HAL
open science

Modélisation et animation de la mer en temps réel

Damien Hinsinger

To cite this version:

Damien Hinsinger. Modélisation et animation de la mer en temps réel. Synthèse d'image et réalité virtuelle [cs.GR]. 2001. inria-00598391

HAL Id: inria-00598391

https://inria.hal.science/inria-00598391

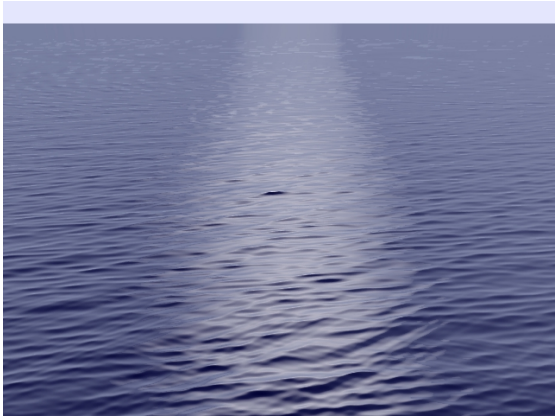
Submitted on 6 Jun 2011

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

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

DEA Imagerie, Vision et Robotique

Modélisation et animation de la mer en temps réel



Présenté par Damien HINSINGER

Sous la responsabilité de Marie-Paule CANI
et Fabrice NEYRET

Projet iMAGIS
Laboratoire GRAVIR (CNRS,INPG,INRIA,UJF)

Juin 2001

Table des matières

1 État de l'art **2**
1.1 Introduction au phénomène à reproduire . . . 2
1.1.1 La notion de particule fluide . . . 2
1.1.2 Détermination de l'écoulement . . . 3
1.1.3 Ondes de surface . . . 3
1.1.4 Eau peu profonde . . . 4
1.2 Les modèles paramétriques de houle . . . 5
1.2.1 La houle de Gestner . . . 5
1.2.2 La houle de Stokes . . . 6
1.2.3 Le modèle de Biesel . . . 7
1.3 Génération . . . 8
1.3.1 Le vent . . . 8
1.3.2 Influence des courants . . . 8
1.3.3 Propagation . . . 8
1.3.4 Génération procédurale . . . 9
1.4 La mer en synthèse d'images . . . 10
1.4.1 Modèles du phénomène inspirés de la physique . . . 10
1.4.2 Rendu et géométrie . . . 14

2 Mise au point d'une méthodologie **17**
2.1 Les niveaux de détail . . . 17
2.2 Modélisation . . . 17
2.3 Application à la mer . . . 18
2.4 Les primitives de mouvement . . . 19
2.4.1 Énumération des phénomènes . . . 19
2.4.2 Choix d'une modélisation . . . 19
2.4.3 Composition des primitive (fig. 2.2) . . . 20
2.4.4 Génération des primitives . . . 21
2.5 Les effecteurs : de la géométrie au rendu . . . 22
2.5.1 Étude du phénomène . . . 22

3 Niveaux de détail et aliasage **26**
3.1 Adaptation de maillage . . . 26
3.1.1 Conservation de la topologie . . . 26
3.1.2 Aliasage des primitives : introduction . . . 26
3.1.3 Adaptation du maillage . . . 27
3.2 Aliasage et niveaux de détail : filtrage . . . 29
3.2.1 Calcul des normales . . . 30
3.2.2 Filtrage . . . 33
3.3 Adaptation et temps réel . . . 34
3.3.1 Génération des primitives pour la mer . . . 34

4 Application et résultats **37**
4.1 Architecture du modèle 37
4.2 Résultats : la mer 38
4.2.1 Paramétrage de l'application 38
4.3 Un autre type de primitive : les sillages 42

Introduction

Depuis toujours, dans le domaine de la synthèse d'images, la mer fait l'objet de nombreux travaux. Mais il est impossible de représenter tous les aspects du phénomène de la même façon. Depuis quelques années, sa simulation, dans le domaine des effets spéciaux, atteint une qualité visuelle photoréaliste. Mais la mer, malgré la place qu'elle occupe sur nore planète, est pratiquement absente du principal domaine du temps réel, les jeux vidéos.
Le but de se travail est developper une méthode en temps réel d'animation et de rendu de la mer , scène relativement complexe par nature qui se prète bien à la recherche de techniques d'optimisations des méthodes d'animation en temps réel.
Il est difficile de se baser sur des travaux déjà réalisés pour atteindre ces buts, car il existe peu de jeux où la mer joue le premier rôle faute de moteurs à la hauter. La meilleure approche consiste donc à s'orienter vers des méthodes introduites à d'autres fins, que ce soit en synthèse d'image (effets spéciaux) ou en physique (hydrodynamique, océanographie).
Pour gérer au mieux la complexité des modèles, nous allons utiliser une méthodologie particulière basée sur l'indépendance entre la visualisation d'un phénomène et son animation. Ici, cela signifie séparer le rendu de la modélisation et l'animation de la mer elle-même.
En nous basant sur un modèle paramétrique du phénomène, nous allons générer à la volée un maillage correspondant au mieux avec le contexte de la vue et de la scène.
Enfin, nous allons traiter l'adaptation des détails visuels ainsi que les importants problèmes d'aliasage dus à l'étendue des échelles de représentation par un filtrage en temps réel des informations à calculer et à afficher.
Le traitement des niveaux de détail et l'adaptation de maillage constituent, aujourd'hui, un axe de recherche très actif dans le cadre du temps réel dans la mesure où la communauté d'intersse à la représentation de scènes très complexes. Les théories développés dans ce projet ne sont donc pas restreintes à la représentation de la mer, même si celle-ci s'est avérée un excellent sujet d'étude.

Les résultats, bien qu'étant fortement sensibles aux ressources de calculs, sont d'une excellente qualité visuelle et ouvrent même la possibilité d'une adaptation pour les domaines non temps réel.

Chapitre 1

État de l'art

Nous allons présenter ici les travaux effectués en synthèse d'images dans les différents domaines de modélisation, animation et rendu de la mer, et plus généralement de liquides. Afin d'en faciliter la compréhension, nous allons commencer par introduire les principes physiques qui régissent les phénomènes que nous allons étudier. Nous verrons ensuite les différentes modélisations qui existent dans le domaine de la physique et enfin, les précédents travaux effectués en synthèse d'images.

1.1 Introduction au phénomène à reproduire

Dans le domaine de l'étude physique des fluides[1] , il est une modélisation particulièrement utile à l'informatique puisqu'elle se base sur la notion de *particule fluide*.

1.1.1 La notion de particule fluide

Les molécules d'un fluide sont, en raison de son état, en constant mouvement les unes par rapport aux autres. Elles sont plus particulièrement caractérisées par une plus ou moins importante fluctuation autour d'une vitesse moyenne. Cette fluctuation est directement liée à une excitation énergétique puisque leur amplitude définit la température du fluide.
Les valeurs moyennes de la cinétique des molécules sont évidemment conséquences de phénomènes de plus grande échelle, il s'agit de l'écoulement du fluide.
Pour l'étude de la cinétique (et indirectement de la dynamique), il faut donc faire abstraction des phénomènes intermoléculaires (du domaine de la thermodynamique) et trouver une structure qui soit caractéristique de l'écoulement du fluide (à son échelle).
Les physiciens introduisent, pour cela, la notion de *particules fluide*. Cette dernière regroupe un ensemble de molécules afin d'en tirer des caractéristiques moyennes. Si l'on considère les données thermodynamiques constantes dans le problème (pas de fortes variations de température ou de pression, c'est à dire système énergétiquement fermé), on peut introduire un regroupement des molécules selon un volume choisi négligeable devant la taille de l'écoulement et le plus grand possible devant la taille moléculaire.
Cette modélisation a plusieurs conséquences intéressantes. La première est que les caractéristiques de la particule se trouvent être locales et continues à l'échelle de l'écoulement ; cela justifie en particulier l'utilisation des dérivées spatiales. La deuxième est plus intéressante pour la modélisation numérique, il s'agit du fait que l'on peut utiliser des particules fluides de forme libre à condition de considérer toutes les interactions correctement, et en prenant en compte le fait que la particules est déformable (car les trajectoires des molécules à l'intérieur ne sont par forcement identiques). Cela permet d'adapter la forme à la résolution ainsi qu'aux conditions de l'écoulement. Kass et Miller [8] ont utilisé des particules de type colonne d'eau pour modéliser un phénomène d'ondes gravitationnelles où les interaction horizontales sont négligeables.
La particule fluide fournit donc à l'informatique une base solide pour des modèles numériques relativement proches de la théorie physique.

[1] informations extraites de [1],[2],[9],[10],[11] et [12].

Cependant, en pratique, les forces qui régissent l'écoulement ont une complexité suffisante pour interdire une modélisation totalement fidèle pour beaucoup de phénomènes, comme nous allons le voir.

1.1.2 Détermination de l'écoulement

Si l'on applique les lois de la dynamique à la particule fluide définie précédemment, on obtient l'équation connue comme «équation de Navier-Stokes» (du nom de ses inventeurs simultanés).

$$\rho\tau\vec{a} = \vec{f}\tau + \iint_{\Sigma} (-P + \sigma)\mathrm{d}\vec{S} \tag{1.1}$$

pour une particule de surface fermée S, de volume τ et de masse volumique ρ. $\vec{f}$ représente les forces volumiques extérieures et $-P_\sigma$ la somme des contraintes de pression et friction visqueuse exercés par le fluide sur la particule.

Cette équation traduit l'évolution spatio-temporelle de la particule dans un fluide newtonien incompressible. Avec quelques informations supplémentaires (champ de pression, continuité du fluide), on peut ainsi calculer le champ de vitesse dans le milieu et donc déterminer l'écoulement.
La plupart des simulations physiques d'écoulement sont aujourd'hui réalisées avec cette technique ; néanmoins, la complexité de la résolution réserve son utilisation aux cas où l'on veut obtenir une solution précise, et réciproquement où on a le temps de la calculer.
Dans le domaine de la synthèse d'images, l'usage de la méthode numérique multiplie la complexité (et les ressources nécessaire) bien au delà des limites habituellement acceptables en raison du niveau de détail demandé (autant visuel que phénoménologique pour la plupart des cas). En réalité, on retrouve ici les problèmes classiques de la modélisation physique en synthèse d'image, à savoir que nous recherchons un modèle visuellement réaliste et aisément contrôlable alors que la physique fournit uniquement l'assurance d'une solution consistante.
De plus, les travaux effectués sur les fluides dans le domaine de la physique sont rarement transposables a la synthèse d'image. En effet, les calculs de marée ou de déformation de structures ne sont pas effectués à une échelle visuellement pertinente, l'objectif étant souvent d'obtenir un échantillon de valeurs caractéristiques.

Néanmoins, certaines hypothèses permettent de trouver des solutions physiques paramétriques des plus intéressantes pour l'image.

1.1.3 Ondes de surface

L'étude de l'animation de la surface de la mer est celle des ondes de surface qui la déforment. Physiquement, cela revient à étudier le mouvement oscillatoire de l'interface entre deux fluides (l'air et l'eau). Il faut d'abord souligner que le milieu est dispersif : la vitesse varie avec la longueur d'onde.
Pour les ondes d'interface entre l'air et l'eau, la relation de dispersion, liant la pulsation $\omega = 2\pi/T$ et le nombre d'onde $k = 2\pi/\lambda$ (avec T période et λ longueur d'onde) peut s'écrire comme suit :

$$\omega^2 = (\mathrm{g}k + \gamma k^3/\rho)\tanh(kH) \tag{1.2}$$

où γ est le rapport des chaleurs massiques des milieux et H la profondeur.

On voit que l'équation de dispersion met en valeur deux régimes distincts (fig. 1.1).

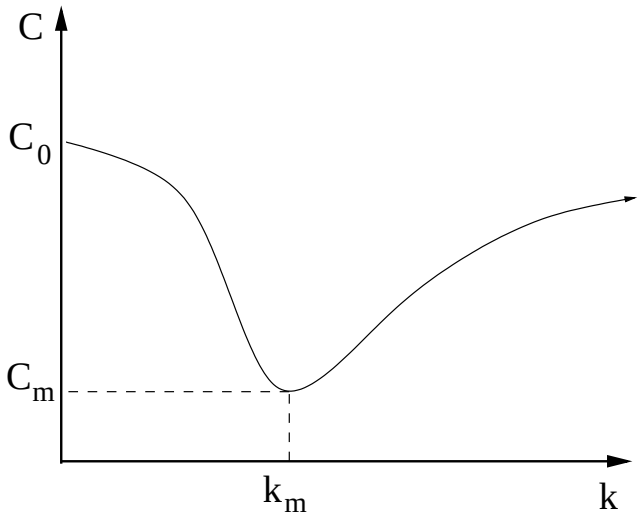


FIG. 1.1 – Vitesse en fonction du vecteur d'onde ($C = \omega/k$)

– Le plus important est celui des ondes gravitationnelles (grandes longueur d'onde). La force de rappel de l'oscillation est alors la pesanteur et la vitesse (on montre que $c = \omega/k$) décroît avec la racine carré de la longueur d'onde, c'est le terme en $gk\tanh(kH)$, on a :

$$c = \sqrt{g/k\tanh(kH)}$$

– Pour les petites longueurs d'onde (en deçà de 1,7cm) on parle d'«onde capillaires», dues à la tension superficielle à l'interface entre les deux milieux, c'est le terme en $\gamma k^3/\rho$. On montre que la vitesse décroît alors avec la longueur d'onde.

$$c = \sqrt{\gamma k/\rho\tanh(kH)}$$

En raison de l'échelle des phénomènes, nous allons surtout porter notre intérêt sur les ondes gravitationnelles.

1.1.4 Eau peu profonde

La notion d'eau *profonde* ou *peu profonde* s'attache la plupart du temps au rapport entre la profondeur et la longueur de l'onde. On remarque que l'équation de dispersion fait encore une fois apparaître deux régimes, car le terme $tanh(kH)$ tend vers 1 en eau profonde.
En eau peu profonde (approche des plages), les ondes conservent leur pulsation ω et le vecteur d'onde $\vec{k}$ obéit alors à la loi de dispersion. Cela donne naissance à deux phénomènes.

– D'abord, la longueur d'onde diminue, les vagues se rapprochent les unes des autres. C'est un phénomène de réfraction, si l'on considère le milieu caractérisé par sa profondeur variable.
  C'est ce phénomène de réfraction qui entraîne une déformation du front d'onde à l'approche des côtes.
– Ensuite, si l'amplitude est suffisante, on constate un déphasage spatial progressif de l'onde entre sa crête et son creux (fig. 1.2) du à une variation du vecteur d'onde entre ces deux positions. C'est ce qui fait que les vagues se penchent en avant à l'approche de la plage.

Dans les deux cas, il existe une limite au rapport entre l'amplitude et le vecteur d'onde qui, une fois dépassée, produit le déferlement de la vague. Celui-ci se traduit entres autres par une perte d'énergie de l'onde.

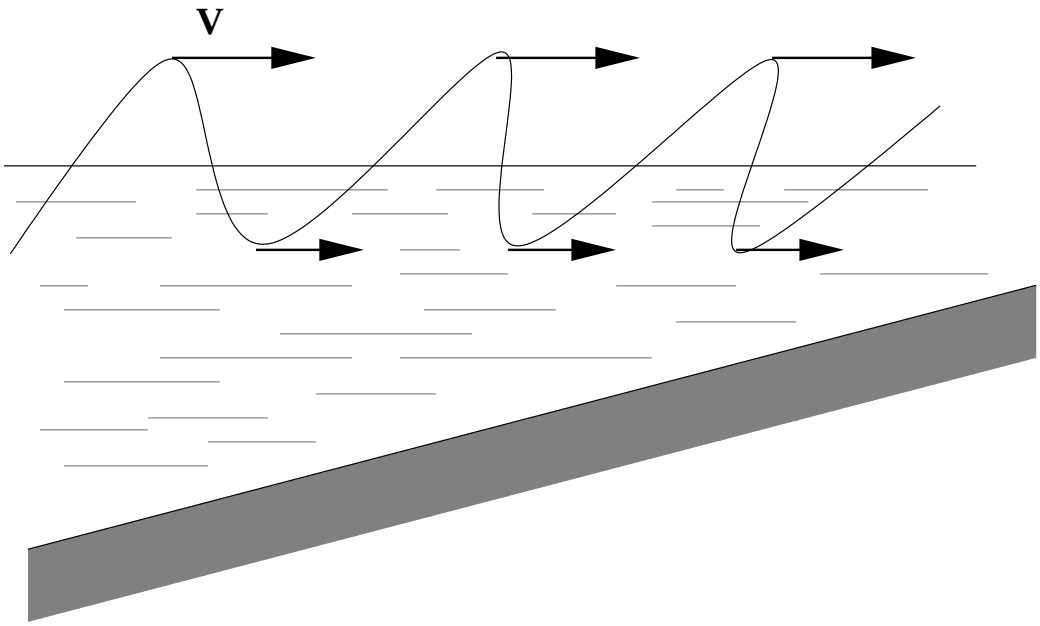


FIG. 1.2 – Déferlement des vagues

1.2 Les modèles paramétriques de houle

Les modèles paramétriques sont des modèles fournissant des équations décrivant paramétriquement la surface à un instant donné. Comme la dynamique générale fournit des équations différentielles, les modèles paramétriques se basent sur des solutions particulières. Certaines approximations paramétriques sont attractives car les trajectoires suivies par les particules sont fermées, ce qui a pour nous un certain intérêt comme nous allons le voir.

1.2.1 La houle de Gestner

Le modèle de houle de Gestner a été introduit en 1802. Il suppose que la longueur d'onde est négligeable devant la profondeur et que l'écoulement est irrotationnel, ce qui implique qu'il dérive d'un potentiel.

Trajectoire des particules fluides (1.3)

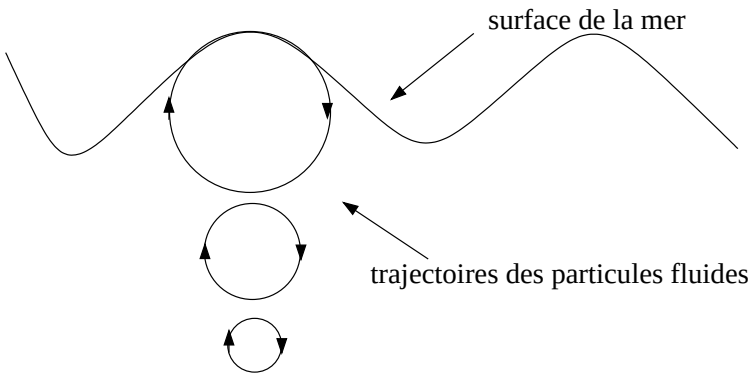


FIG. 1.3 – Trajectoire des particules de fluide en eau profonde en fonction de leur profondeur

Dans le cas des ondes gravitationnelles, l'écoulement généré par les vagues est assez particulier. A l'image du bouchon ou du baigneur qui flotte, les particules sont globalement stationnaires (hors courant) et sont régies par des mouvement oscillatoires locaux.
Il s'agit en fait d'une oscillation horizontale (dans le sens de propagation, c'est à dire perpendiculairement au front de l'onde) et d'une oscillation verticale.

En pratique, la détermination du champ de vitesse fournit une composante verticale sinusoïdale et une composante horizontale également sinusoïdale déphasée de 90 degrés, ce qui donne une trajectoire circulaire pour chaque particule. L'intégration autour d'un *centre* (x_0, z_0) donne les équations paramétrique suivantes.

$$\begin{cases} x - x_0 = -\xi_0 \, \mathrm{e}^{kz_0} \sin(kx_0 - \omega t) \\ z - z_0 = -\xi_0 \, \mathrm{e}^{kz_0} \cos(kx_0 - \omega t) \end{cases} \tag{1.3}$$

On note que le rayon décroît exponentiellement avec la profondeur (fig. 1.3). La courbe résultant des mouvements simultanés de particules dont les centres consituent la courbe continue $z = z_0$ est appelée *trochoïde* (1.4). On peut étendre cette consrtuction par les mouvement autour de centres consituant une surface de topologie planaire. La trochoïde ne ressemble à une sinusoïde que pour les faibles amplitudes.

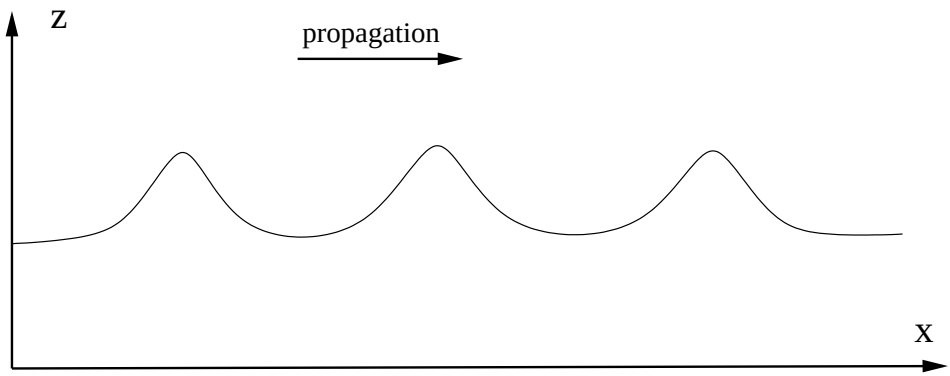


FIG. 1.4 – Trochoïde

L'équation de dispersion introduite plus haut (1.2) fournit une relation entre le vecteur d'onde et la fréquence de la vague. La valeur de l'amplitude, quand à elle, ne dépend que du phénomène qui a engendré la houle. Elle prend donc des valeurs indépendantes des paramétres de fréquence et longueur d'onde. Si on dérive spatialement l'équation de la houle, on voit aisément qu'il existe une amplitude au delà de laquelle la trochoïde *boucle*. Cette limite, correspondant à un rapport entre la longueur d'onde et l'amplitude, représente le déferlement de la vague. La valeur limite peut aisément être calculée mais l'expérience montre qu'en raison notamment de la tension superficielle, l'amplitude peut aller légèrement au delà de la prévision théorique.

1.2.2 La houle de Stokes

Partant des mêmes hypothèses, Stokes introduit un modèle prenant en compte la faible profondeur. Ce modèles est basé sur l'intégration par développement limité des équations différentielles déterminées avec l'hypothèse selon laquelle les vitesses dérivent de potentiels (irrotationalité).
Pour utiliser le modèle de Stokes au premier ordre, certaines hypothèses doivent être prises en compte. La cambrure doit être faible, c'est à dire que l'amplitude doit être faible devant la longueur d'onde ainsi que la profondeur. On obtient alors au premier ordre :

$$\begin{cases} \xi_1 = \frac{H}{2} \cos(kx - \omega t) \\ \Phi_1 = \frac{gHT}{4\pi} \frac{\cosh(k(z+h))}{\sinh(kh)} \sin(kx - \omega t) \\ \lambda = \frac{gT^2}{2\pi} \tanh\left(\frac{2\pi h}{\lambda}\right) \end{cases} \tag{1.4}$$

ξ est l'expression de la surface libre et Φ celle du potentiel des vitesses. H est l'amplitude (arbitraire) de la vague. h est la profondeur.

Avec cette solution, les trajectoires des particules deviennent des ellipses, qui s'aplatissent en approchant du fond jusqu'à avoir un rayon vertical nul.
On remarque ici que lorsque le rapport entre la longueur d'onde et la profondeur tend vers 0, on retrouve le modèle de Gestner.

Les développements limités à des ordres supérieurs donnent des solutions non linéaires. Les trajectoires ne sont plus fermées. Elles sont surtout utilisées pour les calculs d'efforts sur des structures telles les plates-formes pétrolières.

1.2.3 Le modèle de Biesel

Malgré la prise en compte des faibles profondeurs, le modèle de Stokes ignore l'aspect énergétique de la réfraction sur le fond ce qui a pour conséquence de l'invalider lorsqu'il y a trop d'augmentation du vecteur d'onde, c'est à dire lorsque les vagues s'approchent du déferlement. En pratique, Stokes n'est plus valable pour des profondeurs inférieures à la moitié de la longueur d'onde.
L'approche de Biesel consiste à tenir compte des modifications d'amplitude impliquées par la variation de la longueur d'onde selon la loi suivante.

$$\left(\frac{H}{H_\infty}\right)^2 = \frac{1}{\tanh(Kh)\left(1+\frac{2Kh}{\sinh(2Kh)}\right)} \tag{1.5}$$

où $\frac{H}{H_\infty}$ représente la variation de l'amplitude par rapport à sa valeur en eau profonde.

Ce modèle permet ainsi de calculer plus précisément le déferlement et de s'approcher encore du rivage. Les équations paramétriques du modèle sont :

$$\begin{cases} x - x_0 = A_x \sin\left(\int_0^{x_0} K\mathrm{d}x - \omega t\right) - B_x \cos\left(\int_0^{x_0} K\mathrm{d}x - \omega t\right) \\ z - z_0 = A_z \sin\left(\int_0^{x_0} K\mathrm{d}x - \omega t\right) - B_z \cos\left(\int_0^{x_0} K\mathrm{d}x - \omega t\right) \end{cases} \tag{1.6}$$

avec A et B fournis par des développements limités de l'équation 1.5.
En fait, les orbites des particules deviennent des ellipses qui s'alignent avec la direction du fond (fig. 1.5).

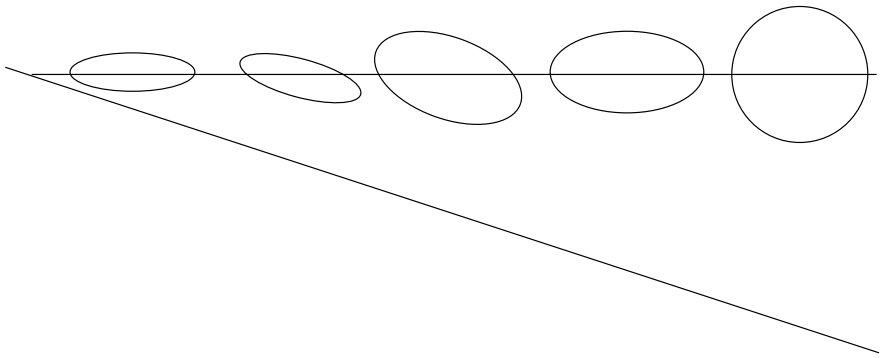


FIG. 1.5 – Trajectoire des particules avec le modèle de Biesel

Cette méthode permet de s'approcher du point de déferlement mais elle reste paramétrique donc ne va en aucun cas au delà. La physique d'un déferlement fait intervenir d'autres phénomènes non pris en compte dans ce modèle.

1.3 Génération

Une composante relativement importante de la modélisation physique de la mer est la génération de l'ensemble de vagues que l'on va utiliser. Pour traiter cela de manière efficace, il est nécessaire de s'intéresser aux phénomènes qui produisent les environnements que l'on va vouloir modéliser.

1.3.1 Le vent

Bien que le lien entre le vent et les vagues soit difficile à établir précisément, les océanographes en ont une bonne connaissance empirique. L'état de la mer est la résultante d'une superposition d'ondes de longueur et périodes différentes engendrées par le vent. En dehors de la zone où souffle le vent, ces ondes se nomment la houle.
Les vagues générées vont avoir des caractéristiques dépendant de plusieurs variables. Il faut prendre en compte la force du vent et sa durée d'action, ainsi que la dimension de la zone génératrice (fetch), zone dans laquelle souffle un vent d'une vitesse et direction données.

Périodes dominantes :

- Pour un fetch court (de petite taille) ou un temps d'exposition de 1 à 2 heures, la période (en seconde) est le cinquième de la force (vitesse en noeuds[2]) du vent.
- En condition moyenne, la période est le quart de la force du vent.
  Pour un vent de 40 noeuds, la période est d'environ 10 secondes (entre 8 et 13 avec période dominante à 10).
- Pour une longue durée (12-24h), sans limitation de distance (plusieurs miles), on a un état stationnaire : la mer est complètement levée et, pour obtenir la période, on divise la force par 3.7

Il est difficile d'estimer la hauteur des vagues. En pratique, la mesure est réalisée de manière statistique. Précisons seulement que sur dix milles vagues, certaines peuvent atteindre le double de la hauteur moyenne.

1.3.2 Influence des courants

Lors de la rencontre d'une vague avec un courant, la longueur d'onde et l'amplitude varient. Le point limite de déferlement est également plus élevé si la vitesse du courant est proche du quart de celle de la vague. C'est un phénomène fréquemment observé près des côtes.
Le type de courant observable à l'échelle qui nous intéresse est celui généré par le vent ; il s'observe jusqu'à une certaine profondeur et il atteint théoriquement 3% de la force du vent. De plus, on observe un décalage de son orientation à droite ou à gauche selon l'hémisphère de l'ordre de $45\ ^o$ et qui peut atteindre 180° selon la profondeur.

1.3.3 Propagation

A l'intérieur du fetch sont générés un grand nombre de longueurs d'ondes différentes, parmi lesquelles certaines ont une durée de vie très limitée. Les pertes énergétiques engendrées par les déferlements et l'amortissement, par la viscosité, des petites longueurs d'onde, entraînent que seule une certaine gamme de fréquences est présente à l'extérieur du fetch. Ce sont les longueurs d'onde les plus énergétiques et ce sont elles qui vont se propager loin du fetch donc généralement elles que l'on va vouloir représenter.

Pour les cas établies , c'est à dire hors du fetch ou pour une mer non susceptible d'évloluer, Pierson et Moskowitz ont proposé en 1964, à partir de mesures expérimentales, un filtre décrivant le spectre des ondes de surface générés par le vent (1.3.3). Ce filtre est défini par :

$$F_{PM}(f) = \frac{a.g^2}{(2\pi)^4 f^5} \mathrm{e}^{\frac{5}{4}\left(\frac{f_m}{f}\right)^4} \tag{1.7}$$

[2] 1 mille (1 852 m) par heure.

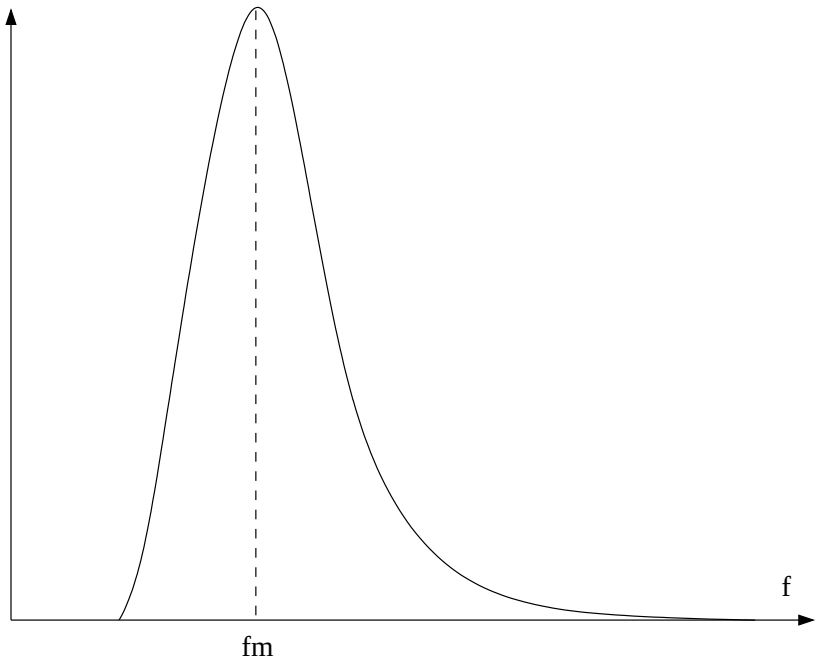


FIG. 1.6 – Filtre de Pierson-Moskowitz

où a est la constante de Philips, g l'accélération de la pesanteur et $f_m = \frac{0.13g}{U_{10}}$ représente un pic de fréquence dépendant de U_{10}, vitesse du vent à une hauteur de 10 mètres.

Ce filtre a été étendu en deux dimensions par Hasselmann, Dunckel et Ewing de la façon suivante :

$$F(f, \alpha) = F_{PM}(f).D(f, \alpha) \tag{1.8}$$

où $D(f, \alpha)$ pondère le filtre de Pierson-Moskowitz selon la direction α.

Mastin, Watterberg et Mareda [13] ont utilisé directement le filtre de Pierson-Moskowitz pour filtrer une image de bruit blanc et en tirer un champ de hauteur. Thon, Dischler et Ghazanfarpour [20, 21, 22] ont utilisé la version 2D du filtre de Pierson-Moskowitz pour générer une image dont ils tirent les caractéristiques à injecter dans leurs trochoïdes.

1.3.4 Génération procédurale

Beaucoup de travaux utilisent à différentes échelle une génération procédurale de surface. Beaucoup sont empiriques comme celle de Perlin [16], certains tiennent compte de données spectrales [20], toujours sur le modèle de Perlin, et nombreux sont ceux qui utilisent des fonctions de turbulence fortement aléatoires, Fournier et Reeves [3] proposent de bruiter les paramètres de leur modèle, Gonzato [6],[5],[7] utilise une fonction de bruit pure pour simuler les vagues capillaires.
La discutable efficacité des bruitages pour la simulation des phénomènes ondulatoires de la mer tient en grande partie au fait que l'on peut retrouver la forme du filtre de Pierson-Moskowitz de manière empirique (car la plupart des bruits utilisés sont de nature gaussienne).
Les meilleurs résultats sont obtenus par la fonction de perturbation de Perlin qui, par nature, revient à appliquer une fonction de bruit, contrôlée par filtrage originalement empirique, et assujettit au filtre de Pierson-Moskowitz par Thon [20].

1.4 La mer en synthèse d'images

Maintenant que nous avons exposé les bases nécessaires à la compréhension du phénomène, nous allons voir comment le reproduire en synthèse d'images. Ce sujet a fait l'objet de plusieurs travaux vers la fin des années 80. Malheureusement, la complexité du phénomène l'a par la suite réservé à des applications ne reculant pas devant les moyens, tant matériels qu'humains (graphistes), comme les effets spéciaux.
Malgré cela, les rendus de la mer sont d'excellente qualité (Titanic, Waterworld [17], The perfect storm).

1.4.1 Modèles du phénomène inspirés de la physique

La première étape pour créer une visualisation de la mer est de construire une surface, bien que certains phénomènes comme le déferlement puissent rompre la surface de l'eau, l'utilisation de modèles paramétriques entraîne qu'il est le plus souvent pertinent de dire que la surface n'a pas de trous (continuité C^0). Pour construire et animer la surface, il y a deux approches différentes. La première est de créer un modèle numérique d'intégration des équation de la dynamique du fluide. Elle est rarement utilisée car lourde en temps de calcul. La deuxième consiste à utiliser des solutions particulières de la physique pour en tirer des modèles paramétriques.

Modèles physiques dynamiques

Dans cette optique, Kass et Miller [8] ont introduit un modèle dynamique, basé sur certaines approximations, intégrant un système simple et stable d'équations différentielles en eau peu profonde. Néanmoins, ce modèle revouvre des cas très limités et cela ne justifie pas sa complexité pour une utilisation dans un cadre plus général.

Modèles paramétriques

Pour les raisons exposées ci-dessus, les modèles paramétriques sont de loin les plus usités.

Modèle sinusoïdal

L'approche la plus simpliste est d'utiliser des sinusoïdes pures pour modéliser les vagues. Lorsque l'amplitude est faible, la trochoïde ressemble à une sinusoïde. Les travaux effectués par Peachey [14] et Ts'o et Barsky [23] s'appuient sur cette idée. Cette approche permet une modélisation des plus simple de surfaces peu agitées comme c'est souvent le cas pour les faibles étendues (piscines, lacs). Cela permet également de résoudre simplement le problème de la sommation de vagues paramétrées différemment et de fournir un champ de hauteurs qui se prête bien a la modélisation et au rendu (triangulations etc.). Par contre, l'utilisation d'une fonction au sens algébrique (donc générant un champ de hauteurs) interdit certaines formes aux vagues, car on ne peut pas avoir plusieurs points sur une même verticale. Ce modèles exclut donc les vagues «penchées» proches du déferlement.

Le modèle de Fournier-Reeves [3]

Le modèle proposé par Fournier et Reeves est basé sur les modèles de houle de Gestner et Biesel présentés au paragraphe 1.2. L'approche consiste à modifier les équation de Gestner par la méthode de sommation sur le vecteur d'onde utilisée par Biesel.
En pratique, cela donne :

$$\begin{cases} x = x_0 + R\sin(\Phi) \\ z = z_0 + R\cos(\Phi) \\ \Phi = -\omega t + \int_o^{x_0} k(h(x))\mathrm{d}x \end{cases}$$

où R représente l'amplitude fixée empiriquement.

Ainsi, cette approche permet aux auteurs de créer une continuité entre les modèles de Gestner et Biesel avec les variations de profondeur.

Pour déterminer la valeur de k en fonction de la profondeur, Fournier et Reeves proposent une valeur approchée (à 5%)

$$K(h_x) = \frac{K_\infty}{\sqrt{\tanh(K_\infty h_x)}} \tag{1.9}$$

Cette méthode implique de sommer pour tout point de l'espace les valeurs de k depuis le large. Pour implémenter cela de façon pratique, les auteurs discrétisent la somme par la technique classique de Riemann, et développent pour obtenir une somme sur x de la forme :

$$\Phi = -\omega t + \sum_{0}^{x_0} \frac{K_\infty}{\sqrt{\tanh(K_\infty h(x))}} \Delta x$$

La somme ne dépend plus alors que de K_∞ et de la géométrie du fond, ce qui est invariant dans le temps pour chaque point.

Pour les modifications d'amplitude, les auteurs se basent sur la théorie de Biesel en utilisant des ellipses qui s'alignent sur le fond, mais ils en simplifient les paramètres pour gagner en facilité d'implémentation et de contrôle.

La réfraction des trains d'onde sur le fond est implémentée par la classique loi de réfraction de Descartes qui permet d'orienter la direction du front localement. Avec tous les paramètres, les équations deviennent :

$$\begin{cases} x = x_0 + R\cos\alpha.S_x \sin\Phi + sin\alpha.S_z \cos\Phi \\ z = z_0 - R\cos\alpha.S_z \cos\Phi + sin\alpha.S_x \sin\Phi \end{cases} \tag{1.10}$$

$$S_x = -\frac{1}{1 - \mathrm{e}^{-K_x h}} \qquad \text{et} \qquad S_z = S_x(1 - \mathrm{e}^{-K_x h})$$

avec Φ défini comme précédemment et $\sin\alpha = \sin(\gamma)\mathrm{e}^{-K_x h}$ où γ est donné par la réfraction sur le fond.

Ce modèle gère donc la surface par trains d'ondes et calcule leur réfraction en fonction du fond jusqu'au bord de la plage. La technique de modélisation par trains de vagues permet de générer les interférences dues à la réfraction (fig. 1.7).

Un autre avantage de la modélisation par train est de permettre aisément la réflexion contre des obstacles puisque, dans ce cas, chaque train n'interfère pas avec lui-même . Le fait de connaître les caractéristiques cinématique de chaque train d'onde permet de fournir des données dynamiques pour traiter le déferlement, par exemple avec des systèmes de particules comme présenté dans l'article de Fournier et Reeves.
Enfin, pour rendre la surface plus réaliste, les auteurs proposent d'utiliser un train bruité de faible longueur d'onde qui vient se rajouter aux vagues de grandes longueurs d'onde.

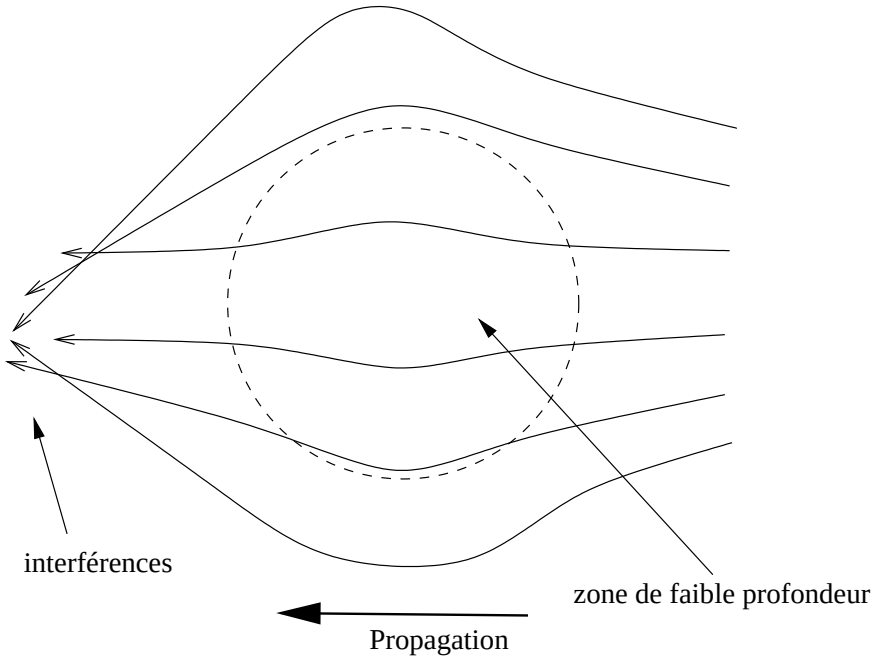


FIG. 1.7 – Interférences dues à la réfraction du fond (vue de dessus)

FIG. 1.8 – Modèle de Founnier et Reeves [3]

En conclusion, cet article présente une approche très intéressante d'implémentation des modèles physiques paramétriques en choisissant de composer la surface par une division du phénomène en fonction des paramètres. Notons tout de même l'inconvénient de ce type de technique qui est principalement l'augmentation de la complexité de modélisation et de calcul lorsqu'on souhaite utiliser un grand nombre de trains

de de vagues. Mais cela n'apparaît pas ici, car la philosophie du modèle même le cantonne à l'utilisation d'un faible nombre de train, en permettant de modifier aléatoirement ou stochastiquement son apparence (notamment le long de la crête).

Approche statistique

Une approche fondamentalement différente de celle exposée précédemment est celle utilisée par Mastin, Watterberg et Mareda [13]. Cette approche consiste à synthétiser un champ de hauteur à partir d'un filtrage de bruit blanc en tenant compte des particularités fréquentielles de la mer mesurés expérimentalement (fig. 1.9) ou parfois générées de façon procédurales (théorique). La méthode est contrôlable par plusieur paramètres, agissant sur le contrôle du filtre (cf [4]).

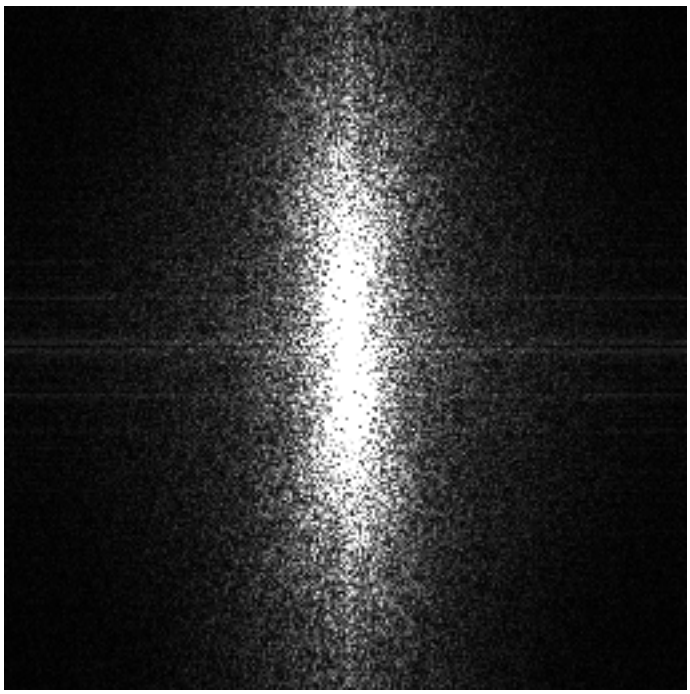


FIG. 1.9 – Spectre d'une image de mer, (on ne représente ici que les amplitudes). On peut en tirer un filtre permettant de synthétiser un champ de hauteurs

Le résultat est d'une certaine qualité car la génération est directement dépendante de la discrétisation (résolution de l'image de bruit) ce qui permet d'obtenir la meilleure complexité possible dans le spectre des vagues générées tout en évitant l'aliasage du à la discrétisation. Cette méthode a été utilisée dans un cadre sensiblement différent par Jos Stam [18], qui produit un schéma de répétition d'images de bruit pour générer un champ de hauteur apériodique.

Il apparaît néanmoins un problème non négligeable lorsque l'on veut animer les résultats obtenus puisqu'on ne contrôle pas les phases des ondes. Notons qu'il est très difficile de retrouver les paramètres de trochoïdes présentes dans les images obtenues par filtrage pour les utiliser dans l'animation. En fait, cela revient à déconvoluer par des trochoïdes et il faut, pour cela, connaître idéalement la fréquence, l'amplitude et l'orientation de celles-ci.

Afin de palier à ce défaut, une approche hybride a été introduite par Thon, Dischler et Ghazanfarpour [20] qui consiste à utiliser la technique précédente pour contrôler les paramètres de trochoïdes.
Pour cela, les auteurs discrétisent une image générée comme précédemment et, après sélection des compo-

santes significatives, utilisent ces données (longueur d'onde et orientation) comme paramètres d'un modèle basé sur la houle de Gestner. Ils justifient l'application des paramètres issus de sinusoïdes (base de l'espace fréquentiel de la transformé de Fourier) à des trochoïdes par la similitude entre les deux spectres, ce qui entraîne implicitement que l'on se cantonne au cas des vagues de faible amplitude.
Pour rendre les détails de petite échelle, la méthode consiste à utiliser une fonction de turbulence de Perlin [16] paramétrée pour approcher le spectre du filtre de Pierson-Moskowitz (cf. 1.3).

1.4.2 Rendu et géométrie

Le comportement optique des surfaces liquides calmes, est suffisamment bien connu pour permettre une étude précise dans le domaine de l'informatique graphique. Fondamentalement, les surfaces liquides sont des réflecteurs presque parfaitement spéculaires, dont les fonctions de réflexion et transmission sont parfaitement connues.

Réflexion et transmission

La réflexion et la réfraction sont parfaitement modélisés par les lois de l'optique. La direction de réflexion obéit à la loi classique de la symétrie du rayon incident par rapport à la normale et la réfraction suit la loi de Descartes. Enfin, Fresnel fournit un coefficient pour calculer la part de réflexion et de réfraction en fonction de l'angle normal d'incidence (angle entre le rayon incident et la normale) (fig. 1.10).

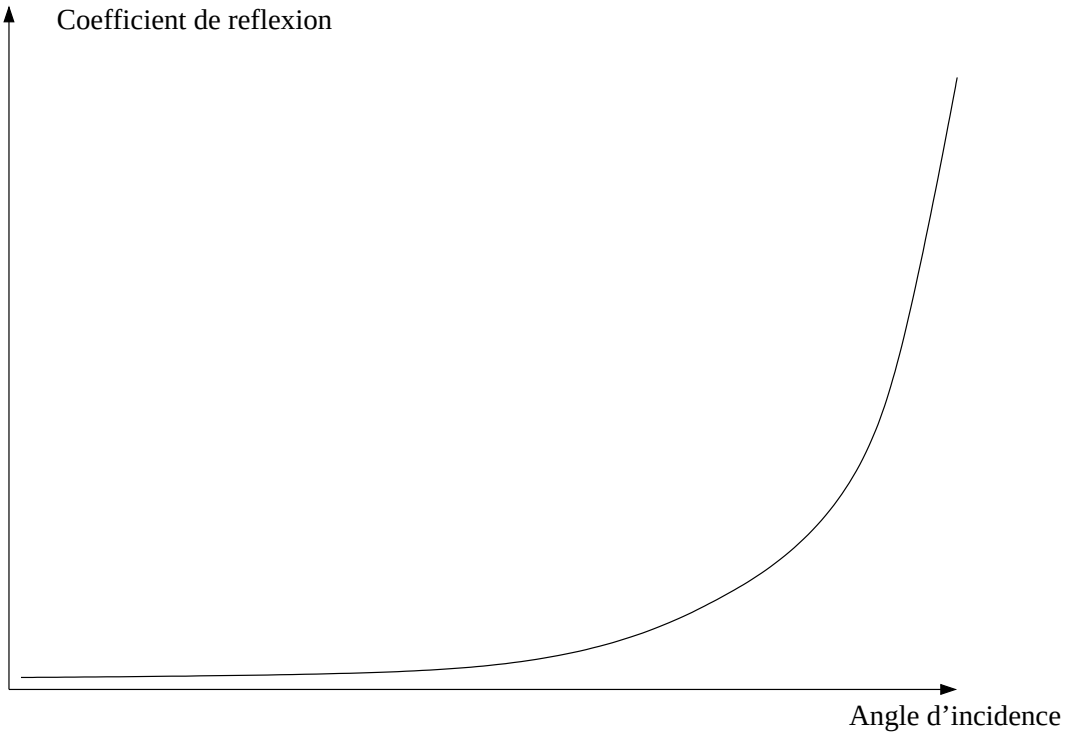


FIG. 1.10 – Allure de la courbe de Fresnel

Le cas des mers et océans est quelque peu différent du cas théorique pour plusieurs raisons. Les océans ne sont pas parfaitement transparents, il s'y produit des phénomènes d'absorption et de diffusion. Sous l'effet de la diffusion produite par les particules parasites (déchets organiques, plancton), un rayon est retransmis dans un grand nombre de directions. Certains de ces rayons sont retransmis vers la surface.

D'autre part, l'absorption variable de la lumière blanche du soleil produit un spectre coloré propre au milieu et dépendant de l'épaisseur traversée qui va être ensuite diffusé et donner une couleur particulière à

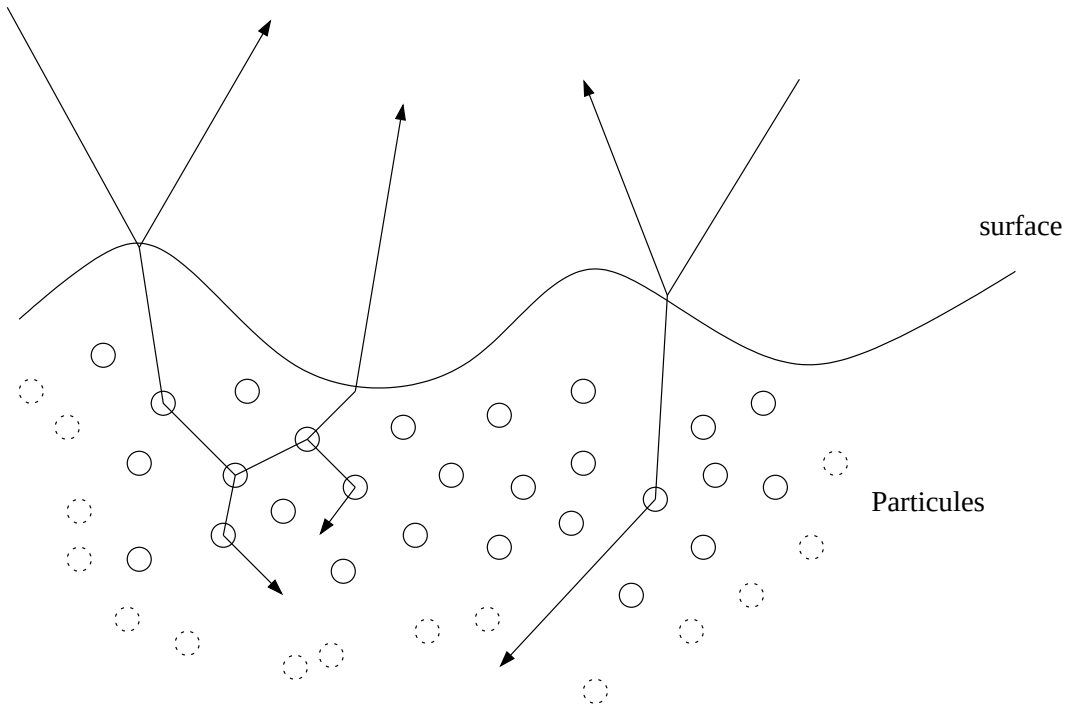


FIG. 1.11 – Diffusion des rayons par les particules présentes dans l'océan

l'océan. Ainsi, l'océan a une couleur dominante bleu mais la présence de plancton peut faire tendre cette couleur vers le vert.

Modélisation de l'optique en synthèse d'images

Malgré cette base théorique assez précise sur les caractéristiques optiques de la mer, les modélisations diffèrent sensiblement, bien que la géométrie soit globalement la même. Ainsi Peachey [14] utilise un maillage rendu par une interpolation linéaire sur chaque face des propriétés aux sommets. L'inconvénient de ce type d'approche est le fait que l'interpolation linéaire altère les reflets appliqués par une texture d'environnement. Cependant l'auteur appuie son choix sur le fait que les reliefs de la mer perturbent par eux même les reflets de telle sorte que l'on ne peut pas, dans la plupart des cas, reconnaître de forme précise sur la surface.
Mastin, Watterberg et Mareda [13] utilisent la même grille de hauteurs régulière mais hiérarchisent leur modèle par boîtes englobantes organisées en Quad-Tree. La méthode est alors basé sur un lancer de rayons mais les coefficients de réflexion et réfraction restent fixés et identiques pour toute la surface.
Ts'o et Barsky utilisent des textures de réflexion et réfraction composées selon le coefficient de Fresnel (approché par une fonction quadratique) pour palier à l'aspect plastique des deux approches précédentes.
Enfin, Fournier et Reeves [3] utilisent une texture d'environnement avec un rendu du type lancer de rayons, et composent avec des termes diffus et ambiants qui donnent une couleur, propre à l'océan, déterminée de manière empirique.

Limites pratiques

En pratique, plusieurs problèmes apparaissent. Le premier est la conséquence directe de la nature spéculaire de la surface. Il apparaît fréquemment des réflexions multiples à la surface de la mer (fig. 1.13) et cela ne peut pas être gérer qu'avec un algorihtme de lancer de rayons. Le deuxième problème exposé par Tessendorf [19] est du à l'aliasage généré par la résolution limitée de la camera. En effet, l'effet de réflexion produit par les longueurs d'ondes trop petites pour être affichées peut être modélisé comme une

FIG. 1.12 – Réflexion et rétro-diffusion de la mer

fonction de réflexion (BRDF). Cela modifie quelque peut la nature parfaitement spéculaire de la mer pour en faire une surface plus complexe à traiter.

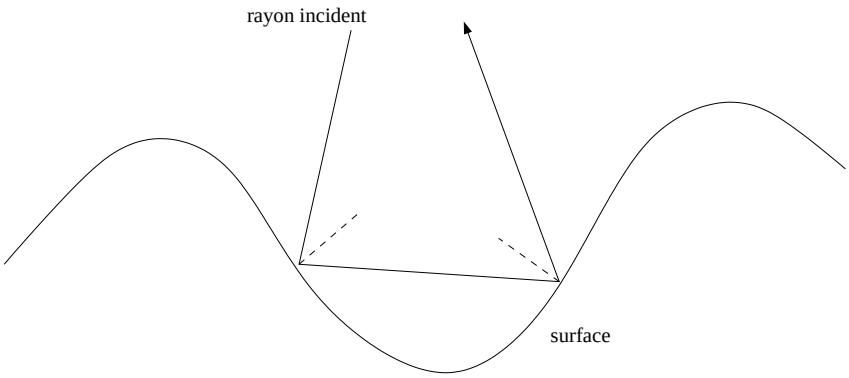


FIG. 1.13 – Réflexion multiple d'un rayon sur la surface de la mer

Chapitre 2

Mise au point d'une méthodologie

Les problèmes de la visualisation et de l'animation de la mer en temps réel couvrent un vaste domaine. En effet, comme cela apparaît dans les travaux précédemment réalisés, il existe d'une part, des approches phénoménologiques diverses offrant chacune leurs avantages et leur inconvénients et d'autre part, de nombreuses techniques de modélisation et de gestion de la géométrie dans une optique temps réel.
Malgré l'apparition de technologies qui tendent à utiliser des niveaux d'abstraction plus élevés [1] afin de s'affranchir des problèmes de bus sur PC[2], la grande majorité des cartes actuelles sont basées sur des primitives polygonales pour générer un rendu temps réel. Notre modèle devra donc fournir un ensemble de faces polygonales, si possible structurés de façon a être envoyés le plus efficacement possible à travers un bus, point faible du PC.
Dans cette optique, nous nous trouvons confrontés au problème de la gestion d'un maillage, géré directement ou généré à partir de structures plus complexes comme des surfaces paramétriques ou implicites.

Les caractéristiques du phénomène rendent la recherche d'une solution idéale particulièrement difficile. En effet, la vue peut couvrir un espace de grande dimension (jusqu'à l'horizon), sous un grand nombre d'échelles (premier plan et plans éloignés) mais aussi un espace restreint : lorsque la caméra est penchée vers le sol.
Ces contraintes nous obligent à nous appuyer sur une approche hiérarchique et sur une gestion de la simplification des détails pour gagner une efficacité. Il est donc nécessaire de mettre au point un méthodologie efficace pour traiter cet aspect.

2.1 Les niveaux de détail

Pour le problème auquel nous sommes confrontés, la notion de niveau de détail entre en jeu à plusieurs niveaux.
Tout d'abord, il s'agit de traiter le maillage de la surface de telle sorte qu'il s'adapte au mieux au contexte d'observation de la scène. Ensuite, en raison de la diversité d'échelles au sein même du phénomène et de l'ampleur des traitements entraînés, il nous faut également hiérarchiser le modèle en vue de choisir au mieux les détails à afficher pour maintenir des performances d'affichage constantes.

2.2 Modélisation

Pour les raisons exposées ci-dessus, nous avons décidé de séparer le traitement des niveaux de détail au sein du phénomène et de sa représentation géométrique. Cette approche est inspirée des travaux réalisés par Frank Perbet [15] dans le cadre de modélisation et animation de prairies en temps réel.

[1] La dernière génération de cartes Nvidia donne la possibilité de passer directement des surfaces (splines) à la carte qui les maille.
[2] Sur les architectures de type PC, la carte graphique qui pocède un processeur dédié aux fonctions de calcul 3d, est un périphérique dont la faible bande passante constitue un goulot d'étranglement limitant, dramatiquement parfois, la fréquence d'affichage.

Pour cela, nous allons créer des groupes élémentaires de phénomènes animés, ou «primitives de mouvement», qui vont agir sur des éléments de rendu, appelés «effecteurs» (fig. 2.1).

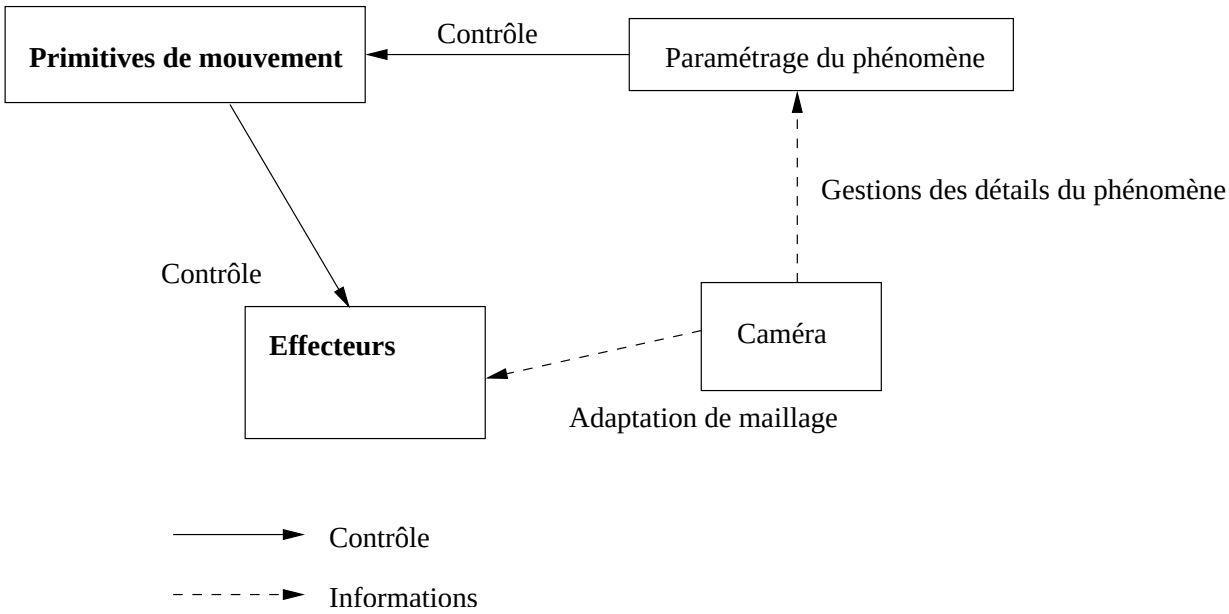


FIG. 2.1 – Modélisation par Primitives / Effecteurs

2.3 Application à la mer

En pratique, l'idéal pour préserver la souplesse de la méthode serait de séparer les primitives des effecteurs de façon à ce que les primitives permettent n'importe quelle modélisation des effecteurs, par exemple une grille de hauteur ou une surface implicite.
Concrètement, ici, les primitives vont représenter une vague (ou un ensemble de vagues) et les effecteurs un maillage destiné à être animé par les vagues.
Pour cela, notre choix va être de considérer la surface de la mer comme une surface paramétrique, c'est à dire dont on peut tirer les coordonnées en tout point en fonction de paramètres. Notons que cette modélisation est plus intéressante que celle considérant une surface comme paramétrée sur le plan horizontal puisque cette dernière interdit les recouvrements (problème des champs de hauteur).
Ce modèle paramétrique représentera, dans la suite de l'étude, la contrainte explicite de modélisation, à la fois pour les primitives et pour les effecteurs, qui permettra de mettre en place une liaison entre les deux.

2.4 Les primitives de mouvement

Pour obtenir une approche efficace, il convient de bien étudier les différents phénomènes à représenter.

2.4.1 Énumération des phénomènes

Le principal phénomène à l'origine de perturbations sur la surface de la mer est la houle. Comme exposé précédemment, il s'agit d'ondes gravitationnelles qui peuvent être modélisées par des trochoïdes dont on contrôle la longueur d'onde et l'amplitude, la pulsation étant asservie au vecteur d'onde par la physique.
A une plus petite échelle, on trouve également les ondes capillaires, générées souvent par le vent et qui doivent donc obéir à un contrôle particulier.
On peut ensuite ajouter un certain nombre de phénomènes qui dépendent du but de l'application, comme des sillages de bateaux.

2.4.2 Choix d'une modélisation

Les différentes approches possibles sont essentiellement basées sur deux approches exposées en état de l'art, celle d'une génération stochastique et celle d'une génération physique.
Le modèle choisi dans le cadre de ce projet a été celui de l'approche paramétrique tiré de la théorie des ondes pour différentes raisons. La première est la souplesse de ce modèle pour l'animation, qui constitue un aspect essentiel du projet. D'autres raisons, que nous expliciterons plus loin, sont liées à l'aspect temps réel. Tout d'abord, soulignons que, dans le cas du modèle stochastique, la génération par transformée de Fourier (algorithme FFT : Fast Fourier Transform) pose des problèmes de temps de calcul, car la complexité de ce type d'algorithme le limite à une implémentation de faible résolution lorseque le calcul doit être effectué en temps réel. De plus, l'ampleur des échelles différentes interdit pratiquement tout pré-calcul, et le filtrage en fonction de la vue constitue une méthode de gestion de niveau de détail particulièrement efficace comme nous le verrons au chapitre 3.

Pour ces différentes raisons, nous avons choisi d'implémenter des modèles classiques de houle basés sur des trochoïdes. Le premier problème apporté par cette approche réside dans la trochoïde elle même.

$$\begin{cases} x - x_0 = -\xi_0 \, \mathrm{e}^{k z_0} \sin(k x_0 - \omega t) \\ z - z_0 = -\xi_0 \, \mathrm{e}^{k z_0} \cos(k x_0 - \omega t) \end{cases}$$

Étant donné sa nature même (une oscillation horizontale et une verticale), la paramétrisation classique par le plan horizontal (champ de hauteur) n'est pas possible. Rappelons encore une fois que c'est ce qui permet à la trochoïde de modéliser les recouvrements engendrés par certaines vagues dont la crête passe au dessus de la base.
Thon[20] se ramène tout de même à un champ de hauteurs en pré-calculant l'aspect de la trochoïde sur une période. Nous n'allons pas utiliser cette technique car elle est contraignante pour le résultat (les vagues ne peuvent plus se retourner) et n'est pas, à notre sens, nécessaire. En effet, en raison de la faible amplitude théorique des oscillations des particules fluides, nous allons considérer la surface comme paramétrée par sa position au repos, c'est à dire la surface constituée par un maillage dont les sommets sont les centres des particules. Il s'agit en réalité, dans le cas de la mer, de la surface horizontale de la mer au repos.
Même si ce paramétrage peut paraître inapte à fournir suffisamment d'information en fonction seulement du centre (notamment pour l'adaptation de maillage), les données de l'oscillation nous garantissent une marge d'erreur connue et les caractéristiques de la trochoïde garantissent une topologie identique à celle de la surface au repos.
Nous avons donc choisi un type de modélisation de primitives et un paramétrage apte à contrôler les effecteurs de la façon introduite au chapitre 2.

Parmi les travaux déjà effectués la base des trochoïdes, la plupart couplent cette méthode avec des bruits ou perturbations plus ou moins complexes. Cela implique un manque de souplesse, soit au niveau de la complexité soit au niveau de la modélisation, de la gestion des trochoïdes. En raison du fait que le phénomène

à modéliser n'est qu'une composition d'un très grand nombre de primitives, nous avons décidé de choisir une approche différente des méthodes classiques. En effet, plutôt que de bruiter nos données, nous allons construire un modèle de traitement des primitives par des méthodes de niveaux de détail qui vont nous permettre d'en afficher le maximum possible et ainsi de rompre les motifs répétitifs qui pourraient apparaître.Néanmoins, le positionnement de notre approche n'est pas de remplacer l'enrichissement par perturbation, mais seulement de modéliser le point fort de la méthode, la trochoïde, de façon efficace, ce qui peut permettre dans certains cas de se passer des perturbations.

2.4.3 Composition des primitive (fig. 2.2)

La modélisation est implicitement basée sur la facultée des primitives à se composer entre elles. Néanmoins, la modélisation paramétrique de celles-ci ne fournit pas de solution a priori pour la composition. En effet, il s'agit de mélanger les oscillations de particules ayant lieu dans divers plans, et la physique ne fournit pas d'information triviale sur cette composition.
Dans un soucis de simplicité et d'efficacité, nous avons choisi de composer les oscillations de manière additive selon le centre (position au repos) de la particule. Nous n'apporterons, pour ce choix aucune justification théorique mais la surface obtenue semble correspondre à celle recherchée. Une motivation a ici été la nécessité de pouvoir composer les primitives indifféremment dans n'importe quel ordre, ce qui n'aurait pas été le cas si, par exemple, nous avions choisi comme position au repos pour la deuxième primitive la surface générée par la première.

Intuitivement, cette solution correspond à composer d'une part deux oscillations verticales (comme dans le cas d'un champ de hauteur pur) et deux oscillations horizontales (selon les deux variables x et y). on obtient donc, sur le plan horizontal, une ellipse et sur l'axe vertical, la composition de deux sinusoïdes.

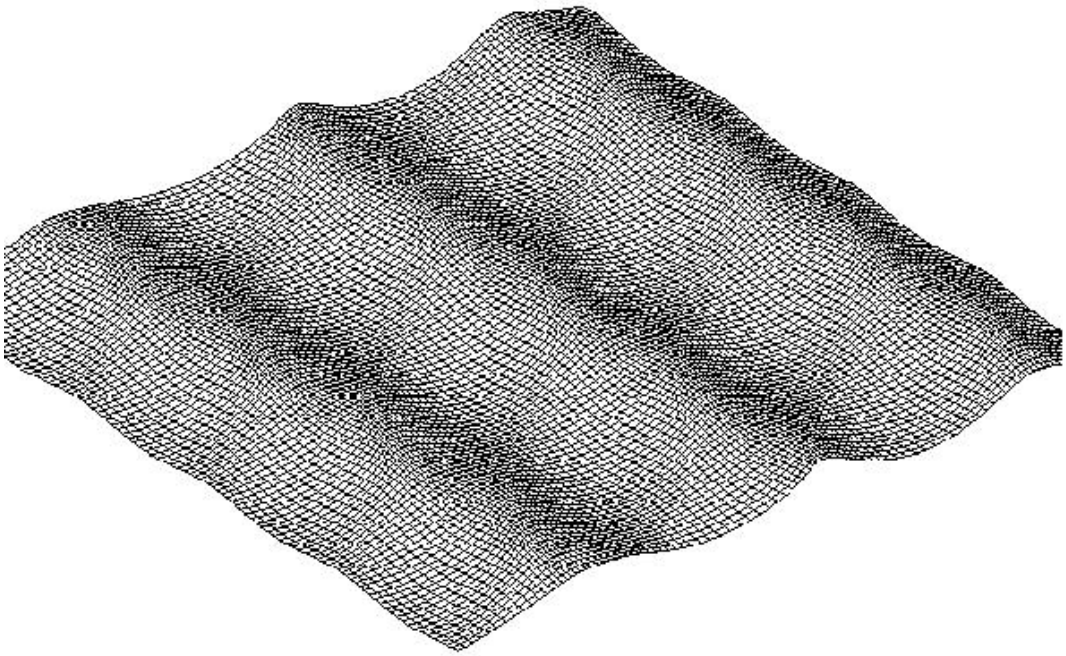


FIG. 2.2 – Composition de deux trochoïdes

2.4.4 Génération des primitives

La génération des primitives doit être adaptée de façon automatique, afin de permettre un maximum de diversité, et hiérachiser efficacement les primitives dans une strucutre adéquate. En pratique, si une primitive contient une vague, chaque vague sera traitée en niveau de détail d'où une grande efficacité. Mais si plusieurs vagues ont les mêmes caractéristiques du point de vue du traitement, elles peuvent être regroupés dans une seule primitive afin de gagner du temps de traitement.

Une approche non étudiée ici pourrait être de gérer les niveaux de détails sur les primitives à travers une structure hiérarchique, c'est à dire de faire du niveau de détail de niveau de détail (fig. 2.3). Ici, nous nous contenterons de hiérarchiser les primitives pour traiter celles de dernier niveau. L'automatisation de la génération n'aura alors plus, comme tache, qu'à choisir le niveau de regroupement des primitives en fonction d'un paramètre choisi (donc de façon statique et non dynamique). Ces notions seront étudiées plus en détail dans le chapitre 3. Nous verrons également au chapitre 4 que la structure implémentée dans notre modèle est naturellement orienté vers cette approche dynamique.

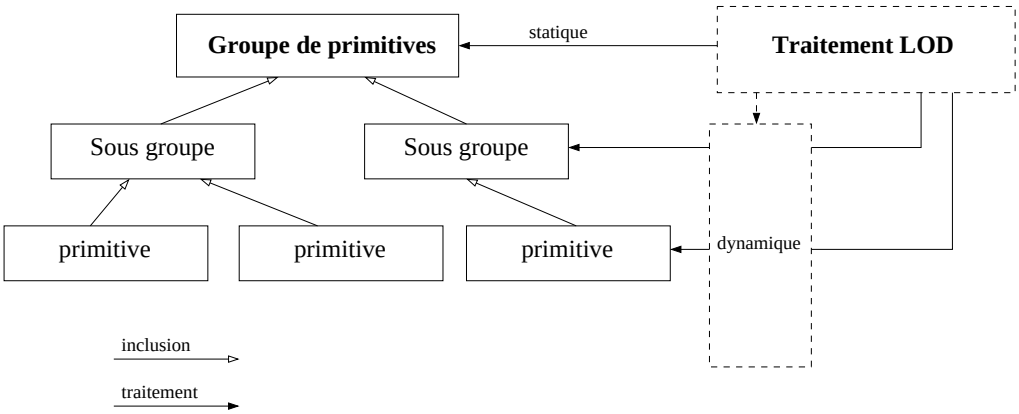


FIG. 2.3 – Hiérarchisation des primitives

2.5 Les effecteurs : de la géométrie au rendu

Comme défini par la méthodologie, les effecteurs sont les connexions entre les primitives de mouvement et le rendu visuel. Dans une approche temps réel, il est difficile de contourner l'asservissement des effecteurs à un maillage. Dans ce chapitre, nous allons étudier le rendu de la mer afin de déterminer une technique de modélisation efficace.

2.5.1 Étude du phénomène

Bien que l'optique de la surface soit relativement simple, le rendu de la mer pose un certain nombre de problèmes. Dans une approche temps réel, il est très difficile de gérer la complexité nécessaire pour rendre compte fidèlement de toutes les perturbations de la surface qui lui donnent cet aspect si particulier (photo 2.4).

FIG. 2.4 – Image réelle

Comme nous l'avons indiqué précédemment dans la modélisation des primitives, l'aspect de la surface de l'océan tient à sa composition d'un très grand nombre d'ondes de directions et longueurs d'ondes différentes. Pour continuer en cohérence avec notre méthodologie générale, nous allons ici nous intéresser au rendu de l'eau indépendamment de sa méthode de génération. Nous allons donc étudier comment nous pouvons rendre la surface de la mer sur une surface maillée pour un affichage temps réel.

Réflexion

L'aspect primordial de la surface de la mer est la réflexion spéculaire parfaite. L'approche de rendu la plus adaptée est, comme l'a montré l'état de l'art, l'utilisation d'une texture d'environnement. Néanmoins,

les textures d'environnement implémentés dans la librairie openGL sont limitées, elles ne tiennent pas compte de l'emplacement de l'observateur, et se contentent de calculer une projection de la texture en fonction des normales uniquement.
Afin de préserver une cohérence avec l'animation et l'interactivité (à travers les mouvements de la caméra), nous allons construire notre propre système de calcul des coordonnées de plaquage de texture pour chaque point en fonction de sa normale. Ce calcul est peu pénalisant pour l'application car il consiste simplement à calculer la réflexion du vecteur reliant le point de vue au sommet considéré.

Comme indiqué sur la fig 2.5, en considérant le trajet comme inverse par rapport à celui de la lumière (nous partons de l'observateur pour arriver à l'objet reflété), le rayon réfléchi est calculé géométriquement à partir de la direction du rayon incident et de la normale. Il s'agit de calculer le symétrique du rayon incident par la normale. Si l'on considère une base dans laquelle la normale constitue le vecteur z, il suffit alors simplement d'inverser la coordonnée z du rayon incident.

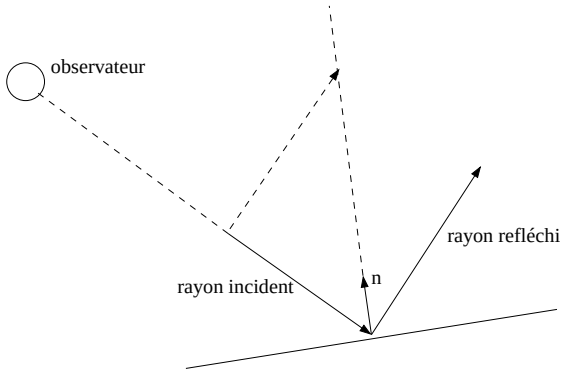


FIG. 2.5 – Calcul du rayon réfléchi

Une fois le rayon réfléchi calculé, il ne reste plus qu'à trouver le pixel vers lequel il pointe dans la texture d'environnement. Nous avons pour cela simplifié en choisissant une texture hémisphérique dont la base est horizontale. Il suffit alors de prendre la projection du rayon réfléchi sur le plan horizontal pour obtenir les coordonnées du point dans la texture (fig. 2.6).

L'interpolation réalisé par l'algorithme de texturage étant linéaire, elle ne produit pas exactement le résultat que nous devrions avoir. En fait, notre texture d'environnement est sphérique, donc cela considère les face comme ayant une géométrie localement sphérique entre les normales fournies aux sommets : on ne peut pas avoir de reflet planaire sur une face). Néanmoins cela est peu perçu car la courbure réelle des faces est souvent assez proche de celle d'une sphère de grand rayon (comme ici). Il y a cependant des cas où cette tendance au lissage "rond" peut être ennuyeuse (cas où la face devrait contenir une arête saillante), c'est un problème classique d'aliasage, il n'existe pas de solution à priori. Dans les cas où les ressources le permettent, le sur-échantillonnage est une solution acceptable.
Néanmoins, dans notre cas, ce problème apparaît assez rarement, considérant la relative continuité de la courbure de la mer (dans la plupart des cas) et la qualité de l'échantillonnage permise par le matériel. De plus, quand ce problème apparaît, il passe le plus souvent inaperçu en raison de l'aspect quelque peu imprévisible de la surface.

Habillage

Nous avons vu comment plaquer une texture d'environnement sur les faces de notre modèle. Ici, nous allons étudier la possibilité d'afficher des détails de taille inférieure à la résolution du maillage.

Il faut souligner le fait que nous allons tenter d'exploiter, de plusieurs façons différentes (géométrie et bump mapping), des informations fournis de la même manière à toutes les échelles (petites et grandes : pas de fonctions de pertrurbation). Cela a comme intérêt de rendre le modèle adaptable à l'échelle (niveaux de détails) et comme inconvénient de nécessiter une continuité parfaite entre les différents types de

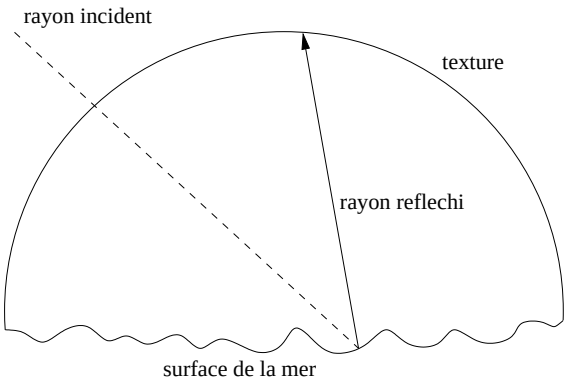


FIG. 2.6 – Détermination des coordonnées de texture

représentation des données.

En pratique, nous allons vouloir ajouter des détails sur une face, sachant que nous connaissons les perturbations à apporter à celle-ci et que ces perturbations sont de nature totalement spéculaire.

La première approche qui vient directement à l'esprit est le bump-mapping. Cela consiste, par définition, à perturber l'interpolation de la normale entre les sommets de la géométrie en fonction d'une texture de variations de hauteurs. Néanmoins, en pratique, la technique la plus utilisée, connue sous le nom d'«emboss bump mapping», consiste à composer la texture à appliquer avec une texture calculée modifiant la luminosité de façon à éclaircir ou assombrir les texels[3] en fonction de l'orientation des normales au relief avec la lampe (fig. 2.7).

En pratique, la soustraction de la texture de relief avec une autre version d'elle-même décalée dans une direction fournit une texture de gradient dans cette direction. En réitérant ce calcul dans une deuxième direction ou en choisissant une direction adéquate (direction de la lampe), on obtient une texture de gradients en niveaux de gris,entre 0 et 1, dont l'origine est en 0.5 (décalage de $+0.5$). Ce qui a la propriété de fournir du blanc (respectivement du noir) dans les zones de gradient positif (respectivement négatif). Cette texture peut alors directement être mélangé avec la texture d'habillage pour donner une impression de relief.

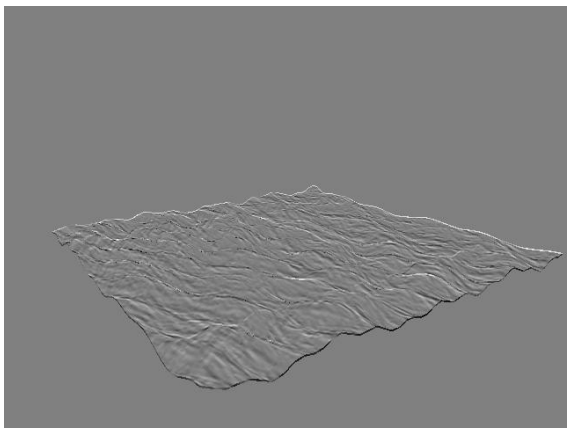


FIG. 2.7 – Emboss bump mapping

[3]Point d'une texture.

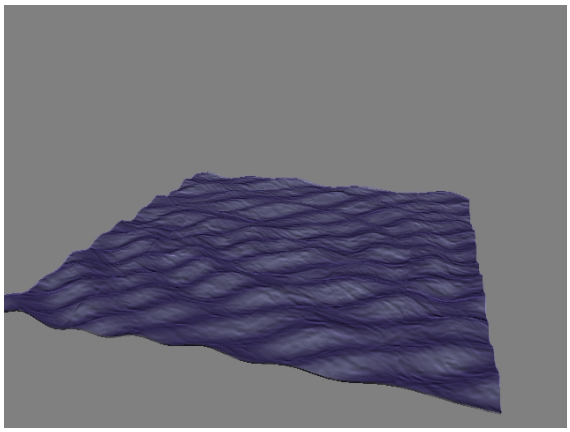


FIG. 2.8 – Réflexion et bump mapping

Néanmoins, l'utilisation que nous voudrions faire du bump mapping est de véritablement modifier les normales dans le calcul de réflection. Sans cela, nous donnerions un aspect plastique à la surface (fig. 2.8) car ses caractéristiques seraient un éclairage diffus et non une réflexion pour les détails. Mais cela n'est pas réalisable sur la plupart des cartes 3D actuelles. En effet, la remplissage (et texturage) des primitives d'affichage est forcement fait de la même façon : un texturage linéaire mélangé avec un éclairage lui aussi linéaire.
Certains matériels, notamment les récentes cartes Nvidia sur PC, permettent d'écrire soit même les opération à réaliser en fonction de textures et donc de réaliser un véritable bump mapping (comme c'est le cas des rendus non temps réels).
Mais le bump mapping pose encore d'autres problèmes. Pour être implémenté, il nécessite de fournir au matériel des textures comportant (au minimum) les caractéristiques du relief. Or, dans notre modélisation, ce relief est animé, et pour pouvoir être adapté à des échelles différentes, doit pouvoir être généré à tout moment. La complexité de la génération de textures de relief en temps réel ainsi que le coût de son passage à travers les bus des PC rendent son implémentation trop coûteuse pour le but à atteindre. En effet, selon le matériel, il peut être plus avantageux de passer à travers le bus une géométrie adaptée (les textures ne sont souvent pas affichés à la résolution pour laquelle elles sont optimale). C'est l'approche que nous avons choisi. Ce choix est néanmoins discutable, en particulier en fonction du matériel vers lequel est ciblée l'application.

Un approche qui pourrait régler le problème de la génération des textures pour chaque image consisterait à pré-calculer un relief animé. On revient alors à l'introduction de bruit ou de perturbations et cela aurait pour effet, à ce niveau là de rompre la continuité nécessaire aux niveaux de détails.

Chapitre 3

Niveaux de détail et aliasage

L'architecture du modèle a été construite autour de l'utilisation de niveaux de détails ou plus précisément de sa version continue, c'est à dire l'adaptation de la complexité de la scène en fonction des paramètres de la vue. Comme introduit au chapitre 2, la gestion de la complexité entre en jeu pour deux aspects distincts du problème : les primitives de mouvement et les effecteurs. Nous allons commencer par nous intéresser au cas des effecteurs avec la mise en place et l'adaptation d'un maillage avec les paramètres de caméra. Nous verrons ensuite comment un filtrage des primitives est utilisé, à la fois pour accélérer le traitement et pour régler les problèmes d'aliasage.

3.1 Adaptation de maillage

L'architecture du système de primitive nous permet de connaître l'emplacement de n'importe quel point de la surface au repos (ainsi que l'amplitude de la variation). En considérant que la variation est peu importante par rapport à la taille de l'espace observé, nous allons construire un maillage et adapter la position de ses sommets à chaque modification des paramètres de la vue.

3.1.1 Conservation de la topologie

Nous allons utiliser la propriété suivante : le paramétrage des primitive est supposé conserver la topologie de la surface (deux points ne peuvent se croiser au cours de leurs oscillations).Cette propriété nous permet de construire le maillage sur la surface au repos au lieu d'avoir à mailler un ensemble de points auxquels on a appliqué les primitives de mouvement.
De plus, cette approche va nous permettre de conserver le maillage en mémoire (la topologie : les polygones de la surface au repos), ce qui a pour intérêt premier de limiter les gestions complexes d'espace mémoire. Enfin, cela entraîne que la densité de polygones affichés (et calculés) reste constante (pas de création ou de suppression), ce qui est généralement ce que l'on recherche dans une application temps réel car cela permet une certaine constance (modulé par le reste de l'application) autour de la fluidité choisie (fixée par la complexité du maillage).

3.1.2 Aliasage des primitives : introduction

Il est difficile de décrire séparément le traitement des primitives de celui des effecteurs, car ils s'appuient mutuellement l'un sur l'autre malgré leur volontaire séparation. C'est pourquoi nous devons commencer par introduire le concept de filtrage qui sera explicité plus en détail en 3.2.
Si l'on considère un maillage ayant une densité spatiale (autrement dit une certaine densité dans l'espace du phénomène, ici la mer), ce maillage sera apte à afficher uniquement des primitives de taille supérieure à un seuil en deçà duquel on aura un phénomène d'aliasage. C'est un problème qui apparait lorseque l'on

cherche à discrétiser un phénomène dont la fréquence de variation n'est pas suffisament supérieure au pas de discrétisation[1] (fig. 3.1). Le phénomène obtenu est alors une convolution des deux fréquences, pouvant donné un résultat pseudo aléatoire ou un motif parasite dans le cas où les fréquences sont multiple l'une de l'autre.

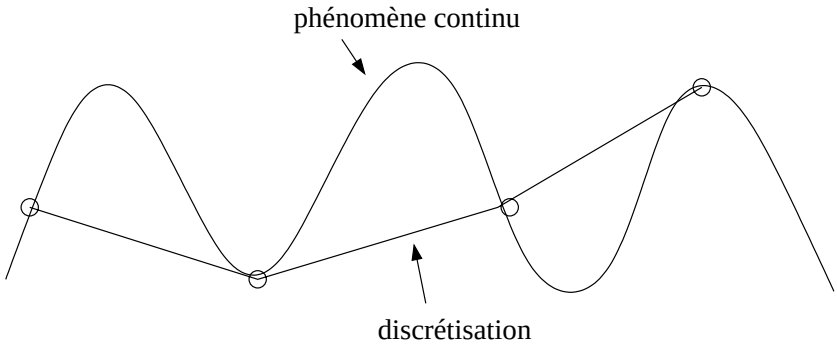


FIG. 3.1 – Aliasage

Pour la mer, les primitives sont des vagues d'une longueur d'onde donnée. Si l'on considère des longueurs d'onde inférieure à la taille d'un polygone, ses sommets tomberont indifféremment sur des points qui n'auront plus de cohésion visuelle entre eux ; c'est l'aliasage. En fait, cette limite se trouve au delà d'un polygone par période d'oscillation. Il faut compter un minimum de quatre polygones dans une période pour ne pas avoir d'aliasage, et cette limite ne garantie pas un résultat visuellement acceptable.
La conséquence de ce phénomène dont nous allons nous servir dans ce qui suit est qu'il existe une limite inférieure des détails qu'un polygone peut afficher, et cette limite est fonction de sa taille, donc, dans le cas d'un maillage, de sa densité.

3.1.3 Adaptation du maillage

Nous allons réaliser une adaptation de la densité spatiale du maillage afin de garder une surface constante sur l'écran pour chaque polygone. C'est une méthode qui peut possède les caractéristiques d'un lancer de rayons pour ce qui est des problèmes d'aliasage : le maillage introduit une variation de densité d'échantillonnage du phénomène qui, lui, a une fréquence de détail constante dans l'espace pour permettre d'être visualisé d'un point de vue quelconque.

Projection du maillage

Pour calculer la projection du maillage désiré sur la surface au repos, nous avons implémenté un lancer de rayons[2]. Il s'agit de parcourir le plan image (dans une base propre à la caméra) et calculer l'intersection du rayon projeté à travers chaque point[3] avec la surface au repos (fig. 3.2 et 3.3). Cette approche se trouve être rentable en temps de calcul puisque la surface au repos est un plan et qu'il s'agit du plan horizontal dans la base du monde. Notons qu'une technique plus générale, par exemple en calculant la matrice de projection, serait souhaitable pour une application dans des cas plus généraux.

[1] on admet qu'un pas de discrétisation 4 fois inférieur à la fréquence du phénomène constitue un minimum acceptable.
[2] Au sens premier : calculer un rayon partant du point de vue et arrivant sur la surface en passant par un point donné du plan de la caméra.
[3] moyennant un échantillonnage de la surface en fonction de la surface désirée sur le plan caméra pour chaque polygone.

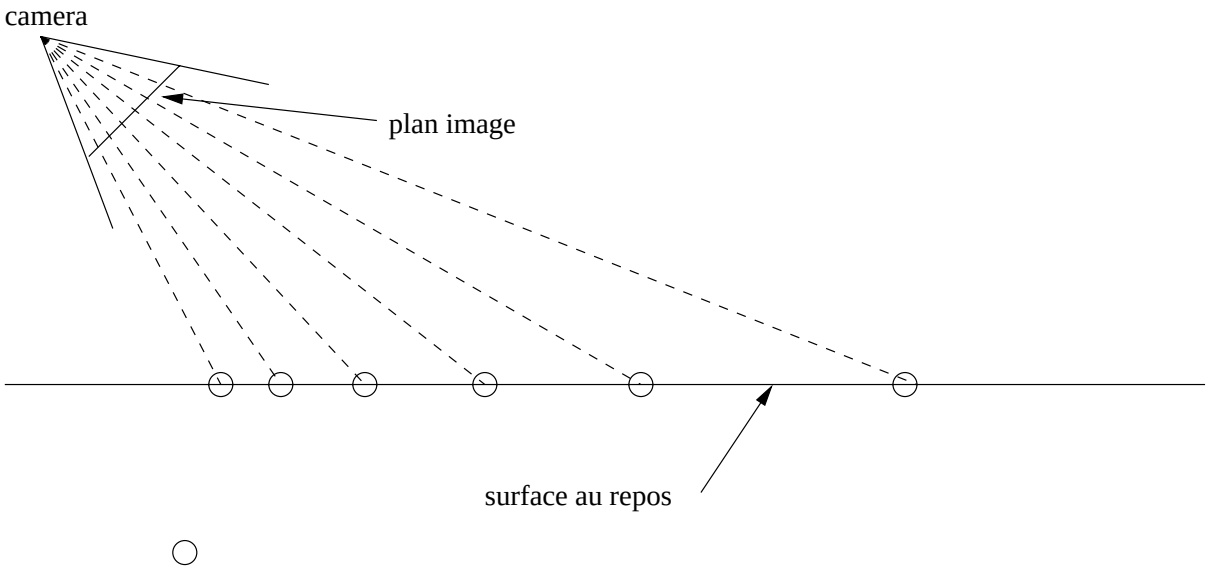


FIG. 3.2 – Projection du maillage

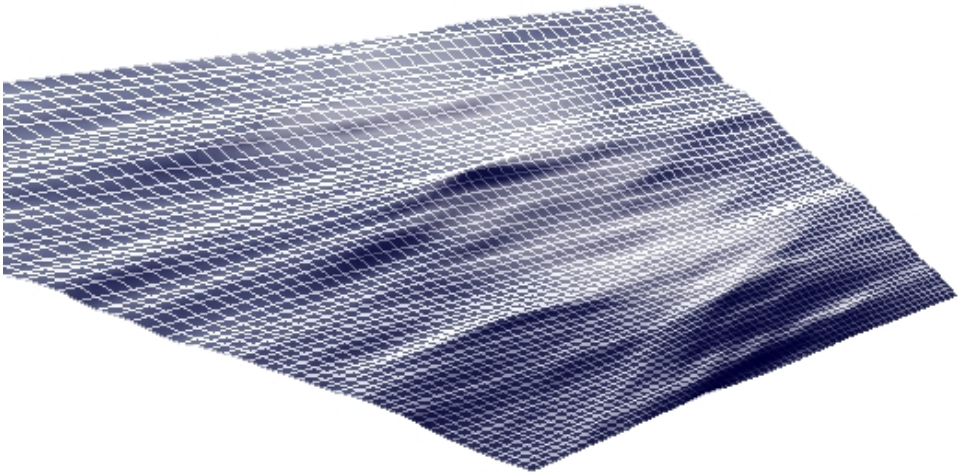


FIG. 3.3 – Projection du maillage, résultat

Néanmoins, cette approche, si elle parait naturelle, peut être largement discutée. En effet, lorsque la vue se trouve au voisinage de l'horizontale, c'est à dire lorsque l'observateur voit l'horizon et ainsi la mer à plusieurs échelles, il peut porter son attention sur différents plans (fig. 3.4).
Augmenter la densité de maillage au premier plan (surface en pixels des polygones moins importantes) correspond surtout à diminuer la densité à l'horizon et ainsi à considérer que les détails ne sont pas intéressants. Cette approche peut-être appuyée par le fait que, pour une surface constante, la densité spatiale à l'horizon est déjà faible, et il arrive souvent (mer calme) que la taille du maillage soit trop importante pour rendre la plus grande longueur d'onde. On peut alors choisir de mailler le plus grossièrement possible, ou d'afficher l'horizon avec une technique différente. Cette approche ne sera pas étudiée ici, nous nous sommes cantonnés à une seule représentation pour toutes les échelles afin de conserver au maximum la continuité

des transitions.
L'inverse est aussi envisageable : il serait possible de défendre le technique qui consiste à augmenter la densité de maillage à l'horizon pour la faire au maximum correspondre aux plus gros détails à afficher. Cela entraînerait bien sur une diminution de la densité de maillage au premier plan. Ce serait probablement plus efficace pour le cas où l'intérêt de l'observateur se porte sur un plan médian.

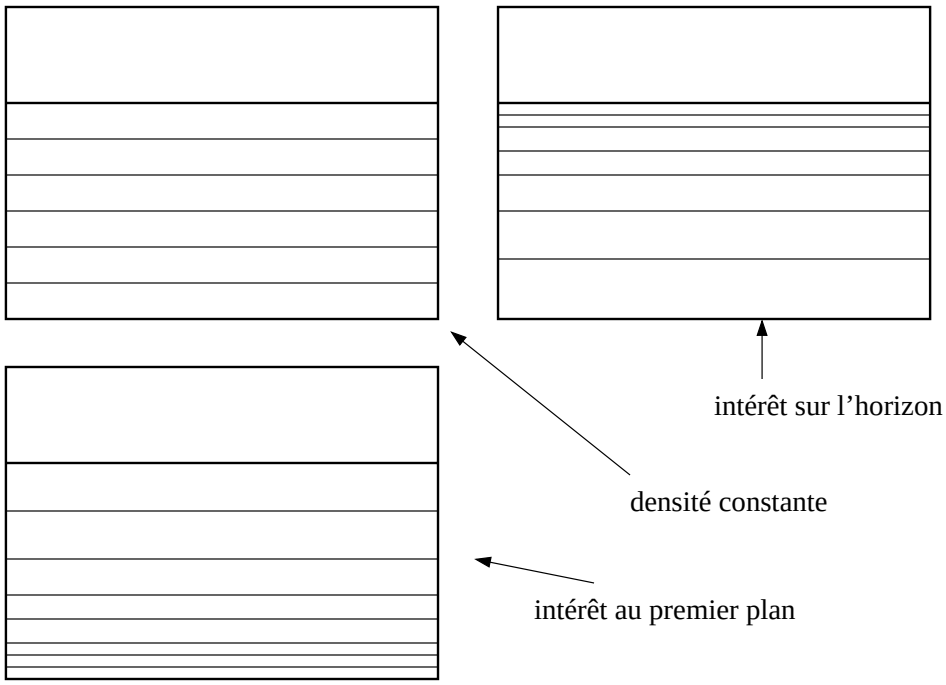


FIG. 3.4 – Les différentes approches de maillage

En fait, aucune solution n'est optimale, si ce n'est celle qui consiste à adapter la densité de maillage à la fois au phénomène et au centre d'intérêt de l'observateur (la mer a peu d'intérêt à être visualisée seule dans le cadre d'une application interactive, il existe forcément un centre d'intérêt dicté par l'action, même s'il s'agit d'une vague plus grosse que les autres).

3.2 Aliasage et niveaux de détail : filtrage

L'aliasage est un problème essentiel dans le cadre de la modélisation de la mer. En effet, beaucoup d'échelles de détail différentes sont présentes, et la discrétisation nécessaire à la modélisation entraîne obligatoirement un phénomène l'aliasage. Il est donc essentiel de mettre en place un système de filtrage.
Comme introduit en 3.1.2, l'aliasage apparaît avec la discrétisation d'un phénomène dont la fréquence est supérieure ou égale à celle d'échantillonnage. Il existe différentes méthodes de traitement, mais la plupart se rapportent directement ou indirectement à une technique de filtrage, car la solution la plus efficace consiste à supprimer les fréquences supérieures ou égales au taux d'échantillonnage.

Dans le cas de la mer, l'approche est particulièrement simple puisque les données décrivant le phénomène ont comme caractéristique intrinsèque une longueur d'onde (caractéristique des primitives), que l'on peut donc directement comparer à la densité de maillage (effecteurs). De plus, notre approche permet d'avoir des informations de densité locale, ce qui permet d'adapter le filtrage différemment en chaque point pour une même vue. Cela s'avère essentiel dans le cas d'une vue qui s'étend jusqu'à l'horizon, et l'expérience

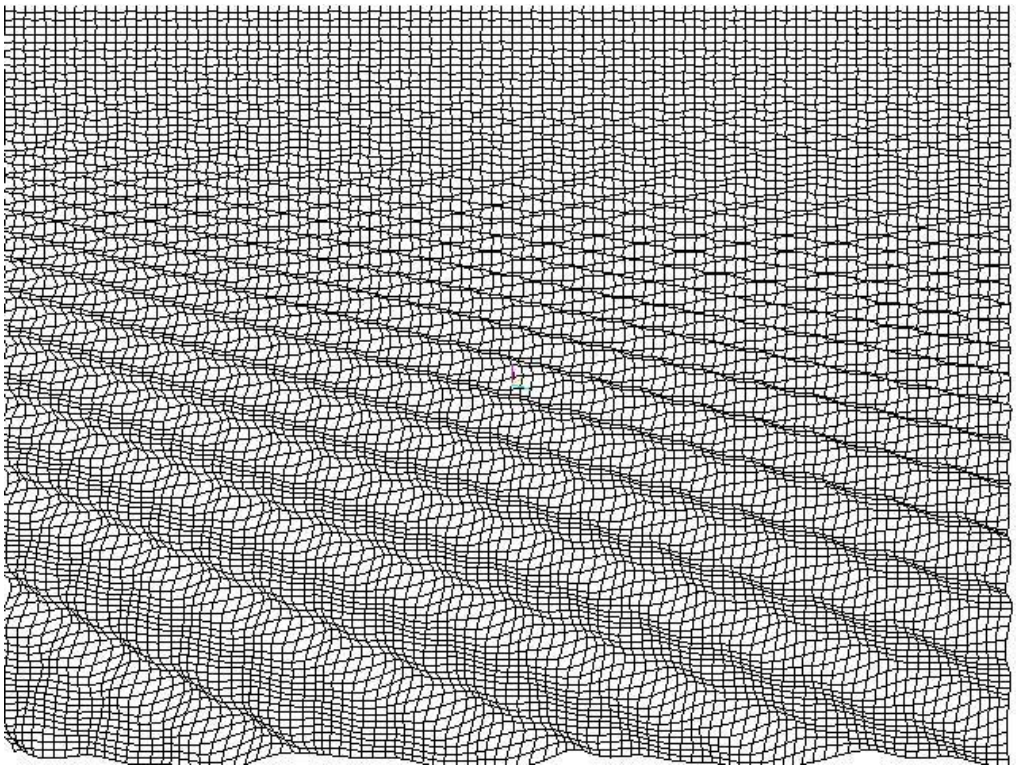


FIG. 3.5 – Phénomène d'aliasage géométrique lors de la projection du maillage : au deuxième plan, l'onde est altérée, à l'arrière plan : l'onde n'est plus visible

montre que c'est souvent le cas pour des vues plus restreintes.

3.2.1 Calcul des normales

Avant d'étudier le filtrage proprement dit, nous allons nous intéresser à un phénomène qui altère sensiblement la façon de traiter le problème, il s'agit de la façon de calculer les normales.
En réalité, l'aliasage purement géométrique (celui introduit en 3.1.2) est peu visible pour le cas qui nous concerne, car la visualisation qu'on utilise etant uniquement un reflet, le rendu est essentiellement basé sur les normales affectées aux polygones. L'aliasage qui apparaît dans la figure 3.5 passe relativement inaperçut avec l'habillage. Par contre, l'aliasage du à la discrétisation des normales sur le maillage est très visible, c'est lui que nous allons devoir traiter. Ce phénomène dépend bien sur fortement de la façon dont on calcule les normales (fig. 3.6 et 3.9).

En effet, il existe deux techniques qui illustrent parfaitement ce discours. La première consiste à calculer les normales numériquement à partir du maillage. Dans ce cas, il n'y aura pas d'aliasage du aux normales puisque elles sont cohérentes avec le maillage lui même. Le seul aliasage gênant sera purement géométrique.

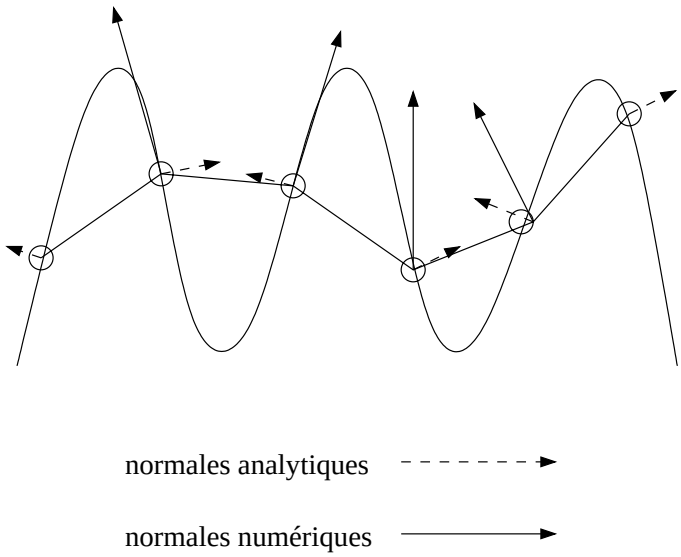


FIG. 3.6 – Normales analytiques et numériques

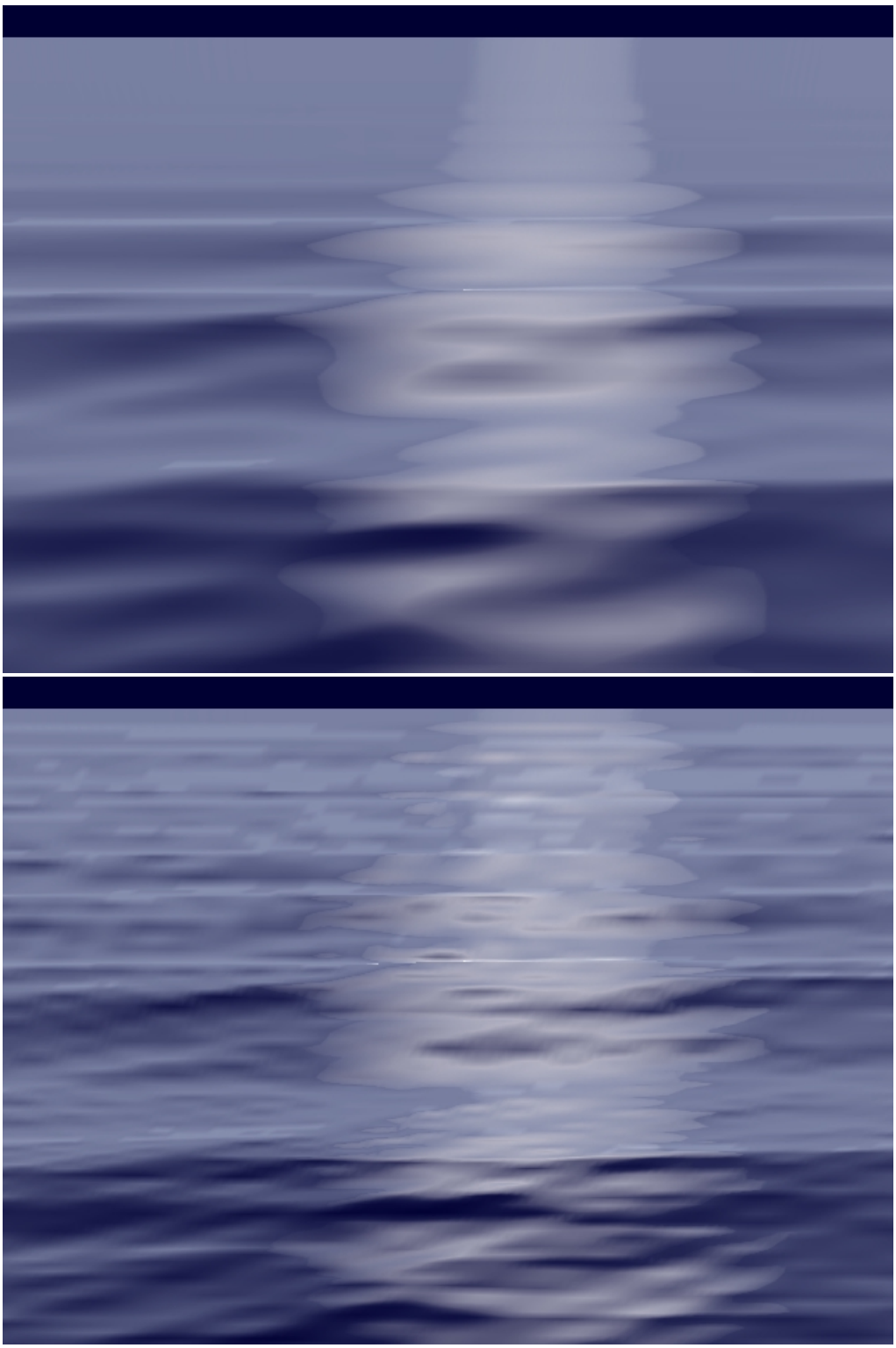


FIG. 3.7 – Normales analytiques et numériques : plus de détail mais aliasage à l'horizon dans le second cas.

Néanmoins, cette approche entraîne évidement une perte de donnée (il s'agit en fait d'un lissage). Cette perte est non négligeable dans le cas où la taille de la géométrie est de l'ordre de celle des détails à rendre et c'est effectivement le cas recherché en temps réel.
Mais l'utilisation de normales calculées analytiquement (et fournissant donc des informations supplémentaires) entraîne un aliasage tellement important que les informations en deviennent complètement corrompues. Nous allons donc traiter ce problème avec un filtrage comme suggéré ci-dessus.

3.2.2 Filtrage

Cette technique va simplement consister à appliquer un filtre sur le spectre des ondes de la mer en chaque point du maillage pour ne garder les déplacements et surtout les tangentes que lorsque la fréquence est affichable, selon la taille locale du maillage. Rappelons que la densité d'échantillonnage (donc la taille du maillage) est choisie constante sur le plan image, donc varie dans l'espace du phénomène (fig. 3.2).
En pratique, le filtre doit être progressif pour ne pas entraîner de frontières trop brusque sur l'image (celles-ci seraient alors fonction de la variation de répartition du maillage, donc, dans notre cas, horizontales sur l'écran).

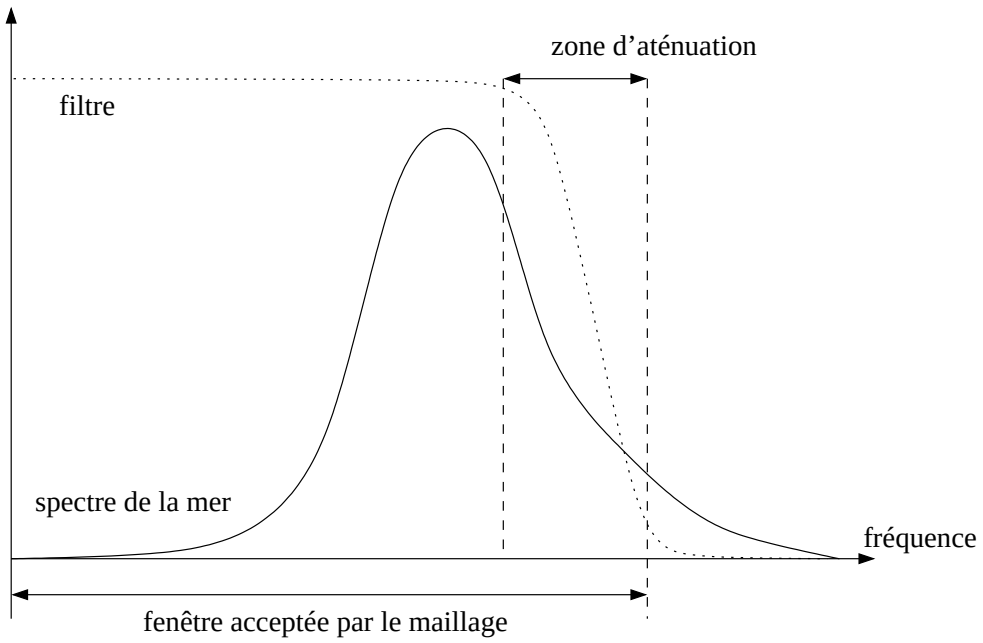


FIG. 3.8 – Allure du filtre (localement, les limite de maillage varient avec sa densité)

Dans notre cas, le phénomène d'aliasage peut être recherché dans une certaine mesure. En effet, lorsque l'on omet les trop grandes longueurs d'onde (souvent le cas car les variations sont très peu visibles à petite échelle), l'horizon apparaît relativement uniforme. Un certain aliasage peut alors rendre sa représentation plus agréable.
En fait l'expérience montre que le rendu accepte un filtrage beaucoup plus souple que la théorie ne le permet. Cela peut-être expliqué par la raison ci-dessus, et sans doute par le fait que nous préférons avoir plus de détails même si nous savons qu'ils ne représentent rien.
Le problème est que ce type d'aliasage visuellement acceptable sur une image fixe ne l'est plus dans une animation, car il s'apparente à un bruit incontrôlé.
La solution optimale, implémentable par exemple dans une approche non temps réel, serait de mailler en fonction de l'échelle des primitives et de filtrer ensuite pour l'aliasage à l'échelle des pixels (limite inférieure de la taille du maillage).

Une autre conséquence intéressante du filtrage est qu'il permet de ne calculer que les primitives qui vont réellement être vues en fonction de l'échelle (susceptible de varier) à laquelle on visualise le phénomène. C'est en quelque sorte une gestion du niveau de détail presque continue, car les primitives ne sont pas générés à la volée[4], elle correspondent à un phénomène que l'on veut représenter.

3.3 Adaptation et temps réel

Finalement, nous allons étudier l'implémentation du modèle présenté dans le cadre d'une application temps réel.
Dans un premier temps, rappelons que la constance de la taille du maillage nous garantit une complexité constante pour le rendu (pris en charge par les cartes). Dans le cas d'un trop grand nombre de points, il se peut que le bus des PC soit un facteur limitant en qualité. Cela dépend essentiellement du matériel, mais les essais que nous avons réalisé sur un PC équipé d'une pentium III à 800 MHz et d'une carte Nvidia ont montré que les calculs étaient l'aspect le plus pénalisant pour la rapidité de l'application. Autrement dit, quelle que soit la densité d'échantillonnage choisie, le processeur n'est pas capable de calculer suffisamment de primitives pour donner l'aspect idéal. Évidement, cela dépend essentiellement du phénomène, mais notons que l'augmentation de complexité du calcul pour chaque point supplémentaire est multipliée par le nombre de primitives (moyennant filtrage). Comme il s'agit d'une surface, la densité augmente avec le carré de la résolution, on se retrouve donc logiquement rapidement limité par le calcul.

Nous allons donc rapidement étudier la gestion des primitives pour s'assurer de l'optimalité du calcul.
Notre approche a été de gérer les primitives de façon statique (pas de génération et destruction en temps réel) en limitant le nombre total de primitives, c'est à dire au moment de la génération. Comme le nombre de ces dernières peut décroître en fonction du filtrage, il apparaît dans l'application des variations de fluidité (gain car on supprime des calculs par rapport au pire des cas), cela n'est pas une solution optimale car il en découle une perte de ressource (que l'application aurait put utiliser).

3.3.1 Génération des primitives pour la mer

Nous avons vu dans l'état de l'art qu'il existait des bases utilisables pour la génération des primitives de vagues. Le mieux est bien sûr d'utiliser des approches stochastiques, car cela augmente en général nettement le naturel des scènes à modéliser. Rappelons qu'il existe le filtre de Pierson-Moskowitz pour créer une mer relativement calme loin du fetch, et que les études statistiques constituent une bonne base à exploiter pour modéliser les divers phénomènes souhaités.
Pour donner une idée précise de la diversité des configurations de vagues, rappelons simplement que le fait de se trouver hors du fetch engendre un certain nombre de situations différentes.
Par exemple, il se peut qu'il y ait composition entre des vagues créées très loin et des toutes petites vagues créées localement par un petit vent ; on a alors un spectre à deux dominantes séparés. Ce phénomène est relativement courant bien que les différentes dominantes puissent être plus ou moins proches.
A partir de cela, nous pouvons aisément conclure que quelle que soit la technique de génération choisie, elle se rapporte à la spécification d'un spectre donnant les caractéristiques des ondes à générer. En fait, cela correspond à un filtre qu'on appliquerait en première approche à une disrtibution continue, mais on pourrait imaginer d'appliquer ce filtre à un spectre plus représentatif de certains phénomènes. Mais cela constitue une étude sans doute beaucoup plus complexe qu'il n'y parait.
Enfin, il existe une autre technique, qui consiste à créer le spectre «à la main» de manière totalement empirique. Cette méthode fonctionne relativement bien, car l'allure des filtres générés avec des techniques statistiques est finalement assez simple. Comme toujours, l'idéal est probablement de mélanger toutes ces techniques pour obtenir une simplicité (méthodes stochastiques) et un contrôle (méthode empirique) optimal.

[4] On pourrait imaginer de générer uniquement les primitives que l'on veut afficher. cf 3.3.1.

Contentons nous ici et considérer que nous avons un spectre (ou un filtre, ce qui revient au même) décrivant le phénomène à modéliser. Nous savons d'autre part que nous voulons générer un certain nombre de primitives. Il suffit alors de discrétiser le spectre en autant de primitives que nécessaires. Mais si nous générons automatiquement les primitives, c'est pour nous permettre d'en adapter le nombre à la puissance de la machine, donc les spécifications du phénomène doivent être dans le cadre le plus général. Dès lors, si l'on discrétise tout le spectre, il se peut qu'il y ait génération de primitives qui ne seront jamais affichées car la densité de maillage choisie ne le permet pas. Nous devons donc prendre ce paramètre en compte dans la génération pour calculer la meilleure répartition possible des primitives.
Remarquons tout de même que le choix des paramètres n'est pas trivial, car il s'agit de déterminer le maillage optimal avec le plus grand nombre de primitives. Pour déterminer les paramètres, il faut rajouter une information comme, par exemple, un nombre idéal de primitives par polygone.

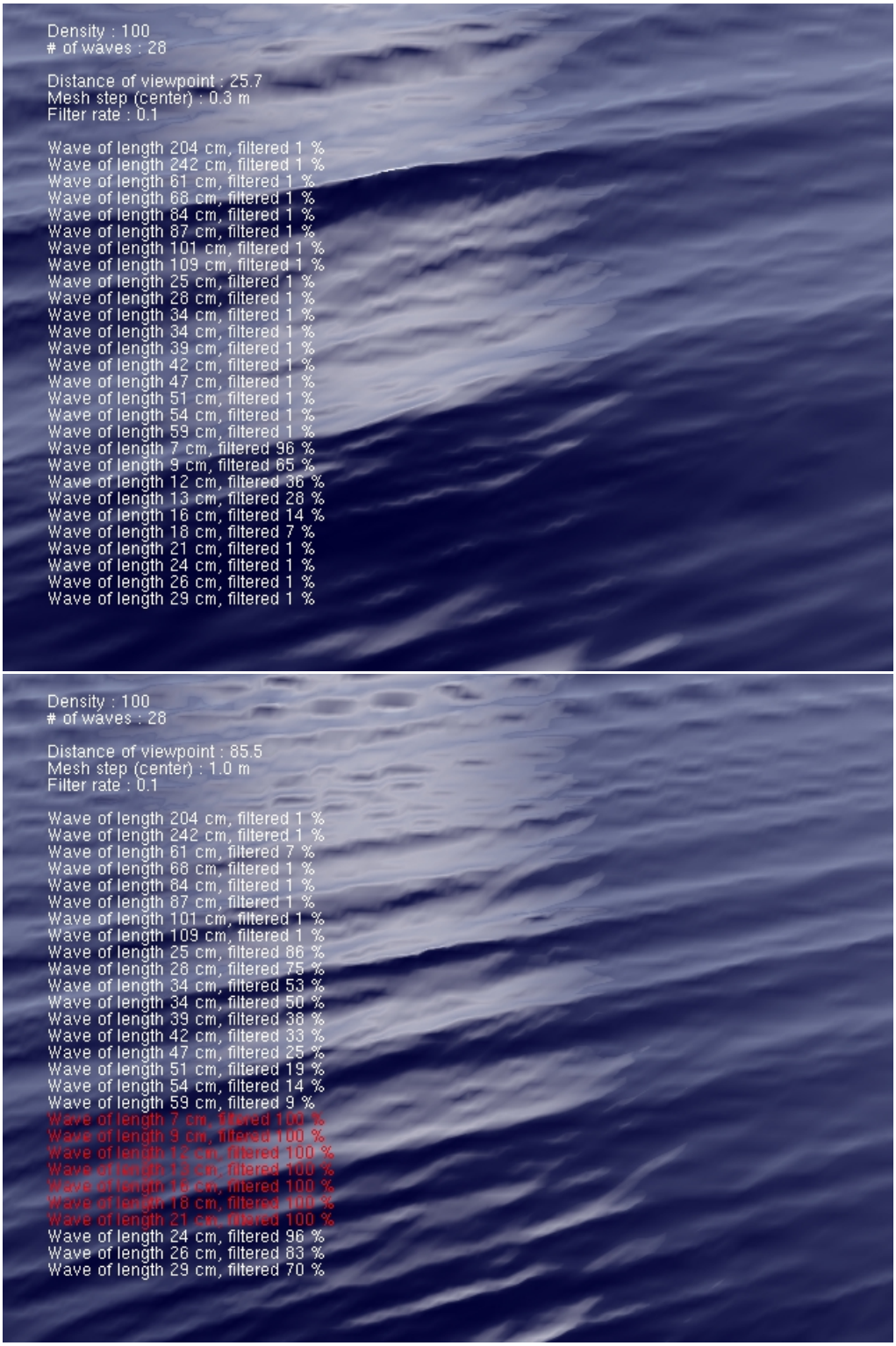


FIG. 3.9 – Filtrage à deux distances différentes : les primitives en rouge sont complètement filtrés (i.e. jamais affichées)

Chapitre 4

Application et résultats

L'application du modèle qui a été présenté a pour cadre essentiel le temps réel, que ce soit pour les jeux vidéos ou tout autre domaine de visualisation interactive comme les simulateurs de navigation. Néanmoins, l'architecture en primitives d'animation et effecteurs pour le rendu est aisément orientable vers d'autres applications, par exemple des applications en rendu non temps réel en utilisant des moteurs de rendu plus efficaces (comme RenderMan).

4.1 Architecture du modèle

Dans le cadre temps réel, nous avons mis en place une architecture pensée de manière à pouvoir être adaptée le plus facilement et le plus efficacement possible à une application de type jeu vidéo.
Cette architecture implémente le modèle primitive/effecteur ainsi que le modèle d'adaptation de géométrie décrit plus haut.
L'intérêt de cette modélisation modulaire est de permettre de modifier aisément et indépendamment les différentes parties du modèle.

Bien que basé sur une philosophie objet (fig. 4.1), seule l'implémentation des primitives se prête pleinement à cette optique, les parties géométrie et rendu ne nécessitant pas à priori d'héritages ou d'instanciation multiples.

La synchronisation avec le temps réel est effectuée par l'application selon une architecture classique : la mise à jour de la vue est effectuée lors de la latence de l'application (idle) et le temps depensé en calcul entre deux images est chronométré et écrit dans un compteur propre à l'objet de gestion de la géométrie. Ce compteur contient donc en permanence le temps *réel*. De la même façon, les mouvements de la caméra mettent à jour les informations nécessaires au calcul de la géométrie. Parallèlement, la même méthode est appliquée pour le contrôle des primitives.
Pour obtenir un résultat satisfaisant, nous avons fait attention à n'avoir aucun contrôle actif [1] au niveau du rendu lui même ainsi qu'à celui des primitives. En fait, des calculs sont uniquement effectués lorsque l'application commande un rendu.

[1] Qui provoque un calcul par opposition à «passif» : qui se contente de mettre à jour des paramètres.

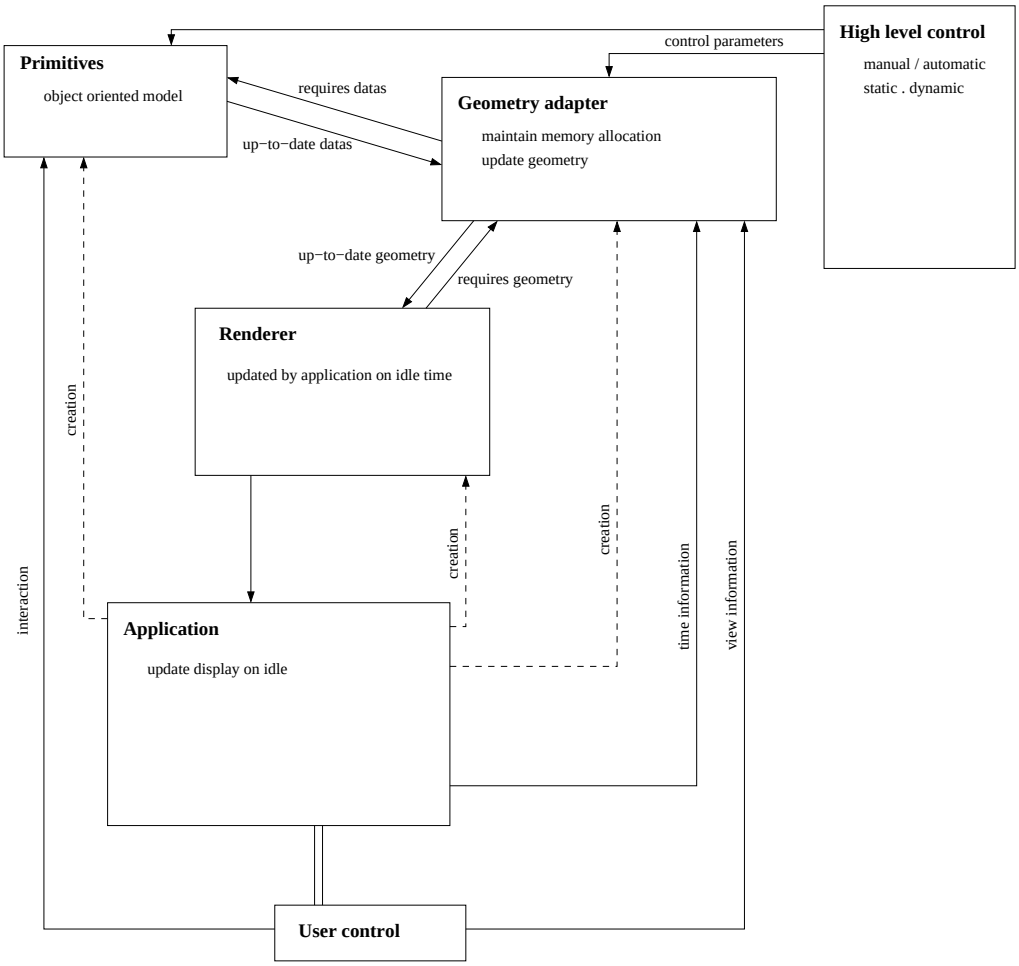


FIG. 4.1 – Architecture du modèle

4.2 Résultats : la mer

En pratique, pour les faibles résolutions, il est utile de laisser un peu d'aliasage au premier plan : cela rajoute un semblant de détails. Mais le filtre étant le même à toutes les échelles, il apparaît de l'aliasage non désiré à l'arrière plan comme on peut le voir sur les images suivantes (figures 4.2 et 4.3).

4.2.1 Paramétrage de l'application

Selon la nature de l'application, il y a différents paramètres réglables en fonction du contexte.
Tout d'abord, on peut régler la densité du maillage. Pour le temps réel nous avons essentiellement testé des maillages carrés de 100 ou de 50 points de coté.
On obtient avec un nombre de primitives comparable :

- Pour les maillages 100 points, le résultat obtenu se situe entre 4 et 10 images par seconde.

- Pour les maillages de 50 points, on obtient un résultat entre 20 et 30 images par seconde.

Le coût du calcul évoluant avec la surface du maillage, il est intéressant de noter que l'on peut gagner beaucoup de ressources en augmentant légèrement la taille des polygones (par exemple en passant de 50 points à 40 points de coté).

L'autre principal paramètre à régler est le nombre de primitives à utiliser. En effet, celles-ci vont être calculées pour chaque point du maillage, ce réglage agit uniquement sur le temps de calcul disponible sur le processeur. L'intérêt d'utiliser le plus de primitives possible est d'éviter le plus possible l'apparition de motifs.
Pour une certaine marge de longueurs d'onde proches, il faut compter au minimum 5 primitives pour ne pas avoir de motifs trop prononcés, 10 est une bonne moyenne. Mais on peut moduler ce choix en fonction de l'importance visuelle de la longueur d'onde.

Enfin, le dernier paramètre réglable est la sensibilité du filtrage. En théorie, l'aliasage se produit en deçà de quatre polygones par période, mais en pratique on peut afficher jusqu'à 4 ou 5 périodes dans un polygone sans ressentir trop l'aliasage (c'est un cas limite qui produit un mauvais résultat à l'horizon). Le réglage du filtre est très important car il constitue une bonne façon de gagner des détails : Nous avons obtenu un résultat équivalent à 500 et 300 points de côté, seulement en modifiant le filtrage (il s'agit d'un cas extrême, le filtrage étant idéal à 500, c'est à dire vers 4 et peu sensible à 300, vers 0.5). Mais le problème est que les détails gagnés grâce au phénomène d'aliasage s'apparentent à du bruit lors de l'animation. Pour les résultats temps réel (50 à 100 points de côtê), les réglages sont aux alentours de 0.3 polygones par période et 0.5 polygones par période. Les valeurs peuvent être baissés si la vue n'inclue par l'horizon. Pour les rendus non temps réel, on fixe les valeurs beaucoup plus hautes, quitte à augmenter la résolution, de façon a bien rendre l'horizon (car c'est là que l'aliasage est le plus sensible). Une bonne valeur se situe entre 1 et 4 polygones par période. L'expérience montre qu'au delà, l'effet est peu sensible.

Quels que soient les buts recherchés, le réglage des paramètres est plutôt difficile à réaliser. En réalité, la marge dans laquelle on obtient des résultats visuellement intéressant est faible et la variation du coût de calcul est forte. En d'autre termes, il est relativement aisé d'avoir un résultat de bonne qualité aux alentours de 10 images par seconde mais pratiquement impossible d'avoir un résultat même médiocre à 40 images par seconde.

Remarque

Les temps indiqués sont indicatifs car il reste encore de nombreuses optimisations à réaliser. Par exemple, il semble que la marge des 10000 points commence à saturer le bus AGP de la sortie graphique (Nvidia Quadro Pro) utilisée pour les essais. Le maillage est actuellement passé par quad-strips, ce qui permet de réduire le nombre de points envoyés dans le bus par deux par rapport à des triangle-strip et par quatre par rapport à des polygones non connectés (il y a redondance des sommet car nous avons un maillage). Mais il existe encore la possibilité d'utiliser des vertex arrays implémentés dans les cartes Nvidia qui permettrait de ne passer chaque point qu'une seule fois (contre quatre au pire des cas).

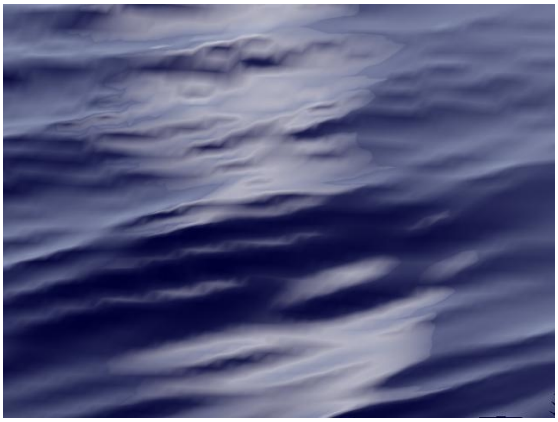


FIG. 4.2 – Résolution de 2500 points (20 images par seconde en 40x40 et 15 images par seconde en 50x50)

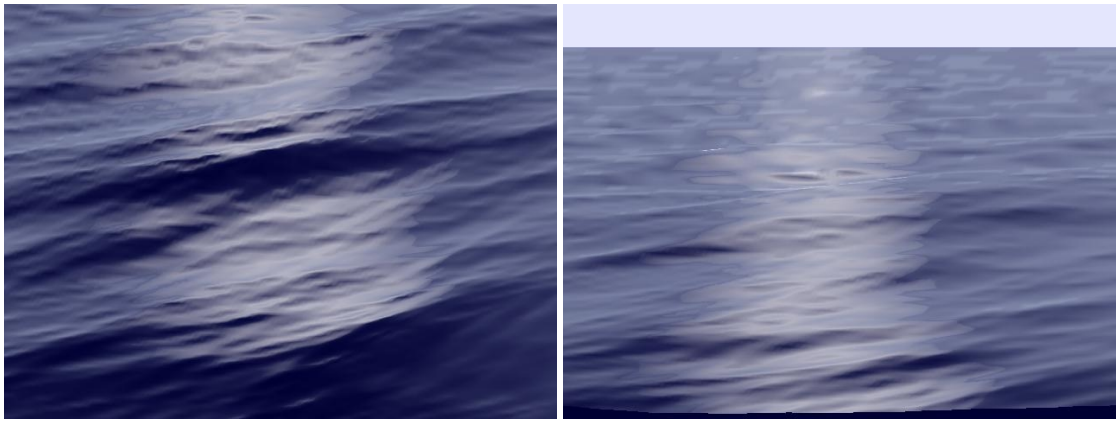


FIG. 4.3 – Résolution de 10000 points, 30 primitives (3-5 images par seconde, jusqu'à 10 selon la vue)

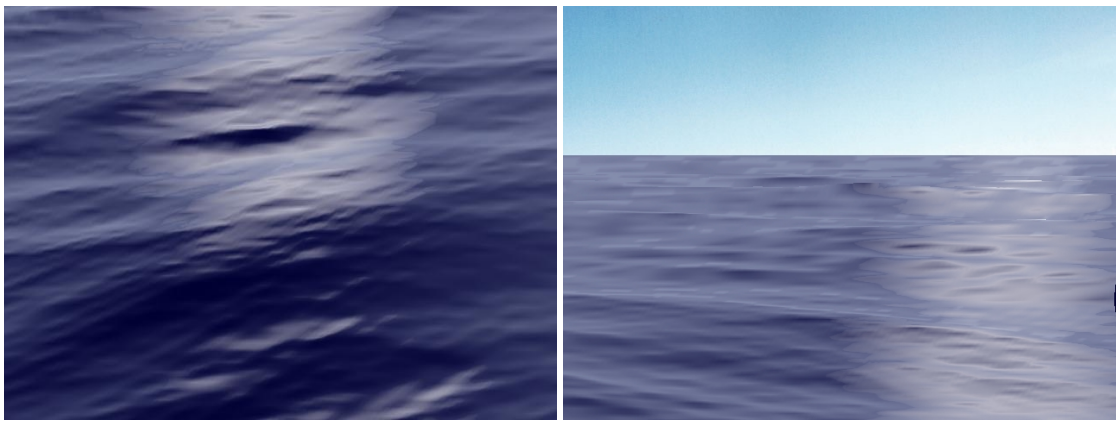


FIG. 4.4 – Résolution de 10000 points, 40 primitives (2-3 images par seconde)

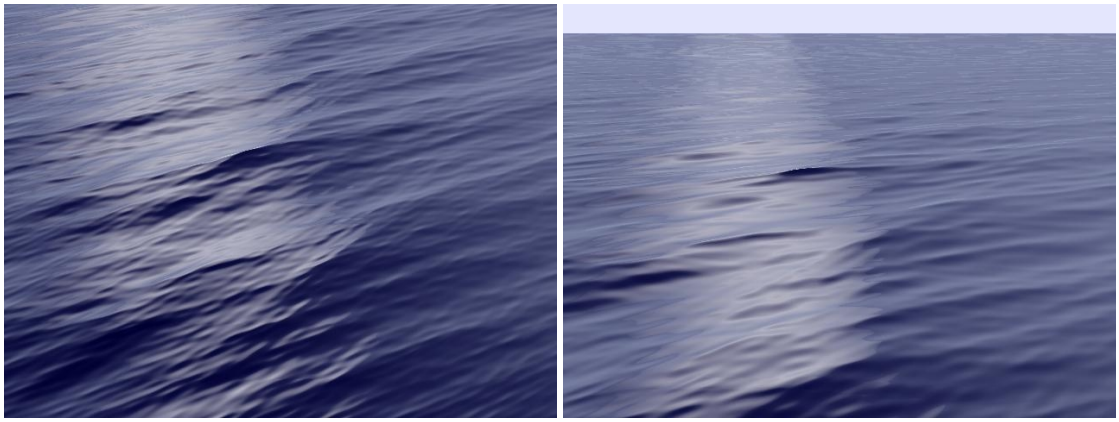


FIG. 4.5 – 250000 points, 60 primitives (10s)

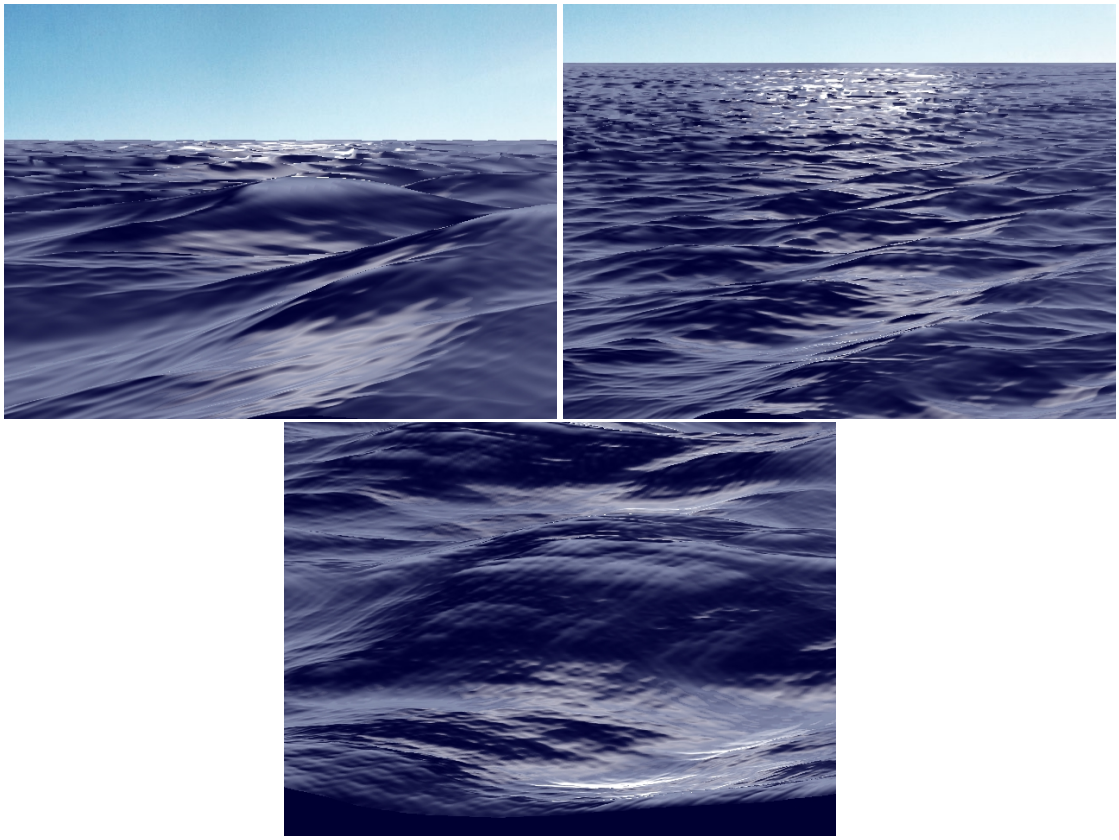


FIG. 4.6 – Mer agitée

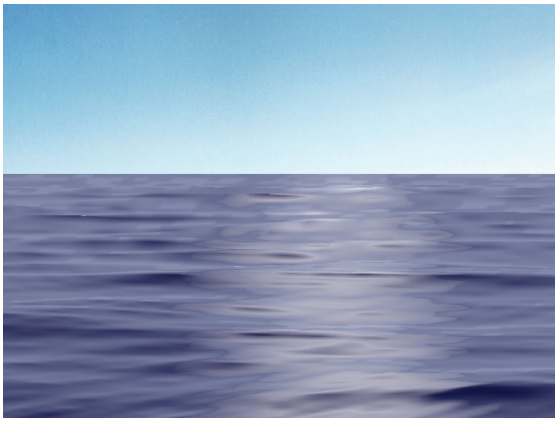


FIG. 4.7 – Mer calme, temps réel

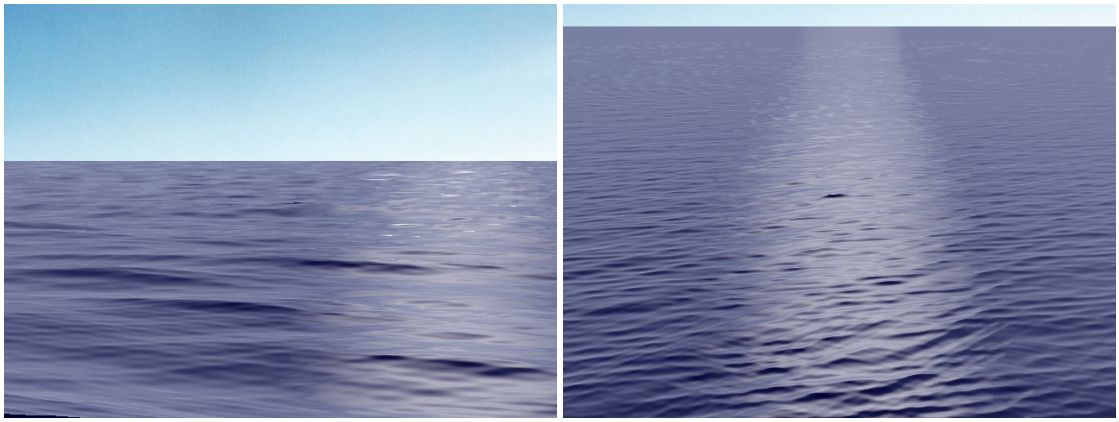


FIG. 4.8 – Mer calme

4.3 Un autre type de primitive : les sillages

Les sillages constituent une primitive incontournable pour une application basée sur la visualisation de la mer. Nous en avons implémenté une approche pour montrer la composition de primitives de mouvement de différente nature.

Modélisation des sillages

Les sillages sont des phénomènes relativement complexes. Il s'agit de la composition d'ondes gravitationelles ou capillaires. Ce sont des ondes de choc produites par le déplacement d'un objet dans le milieu liquide (ou inversement dans le cas d'un cailloux dans une rivière). On montre que les ondes des sillages sont stationaires par rapport au phénomène qui les engendre, ce qui va, entre autres, permettre de précalculer un motif. Introduisons brièvement le phénomène : le déplacement d'un objet dans l'eau produit, autour de lui, des ondes dont la vitesse est celle de l'objet dans le sens du déplacement et qui décroit jusqu'à 0 avec le cosinus de l'angle entre la trajectoire et la direction dans laquelle elles sont émise. On montre que le sillage résultant d'une onde de choc est contenu dans un cone d'angle constant.
En considérant un sillage rectiligne, on peut calculer, pour chaque point de l'espace (atteind par le sillage) quelles ondes sont en interférence à un instant donné. Dans le cas général, ces ondes sont au nombre de

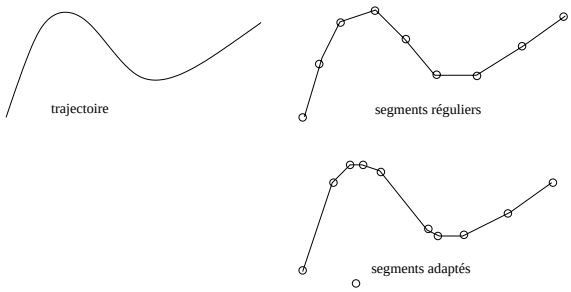


FIG. 4.9 – Trajectoire des sillages

deux. Il suffit alors de résoudre une équation du second degré pour déterminer la position des centres et ainsi le mouvement d'oscillation en ce point.
Les deux ondes composés sont de nature différente. La première, d'une longueur d'onde relativement grande (de l'ordre du mètre mais cela dépend de la vitesse) est perpendiculaire à la trajectoire de l'objet. La deuxième a une longueur d'onde beaucoup plus petite et forme la bordure du cone du sillage. C'est celle que l'on a l'habitude d'observer.
Dans la majorité des cas l'objet qui forme le sillage (surtout les bateaux), masque une partie des longueurs d'ondes générés en fonction de sa taille. C'est une effet de filtrage qui entraîne que l'on voit rarement l'ensemble du phénomène décrit ci-dessus.

Pour déterminer les ondes engendrées par les sillages, nous avons developpé une méthode permettant de calculer le champ de hauteur engendré en tout point selon une trajectoire rectiligne d'après la méthode exposée.
Pour obtenir des trajectoires courbes, il suffit alors de composer des segments de sillages rectilignes. Même dans le cas du précalcul d'un motif, cette modélisation permet de composer les ondes produites par des sections différentes entre elles (car elles sont composés additivement). Notons qu'une section n'a aucune raison d'interférer avec elle-même puisqu'elle est droite. Le nombre de sections à utiliser dépend de la courbure de la trajectoire. Pour obtenir un sillage optimal, il suffit de discrétiser la trajectoire avec un pas variable en fonction de la courbure (fig. 4.3).

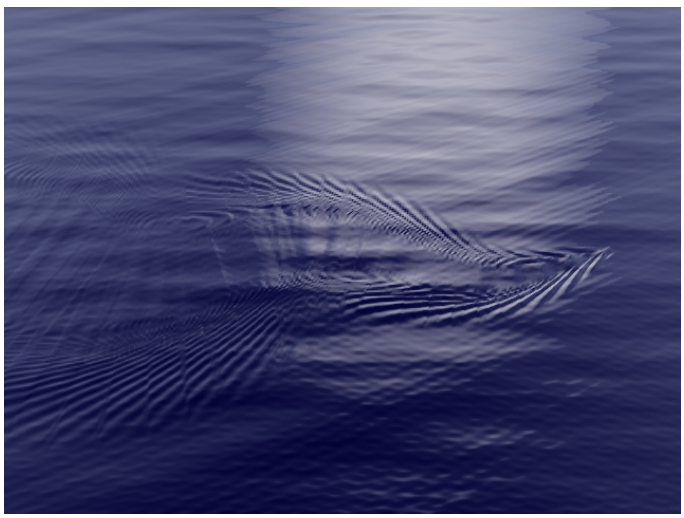


FIG. 4.10 – Sillage

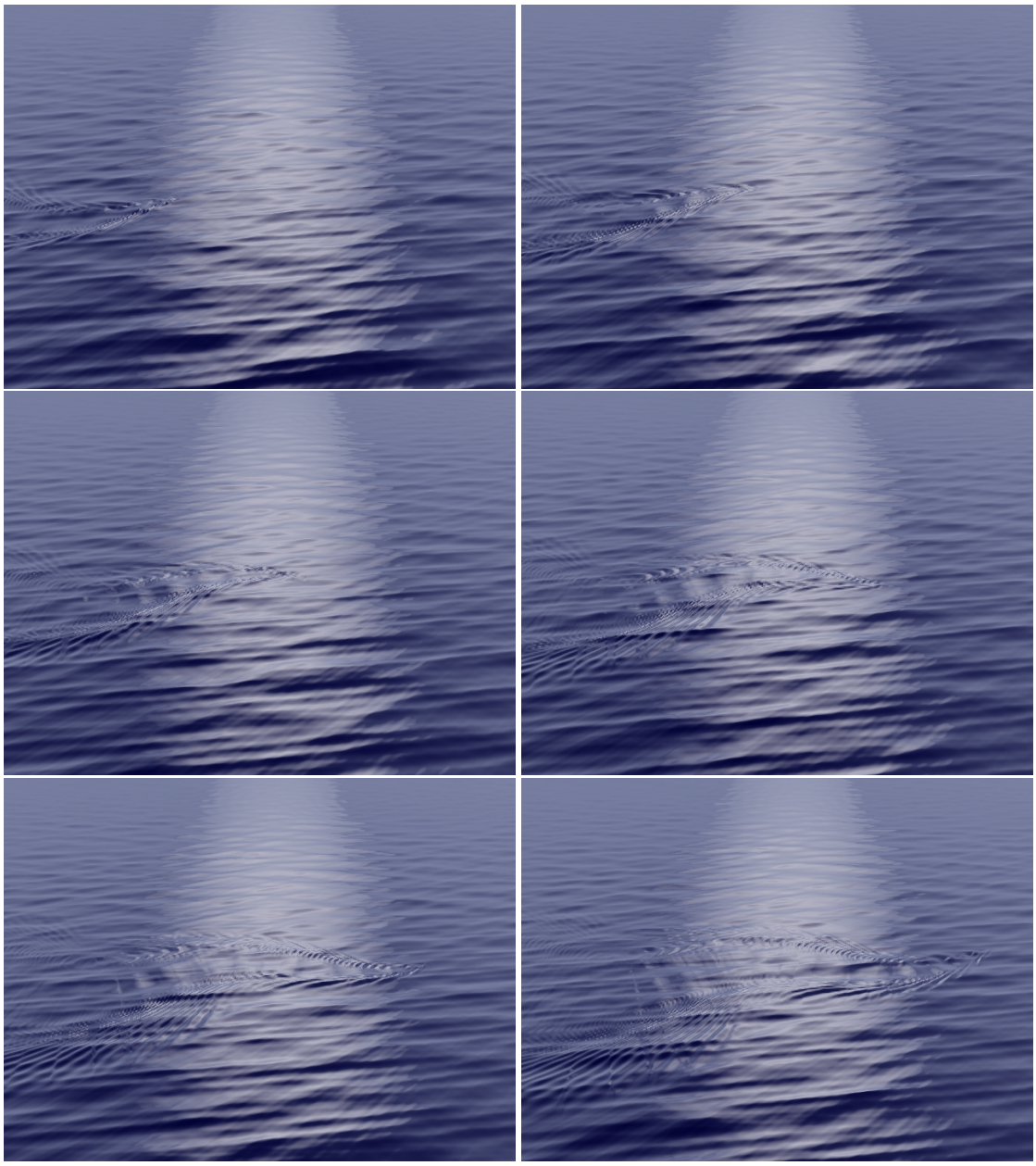


FIG. 4.11 – Sillage animé

Conclusion et perspectives

Nous avons, au cours de ce projet, mis au point une méthode de modélisation des surfaces océaniques en temps réel. Nous avons, pour cela, implémenté une formalisation, à base de primitives et d'effecteurs, permettant une certaine souplesse à l'implémentation de niveaux de détails. L'approche que nous avons utilisé est de type paramétrique, ce qui confère à notre modèle une certaine stabilité et nous permet de générer un maillage à la volée afin de l'adapter au mieux au contexte.

Il reste néanmoins énormément de travail à réaliser, et les directions de recherche sont nombreuses et diverses. Le point essentiel que nous n'avons pas abordé est l'introduction d'une composante dynamique dans le modèle. Les primitives sont pensées pour être contrôlés. On pourrait imaginer un système dynamique qui règle lest paramètres des primitives. L'effet produit serait, d'abord, une meilleure interaction du phénomène avec le milieu (digues ou bateaux), et ensuite une meilleure gestion de l'animation.Une approche serait de pré-calculer dynamiquement les réflexions et peut-être les diffractions des ondes sur la côte et les autres objets susceptibles d'interagir avec l'écoulement.

Mais l'implémentation a montré que notre approche fournissait des résultats satisfaisants dans un cadre plus général que le temps réel puisque les modélisations et les traitement de l'aliasage permettent également d'augmenter la qualité visuelle dans les cas plus complexe. La recherche dans ce domaine diverge quelque peu des axes principaux du temps réel mais il serait intéressant de finaliser, dans ce cadre, les techniques non temps réel introduites ici et de fournir au modèle une méthode de rendu plus adaptée en s'affranchissant des contraintes d'implémentation propre au temps réel.

Enfin, un axe de recherche, à mon sens essentiel, est celui de l'architecture de primitives et effecteurs qui constituent le squelette de notre modélisation, et tient le premier rôle dans notre adaptation de maillage en temps réel.

Bibliographie

[1] P. Chassaing. *Mécanique des fluides. Eléments d'un premier parcours*. Cepadues éditions, 1997.

[2] R. Feynman. *Lectures on physics*. Addison-Weisley Publishing Compagny, 1977.

[3] Alain Fournier and William T. Reeves. A simple model of ocean waves. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 75–84, August 1986.

[4] B. Gates. An overview of modeling liquids for computer graphics. In *Siggraph Course Notes*. Addison-Wesley, 1994.

[5] J.-C. Gonzato and B. L. Saec. On modeling and rendering ocean scenes (diffraction, surface tracking and illumination). In *WSCG '99 (Seventh International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media)*, pages 93–101, Plzen-Borey, Czech Republic, February 1999. University of West Bohemia.

[6] Jean-Christophe Gonzato and Bertrand Le Saëc. A phenomenological model of coastal scenes based on physical considerations. In D. Thalmann and M. van de Panne, editors, *Computer Animation and Simulation '97*, Eurographics, pages 137–148. Springer-Verlag Wien New York, 1997.

[7] Jean-Christophe Gonzato and Bertrand Le Saëc. On modelling and rendering ocean scenes. *The Journal of Visualization and Computer Animation*, 11(1) :27–37, 2000.

[8] Michael Kass and Gavin Miller. Rapid, stable fluid dynamics for computer graphics. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 49–57, August 1990.

[9] B. Kinsman. *Wind Waves, Their Generation and Propagation on the Ocean Surface*. Dover Publication, 1984.

[10] A. Lencastre. *Manuel d'hydraulique générale*. Eyrolles, 1966.

[11] J. Lighthill. *Waves in fluids*. Cambridge University Press, 1978.

[12] J. Lighthill. *An informal introduction to the theoretical fluid mecanics*. Oxford University Press, 1986.

[13] G. A. Mastin, P. A. Watterberg, and J. F. Mareda. Fourier synthesis of ocean scenes. *IEEE Computer Graphics and Applications*, 7(3) :16–23, March 1987.

[14] Darwyn R. Peachey. Modeling waves and surf. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 65–74, August 1986.

[15] Frank Perbet and Marie-Paule Cani. Animating prairies in real-time. In *ACM Interactive 3D Graphics*, USA, Mar 2001.

[16] Ken Perlin. An image synthesizer. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19(3), pages 287–296, July 1985.

[17] Areté software. `http ://www.areteis.com`.

[18] Jos Stam. Stochastic dynamics : Simulating the effects of turbulence on flexible structures. *Computer Graphics Forum*, 16(3) :159–164, August 1997. Proceedings of Eurographics '97.

[19] J. Tessendorf. Simulating ocean water. In *Siggraph Course Notes*. Addison-Wesley, 1999.

[20] Sebastien Thon, Jean-Michel Dischler, and Djamchid Ghazanfarpour. Ocean waves synthesis using a spectrum-based turbulence function. In *Computer Graphics International Proceeding*, 2000.

[21] Sebastien Thon, Jean-Michel Dischler, and Djamchid Ghazanfarpour. A simple model for visually realistic running waters. In *Eurographics UK*, 2000.

[22] Sebastien Thon and Djamchid Ghazanfarpour. A semi-physical model of running waters. In *Eurographics UK*, 2001.

[23] Pauline Y. Ts'o and Brian A. Barsky. Modeling and rendering waves : Wave-tracing using beta-splines and reflective and refractive texture mapping. *ACM Transactions on Graphics*, 6(3) :191–214, July 1987.