} \quad (3.19)$$

avec  $V_i = \frac{m_i}{\rho_i}$  le volume occupé par une particule  $i$ ,  $h_i$  (resp.  $h_j$ ) le rayon d'interaction de la particule  $i$  (resp.  $j$ ) et  $r$  la distance entre les deux particules  $i$  et  $j$ .

### 3.3.4 Algorithme général

Nous modifions l'algorithme 1 de la manière suivante. A chaque pas de temps, nous identifions les particules fusionnées ou divisées. Si une particule est en cours de fusion, nous lui transférons une partie de la masse de ses filles. Dans le cas de la division, nous transférons une partie de sa masse vers ses filles. Nous évaluons alors le nouveau rayon d'interaction de chaque particule mère et fille à l'aide de l'équation 3.11 en prenant en compte sa nouvelle masse. Nous définissons une particule voisine  $j$  d'une particule  $i$  si  $r \leq \max(h_i, h_j)$ . Enfin, nous calculons les forces de pression et de viscosité à l'aide du système 3.19.

Nous avons utilisé une structure de données adaptative de type "KdTree" selon le rayon d'interaction des particules dans une région donnée. Pour déterminer les voisines d'une particule, nous parcourons alors seulement les voxels adjacents auquel elle appartient, ce qui permet d'accélérer le calcul du voisinage.

### 3.3.5 Performances

Pour observer les gains obtenus par notre approche, plusieurs simulations ont été menées sur un processeur AMD64 à 2,4 Ghz et 1 Go de mémoire vive. Le gain apporté par la multirésolution, en nombre de particules désactivées, croît en fonction du nombre de particules présentes dans la simulation ; de façon similaire, les temps de calculs sont accélérés de façon linéaire, comme le montre la figure 3.14. Par exemple, pour une simulation avec 60K particules initiales, nous obtenons un gain de 40% en termes de particules désactivées, et ainsi une simulation 1,5 fois plus

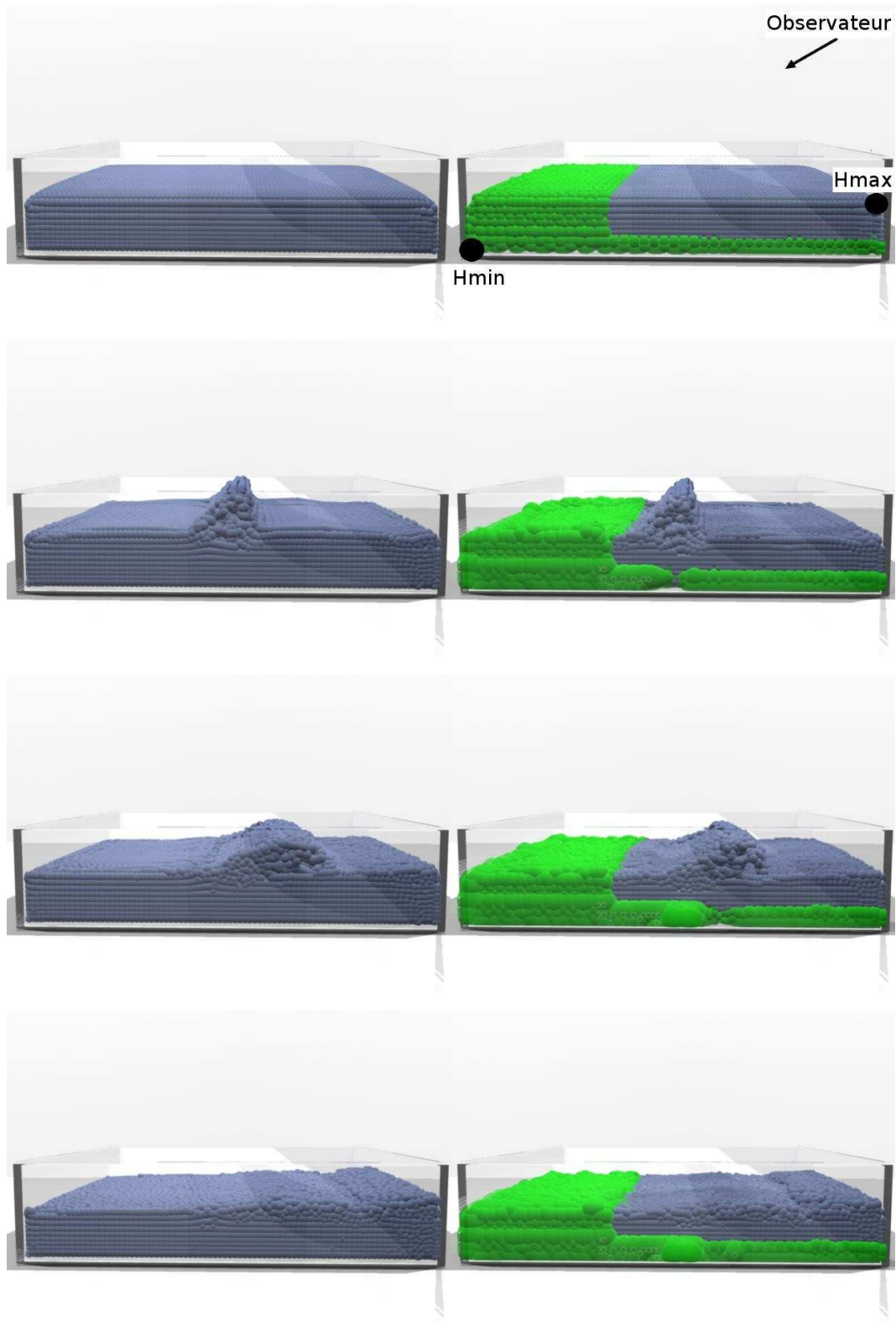


FIG. 3.12 – Comparaison de la simulation obtenue sans (à gauche) et avec (à droite) multirésolution avec 100K particules initiales et un soliton de paramètre  $(0.35, 0.1, 0, 0)$ . Le rayon des particule correspond au rayon d'interaction qui a été agrandi pour une meilleur visualisation.

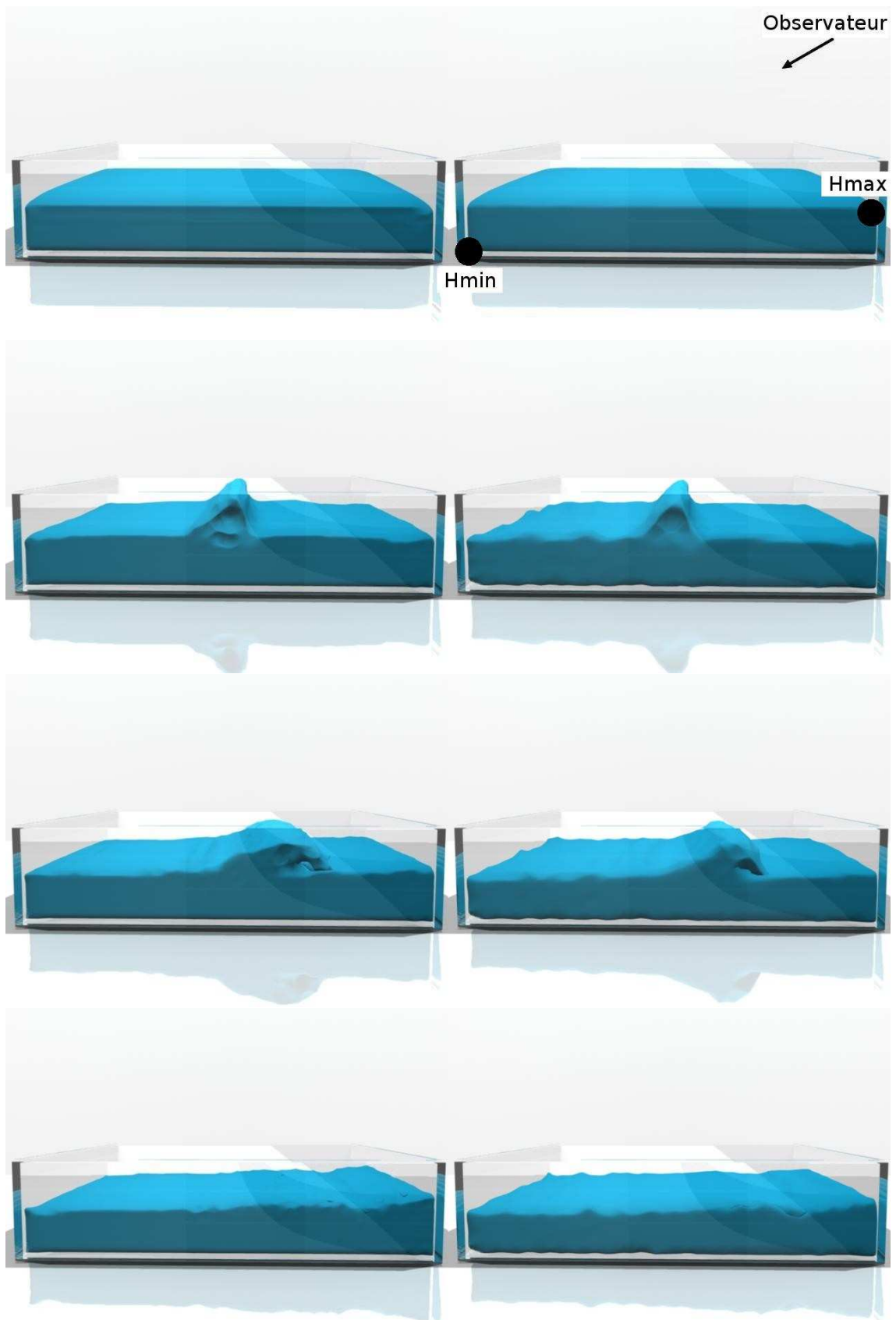


FIG. 3.13 – Comparaison des surfaces extraites des simulations de la figure 3.12.



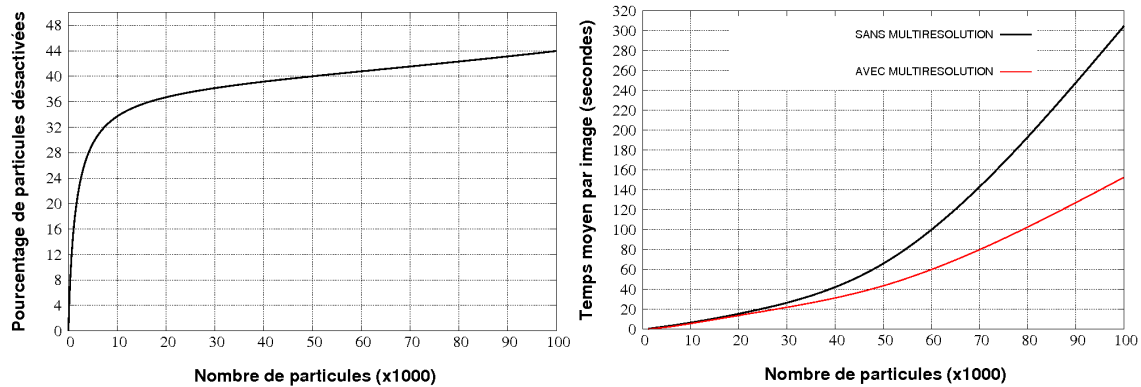


FIG. 3.14 – Gain amené par la multirésolution en pourcentage de particules désactivées (à gauche) et évolution du temps de calcul moyen en secondes par image (à droite) entre une simulation standard et multirésolution en fonction du nombre de particules initial.

rapide (environ 62 sec. par image en moyenne avec la multirésolution contre 108 en simulation standard). L'accélération qu'apporte notre approche est également illustrée sur la figure 3.1 qui montre une comparaison de la répartition en pourcentages des calculs avec et sans multirésolution. Comme nous pouvons le voir, notre méthode permet de diminuer les calculs de voisinages et donc par conséquent de réduire les temps de calculs du processus de résolution.

Comme le montre les figures 3.15 et 3.16, l'évolution de la moyenne des densités et des forces au cours d'une animation de 600 images avec multirésolution sont comparables à celles d'une simulation sans multirésolution. Au niveau des densités, l'erreur relative commise avec notre approche multirésolution est très faible (inférieure à 4%) alors qu'au niveau des forces, cette erreur est assez importante (supérieure à 20%). Pourtant, comme le confirme le résultat visuel des figures 3.12 et 3.13, les simulations avec et sans multirésolutions restent assez proches.

Une comparaison avec la méthode de [APKG07] a été effectuée (voir figure 3.17). Les différences fondamentales entre nos deux schémas multirésolution sont la prise en compte de la force d'impulsion comme critère de fusion et de division, et la mise en place d'heuristiques simples, ainsi qu'une approximation initiale de la surface, ce qui ne nécessite pas de calculs complexes de champs de distance entre une particule et la surface. Dans le cas présent, nous avons remarqué que dans la méthode de [APKG07] peu de particules n'étaient réellement supprimées du système. Ceci s'explique par le fait que cette méthode s'appuie principalement sur la distance entre une particule et l'interface, ce qui se traduit également par des calculs plus complexes

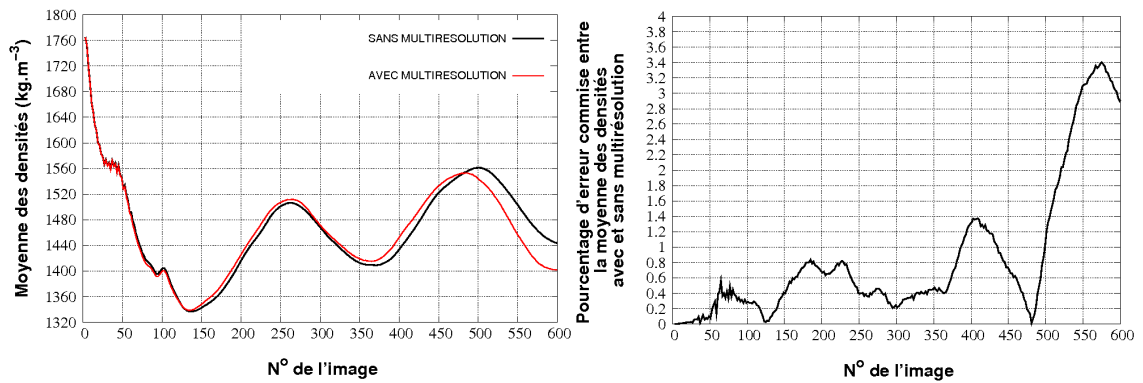


FIG. 3.15 – Comparaison de l'évolution de la moyenne des densités par image sur l'ensemble des particules actives de la simulation (à gauche) et estimation de l'erreur relative commise correspondante avec et sans multirésolution

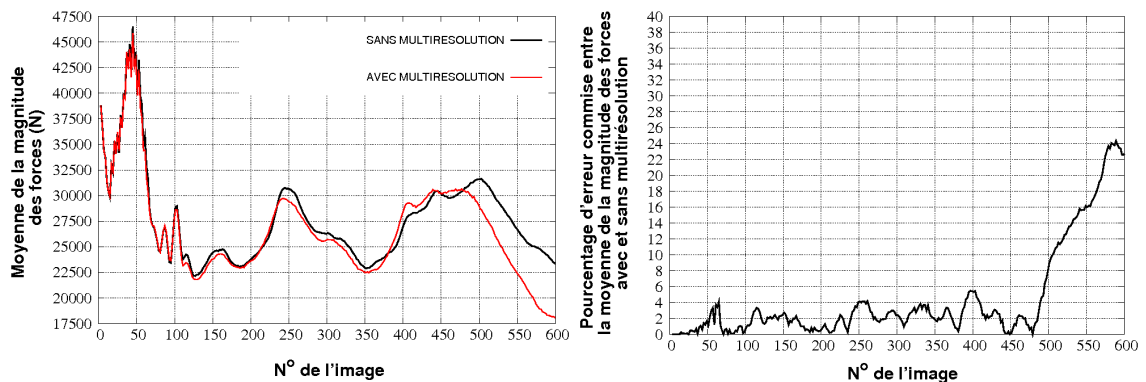


FIG. 3.16 – Comparaison de l'évolution de la moyenne de la magnitude des forces par image sur l'ensemble des particules actives de la simulation (à gauche) et estimation de l'erreur relative commise correspondante avec et sans multirésolution

| étape           | 50K         | 100K        | 200K        |
|-----------------|-------------|-------------|-------------|
| voisinage       | 37.9 [51.3] | 31.9 [42.9] | 35.5 [54.9] |
| densité         | 6.9 [6.4]   | 8.3 [7.5]   | 10.0 [7.7]  |
| forces          | 15.5 [12.8] | 11.5 [18.8] | 12.7 [17.6] |
| intégration     | 5.2 [6.4]   | 5.5 [6.8]   | 5.5 [5.2]   |
| multirésolution | 12.1 [-]    | 20.9 [-]    | 17.3 [-]    |
| embruns marins  | 22.4 [23.1] | 21.9 [24.0] | 19.0 [14.6] |

TAB. 3.1 – Comparaison des pourcentages des temps de calculs par étape avec et sans (entre crochets) multirésolution avec 50K, 100K et 200K particules initiales.

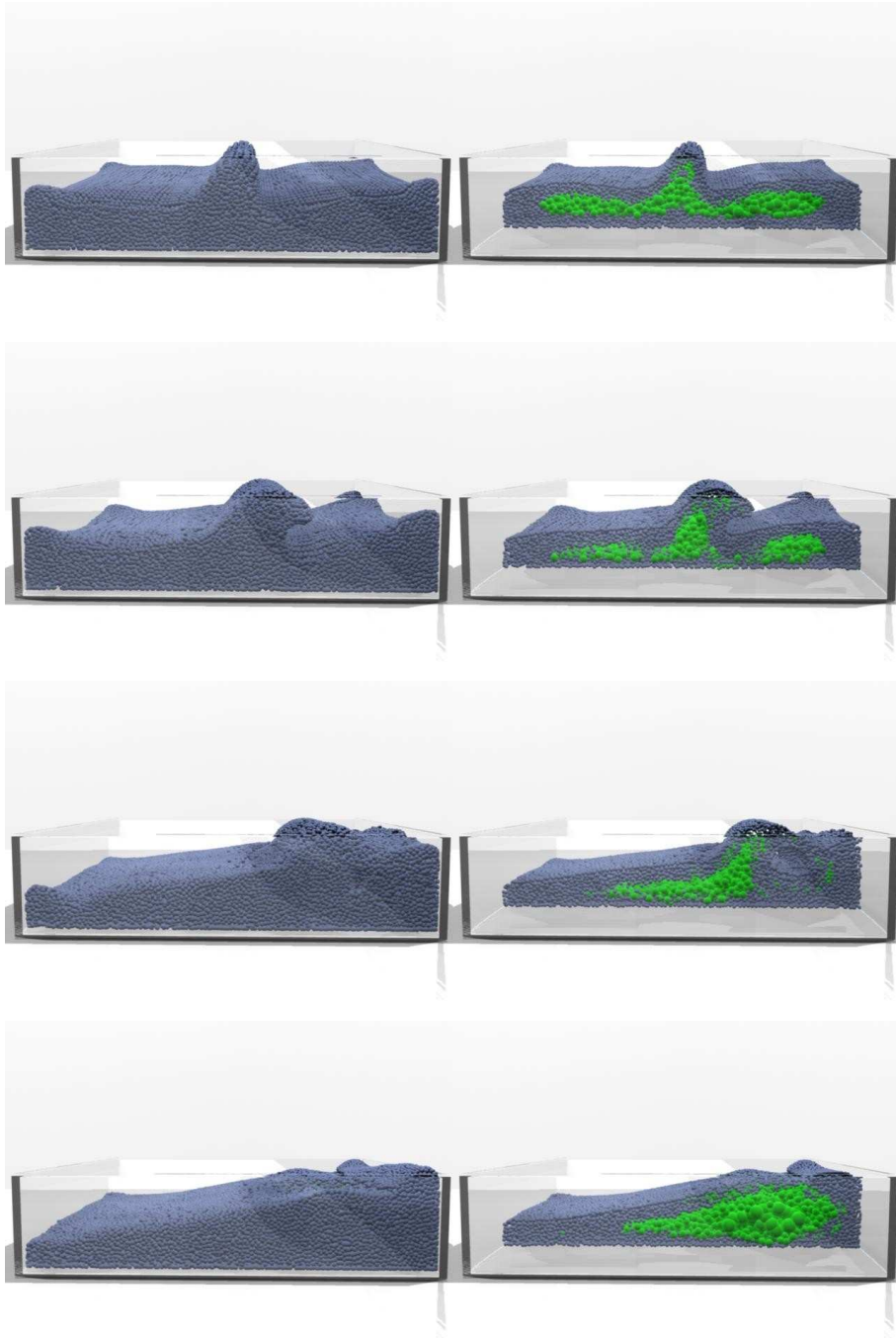


FIG. 3.17 – Simulation avec la méthode de [APKG07] (les images de la colonne de droite permettent de montrer les particules fusionnées en vert)

et plus longs. Or, dans le cas présent, nous n'avons besoin que peu de volume mais largement étendu afin de simuler une surface d'océan. Nous pouvons donc noter que cette approche n'est pas bien adaptée dans ce cas.

### 3.4 Définition de la surface libre

Pour le rendu des images présentes dans ce chapitre, nous avons extrait la surface à l'aide de la méthode des marching-cubes [LC87]. Nous avons également plaqué une texture d'océan en utilisant celle présente dans *Autodesk Maya* et le lancer de rayon *Mental-Ray* afin de calculer les effets de réflexions et de réfraction à la surface. Concernant l'extraction de la surface, nous avons également brièvement exploré au cours de ses travaux l'utilisation de surface variationnelle permettant la prise en compte du temps afin d'obtenir une animation cohérente. Nous décrivons ici ces deux approches.

#### 3.4.1 Extraction par l'algorithme des Marching Cubes

Comme nous l'avons vu dans le chapitre 1, la technique des Marching-Cubes permet d'extraire une surface explicite d'une surface définie implicitement. Comme nous l'avons souligné également, c'est une approche très utilisée par les systèmes lagrangiens car elle permet de déduire une triangulation de l'interface du fluide en fonction de la position de l'ensemble des particules en représentant implicitement l'interface à l'aide de la fonction suivante implicite [CBP05] :

$$\phi(x) = \left( \sum_N^i (1 - W(x, x_i)/R)^2 \right)^{1/2} \quad (3.20)$$

avec :

$$W(x, x_i) = \begin{cases} \|x - x_i\| & \text{si } \|x - x_i\| \leq h_i \\ 0 & \text{sinon} \end{cases}$$

avec  $x$  la position d'un coin d'un cube,  $x_i$  la position d'une particule et  $R$  désignant le coté de cube.

En utilisant cette formulation, nous pouvons prendre en compte la taille adaptative des particules caractérisées par des rayons d'interaction différents afin de déterminer une polygonisation de l'interface. Cependant, un inconvénient majeur provient du manque de cohérence temporelle du maillage généré. En effet, d'un instant à l'autre, la position des particules évolue, et certains triangles peuvent subitement apparaître ou disparaître. Nous présentons dans la section qui suit une approche pouvant limiter ces effets d'aliassages temporels.

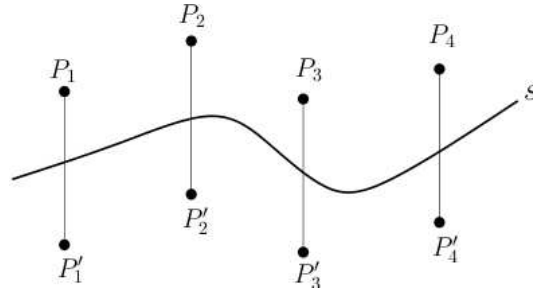


FIG. 3.18 – Localisation de la surface en fonction de la position des centres et de leurs valeurs.

### 3.4.2 Surfaces variationnelles

Les surfaces variationnelles sont un moyen très utilisé afin de représenter une surface. Le principe est d'utiliser des fonctions de bases radiales et de définir des points dans l'espace aidant à localiser la surface. Ces points sont appelés *centres* et possèdent une valeur scalaire représentant la distance signée à la surface. Par exemple, comme cela est illustré sur la figure 3.18, si nous définissons les centres positifs  $P_1$ ,  $P_2$ ,  $P_3$  et  $P_4$  de valeur  $+0.5$  et les centres négatifs  $P'_1$ ,  $P'_2$ ,  $P'_3$  et  $P'_4$  de valeur  $-0.5$ , nous pouvons déduire qu'une surface passe entre les centres positifs et négatifs.

Le champ scalaire définissant la surface  $s$  est défini par :

$$s(x) = P(x) + \sum_{i=1}^N \lambda_i \Phi(x - x_i) \quad (3.21)$$

avec :

- $x_i$  un point 3D appelé "centre"
- $P$  un polynôme de faible degré, typiquement linéaire ou quadratique
- $\lambda_i$  sont un ensemble de coefficients
- $\Phi$  une fonction appelée "fonction de base"

La surface est alors définie par l'ensemble des points  $x$  de l'espace tels que  $s(x) = 0$ . Notons  $P = \{p_1, \dots, p_l\}$  un ensemble de monômes associés à  $p$ ,  $C = \{c_1, \dots, c_l\}$  l'ensemble des coefficients du polynôme  $P$  et  $F$  l'ensemble des valeurs scalaires associées au centre  $x_i$ . L'équation 3.21 est alors équivalente au système linéaire suivant, écrit sous forme matricielle :

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ C \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \quad (3.22)$$

où :

$$A_{ij} = \Phi(|x_i - x_j|) \quad i, j = 1, \dots, N$$

et

$$P_{ij} = p_j(x_i) \quad i = 1, \dots, N \quad j = 1, \dots, l$$

En résolvant le système linéaire 3.22 nous pouvons déterminer  $\lambda$  et  $c$ , et ainsi connaître la valeur de la fonction  $s$  en tout point de l'espace.

Une propriété intéressante de la représentation variationnelle est la possibilité d'injecter une variable temporelle au sein de l'équation 3.21 et d'utiliser une surface variationnelle 4D comme cela est présenté dans [TO99]. L'idée de cette technique consiste alors à extraire une surface à différents instants clés et d'interpoler entre ces instants afin d'obtenir une animation cohérente.

Le problème de l'application de cette méthode avec un système de particule est la connaissance des points situés au-dessus et au-dessous de l'interface, puisque celle-ci n'est pas non plus connue. Nous proposons ici une idée de méthode permettant de calculer les centres négatifs et positifs en fonction du système de particules de simulation. En première étape, le principe serait d'utiliser une grille adaptative de type KdTree discrétisée selon le rayon d'interaction des particules. Les centres négatifs seraient les particules les plus proches de l'interface, identifiées en considérant les cellules voisines de la cellule à laquelle elles appartiennent et si au moins une de ces voisines est vide. Les centres positifs seraient identifiés comme les centres des cellules vides les plus proches de chacun des centres négatifs. La distance signée associée à un centre positif pourrait être calculée par :

$$d = \frac{\|x_c - x_i\| - m_i}{2} \quad (3.23)$$

avec  $x_c$  la position du centre positif,  $x_i$  la position de la particule la plus proche de  $x_c$  et  $m_i$  sa masse. Chaque centre négatif aurait alors une valeur scalaire égale à  $-d$ . Après cette phase d'identification, nous pourrions alors obtenir une surface en résolvant le système 3.22 à différents instants clés et en interpolant la surface entre deux instants clés.

Nous n'avons pas eu malheureusement le temps d'explorer plus cette piste mais elle pourrait éviter efficacement les effets d'aliassage temporel souvent observés avec la technique des marching-cubes.

## 3.5 Conclusion

Nous avons étudié dans cette partie une nouvelle méthode de représentation des vagues déferlantes en eaux peu profondes. Le modèle de forces d'impulsion finie

que nous avons introduit nous a permis de simuler ce phénomène sans limitation temporelle et de pallier aux inconvénients des approches existantes. Un schéma multirésolution a été également mis en oeuvre afin de limiter la complexité calculatoire. Nous avons vu que la méthode proposée pour la génération automatique des embruns marins nous a permis d'enrichir le rendu et d'obtenir des images plus réalistes. Enfin, la méthode d'extraction de la surface du fluide a été discutée et nous a permis d'envisager d'autres pistes afin de garantir la cohérence temporelle lors de l'animation.

# Chapitre 4

## Contrôle de fluide

L'approche lagrangienne que nous avons proposé dans le chapitre précédent ne permet pas de simuler rapidement le comportement de grandes étendues d'eau. Bien que l'emploi d'une approche adaptative soit capable de réduire les temps de calculs, elle ne permet pas à un utilisateur de constater rapidement si le résultat donné est proche du résultat souhaité. Nous proposons dans ce chapitre une méthode alternative permettant de remédier à ce problème. Ces travaux sont encore en cours de tests et nous présentons ici les premiers résultats.

Nous proposons une nouvelle approche de simulation lagrangienne basée sur l'emploi d'une structure de données hiérarchique, hautement contrôlable, permettant de visualiser rapidement la simulation physique, même avec un grand nombre de particules. Dans un premier temps, nous décrivons les fondements théoriques de notre méthode, notamment la construction de cette structure de données et le calcul des interactions des particules entre les différents niveaux hiérarchiques de ce système. Dans un second temps, nous montrons les résultats obtenus en discutant sur les performances de notre algorithme. Enfin nous concluons sur cette approche en examinant les différents avantages et inconvénients qu'elle présente.

### 4.1 Simulation SPH hiérarchique

Comme nous l'avons vu, l'approche multirésolution proposée dans le chapitre précédent s'appuie sur des heuristiques simples et efficaces. Bien qu'elle soit capable de réduire les temps de calculs, elle ne permet pas d'obtenir une visualisation rapide de la vague déferlante, obligeant l'utilisateur à effectuer plusieurs simulations afin d'obtenir l'effet souhaité. Plus largement, même l'emploi de schémas multirésolution tel que celui proposé par [APKG07] contraint souvent un utilisateur à effectuer plusieurs simulations afin de parvenir à l'objectif recherché.



Le principe de notre approche est d'introduire le concept de système SPH hiérarchique dont le but est de contrôler la simulation à plusieurs niveaux et d'en obtenir une pré-visualisation en utilisant le niveau le plus haut. Les particules d'un niveau supérieur peuvent contrôler dynamiquement les particules des niveaux inférieurs, ce qui permet de réduire considérablement les calculs. Nous détaillons dans cette partie les différentes étapes de notre méthode.

### 4.1.1 Fusion

Le processus de fusion proposé est très différent de celui présenté dans le chapitre précédent puisqu'il s'effectue au sens hiérarchique. Le but de cette opération n'est pas d'englober des particules dans une seule et même particule du système initial mais de regrouper certaines particules dans une particule appartenant à un autre système de simulation, moins dense en particules, ce qui a pour effet de réduire les calculs de voisinage et d'accélérer le processus de résolution.

Le principe de notre approche est la suivante. Au départ du processus, un utilisateur choisit le niveau maximal de la représentation hiérarchique, c'est à dire le nombre maximal de systèmes de particules générés en dehors du système de simulation classique. Chaque particule située au niveau le plus bas (le niveau 0) est englobée par une particule au niveau supérieur, elle-même englobée par une autre particule située au niveau plus supérieur et ceci récursivement. Les particules d'un niveau  $n$  sont donc englobées par une particule d'un niveau  $n + 1$ . Ce principe est illustré sur le schéma 4.1. Au départ de la simulation, nous parcourons la liste des particules du niveau le plus bas. Nous choisissons la particule la plus proche de la particule traitée et nous la marquons. Si cette particule se trouve à une distance inférieure à son rayon d'interaction alors nous créons une nouvelle particule au niveau  $n + 1$  dont le centre est situé au milieu des deux particules. La masse de chaque particule créée est calculée par la somme des masses des particules qu'elle englobe. Nous calculons alors son rayon d'interaction à l'aide de l'équation 3.11 (voir chapitre 3). Nous procédons de façon récursive sur l'ensemble des particules créées jusqu'à atteindre le niveau maximal, préalablement défini par l'utilisateur.

### 4.1.2 Calcul de la vitesse

Si nous observons les résultats obtenus dans le chapitre précédent, nous pouvons constater que les calculs de voisinage représentent environ 35% de l'ensemble des calculs effectués sur une particule en moyenne. Afin de réduire cette part de calculs,

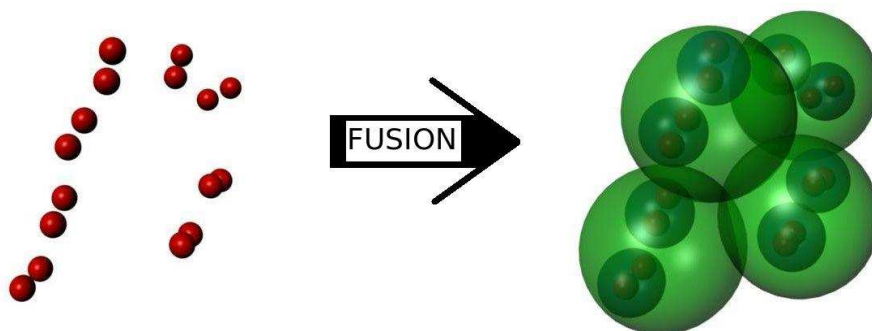


FIG. 4.1 – Processus de fusion hiérarchique

nous proposons de n'effectuer les calculs de voisinage qu'au niveau le plus supérieur, niveau possédant le moins de particules.

Afin de guider les particules des niveaux intermédiaires et de ne pas limiter les échanges à ceux calculés au niveau le plus supérieur, nous proposons aussi de décomposer la vitesse de chaque particule. En effet, nous soumettons chaque particule d'un niveau à deux formes d'interactions, celles représentées par la mère, elle-même soumise à son voisinage, et d'un autre côté, celles exercées par les autres particules filles.

Nous définissons alors la vitesse  $\vec{U}'_i$  d'une particule  $i$  située à un niveau  $n$  en fonction de la vitesse  $\vec{U}$  de sa particule mère située au niveau  $n + 1$  et en fonction de la vitesse  $\vec{U}_i$  calculée en prenant en compte les interactions avec ses soeurs appartenant à la même particule mère et évaluée de la même façon que pour une approche SPH classique.

$$\vec{U}'_i = \vec{U}_i + \tilde{U}W(r) \quad (4.1)$$

avec  $r = \|x - \tilde{x}\|$  étant la distance entre la particule fille et la particule mère et  $W$  une fonction de pondération. Nous avons utilisé une fonction gaussienne :

$$W(r) = e^{-\frac{R}{2}}$$

avec :

$$R = \begin{cases} \frac{\|x - \tilde{x}\|}{\tilde{h}} & \text{si } \|x - \tilde{x}\| \leq \tilde{h} \\ 0 & \text{sinon} \end{cases}$$

avec  $x$  et  $\tilde{x}$  la position de la particule fille et de la particule mère et  $\tilde{h}$  le rayon d'interaction de la particule mère.

En effet, si nous considérons la distance d'une particule fille à sa mère, celle-ci pourra conduire dynamiquement sa fille si cette distance est assez faible, c'est à dire si elle

est inférieure au rayon d'interaction  $\tilde{h}$  de la particule mère.

Nous procédons selon un ordre descendant. Dans un premier temps, nous calculons les interactions au plus haut niveau de façon classique. Dans un second temps, nous procédons par récursion sur l'ensemble des niveaux, en calculant les sous-interactions entre les particules filles. Sur chaque particule de chaque niveau, nous calculons la position de la particule en fonction de la vitesse de sa mère, mais aussi de celle déduite par les échanges avec ses soeurs. L'ensemble de ces étapes est illustré dans l'algorithme 2.

### 4.1.3 Evaluation de la densité

Au niveau le plus haut, les interactions sont calculées de manière classique en fonction du voisinage de la particule. La densité d'une particule située sur ce niveau est donc évaluée en fonction de son voisinage dans le système global.

Comme le calcul des interactions entre les particules filles s'effectue lui aussi de manière interne à la particule, nous estimons la densité d'une particule fille également de façon interne, c'est à dire en ne considérant que les particules appartenant à la même particule mère. Ceci a pour effet de réduire considérablement les temps de calcul. En effet, comme chaque particule mère ne contient qu'un faible nombre de particules filles, cette quantité est ainsi rapidement évaluée.

### 4.1.4 Changement de mère

Au cours de la simulation, nous choisissons de transférer une particule fille vers une autre particule mère si la distance avec le centre de sa mère d'origine est supérieure au rayon d'interaction  $\tilde{h}$  de celle-ci. En effet, si c'est le cas, cela signifie que la particule fille est trop éloignée de la particule mère et qu'elle ne peut donc plus être représentée dynamiquement par celle-ci. Nous procédons avec le niveau le plus inférieur et nous recalculons la nouvelle particule mère en fonction du voisinage calculé à la volée de la mère d'origine.

Quand nous avons identifié la particule mère, nous transférons alors toute la masse de la nouvelle particule fille et nous recalculons son nouveau rayon d'interaction (équation 3.11). Nous procédons alors par récursion sur l'ensemble des niveaux du système hiérarchique de manière ascendante afin de privilégier les particules du niveau le plus bas (voir figure 4.2).

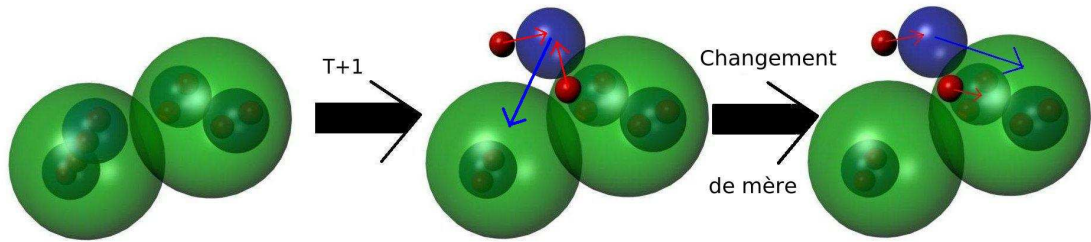


FIG. 4.2 – Processus de changements de mère - Les flèches indiquent les liaisons entre les particules filles et les particules mères.

### 4.1.5 Algorithme Général

Les différentes étapes de notre méthode sont illustrées dans l’algorithme 2. Nous procédons par récursion sur l’ensemble des niveaux. Au niveau le plus haut, nous utilisons une simulation classique, c’est à dire l’algorithme 1. Dans chacun des niveaux intermédiaires et pour chaque particule, nous calculons sa densité, les interactions avec ses soeurs puis sa vitesse. Ceci revient à utiliser une approche classique interne à chaque particule mère. Enfin, nous modifions sa vitesse en fonction du déplacement de sa mère, comme cela est expliqué dans la section précédente.

## 4.2 Résultats

Dans un premier temps, nous nous sommes intéressés à comparer notre approche hiérarchique avec une approche standard en utilisant différents niveaux maximal. nous avons choisi d’englober au plus 4 particules dans chacune particule d’un niveau supérieur. La figure 4.4 présente les résultats de ces simulations en utilisant 2 et 3 niveaux. Comme nous pouvons le voir, le résultat de la seconde colonne est assez proche de celui de la simulation standard (1ère colonne).

La figure 4.3 illustre les différents niveaux hiérarchiques pour une simulation avec 3 niveaux. Pour des raisons de clarté, les particules sont affichées avec une taille fixe, ne correspondant pas à leur rayons d’interaction ou à leur masse. Nous pouvons observer sur ces images que le comportement dynamique du fluide reste réaliste malgré l’emploi des approximations qui sont effectuées au sein de notre méthode.

Quantitativement, la figure 4.5 montre l’évolution de la moyenne du nombre de Reynolds sur l’ensemble des particules du niveau le plus bas. Ce nombre sans dimension

**Algorithme 2** Algorithme de simulation SPH hiérarchique, avec  $N_{MAX}$  correspondant au nombre de niveau hiérarchique maximal et  $n_N$  représentant le nombre de particules d'un niveau  $N$

**Algorithme SPH hierarchique**

**Variables**

|  $N_{MAX}$  : entier

**Début**

| **Pour**  $i$  **variantDe** 1 à  $n_{N_{MAX}}$  **Faire**

| | SPH( $i$ )

| **FinPour**

| **Pour**  $N$  **variantDe**  $N_{MAX} - 1$  à 1 **Faire**

| | **Pour**  $i$  **variantDe** 1 à  $n_N$  **Faire**

| | | SPH( $i$ )

| | |  $U_{i_{t+1}} \leftarrow U_{i_t} + \tilde{U}W(r)$

| | **FinPour**

| | **Pour**  $N$  **variantDe** 1 à  $N_{MAX} - 1$  **Faire**

| | | **Pour**  $i$  **variantDe** 1 à  $n_N$  **Faire**

| | | | Division( $i$ )

| | | **FinPour**

| | **FinPour**

| **Fin**

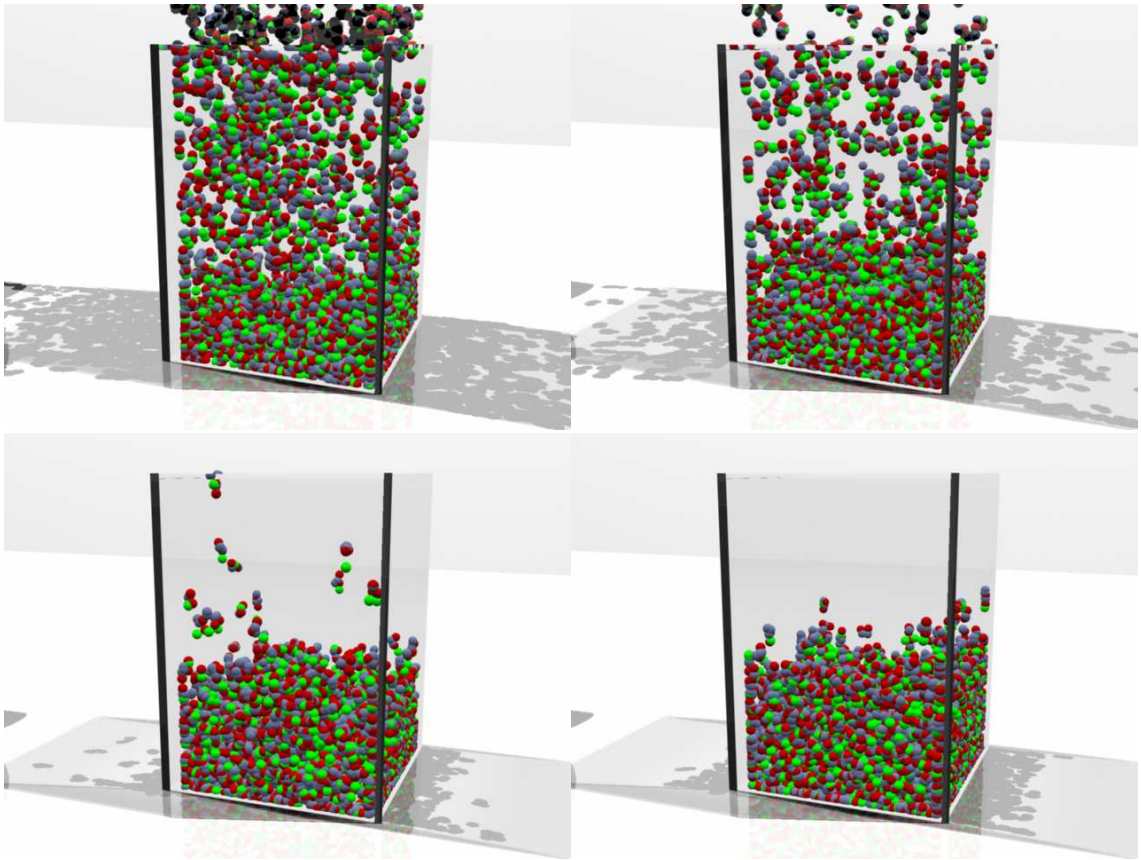


FIG. 4.3 – Exemple de simulation SPH hiérarchique sur 3 niveaux - Les particules en bleu sont celles du niveau 1, en rouge, du niveau 2 et en vert du niveau 3.

est calculé en utilisant la formule suivante :

$$Re_i = \frac{\|\vec{U}_i\|}{\rho_i \eta} \quad (4.2)$$

avec  $\vec{U}_i$  la vitesse de la particule  $i$ ,  $\rho_i$  sa densité et  $\eta$  la viscosité du fluide considéré (l'eau dans notre cas). Selon le principe de similitude en mécanique des fluides, deux écoulements sont dits similaires si leurs nombres de Reynolds sont égaux. Nous pouvons noter que l'évolution de ce nombre pour une simulation avec 2 et 3 niveaux est très proche de celle de la simulation standard (en noir). Cependant, au niveau 5 l'évolution de ce nombre diffère de celle de la simulation de référence. Ceci nous permet de supposer qu'il doit exister un niveau seuil qui dépend du nombre de particules présents sur ce niveau.

Nous pouvons voir dans le tableau 4.1 que notre approche permet d'accélérer considérablement les temps de calculs par rapport à une méthode classique. Par exemple, pour un système contenant  $50K$  particules au niveau le plus bas, nous obtenons une simulation 3, 5 et 13, 5 fois plus vite en utilisant deux et trois niveaux hiérarchiques qu'en utilisant un seul niveau. Dans ces résultats, le calcul du voisinage au niveau le plus haut est effectué à l'aide d'une double itération sur l'ensemble des particules, ce qui peut être optimisé en employant une grille spatiale 3D, comme nous l'avons fait dans le chapitre précédent.

Cette approche permet aussi un fort contrôle utilisateur en se basant sur le même principe que la méthode de [TKRP06], c'est à dire en ajoutant des forces de contrôle définies par l'utilisateur sur certaines particules SPH. En effet, un utilisateur peut agir directement sur les particules du plus haut niveau hiérarchique en rajoutant des forces, ce qui a pour effet de modifier le comportement dynamique des particules des niveaux inférieurs. Ce principe peut être utilisé pour la simulation de vagues déferlantes avec la méthode que nous avons proposé dans le chapitre précédent (voir figure 4.6).

### 4.3 Conclusions

Nous avons présenté dans ce chapitre une nouvelle méthode de simulation SPH, permettant une représentation hiérarchique du système global et une très forte accélération du processus de simulation. Nous avons montré comment il était possible d'approximer hiérarchiquement les échanges entre les particules des différents niveaux. Ce type de représentation permet aussi d'induire un contrôle utilisateur qui peut agir directement sur les particules du niveau le plus haut, conduisant ainsi les

| $N_{MAX}$ | 10K  | 20K  | 50K   |
|-----------|------|------|-------|
| 1         | 25.1 | 93.9 | 364.0 |
| 2         | 7.2  | 25.3 | 99.2  |
| 3         | 1.8  | 7.2  | 27.4  |
| 5         | 0.6  | 1.3  | 4.1   |

TAB. 4.1 – Temps de calculs moyen par images (en secondes) avec différents niveaux hiérarchiques pour 10K, 20K et 50K particules initiales.

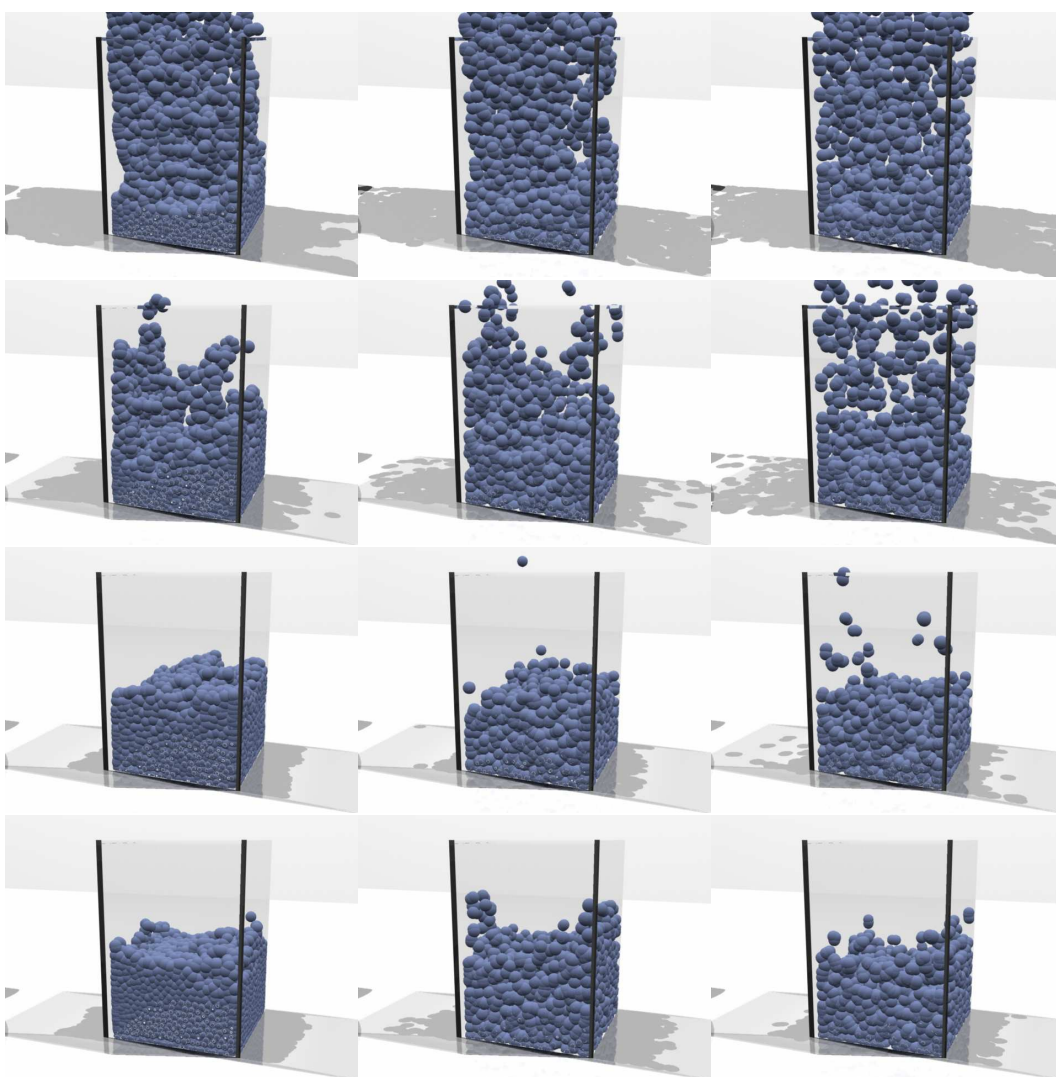


FIG. 4.4 – Comparaison de notre approche avec une approche classique - La première colonne représente la simulation SPH standard (avec 1 seul niveau). Les deuxième et troisième colonnes représentent respectivement une simulation SPH hiérarchique avec 2 et 3 niveaux de simulation



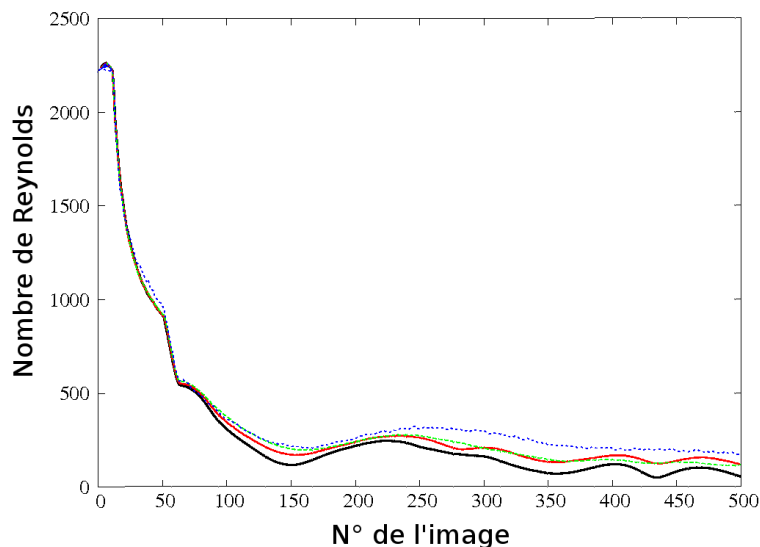


FIG. 4.5 – Comparaison du nombre de Reynolds avec une approche classique (en noir) pour différentes simulations hiérarchiques avec 2 niveaux (en rouge), 3 niveaux (en vert) et 5 niveaux (en bleu)

particules des sous-niveaux. Ce contrôle peut être utilisé dans le cadre de la simulation de vagues déferlantes par exemple, en utilisant le modèle de la force d'impulsion proposé au chapitre précédent.

Cette méthode est actuellement encore en cours de tests mais les premiers résultats obtenus semblent assez prometteurs. Plusieurs pistes restent ouvertes. Par exemple, l'utilisation de d'autres fonctions de pondération pour l'approximation de la vitesse d'une particule fille, devra être discutée. D'autre part, étant donné que les structures de données utilisées sont de simples tableaux de particules pouvant être transposées sous forme de textures, cette méthode peut être facilement implémentée sous GPU, ce qui permettrait d'accélérer les calculs et d'obtenir, peut être, un résultat en temps réel, même avec un grand nombre de particules. Nous souhaiterions explorer cette piste dans un futur proche.

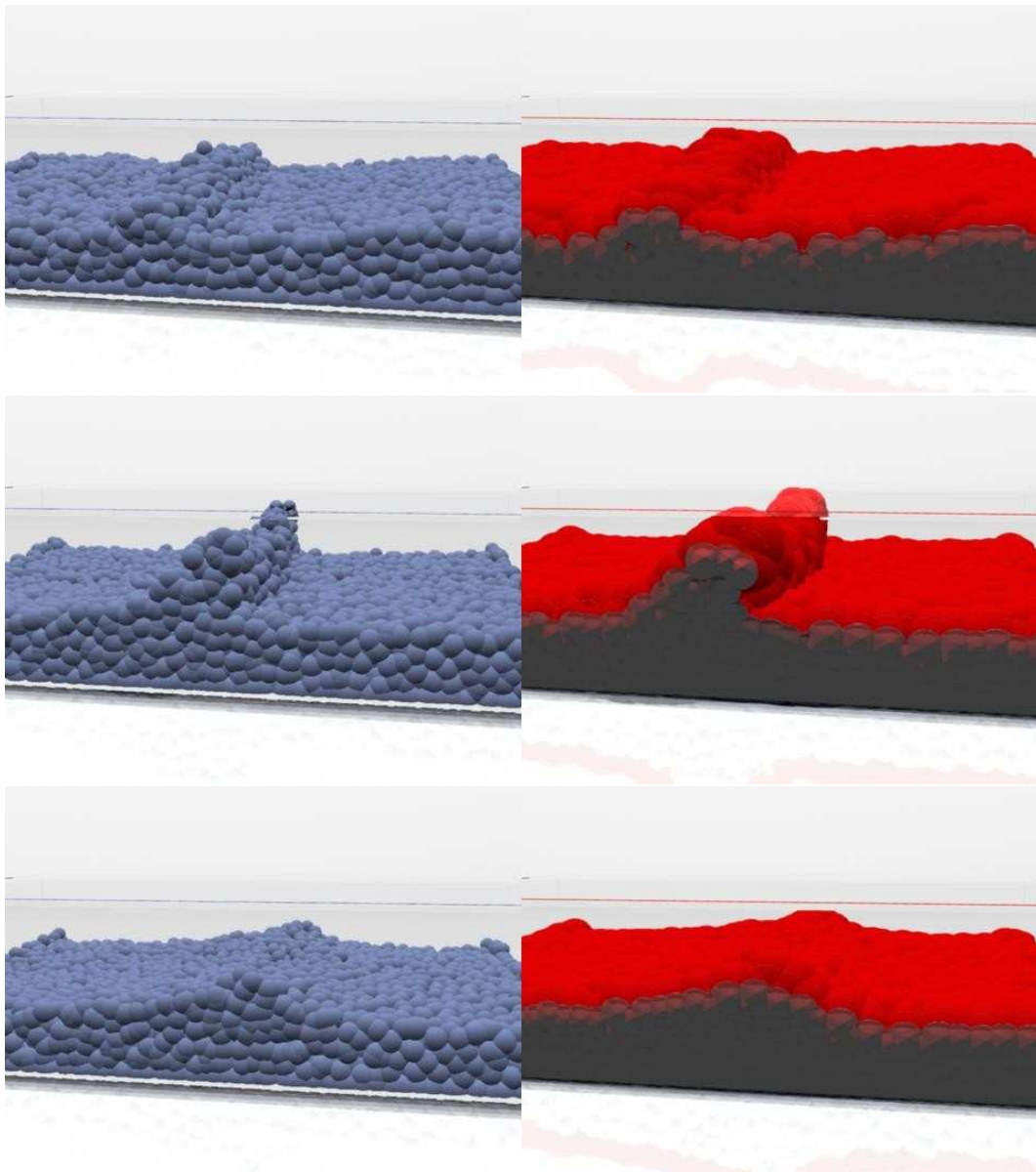


FIG. 4.6 – Simulation de vagues déferlantes avec notre système hiérarchique - La première colonne représente les particules du niveau le plus bas qui sont contrôlées hiérarchiquement par les particules du niveau le plus haut, affichées sur la seconde colonne.



# Conclusion

## Travaux réalisés

Comme nous l'avons vu, l'océan est un sujet de recherche complexe autant d'un point de vue physique que d'un point de vue informatique. Dans ce contexte, nous avons vu qu'il existait plusieurs problématiques. D'une part, au niveau du rendu, où il manquait une méthode de rendu unifiée et une alternative au calcul coûteux du point d'intersection entre un rayon et les modèles de représentation paramétriques utilisés pour la modélisation de la surface de l'océan. Dans ce contexte, nous avons proposé un nouvel algorithme permettant un rendu réaliste en limitant les coûts de calculs. Nous avons montré comment cette méthode pouvait intégrer la simulation de divers phénomènes tels que les échanges lumineux surfaciques ou sous-surfaciques. Nous avons également proposé une nouvelle approche de simulation de l'écume et de l'éblouissement, en gardant le même souci de réduire les temps de calcul et l'occupation mémoire dédiée à la simulation de ces effets.

D'autre part, nous nous sommes intéressés au déferlement des vagues, phénomène complexe, riche scientifiquement et visuellement. Comme nous l'avons vu dans le premier chapitre, aucune méthode ne permettait sa représentation à la fois dans sa diversité mais aussi dans son intégralité. Pour pallier à ces limitations, nous avons proposé une nouvelle méthode, basée physique et adaptative, capable de simuler différents types de déferlement. Le modèle proposé permet également un contrôle utilisateur afin de définir par exemple la géométrie de la vague déferlante. Nous nous sommes aussi intéressés au rendu des embruns marins en proposant une nouvelle en réduisant les coûts de calcul engendrés par la représentation de ce phénomène sur de grandes étendues d'eau.

Dans le dernier chapitre, nous avons également proposé une nouvelle méthode basée physique pour résoudre les équations de Navier-Stokes de façon plus rapide. Nous avons introduit la notion de système SPH hiérarchique qui permet une très forte réduction des temps de calculs et d'obtenir des résultats de simulation proches de

l'interactivité. Comme nous l'avons vu, cette approche permet également de rajouter un haut degrés de contrôle utilisateur, et peut s'adapter au modèle proposé dans le troisième chapitre pour le déferlement des vagues.

## Perspectives

Les perspectives possibles de ces travaux sont multiples. D'une part, au niveau du rendu, d'autres phénomènes peuvent être pris en compte, tel que les caustiques. Comme nous l'avons vu, il serait également intéressant d'intégrer une méthode permettant de représenter des embruns marins au sein de notre algorithme de rendu. D'autre part, notre méthode de simulation de vagues déferlantes ne permet pas de représenter l'écume due au déferlement. Plus largement, il faudrait étudier une méthode de rendu unifiée de vagues déferlantes, un peu de la même façon que celle que nous avons proposé en eaux profondes dans le second chapitre.

Comme nous l'avons vu dans le chapitre 4, notre système SPH hiérarchique repose sur l'approximation de la vitesse des niveaux hiérarchiques intermédiaires en fonction de celles calculées au niveau le plus supérieur. Nous avons utilisé une fonction gaussienne. Pourtant, il serait judicieux de mener une étude sur d'autres fonctions d'interpolation afin de trouver celle la plus adaptée. Nous avons présenté les premiers résultats de notre méthode. D'autres tests devront être effectués pour déduire les limites possibles de cette méthode.

Enfin, il serait intéressant d'étudier un modèle capable de faire la jonction entre le modèle en eaux profondes et celui proposé en eaux peu profondes. Ceci représente un sujet complexe car ces deux modèles de représentation sont très différents autant du point de vue de la modélisation mais aussi de l'animation. Pourtant, un modèle capable de faire cette jonction permettrait d'obtenir une représentation de la surface de l'océan par niveaux de détails, en utilisant une approche paramétrique peu coûteuse au loin, et une approche physique près des côtes ou de l'observateur.

Je tiens à remercier le Conseil Régional Limousin pour le financement de ces travaux de thèse.

## Annexe A : Coefficient d'atténuation de l'eau pure

Le coefficient d'atténuation de l'eau pure est utilisé notamment dans le calcul de la diffusion sous-surfacique (voir chapitre 2).

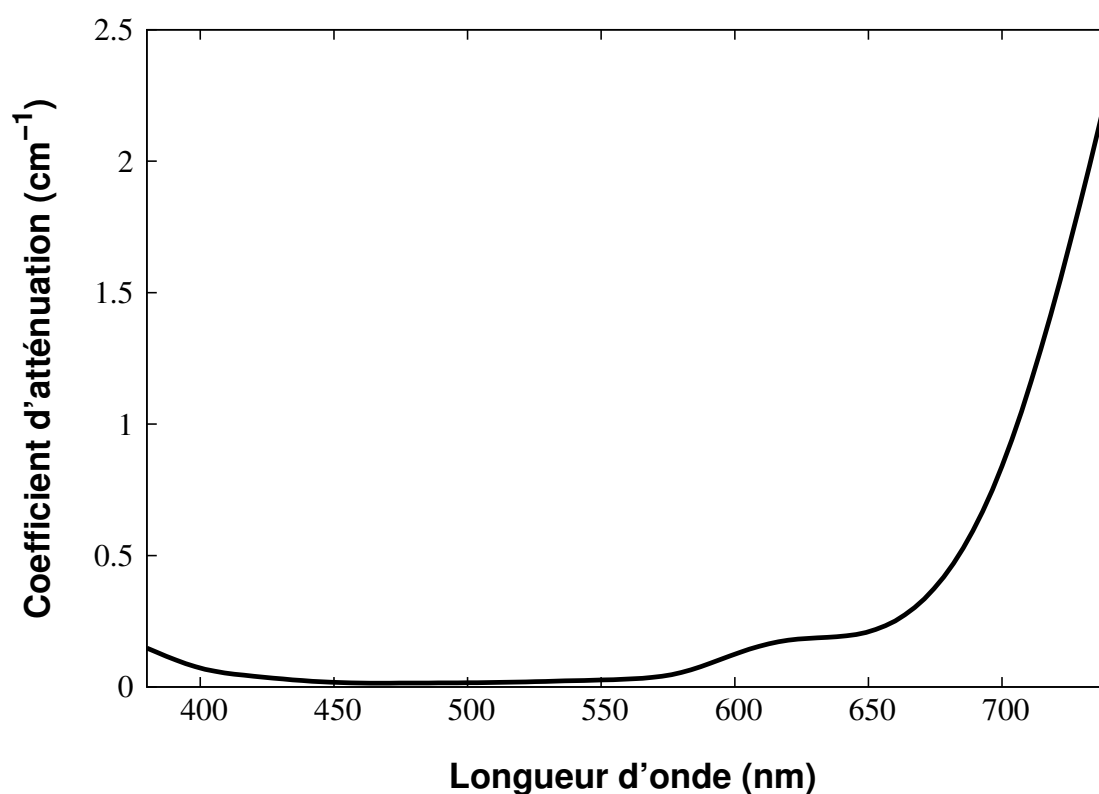


FIG. 4.7 – Coefficient d'atténuation de l'eau pure - données extraites de [GM83]



## Annexe B : Expression des fonctions $W_{poly6}$ , $W_{spiky}$ et $W_{visc}$

La fonction  $W_{poly6}$  utilisée dans le calcul de la densité et des forces de tension de surface est exprimée par :

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

$$\nabla W_{poly6}(\mathbf{r}, h) = -\frac{945}{32\pi h^9} \begin{cases} \mathbf{r}(h^2 - \|\mathbf{r}\|^2)^2 & \text{si } 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \text{sinon} \end{cases}$$

$$\nabla^2 W_{poly6}(\mathbf{r}, h) = -\frac{945}{32\pi h^9} \begin{cases} (h^2 - \|\mathbf{r}\|^2)(3h^2 - 7\|\mathbf{r}\|^2) & \text{si } 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \text{sinon} \end{cases}$$

La fonction  $W_{spiky}$  utilisée pour le calcul des forces de pression est exprimée par :

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

$$\nabla W_{spiky}(\mathbf{r}, h) = -\frac{45}{\pi h^6} \begin{cases} \frac{\mathbf{r}}{\|\mathbf{r}\|}(h - \|\mathbf{r}\|)^2 & \text{si } 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \text{sinon} \end{cases}$$

$$\nabla^2 W_{spiky}(\mathbf{r}, h) = -\frac{90}{\pi h^6} \begin{cases} \frac{1}{\|\mathbf{r}\|}(h - \|\mathbf{r}\|)(h - 2\|\mathbf{r}\|) & \text{si } 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \text{sinon} \end{cases}$$

La fonction  $W_{visc}$  utilisée pour le calcul des forces de viscosité est définie par :

$$W_{visc}(\mathbf{r}, h) = \frac{15}{2\pi h^3} \begin{cases} \left(-\frac{\|\mathbf{r}\|^3}{2h^3} + \frac{\|\mathbf{r}\|^2}{h^2} + \frac{h}{2\|\mathbf{r}\|} - 1\right) & \text{si } 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \text{sinon} \end{cases}$$

$$\nabla W_{visc}(\mathbf{r}, h) = \frac{15}{2\pi h^3} \begin{cases} \mathbf{r}\left(-\frac{3\|\mathbf{r}\|}{2h^3} + \frac{2}{h^2} - \frac{h}{2\|\mathbf{r}\|^3}\right) & \text{si } 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \text{sinon} \end{cases}$$



$$\nabla^2 W_{visc}(\mathbf{r}, h) = \frac{45}{\pi h^6} \begin{cases} (h - \|\mathbf{r}\|) & \text{si } 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \text{sinon} \end{cases}$$

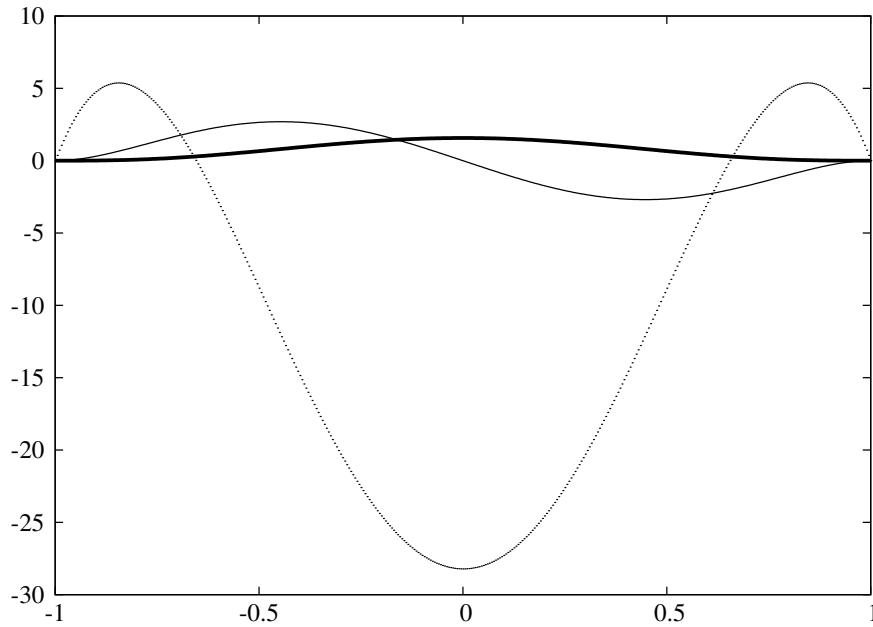


FIG. 4.8 – Représentation graphique de la fonction  $W_{poly6}(\mathbf{r}, h)$  (en trait plein),  $\nabla W_{poly6}(\mathbf{r}, h)$  (en trait fin) et  $\nabla^2 W_{poly6}(\mathbf{r}, h)$  (en pointillé) pour  $h = 1.0$

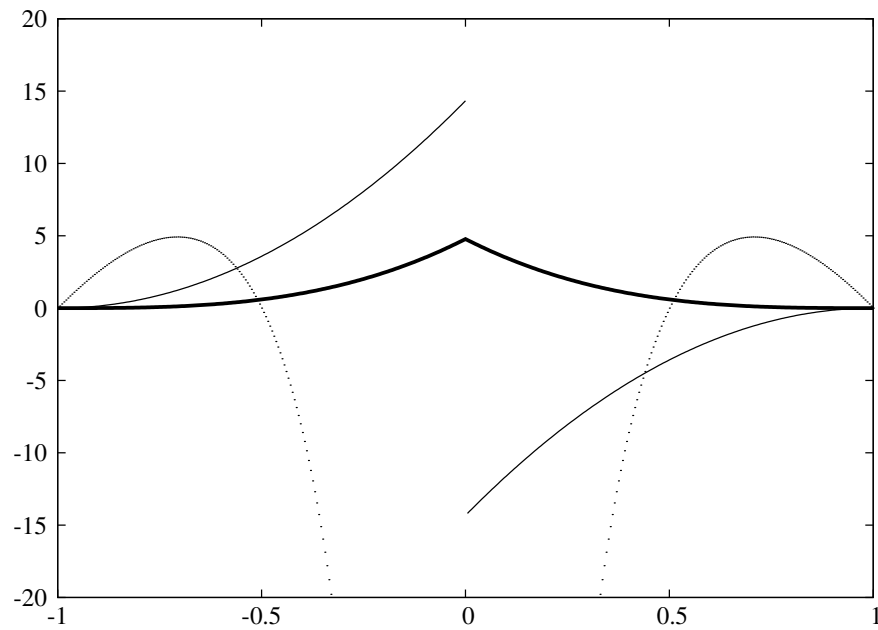


FIG. 4.9 – Représentation graphique de la fonction  $W_{spiky}(\mathbf{r}, h)$  (en trait plein),  $\nabla W_{spiky}(\mathbf{r}, h)$  (en trait fin) et  $\nabla \nabla W_{spiky}(\mathbf{r}, h)$  (en pointillé) pour  $h = 1.0$

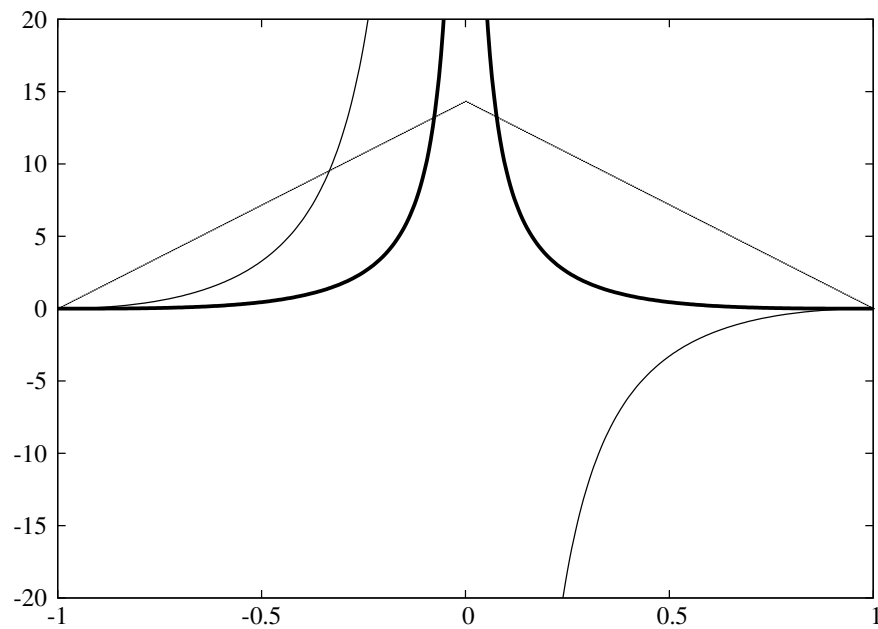


FIG. 4.10 – Représentation graphique de la fonction  $W_{visc}(\mathbf{r}, h)$  (en trait plein),  $\nabla W_{visc}(\mathbf{r}, h)$  (en trait fin) et  $\nabla \nabla W_{visc}(\mathbf{r}, h)$  (en pointillé) pour  $h = 1.0$



---

## Bibliographie

- [APKG07] B. Adams, M. Pauly, R. Keiser, and L. J. Guibas. Adaptively sampled particle fluids. In *ACM Transactions on Graphics (SIGGRAPH '07 papers)*, volume 26, New York, NY, USA, 2007. ACM Press.
- [Aub02] M. Aube. *Initiation a la teledetection et a la modelisation de l'atmosphere*. Groupe de recherche et d'application en physique au College de Sherbrook, 2002.
- [Bat74] J. A. Battjes. Surf similarity. In *Proceedings of 14th Coastal Engineering Conference*, pages 466–480, 1974.
- [BD06] L. Baboud and X. Décoret. Realistic water volumes in real-time. In *Eurographics Workshop on Natural Phenomena*. Eurographics, 2006.
- [Bel03] V. Belyaev. Real-time simulation of water surface. In *MAX Press Editor. Conference Proceedings*, 2003.
- [Bie52] F. Biesel. Study of wave propagation in water of gradually varying depth. In *Gravity Waves*, pages 243–253, 1952.
- [BMFG06] R. Bridson, M. Müller-Fischer, and E. Guendelman. Fluid simulation. *SIGGRAPH 2006 Course Notes*, 2006.
- [BN76] J.F. Blinn and M.E. Newell. Texture and reflection in computer generated images. *Commun. ACM*, 19(10) :542–547, 1976.
- [BT07] M. Becker and M. Teschner. Weakly compressible sph for free surface flows. In *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 209–217, 2007.
- [BYM05] N. Bell, Y. Yu, and P. J. Mucha. Particle-based simulation of granular materials. In *SCA '05 : Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 77–86, New York, NY, USA, 2005. ACM.
- [CBP05] S. Clavet, P. Beaudoin, and P. Poulin. Particle-based viscoelastic fluid simulation. In *Symposium on Computer Animation 2005*, pages 219–228, july 2005.

- [CC06] Y-F Chiu and C-F Chang. Gpu-based ocean rendering. *IEEE International Conference on Multimedia and Expo*, pages 2125–2128, 2006.
- [CL95] J. X. Chen and N. V. Lobo. Toward interactive-rate simulation of fluids with moving obstacles using navier-stokes equations. *Graph. Models Image Process.*, 57(2) :107–116, 1995.
- [CMT04] M. Carlson, P. J. Mucha, and G. Turk. Rigid fluid : animating the interplay between rigid bodies and fluid. In *SIGGRAPH '04 : ACM SIGGRAPH 2004 Papers*, pages 377–384, New York, NY, USA, 2004. ACM.
- [CS04] E. Cerezo and F. J. Seron. Rendering natural waters taking fluorescence into account. *Computer Animation and Virtual Worlds*, 15(5) :471–484, 2004.
- [DC96] M. Desbrun and M.P. Cani. Smoothed particles : A new paradigm for animating highly deformable bodies. In R. Boulic and G. Hegron, editors, *Eurographics Workshop on Computer Animation and Simulation (EGCAS)*, pages 61–76. Springer-Verlag, Aug 1996. Published under the name Marie-Paule Gascuel.
- [DC99] M. Desbrun and M.-P. Cani. Space-time adaptive simulation of highly deformable substances. Rapport de recherche 3829, INRIA, December 1999.
- [DCG07a] E. Darles, B. Crespín, and D. Ghazanfarpour. Accelerating and enhancing rendering of realistic ocean scenes. In *WSCG*, 2007.
- [DCG07b] E. Darles, B. Crespín, and D. Ghazanfarpour. Une approche multirésolution lagrangienne pour la simulation de vagues déferlantes. In *Journées AFIG*, 2007. Prix du meilleur papier, sélectionné pour publication dans REFIG.
- [DCG08] E. Darles, B. Crespín, and D. Ghazanfarpour. A multiresolution approach for the sph simulation of breaking waves. Technical report, XLIM, Université Limoges, 2008.
- [EMF02] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. *ACM Trans. Graph.*, 21(3) :736–744, 2002.
- [FF01] N. Foster and R. Fedkiw. Practical animation of liquids. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30, New York, NY, USA, 2001. ACM.
- [Fin04] M. Finch. Effective water simulation from physical models. *GPU Gems*, pages 5–29, 2004.

- [FM96] N. Foster and D. Metaxas. Realistic animation of liquids. *Graph. Models Image Process.*, 58(5) :471–483, 1996.
- [FM97] N. Foster and D. Metaxas. Controlling fluid animation. In *CGI '97 : Proceedings of the 1997 Conference on Computer Graphics International*, page 178, Washington, DC, USA, 1997. IEEE Computer Society.
- [FR86] A. Fournier and W.T. Reeves. A simple model of ocean waves. *ACM Computer Graphics (SIGGRAPH '86 Proceedings)*, pages 75–84, 1986.
- [GH04] S. T. Greenwood and D. H. House. Better with bubbles : enhancing the visual realism of simulated fluid. In *SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 287–296, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
- [GM82] R. Gingold and J. Monaghan. Kernel estimates as a basis for general particle method in hydrodynamics. *Journal of Computational Physics*, 46 :429–453, 1982.
- [GM83] H. R. Gordon and A. Morel. Remote assesment of ocean color for interpretation of satellite visible imagery, a review. *Lecture Notes on Coastal and Estuarine Studies*, 4, 1983.
- [GM02] M. Gamito and F. K. Musgrave. An accurate model of wave refraction over shallow water. *Computer and Graphics*, 26(2) :291–307, 2002.
- [Gon99] J.C Gonzato. *Modélisation de scènes océaniques*. PhD thesis, Université de Bordeaux 1, 1999.
- [GS00] J. C. Gonzato and B. Le Saec. On modeling and rendering ocean scenes. *Journal of Visualisation and Computer Simulation*, 11 :27–37, 2000.
- [Ha73] K. Hasselmann and al. Measurements of wind-wave growth and swell decay during the joint north sea wave project (jonswap). *Deutsche Hydr. Zeitschrift*, A8(12), 1973.
- [Har96] J.C Hart. Sphere tracing : A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10) :527–545, 1996.
- [HHK07] W. Hong, D.H. House, and J. Keyser. Adaptive particles for incompressible fluid simulation. Technical Report tamu-cs-tr-2007-6-1, Department of Computer Science, Texas A&M University, 2007.
- [HMT01] S. Hadap and N. Magnenat-Thalmann. Modeling dynamic hair as a continuum. *Computer Graphics Forum*, 20(3) :329–338, 2001.
- [HNC02] Hinsingert, Neyret, and Cani. Interactive animation of ocean waves. *ACM Symposium on Computer Animation*, pages 161–166, 2002.

- [HVT<sup>+</sup>06] Y. Hu, L. Velho, X. Tong, B. Guo, and H. Shum. Realistic, real-time rendering of ocean waves. *Computer Animation and Virtual Worlds*, 17(1) :59–67, 2006.
- [HW65] F. Harlow and J. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8 :2182–2189, 1965.
- [HW96] J.C Hart and S.P Worley. Hyper-rendering of hyper-textured surfaces. *Implicit Surfaces*, pages 99–104, 1996.
- [HW04] Holmberg and Wunsche. Efficient modeling and rendering of turbulent water over natural terrain. *Computer Graphics*, pages 15–22, 2004.
- [IDN02] K. Iwasaki, Y. Dobashi, and T. Nishita. An efficient method for rendering underwater optical effects using graphics hardware. *Computer Graphics Forum*, 21(4) :701–711, 2002.
- [IDN03] K. Iwasaki, Y. Dobashi, and T. Nishita. A volume rendering approach for sea surfaces taking into account second order scattering using scattering maps. In *VG '03 : Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics*, pages 129–136, New York, NY, USA, 2003. ACM.
- [Igl04] A. Iglesias. Computer graphics for water modeling and rendering : a survey. *Future Generation Computer Systems*, 20(8), 2004.
- [IVB02] J. R. Isidoro, A. Vlachos, and C. Brennan. Rendering ocean water. *Direct3D ShaderX : Vertex and Pixel Shader Tips and Tricks*, Wordware, 2002.
- [JBS03] S. Jeschke, H. Birkholz, and H. Shumann. A procedural model for interactive animation of breaking ocean waves. *Computer Graphics*, 2003.
- [Jer76] N.G Jerlov. *Marine Optics*. Elsevier, 1976.
- [JG01] L.S. Jensen and R. Goliás. Deep-water animation and rendering. <http://www.gamasutra.com/gdce/2001/jensen>, 2001.
- [KAG<sup>+</sup>05] R. Keiser, B. Adams, D. Grasser, P. Bazzi, P. Dutré, and M. Gross. A unified lagrangian approach to solid-liquid animation. *Eurographics Symposium on Computer Animation*, 2005.
- [KCC<sup>+</sup>06] J. Kim, D. Cha, B. Chang, B. Koo, and I. Ihm. Practical animation of turbulent splashing water. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 335–344, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

- [Kel06] M. Kelager. Lagrangian fluid dynamics using smoothed particle hydrodynamics. Technical report, Department of Computer Science, University of Copenhagen, 2006.
- [KH84] J. T. Kajiya and B. P Von Herzen. Ray tracing volume densities. *SIGGRAPH Comput. Graph.*, 18(3) :165–174, 1984.
- [KM90] M. Kass and G. Miller. Rapid, stable fluid dynamics for computer graphics. In *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 49–57, New York, NY, USA, 1990. ACM.
- [KO96] S. Koshizuka and Y. Oka. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science Engineering*, (123) :421–434, 1996.
- [Koe84] P. Koepke. Effective reflectance of oceanic whitecaps. *Appl. Optics*, 23 :1816–1824, 1984.
- [Kry05] Y. Kryachko. Using vertex texture displacement for realistic water rendering. *GPU Gems 2*, pages 283–294, 2005.
- [KW06] P. Kipfer and R. Westermann. Realistic and interactive simulation of rivers. In *GI '06 : Proceedings of Graphics Interface 2006*, pages 41–48, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.
- [LC87] W. E. Lorensen and H. E. Cline. Marching cubes : A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4) :163–169, 1987.
- [LGF04] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. In *SIGGRAPH '04 : ACM SIGGRAPH 2004 Papers*, pages 457–462, New York, NY, USA, 2004. ACM.
- [LL77] Lucy and B. Leon. A numerical approach to the testing of the fission hypothesis. *Astron Journ.*, 82(12) :1013–1924, December 1977.
- [LSSF06] F. Losasso, T. Shinar, A. Selle, and R. Fedkiw. Multiple interacting liquids. *ACM Trans. Graph.*, 25(3) :812–819, 2006.
- [LTKF08] F. Losasso, J. Talton, N. Kwatra, and R. Fedkiw. Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics*, 14(4) :797–804, 2008.
- [Max81] N. L. Max. Vectorized procedural models for natural terrain : Waves and islands in the sunset. In *SIGGRAPH '81 : Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, pages 317–324, New York, NY, USA, 1981. ACM.



- [MCG03] M. Müller, P. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. *In Proceedings of 2003 ACM SIGGRAPH Symposium on Computer Animation*, pages 154–159, 2003.
- [Mit05] Jason L. Mitchell. Real-time synthesis and rendering of ocean water. Technical report, ATI Research, 2005.
- [MKH05] T. Nishita T. Naemura M. Kakimoto, K. Matsuoka and H. Harashima. Glare generation based on wave optics. *Computer Graphics*, 24(2) :185–193, 2005.
- [MKN<sup>+</sup>04] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2004.
- [MM86] E. Monahan and G. MacNiocail. *Oceanic Whitecaps : Their Role in Air Sea Exchange Processes*. D. Reidel, 1986.
- [MMS04] V. Mihalef, D. Metaxas, and M. Sussman. Animation and control of breaking waves. *In SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 315–324, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
- [Mon92] J. Monaghan. Smoothed particle hydrodynamics. *Ann. Rev. Astron. Astrophys.*, 30 :543–574, 1992.
- [Mor91] A. Morel. Light and marine photosynthesis : a spectral model with geochemical and climatological implications. *Prog. Oceanogr.*, 63 :199, 1991.
- [Mor99] J. Morris. Simulating surface tension with smoothed particle hydrodynamics. *Int. J. Numer Meth. Fluids*, 33 :333–353, 1999.
- [MP89] G. Miller and A. Pearce. Globular dynamics : a connected particle system for animating viscous fluids. *Computer Graphics 13*, 3 :305–309, 1989.
- [MSKG05] M. Müller, B. Solenthaler, R. Keiser, and M. Gross. Particle-based fluid-fluid interaction. *In Proceedings on Symposium on Computer Animation*, pages 237–244, 2005.
- [MST04] M. Müller, S. Schirm, and M. Teschner. Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics. *Technol. Health Care*, 12(1) :25–31, 2004.
- [Mun47] W.H. Munk. A critical wind speed for air-sea boundary processes. *Journal of Marine Research*, 6 :203, 1947.

- [MWM87] G. A. Mastin, P. A. Watterberg, and J. F. Mareda. Fourier synthesis of ocean scenes. *IEEE Comput. Graph. Appl.*, 7(3) :16–23, 1987.
- [NKON90] E. Nakamae, K. Kaneda, T. Okamoto, and T. Nishita. A lighting model aiming at drive simulators. *Computer Graphics (Siggraph '90 proceedings)*, pages 395–404, 1990.
- [OH95] J. F. O'Brien and J. K. Hodgins. Dynamic simulation of splashing fluids. In *Computer Animation '95*, pages 198–205, 1995.
- [OS88] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed : Algorithms based on hamilton-jacobi formulations. *J. Comp. Phys*, 79 :12–49, 1988.
- [PA01] S. Premoze and M. Ashikhmin. Rendering natural waters. *Comput. Graph. Forum*, 20(4) :189–199, 2001.
- [PCPS97] F. Perez-Cazorla, X. Pueyo, and F. Sillion. Global illumination techniques for the simulation of participating media. In *Proceedings of the Eighth Eurographics Workshop on Rendering*, Saint Etienne, France, June 1997. <http://ima.udg.es/frederic/publications.html>.
- [Pea86] R. Darwin Peachey. Modeling waves and surf. *ACM Computer Graphics*, 20(4) :65–74, 1986.
- [Per85] K. Perlin. An image synthesiser. *Computer Graphics*, 19(3) :287–296, 1985.
- [PM64] W.J Pierson and L. Moskowitz. A proposed spectral form for fully developed wind seas based on similarity theory of s.a kilaigorodskii. *Journal of Geophysical Research*, pages 5281–5190, 1964.
- [PTB<sup>+</sup>03] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, and R. T. Whitaker. Particle-based simulation of fluids. In *Proceedings of Eurographics 2003*, pages 401–410, 2003.
- [Ree83] W.T. Reeves. Particle systems : A technique for modeling a class of fuzzy objects. In *SIGGRAPH '83 : Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 359–375, New York, NY, USA, 1983. ACM.
- [RO98] R. Radovitzky and M. Ortiz. lagrangian finite element analysis of newtonian fluid flows. *Int J. Numer. Meth. Engng*, 43 :607–619, 1998.
- [Rok93] P. Rokita. A model for rendering highintensity lights. *Computer Graphics*, 17(4) :431–437, 1993.
- [SAC<sup>+</sup>99] D. Stora, P. O. Agliati, M. P. Cani, F. Neyret, and J. D. Gascuel. Animating lava flows. In I. Scott Mackenzie and James Stewart, editors, *Gra-*

- phics Interface, GI'99, June, 1999*, pages 203–210, Kingston, Ontario, Canada, 1999. Canadian Human-Computer Communications Society.
- [Sch94] C. Schlick. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum*, 13(3) :233–246, 1994.
- [SF79] E. P Schettle and W. Fenn. Models for the aerosols for the lower atmosphere and the effects of humidity variations on their optical properties. AFGL-TR-79-0214 676, Environmental Research Papers, 1979.
- [SF95] J. Stam and E. Fiume. Depicting fire and other gaseous phenomena using diffusion processes. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 129–136, New York, NY, USA, 1995. ACM.
- [SSP06] B. Solenthaler, J. Schläfli, and R. Pajarola. From fluid to solid and back in a unified particle model. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation - Posters and Demos*, pages 13–14, 2006.
- [SST89] M. Shinya, T. Saito, and T. Takahashi. Rendering techniques for transparent objects. *Graphics Interface '89*, pages 173–181, 1989.
- [SSZG95] G. Spencer, P. Shirley, K. Zimmerman, and D.P. Greenberg. Physically-based glare effects for digital images. *Computer Graphics (Siggraph '95 proceedings)*, 29 :325–334, 1995.
- [SW01] J. Schneider and R. Westermann. Towards real-time visual simulation of water surfaces. *The journal of Vision, Modeling, and Visualization*, 2001.
- [TB87] P. Y. Ts'o and B. A. Barsky. Modeling and rendering waves : wave-tracing using beta-splines and reflective and refractive texture mapping. *ACM Trans. Graph.*, 6(3) :191–214, 1987.
- [TDG00] S. Thon, JM. Dischler, and D. Ghazanfarpour. Ocean waves synthesis using a spectrum-based turbulence function. In *Computer Graphics International*, 2000.
- [Tes01] J. Tessendorf. Simulating ocean waters. 2001.
- [TFK<sup>+</sup>03] T. Takahashi, H. Fujii, A. Kunitatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki. Realistic animation of fluid with splash and foam. *Computer Graphics Forum (Eurographics Proceedings)*, 22(3) :391–400, 2003.
- [Tho01] S. Thon. *Representation de l'eau en synthèse d'images : exemples de grandes étendues d'eau*. PhD thesis, Université de Limoges, Limoges, 2001.

- [TKRP06] N. Thürey, R. Keiser, U. Rüdè, and M. Pauly. "detail-preserving fluid control". *Proceedings of the 2006 Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 7–15, 2006.
- [TO99] G. Turk and J. F. O'Brien. Shape transformation using variational implicit functions. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 335–342, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [Ton91] D. Tonnesen. Modeling liquids and solids using thermal particles. In *Graphics Interface*, pages 255–262, 1991.
- [TPF89] D. Terzopoulos, J. Platt, and K. Fleischer. Heating and melting deformable models. In *Graphics Interface*, pages 219–226, 1989.
- [TRS06] N. Thürey, U. Rüdè, and M. Stamminger. Animation of open water phenomena with coupled shallow water and free surface simulations. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 157–164, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [TSS<sup>+</sup>07] N. Thürey, F. Sadlo, S. Schirm, M. Müller-Fischer, and M. Gross. Real-time simulations of bubbles and foam within a shallow water framework. In *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 191–198, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [Whi80] T. Whitted. An improved illumination model for shaded display. In *Commun. ACM*, volume 23, pages 343–349, 1980.
- [WMT05] H. Wang, P. J. Mucha, and G. Turk. Water drops on surfaces. *ACM Trans. Graph.*, 24(3) :921–929, 2005.
- [WZC<sup>+</sup>06] Q. Wang, Y. Zheng, C. Chen, T. Fujimoto, and N. Chiba. Efficient rendering of breaking waves using mps method. *Journal of Zhejiang University SCIENCE A*, 7(6) :1018–1025, 2006.
- [ZYP06] W. Zheng, J. H. Yong, and J. C. Paul. Simulation of bubbles. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 325–333, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.