Université de Montréal

Extension d'un modèle d'eau peu profonde à des dynamiques de fluides complexes

par
Paul Godbert

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Juillet, 2026

© Paul Godbert, 2026.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

Extension d'un modèle d'eau peu profonde à des dynamiques de fluides complexes

présenté par:

Paul Godbert

a été évalué par un jury composé des personnes suivantes:

Pr. Pierre Poulin, Université de Montréal	directeur de recherche
M. Arnaud Schoentgen, La Forge, Ubisoft Montréal	codirecteur
Pr. Benoit Baudry, Université de Montréal	membre du jury

Mémoire accepté le: 12 Août 2026

RÉSUMÉ

Au cours des dernières années, l'augmentation de la puissance de calcul des GPU a permis l'apparition d'effets graphiques de plus en plus poussés dans les jeux vidéo. Ce gain de performance a notamment favorisé l'intégration de simulations physiques en temps réel dans certaines productions à gros budget, contribuant à des environnements plus riches, dynamiques et immersifs. Dans la continuité de ces avancées, nous présentons un système de simulation d'eau en temps réel destiné à l'utilisation dans les jeux vidéo.

Basé sur un modèle de *Shallow Water*, notre système permet la gestion de différents types d'environnements aquatiques, incluant rivières, régions côtières ainsi qu'océans ouverts. Bien qu'intéressant par son faible coût et sa capacité à prendre en compte le terrain, ce modèle de base est limité, tant par les comportements qu'il peut modéliser que par son niveau de détails. Afin d'améliorer le réalisme visuel de la simulation, nous intégrons plusieurs extensions proposées dans la littérature en infographie. Ces améliorations comprennent notamment différentes représentations de l'écume, l'ajout de détails plus petits qu'un texel de simulation ainsi que la simulation de vagues déferlantes. L'ensemble de ces composantes contribue à reproduire une grande diversité de phénomènes observables dans les environnements marins et fluviaux réels.

L'implémentation repose sur l'utilisation intensive de nuanceurs de calcul exécutés sur GPU, permettant d'effectuer l'intégralité de la simulation sans transfert coûteux de données entre CPU et GPU. Cette approche favorise une forte parallélisation des calculs et permet l'intégration de nombreuses composantes de simulation et de rendu au sein d'un même système temps réel. Afin d'évaluer la viabilité de cette approche, nous présentons une analyse expérimentale des performances du système mis en place dans plusieurs scénarios représentatifs. Les résultats obtenus montrent qu'il est possible de reproduire une large gamme de phénomènes tout en maintenant des performances compatibles avec les contraintes des productions de jeu.

Mots-clés : Infographie, Simulation physique, Équations d'eau peu profonde, Temps réel

ABSTRACT

Over the course of the past few years, the rapid increase in the computing power of GPUs has enabled the emergence of increasingly sophisticated graphical effects in video games. This gain in performance has notably facilitated the integration of real-time physical simulations in certain high-budget productions, thereby contributing to the creation of richer, more dynamic, and more immersive environments. In continuity with these advancements, we present a real-time water simulation system intended for use in video games.

Built upon a Shallow Water model, our system supports the simulation of various types of aquatic environments, including rivers, coastal regions, and open oceans. Although this model is interesting due to its low simulation cost and its capability to handle terrains, it is limited both in the phenomena it can represent and in its level of detail. In order to enhance the visual realism of the simulation, we integrate several extensions proposed in the computer graphics literature. These improvements notably include different foam representations, the addition of surface details smaller than a simulation texel, as well as the simulation of breaking waves. Together, these components contribute to reproducing a wide variety of phenomena observable in real-world marine and fluvial environments.

The implementation relies on the intensive use of compute shaders executed on GPU, enabling the entire simulation to be carried out without costly data transfers between CPU and GPU. This approach promotes a high degree of parallelism and allows the integration of numerous simulation and rendering components within a single real-time system. To evaluate the viability of this approach, we present an experimental performance analysis of the system applied to several representative scenarios. The results demonstrate that it is possible to reproduce a wide range of phenomena while maintaining performance levels compatible with the constraints of modern video game applications.

Keywords : Computer Graphics, Physics Simulation, Shallow Water Equations, Real Time

TABLE DES MATIÈRES

RÉSUMÉ . . . . . **iii**

ABSTRACT . . . . . **iv**

TABLE DES MATIÈRES . . . . . **v**

LISTE DES TABLEAUX . . . . . **viii**

LISTE DES FIGURES . . . . . **ix**

LISTE DES ANNEXES . . . . . **xiii**

LISTE DES ABRÉVIATIONS . . . . . **xiv**

REMERCIEMENTS . . . . . **xv**

CHAPITRE 1 : INTRODUCTION **1**

1.1 Contexte 1

1.2 Ubisoft et La Forge 1

1.3 Simulation de fluides en temps réel 2

CHAPITRE 2 : CADRE THÉORIQUE **4**

2.1 Équations de Navier-Stokes 4

2.2 Équations d'eau peu profonde 5

2.2.1 Dérivations 6

2.2.2 Limitations 9

2.3 Résolution numérique 9

2.3.1 Grille MAC 9

2.3.2 *Operator splitting* 11

CHAPITRE 3 : TRAVAUX CONNEXES 14
3.1 Méthodes Eulériennes 14
3.1.1 Simulation en temps réel de vastes étendues d'eau avec des détails fins 14
3.1.2 Rendu et simulation d'eau en temps réel dans un monde ouvert . 17
3.1.3 Vagues déferlantes en temps réel pour modèle d'eau peu profonde 18
3.1.4 Généralisation de la simulation d'eau peu profonde avec des ondes de surface dispersives 21
3.2 Méthodes Lagrangiennes 21
3.2.1 Simulation d'eau peu profonde à base de SPH 21
3.2.2 Simulation d'eau peu profonde à base de particules pour des domaines irréguliers 23
3.2.3 Simulation d'eau peu profonde à base de particules avec éclaboussures et vagues déferlantes 24
3.3 Méthodes hybrides 26
3.3.1 Animation de l'eau utilisant l'équation d'onde, couplée à une simulation SPH 26
3.4 Conclusion 27

CHAPITRE 4 : SYSTÈME PROPOSÉ 29
4.1 Solveur de base 29
4.2 Améliorations du solveur 30
4.2.1 Améliorations de stabilité 30
4.2.2 Gestion des obstacles internes à la simulation 31
4.2.3 Bordures absorbantes 33
4.2.4 Schéma d'advection MacCormack 34
4.3 Advection de détails 36
4.4 Système de particules GPU 39
4.4.1 Système de base 39
4.4.2 Couplage bidirectionnel entre particules et grille 40

4.5 Vagues déferlantes 42
4.5.1 Particules de *splash* et de *spray* 42
4.5.2 Détection de vagues 43
4.5.3 Instanciation des particules 44
4.5.4 Rendu des particules 45
4.6 Rendu de la surface 51
4.7 Écume 51
4.7.1 Pseudo-simulation d'écume sur grille 52
4.7.2 Particules d'écume 55
4.7.3 Couplage des deux représentations 56

CHAPITRE 5 : RÉSULTATS 59

CHAPITRE 6 : CONCLUSION 62

BIBLIOGRAPHIE 65

LISTE DES TABLEAUX

5.I Caractéristiques des scènes de test. 59
5.II Comparaison des performances du système sur différentes scènes. 61

LISTE DES FIGURES

2.1 Grille MAC 2D et ses variables. 10

2.2 Méthode semi-Lagrangienne. L'étape de retour en arrière est ici faite en ligne droite sur un seul pas de temps Δt, sans subdivision. La valeur de la grandeur transportée est ensuite interpolée bilinéairement à partir des valeurs des quatre cellules voisines. 11

2.3 Schéma *Upwind*. 13

3.1 a) Avant réduction d'*overshoot* ; b) Après réduction d'*overshoot*. Réduction des pics visibles sur l'image de gauche près des rochers. Image extraite de l'article de Chentanez et Müller, 2010 [2]. . . . 15

3.2 Gauche : simulation d'eau peu profonde sans détails ; Milieu : détails de hautes fréquences et poids constant ; Droite : détails de hautes fréquences et poids variables. Image extraite de l'article de Chentanez et Müller, 2010 [2]. 16

3.3 Gauche : cascades avec des particules générées en haut des plateaux des cascades ; Droite : zone de création des particules (boîte rouge). Image extraite de l'article de Chentanez et Müller, 2010 [2]. 17

3.4 Texture d'écume rendue à la surface d'une rivière. Image extraite de l'article de Mao et Wu, 2017 [10]. 18

3.5 Construction de la courbe formant le front d'une vague. Image extraite de l'article de Thürey et al., 2007 [13]. 19

3.6 Génération géométrique d'une vague. Image extraite de l'article de Thürey et al., 2007 [13]. 20

3.7 Champ de densité et noyaux de lissages correspondants en dessous des images - Gauche : non normalisé, bosses à la surface ; Milieu : normalisé, surface lisse mais hauteurs incorrectes aux bordures ; Droite : interpolé. Image extraite de l'article de Solenthaler et al., 2011 [12]. 23

3.8 Classification des trois types de particules 3D. Image extraite de l'article de Fujisawa et al., 2017 [5]. 25
3.9 Évolution d'une vague en fonction du temps. Image extraite de l'article de Fujisawa et al., 2017 [5]. 25
3.10 SPH couplé avec Équation d'onde. Image extraite de l'article de Ramakrishnan et al., 2023 [11]. 27

4.1 Schéma de notre solveur de base, encodé sur une texture. 29
4.2 Gestion des obstacles internes - Gauche : sans | Droite : avec. Les faces noires correspondent à un artefact du *shadow map*, cela n'affecte pas notre technique. 32
4.3 Évolution d'une onde (vague) dans un domaine possédant deux bordures absorbantes gauche-droite et deux bordures réfléchissantes haut-bas. On peut observer la qualité de l'absorption de la vague sur les côtés gauches et droits alors que la vague poursuit son parcours de haut en bas. 34
4.4 Schéma d'advection MacCormack. 35
4.5 Gauche : Semi-Lagrangien | Droite : MacCormack. Ajout de détails fins, mais aussi quelques instabilités près du centre du vortex. 36
4.6 Impact du paramètre Ω sur l'apparence des détails de surface. . . 38
4.7 Rivière sans détails (Gauche) et avec détails d'écoulement (Droite). 39
4.8 Système d'accumulation des particules illustré en 2D. Les *i* encerclés représentent des particules dans la simulation, tandis que les *i* encadrés en noir représentent les indices de particules réinjectées dans la cellule correspondante. La particule 6 n'est pas en contact avec sa colonne d'eau et n'est donc pas réinjectée. 42
4.9 Zone d'émission des particules de *splash* et de *spray* (en rouge). . 44
4.10 Particules de *spray* projetées par une vague déferlante. À noter l'étirement des particules selon leur vitesse, ce qui accentue l'impression de mouvement. 46

4.11 Carte de profondeur des particules de *splash*. Le niveau de gris indique la profondeur de la sphère correspondante. 47
4.12 Carte d'épaisseur des particules de *splash* selon l'écran. Le niveau de gris est associé à la profondeur. Plus blanc signifie plus épais. . 47
4.13 Gauche : carte de profondeur floutée | Droite : carte d'épaisseur floutée. A noter que les silhouettes demeurent nettes alors que l'intérieur devient plus lisse. 48
4.14 Normales reconstruites. Les variations de couleur sont associées à la direction de la normale. Dans cet encodage, le violet correspond à une normale pointant vers le haut, tandis que le rouge et le vert représentent les directions horizontales. 49
4.15 Évolution d'une vague déferlante en quatre temps, de gauche à droite, de haut en bas. Dans un premier temps, on observe l'écume résiduelle de la vague antérieure. Dans un deuxième temps, on distingue la disparition de l'écume ainsi que la formation d'une nouvelle vague déferlante. Dans un troisième temps, la vague déferlante progresse avec un mouvement balistique. Finalement, la vague s'échoue sur le rivage, créant à nouveau de l'écume. 51
4.16 Rendu final du volume d'eau incluant réflexion, réfraction et absorption. 52
4.17 Texture de pseudo-simulation de l'écume sur une plage sur laquelle se trouvent des rochers et les poteaux d'un ponton. Des vagues viennent s'y écraser, créant de l'écume. Les zones blanches correspondent à une forte concentration d'écume, tandis que les zones noires indiquent l'absence d'écume. 54
4.18 Résultat de la technique d'écume sur grille dans un courant d'eau rapide provenant d'une chute. 55
4.19 Particules d'écume à la surface de l'eau suite à une vague déferlante s'écrasant sur la plage et les rochers. 56

4.20 Comparaison entre la pseudo-simulation d'écume de base et la version couplée avec les particules d'écume sur une scène de plage avec des rochers et un ponton. Gauche : pseudo-simulation sans couplage | Droite : pseudo-simulation avec couplage. 57

5.1 Comparaison entre la simulation de base et le système complet sur une scène représentant une plage. 59

5.2 Impact du système de particules sur la représentation des vagues déferlantes et des éclaboussures. 60

5.3 Comparaison entre la simulation de base et le système complet sur une scène représentant une rivière avec une cascade. 60

LISTE DES ANNEXES

Annexe I : Dérivation de la variation de la hauteur de l'eau dans le contexte des équations de *Shallow Water* . . . . . **xvi**

LISTE DES ABRÉVIATIONS

CPU	Processeur (***C**entral **P**rocessing **U**nit*)
FFT	Transformée de Fourier rapide (***F**ast **F**ourier **T**ransform*)
GPU	Carte graphique (***G**raphics **P**rocessing **U**nit*)
MAC	***M**arker-**A**nd-**C**ell*
SPH	***S**moothed **P**articles **H**ydrodynamics*
SWE	Équation d'eau peu profonde (***S**hallow **W**ater **E**quations*)

REMERCIEMENTS

Ce projet est le fruit de plus d'une année de travail. Il n'aurait pu voir le jour sans le soutien, les conseils et la confiance de nombreuses personnes, auxquelles je souhaite exprimer ma profonde gratitude.

Je tiens tout d'abord à remercier mon directeur de maîtrise, Pr. Pierre Poulin, pour ses précieux conseils tout au long de ce projet. Ses perspectives complémentaires, tant sur les aspects scientifiques que pratiques, ont grandement contribué à orienter mes réflexions et à enrichir la qualité de ce travail.

J'aimerais également remercier mon codirecteur Arnaud Schoentgen ainsi que Carlos Caicedo, mon gestionnaire, tous les deux chez Ubisoft La Forge. Leur mentorat, leur confiance et leurs nombreux conseils ont joué un rôle déterminant dans mon développement professionnel. Travailler à leurs côtés m'a permis de grandir en tant qu'ingénieur et d'aborder les défis de ce projet avec davantage de rigueur et de confiance.

Je tiens également à remercier mes collègues à Ubisoft La Forge pour leur soutien, leur expertise et les nombreuses discussions enrichissantes qui ont accompagné mon passage à La Forge. Ce fut un privilège de travailler à leurs côtés.

Ce projet n'aurait pas non plus été possible sans le soutien d'Ubisoft, MITACS et le fonds de recherche CRSNG Découverte qui ont financé cette initiative et m'ont offert l'opportunité de me consacrer à temps plein à ces travaux. Je leur suis reconnaissant pour la confiance qu'ils ont accordée à ce projet et pour leur engagement envers la recherche.

Enfin, je ne pourrais terminer sans remercier ma partenaire, ma famille et mes amis pour leur soutien indéfectible. Leur capacité à me pousser à donner le meilleur de moi-même a été essentielle à l'aboutissement de ce mémoire. Je leur suis profondément reconnaissant de m'avoir accompagné à chaque étape de ce parcours.

CHAPITRE 1

INTRODUCTION

1.1 Contexte

Ce mémoire présente les travaux réalisés dans le cadre d'un stage de maîtrise d'un an effectué à l'Université de Montréal en collaboration avec Ubisoft. Ce projet s'inscrit dans le cadre d'un partenariat entre le milieu académique et l'industrie visant à explorer de nouvelles approches de simulation de liquides adaptées aux contraintes des applications interactives.

Les travaux présentés dans ce document ont été réalisés au sein du laboratoire de recherche La Forge, où chercheurs universitaires et professionnels de l'industrie collaborent sur des projets de recherche appliquée. L'objectif de ce stage était d'étudier et de développer des techniques permettant d'étendre les capacités d'un modèle de simulation d'eau en temps réel basé sur les équations d'eau peu profonde (ou *Shallow Water Equations*).

1.2 Ubisoft et La Forge

Fondée en 1986, Ubisoft est l'une des plus importantes entreprises de développement et d'édition de jeux vidéo au monde. L'entreprise possède de nombreux studios répartis à l'international et développe des technologies utilisées dans une grande variété de productions interactives.

Afin de soutenir ses activités de recherche et développement, Ubisoft a créé en 2016 le laboratoire de recherche *La Forge*. Présentes au Canada, en France et en Chine, les équipes de La Forge ont pour mission d'explorer de nouvelles technologies à partir des avancées récentes de la recherche académique. Le laboratoire favorise la collaboration entre chercheurs universitaires et employés d'Ubisoft afin de concevoir des prototypes

dans des domaines tels que le rendu graphique, l'animation, l'intelligence artificielle, l'audio et la physique. Les connaissances acquises dans le cadre de ces projets contribuent à l'évolution des technologies utilisées dans les productions d'Ubisoft tout en alimentant la recherche académique.

1.3 Simulation de fluides en temps réel

Au cours des dernières années, l'augmentation de la puissance de calcul des GPU a permis l'intégration d'effets visuels et de simulations physiques de plus en plus complexes au sein des jeux vidéo. Cette évolution a notamment rendu possible l'exécution de modèles numériques auparavant trop coûteux pour une utilisation en temps réel. Ces avancées ont favorisé l'apparition d'environnements plus dynamiques, plus vastes et plus complexes dans lesquels les éléments physiques tels que les fluides, les destructions ou les interactions avec le terrain contribuent grandement à l'immersion du joueur.

Parmi les phénomènes physiques simulés en temps réel, l'eau représente un défi particulier en raison de la diversité des comportements qu'elle peut adopter. Une simulation réaliste doit être capable de représenter à la fois des mouvements de grande échelle, comme les courants ou les vagues, et des phénomènes plus fins tels que les éclaboussures ou les interactions locales avec l'environnement. Les modèles basés sur les équations d'eau peu profonde constituent une approche couramment utilisée dans ce contexte puisqu'ils offrent un compromis intéressant entre comportement physique plausible, faible coût de calcul et interaction avec les terrains. En représentant le liquide sous la forme d'un champ de hauteur défini sur une grille bidimensionnelle, ces modèles permettent de simuler efficacement des phénomènes comme les vagues, les écoulements de rivière ou les variations du niveau d'eau, tout en conservant des performances compatibles avec une exécution en temps réel et un coût mémoire raisonnable.

Cependant, cette représentation simplifiée impose certaines contraintes. En considérant principalement une surface d'eau bidimensionnelle, les équations d'eau peu pro-

fonde ne permettent pas de représenter directement certains comportements d'eau fortement tridimensionnels tels que les cascades, les vagues déferlantes ou les projections d'eau. De plus, la résolution de la grille utilisée pour la simulation limite la quantité de détails pouvant être reproduits, ce qui peut entraîner une perte de richesse visuelle lorsque des interactions à petite échelle sont nécessaires.

L'objectif de ce projet est donc d'étendre les capacités d'un solveur d'eau peu profonde exécuté sur GPU afin d'améliorer sa stabilité, sa qualité visuelle et la diversité des phénomènes pouvant être représentés. Pour atteindre cet objectif, plusieurs modifications seront apportées au solveur existant afin d'améliorer son comportement et ses capacités de simulation. Une implémentation d'une approche hybride combinant une simulation principale sur grille avec un système de particules exécuté sur GPU a été réalisée. Cette combinaison permet de conserver l'efficacité des méthodes à base de champs de hauteur pour les phénomènes de grande échelle, tout en ajoutant une représentation plus flexible pour modéliser des scénarios dépassant les capacités du modèle d'eau peu profonde.

Ce projet s'inscrit dans le contexte du développement de techniques de simulation adaptées aux contraintes de la production vidéoludique moderne, où la recherche d'un équilibre entre qualité visuelle et performance demeure essentielle. L'objectif n'est donc pas uniquement d'augmenter la qualité visuelle du modèle, mais également de proposer une solution pouvant être intégrée dans un environnement temps réel et de s'adapter aux diverses contraintes imposées par les besoins des productions.

Le reste de ce mémoire est organisé comme suit. Le chapitre 2 présente les notions théoriques nécessaires à la compréhension du projet. Le chapitre 3 présente un état de l'art des différentes approches utilisées pour la simulation de l'eau à base de modèles d'eau peu profonde. Le chapitre 4 décrit le système proposé ainsi que les différentes améliorations apportées au solveur. Le chapitre 5 présente les résultats obtenus ainsi qu'une analyse des performances du système sur différentes scènes de test. Enfin, dans le chapitre 6 nous proposons une conclusion et discutons de futures pistes d'exploration.

CHAPITRE 2

CADRE THÉORIQUE

2.1 Équations de Navier-Stokes

La simulation d'un volume de fluide en temps réel repose généralement sur une modélisation physique de son comportement. Dans le cadre de modèles simplifiés, le fluide est souvent considéré comme incompressible et sa viscosité supposée constante. Comme pour tout système physique, l'évolution du mouvement du fluide est régie par la loi fondamentale de la dynamique :

$$m\vec{a} = \vec{F}.$$

Dans le cas d'un fluide, cette dynamique se traduit par un système d'équations différentielles appelé équations de Navier-Stokes :

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{f} + \nu \nabla^2 \vec{u}$$
$$\nabla \cdot \vec{u} = 0$$

où :

- $\vec{u}$ est le champ de vitesse du fluide, qui donne la vitesse en chaque point de l'espace et du temps.
- ρ est la densité du fluide, considérée comme constante pour les fluides incompressibles, typiquement $1000\,\mathrm{kg/m^3}$ pour l'eau.
- ∇p est le gradient de pression, avec p calculé de manière à satisfaire la condition d'incompressibilité.
- ν est la viscosité cinématique.
- $\vec{f}$ représente les forces extérieures, comme la gravité ou d'autres forces imposées.

— ∇^2 est l'opérateur différentiel Laplacien, qui mesure localement comment une valeur diffère de la moyenne de ses voisines.
— la dérivée $\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = \frac{D\vec{u}}{Dt}$ correspond à l'accélération en un point de l'espace, appelée dérivée matérielle ou particulaire.

La seconde équation, $\nabla \cdot \vec{u} = 0$, impose l'incompressibilité du fluide. Elle garantit qu'il n'y a ni création ni destruction de matière dans le volume simulé.

Souvent, le choix est fait de simplifier les équations de Navier-Stokes en négligeant la viscosité. Dans le cas spécifique d'une simulation d'un volume d'eau, il est possible de prendre cette liberté, car la viscosité de l'eau est faible ($1.01 \times 10^{-3} Pa.s$ pour l'eau, contre $7.72 \times 10^{-2} Pa.s$ pour le miel). Dans ce cas, l'effet de la viscosité est souvent considéré comme négligeable. Simplifier les équations précédentes en assumant aucune viscosité nous permet d'obtenir les équations d'Euler :

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{f}, \tag{2.1}$$
$$\nabla \cdot \vec{u} = 0.$$

Ces équations plus simples conservent l'essentiel de la dynamique du fluide tout en étant plus simples à résoudre numériquement, ce qui constitue un avantage important pour la simulation en temps réel.

2.2 Équations d'eau peu profonde

Les équations de Navier-Stokes sont des équations complexes à résoudre, encore plus dans un contexte de simulation en temps réel. Pour ces raisons, certaines méthodes proposent de réduire le domaine de simulation, passant d'un domaine en 3D à une simulation en 2D capable d'être résolue en temps réel. La plupart de ces méthodes choisissent de représenter le volume d'eau sous la forme d'un champ de hauteur avec $y = h(x,z)$ la hauteur d'une colonne d'eau au point (x,z). Cette dernière peut être retrouvée à partir de deux champs de hauteur : la hauteur du terrain et la profondeur de l'eau. Ces deux

quantités peuvent être annotées respectivement $b(x,z)$ et $d(x,z)$. Ainsi, on obtient que $h(x,z) = b(x,z) + d(x,z)$. Pour simuler le phénomène d'écoulement de l'eau, notre variable principale sera la profondeur de cette dernière, évoluant au cours du temps.

Plusieurs méthodes utilisent cette représentation. Selon le contexte, certaines peuvent paraître plus appropriées que d'autres. Dans le cadre de notre projet, notre intérêt se porte sur les équations dites d'eau peu profonde. Celles-ci reposent sur l'hypothèse que la profondeur de l'eau est très faible par rapport à la longueur d'onde des vagues. Cette approximation permet de négliger les variations verticales de la vitesse, en considérant que le mouvement du fluide est essentiellement horizontal. Le principal intérêt de cette méthode est le fait qu'elle supporte naturellement les interactions avec le terrain.

2.2.1 Dérivations

Afin de dériver les équations d'eau peu profonde à partir des équations de Navier-Stokes, il est possible de poser une série d'hypothèses visant à simplifier le modèle. Pour récapituler, les hypothèses admises sont les suivantes :

— Les vitesses sont essentiellement horizontales, la vitesse verticale quant à elle est négligée. Le liquide qui passe d'une colonne à la suivante sera alors répercuté immédiatement dans cette colonne.

— La topographie du terrain sur lequel repose le liquide est connue via un champ de hauteur $b(x,z)$. Les pentes de ce dernier sont faibles.

— La profondeur de l'eau est faible par rapport aux longueurs d'onde des vagues.

— Les accélérations dans le fluide sont faibles comparées à la gravité.

Une dernière hypothèse fondamentale que nous allons émettre est celle d'assumer une pression hydrostatique. En effet, en réécrivant l'équation du mouvement (2.1), on obtient :

$$\frac{\partial \vec{v}}{\partial t} + u\frac{\partial \vec{v}}{\partial x} + v\frac{\partial \vec{v}}{\partial y} + w\frac{\partial \vec{v}}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial y} = -g. \tag{2.2}$$

où $g \approx 9.81\text{m.s}^{-2}$ est l'accélération de la pesanteur terrestre. On assumera ici que, comme l'eau est peu profonde et relativement calme, les termes dominants sont alors le gradient de pression et la gravité. En négligeant les termes minoritaires, on obtient l'équation de la pression hydrostatique :

$$\frac{1}{\rho}\frac{\partial p}{\partial y} = -g.$$

En intégrant sur l'axe vertical Y et en posant les conditions $y = h$ et $p = 0$ (on se trouve à la surface et la pression y est considérée comme nulle), on obtient une expression de la pression au temps t ne dépendant que de y :

$$p(x,y,z,t) = \rho g(h(x,z,t) - y).$$

Nous obtenons ainsi une expression permettant d'approximer directement la pression nécessaire, sans avoir à résoudre un système linéaire complexe. Cette approximation constitue une des accélérations principales qui en font une méthode adaptée au temps réel.

Étant donné que nous avons déduit une expression explicite de la pression, nous pouvons injecter celle-ci dans les équations de Navier-Stokes, réduites aux deux composantes horizontales x et z. Dans l'équation (2.2), en supposant que u et w sont constantes sur y, alors $\frac{\partial u}{\partial y} = \frac{\partial w}{\partial y} = 0$. Les équations de quantité de mouvement deviennent alors :

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + w\frac{\partial u}{\partial z} + g\frac{\partial h}{\partial x} = 0,$$
$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + w\frac{\partial w}{\partial z} + g\frac{\partial h}{\partial z} = 0.$$

On observe alors que l'accélération est proportionnelle à la gravité et au gradient de la hauteur.

En ce qui concerne la vitesse verticale, bien que négligée dans les équations princi-

pales, il est possible de la reconstituer à partir des vitesses horizontales pour satisfaire la condition d'incompressibilité.

La dernière inconnue fondamentale du modèle est la hauteur d'eau $d(x,z,t)$, définie comme la distance verticale entre la surface libre de l'eau $h(x,z,t)$ et la topographie du terrain $b(x,z)$, qui ne change généralement pas dans le temps :

$$d(x,z,t) = h(x,z,t) - b(x,z).$$

Après dérivation (voir l'annexe I), on obtient l'expression suivante décrivant la variation de la hauteur d'eau :

$$\frac{\partial d}{\partial t} = -\nabla \cdot (d\vec{u}).$$

Cette équation exprime que toute variation locale de la profondeur d'eau résulte des flux horizontaux, permettant ainsi de déterminer facilement la hauteur de l'eau en un point du champ de hauteur.

Ainsi, les équations d'eau peu profonde peuvent être exprimées comme suit :

$$\frac{D\vec{u}}{Dt} = -g\nabla h + a^{\text{ext}},$$
$$\frac{Dh}{Dt} = -d\nabla \cdot \vec{u}.$$

Maintenant que les expressions de pression, de vitesses et de hauteur ont été établies, posons des conditions sur les bords du domaine de simulation. Pour un mur solide et statique, on pose que, pour $\hat{n}$, la normale en 2D du mur sur le plan XZ, on a :

$$(u,w) \cdot \hat{n} = 0.$$

Le gradient de la hauteur doit aussi être restreint :

$$\left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y}\right) \cdot \hat{n} = 0.$$

Cette technique fonctionne pour des bordures solides et réfléchissantes avec collisions élastiques sans friction.

2.2.2 Limitations

La réduction de la dimensionnalité du modèle limite évidemment son expressivité. Cela rend notamment le modèle incapable de simuler certains effets tels que les chutes d'eau, les éclaboussures, la cassure des vagues ou encore la formation de gouttes ou de jets d'eau. En pratique, ces hypothèses restreignent souvent l'utilisation des modèles d'eau peu profonde à la simulation d'environnements relativement calmes avec des terrains peu abrupts ou encore des vitesses d'écoulement d'eau relativement faibles.

2.3 Résolution numérique

Les équations de Navier-Stokes, et par extension les équations d'eau peu profonde, ne possèdent généralement pas de solutions analytiques explicites. Il est donc nécessaire d'adopter une approche numérique pour en simuler l'évolution. Pour cela, l'approche standard est une approche Eulérienne, particulièrement adaptée à l'utilisation d'un champ de hauteur.

2.3.1 Grille MAC

Le domaine de simulation est discrétisé à l'aide d'une grille de type *MAC* (*Marker-And-Cell*), qui stocke les variables scalaires comme la hauteur d'eau d ou la hauteur du terrain b au centre de chaque cellule, et les composantes de vitesse horizontales et verticales respectivement sur la face gauche et basse de chaque cellule. La figure 2.1 illustre cette organisation. Cette structure permet de calculer sans biais le gradient de p sur les bords des cellules de la grille.

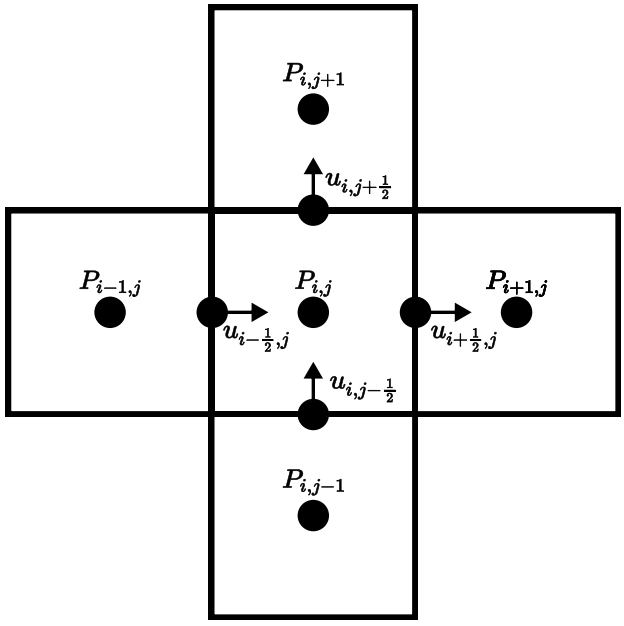


FIGURE 2.1 – Grille MAC 2D et ses variables.

Il est important de noter qu'il est possible de calculer facilement la vitesse du fluide au centre d'une cellule en interpolant les vitesses stockées sur les bords de cette dernière. Soit la cellule (i, j) :

$$\vec{u}_{i,j} = \left(\frac{u_{i-\frac{1}{2},j} + u_{i+\frac{1}{2},j}}{2}, \frac{v_{i-\frac{1}{2},j} + v_{i+\frac{1}{2},j}}{2} \right).$$

Finalement, cette structure permet également de calculer efficacement des dérivées partielles par différences finies :

$$\frac{\partial d_{i-\frac{1}{2},j}}{\partial x} = \frac{d_{i,j} - d_{i-1,j}}{\Delta x}.$$

La grille MAC constitue ainsi une structure robuste permettant de discrétiser notre simulation efficacement. De plus, cette grille peut être facilement encodée dans une texture, permettant d'être portée efficacement sur GPU. Dans le cadre de ce projet, nous considérons une grille constituée de cellules carrées, où $\Delta x = \Delta y$.

2.3.2 *Operator splitting*

Résoudre les équations d'eau peu profonde directement est une tâche complexe. Pour simplifier la tâche, il est possible de séparer ces équations complexes en plusieurs sous-équations, plus simples à résoudre et pour lesquelles il existe des algorithmes spécialisés. On fonctionne alors comme suit : Pour chaque pas de temps Δt, la simulation est divisée en étapes indépendantes, traitant séparément l'advection des vitesses, l'intégration des vitesses, la mise à jour des hauteurs de l'eau, et enfin l'application des conditions aux frontières. Cette approche modulaire permet une meilleure maîtrise de la stabilité numérique et facilite la compréhension du processus.

2.3.2.1 Advection

L'advection des quantités conservées est réalisée à l'aide d'un schéma semi-Lagrangien : Pour chaque cellule, on remonte dans le temps le long du champ de vitesse pour trouver la position au pas de simulation précédent d'une particule imaginaire ayant fini son trajet au centre de la cellule concernée. La valeur de la grandeur transportée est ensuite interpolée. Cela est reproduit dans la figure 2.2.

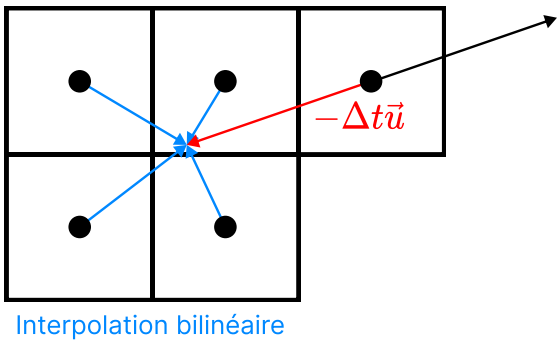


FIGURE 2.2 – Méthode semi-Lagrangienne. L'étape de retour en arrière est ici faite en ligne droite sur un seul pas de temps Δt, sans subdivision. La valeur de la grandeur transportée est ensuite interpolée bilinéairement à partir des valeurs des quatre cellules voisines.

Dans le cas d'un modèle d'eau peu profonde, cette méthode est utilisée pour transporter

le champ de vitesse par lui-même, un phénomène appelé *auto-advection*. La résolution numérique de ce terme reste une approximation du comportement physique réel. Le schéma semi-Lagrangien introduit notamment une dissipation numérique qui entraîne une perte progressive des variations à haute fréquence du champ simulé, provoquant ainsi un lissage des détails de la simulation ainsi qu'une perte d'énergie au cours du temps. L'importance de cette dissipation dépend de l'ordre spatial et temporel de la méthode d'interpolation utilisée lors de l'étape de rétro-projection. Une interpolation bicubique permet par exemple de mieux préserver les variations du champ qu'une interpolation bilinéaire, mais nécessite davantage d'accès mémoire et d'opérations, ce qui augmente son coût de calcul.

2.3.2.2 Intégration de la hauteur d'eau

La hauteur de l'eau d est mise à jour par une intégration explicite de son évolution temporelle sur un pas de temps Δt :

$$d_{i,j}^{t+1} = d_{i,j}^{t} + \frac{\partial d_{i,j}^{t}}{\partial t}\Delta t.$$

La dérivée temporelle est obtenue à partir de la forme discrète de l'équation de conservation de masse. Pour une cellule (i, j), elle s'écrit :

$$\frac{\partial d_{i,j}}{\partial t} = \frac{(u\bar{d})_{i+\frac{1}{2},j} - (u\bar{d})_{i-\frac{1}{2},j}}{\Delta x} + \frac{(w\bar{d})_{i,j+\frac{1}{2}} - (w\bar{d})_{i,j-\frac{1}{2}}}{\Delta z}. \tag{2.3}$$

Les termes de flux $(u\bar{d})$ et $(w\bar{d})$ représentent le transport de masse à travers le champ de vitesse. Pour évaluer la hauteur d'eau $\bar{d}$, on utilise un schéma *upwind* (figure 2.3) basé sur le signe de la vitesse. Par exemple, pour la face située en $i+\frac{1}{2}, j$ d'une cellule (i, j), on a :

$$(\bar{d})_{i+\frac{1}{2},j} = \begin{cases} d_{i,j} & \text{si } u_{i+\frac{1}{2},j} > 0, \\ d_{i+1,j} & \text{sinon.} \end{cases}$$

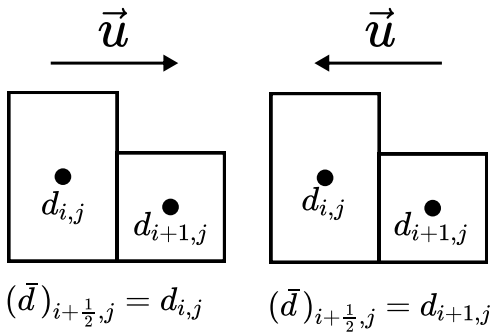


FIGURE 2.3 – Schéma *Upwind*.

Ce schéma permet de sélectionner la valeur située en amont du flux, limitant ainsi les oscillations numériques et respectant le sens physique du transport de matière.

2.3.2.3 Intégration des vitesses

Les vitesses sont mises à jour selon le gradient de hauteur et les accélérations extérieures :

$$u^{n+1}_{i+\frac{1}{2},j} = u^{n}_{i+\frac{1}{2},j} + \left(-g\frac{h_{i+1,j}+h_{i,j}}{\Delta x} + a_x^{\text{ext}}\right)\Delta t,$$

$$w^{n+1}_{i,j+\frac{1}{2}} = w^{n}_{i,j+\frac{1}{2}} + \left(-g\frac{h_{i,j+1}+h_{i,j}}{\Delta x} + a_z^{\text{ext}}\right)\Delta t.$$

Il est important de noter qu'étant données les contraintes de temps réel, l'utilisation d'un intégrateur explicite est de mise. Ce choix d'intégrateur apporte certes des avantages en termes de performance, mais est en général moins robuste que d'autres types d'intégrateurs tels que les intégrateurs semi-implicites ou implicites, car beaucoup plus soumis à des problèmes de stabilité numérique.

CHAPITRE 3

TRAVAUX CONNEXES

Dans le cadre de ce projet, nous proposons une revue de littérature consacrée aux méthodes de simulation de liquides dérivées du modèle d'eau peu profonde. Nous nous concentrons sur les méthodes adaptées à la simulation en temps réel, qui sont les plus pertinentes pour le rendu d'eau dans des jeux vidéo. Ce chapitre est divisé en trois parties, chacune abordant une famille de méthodes : les méthodes Eulériennes, discrétisant le volume d'eau sur une grille, les méthodes Lagrangiennes, utilisant des particules afin de modéliser le comportement du liquide et enfin les méthodes hybrides. À noter que les articles présentés dans ce chapitre ne sont pas classés par ordre chronologique, mais selon leur pertinence vis-à-vis de notre projet.

3.1 Méthodes Eulériennes

Les méthodes Eulériennes sont largement utilisées dans un contexte de simulation en temps réel, notamment en raison de leur stabilité numérique et de leur formulation bien adaptée à une implémentation GPU. Toutefois, leur précision dépend fortement de la résolution de la grille, ce qui introduit un compromis direct entre coût de calcul et qualité des détails simulés.

3.1.1 Simulation en temps réel de vastes étendues d'eau avec des détails fins

Publié en 2010, l'article *Real-time Simulation of Large Bodies of Water with Small Scale Details* de Chentanez et Müller [2] se fonde sur le solveur de base introduit au chapitre 2 et se concentre sur divers ajouts pour en augmenter les capacités :

1. Ajout de techniques pour améliorer la stabilité de l'intégration des vitesses et des hauteurs :

(a) Une condition CFL est introduite en limitant la norme de la vitesse $\|\vec{u}\|$ par $\alpha\frac{\Delta x}{\Delta t}$, avec $\alpha = 0.5$ dans cet article.

(b) La hauteur de l'eau est limitée pour l'intégration des hauteurs, en remplaçant $\bar{d}$ dans l'équation (2.3) par $\bar{d} - \max\left(0, \frac{\bar{d}_{i+1,j}+\bar{d}_{i-1,j}+\bar{d}_{i,j+1}+\bar{d}_{i,j-1}}{4} - d_{\text{avgmax}}\right)$, avec

$d_{\text{avgmax}} = \beta\frac{\Delta x}{g\Delta t}$, où β est un paramètre au choix, dans cet article $\beta = 2$.

(c) Pour réduire les problèmes d'*overshoot*, c'est-à-dire l'apparition de valeurs excessives non physiques, on peut détecter les vagues problématiques et réduire leur hauteur en conséquence. Par exemple, sur l'axe X au temps t :

— Si $h_{i,j} - h_{i-1,j} > \lambda_{\text{edge}}$ et $h_{i,j} > h_{i+1,j}$, alors
$d_{i,j}^{t+1} = d_{i,j}^{t} + \alpha_{\text{edge}}(\max\left(0, \frac{1}{2}(d_{i,j} + d_{i+1,j})\right) - d_{i,j}\big)$,

— Si $h_{i,j} - h_{i+1,j} > \lambda_{\text{edge}}$ et $h_{i,j} > h_{i-1,j}$, alors
$d_{i,j}^{t+1} = d_{i,j}^{t} + \alpha_{\text{edge}}(\max\left(0, \frac{1}{2}(d_{i,j} + d_{i-1,j})\right) - d_{i,j}\big)$.

Ces techniques permettent de limiter les problèmes d'instabilité sans trop affecter la fidélité du modèle, comme illustré dans la figure 3.1.

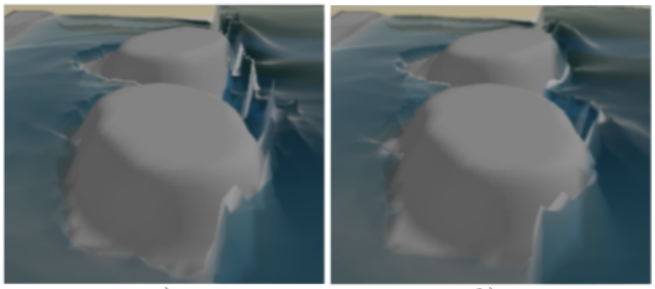


FIGURE 3.1 – a) Avant réduction d'*overshoot* ; b) Après réduction d'*overshoot*. Réduction des pics visibles sur l'image de gauche près des rochers. Image extraite de l'article de Chentanez et Müller, 2010 [2].

2. Afin de représenter des détails fins non capturés par la simulation (i.e., plus petits qu'une cellule de la grille MAC), un mélange de textures de détails est utilisé. Le transport de ces textures est modélisé par l'advection semi-Lagrangienne de trois ensembles de coordonnées de texture, régulièrement réinitialisées toutes les τ secondes avec un décalage de phase pour permettre un mélange plus lisse.

L'importance de chaque texture dans le mélange est pondérée par un facteur dépendant de l'étirement local des textures, calculé via la mesure de déformation de Green (*Green Strain*), afin d'atténuer les détails dans les zones très déformées. Le déplacement final est obtenu en combinant les trois textures avec des poids temporellement décalés pour une transition continue.

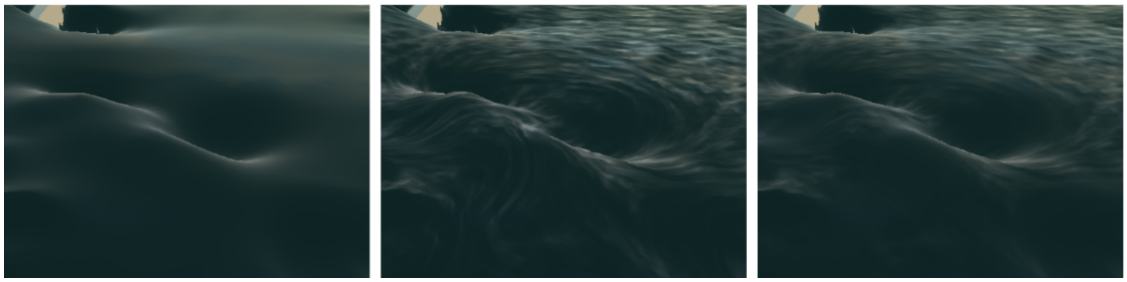


FIGURE 3.2 – Gauche : simulation d'eau peu profonde sans détails ; Milieu : détails de hautes fréquences et poids constant ; Droite : détails de hautes fréquences et poids variables. Image extraite de l'article de Chentanez et Müller, 2010 [2].

3. Pour permettre au modèle de prendre en compte les cas de discontinuités du terrain ou encore les cassures de vagues, le modèle est augmenté par l'ajout d'un système de particules couplé à la simulation 2D. Ces dernières sont instanciées à des endroits spécifiques tels que le haut d'une chute d'eau, ou la crête d'une vague en train de se briser. Ces particules dites de *splash* et de *spray* ont pour utilité le transport de quantités de fluide dans des moments où la simulation à base de champ de hauteur atteint ses limites. Ces particules sont illustrées dans la figure 3.3. Un autre type de particules, les particules de *foam*, répondent quant à elles à un but purement esthétique. Une fois instanciées, ces particules se voient assignées une vitesse initiale puis sont simulées selon un comportement balistique jusqu'à entrer de nouveau en collision avec le champ de hauteur, où la quantité transportée sera ajoutée au volume.
4. Des critères sont introduits pour reconnaître les vagues déferlantes, ce qui entraîne la génération de particules mais pas de déformation de la géométrie de la vague :
   (a) $\left|\nabla h_{i,j}\right| > \alpha_{\text{minSplash}} \frac{g\Delta x}{\Delta t}$: pente assez raide pour se briser.

(b) $\frac{d_{i,j}-d_{i,j}^{\text{prev}}}{\Delta t} > v_{\text{minSplash}}$: élévation rapide à l'avant de la vague.

(c) $\nabla^2 h_{i,j} < l_{\text{minSplash}}$: haut de la vague.

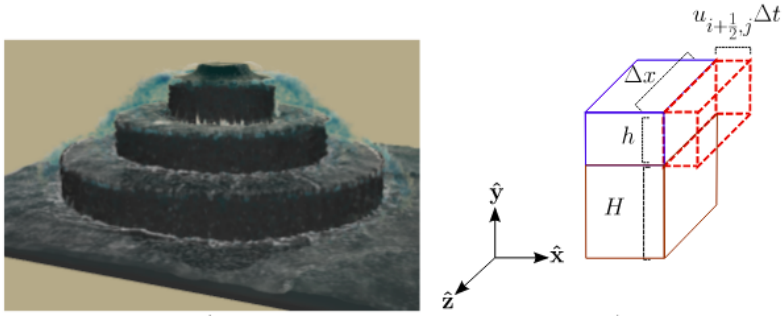


FIGURE 3.3 – Gauche : cascades avec des particules générées en haut des plateaux des cascades ; Droite : zone de création des particules (boîte rouge). Image extraite de l'article de Chentanez et Müller, 2010 [2].

3.1.2 Rendu et simulation d'eau en temps réel dans un monde ouvert

La présentation de 2017 *Open-World Water Rendering and Real-Time Simulation* par Mao et Wu [10] décrit un outil complet servant à l'intégration d'eau dans des mondes ouverts.

1. Cet article utilise la même méthode que Chentanez et Müller [2] pour la simulation en temps réel de vastes volumes d'eau, reprenant les idées de création de particules pour gérer les discontinuités du terrain comme des chutes d'eau.
2. La simulation d'écume sur la grille de simulation est basée sur deux critères :
   (a) $\phi_{\text{ta}} = \left\| \left(u_{i-\frac{1}{2},j+1} - u_{i-\frac{1}{2},j}, v_{i+1,j-\frac{1}{2}} - v_{i,j-\frac{1}{2}} \right) \right\|$: quantité d'air emprisonné dans une cellule.
   (b) $\phi_{\text{steep}} = \left\| \left(\frac{\partial H+h}{\partial x}, \frac{\partial H+h}{\partial y} \right) \right\|$: norme du gradient de la hauteur de l'eau.

   Ces deux critères permettent d'ajouter une certaine quantité d'écume dans une pseudo-simulation. Les quantités d'écume sont transportées par le champ de vitesse du fluide et se dissiperont au cours du temps. Cette pseudo-simulation permet de déterminer l'opacité de la texture en un point donné. Cette dernière est

ensuite transportée par le champ de vitesse de notre simulation d'eau. L'écume ainsi générée est facilement visible en figure 3.4.

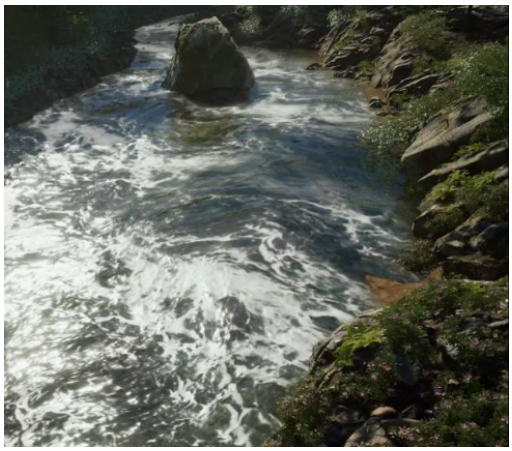


FIGURE 3.4 – Texture d'écume rendue à la surface d'une rivière. Image extraite de l'article de Mao et Wu, 2017 [10].

3. Enfin, cet article apporte une méthode de pré-calcul de champs de vitesse provenant d'une simulation physique utilisant la méthode *Lattice-Boltzmann*, plus précisément LBM-SWE [9]. Cette méthode permet de régler certains problèmes des champs de vitesse classiques. En effet, elle permet de représenter des flots turbulents, elle est hautement parallélisable et elle est facilement éditable par un artiste. Cette méthode n'est pas utilisée en conjonction avec le modèle d'eau peu profonde et ne sera donc pas discutée plus en profondeur.

3.1.3 Vagues déferlantes en temps réel pour modèle d'eau peu profonde

Dans l'article *Real-time Breaking Waves for Shallow Water Simulations* [13], Thürey et al. présentent une méthode permettant d'étendre les capacités visuelles du modèle d'eau peu profonde en y ajoutant une représentation polygonale de vagues déferlantes. Les différentes étapes sont :

1. Détection de régions où les vagues pourraient casser :

$$|\nabla H(x)| > t_H \text{ et } \nabla H(x) \cdot u(x) < 0$$

avec H la hauteur totale de l'eau et du terrain, $t_H = \frac{p_H g \Delta t}{\Delta x}$, et p_H un paramètre au choix, souvent $\frac{1}{4}$ dans cet article.

2. Ces régions génèrent une série de points représentant la courbe directrice de la vague. Les trous dans cette courbe sont complétés en étendant selon les points proches. Pour ce faire, on sélectionne un point au hasard qui répond aux critères, on suit la tangente du gradient de hauteur, dans un sens puis dans l'autre, en changeant de sens lorsque le point sur lequel on tombe ne fait pas partie de la région considérée comme déferlante. Si un point est ajouté sur le front de vague, on le recentre au point de la cellule selon le gradient le plus fort. La figure 3.5 illustre ce processus.

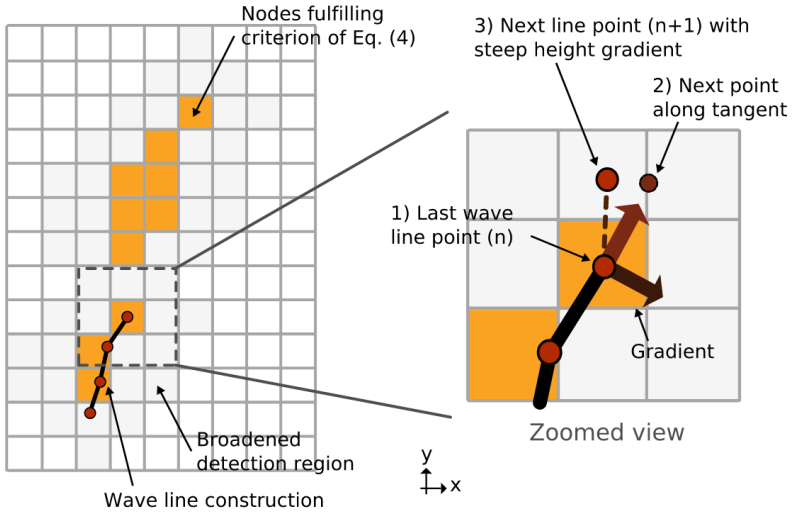


FIGURE 3.5 – Construction de la courbe formant le front d'une vague. Image extraite de l'article de Thürey et al., 2007 [13].

3. Les points sont ensuite transportés selon la pente du champ de hauteur précédent, à vitesse $c = \sqrt{gH}$, puis projetés vers la zone de plus forte pente pour suivre le front de vague. Si la distance entre l'ancien et le nouveau point est supérieure à $2c$, alors on retire le point de la courbe. Finalement, de nouveaux points sont insérés lorsque la distance entre deux points voisins dépasse $2\Delta x$. Les points qui se situent à une distance inférieure à $\frac{\Delta x}{2}$ sont fusionnés.

4. Un maillage représentant une vague est généré à partir de particules connectées le long de la courbe de vague à intervalles réguliers t_g, formant des couches successives de quadrilatères pour construire une surface continue, même en cas de changement dans la structure de la courbe. Ces particules sont initialement déplacées vers la crête en suivant la vitesse inverse de la courbe de vague $-u_l$, puis reçoivent une vitesse amplifiée u_s selon leur hauteur. On soustrait également $t_g u_s$ à la position de la particule après qu'elle ait été déplacée sur la crête pour s'assurer que sa position coïncide bien avec la surface de la simulation. Tout cela permet de simuler le renversement de la vague en donnant une vitesse plus élevée aux particules se situant au sommet. Une fois détachées du liquide, elles sont soumises à un comportement balistique et peuvent déformer la surface de l'eau en cas de collision. Pour ce faire, on déplace de $-p_\mathrm{m}$ la cellule dans laquelle la particule retombe et de $\frac{p_\mathrm{m}}{8}$ les cellules voisines, p_m étant la masse d'une particule choisie par un utilisateur. Ainsi, même sans transport réel de liquide, les vagues ont un impact sur la simulation.

5. Finalement, le rendu d'une vague est exécuté en générant une deuxième version du maillage, permettant de donner de l'épaisseur à la vague, puis en décalant ce deuxième maillage dans le sens de la normale du premier, pondéré par la masse choisie d'une particule, puis en connectant les extrémités par des quadrilatères.

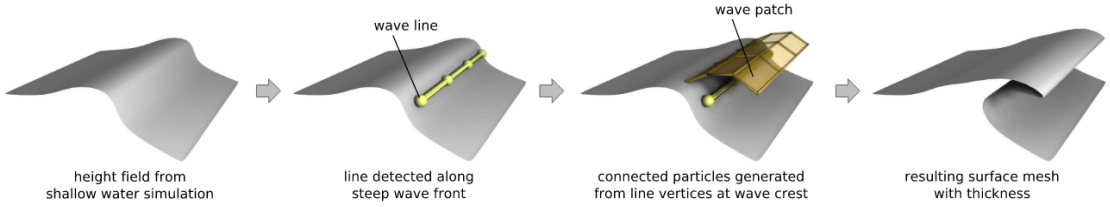


FIGURE 3.6 – Génération géométrique d'une vague. Image extraite de l'article de Thürey et al., 2007 [13].

3.1.4 Généralisation de la simulation d'eau peu profonde avec des ondes de surface dispersives

Dans l'article *Generalizing Shallow Water Simulation with Dispersive Surface Waves* [8], Jeschke et Wojtan proposent une méthode permettant d'étendre la précision du modèle d'eau peu profonde dans les hautes fréquences en le couplant au modèle *Airy Wave Theory* [1], qui consiste en une approximation linéaire des équations de la mécanique des fluides décrivant la propagation de vagues de faible amplitude sur une surface libre, tout en prenant en compte les effets de dispersion. Cette méthode permet d'obtenir un nouveau solveur hybride capable de combler les faiblesses des deux modèles et ainsi d'obtenir un modèle combinant la précision de *Airy Wave Theory* et la prise en compte des terrains qu'apporte le modèle d'eau peu profonde. Pour cela, les auteurs proposent de coupler les deux systèmes en décomposant h et $q = hu$ en deux parties à l'aide d'une équation de diffusion permettant de filtrer les parties de hautes fréquences. Cela permet de récupérer les parties correspondant au modèle d'eau peu profonde et au modèle dû à *Airy Wave Theory*. Ensuite, les deux parties sont simulées séparément. La partie due à *Airy Wave Theory* est transportée horizontalement par les vitesses du modèle d'eau peu profonde. Enfin, les deux parties sont fusionnées pour récupérer le déplacement final.

3.2 Méthodes Lagrangiennes

Bien que moins répandues, les méthodes de simulation d'eau peu profonde Lagrangiennes présentent des avantages intéressants en ce qui concerne la gestion de terrains présentant une topologie arbitraire ainsi que le couplage avec d'autres simulations de fluide Lagrangiennes.

3.2.1 Simulation d'eau peu profonde à base de SPH

La méthode tirée de l'article *SPH-Based Shallow Water Simulation* par Solenthaler et al. [12] utilise un système de particules pour résoudre les équations d'eau peu profonde. Ces particules sont simulées au niveau du terrain et sont principalement soumises à la force $f_i^{\text{terrain}} = -h\nabla H_i$ qui agit le long du gradient du terrain. Le terrain est modélisé par

un champ de hauteur comme dans les méthodes vues précédemment. La densité de particules en une cellule i est interprétée comme la hauteur du fluide. Faire cela naïvement en utilisant directement la mesure de densité résulte en une surface très inégale, criblée de bosses. La solution à ce problème est de lisser le champ de densité en appliquant un noyau à la densité discrète :

$$\rho(r) = \sum_j \rho_j W(r - r_j, l).$$

On récupère également une version normalisée de ce champ de densité :

$$\hat{\rho}(r) = \frac{\rho(r)}{\sum_j W(r - r_j, l)}$$

avec le paramètre l représentant la taille du noyau de lissage.

Finalement, on interpole linéairement entre ces deux champs de densité en fonction de

$$\alpha = \frac{\rho(r) - \rho_{\min}(r)}{\rho_{\max}(r) - \rho_{\min}(r)}$$

avec $\rho_{\max}(r)$ et $\rho_{\min}(r)$ les valeurs de densité maximales et minimales dans le voisinage de la position r.

Un dernier problème subsiste. En effet, dans les régions où la densité est faible, on peut observer l'apparition de monticules d'eau isolés. Pour réduire ce problème, on augmente la taille du noyau de lissage à mesure que la densité diminue. La figure 3.7 illustre les effets des différents lissages sur la surface de l'eau.

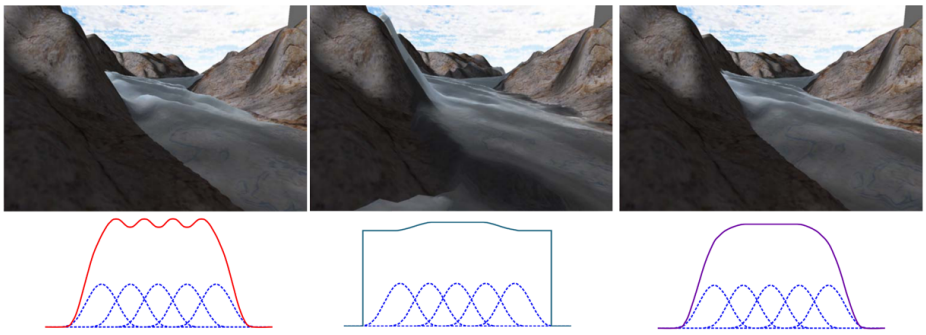


FIGURE 3.7 – Champ de densité et noyaux de lissages correspondants en dessous des images - Gauche : non normalisé, bosses à la surface ; Milieu : normalisé, surface lisse mais hauteurs incorrectes aux bordures ; Droite : interpolé. Image extraite de l'article de Solenthaler et al., 2011 [12].

3.2.2 Simulation d'eau peu profonde à base de particules pour des domaines irréguliers

L'article *Particle-Based Shallow Water Simulation for Irregular and Sparse Domains* [4] de Chládek et Ďurikovič propose de légères améliorations de la gestion des conditions aux frontières et de la précision de la simulation par rapport à l'article de Solenthaler et al. [12]. En effet, bien qu'une simulation réaliste puisse être atteinte grâce à une méthode de simulation classique, des erreurs de précision dues aux régions contenant peu de particules peuvent se glisser dans la simulation. Pour réduire ce genre de problème lors du calcul des hauteurs, le noyau utilisé pour calculer la densité est modifié en utilisant le filtre de Shepard, qui normalise la valeur du noyau W_{ij}^{2D} par la valeur du même noyau appliquée à tous les voisins j de la particule i.
Ainsi, le noyau devient :

$$\tilde{W}_{ij} = \frac{W_{ij}}{\sum_j W_{ij} \frac{V_j}{h_j}}.$$

Une solution similaire est utilisée au niveau du calcul du gradient, bien que plus coûteuse car nécessitant l'inversion d'une matrice.

Finalement, une amélioration est apportée à la gestion des limites de la simulation. En

effet, pour certaines configurations de terrain, des particules trop proches des bordures peuvent entraîner des forces trop importantes. Pour éviter ce problème, des particules sont échantillonnées aux bordures et intégrées au calcul des forces.

3.2.3 Simulation d'eau peu profonde à base de particules avec éclaboussures et vagues déferlantes

La méthode présentée dans l'article *Particle-based Shallow Water Simulation with Splashes and Breaking Waves* [5] de Fujisawa et al. se base sur un champ de hauteur représenté par des particules 2D (similaire à la méthode de Solenthaler et al. [12]). Elle ajoute des particules 3D instanciées en fonction du comportement des vagues. Les deux systèmes peuvent être couplés sans grande difficulté étant donné que les deux utilisent une représentation Lagrangienne. Pour générer des effets en 3D, cet article considère seulement deux types d'effets : des éclaboussures verticales produites lors du déferlement des vagues et des projections d'eau horizontales engendrées par leurs pentes abruptes. Des particules 3D sont générées si :

$\frac{h_i - h_i^{\text{prev}}}{\Delta t} > \gamma_{3\text{D}}$: la variation de hauteur est supérieure à un certain seuil $\gamma_{3\text{D}}$.

Ces dernières sont classifiées en trois catégories : *top*, *steep* et *slope*. La création de ces particules est gérée selon la valeur $h_{\text{diff}} = \max(h_j) - \min(h_j)$.

1. Les particules *top* sont créées lorsque $h_{\text{diff}} \leq \gamma_{\text{top}}$. Elles sont générées avec la vitesse
$$u_{xz}^{\text{top}} = 0, \quad u_y^{\text{top}} = k^{\text{top}} \frac{h_i - h_i^{\text{prev}}}{\Delta t}.$$
2. Les particules *steep* sont créées lorsque $\gamma_{\text{steep}} \leq h_{\text{diff}}$. Elles sont générées avec la vitesse oblique
$$u_{xz}^{\text{steep}} = \frac{a_{xz}^{2D}}{|a_{xz}^{2D}|} \sqrt{g h_i}, \quad u_y^{\text{steep}} = k^{\text{steep}} \frac{h_i - h_i^{\text{prev}}}{\Delta t}.$$
3. Les particules de *slope* ont une vitesse correspondant à l'interpolation linéaire

des deux types de particules précédents en fonction d'un paramètre α_{slope} choisi par l'utilisateur.

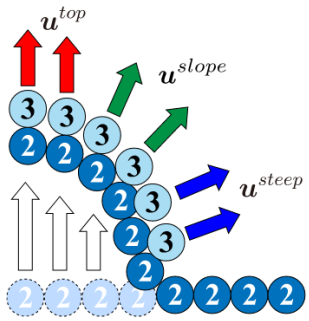


FIGURE 3.8 – Classification des trois types de particules 3D. Image extraite de l'article de Fujisawa et al., 2017 [5].

La position initiale de chacune de ces particules 3D est définie comme suit :

$$x_{\text{xz}}^{\text{3D}} = x_{\text{xz}}^{\text{2D}}$$
$$x_{\text{y}}^{\text{3D}} = (1-\alpha_{\text{height}})h_i^{\text{2D}} + \alpha_{\text{height}} \max(h_j) + H(x^{\text{2D}})$$

avec α_{height} un paramètre au choix, 0.0 ou 0.3 selon l'article.

Cette méthode permet ainsi d'obtenir des résultats exhibant des niveaux de détails habituellement retrouvés dans les méthodes en 3D, tout en simplifiant la complexité du modèle. La figure 3.9 donne une bonne illustration des résultats possibles.

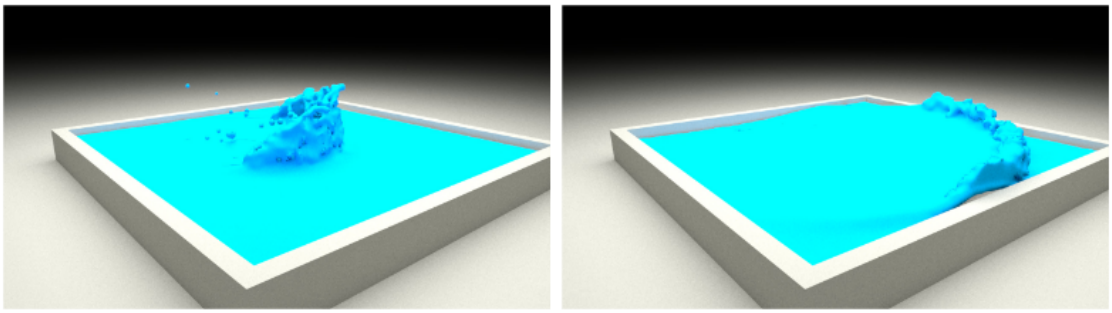


FIGURE 3.9 – Évolution d'une vague en fonction du temps. Image extraite de l'article de Fujisawa et al., 2017 [5].

3.3 Méthodes hybrides

Les méthodes hybrides combinent les avantages des méthodes Eulériennes et Lagrangiennes, permettant de bénéficier de la stabilité et de la précision des deux approches. Ces méthodes sont particulièrement utiles pour simuler des phénomènes complexes où les interactions entre différentes échelles et types de fluides sont importantes.

3.3.1 Animation de l'eau utilisant l'équation d'onde, couplée à une simulation SPH

L'article *Water Animation using Coupled SPH Wave Equation* [11] par Ramakrishnan et al. propose une méthode pour coupler deux méthodes de simulation fondamentalement différentes : une méthode Eulérienne et une méthode Lagrangienne. Le but est d'exploiter le niveau d'expressivité d'une méthode SPH en 3D tout en réduisant le coût grâce à une méthode à base de champ de hauteur, dans ce cas-ci, le modèle Équation d'onde. Bien que cette méthode couple deux simulations différentes, il est important de noter que la simulation d'Équation d'onde affecte le modèle SPH mais pas l'inverse. Pour coupler les deux approches, un système de particules fantômes est utilisé. Les positions de ces particules fictives sont calculées en fonction des particules SPH : pour une particule dont la position est x, y, z, une possible particule fictive se trouverait à la position x_g, y_g, z_g suivante :

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{bmatrix} x + ir \\ y + jr \\ w(x + ir, y + jr) - kr \end{bmatrix}.$$

Cette particule fictive aurait alors la vitesse verticale du modèle d'Équation d'onde en ce point et pour densité celle de l'eau au repos. Une interaction entre une particule

fictive et une particule SPH n'a lieu que lorsque

$$\left\| \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} - \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right\| < h.$$

Les particules fictives ne sont pas stockées en mémoire et sont donc évaluées seulement lors du calcul de force des particules SPH suffisamment proches de la surface du champ de hauteur. Cette approche offre un couplage permettant de transmettre le mouvement d'Équation d'onde vers la simulation SPH, qui peut par la suite modéliser les effets plus complexes en 3D. La figure 3.10 illustre le potentiel de ce couplage.

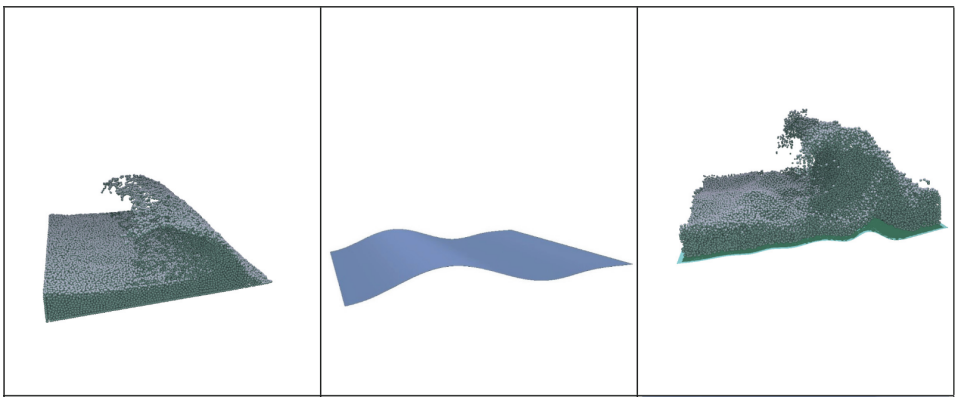


FIGURE 3.10 – SPH couplé avec Équation d'onde. Image extraite de l'article de Ramakrishnan et al., 2023 [11].

3.4 Conclusion

Dans ce chapitre, nous avons présenté un ensemble de méthodes de simulation de liquides basées sur le modèle d'eau peu profonde. Nous avons exploré les méthodes Eulériennes, Lagrangiennes et hybrides, en détaillant les articles clés et les techniques qu'ils proposent pour améliorer la stabilité, la précision et le réalisme visuel des simulations. Ces travaux connexes fournissent une base solide pour le développement d'un système robuste de simulation d'eau efficace dans un contexte de jeu vidéo. Dans le cadre de ce projet, nous nous appuyons principalement sur les articles de Chentanez et Müller [2]

et de Mao et Wu [10], qui offrent des solutions prometteuses pour la simulation d'eau peu profonde qui reste efficace, tout en offrant des comportements complexes et variés désirés dans un contexte de jeu vidéo.

CHAPITRE 4

SYSTÈME PROPOSÉ

Tout au long de ce chapitre, nous allons aborder les systèmes qui ont été implémentés dans le cadre de ce projet. Nous proposons une réimplémentation robuste de méthodes procédurales discutées dans le chapitre 3, principalement celle de Chentanez et Müller, ainsi qu'une méthode originale de couplage de l'écume des particules vers la grille. Toutes ces contributions augmenteront d'autant les capacités et la qualité visuelle du modèle d'eau peu profonde.

4.1 Solveur de base

Pour l'implémentation de ce prototype, il a été décidé d'utiliser l'approche Eulérienne en discrétisant le volume d'eau sur une grille MAC (voir le chapitre 2). Ainsi, l'entièreté des données nécessaires à la simulation peut être encodée dans une seule texture RVBA, en stockant la profondeur de l'eau dans le canal R, la hauteur du terrain dans le canal V, et les vitesses horizontales aux faces dans les canaux B et A. L'algorithme de simulation étant hautement parallélisable, il est possible d'implémenter entièrement le système en utilisant des nuanceurs de calcul sur GPU, rendant possible la simulation en temps réel.

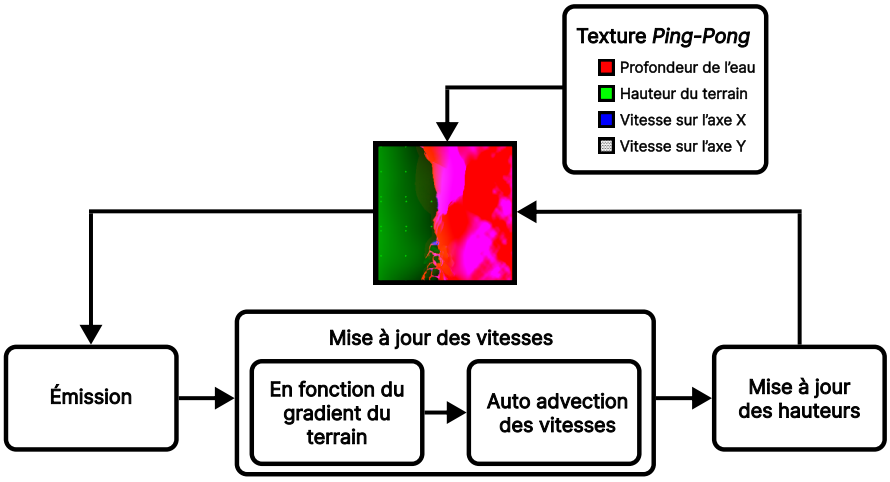


FIGURE 4.1 – Schéma de notre solveur de base, encodé sur une texture.

Les informations de hauteurs stockées dans la texture de simulation peuvent être utilisées pour déplacer verticalement les sommets d'un maillage 3D, représentant ainsi la surface de l'eau dans le monde. Finalement, le rendu du liquide est réalisé grâce à diverses techniques permettant un rendu semi-transparent, prenant en charge les réfractions ainsi que l'absorption de la lumière par le volume d'eau.

4.2 Améliorations du solveur

Pour pallier certaines des limitations du solveur de base, il est possible d'apporter quelques modifications à l'algorithme de simulation. Dans cette section, nous allons aborder les améliorations que nous avons choisi d'implémenter dans ce prototype.

4.2.1 Améliorations de stabilité

Le solveur de base est sujet à des instabilités numériques pouvant souvent briser sa crédibilité, allant parfois jusqu'à entraîner la divergence de la simulation. De plus, en raison de certaines imprécisions numériques, la hauteur de l'eau peut parfois devenir négative, causant encore plus de problèmes d'instabilités.

En suivant la méthode de Chentanez et Müller [2], plusieurs améliorations sont apportées au solveur de base afin d'en améliorer la stabilité numérique. Nous imposons tout d'abord une hauteur d'eau strictement positive afin d'éviter l'apparition de profondeurs négatives au cours de la simulation.

Nous appliquons également une limitation de la vitesse horizontale afin que sa norme ne dépasse pas $\alpha \frac{\Delta x}{\Delta t}$, avec $\alpha = 0.5$. Cette contrainte limite la vitesse maximale de propagation des vagues, mais améliore significativement la stabilité de la simulation dans les régions fortement turbulentes. La valeur α est conservée comme paramètre utilisateur afin de rendre cette limitation plus ou moins stricte.

Lors de l'intégration des hauteurs, nous remplaçons la valeur $\bar{d}$ par $\bar{d} - d_{\text{adj}}$, avec

$d_{\text{adj}} = \max\left(0, \frac{\bar{d}_{i+1,j}+\bar{d}_{i-1,j}+\bar{d}_{i,j+1}+\bar{d}_{i,j-1}}{4} - d_{\text{avgmax}}\right)$ et $d_{\text{avgmax}} = \beta\frac{\Delta x}{g\Delta t}$, où $\beta = 2.0$ dans l'ensemble de nos expériences. Cette correction limite l'amplitude des vagues en eau profonde et améliore la stabilité du solveur tout en conservant un comportement visuellement plausible.

Enfin, afin de réduire les artefacts d'*overshoot* observés lors de la transition entre les eaux profondes et peu profondes (section 3.1), nous appliquons également la correction suivante :

— Si $h_{i,j} - h_{i-1,j} > \lambda_{\text{edge}}$ et $h_{i,j} > h_{i+1,j}$,
alors $d_{i,j}^{n+1} = d_{i,j}^{n} + \alpha_{\text{edge}}(\max(0, \frac{1}{2}(d_{i,j} + d_{i+1,j})) - d_{i,j})$

— Si $h_{i,j} - h_{i+1,j} > \lambda_{\text{edge}}$ et $h_{i,j} > h_{i-1,j}$,
alors $d_{i,j}^{n+1} = d_{i,j}^{n} + \alpha_{\text{edge}}(\max(0, \frac{1}{2}(d_{i,j} + d_{i-1,j})) - d_{i,j})$

avec $\lambda_{\text{edge}} = 2\Delta x$ et $0.1 < \alpha_{\text{edge}} < 0.5$.

Bien que cette méthode permette de réduire le problème, elle ne permet pas de l'effacer complètement. Ces artefacts pourraient donc réapparaître dans certains cas plus extrêmes.

4.2.2 Gestion des obstacles internes à la simulation

Le solveur de base ne gère pas correctement la présence d'obstacles internes à la simulation. Cela peut engendrer des instabilités numériques ainsi que des artefacts visuels marqués, particulièrement à proximité des côtes ou des zones émergées.

Pour remédier à ce problème, il est possible d'identifier certaines faces de la grille comme étant réfléchissantes. Une face horizontale $(i+\frac{1}{2}, j)$ est considérée comme un mur si

$$\left(d_{i,j} \leq \varepsilon \text{ et } b_{i,j} > b_{i+1,j}\right) \text{ ou } \left(d_{i+1,j} \leq \varepsilon \text{ et } b_{i+1,j} > d_{i,j}\right),$$

où $\varepsilon > 0$ représente une petite constante permettant d'identifier les cellules ne contenant

pas d'eau. Dans toutes nos expériences, nous utilisons $\varepsilon = 10^{-4}\Delta x$. À partir de cette heuristique, il est possible de construire un masque binaire à deux canaux indiquant quelles faces doivent être traitées comme des murs. Pour les faces perpendiculaires à l'axe X, on définit le masque

$$m_x\left(i+\frac{1}{2},j\right) = \begin{cases} 0, & \text{si la face est réfléchissante,} \\ 1, & \text{sinon.} \end{cases}$$

Lors de l'intégration des vitesses, ce masque est appliqué directement aux composantes de vitesse afin d'annuler les vitesses traversant les frontières solides :

$$u_{i+\frac{1}{2},j} \leftarrow m_x\left(i+\frac{1}{2},j\right) u_{i+\frac{1}{2},j}.$$

Une construction identique est utilisée pour les faces perpendiculaires à l'axe Y et la composante de vitesse v.

Finalement, les faces identifiées comme des murs sont exclues de l'étape d'advection. Cela empêche le transport de quantité de mouvement à travers les frontières solides et réduit significativement les artefacts observés à proximité des côtes, tout en améliorant la stabilité générale de la simulation.

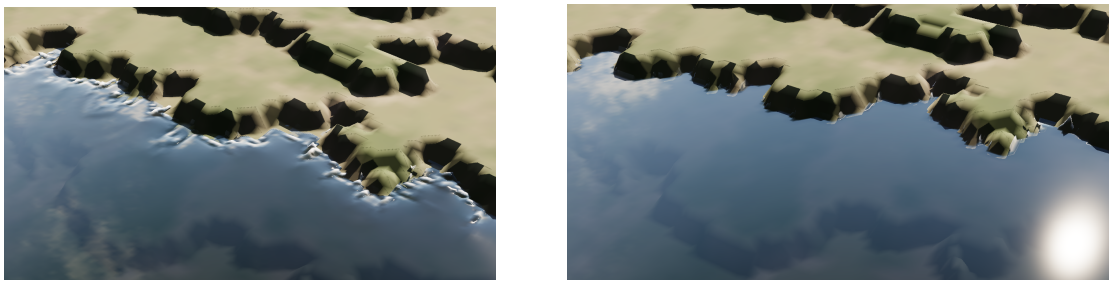


FIGURE 4.2 – Gestion des obstacles internes - Gauche : sans | Droite : avec. Les faces noires correspondent à un artefact du *shadow map*, cela n'affecte pas notre technique.

Sur cette figure, on observe comment les replis de la surface de l'eau sont fortement réduits par notre traitement.

4.2.3 Bordures absorbantes

Dans certains scénarios, il est souhaitable de pouvoir simuler une portion d'océan ouvert. Cependant, si les bordures du domaine du solveur de base sont toutes réfléchissantes, toute vague atteignant le bord du domaine sera réfléchie vers l'intérieur de la simulation. Ce comportement brise rapidement l'illusion d'un océan s'étendant au-delà des limites du domaine simulé.

Pour résoudre ce problème, nous utilisons une méthode inspirée des *Perfectly Matched Layers* (PML), adaptée aux équations de *Shallow Water* par Chentanez et Müller [2]. L'idée principale consiste à amortir progressivement l'amplitude des vagues avant qu'elles n'atteignent les frontières du domaine, de manière à ce que leur énergie devienne négligeable au moment où elles quittent la zone simulée, afin d'éviter les réflexions aux bords du domaine.

Pour ce faire, une région d'amortissement est définie le long des frontières ouvertes du domaine. À l'intérieur de cette région, un champ scalaire ϕ est introduit. Ce champ est initialisé à zéro puis utilisé pour modifier l'évolution de la hauteur d'eau et des vitesses horizontales selon les équations suivantes :

$$
\begin{aligned}
d_{i,j} &\leftarrow d_{i,j} + \left(-\sigma_{i,j}(d_{i,j} - d_{\text{rest}}) + \phi_{i,j}\right)\Delta t, \\
u_{i+\frac{1}{2},j} &\leftarrow u_{i+\frac{1}{2},j} - \frac{1}{2}\left(\phi_{i+1,j} + \phi_{i,j}\right) u_{i+\frac{1}{2},j}\Delta t, \\
\phi_{i,j} &\leftarrow \phi_{i,j} - \lambda_{\text{update}}\phi_{i,j}\frac{w_{i,j+\frac{1}{2}} - w_{i,j-\frac{1}{2}}}{\Delta x}\Delta t, \\
\phi_{i,j} &\leftarrow \lambda_{\text{decay}}\phi_{i,j}.
\end{aligned}
$$

Les équations ci-dessus correspondent à une couche d'amortissement agissant le long l'axe X ; une formulation correspondante peut être appliquée aux frontières associées à l'axe Y.

Dans toutes nos expériences, nous utilisons $\lambda_{\text{decay}} = 0.9$ et $\lambda_{\text{update}} = 0.1$. Ces paramètres peuvent toutefois être ajustés afin d'améliorer la stabilité. Le coefficient d'amortissement $\sigma_{i,j}$ varie à l'intérieur de la région d'amortissement et augmente progressivement vers la frontière du domaine. En pratique, une interpolation quadratique ou cubique entre le bord interne de la région et la frontière extérieure produit de bons résultats. Des valeurs élevées de $\sigma_{i,j}$ permettent une dissipation plus rapide de l'énergie et autorisent l'utilisation de régions d'amortissement plus étroites. Toutefois, si ce coefficient devient trop important, des réflexions numériques peuvent apparaître lors de l'entrée des vagues dans la zone d'amortissement. Enfin, dans la majorité des cas, nous fixons la largeur de la région d'amortissement à dix cellules. Cette valeur offre un bon compromis entre stabilité numérique et efficacité d'absorption, mais peut être adaptée selon les besoins de la simulation.

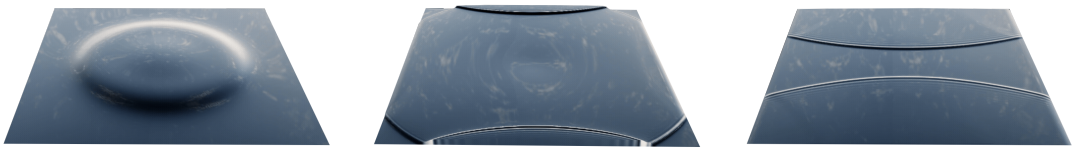


FIGURE 4.3 – Évolution d'une onde (vague) dans un domaine possédant deux bordures absorbantes gauche-droite et deux bordures réfléchissantes haut-bas. On peut observer la qualité de l'absorption de la vague sur les côtés gauches et droits alors que la vague poursuit son parcours de haut en bas.

4.2.4 Schéma d'advection MacCormack

Il est possible d'améliorer l'étape d'advection du solveur de base. L'advection semi-Lagrangienne constitue une approximation du transport réel induit par le liquide, introduisant une dissipation numérique importante. En pratique, les quantités advectées perdent progressivement leurs détails fins et deviennent de plus en plus lissées au cours du temps. Pour réduire cette dissipation, il est possible d'utiliser le schéma d'advection de MacCormack, qui consiste à estimer l'erreur de l'advection semi-Lagrangienne puis à l'utiliser pour corriger le résultat.

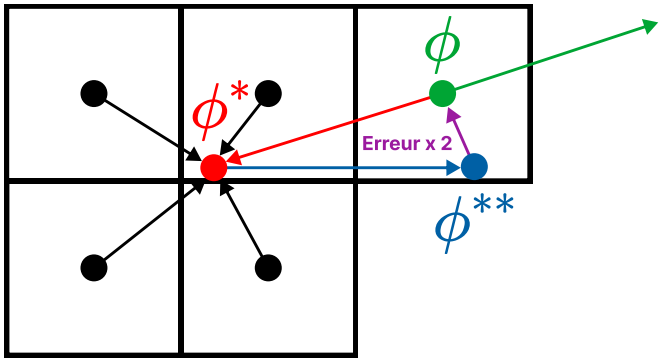


FIGURE 4.4 – Schéma d'advection MacCormack.

Le procédé est le suivant. Une première advection semi-Lagrangienne produit une valeur intermédiaire ϕ^*. Une seconde advection est ensuite réalisée en remontant le champ de vitesse afin d'obtenir une approximation reconstruite ϕ^{**} de la quantité initiale ϕ. Dans le cas idéal, on retrouverait $\phi^{**} = \phi$, mais les erreurs d'interpolation rendent cette égalité inexacte. On estime alors l'erreur d'advection par :

$$e = \frac{1}{2}(\phi - \phi^{**}),$$

puis on corrige la valeur advectée :

$$\hat{\phi} = \phi^* + e.$$

Cette correction réduit partiellement la dissipation et permet de mieux conserver les hautes fréquences du champ transporté. Cependant, elle peut engendrer des dépassements (*overshoots*) ou sous-dépassements (*undershoots*), pouvant produire des valeurs non physiques et mener à des cas d'instabilités. Pour garantir la stabilité, on applique une étape de limitation :

$$\hat{\phi} \leftarrow \min\left(\phi_{\max}, \max\left(\phi_{\min}, \hat{\phi}\right)\right).$$

Une alternative consiste à revenir à la valeur ϕ^* en cas de détection d'un dépassement.

Ces corrections permettent de conserver une simulation stable tout en réduisant significativement la dissipation numérique. La figure 4.5 montre bien la dissipation et la correction par l'advection MacCormack.



FIGURE 4.5 – Gauche : Semi-Lagrangien | Droite : MacCormack. Ajout de détails fins, mais aussi quelques instabilités près du centre du vortex.

En pratique, ce schéma d'advection est appliqué uniquement à l'étape d'auto-advection du fluide. Cette approche nécessite de stocker un nombre nettement plus élevé de variables, résultant en un plus grand nombre d'opérations de lecture et d'écriture, rendant cette approche moins performante. Pour ces raisons, et compte tenu du gain visuel plus limité que cette méthode apporte, nous ne conseillons pas son implémentation dans le cas d'une utilisation en jeu sans optimisation au préalable. Dans notre implémentation, cette fonctionnalité restera désactivée par défaut.

4.3 Advection de détails

Comme discuté dans la section 3.1.1, la simulation ne permet pas de représenter les détails plus fins qu'une cellule de simulation, perdant de ce fait toute notion de mouvement dans des eaux où les mouvements verticaux sont négligeables. Dans cette section, nous décrivons la méthode utilisée dans l'article de Chentanez et Müller [2] pour résoudre ce problème.

Cette méthode consiste à introduire une texture de détails indépendante de la simulation basse résolution. Cette texture de détails peut être générée à partir d'un spectre

FFT d'océan, de toute autre représentation périodique ou bien simplement produite par un artiste dans le cas d'un rendu stylisé. Ses valeurs sont ensuite échantillonnées pour enrichir visuellement la surface simulée. Dans notre implémentation, nous utilisons principalement cette technique pour déplacer une carte de normale. Elle pourrait tout à fait être utilisée pour transporter une carte de déplacement et utiliser cette dernière afin de déplacer un maillage haute résolution.

Pour chaque cellule de la grille (i, j), on stocke trois ensembles de coordonnées de texture :

$$\left(s_{i,j}^k, t_{i,j}^k\right), \quad k \in \{1,2,3\}.$$

Initialement, ces coordonnées étaient définies comme $(i\Delta x, j\Delta y)$, ce qui introduisait des discontinuités lors du déplacement du domaine de simulation dans l'espace monde. En pratique, nous utilisons des coordonnées définies en espace monde afin de garantir la cohérence lors des translations des instances de simulation. Les coordonnées de texture sont transportées par le champ de vitesse via une advection semi-Lagrangienne. Afin d'éviter une distorsion excessive, chaque ensemble (s^k, t^k) est réinitialisé périodiquement toutes les τ secondes, avec un décalage de phase de $\tau/3$ entre les trois ensembles. Dans la majorité des cas, nous utilisons $\tau = 3$ secondes.

Cependant, même avec cette stratégie, les coordonnées ST des textures peuvent subir des étirements importants. Pour atténuer ce problème, il est possible de calculer la déformation maximale subie par chaque cellule. Pour ce faire, étant donnée la coordonnée de texture associée à la cellule de la grille de coordonnées (i, j), dans l'ensemble de coordonnées de textures (s^k, t^k), on définit la matrice de déformation de Green :

$$G = \frac{1}{2}(DD^\top - I).$$

telle que

$$D = \begin{bmatrix} \frac{\partial s_{i,j}^k}{\partial x} & \frac{\partial s_{i,j}^k}{\partial z} \\ \frac{\partial t_{i,j}^k}{\partial x} & \frac{\partial t_{i,j}^k}{\partial z} \end{bmatrix}.$$

On calcule ensuite $\mu_{i,j}^k$, la valeur propre de plus grande magnitude de G pour chaque ensemble de coordonnées ST, afin de garder uniquement la valeur maximale parmi les trois ensembles :

$$\mu_{i,j} = \max_{k \in \{1,2,3\}} \mu_{i,j}^k.$$

Le poids associé à chaque cellule est défini par :

$$\omega_{i,j} = e^{-\Omega \mu_{i,j}}.$$

Le paramètre $\Omega \in [0.0, 2.0]$ contrôle la vitesse à laquelle les détails disparaissent sous l'effet de l'étirement. La figure 4.6 montre les effets pour différentes valeurs de Ω.

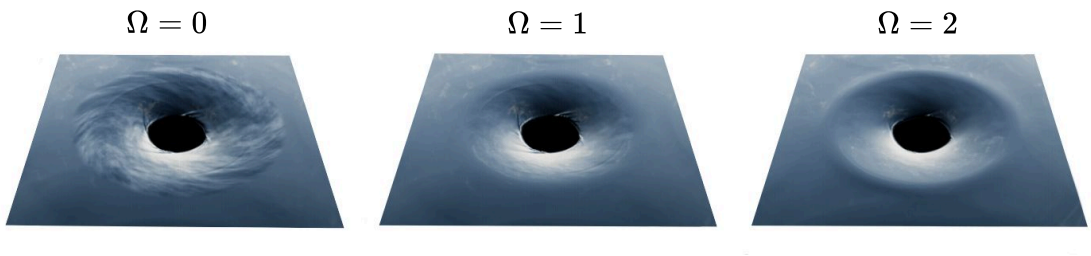


FIGURE 4.6 – Impact du paramètre Ω sur l'apparence des détails de surface.

Les normales finales peuvent alors être obtenues en combinant les contributions des trois ensembles de coordonnées ST :

$$\text{Normale}_{i,j} = \omega_{i,j} \sum_{k=1}^{3} w^k F(s_{i,j}^k, t_{i,j}^k),$$

où ω, s^k et t^k sont interpolés bilinéairement sur la grille, F est la fonction d'échantillonnage de la texture de détails, et les poids temporels sont définis par :

$$w^k = \frac{1}{3}\left(1 - \cos\left(2\pi \frac{t - t_0^k}{\tau}\right)\right),$$

avec t_0^k représentant le dernier instant de régénération de l'ensemble k.

Cette approche permet de réduire fortement la limitation liée à la résolution de la grille et de visualiser des mouvements de surface à des échelles beaucoup plus fines que la simulation sous-jacente. La figure 4.7 montre cette technique à l'œuvre lors de l'écoulement d'une rivière.

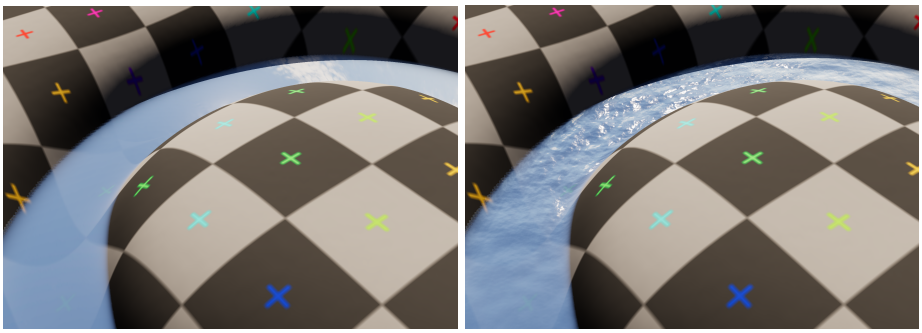


FIGURE 4.7 – Rivière sans détails (Gauche) et avec détails d'écoulement (Droite).

4.4 Système de particules GPU

Bien que convaincant sur plusieurs aspects, le modèle de *Shallow Water* souffre d'une limitation majeure. En effet, le fait que le volume d'eau soit représenté seulement par un champ de hauteur empêche la modélisation d'effets tels que les surplombs, les vagues déferlantes ou encore les cascades. Pour pallier cette limitation, Chentanez et Müller [2] et Mao et Wu [10] proposent une représentation hybride à base de particules permettant le transport de matière en 3D. Nous proposons une implémentation de cette approche hybride basée sur un système de particules GPU. Elle constitue une grande partie des efforts investis dans notre projet.

4.4.1 Système de base

Dans cette implémentation, chaque particule est identifiée par un indice unique et ses attributs sont stockés dans plusieurs tampons distincts, contenant notamment la position, la vitesse et la durée de vie. Il est aussi possible d'étendre facilement le système de base avec divers paramètres spécifiques dépendant du type de particules, tels que le rayon (taille) ou la transparence. Cette organisation permet un accès efficace aux données et

favorise un traitement massivement parallèle.

La gestion du cycle de vie des particules s'appuie sur deux listes d'indices : une liste de particules actives et une liste de particules inactives, chacune associée à un compteur indiquant le nombre d'éléments qu'elle contient. Lors de l'émission, des indices sont prélevés dans la liste des particules inactives, les attributs correspondants sont initialisés, puis ces indices sont ajoutés à la liste des particules actives. À chaque mise à jour, seules les particules actives sont traitées. Les particules dont la durée de vie a expiré sont retirées de la liste active et réinsérées dans la liste inactive, permettant ainsi la réutilisation de leur emplacement mémoire sans allocation dynamique. Les insertions et suppressions au sein de ces listes reposent sur des opérations atomiques afin de garantir la cohérence des compteurs et des indices manipulés simultanément par un grand nombre de fils d'exécution.

La simulation est réalisée à l'aide de nuanceurs de calcul, chaque fil d'exécution étant responsable de la mise à jour d'une particule active. À l'issue de cette étape, une nouvelle liste de particules actives est construite à partir des particules encore valides, ce qui réalise implicitement une compaction des données. Le nombre de groupes de calcul à exécuter est déterminé dynamiquement à partir du compteur de particules actives grâce à un appel de calcul indirect, permettant d'adapter automatiquement la charge de travail au nombre réel de particules simulées.

4.4.2 Couplage bidirectionnel entre particules et grille

Pour permettre une représentation hybride, il est nécessaire de mettre en place des moyens permettant de faire communiquer la grille de simulation et les systèmes de particules. Pour ce faire, nous procédons comme suit : un ensemble d'heuristiques est évalué pour chaque cellule afin de déterminer si une particule doit être générée, ainsi que les paramètres associés à son émission. Lorsqu'une émission est requise, une requête est transmise au système de particules et une quantité de fluide correspondante est prélevée de la grille. Les particules ainsi créées sont ensuite transportées indépendamment

selon une dynamique balistique. Lorsqu'une particule entre en collision avec la surface libre représentée par le modèle de *Shallow Water*, sa masse est réinjectée dans la cellule correspondante de la grille et sa quantité de mouvement est transférée au fluide sous la forme d'une impulsion résultant de la collision.

L'émission des particules est réalisée au moyen d'une passe de calcul parcourant une texture d'heuristiques préalablement générée à partir de l'état de la simulation. Chaque fil d'exécution traite une cellule de la grille et détermine le nombre de particules à émettre à partir d'une estimation du flux sortant, obtenu en rapportant la quantité d'eau disponible au volume représenté par une particule, tout en respectant une limite maximale d'émission par cellule.

La réinjection des particules dans la simulation repose quant à elle sur une structure de collecte associée à chaque cellule, composée d'un compteur et d'un tampon d'indices de taille fixe. Lorsqu'une particule atteint la surface libre, une opération atomique est utilisée pour incrémenter le compteur de la cellule correspondante et réserver un emplacement dans le tampon où son indice est enregistré. Une passe d'accumulation exécutée sur la grille parcourt ensuite ces listes d'indices afin de récupérer les attributs des particules concernées et d'en déduire leurs contributions respectives. Les volumes associés sont additionnés à la masse déjà présente dans la cellule, tandis que la vitesse est mise à jour à partir de la quantité de mouvement totale du système composé de la cellule et des particules réinjectées :

$$u_{i+\frac{1}{2},j} \leftarrow \frac{u_{i+\frac{1}{2},j} h_{i,j} \Delta x^2 + \sum_p u_p V_p}{h_{i,j} \Delta x^2 + \sum_p V_p}.$$

Dans cette expression, $h_{i,j}\Delta x^2$ représente le volume de fluide initialement contenu dans la cellule (pour rappel, $\Delta x = \Delta y$), tandis que V_p et u_p désignent respectivement le volume et la vitesse de chaque particule réinjectée. Le numérateur correspond à la quantité de mouvement totale avant mélange, et le dénominateur au volume total de fluide. Cette formulation permet de conserver simultanément la masse et la quantité de

mouvement lors du transfert des données de la représentation particulaire vers la grille de simulation.

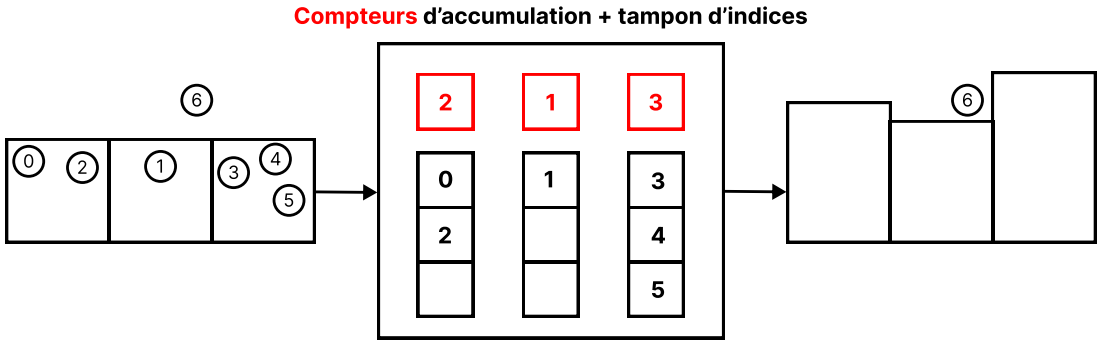


FIGURE 4.8 – Système d'accumulation des particules illustré en 2D. Les *i* encerclés représentent des particules dans la simulation, tandis que les *i* encadrés en noir représentent les indices de particules réinjectées dans la cellule correspondante. La particule 6 n'est pas en contact avec sa colonne d'eau et n'est donc pas réinjectée.

4.5 Vagues déferlantes

Bien qu'il puisse être étendu à d'autres cas d'utilisations, nous utilisons principalement ce système de couplage particules-grille dans le but de modéliser des vagues déferlantes. Dans cette section, nous allons aborder les spécificités liées à ce cas d'usage.

4.5.1 Particules de *splash* et de *spray*

Afin de reproduire les différents types de projections observés à la surface du fluide, les particules sont réparties en deux catégories distinctes : les particules de *splash* et les particules de *spray*. Ces deux types de particules reposent sur la même infrastructure de simulation et partagent les mêmes mécanismes d'émission, de transport et de réinjection dans la grille. Ils se distinguent principalement par leurs paramètres visuels et physiques, notamment leur taille apparente, leur méthode de rendu et la quantité de forces de frottement qui leur sont appliquées. Par exemple, les particules de *splash* sont environ deux fois plus grosses que les particules de *spray*, même si leur taille exacte varie selon les scènes. Cette séparation permet de représenter à la fois les gouttes volumineuses issues

du déferlement des vagues et les fines particules en suspension dans l'air générées par les éclaboussures.

4.5.2 Détection de vagues

Les phénomènes de déferlement ne peuvent pas être représentés directement par un modèle basé sur un champ de hauteur. Les vagues qui se briseraient naturellement dans un environnement tridimensionnel tendent à engendrer des instabilités numériques dans une simulation bidimensionnelle. Afin de limiter ces artefacts tout en préservant l'apparence visuelle du phénomène, les vagues susceptibles de déferler sont détectées puis amorties, tandis qu'une partie de leur énergie est transférée vers le système de particules.

De manière similaire à la méthode de Chentanez et Müller [2], une cellule est considérée comme contenant une vague déferlante lorsqu'elle satisfait simultanément les critères suivants :

$$|\nabla h_{i,j}| > \alpha_{\text{minSplash}} \frac{g\Delta t}{\Delta x},$$
$$\frac{d_{i,j} - d_{i,j}^{\text{prev}}}{\Delta t} > v_{\text{minSplash}},$$
$$\nabla^2 h_{i,j} < l_{\text{minSplash}},$$

où $\alpha_{\text{minSplash}}$, $v_{\text{minSplash}}$ et $l_{\text{minSplash}}$ désignent des seuils configurables par l'utilisateur. Le premier critère garantit une pente suffisante, le second permet de détecter des vagues ayant une vitesse d'élévation suffisamment rapide, tandis que le troisième limite l'émission de particules aux régions proches des crêtes. En pratique, $\alpha_{\text{minSplash}} = 0.45$, $v_{\text{minSplash}} = 4$ et $l_{\text{minSplash}} = -4$ dans la plupart de nos scènes de test.

Le gradient de hauteur est évalué à l'aide de différences finies décentrées :

$$\nabla h_{i,j} = \begin{bmatrix} \frac{\max_{\text{abs}}(h_{i+1,j}-h_{i,j}, h_{i,j}-h_{i-1,j})}{\Delta x} \\ \frac{\max_{\text{abs}}(h_{i,j+1}-h_{i,j}, h_{i,j}-h_{i,j-1})}{\Delta x} \end{bmatrix},$$

avec

$$\max_{\text{abs}}(a,b) = \begin{cases} a & \text{si } |a| > |b| \\ b & \text{sinon.} \end{cases}$$

Le Laplacien est approximé à l'aide de différences centrales :

$$\nabla^2 h_{i,j} = \frac{h_{i+1,j} + h_{i-1,j} + h_{i,j+1} + h_{i,j-1} - 4h_{i,j}}{(\Delta x)^2}.$$

4.5.3 Instanciation des particules

Lorsque les critères de détection sont satisfaits, de nouvelles particules sont générées dans une région localisée autour de la cellule détectée et orientée selon la direction du gradient de hauteur.

La position initiale de chaque particule est déterminée à partir du centre de la cellule, puis décalée aléatoirement dans une zone d'émission perpendiculaire au gradient de hauteur. Un léger bruit tridimensionnel supplémentaire est également appliqué afin de répartir les particules à l'intérieur du volume émis et de limiter l'apparition de motifs réguliers.

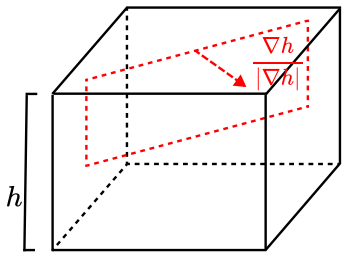


FIGURE 4.9 – Zone d'émission des particules de *splash* et de *spray* (en rouge).

La vitesse initiale est calculée à partir de la vitesse locale du fluide échantillonnée à la position de la particule. Une composante horizontale supplémentaire, alignée avec la direction de l'écoulement local, est ajoutée afin d'accentuer la projection des particules, tandis qu'une composante verticale proportionnelle à la vitesse d'élévation de la surface contrôle leur impulsion ascendante. Enfin, la position initiale est légèrement avancée le long de cette vitesse à l'aide d'un décalage temporel aléatoire, ce qui réduit les artefacts liés à une émission simultanée de toutes les particules. Pour une cellule donnée, on instancie tout d'abord des particules de *splash* selon le calcul de flux mentionné précédemment. On instancie ensuite les particules de *spray* selon le même calcul de flux, cette fois-ci basé sur la quantité d'eau restante. Cette séparation permet d'appliquer ultérieurement un traitement visuel spécifique à chaque type de particules lors du rendu.

4.5.4 Rendu des particules

Les différents types de particules introduits dans le système possèdent des caractéristiques visuelles distinctes et sont donc rendues à l'aide de techniques adaptées à leur apparence. Cette séparation permet de reproduire à la fois les fines gouttelettes en suspension associées au *spray* et les quantités d'eau plus volumineuses représentées par les particules de *splash*. Toutes ces particules peuvent également être étirées selon leur vitesse lors du rendu, permettant de mieux transmettre l'idée de leur mouvement. Cet étirement est configurable selon un paramètre utilisateur.

4.5.4.1 *Spray*

Les particules de *spray* sont rendues sous la forme de panneaux orientés vers la caméra (*billboards*). Chaque particule est représentée par un quadrilatère dont l'orientation est mise à jour afin de faire face à l'observateur. Un nuanceur de pixels est ensuite utilisé pour générer une représentation circulaire de la particule à l'intérieur de ce quadrilatère en rendant transparents les fragments à l'extérieur d'un rayon prédéfini. Cette approche permet d'obtenir l'apparence de petites gouttelettes tout en limitant le coût associé au rendu d'un grand nombre de particules en utilisant des modèles complexes. La

figure 4.10 illustre l'apparence des particules de *spray* projetées par une vague déferlante.

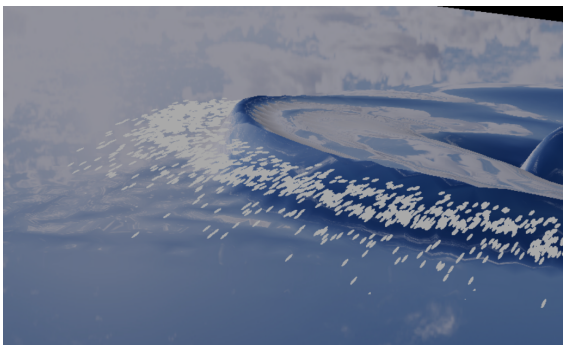


FIGURE 4.10 – Particules de *spray* projetées par une vague déferlante. À noter l'étirement des particules selon leur vitesse, ce qui accentue l'impression de mouvement.

4.5.4.2 *Splash*

Le rendu des particules de *splash* repose sur une technique de reconstruction de surface en espace écran tirée des travaux de *Green* [6]. Cette approche a été retenue pour sa capacité à produire une surface continue à partir d'un ensemble discret de particules tout en conservant un coût compatible avec les contraintes du rendu temps réel.

La première étape consiste à générer une représentation intermédiaire du fluide sous la forme d'une carte de profondeur sur l'écran (figure 4.11) et d'une carte d'épaisseur. Les particules sont rendues à l'aide de quadrilatères orientés vers la caméra, de manière similaire aux particules de *spray*. Toutefois, le nuanceur de pixels reconstruit localement une sphère centrée sur le quadrilatère afin d'en calculer analytiquement la profondeur. Cette approximation permet d'obtenir une silhouette sphérique ainsi qu'une profondeur cohérente sans avoir à rasteriser une géométrie tridimensionnelle complexe. Le coût de rendu demeure ainsi comparable à celui d'un simple panneau tout en produisant une représentation beaucoup plus fidèle des particules.

Une approche similaire est utilisée pour générer la carte d'épaisseur (figure 4.12).

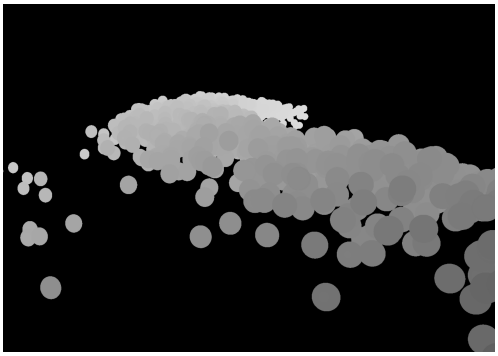


FIGURE 4.11 – Carte de profondeur des particules de *splash*. Le niveau de gris indique la profondeur de la sphère correspondante.

Pour chaque fragment, l'épaisseur traversée à l'intérieur de la sphère est évaluée analytiquement puis accumulée de manière additive. Cette quantité correspond à la distance effectivement parcourue par un rayon de vue à travers le volume représenté par la particule. L'épaisseur obtenue est donc maximale au centre de la projection de la sphère et décroît progressivement vers ses contours, où elle tend vers zéro. L'accumulation additive de ces contributions fournit une estimation de l'épaisseur totale du fluide observée pour chaque pixel. Pour sauver de la performance, cette texture est rendue à un quart de la résolution de l'image finale, ce qui n'affecte pas significativement la qualité visuelle du rendu final.

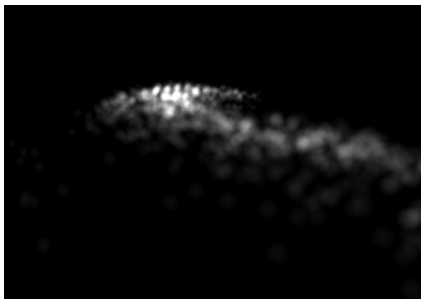


FIGURE 4.12 – Carte d'épaisseur des particules de *splash* selon l'écran. Le niveau de gris est associé à la profondeur. Plus blanc signifie plus épais.

Les cartes de profondeur et d'épaisseur obtenues présentent néanmoins des discontinuités. Afin de reconstruire une surface plus régulière, un filtrage gaussien dépendant de la profondeur est appliqué à ces deux cartes. Dans un souci d'efficacité, le flou bidi-

mensionnel est décomposé en deux passes séparables, horizontale puis verticale. Cette factorisation réduit la complexité du filtrage de $O(N^2)$ à $O(N)$, où N représente une rangée ou une colonne de pixels. Cela représente un gain significatif pour les noyaux de grande taille employés dans le cadre de la reconstruction de surface.

Cette opération introduit toutefois certaines erreurs locales dans le champ de profondeur reconstruit. En effet, en rendant le flou dépendant de la profondeur, ce dernier devient non séparable. Lors de l'estimation des normales, ces erreurs peuvent se manifester sous la forme de légers artefacts géométriques ou de variations de courbure non physiques. En pratique, ces défauts deviennent difficilement perceptibles après l'application des effets optiques du matériau et demeurent largement compensés par le gain de qualité visuelle apporté par le lissage. La figure 4.13 montre l'effet du filtrage gaussien sur les deux cartes.

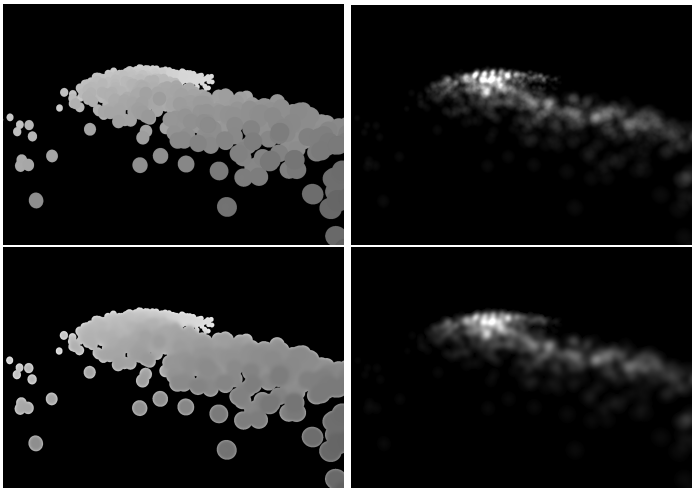


FIGURE 4.13 – Gauche : carte de profondeur floutée | Droite : carte d'épaisseur floutée. A noter que les silhouettes demeurent nettes alors que l'intérieur devient plus lisse.

Les normales de surface sont reconstruites directement à partir des variations de profondeur en espace écran. Les positions correspondant aux pixels voisins sont d'abord reconstruites en espace monde à partir de la profondeur filtrée. Deux vecteurs tangents sont ensuite calculés à l'aide de différences finies, puis leur produit vectoriel est uti-

lisé pour obtenir une approximation de la normale locale de la surface du fluide. La figure 4.14 illustre le résultat.

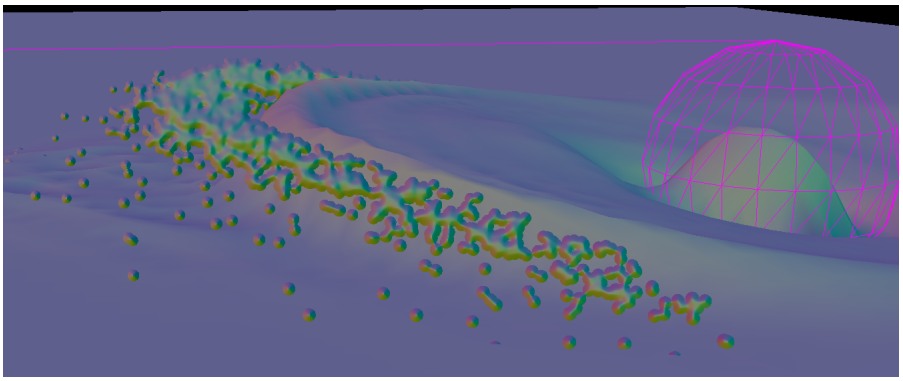


FIGURE 4.14 – Normales reconstruites. Les variations de couleur sont associées à la direction de la normale. Dans cet encodage, le violet correspond à une normale pointant vers le haut, tandis que le rouge et le vert représentent les directions horizontales.

Une fois les normales reconstruites, la surface du fluide peut être intégrée au reste de la scène. L'éclairage repose principalement sur trois phénomènes physiques : la réflexion, la réfraction et l'absorption.

La réflexion est modulée par l'approximation de Schlick du terme de Fresnel :

$$F(\theta) = F_0 + (1 - F_0)(1 - \cos\theta)^5,$$

où F_0 représente la réflectance à incidence normale et θ l'angle entre la direction d'observation et la normale à la surface. Nous utilisons $F_0 = 0.02$ dans tous nos exemples. Cette formulation permet d'augmenter progressivement la contribution spéculaire lorsque l'angle de vue devient rasant, reproduisant ainsi le comportement observé sur les surfaces liquides réelles.

La contribution de la réfraction est déterminée par une seule réfraction en espace écran, réalisée à partir de la normale reconstruite. Le vecteur de réfraction est utilisé pour décaler les coordonnées d'échantillonnage dans l'image de la scène préalablement rendue. Cette approche permet d'obtenir un effet visuellement convaincant à faible coût

de calcul. Elle présente cependant les limitations classiques des méthodes en espace écran : seules les informations déjà visibles à l'écran peuvent être réfractées. Les objets masqués ou situés hors du champ de vision ne peuvent donc pas contribuer correctement à l'image finale, ce qui peut occasionnellement entraîner des discontinuités, des zones d'information manquante ou de légers étirements sur les bords de l'écran. Dans l'ensemble des cas de plan d'eau, ces artefacts sont très peu observables.

L'absorption de la lumière à travers le liquide est évaluée à partir de la carte d'épaisseur reconstruite. Pour chaque pixel, l'épaisseur totale accumulée est interprétée comme une approximation de la distance parcourue par la lumière à travers le volume de fluide. La transmission est alors calculée à l'aide de la loi de Beer-Lambert :

$$T = e^{-k\sigma_a d},$$

où d désigne l'épaisseur accumulée et σ_a le coefficient d'absorption du milieu, et k un facteur artistique permettant de contrôler l'intensité de l'absorption. Cette formulation reproduit naturellement le fait que les régions où le fluide est plus épais absorbent davantage de lumière que les zones minces. Elle permet également d'introduire une coloration du fluide en utilisant des coefficients d'absorption distincts pour chaque composante spectrale, reproduisant ainsi la teinte caractéristique observée dans les grands volumes d'eau. Dans notre implémentation, les valeurs d'absorption sont déterminées par deux couleurs, choisies par des artistes : une couleur de faible profondeur et une couleur pour les eaux profondes. Nous interpolons entre ces couleurs en fonction de T. Pour décrire à quel point l'eau est trouble, il est possible de modifier le paramètre k. Dans nos scènes nous utilisons généralement $k = 5$.

La couleur finale du fluide est obtenue en combinant les contributions réfléchies et réfractées selon le terme de Fresnel, tandis que la composante réfractée est atténuée par le facteur de transmission issu de la loi de Beer-Lambert. Cette combinaison permet de reproduire les principales propriétés optiques d'un milieu aquatique semi-transparent

tout en conservant un coût compatible avec les contraintes du rendu en temps réel. La figure 4.15 illustre l'apparence finale d'une vague déferlante.

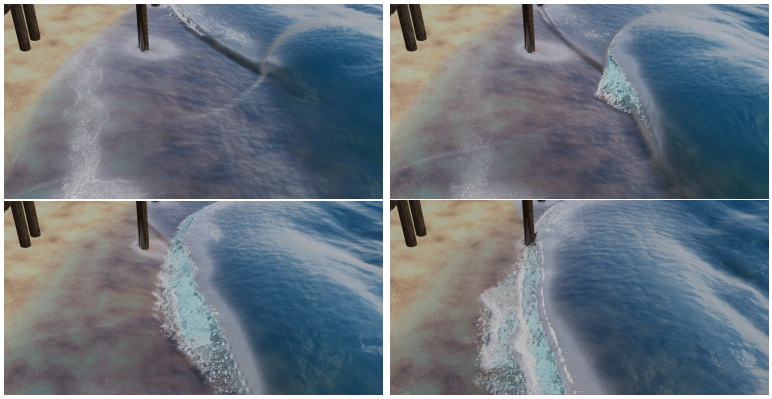


FIGURE 4.15 – Évolution d'une vague déferlante en quatre temps, de gauche à droite, de haut en bas. Dans un premier temps, on observe l'écume résiduelle de la vague antérieure. Dans un deuxième temps, on distingue la disparition de l'écume ainsi que la formation d'une nouvelle vague déferlante. Dans un troisième temps, la vague déferlante progresse avec un mouvement balistique. Finalement, la vague s'échoue sur le rivage, créant à nouveau de l'écume.

4.6 Rendu de la surface

Pour rendre le volume d'eau, nous utilisons les mêmes étapes d'éclairage, d'absorption et de réfraction en espace écran que pour les particules de *splash*. La seule différence réside dans le fait qu'il est bien plus aisé d'estimer la profondeur du fluide en espace écran en effectuant le rendu de la surface du fluide après le reste de la scène et en comparant la profondeur de la surface du fluide à celle des autres objets. La figure 4.16 illustre un résultat correspondant.

4.7 Écume

Les mécanismes présentés jusqu'à présent permettent de reproduire un large éventail de phénomènes, allant des écoulements de faible amplitude aux vagues déferlantes accompagnées de projections de fluide. Néanmoins, un élément visuel essentiel demeure

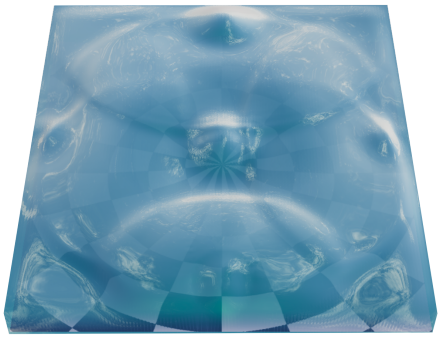


FIGURE 4.16 – Rendu final du volume d'eau incluant réflexion, réfraction et absorption.

absent : l'écume présente à la surface de l'eau. Cette dernière joue un rôle majeur dans la perception des mouvements du fluide et constitue souvent un indicateur visuel des zones fortement turbulentes.

Plusieurs travaux, tels que ceux de Chentanez et Müller [2] et de Mao et Wu [10], se sont intéressés à la modélisation de ce phénomène dans un contexte de simulation en temps réel. Dans cette section, nous présentons une réimplémentation de certaines des techniques proposées dans ces travaux, ainsi qu'une extension visant à assurer une meilleure cohérence spatio-temporelle pour l'écume.

4.7.1 Pseudo-simulation d'écume sur grille

La première composante du système repose sur une représentation de l'écume directement intégrée à la grille de simulation. À cette fin, une texture scalaire supplémentaire est introduite afin de stocker localement la quantité d'écume présente à la surface du fluide. Cette quantité est simulée indépendamment de la hauteur d'eau et de la vitesse, puis elle est utilisée lors du rendu pour moduler l'apparition d'une texture d'écume. L'évolution de cette quantité d'écume est réalisée en trois étapes : l'injection, la diffusion et l'advection.

L'injection vise à identifier les régions où l'écume est susceptible d'être générée. Deux heuristiques complémentaires sont utilisées. La première repose sur une estimation

de la quantité d'air piégée dans le fluide Φ_{air}, obtenue à partir des variations de vitesse observées aux interfaces de chaque cellule :

$$\Phi_{\text{air}} = \left\| \left(u_{i-\frac{1}{2},j+1} - u_{i-\frac{1}{2},j}, v_{i+1,j-\frac{1}{2}} - v_{i,j-\frac{1}{2}} \right) \right\|.$$

Cette mesure tend à devenir importante dans les zones où les mouvements du fluide sont fortement divergents ou turbulents.

La seconde heuristique repose sur l'amplitude du gradient de hauteur Φ_{pente} :

$$\Phi_{\text{pente}} = \left\| \left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y} \right) \right\|$$

qui permet d'identifier les fronts de vague suffisamment abrupts pour produire de l'écume.

Les deux contributions sont additionnées, puis normalisées à l'aide d'une fonction de saturation inspirée de Ihmsen et al. [7] :

$$\frac{\min(F_{\max}, \Phi_{\text{air}} + \Phi_{\text{pente}}) - \min(F_{\min}, \Phi_{\text{air}} + \Phi_{\text{pente}})}{F_{\max} - F_{\min}}$$

où $F_{\min}$ et $F_{\max}$ désignent deux seuils configurables. Le résultat obtenu est compris entre 0 et 1 et constitue un masque représentant la quantité locale d'écume nouvellement générée.

L'écume ainsi produite est ensuite soumise à un processus de diffusion afin de reproduire son étalement progressif à la surface de l'eau. Pour une valeur d'écume $\Phi_{i,j}$, l'évolution temporelle au temps $t+1$ est approchée par :

$$\Phi_{i,j}^{t+1} = \Phi_{i,j}^{t} + \alpha(\Phi_{i-1,j}^{t} + \Phi_{i+1,j}^{t} + \Phi_{i,j-1}^{t} + \Phi_{i,j+1}^{t} - 4\Phi_{i,j}^{t})$$

où α représente un paramètre utilisateur contrôlant la vitesse de diffusion. Dans la plupart de nos exemples, α est compris dans l'intervalle $[0.95, 0.99]$.

Enfin, l'écume est transportée par le champ de vitesse du fluide à l'aide d'une étape d'advection semi-Lagrangienne. Cette approche permet aux quantités d'écume de suivre naturellement les mouvements du fluide. La figure 4.17 illustre une pseudo-simulation d'écume typique de vagues déferlant sur une plage.

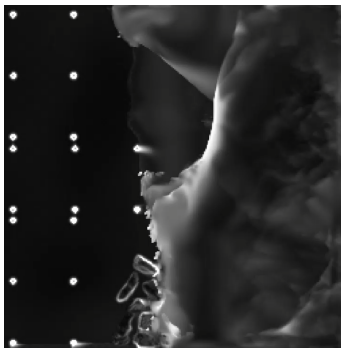


FIGURE 4.17 – Texture de pseudo-simulation de l'écume sur une plage sur laquelle se trouvent des rochers et les poteaux d'un ponton. Des vagues viennent s'y écraser, créant de l'écume. Les zones blanches correspondent à une forte concentration d'écume, tandis que les zones noires indiquent l'absence d'écume.

Lors du rendu, le masque obtenu est utilisé afin d'afficher une texture représentant visuellement l'écume. Cette texture est ensuite transportée selon la même technique employée pour les détails de surface. Cette séparation entre la simulation de la quantité d'écume et son apparence permet d'obtenir des motifs visuellement riches tout en conservant un coût de simulation limité. La figure 4.18 montre l'écume transportée sur une rivière très dynamique.

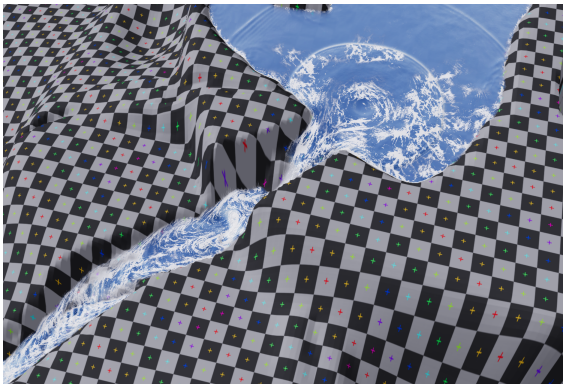


FIGURE 4.18 – Résultat de la technique d'écume sur grille dans un courant d'eau rapide provenant d'une chute.

4.7.2 Particules d'écume

Bien que la simulation sur grille permette de reproduire les principales structures d'écume observées à la surface de l'eau, elle demeure limitée par la résolution de la simulation. Les détails de très petite échelle, tels que les amas de bulles ou les résidus d'écume persistants, ne peuvent pas être représentés correctement à l'aide de la technique précédente, dont la résolution est directement liée à celle de la grille de simulation.

Afin d'enrichir le niveau de détails, une seconde représentation de l'écume est introduite sous la forme de particules. Ces dernières sont contraintes à demeurer à la surface du fluide et sont transportées à l'aide du champ de vitesse de la simulation. Leur position verticale est recalculée à chaque image à partir de la hauteur d'eau locale, ce qui garantit leur adhérence à la surface.

L'émission de ces particules est directement couplée au système de particules de *splash*. Lorsqu'une particule de *splash* entre à nouveau en contact avec la surface du fluide et réinjecte son volume dans la grille, une particule d'écume peut être générée avec une probabilité configurable. Cette approche permet de concentrer naturellement les particules d'écume dans les zones où des phénomènes de déferlement ou d'éclaboussure ont récemment eu lieu. La figure 4.19 montre de telles particules d'écume produites

par des collisions sur des rochers.



FIGURE 4.19 – Particules d'écume à la surface de l'eau suite à une vague déferlante s'écrasant sur la plage et les rochers.

Chaque particule possède une durée de vie limitée ainsi qu'un rayon qui décroît progressivement au cours du temps. Cette diminution graduelle permet de reproduire la disparition progressive des amas d'écume observés à la surface de l'eau. Bien que relativement simple, cette approche apporte une quantité importante de détails visuels à faible coût de calcul et complète efficacement la représentation sur grille. Additionnellement, cette représentation particulaire permet des accumulations d'écume dans des zones sèches, telle qu'une plage après le retrait d'une vague, ce qui n'est pas possible avec la pseudo-simulation d'écume sur grille.

4.7.3 Couplage des deux représentations

Les représentations de l'écume sur grille et par particules produisent chacune des résultats satisfaisants lorsqu'elles sont utilisées indépendamment. Néanmoins, ces deux systèmes modélisent un même phénomène physique et il apparaît donc naturel de permettre un échange d'information entre eux afin qu'ils puissent se compléter mutuellement.

Dans le cadre de ce travail, nous présentons une nouvelle méthode de couplage unidirectionnel consistant à transférer progressivement la quantité d'écume représentée par les particules vers la simulation sur grille. Cette approche vise à combiner la richesse des détails apportés par les particules avec la plus vaste couverture offerte par la pseudo-simulation de l'écume.

L'implémentation repose sur le mécanisme d'accumulation déjà utilisé pour la réinjection des particules de *splash* dans la simulation. Les indices des particules d'écume sont enregistrés dans la structure d'accumulation associée à la cellule dans laquelle elles se trouvent. Une passe exécutée sur la grille récupère ensuite les particules référencées et convertit leur contribution en une quantité d'écume ajoutée au masque simulé. La quantité transférée dépend notamment du rayon courant de la particule ainsi que des dimensions de la cellule réceptrice.

Contrairement au mécanisme utilisé pour les particules de *splash*, cette accumulation est réalisée avant la disparition effective des particules. La diminution progressive de leur rayon au cours du temps entraîne ainsi un transfert graduel de leur contribution vers la grille. Cette stratégie permet de préserver visuellement la continuité de l'écume tout en évitant les variations brusques qui résulteraient d'une conversion instantanée. La figure 4.20 montre bien la diffusion des particules d'écume dans le volume des vagues.

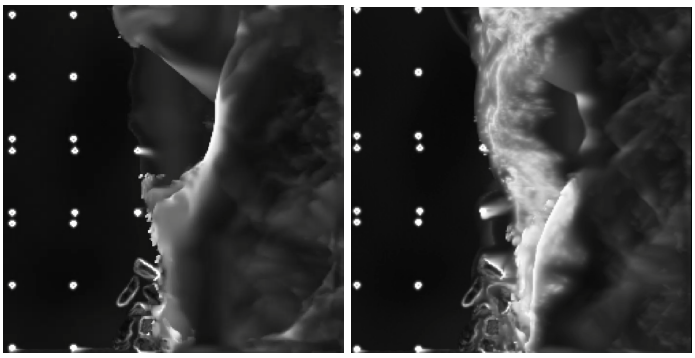


FIGURE 4.20 – Comparaison entre la pseudo-simulation d'écume de base et la version couplée avec les particules d'écume sur une scène de plage avec des rochers et un ponton. Gauche : pseudo-simulation sans couplage | Droite : pseudo-simulation avec couplage.

En pratique, ce couplage enrichit les motifs générés par la pseudo-simulation d'écume en y injectant des détails provenant de la représentation particulaire. Il contribue également à une meilleure conservation de la quantité globale d'écume au sein du système, les particules ne disparaissant plus sans laisser de trace dans la simulation Eulérienne.

CHAPITRE 5

RÉSULTATS

Après avoir présenté en détail notre implémentation, nous nous intéressons aux résultats pratiques de cette dernière. Dans ce chapitre, nous évaluons les apports visuels et la performance du système proposé.

Pour démontrer les fonctionnalités ajoutées, nous utilisons les scènes suivantes :

TABLE 5.I – Caractéristiques des scènes de test.

Figure	Scène	Résolution	Nombre de particules
5.1	Plage	256×256	72k
5.2	Vague	256×256	302k
5.3	Rivière	1024×1024	54k

où la scène "Plage" met en avant les vagues déferlantes, la scène "Vague" teste les limites des systèmes de particules et "Rivière" présente un exemple de simulation de plus haute résolution. Nous réalisons une comparaison visuelle ainsi qu'une capture de temps d'exécution moyen pour chacune de ces scènes. Les figures 5.1, 5.2 et 5.3 illustrent l'impact visuel de ces différentes améliorations sur ces scènes à un moment de la simulation.

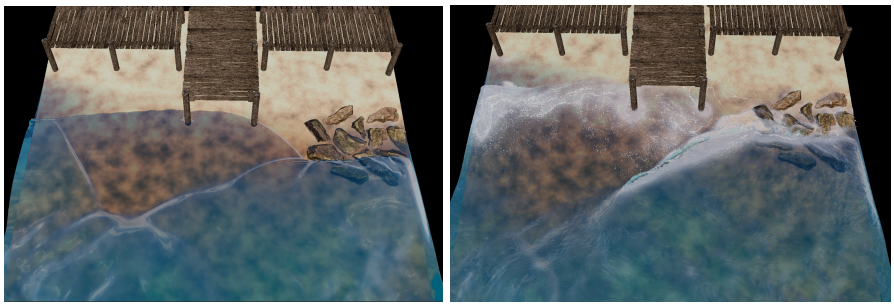


FIGURE 5.1 – Comparaison entre la simulation de base et le système complet sur une scène représentant une plage.

Dans la comparaison de la figure 5.1, on peut observer l'impact des modifications apportées sur la qualité globale de la scène, notamment au niveau des vagues déferlantes et

des éclaboussures. L'ajout de l'écume à base de texture et des particules d'écume permet de générer un niveau de détails supplémentaire qui n'est pas présent dans la simulation de base. Il est également possible d'observer l'effet des détails de surface qui rendent la surface de l'eau plus réaliste.

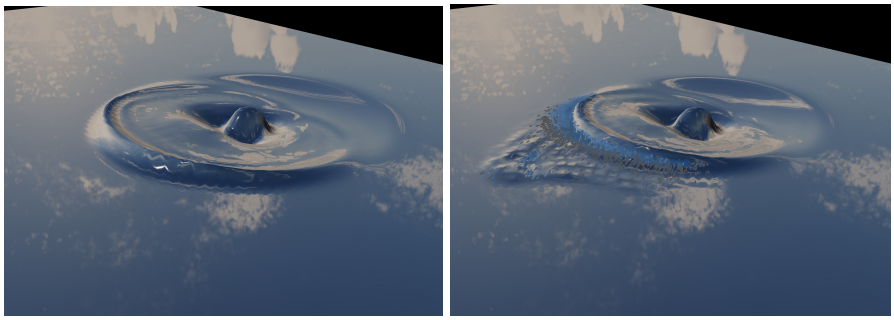


FIGURE 5.2 – Impact du système de particules sur la représentation des vagues déferlantes et des éclaboussures.

Dans la comparaison de la figure 5.2, on peut observer la création d'une vague déferlante ainsi que l'effet du couplage entre les particules et la simulation sur grille, permettant d'ajouter du mouvement et des détails supplémentaires à la scène. On peut noter l'impact du couplage avec les particules de *splash* qui permet la création de vaguelettes à l'avant de la vague.

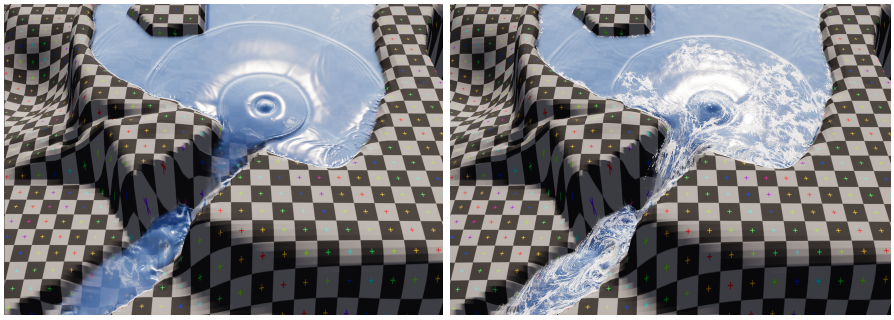


FIGURE 5.3 – Comparaison entre la simulation de base et le système complet sur une scène représentant une rivière avec une cascade.

Dans la comparaison de la figure 5.3, on peut observer l'impact de l'ajout de l'écume sur la scène, notamment au niveau des zones de turbulences au pied de la cascade. L'ajout

de particules d'écume permet de générer un niveau de détails supplémentaire qui n'est pas présent dans la simulation de base. Il est également possible d'observer l'effet des détails de surface qui rendent la surface de l'eau plus réaliste au niveau du bassin supérieur.

Au-delà du gain visuel apporté, il est également important d'évaluer l'impact de ces systèmes sur les performances globales de la simulation. Le tableau 5.II présente les temps d'exécution moyens mesurés sur plusieurs scènes représentatives, en comparant la simulation de base à la version intégrant l'ensemble des améliorations proposées. Les captures de performances ont été réalisées sur un ordinateur équipé d'un GPU NVIDIA GeForce RTX 2070 SUPER, un CPU Intel(R) Xeon(R) W-2255, cadencé à 3.70GHz, ainsi que 64 Go de mémoire vive.

TABLE 5.II – Comparaison des performances du système sur différentes scènes.

Figure	Scène	Solveur de base	Solveur amélioré	Particules	Rendu	Total
5.1	Plage	0.18 ms	0.20 ms	0.31 ms	1.28 ms	1.78 ms
5.2	Vague	0.18 ms	0.35 ms	1.45 ms	1.70 ms	3.5 ms
5.3	Rivière	0.18 ms	0.25 ms	1.20 ms	0.20 ms	1.60 ms

Les mesures obtenues montrent que le coût supplémentaire induit par les systèmes proposés demeure raisonnable au regard du gain visuel obtenu. On observe néanmoins des écarts notables entre les scènes : la scène *Vague*, qui met en jeu environ 300 000 particules, présente les temps les plus élevés, avec 1.45 ms pour la simulation des particules et 1.70 ms pour leur rendu, portant le total à 3.50 ms. Les scènes *Plage* et *Rivière*, dont la population de particules est significativement plus réduite, restent en dessous de 2 ms au total. Malgré l'ajout de plusieurs passes de calcul et de rendu, l'ensemble du système reste compatible avec les contraintes du rendu temps réel.

CHAPITRE 6

CONCLUSION

L'objectif de ce travail était d'étendre les capacités visuelles d'un solveur basé sur les équations d'eau peu profonde tout en conservant les performances nécessaires à une utilisation en temps réel. Bien que ce type de modèle de simulation soit particulièrement efficace pour représenter les écoulements de grande échelle, il demeure limité lorsqu'il s'agit de reproduire des phénomènes tels que les vagues déferlantes, les projections de fluide ou encore l'écume présente à la surface de l'eau. Dans le cadre de ce projet, nous nous sommes appuyés sur une implémentation du solveur de base développée au sein d'un moteur propriétaire reposant sur DirectX 12 et Vulkan.

Afin de pallier ces limitations, plusieurs extensions ont été développées. Un système hybride combinant simulation sur grille et particules a tout d'abord été introduit afin de représenter les particules de *splash* et de *spray*. Cette approche permet de générer des détails qui ne peuvent être correctement représentés par une simulation reposant uniquement sur un champ de hauteur. Des techniques de rendu adaptées ont ensuite été mises en œuvre afin de reconstruire une surface de liquide cohérente à partir des particules et d'améliorer son apparence à l'aide de modèles optiques inspirés des propriétés de l'eau. Enfin, un système d'écume hybride, reposant à la fois sur une pseudo-simulation sur grille et sur des particules dédiées, a été proposé afin d'enrichir davantage le niveau de détails observables.

Certaines optimisations demeurent envisageables, notamment au niveau des mécanismes de couplage entre les différentes représentations du fluide, du type de variables utilisées pour l'encodage des données de simulation (*float 16* au lieu de *float 32*, par exemple), la concaténation de certaines passes de calcul ainsi que la gestion du rendu des particules, qui reste un système multi-passes pouvant s'avérer coûteux. Ces améliorations relèvent toutefois davantage d'un travail d'optimisation pouvant être réalisé par

une équipe de production que d'une remise en question de l'architecture générale du système.

Il est important de mentionner que le portage de ce système du moteur de prototypage vers un engin de production nécessite un travail d'intégration conséquent. En effet, les différences d'architecture des différents moteurs imposent de repenser certaines parties du système afin de garantir une compatibilité avec les pipelines de rendu, les systèmes de gestion des ressources et les outils de production existants. De ce fait, une collaboration étroite avec les équipes de production est essentielle pour assurer une intégration réussie.

L'un des principaux défis observés lors du développement concerne le paramétrage des différents systèmes. Certains paramètres, tels que ceux contrôlant l'émission des particules lors du déferlement des vagues ou encore les seuils utilisés pour la génération d'écume, peuvent être difficiles à interpréter pour les utilisateurs. Bien que ces réglages offrent une grande flexibilité artistique, ils gagneraient à être accompagnés d'outils de visualisation ou d'interfaces plus adaptées afin de faciliter leur ajustement et de réduire le temps nécessaire à l'obtention du résultat souhaité, surtout lorsque ces effets sont combinés. On pourrait par exemple imaginer le support de contrôle local en permettant aux artistes de "peindre" par dessus la simulation pour indiquer, par exemple, des zones où sont générées des particules d'écume (ou dérivé d'après un comportement similaire). Enfin, il est important de noter que le sujet d'une meilleure connexion de la simulation avec les outils procéduraux actuels est une problématique qui mérite d'être soulevée.

Plusieurs pistes d'améliorations peuvent également être envisagées. Le système de particules d'écume pourrait notamment être enrichi par une simulation plus avancée prenant en compte des phénomènes tels que la flottabilité, les échanges de quantité de mouvement ou les interactions entre particules (bien que ce dernier élément puisse s'avérer bien plus coûteux à mettre en place, dû à la nécessité d'implémenter un mécanisme de recherche de voisins). De manière plus générale, l'exploration de méthodes hybrides reposant sur des représentations de type *tall grid cells* [3] constitue une direction particulièe-

rement prometteuse. Ces approches permettraient de conserver l'efficacité des modèles d'eau peu profonde tout en réduisant certaines des limitations associées aux représentations fondées uniquement sur un champ de hauteur. Une grille de simulation de taille adaptative pourrait également améliorer certains détails de niveau intermédiaire.

En définitive, les résultats obtenus démontrent qu'il est possible d'améliorer significativement la richesse visuelle d'une simulation d'eau peu profonde à l'aide de techniques procédurales adaptées aux architectures graphiques modernes. Les méthodes présentées permettent de reproduire efficacement des phénomènes qui échappent traditionnellement aux modèles basés sur des colonnes d'eau, tout en maintenant des performances compatibles avec les exigences du rendu temps réel. Bien que plusieurs axes de recherche demeurent ouverts, les travaux réalisés constituent une base solide pour la réalisation d'effets de simulation crédibles pour des productions de jeux vidéo modernes.

BIBLIOGRAPHIE

[1] George Biddell Airy. Tides and waves. In H. J. Rose, editor, *Encyclopaedia Metropolitana*, volume 3, pages 241–396. B. Fellowes, London, 1841.

[2] Nuttapong Chentanez and Matthias Müller. Real-time simulation of large bodies of water with small scale details. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, page 197–206. Eurographics Association, 2010. URL https://matthias-research.github.io/pages/publications/hfFluid.pdf.

[3] Nuttapong Chentanez and Matthias Müller. Real-time Eulerian water simulation using a restricted tall cell grid. *ACM Trans. Graph.*, 30(4), July 2011. ISSN 0730-0301. doi : 10.1145/2010324.1964977. URL https://doi.org/10.1145/2010324.1964977.

[4] Michal Chládek and Roman Ďurikovič. Particle-based shallow water simulation for irregular and sparse simulation domains. *Computer & Graphics*, 53(PB) :170–176, December 2015. ISSN 0097-8493. doi : 10.1016/j.cag.2015.04.002. URL https://doi.org/10.1016/j.cag.2015.04.002.

[5] Makoto Fujisawa, Takuya Nakada, and Masahiko Mikawa. Particle-based shallow water simulation with splashes and breaking waves. *Journal of Information Processing*, 25 :486–493, 07 2017. doi : 10.2197/ipsjjip.25.486.

[6] Simon Green. Screen space fluid rendering for games. *Game Developer Conference*, 2010. URL https://developer.download.nvidia.com/presentations/2010/gdc/Direct3D_Effects.pdf.

[7] Markus Ihmsen, Nadir Akinci, Gizem Akinci, and Matthias Teschner. Unified spray, foam and air bubbles for particle-based fluids. *The Visual Computer*, 28 : 669–677, 2012. doi : 10.1007/s00371-012-0697-9.

[8] Stefan Jeschke and Chris Wojtan. Generalizing shallow water simulations with dispersive surface waves. *ACM Trans. Graph.*, 42(4), July 2023. ISSN 0730-0301. doi : 10.1145/3592098. URL https://doi.org/10.1145/3592098.

[9] Wei Li, Haozhe Su, Zherong Pan, Xifeng Gao, Zhenyu Mao, and Kui Wu. Flowmap baking with LBM-SWE. 2023. URL https://api.semanticscholar.org/CorpusID:275589669.

[10] Zhenyu Mao and Kui Wu. Open world water rendering and real-time simulation. *Game Developer Conference*, 2023. URL https://media.gdcvault.com/gdc2023/Slides/Open-World+Water+Rendering+and+Real-Time+Simulation_Mao_Zhenyu%26Wu_Kui.pdf.

[11] Varun Ramakrishnan and Tim McGraw. Water animation using coupled sph and wave equation. In *Advances in Visual Computing : 18th International Symposium, ISVC 2023, Proceedings, Part I*, page 303–314. Springer-Verlag, 2023. ISBN 978-3-031-47968-7. doi : 10.1007/978-3-031-47969-4_24. URL https://doi.org/10.1007/978-3-031-47969-4_24.

[12] Barbara Solenthaler, Peter Bucher, Nuttapong Chentanez, Matthias Müller, and Markus Gross. SPH based shallow water simulation. *VRIPHYS 2011 - 8th Workshop on Virtual Reality Interactions and Physical Simulations*, pages 39–46, 2011. doi : 10.2312/PE/vriphys/vriphys11/039-046.

[13] Nils Thürey, Matthias Müller, Simon Schirm, and Markus Gross. Real-time breaking waves for shallow water simulations. *Proceedings - Pacific Conference on Computer Graphics and Applications*, pages 39–46, 2007. doi : 10.1109/PG.2007.33.

Annexe I

Dérivation de la variation de la hauteur de l'eau dans le contexte des équations de *Shallow Water*

La hauteur de l'eau $d(x,z,t)$ dans le modèle d'eau peu profonde est définie comme la distance verticale entre la surface libre de l'eau $h(x,z,t)$ et la topographie du terrain au fond de l'eau $b(x,z)$:

$$d(x,z,t) = h(x,z,t) - b(x,z).$$

Pour dériver une équation régissant l'évolution de cette hauteur, on s'appuie sur le principe de conservation de la masse. Dans notre cas, cela correspond aux conditions d'incompressibilité

$$\nabla \cdot \tilde{\mathbf{u}} = 0$$

en développant cette condition dans les trois directions de l'espace :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0.$$

Afin d'obtenir une expression régissant l'évolution de la hauteur de l'eau, on commence par intégrer l'équation d'incompressibilité sur l'axe vertical, entre le fond $y = b(x,z)$ et la surface libre $y = h(x,z,t)$:

$$\int_{b(x,z)}^{h(x,z,t)} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) dy = 0.$$

Comme les dérivées spatiales $\frac{\partial u}{\partial x}$ et $\frac{\partial w}{\partial z}$ ne dépendent pas directement de y, on peut sortir ces dérivées de l'intégrale et intégrer seulement sur y, ce qui donne :

$$\frac{\partial}{\partial x} \int_b^h u \, dy + \frac{\partial}{\partial z} \int_b^h w \, dy + [v]_b^h = 0.$$

Or, $v(h)$ représente la vitesse verticale de la surface libre, c'est-à-dire :

$$v(h) = \frac{\partial d}{\partial t},$$

et sous l'hypothèse d'un fond fixe, $v(b) = 0$. Ainsi, on obtient :

$$\frac{\partial d}{\partial t} + \frac{\partial}{\partial x}(du) + \frac{\partial}{\partial z}(dw) = 0.$$

Ce qui nous donne au final :

$$\frac{\partial d}{\partial t} = -\nabla \cdot (d\vec{u}).$$