

# Un nouveau modèle pour la génération et le contrôle de vagues déferlantes

Mathias Brousset<sup>1</sup> Emmanuelle Darles<sup>1</sup> Daniel Meneveau<sup>1</sup> Pierre Poulin<sup>2</sup> Benoît Crespin<sup>3</sup>

<sup>1</sup> XLIM, UMR CNRS 7252, Université de Poitiers <sup>2</sup> Dép. I.R.O., Université de Montréal <sup>3</sup> XLIM, UMR CNRS 7252, Université de Limoges

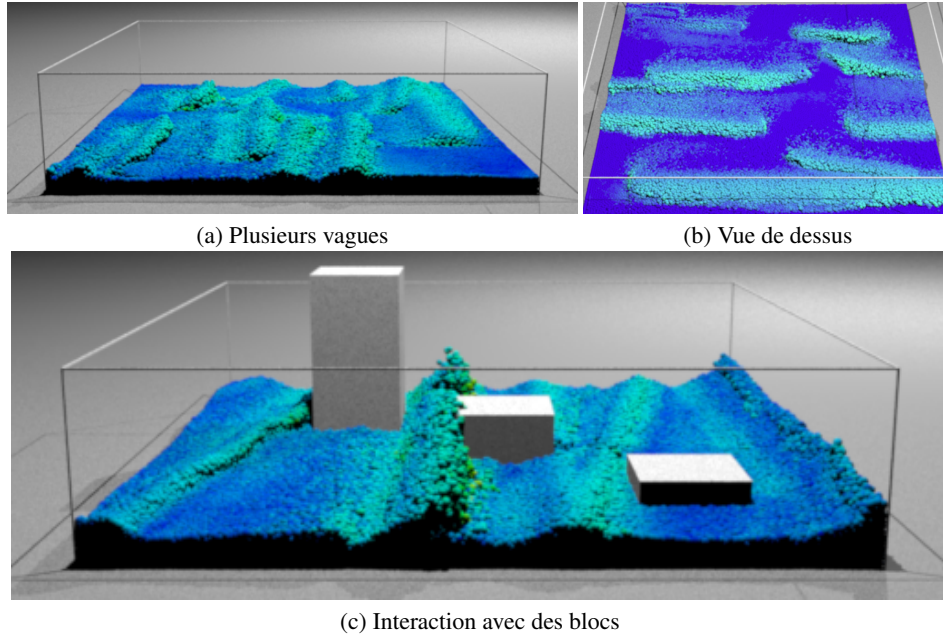


Figure 1: Vagues déferlantes obtenues avec notre modèle: (a) un océan avec plusieurs vagues, vue de profil; (b) la même scène vue de dessus; (c) le reflux des vagues contre des blocs. Les vagues sont combinées et chacune d'entre elle comporte ses propres paramètres (hauteur, largeur, vitesse, orientation, inclinaison de la crête, temps de déferlement)

## Résumé

Cet article présente une nouvelle méthode pour générer et contrôler de la houle ainsi que des vagues déferlantes en utilisant un simulateur de fluides SPH (Smoothed Particle Hydrodynamics). Les approches conventionnelles consistent à utiliser un mur en mouvement pour donner une accélération aux particules, mais le contrôle des vagues est difficile à assurer. Cet article décrit un modèle de forces décrivant physiquement la formation et la propagation des vagues. Nous montrons que son utilisation permet d'obtenir de nombreux types de vagues, et de contrôler de manière intuitive la forme, la vitesse ou encore l'instant du déferlement des vagues. Il est basé sur une représentation en deux dimensions seulement, avec des paramètres de vitesse, de largeur ou encore de hauteur des vagues.

## Abstract

This paper presents a new method for controlling swells and breaking waves using smoothed particle hydrodynamics (SPH). With conventional approaches that generate waves by pushing particles with oscillating planes, the resulting waves cannot be controlled easily, and breaking waves are even difficult to obtain in practice. Instead, we propose to use a wave model that physically describes the behavior of wave forces. We show that mapping those forces to particles produces various types of waves that can be controlled by the user with only a few parameters. Our method is based on a 2D representation that describes wave speed, width, and height. It handles many swell and wave configurations, with various breaking situations.

**Mots clé:** Informatique graphique, animation basée physique, simulation de fluides, SPH

## 1. Introduction

Au cours de la dernière décennie, la simulation de fluides a suscité un intérêt croissant de la part de la communauté informatique graphique. Plusieurs familles de méthodes permettent de simuler des mouvements de fluides de manière réaliste. La plupart d'entre elles approchent les équations de Navier-Stokes, à partir de représentations soit Eulériennes, soit Lagrangiennes. Les résultats obtenus sont réalistes, mais pâtissent de temps de calculs importants, de forte consommation en mémoire, ainsi que d'une faible possibilité de contrôle utilisateur.

Les vagues produites par de vastes étendues d'eau donnent une apparence particulière à notre perception des côtes marines, mais générer des conditions initiales permettant à des solveurs de réaliser des vagues réalistes reste un défi difficile à relever. Notre objectif est de proposer des outils simples pour créer et contrôler différents types de vagues à l'aide d'un modèle pouvant être intégré à n'importe quel type de simulateur de fluides.

Les vagues sont des exemples de phénomènes turbulents souvent utilisés comme résultats démonstratifs par la communauté scientifique. Dans les méthodes basées sur la physique, elles sont la plupart du temps générées à l'aide d'un plan en mouvement poussant les particules, ce qui rend le contrôle par l'utilisateur peu aisé.

Peu d'auteurs ont abordé la génération de vagues de manière physique tout en offrant une possibilité de contrôle artistique, que ce soit avec une paramétrisation intuitive ou avec une intervention dynamique de l'utilisateur.

L'approche proposée par Mihalef et al. [MMS04] aborde ce problème en utilisant une bibliothèque de vagues déferlantes pré-existantes en deux dimensions. Avec ce système, l'utilisateur sélectionne un ensemble de profils de vagues déferlantes et anime la vague 3D avec un solveur Eulérien. La modélisation et le contrôle de la vague dépendent d'un ensemble de profils 2D et de leurs propriétés physiques, mais aucune interaction n'est possible pendant la simulation et le résultat obtenu est difficile à prévoir par un artiste.

Un autre travail intéressant dans ce domaine est celui de Radovitzky et Ortiz [RO98], basé sur une solution de Laitone en 2D, qui représente une vague déferlante à l'aide de fonctions périodiques hyperboliques, avec pour forme celle d'une vague plongeante. Malheureusement ce modèle requiert beaucoup de paramètres, et contrôler les vagues pendant la phase de simulation n'est pas réellement possible.

Notre modèle a pour objectif de faciliter le contrôle utilisateur et de permettre de simuler la propagation de plusieurs types de vagues, à l'aide d'une force extérieure, peu coûteuse en termes de ressources de calcul ou de mémoire, et définie par quelques paramètres pouvant être modifiés de manière dynamique. La figure 1 illustre quelques résultats obtenus grâce à notre modèle. Plus précisément, nos contributions sont les suivantes :

- un modèle de vague simple et basé sur la physique, pouvant représenter des vagues turbulentes, linéaires ou déferlantes, très facile à implémenter car il correspond seulement à une force externe supplémentaire ;
- la possibilité pour un artiste de contrôler la forme et le comportement des vagues générées, grâce à la simplicité du modèle ;
- un contrôle dynamique des vagues durant la phase de simulation.

L'article est organisé en six sections. La section 2 décrit les méthodes existantes pour la simulation de vagues en informatique graphique. La section 3 présente notre modèle et ses différents paramètres de contrôle. La section 4 décrit notre implémentation et présente divers types de résultats. Enfin, la section 5 conclut et présente plusieurs perspectives possibles à ce travail.

## 2. Travaux antérieurs

La simulation de vagues océaniques par ordinateur a suscité l'attention de nombreux auteurs scientifiques [FR86, Tes99, MMS04, DCGG11, NSB13]. Les méthodes existantes peuvent être classifiées selon deux familles : d'une part, les méthodes procédurales (ou spectrales) qui représentent la surface de l'eau et sa hauteur en fonction du temps ; et d'autre part, les méthodes physiques dont l'objectif est d'approcher les équations de Navier-Stokes (NS), en 2D ou 3D.

### 2.1. Animation procédurale de vagues

Pour des océans de grande profondeur, les méthodes procédurales ou spectrales sont souvent utilisées. Le principe est de considérer la surface de l'eau à l'aide d'un champ de hauteurs, déterminées suivant des fonctions périodiques. Leur efficacité en termes de réalisme visuel et de temps de calcul les rend très populaires, bien qu'elles ne reposent pas directement sur les équations de la mécanique des fluides.

Certaines d'entre elles [FR86, Pea86] utilisent des équations paramétriques pour obtenir la surface du liquide, incluant non seulement les phénomènes de houle mais aussi les vagues déferlantes.

Tessendorf [Tes99] anime un champ de hauteurs en utilisant des transformées de Fourier rapides (FFT), auxquelles Bruneton et al. [BNH10] ajoutent du niveau de détail afin d'effectuer un rendu réaliste de l'océan en temps interactif. L'ajustement des paramètres pour ces méthodes étant difficile, Thon et Ghazanfarpour [TG02] proposent d'utiliser des mesures réelles comme paramètres, et les auteurs ajoutent également un bruit de Perlin pour obtenir certains effets visuels. La principale limitation de ces méthodes est qu'elles ne peuvent pas représenter de vagues déferlantes car chaque position horizontale  $(x, y)$  ne peut comporter qu'une seule hauteur  $z$ . De plus, elles restent malgré tout difficiles à paramétrer pour une interaction avec des objets solides.

## 2.2. Vagues et équations de Navier-Stokes

De nombreux auteurs s'intéressent à la simulation de fluides sur la base des équations de Navier-Stokes. Ces simulations numériques utilisent en général une description Eulérienne et des méthodes de différences finies pour approcher une solution [HW65, FM96]. Certaines d'entre elles ajoutent des particules et des surfaces de niveau pour modéliser le volume du fluide et pour réduire les phénomènes de compressibilité [Sta99, FF01, FSJ01, EMF02]. Les méthodes SPH reposent sur une représentation Lagrangienne et modélisent le fluide à l'aide d'un ensemble de particules pour calculer les interactions [MCG03, BT07, SR09, PTB\*03, IOS\*14].

Certaines approches produisent des vagues océaniques en faisant l'hypothèse d'une profondeur finie. À partir d'une librairie de vagues simulées en 2D, Mihalef et al. [MMS04] génèrent des vagues en 3D. Leur méthode consiste à combiner un ensemble de profils 2D pour modéliser une vague à un temps donné ; les vitesses entre les profils sont interpolées linéairement. Dans leur outil, chaque profil correspond à une capture d'une simulation Eulérienne 2D avec ses vecteurs de vitesse associés. En partant d'une géométrie 3D initiale, pouvant être formée comme de la houle ou comme une vague déferlante, la forme de la vague modélisée est finalement obtenue au cours de la simulation. Du fait que chaque profil contient des informations sur la dynamique de la vague (telles que la vitesse), contrôler la vague pendant la phase de simulation n'est pas aisé.

Avec les équations de Saint-Venant (une simplification des équations NS pour les milieux peu profonds), Thurey et al. [TMFSG07] reproduisent des déferlantes en temps interactif. Les vagues sont produites par le mouvement d'un plan dans le volume d'eau. La hauteur de la vague et sa vitesse de propagation peuvent être contrôlées, et la vague déferlante est complétée grâce à un maillage supplémentaire, mais les interactions entre le volume d'eau et des objets solides ne peuvent être simulées de manière physique.

D'après des études océanographiques, la vitesse d'une simulation 2D des équations de NS peut être initialisée avec une combinaison de fonctions hyperboliques représentant la propagation d'une vague solitaire [RO98]. L'idée est de combiner les composantes horizontale et verticale d'une force non linéaire [BGH\*04] :

$$y = H \operatorname{sech}(x - \omega t) \tanh(x - \omega t),$$

où  $H$  représente la hauteur de la vague, et  $\omega$ , la pulsation, représente sa vitesse de propagation. Malheureusement, le contrôle de la vague demeure difficile, et les vagues résultantes sont uniquement valides pour de faibles profondeurs. En outre, le déferlement de la vague est provoqué uniquement par l'élévation du sol, ce qui limite leur utilisation à certaines topographies de fonds sous-marins. Enfin, le calcul des interactions entre plusieurs de ces vagues solitaires requiert d'approcher les équations de Korteweg-de Vries, ce qui est coûteux pour des applications interactives.

Darles et al. [DCG11] étendent ce modèle pour des vagues déferlantes en 3D en ajoutant une force procédurale à un solveur SPH multi-échelle. Cette méthode produit plusieurs types de vagues plongeantes et glissantes avec des hauteurs variables. En revanche, le modèle est très dépendant de la profondeur du fluide, et le déferlement est contrôlé uniquement par la topographie du terrain sous-marin.

Dans cet article, nous proposons une nouvelle méthode pour réduire ces limitations et simplifier le paramétrage des vagues. Nous montrons comment les combiner et les contrôler, et comment notre modèle peut être utilisé pour produire des vagues qui se propagent dans plusieurs directions et qui interagissent entre elles. Les mouvements de houle peuvent être décrits avec des vitesses et des hauteurs variables, et les déferlements sont contrôlables, quelle que soit la profondeur du fluide. Notre travail peut être vu comme une extension et une simplification du modèle proposé par Darles et al. [DCG11], basé sur une force externe hautement contrôlable et facile à intégrer dans un système de simulation SPH.

## 2.3. Équations gouvernantes et méthode SPH

Nous utilisons la forme Lagrangienne des équations de NS, qui décrivent l'écoulement d'un fluide représenté par des particules. La forme Lagrangienne de l'équation du bilan de quantité de mouvement est la suivante :

$$\frac{d\vec{v}_i}{dt} = -\frac{1}{\rho_i} \nabla p_i + \nu \nabla^2 \vec{v}_i + \frac{\vec{F}_i^{ext}}{\rho_i}, \quad (1)$$

$$\frac{d\vec{x}_i}{dt} = \vec{v}_i, \quad (2)$$

où, pour une particule  $i$ ,  $\vec{x}_i$  correspond à sa position,  $\vec{v}_i$  à sa vitesse,  $\rho_i$  représente sa densité,  $p_i$  sa pression,  $\nu$  sa viscosité cinématique, et  $\vec{F}_i^{ext}$  les forces externes. L'équation 1 stipule que, dans le cas Lagrangien, l'accélération d'une particule  $i$  à chaque pas de temps dépend d'une somme de forces internes (pression et viscosité) et externes (gravité et, dans notre cas, des forces de propagation décrites dans la section 3).

Sur la base de cette formulation, les méthodes de type SPH ont été largement utilisées en dynamique numérique des fluides pour approcher les termes différentiels de l'équation 1. Pour chaque particule  $i$ , une quantité donnée  $A$  est interpolée en utilisant les particules voisines  $j$ , avec la formulation suivante [Mon92, Mon05] :

$$A_i = \sum_j \frac{m_j}{\rho_j} A_j W(\vec{x}_i - \vec{x}_j, h), \quad (3)$$

où  $m_j$  représente la masse d'une particule  $j$ ,  $h$  la distance maximale d'interaction entre deux particules voisines, et  $W$  une fonction de distribution radiale avec support compact  $h$ . Des formulations symétriques peuvent être utilisées pour approcher le gradient ou le Laplacien d'un champ  $A$  telles que le gradient de pression et le Laplacien de vitesse, pour approcher une solution à l'équation 1 [IOS\*14].

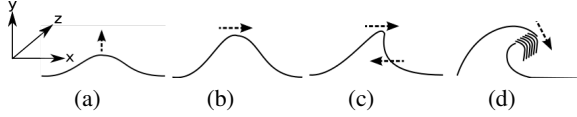


Figure 2: Les quatre phases d'une vague déferlante : (a) la levée, (b) la propagation, (c) la formation du déferlement, et (d) le déferlement.

### 3. Un nouveau modèle générique de vagues

Notre modèle de vague simplifie et généralise la représentation du soliton proposée par Darles et al. [DCG11]. Elle repose sur l'ajout de forces externes aux particules de fluide, grâce à plusieurs paramètres contrôlant la houle et les vagues déferlantes, tels que la hauteur, la vitesse, le moment du déferlement et la durée du déferlement. Nous montrons également comment ce modèle peut être étendu pour modifier l'orientation de la vague, pour faire varier sa crête et pour simuler le croisement de vagues.

#### 3.1. Notre modèle 2D de force de vagues non linéaires

De la houle au déferlement, quatre phases peuvent être identifiées lors de la propagation d'une vague (figure 2) [Kel48] : (a) le fluide s'élève et la vague apparaît ; (b) la vague se propage, et la houle résultante peut varier selon certains paramètres, comme la profondeur de l'eau ou la vitesse du vent ; (c) le déferlement se forme, à cause de l'élévation du fond sous-marin ou encore d'un changement de sens du courant marin ; (d) la vague déferle du fait que les particules au niveau de la crête de la vague ont une vitesse plus élevée que les autres particules.

Durant la phase de houle d'une vague, un mouvement périodique à la surface peut être observé. Les particules du fluide suivent alors un mouvement elliptique. D'après la théorie d'Airy en profondeur finie [WJ10] ; la vitesse verticale des particules augmente lorsque la particule se rapproche de la surface de l'eau. Plus précisément, lorsque la vague avance, chaque particule est assujettie à une accélération verticale et une accélération horizontale qui donnent le déplacement des particules dans la direction de la vague. Lorsque la particule n'est plus dans le domaine de l'onde, elle finit par retomber et les forces résiduelles de reflux provoquent une accélération dans le sens inverse.

Notre modèle permet d'approcher ces effets en utilisant des forces externes à l'intérieur d'un simulateur de fluides. Elles reposent sur une composante horizontale permettant le déplacement de la vague, et sur une composante verticale permettant son élévation. La vitesse de propagation de la vague est décrite par une pulsation  $\omega$  fournie par l'utilisateur ; la phase de levée consiste à initialiser les forces horizontales et verticales au temps  $t = 0$ . Les forces appliquées

sur une particule  $i$  sont définies de la manière suivante :

$$F_{x_i} = -\sqrt{gd} \operatorname{sech}(A d_{z_i}) \tanh(A d_{z_i}) \lambda_x, \quad (4)$$

$$F_{z_i} = \begin{cases} H \operatorname{sech}^2(A(x_i - \omega t - x_{shoal})) \lambda_z & \text{if } x_i < x_{break} \\ 0 & \text{sinon.} \end{cases} \quad (5)$$

avec  $A = \sqrt{3H/4d^3}$ . Dans ce système,  $H$  est la hauteur de la vague définie par l'utilisateur,  $t$  est l'instant courant,  $\omega$  est la vitesse de propagation de la vague,  $d_{z_i} = z_i - z_i^0$  est la profondeur relative de la particule  $i$ ,  $z_i^0$  est la profondeur initiale de la particule, et  $\lambda_x$  et  $\lambda_z$  sont deux paramètres utilisateurs permettant de contrôler la forme de la vague.  $\sqrt{gd}$  est un terme d'atténuation proportionnel à la gravité [RO98], et  $\omega t$  (équation 5) détermine la position de la vague.

La figure 3 montre la force horizontale (équation 4) appliquée à une particule quelconque ; l'axe des abscisses représente la profondeur de la particule  $d_z$ . Ces courbes atteignent leur maximum lorsque la particule est proche de la crête de la vague. La figure 4 montre la force verticale (équation 5). Cette fonction atteint son maximum lorsque la particule est proche du point de levée et est annulée lorsque la particule atteint le point de déferlement.

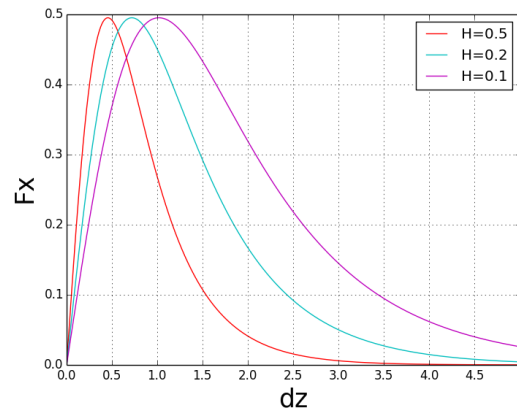


Figure 3: La représentation de la composante horizontale de notre modèle de force  $F_x$ , agissant sur une particule en fonction de sa profondeur  $d_z$  pour différentes valeurs de  $H$ .

Chacune des quatre phases d'une vague peut être distinguée à l'aide de notre formulation :

**Levée** : le point de levée  $x_{shoal}$  représente l'origine d'une vague ; les forces sont initialisées à  $t = 0$ , et débutent à la position  $x_{shoal}$ . La force verticale est relativement grande dans le but d'obtenir une élévation du volume d'eau (partie bleue de la courbe sur la figure 4 (en haut))

**Propagation de la houle** : horizontalement, lorsqu'une vague atteint une particule de position  $x$ , les particules subissent une accélération de manière à obtenir un mouvement elliptique décrit par la théorie d'Airy [BGH\*04], grâce à l'équation 4. Plus une particule est proche de la surface, plus cette force sera importante. Le mouvement de retour subi par une

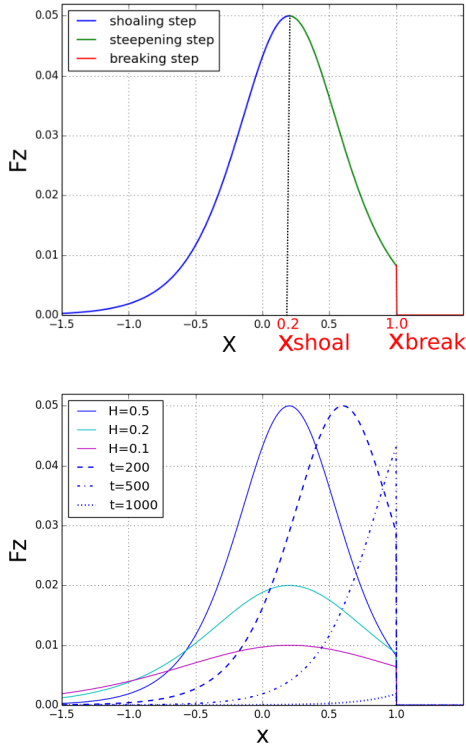


Figure 4: En haut : représentation de la composante verticale  $F_z$  de la force agissant sur une particule avec  $H = 0.5$ ,  $\omega = 0.002$ ,  $t = 0$ , et  $\lambda_z = 0.1$ . En bas : représentation de la composante verticale de force  $F_z$  pour différentes valeurs de  $H$  à différents temps  $t$ .

particule est dû au reflux occasionné par la simulation SPH lorsque le domaine de l'onde dépasse la particule. Verticalement, la force augmente progressivement de manière à provoquer la montée de la particule (mouvement elliptique vertical), fourni par notre modèle grâce à la combinaison de tangentes et sécantes hyperboliques, diminuant de manière linéaire en fonction de la profondeur de la particule (équation 5).

**Formation du déferlement :** la formation du déferlement correspond à une augmentation de la composante verticale des forces de la vague lorsque la position horizontale  $x$  de la particule est proche du point  $x_{shoal}$ . Lorsque la composante horizontale de la force devient plus grande au niveau des particules de la crête, la vague commence à prendre sa forme déferlante. Le déferlement peut alors avoir lieu (partie verte de la première courbe sur la figure 4).

**Déferlement :** le déferlement est dû à la discontinuité de la force dans notre modèle : la vitesse horizontale de la crête est supérieure à celle des particules situées en-dessous, ce qui leur permet de retomber. Cette discontinuité est modélisée

par une annulation (ou une atténuation) de la force verticale (partie rouge de la courbe sur la figure 4 (en haut)).

### Discussion

Les courbes de la figure 3 illustrent la composante horizontale de la force pour trois valeurs de  $H$ . Pour des valeurs faibles de  $H$ , la force atteint une valeur maximale sur une hauteur importante du volume d'eau, permettant de faire avancer la vague. Pour des valeurs plus grandes de  $H$ , la hauteur de l'eau est plus petite et plutôt à la surface, produisant ainsi une différence de vitesses qui conduit au déferlement.

Les courbes de la figure 4 (en haut) montrent l'évolution de  $F_z$  pour différentes valeurs de  $H$  et de  $t$ . Premièrement, la force verticale varie en fonction de  $H$  : avec de faibles valeurs, sa magnitude est faible de même que les variations durant les phases de formation et de déferlement. Dans cette configuration, la houle produite ne permet pas d'obtenir un déferlement et le mouvement des particules résulte principalement de la composante horizontale de la force. La simulation de vagues déferlantes nécessite des valeurs de  $H$  adaptées. De plus, la diminution de la force verticale (qui décrit les phases de levée et de début de déferlement) varie au cours du temps. Plus la valeur de  $t$  augmente, plus la part décroissante de la courbe correspondant au début du déferlement se réduit. Pour de faibles valeurs de  $t$ , cette phase permet aux particules de se lever et de produire un déferlement.

Avec cette représentation, la composante horizontale de la force n'est pas corrélée à la composante verticale. Il est donc possible de produire des vagues de faible amplitude, ou des vagues déferlantes comme cela est présenté dans les prochaines sections. La formule de la composante verticale garantit une croissance linéaire de la vague, qui peut être contrôlée au cours du temps et dans l'espace, en faisant varier les paramètres  $H$ ,  $\omega$ ,  $\lambda_x$  et  $\lambda_z$ .

### 3.2. Contrôle des paramètres

Cette section décrit comment les paramètres de notre modèle peuvent être utilisés pour contrôler localement ou globalement le comportement et la forme de la vague. Cela inclut le contrôle des paramètres de hauteur  $H$ , de pulsation  $\omega$ , de point de levée  $x_{shoal}$  et de point de déferlement  $x_{break}$ . Modifier les paramètres  $\lambda_x$  et  $\lambda_z$  (considérés comme des paramètres globaux additionnels) permet d'obtenir différents types de vagues.

#### Hauteur ( $H$ )

Ce paramètre affecte l'amplitude de la vague (sa hauteur) et sa forme, dérivée de l'équation 4.

Par exemple, de la houle peut être obtenue avec  $H < 0.2$ , alors que des valeurs plus grandes peuvent permettre d'obtenir des vagues déferlantes, comme sur les figures 5 et 6.

#### Pulsation de la vague ( $\omega$ )

La pulsation de la vague contrôle sa vitesse de propagation



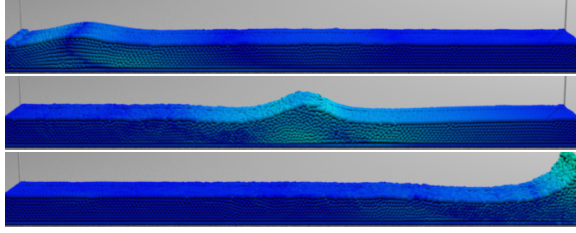


Figure 5: La levée et la propagation d'une houle simple avant qu'elle n'interagisse avec le mur. ( $H = 0.22$ ,  $\omega = 0.002$ ,  $d = 0.2$ ,  $\lambda_x = 1.0$ ,  $\lambda_z = 32.5$ ).

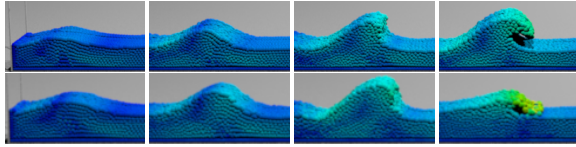


Figure 6: En haut : les quatre phases typiques d'une vague déferlante, qui sont la levée, la propagation, la formation du déferlement, et le déferlement ( $H = 0.24$ ,  $\omega = 0.002$ ,  $d = 0.2$ ,  $\lambda_x = 1.0$ ,  $\lambda_z = 32.5$ ). En bas : vague déferlante obtenue en utilisant une valeur linéairement décroissante de  $H$  jusqu'à 0.

et la durée de la formation du déferlement (voir la figure 7). Plus cette phase est longue, plus les particules de la crête seront hautes. Grâce à ce paramètre, il est possible de contrôler la forme et le type de la vague simulée. Par exemple de petites valeurs ( $\omega < 0.002$ ) permettent de produire des vagues plongeantes, des valeurs intermédiaires ( $0.002 < \omega < 0.003$ ) donnent des vagues glissantes, et des valeurs plus grandes permettent d'obtenir de la houle, qui se produit principalement en eaux profondes.

Combiner des vagues avec différentes valeurs de  $\omega$  permet de représenter des croisements de vagues non uniformes que l'on peut observer sur certaines côtes.

#### Mise à l'échelle de la force des vagues ( $\lambda_x$ et $\lambda_z$ )

Ces paramètres permettent de contrôler globalement le comportement de la vague, en agissant indépendamment dans les directions  $x$  et  $z$ . Ainsi, la levée et le déferlement peuvent être affectés par une modification de  $\lambda_z$  au cours du temps, le choix d'une valeur  $\lambda_z \gg \lambda_x$  garantissant le déferlement. De façon similaire la vitesse de propagation et la direction sont affectées par une modification de  $\lambda_x$  au cours du temps.

#### Point de levée et de déferlement ( $x_{shoal}$ et $x_{break}$ )

Ces deux points permettent à l'utilisateur de définir le point de démarrage  $x_{shoal}$  de la vague, et l'endroit de la scène  $x_{break}$  où celle-ci déferle.

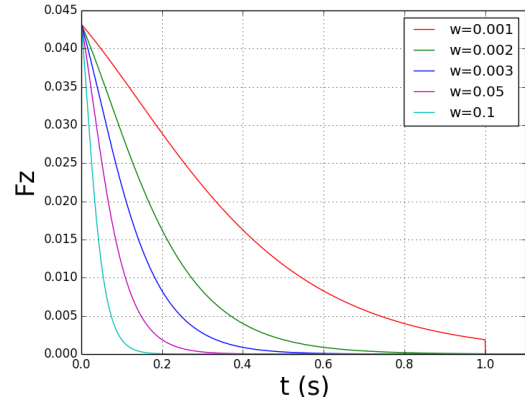


Figure 7: La durée de la phase de début de déferlement pour la composante verticale de la force avec des valeurs différentes de  $\omega$ , évaluée sur une particule située en  $x = 0$  avec  $H = 0.5$ ,  $x_{shoal} = 0.2$ , and  $x_{break} = 1.0$ .

### 3.3. Extension du modèle en 3D

Le modèle décrit précédemment est défini pour une propagation en 1D, le long des abscisses, le paramètre  $z$  correspondant à la hauteur. Dans cette section nous montrons comment nous avons étendu notre modèle pour une propagation en 2D, afin d'avoir un plus grand nombre de configurations.

#### Orientation des vagues

Avec une direction de propagation  $\theta$  donnée dans le plan  $xy$ , la rotation correspondante peut être appliquée sur les composantes des forces et les particules correspondantes, ce qui permet d'avoir une orientation de vague quelconque dans ce plan.

#### Variation de la crête

Un autre exemple de paramétrage de la vague consiste à appliquer plusieurs valeurs de  $H$  dans le but de faire varier la crête de la vague. À l'aide de notre interface graphique, l'utilisateur peut tracer une ligne brisée qui représente la valeur de  $H$  sur la longueur de la crête durant la simulation. La figure 8 illustre une vague ayant une crête oblique.

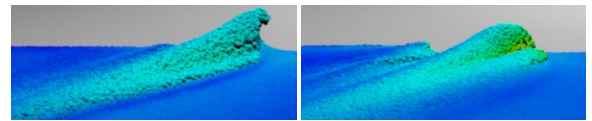


Figure 8: Rotation des particules pour propager la vague dans la direction voulue. La vague déferlante a un angle de propagation de 15-degrés, avec une crête oblique, et  $H_{max} = 0.4$ ,  $\omega = 0.002$ ,  $d = 0.25$ ,  $\lambda_x = 1.0$ , et  $\lambda_z = 32.5$ .

#### Interaction entre vagues

Les interactions entre plusieurs vagues (section 2) nécessitent de calculer une solution pour les équations [KdV95],



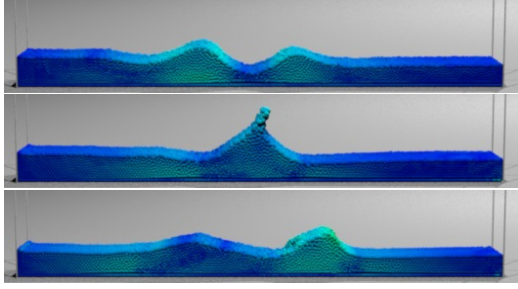


Figure 11: Deux vagues de directions opposées avec la même pulsation  $\omega = 0.002$ . La vague de gauche est plus haute ( $H = 0.27$ ) que celle de droite ( $H = 0.23$ ).

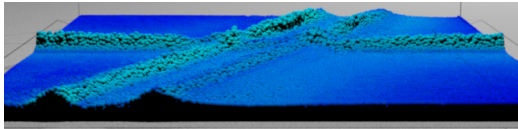


Figure 12: Croisement entre une vague orientée de manière oblique et deux autres. Les deux vagues parallèles ont une orientation de 45-degrés, avec  $H = 0.26$ ,  $d = 0.25$ , et  $\omega = 0.002$ .

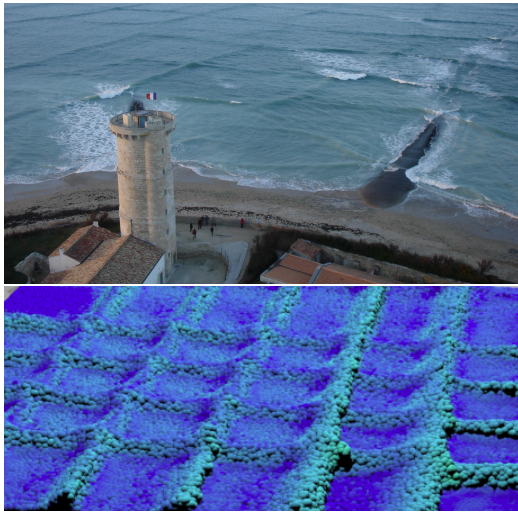


Figure 13: Dessus : photographie de croisements de vagues réelles, courtoisie de Michel Griffon. Dessous : simulation d'un comportement similaire avec notre modèle, avec  $H = 0.23$ ,  $\omega = 0.002$ ,  $d = 0.25$ ,  $\lambda_x = 1.0$  et  $\lambda_z = 32.5$ .

### Dépassement de vagues

La séquence illustrée dans la figure 14 correspond à une vague qui en dépasse une autre, i.e., avec une vitesse de propagation  $\omega$  plus grande. L'interaction entre ces deux vagues produit temporairement une seule vague, séparées ensuite à nouveau, et continuant de se propager avec leur vitesse respective, comme cela peut être observé dans la réalité.

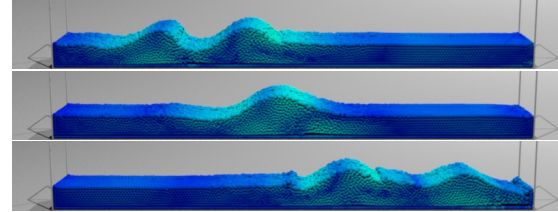


Figure 14: Une vague rapide et de faible amplitude ( $H = 0.23$ ,  $\omega = 0.003$ ) en dépasse une plus grosse et plus lente ( $\omega = 0.002$ ,  $H = 0.27$ ), avec  $d = 0.25$  pour les deux vagues.

### Vagues multiples

Nous avons combiné de nombreux types de vagues, avec des formes, hauteurs et crêtes différentes, et interagissant avec des objets (les interactions sont gérées par le solveur SPH). La figure 1 illustre certaines des configurations que nous avons définies ; la figure 15 montre deux images de la simulation sans obstacles ; la figure 16 utilise une autre configuration de plusieurs vagues, avec des obstacles supplémentaires.

La vidéo présentée à l'URL suivante présente les résultats obtenus à l'aide de notre modèle <https://vimeo.com/133911694>.

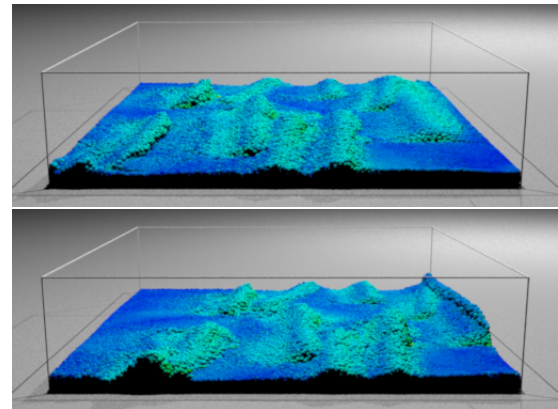


Figure 15: Deux images de plusieurs vagues avec leurs interactions. Leurs largeurs ont été limitées et plusieurs directions différentes leur sont associées.



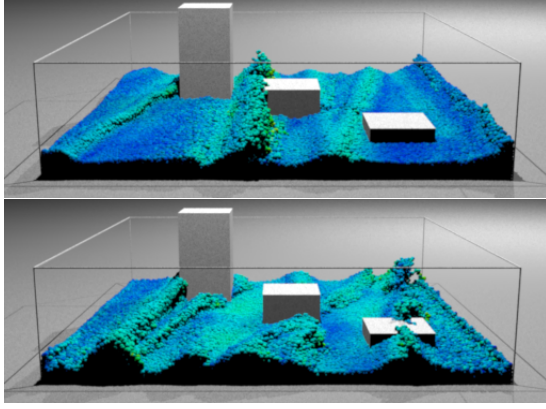


Figure 16: Deux images d'une séquence comportant plusieurs vagues et leurs interactions avec des blocs statiques.

#### 4.2. Performances et discussion

Notre modèle de vagues est implémenté comme une force externe dans notre solveur. La simulation est affichée en temps interactif avec OpenGL, et nous utilisons Mitsuba [Wen10] pour les rendus offline. Le coût du calcul de notre méthode est de l'ordre de 1% du temps de calcul total, comme cela est précisé dans les tableaux 1 et 2.

Les résultats présentés dans cet article ont été obtenus à l'aide d'un processeur Intel Xeon E2620 2.4GHz, équipé de 16GB de RAM. Nous avons parallélisé les calculs SPH sur une carte graphique Nvidia GTX Titan avec CUDA 6.5, incluant également les calculs de notre modèle. Les calculs de voisinage sont réalisés avec une grille uniforme [IABT11]. Nous simulons 10 itérations pour chaque image rendue ( $\Delta t = 0.001$  sec., tableau 2); le calcul de densité et de pression des particules est fait à l'aide des formules présentées par Ihmsen et al. [IOS\*14].

Notre modèle permet de produire de nombreux types de vagues à l'aide d'un nombre de paramètres relativement réduit. Cependant, il existe une corrélation entre ces paramètres; par exemple, modifier la vitesse de la vague  $\omega$  influence également sa hauteur. Outre ces contraintes, les bornes des valeurs doivent être également correctement choisies. Par exemple, lorsque  $H > 0.5$  la force verticale résultante est trop importante, et le résultat produit n'est plus réaliste. Le paramètre  $\omega$  permet de contrôler la vitesse de la vague; lorsque  $\omega > 0.1$ , la vitesse est trop importante.

#### 5. Conclusion et perspectives

Notre modèle de vagues reproduit une grande variété de phénomènes. Il est possible de générer de la houle et des vagues déferlantes, et de les faire interagir entre elles. Nous avons montré qu'avec un contrôle adéquat des paramètres, plusieurs scénarios peuvent être simulés, par exemple une vague de houle progressant en vague déferlante, ou de la

Phase	Houle Figure 5	Plongeante Figure 6	Vagues opposées Figure 11	Mer croisée Figure 12
Calcul voisins	47.89%	48.79%	48.97%	61.48%
Calcul forces internes	41.41%	40.54%	40.58%	31.26%
Calcul solitons	0.94%	0.99%	1.7%	0.86%
Intégration	0.82%	0.8%	0.83%	0.49%
Transferts mémoire	8.94%	8.88%	7.92%	5.91%

Table 1: Répartition des temps de calcul des kernels CUDA pour nos exemples.

Scène	Particules	Temps/image
Faible largeur (figure 6)	30k	0.013s
Largeur importante (figure 12)	100k	0.022s
Profondeur importante (figure 8)	255k	0.076s

Table 2: Temps de calcul moyen par image lors de la simulation.

mer croisée. La liste des effets pouvant être obtenus n'est pas exhaustive et les possibilités sont nombreuses. De plus, comme notre modèle est une force externe, il peut être utilisé dans tout autre solveur SPH (i.e., WCSPH [BT07], PCI-SPH [SR09], IISPH [ICS\*14]), Eulérien [EMF02] ou hybride [CIPT14]).

Prochainement, nous souhaitons corriger certaines limitations liées au contrôle des paramètres. Par exemple, la vitesse de la vague tend à modifier sa forme. Il pourrait être intéressant de proposer des fonctions de plus haut niveau pour contrôler des paramètres à un plus haut niveau (vitesse, hauteur, temps de déferlement).

Nous aimerions également ajouter des détails fins comme de l'écume et des embruns, et utiliser notre modèle pour identifier les régions où ces effets se produisent.

#### Références

- [BGH\*04] BALDWIN D., GÖKTAŞ Ü., HEREMAN W., HONG L., MARTINO R., MILLER J. : Symbolic computation of exact solutions expressible in hyperbolic and elliptic functions for nonlinear pdes. *Journal of Symbolic Computation* 37, 6 (2004), 669–705.
- [BNH10] BRUNETON E., NEYRET F., HOLZSCHUCH N. : Real-time realistic ocean lighting using seamless transitions from geometry to brdf. *Computer Graphics Forum* 29, 2 (2010), 487–496.
- [BT07] BECKER M., TESCHNER M. : Weakly compressible sph for free surface flows. In *SIGGRAPH/Eurographics SCA* (2007), pp. 209–217.
- [CIPT14] CORNELIS J., IHMSEN M., PEER A., TESCHNER M. : Iisph-flip for incompressible fluids. *Computer Graphics Forum* 33, 2 (2014), 255–262. :2014
- [DCG11] DARLES E., CRESPIN B., GHAZANFARPOUR D. : A particle-based method for large-scale breaking wave simulation. *Machine Graphics & Vision* 20, 1 (2011), 3–25.
- [DCGG11] DARLES E., CRESPIN B., GHAZANFARPOUR D.,

- GONZATO J.-C. : A survey of ocean simulation and rendering techniques in computer graphics. *Computer Graphics Forum* 30, 1 (2011), 43–60.
- [EMF02] ENRIGHT D., MARSCHNER S., FEDKIW R. : Animation and rendering of complex water surfaces. In *SIGGRAPH* (2002), pp. 736–744.
- [FF01] FOSTER N., FEDKIW R. : Practical animation of liquids. In *SIGGRAPH* (2001), pp. 23–30.
- [FM96] FOSTER N., METAXAS D. : Realistic animation of liquids. *Graph. Models Image Process.* 58, 5 (1996), 471–483.
- [FR86] FOURNIER A., REEVES W.-T. : A simple model of ocean waves. In *SIGGRAPH* (1986), pp. 75–84.
- [FSJ01] FEDKIW R., STAM J., JENSEN H. : Visual simulation of smoke. In *SIGGRAPH* (2001), pp. 15–22.
- [HW65] HARLOW F., WELCH J. : Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids* 8, 12 (1965), 2182–2189.
- [IABT11] IHMSEN M., AKINCI N., BECKER M., TESCHNER M. : A parallel sph implementation on multi-core cpus. *Computer Graphics Forum* 30, 1 (2011), 99–112.
- [ICS\*14] IHMSEN M., CORNELIS J., SOLENTHALER B., HORVATH C., TESCHNER M. : Implicit incompressible sph. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (2014), 426–435.
- [IOS\*14] IHMSEN M., ORTHMANN J., SOLENTHALER B., KOLB A., TESCHNER M. : Sph fluids in computer graphics. In *Eurographics 2014 - State of the Art Reports* (2014).
- [KdV95] KORTEWEG D. J., DE VRIES G. : On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves. *Philosophical Magazine Series* 5 39, 240 (1895), 422–443.
- [Kel48] KELLER J. : The solitary wave and periodic waves in shallow water. *Communications on Pure and Applied Mathematics* 1, 4 (1948), 323–339.
- [MCG03] MÜLLER M., CHARYPAR D., GROSS M. : Particle-based fluid simulation for interactive applications. In *SIGGRAPH/Eurographics SCA* (2003), pp. 154–159.
- [MMS04] MIHALEF V., METAXAS D., SUSSMAN M. : Animation and control of breaking waves. In *SIGGRAPH/Eurographics SCA* (2004), pp. 315–324.
- [Mon92] MONAGHAN J. : Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics* 30 (1992), 543.
- [Mon05] MONAGHAN J. : Smoothed particle hydrodynamics. *Reports on Progress in Physics* 68, 8 (2005), 1703–1759.
- [NSB13] NIELSEN M. B., SÖDERSTRÖM A., BRIDSON R. : Synthesizing waves from animated height fields. *ACM Trans. Graph.* 32, 1 (2013), 1–9.
- [Pea86] PEACHEY D. R. : Modeling waves and surf. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques* (1986), SIGGRAPH '86, ACM, pp. 65–74.
- [PTB\*03] PREMOZE S., TASDIZEN T., BIGLER J., LEFOHN A., WHITAKER R. : Particle-based simulation of fluids. *Computer Graphics Forum* 22, 3 (2003), 401–410.
- [RO98] RADOVITZKY R., ORTIZ O. : Lagrangian finite element analysis of newtonian fluid flows. *International Journal for Numerical Methods in Engineering* 43, 4 (1998), 607–619.
- [SR09] SOLENTHALER B., R. P. : Predictive-corrective incompressible sph. In *Proceedings of the 36th Annual Conference on Computer Graphics and Interactive Techniques* (2009), SIGGRAPH '09, ACM, pp. 40 :1–40 :6.
- [Sta99] STAM J. : Stable fluids. In *SIGGRAPH* (1999), pp. 121–128.
- [Tes99] TESSENDORF J. : Simulating ocean water. In *SIGGRAPH Course Notes* (1999).
- [TG02] THON S., GHAZANFARPOUR D. : Ocean waves synthesis and animation using real world information. In *Computers and Graphics* (2002), vol. 26, pp. 99–108.
- [TMFSG07] THUREY N., MÜLLER-FISCHER M., SCHIRM S., GROSS M. : Real-time breaking waves for shallow water simulations. In *Pacific Conference on Computer Graphics and Applications* (2007), pp. 39–46.
- [Wen10] WENZEL J. : Mitsuba renderer, 2010. <http://www.mitsuba-renderer.org>.
- [WJ10] WIEGEL R., JOHNSON J. : Elements of wave theory. *Coastal Engineering Proceedings* 1, 1 (2010), 2.