

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par **Jean-Christophe Gonzato**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Modélisation des scènes océaniques

Soutenu le : 15 Décembre 1999

Après avis des rapporteurs :

Alain Fournier rapporteur
Patrick Callet rapporteur

Devant la commission d'examen composée de :

Christophe Schlick (Université Bordeaux I)	Président
Pascal Guitton (Université Bordeaux I)	Rapporteur
Patrick Callet (Ecole Centrale Paris)	Examineur
Alain Fournier (University of British Columbia)	Examineur
Bertrand Le Saëc (Université Bordeaux I)	Examineur
Michel Mériaux (Université de Poitiers)	Examineur

A Patricia et Thibault,

Remerciements

Mes plus vifs remerciements vont à :

- Christophe Schlick, pour avoir accepté d’être le président de mon jury. J’admire l’immensité de ses connaissances, sa grande disponibilité et ses conseils avisés.
- Bertrand Le Saëc, que j’appelle “mon chef”, qui a bien voulu patienter le temps de mon service militaire pour m’encadrer et qui m’a soutenu pendant plus de trois années. J’admire la qualité de ses intuitions, le nombre important de ses idées pour de nouvelles voies de recherches et son ouverture d’esprit. D’un côté plus personnel, Il m’aura permis de découvrir le monde des cichlidées et sa passion pour l’aquariophilie.
- Pascal Guitton, autrement dit “le grand chef”, qui encadre notre équipe avec dextérité. Il fait part d’une gentillesse à toute épreuve. Il a toujours été à mon écoute lorsque je lui demandais conseils, argent, . . .
- Alain Fournier et Patrick Callet, pour avoir accepté de se consacrer à la correction de mon mémoire de thèse, et pour leurs remarques pertinentes, leur gentillesse et leur bonne humeur permanente.
- Michel Mériaux, pour l’honneur qu’il m’a fait de participer à mon jury ainsi que pour sa disponibilité et sa gentillesse.

Merci aussi, dans un ordre tout à fait arbitraire, aux personnes suivantes :

- Benoît (dit “benito”), Laurent (dit “lolo”) et Guillaume (dit “celui qui est toujours en vacances à l’étranger”), pour l’ensemble des remarques pertinentes qu’ils ont su me faire. A ce jour, ils attendent toujours leur bouteille de champagne.
- Dans le désordre et milles excuses aux personnes que j’aurai oubliés : Stefka, Jean-françois, Hélène (dit “la p’tiote”), Hélène (dit “Hachmoffouane”), Anne, Olivier, Sébastien, Jean-Marc, Alexiane, Sébastien, Denis, Xavier, Eric, Céline (amenant un peu d’air pur de Grenoble).
- Enfin, une mention spéciale sera attribué à mes meilleurs amis David et Magali, ainsi qu’à ma femme Patricia pour le temps qu’ils ont passé à relire le texte, à émettre toutes sortes de commentaires plus ou moins provoquants et, enfin, à essayer de me faire articuler un peu mieux.

Table des matières

Introduction	1
1 Modélisation de la houle	5
1.1 Introduction	5
1.2 Résolution des équations de mécanique des fluides	6
1.2.1 Théories physiques	6
1.2.1.1 Les équations de Navier-Stokes	6
1.2.1.2 L'équation de continuité du fluide	7
1.2.2 Modélisation informatique	7
1.2.2.1 Méthode de résolution M.A.C.	7
1.2.2.2 Résolution 2D des équations de mécanique des fluides	10
1.2.3 Bilan	11
1.3 Modélisation paramétrique de la houle	12
1.3.1 Théories physiques	12
1.3.1.1 La houle de Gerstner	14
1.3.1.2 La houle irrotationnelle de Stokes	17
1.3.1.3 La houle de Biesel	21
1.3.1.4 Etude au cas limite : le déferlement	23
1.3.2 Modélisation infographique	26
1.3.2.1 Le modèle de Fournier-Reeves	28
1.3.3 Notre modèle de houle	30
1.3.3.1 Quelques modifications du modèle de Fournier-Reeves	30
1.3.3.2 La fonction d'étirement de la vague : <i>Etirement</i>	31
1.3.3.3 Les effets de la gravité : <i>Chute & Orientation</i>	32
1.3.3.4 Les pentes négatives	33
1.3.3.5 Les équations de notre modèle de houle	34
1.3.4 Bilan	34

2	Influence du sol et de la côte	37
2.1	Introduction	37
2.2	La réfraction des vagues	38
2.2.1	Modélisation physique	38
2.2.1.1	Tracé des plans de vagues	39
2.2.1.2	Tracé des orthogonales au front de vagues	40
2.2.2	Réfraction et infographie	41
2.2.3	Notre modèle de Réfraction : Le Lancer de Vagues Adaptatif	43
2.2.3.1	Principes	43
2.2.3.2	Le calcul de la profondeur en tout point du fond marin	44
2.2.3.3	Calcul de la déviation	45
2.2.3.4	Principe général de la progression des rayons	46
2.2.3.5	La calcul de l'amplitude	49
2.2.4	Résultats	50
2.3	La Diffraction des vagues	51
2.3.1	Un modèle physique	51
2.3.2	Notre modèle et son intégration dans le Lancer de Vagues	54
2.3.2.1	Définition d'une digue	55
2.3.2.2	Calcul de la diffraction	56
2.3.2.3	Diffraction et Réfraction simultanément	57
2.3.3	Résultats	58
2.4	La réflexion des vagues	59
2.4.1	Un peu de physique	59
2.4.2	Notre modèle	60
2.4.2.1	Réflexion sur le musoir	60
2.4.3	Résultats	61
2.5	Facettisation et Multi-trains de vagues	62
2.6	Récapitulatif sur le Lancer Dynamique de Vague complet	63
3	Faire la lumière sur l'océan	67
3.1	Travaux précédents	67
3.2	Identification de la surface	70
3.3	Amélioration de la géométrie : les vagues capillaires	73
3.4	Modèle de rendu	76
3.4.1	Couleur des océans	76
3.4.2	Calcul des coefficients de réflexion et de transmission	78

3.4.3	Atténuation de la lumière	79
3.4.4	Le modèle de couleur utilisé	80
3.5	Résultats	82
4	Le lancer de faces	87
4.1	Introduction	87
4.2	Travaux précédents	90
4.3	Lancer de rayons de Monte Carlo	91
4.4	Le Lancer de Faces	92
4.4.1	Description générale de l’algorithme	92
4.4.2	La scène	93
4.4.3	Pas de progression des faces	94
4.4.4	Subdivision d’une face	94
4.4.5	Intersections avec des objets	96
4.4.6	Le calcul des composantes chromatiques d’un pixel	98
4.4.7	Gestion macroscopique des calculs	100
4.4.8	Points forts de la méthode	100
4.5	Discussion	102
4.5.1	Relation entre niveau de subdivision (LF) et niveau de raffinement (LRMC)	103
4.5.2	Résultats du comparatif	105
4.6	Application à la simulation des mirages	106
4.6.1	Précédents travaux	107
4.6.2	Définition des milieux réfringants	108
4.6.3	Le milieu réfringent et le Lancer de Faces	110
4.6.4	Résultats	110
4.7	Conclusion	110
	Conclusion	113

Table des figures

1	Définition des vecteurs vitesse et de la pression dans un élément de la grille . . .	8
2	Quelques cas particuliers pour la méthode MAC	10
3	Résolution en 2D	11
4	Les différentes données sur les vagues	13
5	Vue générale de la houle de Gerstner	14
6	Niveau de la surface libre	15
7	Vue de la surface de la mer en fonction de K et R	16
8	Propagation des vagues	17
9	Mouvement d'un objet à la surface de l'océan	17
10	Houle de Stokes au premier ordre	19
11	Amplitude relative des vagues en fonction de la profondeur	22
12	Houle de Biesel en bord de rivage (d'après [Bie52])	23
13	Différents types de déferlements	24
14	Type de déferlement en fonction de la pente du fond marin	25
15	Détermination du point de déferlement	26
16	Illustration Japonaise	27
17	Houle et trajet des particules du modèle de Zang	27
18	Une erreur sur la troisième crête	30
19	Cas limite du modèle de Fournier et Reeves	30
20	Une limite de forme de vague dans notre modèle	31
21	a. Fonction Etirement, b. Vue de profil	32
22	a. Fonction Orientation, b. Fonction Displacement, c. Vue de profil	33
23	Vue de profil d'une onde avec une pente négative. La vague continue sa progression dans le sens du train d'onde	34
24	Un exemple final	34
25	Réfraction des vagues dans une baie	39
26	Tracé de front d'ondes par la méthode de Huyghens	39

27	Loi de la réfraction	40
28	Tracé des orthogonales au front de vagues (d'après [Lac65])	41
29	Calcul de l'amplitude en fonction de la réfraction	42
30	Calcul de la réfraction avec le modèle de Fournier-Reeves	42
31	Stockage d'une vague déferlente	43
32	Le "Wave-tracing" de Ts'o sur deux exemples : a. une île, b. une baie	43
33	Définition du sol par des splines (vue de dessus)	44
34	Interpolation Spline-Cardinale 2D	45
35	Interpolation Cardinale-Spline 3D	45
36	Calcul de la courbe de niveau (vue de dessus)	46
37	Méthode de calcul des facteurs de réfraction	47
38	Création de nouveaux rayons	48
39	Création de nouveaux rayons en bord de scène (vue de dessus)	49
40	Calcul de l'amplitude suite à la réfraction	49
41	Calcul de l'amplitude	50
42	Calcul de l'amplitude dans le cas d'un nouveau rayon	50
43	Notre algorithme : a. une île, b. une baie	51
44	Une vue de dessus calculée des mêmes scènes	51
45	Vue 3D facettisée d'une scène avec déferlement	52
46	Vue 3D d'une scène océane grâce à Open Inventor	52
47	Zone d'ombre derrière une digue	53
48	Agitation derrière une digue	53
49	Théorie de la diffraction	54
50	H_i / H_r en fonction de U_i / U_r	55
51	Gestion des musoirs	56
52	Notre méthode pour calculer la diffraction et la réfraction	57
53	a. Les rayons lors du lancer de vagues, b. Les vagues	58
54	Une vue 3D de la diffraction (l'obstacle est perpendiculaire au plan moyen de l'océan)	59
55	Calcul sur le musoir	61
56	Différentes zones de calculs	61
57	a. Une vue Inventor de la réflexion, b. Cas du musoir	62
58	Création des facettes de la scène	63
59	Gestion du voisinage des rayons dans la liste	64
60	Ensemble des rayons se propageant dans une scène	64

61	Approximation des coefficients de Fresnel par Ts'o et Barsky	68
62	Représentation de l'environnement de la scène	69
63	Cas particulier du déferlement	71
64	Recherche de la surface de l'océan	71
65	Différentes fonctions pour évaluer le pas de progression	72
66	Problèmes dans la recherche de la surface	72
67	Quatres niveaux de bruits fractals différents	73
68	Passage continu d'un niveau de bruit au niveau supérieur	74
69	Différentes configurations de surface avec des vagues capillaires	76
70	Les rayons lumineux dans l'océan	77
71	Illumination locale	78
72	Valeur de la réflectance en fonction de l'angle d'incidence du rayon	79
73	Evolution du spectre d'une lumière blanche en fonction de la profondeur	80
74	Le spectre d'absorption de l'eau de mer.	81
75	Modèles de couleur pour les objets sous l'eau	81
76	Transmission d'une lumière blanche jusqu'à 15 mètres	82
77	Transmission d'une lumière blanche jusqu'à 50 mètres	82
78	Multiplés trains de vagues	83
79	Visualisation de l'absorption de la lumière avec un objet plongé dans l'eau	83
80	Vue générale de l'océan de jour	84
81	Coucher de soleil sur l'océan	85
82	Une image "aliasée" et deux images "anti-aliasée" à des niveaux différents.	88
83	Méthode de subdivision des facettes	94
84	Pas de calcul	94
85	Progression avec une intersection	95
86	Un exemple de quadtree	96
87	Intersection avec un objet spéculaire	96
88	Subdivision après une intersection	97
89	Un cas limite	98
90	Rayons et angles solides	98
91	Subdivision hiérarchique d'un pixel	99
92	Calcul du poids d'une face	100
93	Génération d'une image	101
94	Gestion de l'aliasing de texture	101
95	Exemple d'objets semi-spéculaire et semi-transparent (tiré de [Arv95])	102

96	Rendu des objets semi-spéculaires	103
97	Une face réfléchie par un objet semi-spéculaire	103
98	Le graphe à trois niveaux de subdivisions	103
99	Une relation entre le nombre de faces et d'arêtes dans le graphe	104
100	Images résultats produites avec le lancer de faces	105
101	Un mirage inférieur	107
102	Un mirage supérieur	107
103	a. Mirage inférieur, b. Mirage supérieur, c. Fata Morgana	107
104	Parcours de la lumière dans le modèle de Berger	108
105	Deux méthodes pour définir un milieu d'indice continu	108
106	Calcul de la normale à la surface isotherme	109
107	Différents champs de températures	109
108	Un exemple de Fata Morgana	110
109	Mirage sous la tour Eiffel	111

Introduction

Le but de la synthèse d'images réalistes est d'obtenir une image informatique d'une scène tridimensionnelle imitant au mieux son image photographique. Pour atteindre un tel but, il faut connaître le processus complet de génération de l'image. Ceci passe de l'appareil de prise de vue (lentilles, temps de pose, ouverture, ...) au suivi du trajet de la lumière dans la scène virtuelle par une connaissance parfaite de la géométrie de chaque objet et de son interaction avec la lumière.

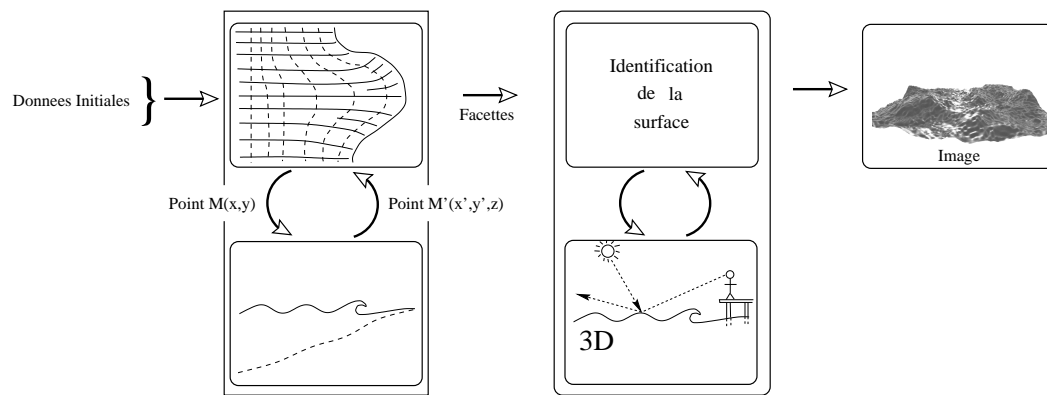
Historiquement, les scènes à visualiser étaient assez simples et composées d'un ensemble d'objets statiques dont la géométrie était une combinaison d'objets de bases (cube, sphère, ...). La synthèse d'images réalistes s'est d'abord intéressée à la qualité des images obtenues. Des modèles d'illumination complets ont donc été présentés. Il fallait connaître parfaitement les propriétés optiques des divers objets de la scène pour appliquer les équations de transport de la lumière. Ces équations étaient directement tirées de la physique et adaptées aux contraintes matérielles de l'époque.

De nos jours, ayant des modèles d'illumination suffisamment précis, une partie importante de l'infographie s'est tournée vers la modélisation des "objets dynamiques" dont la géométrie est soumise à des contraintes physiques. Il n'est alors plus possible de décrire de manière simple cette géométrie. Par exemple, on peut citer la modélisation des nuages, des mouvements des fluides, de la croissance des plantes, des flammes... Chacun de ces "objets" a des propriétés intrinsèques (viscosité pour un fluide, élasticité pour un objet mou, ...) et est soumis à des forces extérieures (vent, pesanteur, ...). Certains objets, tels que les fumées ou les cascades d'eau, devront être découpés en éléments simples pour déterminer leur géométrie globale. Cette partie de l'infographie se nomme *modélisation des phénomènes naturels*.

Si de loin, la physique et l'infographie, travaillent de pair sur les mêmes domaines, elles n'ont pas les mêmes regards sur les phénomènes à modéliser. En effet, d'une part, les physiciens ont pour but de connaître le phénomène dans son intégralité et de manière microscopique. Les lois complexes qui en découlent donneront des résultats précis au détriment des temps de calculs. D'autre part, l'infographie s'impose des contraintes sévères de temps de réponse. En effet, pour la génération d'une animation, il faudra tenir compte du nombre important de modélisation et de

calcul d'images à effectuer (au moins 25 par seconde d'animation). L'infographiste ne s'intéresse donc en général qu'au phénomène macroscopique. De plus, il n'est pas nécessaire d'appliquer des lois physiques complexes pour faire ressortir l'information pertinente. Les détails seront alors remplacés par des algorithmes purement infographiques. Par contre, il semble intéressant d'entrevoir le phénomène sous plusieurs échelles. Une portion de nuage n'est-elle pas un nuage ? Une petite vague n'est-elle pas aussi une vague ?

Nous nous sommes particulièrement intéressés dans cette thèse à la modélisation des scènes océanes. Notre objectif est de présenter une approche permettant, d'une part, de modéliser la géométrie de la surface de l'océan et d'autre part de l'illuminer. La géométrie de l'océan se décompose en deux parties : premièrement, une modélisation des modifications de la forme des vagues et de leurs caractéristiques en fonction du sol (*amplitude, longueur d'onde, déferlement*) est effectuée. Deuxièmement, les différents phénomènes influant sur la propagation des trains d'ondes (*réfraction, diffraction et réflexion*) sont modélisés. En ce qui concerne l'étape de rendu, un modèle d'identification de la surface et d'illumination est proposé.



La figure précédente propose un découpage modulaire des algorithmes nécessaires à l'obtention d'une image. Ils sont indépendants les uns des autres et interchangeable avec des modules du même type. Au fur et à mesure de la modélisation des trains d'ondes, un calcul de la forme de la vague est effectué. Une facettisation de la surface est alors produite et l'étape d'illumination permet de fournir l'image résultante.

Un deuxième objectif de ce travail est une généralisation des algorithmes obtenus en essayant de les appliquer à d'autres données. Nous présenterons une simulation des mirages par un lancer de rayons spécifique utilisant les méthodes déduites de la génération de la réfraction de la houle. Ce lancer de rayons échantillonnera la scène avec un nouveau type de stratification du pixel. Cette stratification sera basé sur des techniques dites de Monte Carlo.

Le chapitre 1 est consacré à la modélisation de la forme en coupe des vagues. Nous passons

tout d'abord en revue les divers modèles physiques. Deux grands types de méthodes sont employés pour permettre cette modélisation : le premier type est une résolution complète des équations de la mécanique des fluides (*Navier-Stokes*). Elles permettent de modéliser les mouvements de tous les types de fluides (liquide ou gaz). Leur aspect généraliste en font un modèle difficile à maîtriser. Le second type est une approximation de la propagation de la houle par des équations paramétriques (*Gerstner, Stokes, Biesel*). Nous détaillons par la suite les divers travaux effectués en infographie (*modèles de Fournier-Reeves [FR86], Peachey [Pea86], Ts'o-Barsky [TB87], etc*). Nous proposons ensuite notre modèle basé sur des équations paramétriques en adaptant les différentes théories physiques. Notre algorithme permet, de plus, d'améliorer la description des vagues déferlantes.

Dans le chapitre 2, nous nous intéressons aux phénomènes conditionnant la propagation des trains de vagues (*réfraction, diffraction et réflexion*). Suivant un même découpage que le précédent chapitre, nous étudions les divers résultats physiques et infographiques existants. Nous proposons un nouvel algorithme général appelé *Lancer de Vagues Dynamique* qui permet de modéliser tout type de surface où ces trois phénomènes interagissent tout en gardant une précision d'échantillonnage identique en tout point de la scène.

Les trains d'onde sont manipulés les uns après les autres. La méthode utilise les orthogonales aux fronts de vagues pour décrire le trajet des ondes dans la scène. La caractéristique principale du lancer de vagues est que si deux orthogonales successives s'écartent trop, alors une nouvelle est créée entre les deux. Ainsi, une même définition de la surface est préservée.

Le chapitre 3 aborde le rendu réaliste du modèle géométrique obtenu par la méthode vue dans les chapitres précédents. Le résultat de la modélisation ne fournit pas explicitement la surface de l'eau. En effet, elle se compose d'un ensemble de facettes indépendantes et superposées. Une première étape basée sur le *Displacement mapping* permet d'identifier la surface réelle de l'océan. Dans une deuxième étape, nous proposons un modèle d'illumination incluant le spectre d'absorption de l'eau de mer afin d'obtenir un rendu réaliste.

Finalement, dans le chapitre 4, nous proposons une adaptation 3D du *Lancer de Vagues Dynamique* au lancer de rayons appelé alors *Lancer de Faces*. En effet, les rayons divergent très vite en se propageant dans une scène. Ils peuvent suivre, de plus, des chemins très différents. Entre deux rayons voisins, une partie importante de la scène peut être omise et ainsi un aliasage géométrique est introduit dans l'image. Le Lancer de Faces décompose la pyramide de vue en faces triangulaires. Lorsque l'aire d'une face devient trop importante, elle est subdivisée pour garantir une bonne détection des objets de la scène. De plus, cette décomposition triangulaire permet de gérer

correctement l'aliasing de texture ainsi que les objets semi-spéculaires.

Cet algorithme peut être utilisé dans un grand nombre d'applications : les mirages, les gemmes, l'endoscopie virtuelle, etc. Dans ce rapport, nous appliquerons le lancer de faces à la génération des mirages.

Chapitre 1

Modélisation de la houle

1.1 Introduction

L'homme a toujours voulu dominer la nature et pour cela il l'a étudiée. Les mouvements des océans et de la houle sont une des manifestations complexes de la nature à laquelle s'intéressent depuis longtemps les physiciens. La maîtrise de la compréhension des phénomènes est nécessaire pour positionner et construire idéalement ports, digues, plate-formes pétrolières.

Une simple observation des vagues montre que la déformation de la surface de l'océan est de type ondulatoire. Par contre, il ne s'agit pas d'onde simple mais d'une combinaison de plusieurs ondes. De plus, les trains de vagues, venant du large, se déforment à l'approche des côtes en subissant manifestement l'influence du sol. Ils se brisent brutalement au bord des plages mais un tel déferlement peut être aussi observé au large bien qu'étant de nature différente. L'amplitude des vagues varie lors de leur propagation. Près des côtes, elle est beaucoup plus importante. Elle dépend aussi des conditions atmosphériques. Par vent faible, l'amplitude est moins importante que par vent fort. Elle peut atteindre une vingtaine de mètres lors de certaines tempêtes.

La vitesse de propagation des vagues dépend aussi des conditions atmosphériques. Allant de simples clapots "mourants" sur la plage, la houle peut traverser l'océan pacifique dans toute sa largeur à une vitesse de plus de 500 kmh^{-1} comme celle générée par l'explosion du *Krakatoa* en 1883 [Kin65].

Deux grandes approches ont été utilisées pour modéliser les vagues :

- Les théories les plus anciennes utilisent des équations paramétriques pour cette modélisation en définissant la surface des vagues comme un mouvement ondulatoire régulier. Leur champ d'application suppose que le vent a cessé de souffler et que la vague se propage sur un fond uniforme et faiblement pentu. Leur simplicité ne permet pas d'établir une théorie générale pour la modélisation des vagues. Par contre, elles sont satisfaisantes pour une visualisation

à l'échelle humaine.

- Le second groupe de théories est basé sur une mise en équation beaucoup plus fine issue de la mécanique des fluides [Com90]. Celles-ci furent introduites par Navier et Stokes. De notre point de vue, l'infographie, leur utilisation est délicate. En effet, la résolution de ces équations par la méthode classique d'éléments finis pose des problèmes de stabilité numérique ainsi que de pragmatiques problèmes de temps de calculs : elles ne sont pas utilisables, par exemple, pour la visualisation d'une plage où l'observateur est trop proche du rivage.

1.2 Résolution des équations de mécanique des fluides

1.2.1 Théories physiques

Les deux plus importantes lois à résoudre en mécanique des fluides sont : les équations de Navier-Stokes et l'équation de continuité du fluide. Par souci de simplification de ces équations, nous nous limiterons ici à des fluides visqueux et incompressibles. De plus, ces derniers sont supposés conservatifs, i.e. qu'il n'y a pas d'apport ou de retrait de fluide pour l'ensemble du système durant la modélisation.

1.2.1.1 Les équations de Navier-Stokes

Les équations de Navier-Stokes établissent l'équilibre des forces agissant sur un fluide : forces internes (accélération des particules) et forces externes (pesanteur, vent, ...). Elles sont exprimées, sous forme vectorielle, par [Com90] :

$$\frac{d\vec{V}}{dt} = \vec{F} - \frac{1}{\rho} \text{grad} p + \nu \Delta \vec{V} \quad (1)$$

Cette équation met en relation le gradient de vitesse des particules $\frac{d\vec{V}}{dt}$ avec les forces extérieures F , les forces internes résultant des variations de pression p pour une masse volumique ρ et de l'écoulement du fluide, fonction de sa viscosité ν .

Les trois composantes du vecteur V sont u, v, w dépendantes du système de coordonnées x, y, z . Le gradient de vitesse se développe classiquement par :

$$\frac{dV}{dt} = \frac{\delta V}{\delta t} + \frac{\delta V}{\delta x} \frac{\delta x}{\delta t} + \frac{\delta V}{\delta y} \frac{\delta y}{\delta t} + \frac{\delta V}{\delta z} \frac{\delta z}{\delta t} = \frac{\delta V}{\delta t} + u \frac{\delta V}{\delta x} + v \frac{\delta V}{\delta y} + w \frac{\delta V}{\delta z}$$

Supposant que la seule force extérieure est la pesanteur g , on obtient :

$$\begin{cases} \frac{\delta u}{\delta t} + u \frac{\delta u}{\delta x} + v \frac{\delta u}{\delta y} + w \frac{\delta u}{\delta z} = -\frac{\delta p}{\rho \delta x} + \nu \left(\frac{\delta^2 u}{\delta x^2} + \frac{\delta^2 u}{\delta y^2} + \frac{\delta^2 u}{\delta z^2} \right) \\ \frac{\delta v}{\delta t} + u \frac{\delta v}{\delta x} + v \frac{\delta v}{\delta y} + w \frac{\delta v}{\delta z} = -\frac{\delta p}{\rho \delta y} + \nu \left(\frac{\delta^2 v}{\delta x^2} + \frac{\delta^2 v}{\delta y^2} + \frac{\delta^2 v}{\delta z^2} \right) \\ \frac{\delta w}{\delta t} + u \frac{\delta w}{\delta x} + v \frac{\delta w}{\delta y} + w \frac{\delta w}{\delta z} = -\frac{\delta p}{\rho \delta z} - g + \nu \left(\frac{\delta^2 w}{\delta x^2} + \frac{\delta^2 w}{\delta y^2} + \frac{\delta^2 w}{\delta z^2} \right) \end{cases} \quad (2)$$

Une version simplifiée de ces équations concernant les fluides parfaits est appelée Equations d'Euler [Com90].

1.2.1.2 L'équation de continuité du fluide

L'équation de continuité du fluide traduit le principe de conservation de masse. La variation de masse du fluide contenu dans un volume donné, durant un laps de temps donné, doit être égale à la somme des masses de fluide entrantes, diminuée de celles sortantes. La relation mathématique correspondante est :

$$\frac{\delta \rho}{\delta t} + \frac{\delta(\rho u)}{\delta x} + \frac{\delta(\rho v)}{\delta y} + \frac{\delta(\rho w)}{\delta z} = \sum \rho Q_v \quad (3)$$

où Q_v est un débit de fluide par unité de volume. Si $Q_v > 0$, le flux est entrant. Si $Q_v < 0$, le flux est sortant. Le fluide est supposé conservatif. Nous avons donc : $\sum \rho Q_v = 0$. De plus, le fluide est supposé incompressible : $\frac{\delta \rho}{\delta t} = 0$. Finalement, dans notre contexte, l'équation de continuité du fluide est :

$$\text{div}(\vec{V}) = \frac{\delta(u)}{\delta x} + \frac{\delta(v)}{\delta y} + \frac{\delta(w)}{\delta z} = 0 \quad (4)$$

1.2.2 Modélisation informatique

1.2.2.1 Méthode de résolution M.A.C.

La méthode M.A.C. (Mark-and-Cell Method) [Fle91] est couramment employée pour résoudre les deux équations des fluides vues précédemment. Elle concerne les fluides ayant une surface libre, i.e. une surface en contact avec l'air. C'est une méthode d'éléments finis où l'espace est découpé à l'aide d'une grille 3D régulière. Les deux équations sont appliquées, et contrôlées, à la frontière de chaque élément de la grille. La résolution se fait en plusieurs passes afin d'obtenir une situation d'équilibre de part et d'autre de ces frontières. En fait, le calcul des dérivées partielles est obtenu par simple soustraction de part et d'autre de chaque élément de la grille. Par exemple, si nous désirons calculer $\frac{\delta v}{\delta y}$ et connaissant les valeurs des vitesses (v_1 et v_2) sur les deux côtés ou parois de la cellule et, si la distance entre ces deux côtés est Y , on aura :

$$\frac{\delta v}{\delta y} = \frac{v_2 - v_1}{Y}$$

Les équations de Navier-Stokes approchées par éléments finis suivant un des trois axes du système sont :

$$\begin{aligned}
 \tilde{u}_{i+\frac{1}{2},j,k} = & u_{i+\frac{1}{2},j,k} + \delta t \left\{ \frac{1}{\delta x} \left[u_{i,j,k}^2 - u_{i+1,j,k}^2 \right] \right. \\
 & + \frac{1}{\delta y} \left[(uv)_{i+\frac{1}{2},j-\frac{1}{2},k} - (uv)_{i+\frac{1}{2},j+\frac{1}{2},k} \right] \\
 & + \frac{1}{\delta z} \left[(uw)_{i+\frac{1}{2},j,k-\frac{1}{2}} - (uw)_{i+\frac{1}{2},j,k+\frac{1}{2}} \right] \\
 & + \frac{1}{\delta x} \left[p_{i,j,k} - p_{i+1,j,k} \right] \\
 & + \frac{\nu}{\delta x^2} \left[u_{i+\frac{3}{2},j,k} - 2u_{i+\frac{1}{2},j,k} + u_{i-\frac{1}{2},j,k} \right] \\
 & + \frac{\nu}{\delta y^2} \left[u_{i+\frac{1}{2},j+1,k} - 2u_{i+\frac{1}{2},j,k} + u_{i+\frac{1}{2},j-1,k} \right] \\
 & \left. + \frac{\nu}{\delta z^2} \left[u_{i+\frac{1}{2},j,k+1} - 2u_{i+\frac{1}{2},j,k} + u_{i+\frac{1}{2},j,k-1} \right] \right\}
 \end{aligned} \tag{5}$$

Ici \tilde{u} est la valeur du vecteur vitesse u après un pas de temps δt et $u_{i,j,k} = \frac{1}{2} \left\{ u_{i+\frac{1}{2},j,k} + u_{i-\frac{1}{2},j,k} \right\}$. Les coordonnées données aux pixels sont entières (i, j, k) et sont situés en leur centre. Afin de définir la position des différents vecteurs vitesse placés sur les faces frontires, la notation $-\frac{1}{2}$ et $+\frac{1}{2}$ est classique.

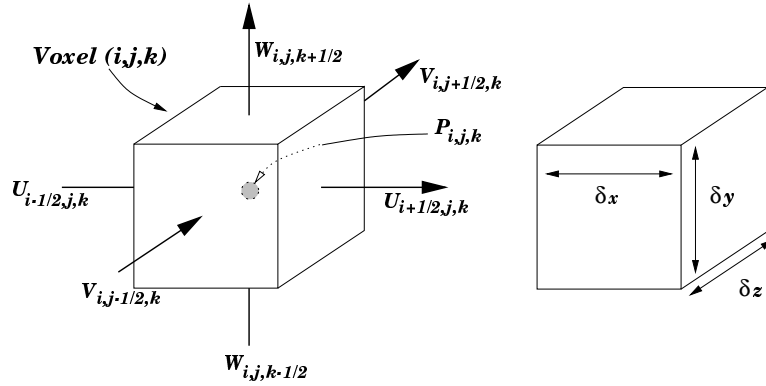


FIG. 1 – Définition des vecteurs vitesse et de la pression dans un élément de la grille

La méthode M.A.C. a été complètement reproduite en informatique graphique par Foster [FM96]. Elle permet de modéliser les mouvements de tous les types de fluides. Le principe général est le suivant : dans un premier temps, une initialisation des voxels (ou cellule de la grille 3D) est effectuée : vecteurs vitesse sur les bords, pression au centre (fig. 1). Le fluide étant incompressible, la valeur initiale de la pression du fluide pour tous les voxels peut être choisi de manière arbitraire. Pour chaque pas de temps, un recalcul des vecteurs vitesse à la frontière de chaque voxel est effectué en résolvant les équations de Navier-Stokes (eq. 1).

Ceci est effectué pour chaque composante u, v, w des vitesses et pour toute la scène. L'ensemble de ces nouvelles vitesses ne permet pas de garantir la conservation de masse et donc de vérifier que

l'équation de continuité du fluide est satisfaite. En effet, nous rappelons que l'ensemble du système doit vérifier l'équation $\text{div}(\vec{V}) = 0$. Pour cela, et par un jeu d'aller-retour dans les voxels, on va modifier localement les pressions et les vitesses pour converger vers un équilibre de l'équation de conservation de masse. La méthode utilisée est la suivante :

- Dans un premier temps, on va calculer la valeur en différence finie de la divergence D du vecteur vitesse moyen d'un voxel :

$$D_{i,j,k} = \frac{1}{\delta x}(u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}) + \frac{1}{\delta y}(u_{i,j+\frac{1}{2},k} - u_{i,j-\frac{1}{2},k}) + \frac{1}{\delta z}(u_{i,j,k+\frac{1}{2}} - u_{i,j,k-\frac{1}{2}}) \quad (6)$$

- Partant de ce calcul et pour satisfaire l'équilibre des masses, les auteurs modifient localement la pression du voxel par $\delta p = \frac{1,6D}{2\delta t} \left\{ \frac{1}{\delta x^2} + \frac{1}{\delta y^2} + \frac{1}{\delta z^2} \right\}$. La nouvelle valeur de la pression est alors égale à : $p = p + \delta p$
- Les vecteurs vitesse des faces du voxel sont alors modifiés par substitution de $p = p + \delta p$ dans l'équation 5 (Le résultat est présenté ici sur une seule des trois composantes, les autres étant similaires) :

$$\begin{cases} u_{i+\frac{1}{2},j,k} = u_{i+\frac{1}{2},j,k} + \frac{\delta t}{\delta x} \delta p \\ u_{i-\frac{1}{2},j,k} = u_{i-\frac{1}{2},j,k} - \frac{\delta t}{\delta x} \delta p \end{cases} \quad (7)$$

Pourquoi un jeu d'aller-retour ? Tout simplement parce que lorsqu'on modifie localement les vecteurs vitesse d'un voxel pour satisfaire la deuxième équation, on modifie les vecteurs vitesse des voxels voisins. Donc on va itérativement modifier l'équilibre pression-vitesse des voxels jusqu'à ce que le système se stabilise et que la divergence de chaque voxel soit inférieure à un certain seuil. Malheureusement, il arrive parfois que le système se mette à osciller.

Pour pouvoir modéliser géométriquement les mouvements du fluide, le vecteur vitesse de chaque particule incluse dans un voxel est calculé par interpolation linéaire entre les vecteurs vitesse définis sur les faces du voxel.

Le problème inhérent à la méthode réside dans le fait que chaque particule de fluide ne doit pas se déplacer de plus d'un voxel à chaque pas de calcul afin d'éviter l'amplification des erreurs commises par l'approximation des équations de Navier-Stokes. Ceci s'explique par le fait que l'on approxime les dérivées partielles des vitesses par des différences finies et que le calcul de ces différences est lié à l'échantillonnage (la largeur d'un voxel). Deux méthodes ont été développées pour les contourner :

- La première, proposée par Foster [FM96], consiste à calculer le pas de temps δt en fonction de la grandeur des vecteurs vitesse dans le système. Malheureusement, cette technique est difficile et coûteuse à mettre en oeuvre pour générer des animations. En effet, l'intervalle de temps entre deux images successives doit être constant. Le pas de temps de la méthode étant variable, la synchronisation des images est délicate. Ceci peut conduire en particulier

à de nombreux recalculs du système entre deux prises de vues.

- La deuxième, proposée par Foster [FM97] et plus récemment par Arques [MAZ98], consiste à déterminer la pression de chaque voxel du système au pas $t + 1$ avant de déterminer les vitesses. Pour plus de détails, nous invitons le lecteur à se reporter aux deux articles précédemment cités.

Enfin, une difficulté majeure est l'obtention des vecteurs vitesse des fluides lorsqu'il existe une interface entre liquide et air dans un même voxel. Ceci est un problème récurrent sur toute la surface du liquide. Foster propose une description des 64 situations de discontinuités possibles que l'on ne détaillera pas ici. Quelques exemples sont présentés figure 2 :

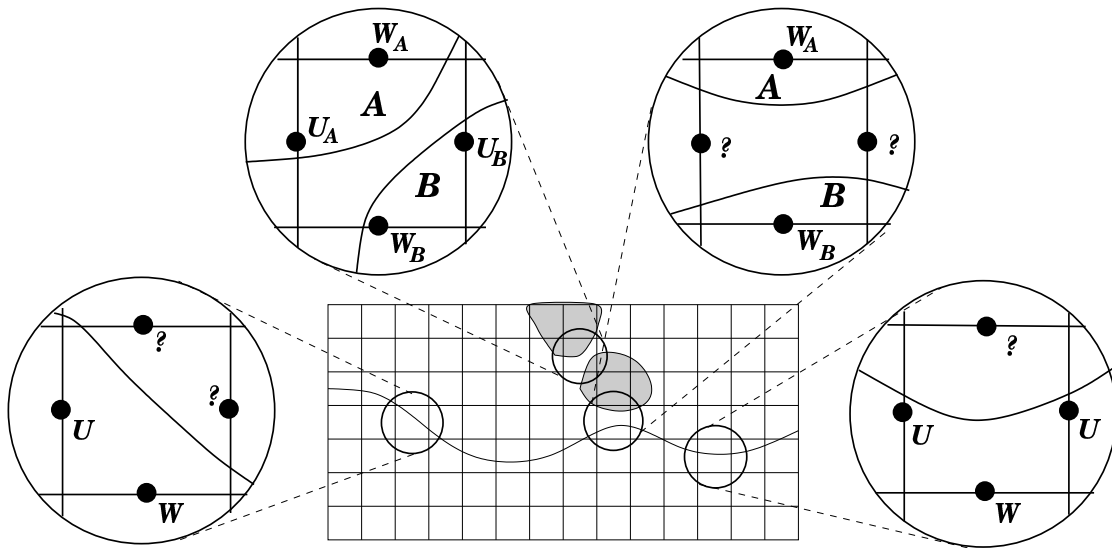


FIG. 2 – Quelques cas particuliers pour la méthode MAC

La figure 2 représente, en vue de coupe, la surface d'un fluide et deux "gouttes" d'eau au dessus. Les zooms effectués à gauche et à droite de la figure présentent les cas où une partie de l'information est manquante. Les deux zooms du haut montrent les cas où sur un même voxel, l'ensemble des vecteurs vitesses sont définis mais ne représentent pas forcément les mêmes fluides (goutte et surface ici).

1.2.2.2 Résolution 2D des équations de mécanique des fluides

Cette méthode proposée par Chen et Lobo [CL95] permet de modéliser l'ensemble des phénomènes régissant un liquide en ne résolvant les équations de mécanique des fluides que sur la surface du liquide. La grille permettant de stocker les différents vecteurs vitesse est alors une simple grille 2D où seules les composantes u et v sont enregistrées.

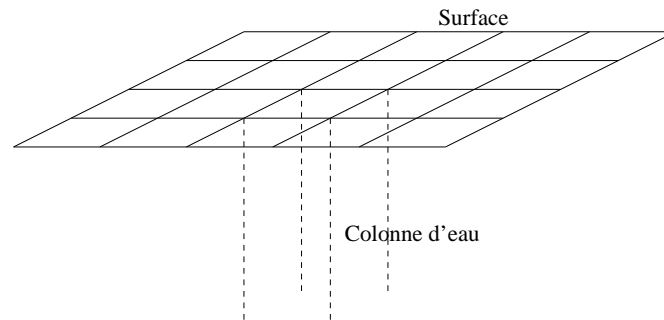


FIG. 3 – Résolution en 2D

La pression d'une cellule représente la colonne d'eau située au dessous d'elle (fig. 3). Le calcul des vecteurs vitesse pour un pas de temps est similaire à celui vu précédemment. Par contre, pour satisfaire l'équation de conservation de masse, plutôt que de faire varier la pression de la cellule, la hauteur de la colonne de fluide va être modifiée. Si la divergence du vecteur vitesse de la cellule est positive, donc s'il y a augmentation de masse, la hauteur de la colonne d'eau de cette cellule va augmenter. Ainsi, grâce à un artifice et avec un simple algorithme 2D, on peut représenter une surface 3D de liquide.

Cet algorithme peut être exécuté en temps réel. Malheureusement, cette méthode a deux grands inconvénients :

- Il n'y a aucune notion de profondeur. On ne peut donc pas tenir compte de l'influence du sol sur la propagation du liquide dans la scène.
 - La méthode de stockage et de calcul 2D ne permet pas non plus d'avoir des liquides déferlants.
- Or, ceci est un de nos buts.

1.2.3 Bilan

Ces méthodes, basées sur une résolution des lois générales de la mécanique des fluides donnent des résultats impressionnants. En effet, elles peuvent être appliquées dans beaucoup de situations. Hélas, leur résolution basée sur les éléments finis est conditionnée par la précision d'échantillonnage de la grille de voxels utilisée. Des images de qualité nécessitent un maillage fin qui impose les temps de calcul beaucoup plus long. Elles ne sont réellement efficaces, pour les scènes océanes, que lorsque l'observateur est placé loin, voire très loin de la scène.

Nous avons testé la méthode de Foster [FM96] avec une grille 100x100, en 2D, où la moitié des voxels contenait du liquide. Le temps de calcul pour passer d'un pas de temps à un autre est de l'ordre de 4 secondes sur une Silicon Graphics O2 avec un processeur R5500. Entre deux images d'une animation, une moyenne de 50 itérations de l'algorithme ont été nécessaires. Malheureusement ces équations, du fait de leurs larges champs d'application, sont assez instables et il faut

pour cela soit diminuer au maximum le pas de temps entre deux étapes de calculs pour réduire les erreurs d'approximations, soit d'inclure des contraintes sur la scène à modéliser. De plus, la méthode programmée s'est avérée complexe à maîtriser numériquement.

1.3 Modélisation paramétrique de la houle

La deuxième approche est basée sur deux faits : premièrement, les vagues sont une onde se déplaçant à la surface et deuxièmement, chaque particule d'eau admet un trajet périodique centré au sein de l'onde. Ceci offre la possibilité d'utiliser des équations paramétriques pour décrire la vue en coupe des vagues.

Dans cette partie, nous verrons dans un premier temps les principaux modèles physique. Après un tour d'horizon des différentes applications effectuées en infographie, nous développerons notre modèle de houle¹. Nous terminerons par quelques exemples de vagues déferlantes générées par notre modèle.

1.3.1 Théories physiques

Hypothèses générales communes aux différentes théories. Tout en restant suffisamment rigoureux pour valider les applications, l'expression mathématique des modèles est assez simple. Par contre, un certain nombre d'hypothèses ont été faites et, selon les modèles :

- La pression atmosphérique p est constante en tout point de la région modélisée,
- l'océan est un fluide parfait, homogène, incompressible et pesant,
- le mouvement est supposé périodique,
- selon les modèles, des hypothèses supplémentaires sont faites sur la profondeur et la pente des fonds marins,
- le vent a cessé de souffler.

Les différents modèles utilisent les coordonnées de Lagrange pour déterminer la position de chaque particule en fonction de leur position à $t = 0$. Le centre du repère de coordonnées est placé au large.

Définitions et Notations.

- La surface de l'océan, en pleine mer, présente une suite infinie d'ondulations parallèles presque identiques qui se propagent de façon sensiblement uniforme vers le rivage. On appelle **houle**, cet ensemble d'ondulations (fig. 4).
- La **surface libre** est le niveau d'eau théorique de l'océan au repos sans aucune ondulation.

¹Une version plus concise des travaux développés dans cette partie se trouve dans [GS97].

- L'**amplitude** ou **hauteur** H de vague est la différence de niveau entre un creux et une crête successifs.
- La **période** T est le temps qui sépare le passage de deux crêtes consécutives en un point fixe. On appelle **longueur d'onde** L , la distance qui sépare deux crêtes successives.
- La **célérité** C de la houle est la vitesse moyenne de propagation des crêtes. Cette vitesse vaut :

$$C = \frac{L}{T} \quad (8)$$

- La **profondeur** h est la distance entre la surface libre de l'eau et le sol.

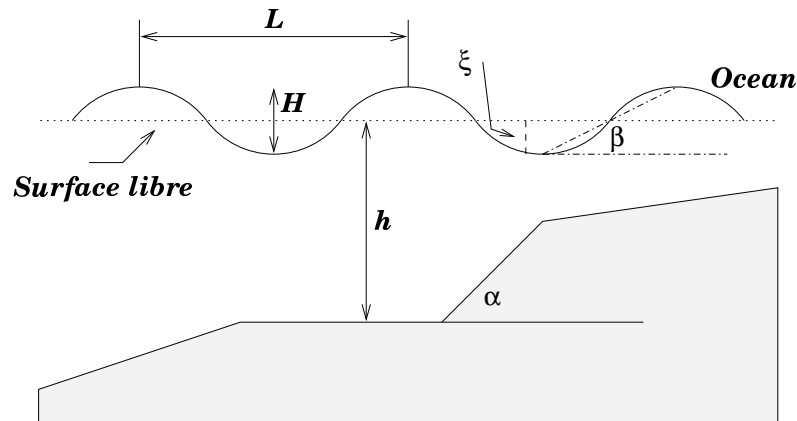


FIG. 4 – Les différentes données sur les vagues

- La **cambrure** δ de la vague est le rapport entre son amplitude H et sa longueur d'onde L . On notera β la pente entre le creux de la vague et la crête :

$$\delta = \frac{H}{L} \quad (9)$$

$$\tan \beta = \frac{H}{\frac{L}{2}} = \frac{2H}{L} = 2\delta \quad (10)$$

- On notera ξ la **cote de la surface libre** en un point courant : $\xi = z_0 - z_r$, où z_r est la hauteur d'eau de référence que l'on prendra dans ce document toujours égale à zéro et z_0 la hauteur d'eau de la surface libre (i.e. sans aucune ondulations).
- On notera p la **pression atmosphérique**.
- On notera ρ la **masse volumique du fluide**.
- On notera g la force de **gravitation**.
- On appellera K **nombre d'onde**. Ce nombre permet de définir la longueur d'onde de la vague :

$$L = \frac{2\pi}{K} \quad (11)$$

1.3.1.1 La houle de Gerstner

Le premier modèle complet, simple et exact, pour les vagues d'amplitudes finies a été présenté par *Gerstner* en 1802 [Ger09]. Il suppose que la profondeur d'eau est infinie. Un modèle identique a été aussi présenté indépendamment par Rankine en 1863 [Ran63].

Modèle général. Chaque particule de fluide est supposée tourner autour d'un point de coordonnées x_0, z_0 en décrivant une circonférence de rayon R_0 , ce dernier décroissant exponentiellement en fonction de sa distance à la surface. Cette décroissance est pondérée par un coefficient, qui, sous les hypothèses de satisfaction de l'équation de continuité du fluide, vaut K . La vitesse de rotation des particules dépend classiquement d'un coefficient ω de vitesse angulaire.

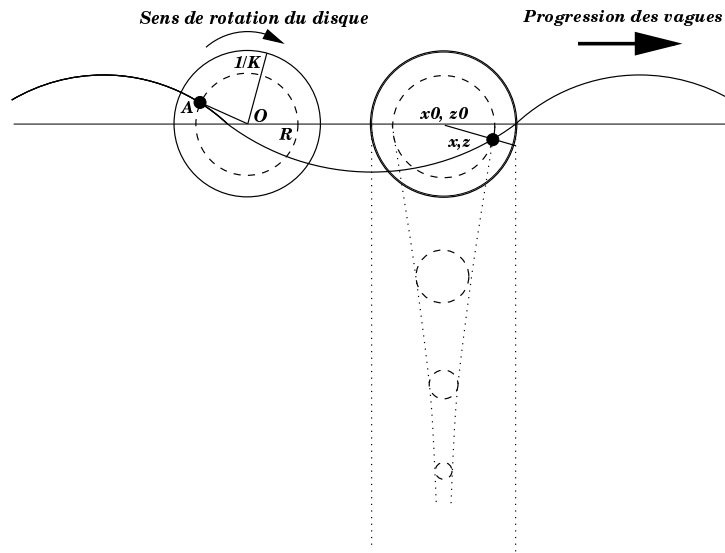


FIG. 5 – Vue générale de la houle de Gerstner

Les coordonnées de chaque particule (voir figure 5) sont :

$$x = x_0 - R_0 e^{Kz_0} \sin(Kx_0 - \omega t) \quad (12)$$

$$z = z_0 + R_0 e^{Kz_0} \cos(Kx_0 - \omega t) \quad (13)$$

Pour simplifier, on notera R le rayon décrivant le mouvement des particules d'eau :

$$R = R_0 e^{Kz_0} \quad (14)$$

Les équations de l'hydrodynamique permettent d'exprimer la pression p en fonction de la densité du fluide, de la pesanteur, etc ... [Bon80] :

$$\frac{p}{\rho} = g\left(\frac{\omega^2 R^2}{2g} - z_0\right) - \left(\frac{\omega^2}{gK} - 1\right)gR \cos(Kx_0 - \omega t)$$

Selon les hypothèses des modèles paramétriques, la pression est constante durant le mouvement. Il existe donc une surface isobare quelque soit le temps si :

$$\frac{\omega^2}{gK} = 1 \Leftrightarrow L = \frac{gT^2}{2\pi} \quad (15)$$

La pression p à la surface s'exprime donc de la manière suivante :

$$p = \rho g\left(\frac{\omega^2 R^2}{2g} - z_0\right) \Leftrightarrow p = \rho g\left(\frac{\pi R^2}{L} - z_0\right) \quad (16)$$

La surface libre, i.e. le lieu décrit par les particules où la pression est par hypothèse constante et nulle ($p = 0$), est caractérisée par :

$$z_0 = \frac{\pi R^2}{L}$$

Le centre de la circonférence décrite par les particules de la surface libre est donc situé à la hauteur :

$$z_0 = \frac{\pi H^2}{4L} \quad (17)$$

Si $z = 0$ à $t = 0$, on a $\xi = \frac{\pi H^2}{4L}$ (fig. 6) :

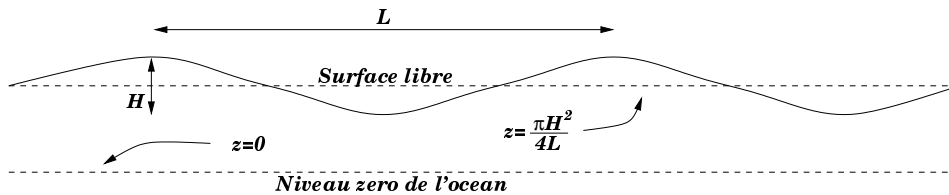


FIG. 6 – Niveau de la surface libre

Visualisation et commentaires. Les particules en déplacement sont situées sur une *trochoïde* dont l'équation paramétrique est donnée par les équations 12 et 13.

Si $KR < 1$, la courbe générée est située sur une trochoïde. Si $KR = 1$, la courbe est une cycloïde. Finalement, dès que $KR > 1$ la trajectoire des particules ne correspond plus à un phénomène

naturel. La figure 7 illustre les différents types de comportements des équations 12 et 13. Les données utilisées sont respectivement $KR = 0.2, 0.4, 0.6, 0.8, 1.0, 1.2$ pour les six courbes.

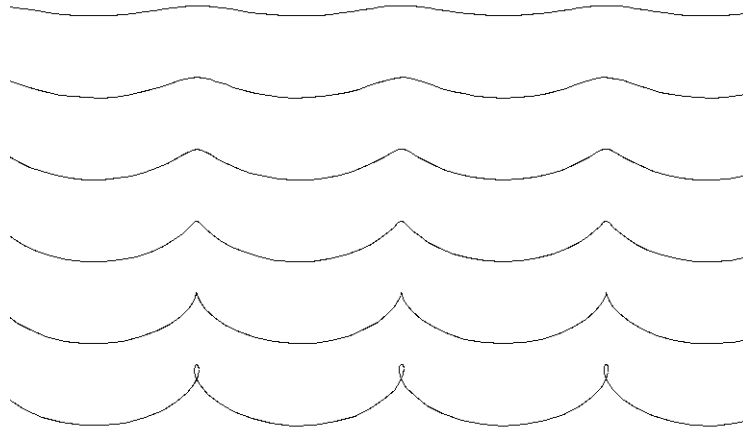


FIG. 7 – Vue de la surface de la mer en fonction de K et R

La surface libre et toutes les surfaces d'égal pression sont caractérisées par la cote z_0 et le rayon R . Les particules correspondantes sont situées sur la trochoïde d'équation paramétrique (12 et 13). Il s'agit du lieu décrit par un point A d'un disque de rayon $\frac{1}{K}$ qui roule sans glisser, tel que :

$$OA = R \text{ (cf. fig. 5)}$$

A l'instant t_1 , tous les points de même z_0 et R auront tourné sur leur trajectoire circulaire d'un angle égal à $\omega(t_1 - t_0) = 2\pi\frac{t_1 - t_0}{T}$.

La surface sera déplacée dans le sens de rotation des cercles ; la trochoïde représentant la surface libre et toutes les surfaces d'égal pression, se déplacent donc d'un mouvement de translation non déformé (Fig. 8). Il faut préciser que seule l'onde se déplace à la surface, les particules de fluide ont un trajet circulaire situé autour de leur position de repos. Ainsi, la houle de Gerstner est aussi qualifiée de **houle trochoïdale**.

On peut remarquer que les vitesses des particules sont tangentes à la crête et au creux. Ces vecteurs vitesse sont dirigés dans le sens de la propagation sur la crête et en sens inverse dans le creux. Ainsi, un objet à la surface aura tendance à aller dans le sens de propagation de la houle lorsqu'il se trouve sur la crête et à revenir en arrière lorsqu'il se trouve dans le creux (fig. 9).

Calculons les composantes de la vitesse des particules. Soit u (resp. w) la composante horizontale (resp. la composante verticale) :

$$u = \frac{dx}{dt} = \omega R \sin(Kx_0 - \omega t) \quad (18)$$

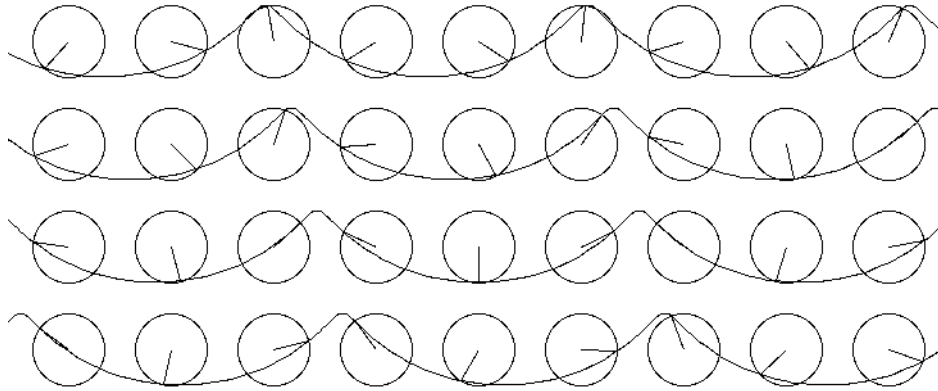


FIG. 8 – Propagation des vagues

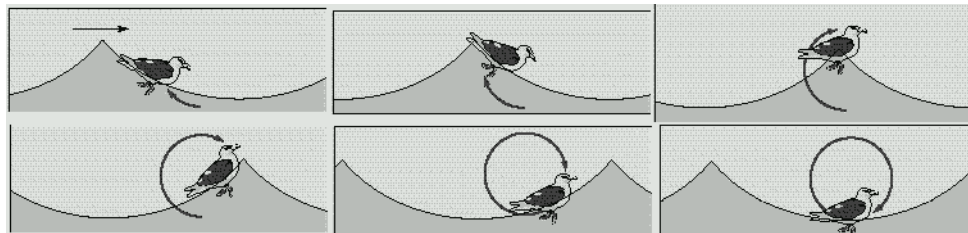


FIG. 9 – Mouvement d'un objet à la surface de l'océan

$$w = \frac{dz}{dt} = \omega R \cos(Kx_0 - \omega t) \quad (19)$$

Enfin, on peut déterminer la cambrure maximale de la houle de Gerstner. L'amplitude maximale H_{max} est obtenue lorsque $R = \frac{1}{K} \Rightarrow H_{max} = \frac{2}{K} = \frac{L}{\pi}$. La valeur maximale de la cambrure équivaut à :

$$\frac{H_{max}}{L} = \frac{\frac{L}{\pi}}{L} = \frac{1}{\pi} = 0,31 \quad (20)$$

1.3.1.2 La houle irrotationnelle de Stokes

Introduction. Le modèle de Stokes est basé sur l'étude des vecteurs vitesse des particules. Ce modèle n'est applicable qu'à partir d'une certaine profondeur.

L'hypothèse forte de base est que le *rotationnel du fluide est nul* i.e. la houle est dite irrotationnelle.

De manière simpliste, on peut dire que le mouvement le plus général d'une particule de fluide peut se décomposer en deux phénomènes bien distincts : une translation de vitesse v et une rotation

autour de son axe à la vitesse $\omega = \frac{1}{2}\text{rot}(\vec{v})$. Cette rotation interne est donnée par une grandeur appelée *rotationnel*. Pour plus de détail, nous invitons le lecteur à consulter des ouvrages spécialisés en mécanique des fluides [Com90].

La houle de Stokes étant irrotationnelle, le mouvement d'une particule du fluide ne dépend que d'une translation et donc, d'un potentiel de vitesses $\Phi(x, z, t)$. Les composantes du vecteur vitesse peuvent s'exprimer de la sorte :

$$u = \frac{\delta\Phi}{\delta x} \quad (21)$$

$$w = \frac{\delta\Phi}{\delta z} \quad (22)$$

Hypothèses

Φ est une fonction harmonique, qui doit être périodique dans l'espace et le temps. Elle est telle que w s'annule sur le fond pour $z = -h$ (fig. 10).

De plus, certaines nouvelles hypothèses doivent être prises en compte pour pouvoir obtenir le modèle de Stokes au premier ordre d'approximation :

- La cambrure doit être faible : $\delta = \frac{H}{L} \ll 1$
- L'amplitude doit être faible : $\frac{H}{h} \ll 1$

Avec ces hypothèses, le potentiel de vitesse doit vérifier les conditions suivantes :

- $\Delta\Phi = 0$ pour $-h < z < 0$, i.e. le rotationnel doit être nul.
- $\frac{\delta^2\Phi}{\delta t^2} + g\frac{\delta\Phi}{\delta z} = 0$ pour $z = 0$. Ceci est une hypothèse de l'hydrodynamique prenant en compte la vitesse des particules par rapport à la vitesse du train d'ondes. [AA85]
- $\frac{\delta\Phi}{\delta z} = 0$ pour $z = -h$. La composante verticale de la vitesse doit être nulle au niveau du sol.

Développement limité au premier ordre. La solution proposée par Stokes, qui suppose le fond à une grande profondeur, développée au premier ordre, est la suivante :

$$\Phi(x, z, t) = -\frac{gHT}{4\pi} \frac{\cosh(K(z+h))}{\sinh(Kh)} \sin(Kx - \omega t) \quad (23)$$

Le mouvement des particules de fluide pendant le déplacement de la houle se déduit du potentiel de vitesse donné précédemment. Ce mouvement peut être déduit à partir d'une loi générale de la mécanique des fluides (appelée loi de conservation de l'énergie) dérivée par l'équation de Bernoulli et présentée ici dans le cadre d'un fluide irrotationnel :

$$\frac{1}{2} \left[\left(\frac{\delta\Phi}{\delta x} \right)^2 + \left(\frac{\delta\Phi}{\delta z} \right)^2 \right] + \frac{p}{\rho} + gz + \frac{\delta\Phi}{\delta t} = F(t)$$

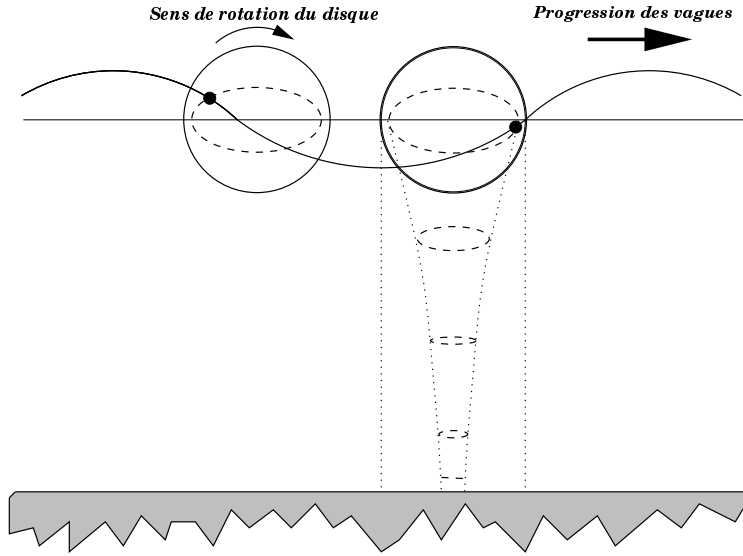


FIG. 10 – Houle de Stokes au premier ordre

A la surface libre, la pression est nulle $p = 0$ et $z = \xi$. La fonction $F(t)$ est une constante en considérant un point au repos à l'infini i.e. sans aucune ondulation. On obtient au premier ordre d'approximation :

$$\frac{\delta\Phi}{\delta t} + g\xi = 0 \quad (24)$$

L'expression de la cote de la surface libre est :

$$\xi(x, t) = -\frac{1}{g} \frac{\delta\Phi}{\delta t} \text{ pour } z = 0$$

$$\xi(x, t) = \frac{H}{2} \cos(Kx - \omega t) \quad (25)$$

L'amplitude de la vague est égale à H . L'onde se propageant à la surface est sinusoïdale.

Les composantes du vecteur vitesse des particules sont :

$$u = \frac{\delta\Phi}{\delta x} = \frac{-gHTK \cosh(K(z+h))}{4\pi \sinh(Kh)} \cos(Kx - \omega t) \quad (26)$$

$$w = \frac{\delta\Phi}{\delta z} = \frac{-gHTK \sinh(K(z+h))}{4\pi \sinh(Kh)} \sin(Kx - \omega t) \quad (27)$$

La vitesse horizontale est en phase avec la variation ξ du niveau de la surface libre, atteignant ses valeurs extrêmes à la verticale des crêtes (où elle est dirigée dans le sens de propagation de l'onde) et des creux (où elle est dirigée dans le sens inverse).

De plus, la condition préalable d'imperméabilité du fond i.e. $w = 0$ pour $z = -h$ est vérifiée. L'absence de frottement au fond se traduit par la non-nullité de u pour $z = -h$. Ces expressions sont surtout utilisées lorsqu'on cherche à évaluer les efforts exercés par la houle sur une structure fixe.

Trajectoires des particules. Des observations dans un canal à houle ont montré que les particules de fluide décrivent des orbites pratiquement fermées dont les dimensions sont assez petites devant la longueur d'onde. En première approximation, les vitesses u et w peuvent donc être confondues avec leur valeur au centre (x_c, z_c) de chaque orbite. D'où la position d'une particule donnée par :

$$\int_0^t \vec{u}(x_c, z_c) d\vec{u} \quad (28)$$

Les orbites sont donc (eq. 26 et 27) des ellipses dont le demi-grand axe R_x (horizontal) et demi-petit axe R_z (vertical) valent respectivement :

$$R_x = \frac{H \cosh(K(z+h))}{2 \sinh(Kh)} \quad (29)$$

$$R_z = \frac{H \sinh(K(z+h))}{2 \sinh(Kh)} \quad (30)$$

A la surface libre ($z = 0$), on obtient $R_z = \frac{H}{2}$. Ceci exprime bien le fait qu'une particule située sur cette surface y reste tout au long de son mouvement. On peut remarquer, de plus, que si la profondeur est importante ($Kh \gg 1$), ces orbites sont pratiquement circulaires. Par contre, les particules qui sont situées au fond de l'océan, subissent un unique mouvement de va et vient.

Développement limité à des ordres supérieurs. En poussant les développements limités à des ordres supérieurs, on obtient les différentes houles non-linéaires de Stokes. Cette étude se traduit par l'introduction d'harmoniques d'ordre supérieur.

Les données suivantes sont tirées de Aristaghes [AA85].

$$\mu = Kx - \omega t \quad (31)$$

Stokes d'ordre 1.

$$\begin{cases} \xi_1 = \frac{H}{2} \cos(\mu) \\ \Phi_1 = \frac{gHT}{4\pi} \frac{\cosh(K(z+h))}{\sinh(Kh)} \sin \mu \\ L = \frac{gT^2}{2\pi} \tanh\left(\frac{2\pi h}{L}\right) \end{cases} \quad (32)$$

Stokes d'ordre 2.

$$\begin{cases} \xi_2 = \xi_1 + \frac{\pi H}{4L} \frac{3 - \tanh^2(Kh)}{\tanh^3(Kh)} \cos(2\mu) \\ \Phi_2 = \Phi_1 + \frac{3\pi H^2}{16T} \frac{\cosh(2K(z+h))}{\sinh^4(Kh)} \sin(2\mu) \\ L = \frac{gT^2}{2\pi} \tanh\left(\frac{2\pi h}{L}\right) \end{cases} \quad (33)$$

Stokes d'ordre 3.

$$\begin{cases} \xi_3 = \xi_2 + \frac{3\pi^2 H^3}{128L^2} \frac{1+8\cosh^6(Kh)}{\sinh^6(Kh)} \cos(3\mu) \\ \Phi_3 = \Phi_2 + \frac{\pi^2 H^3}{128LT} \frac{11-2\cosh(2Kh)}{\sinh^7(Kh)} \cosh(3K(z+h)) \sin(3\mu) \\ L = \frac{gT^2}{2\pi} \tanh\left(\frac{2\pi h}{L}\right) \left[1 + (KI)^2 \frac{14+4\cosh^2(2Kh)}{16\sinh^4(Kh)}\right] \\ H = I + \frac{3I^3}{64L^2} \frac{1+8\cosh^2(2Kh)}{\sinh^6(Kh)} \end{cases} \quad (34)$$

Ces théories sont surtout utilisées pour le calcul des efforts exercés par la houle sur des structures en mer telles que les plate-formes pétrolières. Des calculs aussi complexes ne sont pas nécessaires dans le cas de calculs de visualisation de la surface de la zone portuaire. En effet, la houle n'y est pas régulière et subit beaucoup de modifications dues aux variations du sol, des digues, des brises-lames.

1.3.1.3 La houle de Biesel

Le modèle de Biesel [Bie52] se place dans le cas de la description complexe de la houle arrivant sur un rivage tenant compte des changements de vitesse des particules et de la formation du déferlement qui sera vu dans la partie suivante.

Dans ce modèle, la trajectoire circulaire des particules d'eau situées à la surface se transforme progressivement en ellipse dont le grand axe s'aligne à la pente. Ceci permet de prendre en compte les frottements des particules sur le fond. A faible profondeur, la houle est freinée et ainsi déformée. Malheureusement, les précédentes théories paramétriques ne sont généralement valables qu'à partir d'une certaine profondeur. La limite inférieure est toujours au moins égale à une profondeur d'une demi-longueur d'onde. Entre la surface libre et cette profondeur limite, toutes les particules d'eau sont en mouvement. Lorsque les vagues s'approchent du rivage, des frottements sur le sol interviennent et la houle est ralentie. On peut remarquer que seule la période reste constante. La longueur d'onde est plus courte au bord du rivage qu'au large et peut être calculée par :

$$\frac{L}{L_\infty} = \frac{K_\infty}{K} = \tanh(Kh) \quad (35)$$

L'amplitude théorique de la houle décroît continuellement jusqu'à 91% de sa valeur au large avant d'augmenter jusqu'au point de déferlement. Elle est donnée par la loi [Lac65] (fig. 11) :

$$\left(\frac{H}{H_\infty}\right)^2 = \frac{1}{\tanh(Kh) \left(1 + \frac{2Kh}{\sinh(2Kh)}\right)} \quad (36)$$

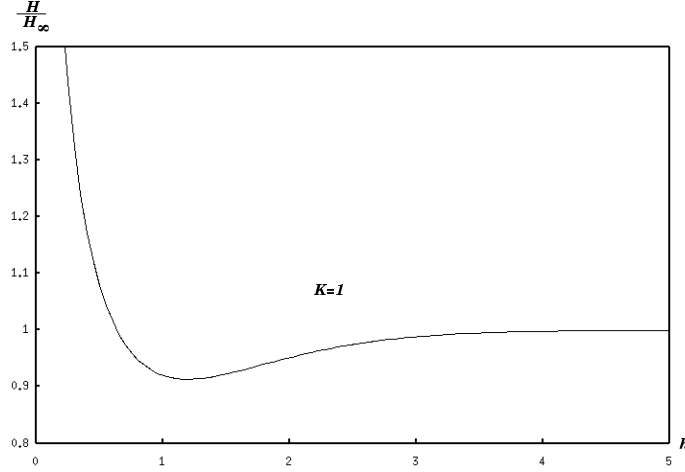


FIG. 11 – Amplitude relative des vagues en fonction de la profondeur

Biesel tient compte de la modification d'amplitude dans sa théorie. De plus, ce modèle permet de décrire une vague jusqu'à son point de déferlement (fig. 12). Les équations paramétriques du modèle sont :

$$x = x_0 + A_x \sin \left(\int_0^{x_0} K dx - \omega t \right) - B_x \cos \left(\int_0^{x_0} K dx - \omega t \right) \quad (37)$$

$$z = z_0 - A_z \cos \left(\int_0^{x_0} K dx - \omega t \right) - B_z \sin \left(\int_0^{x_0} K dx - \omega t \right) \quad (38)$$

avec

$$\begin{cases} A_x = \frac{\Theta}{T_h} \\ A_z = \Theta \\ B_x = \Theta T_\alpha \left(\frac{1+KhT_h}{\Psi^2 T_h^2} + \frac{2KhT_h+(Kh)^2}{2\Psi S_h C_h T_h} - \frac{Kh}{T_h} - 1 \right) \\ B_z = \Theta T_\alpha \left(\frac{Kh+T_h}{\Psi^2 T_h^2} + \frac{2Kh+(Kh)^2 T_h}{2\Psi S_h C_h T_h} - \frac{1}{T_h} - Kh \right) \end{cases}$$

et

$$\begin{cases} S_h = \sinh(Kh) \\ C_h = \cosh(Kh) \\ T_h = \tanh(Kh) \\ T_\alpha = \tan(\alpha) \\ \Psi = \left(1 + \frac{Kh}{\sinh(Kh) \cosh(Kh)} \right) \\ \Theta = \frac{R}{\sqrt{\Psi T_h}} \end{cases}$$

Enfin, la valeur de K varie avec la profondeur suivant la fonction 35.

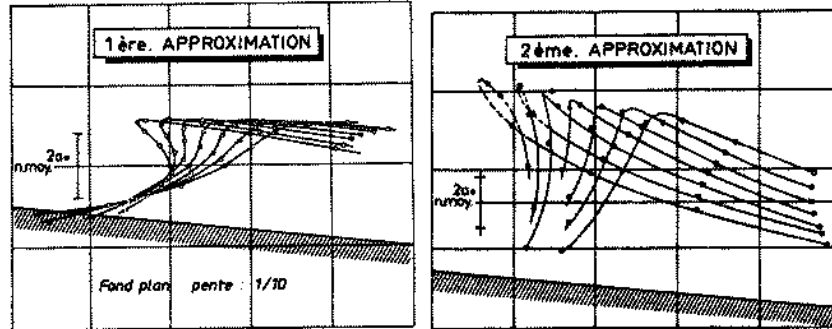


FIG. 12 – Houle de Biesel en bord de rivage (d'après [Bie52])

Comme on le voit sur la figure 12 (partie gauche), ce modèle permet une génération partielle de houle déferlante. Par contre, le déferlement ne peut être complètement modélisé (fig. 12 partie droite). Ceci est aussi le cas avec les modèles paramétriques précédemment exposés.

1.3.1.4 Etude au cas limite : le déferlement

Dans cette partie, nous allons décrire sommairement les différents types de déferlements ainsi que leurs principales caractéristiques.

Introduction. Le déferlement est un phénomène au cours duquel l'onde est partiellement détruite ; il est caractérisé par un haut degré de turbulences et une grande dissipation d'énergie. Une vague ne déferle pas uniquement en eaux peu profondes. Il arrive couramment de voir les vagues se briser en pleine mer. Du point de vue hydrodynamique, le phénomène se produit lors des éventualités suivantes :

- La vitesse des particules sur la crête dépasse celle de l'onde,
- l'accélération des particules sur la crête est supérieure à la pesanteur,
- la surface devient verticale et la gravité écrase le sommet de la vague.

Pour les houles progressives irrotationnelles, le déferlement est lié à une valeur maximale de la cambrure. Les formules proposées sont :

- $\gamma = \frac{H}{L} < 0,142$ en eau profonde (Michell, 1893),
- $\gamma = \frac{H}{L} < 0,14 \tanh(Kh)$ en eau moyennement profonde² (Miche, 1944),
- $\gamma = \frac{H}{h} < 0,78$ en eau peu profonde (Munk, 1949).

²où $h \equiv \frac{L}{2}$

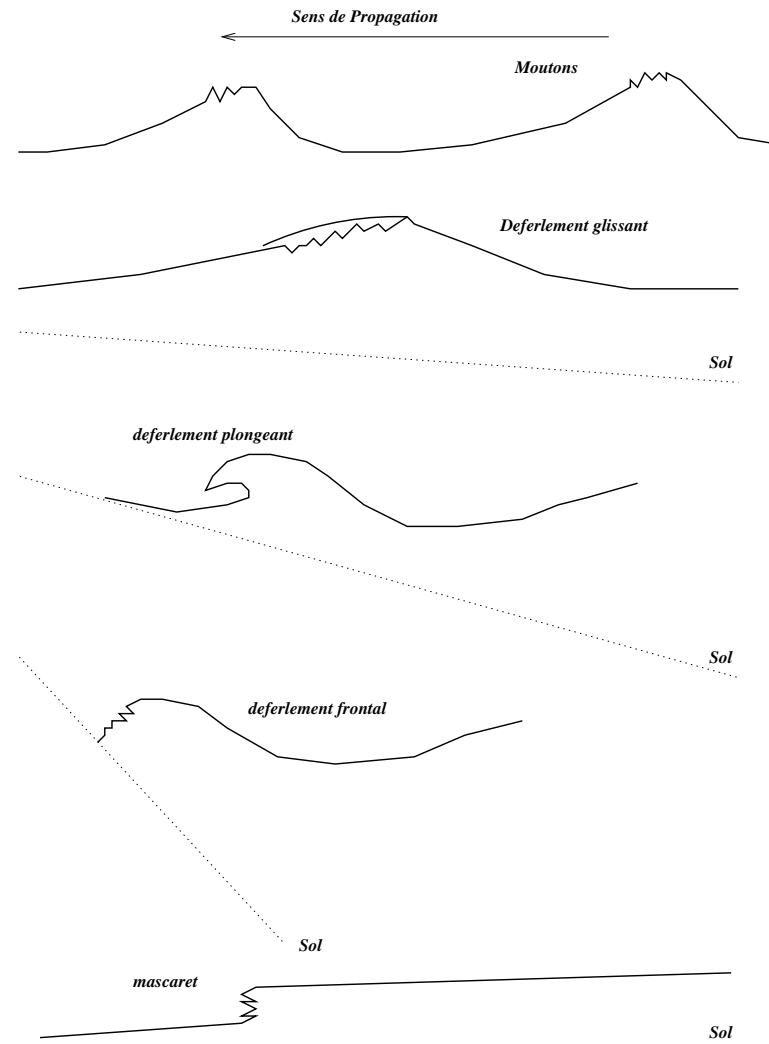


FIG. 13 – Différents types de déferlements

Déferlement en eaux profondes. En pleine mer, l'augmentation de la hauteur d'une vague ne peut pas provenir de l'influence du sol. Le vent, qui est aussi la cause de la houle, est le facteur majeur influençant l'amplitude de la vague. Etant donné qu'en pleine mer la longueur d'onde d'un train de vagues est quasiment stable, lorsque l'amplitude tend vers $0.142L$, la vague devient instable et un moutonnement en crête apparaît (fig. 13).

Michell, comme dit précédemment, a trouvé que la cambrure maximale en pleine mer est de :

$$\frac{H_b}{L_b} = 0,142$$

D'après Lacombe [Lac65], la longueur d'onde limite L_{max} avant déferlement est égale à 1,193 fois sa valeur correspondant aux faibles hauteurs. Or d'après la théorie de Gerstner (eq. 15) :

$$L_{max} = 1,193 \frac{gT^2}{2\pi}$$

On peut donc en déduire la valeur de l'amplitude maximale H_{max} au point de déferlement en fonction de la période de la houle :

$$H_{max} = 0,142 * 1,193 \frac{gT^2}{2\pi} = 0,265T^2$$

Déferlement en eaux peu profondes. Le type de déferlement dépend essentiellement de la pente du fond marin en bord de côte. Sur un fond de pente faible, la vague déferle en glissant sur sa face avant (*déferlement glissant* fig. 13). Lorsque la pente du fond augmente, le déferlement s'effectue sous la forme connue des rouleaux plongeants. Enfin sur fond très incliné, le déferlement se manifeste par l'écoulement progressif d'un front d'ondes.

Le mascaret est apparenté au déferlement, mais ne concerne que les houles de grande longueur d'onde arrivant sur fond quasiment plat.

D'après Bonnefille [Bon80], le graphique 14 permet de déterminer le type de déferlement en fonction de la pente de la plage et de la cambrure de la vague au large.

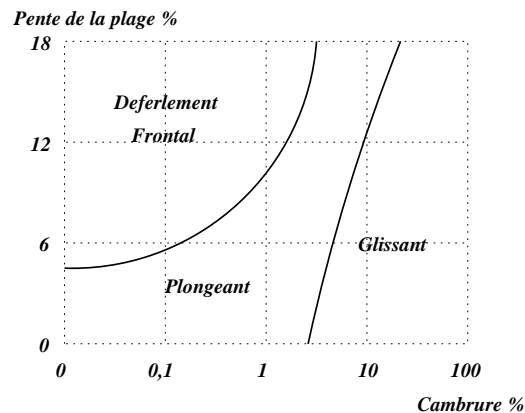


FIG. 14 – Type de déferlement en fonction de la pente du fond marin

De la même manière, on peut déterminer le type de déferlement grâce à un coefficient dit *nombre d'Irribarren* (H et L étant des données prises au large) :

$$I = \frac{\tan(\alpha)}{\sqrt{\frac{H}{L}}}$$

I	0	0,4	2,0	$+\infty$
	Glissant	Plongeant	Frontal	

Finalement, regarder l'angle compris entre le front montant de la vague et le front descendant est un moyen tout aussi efficace pour déterminer le point de départ d'une vague déferlante. Stokes propose comme valeur limite : 120° (fig. 15).

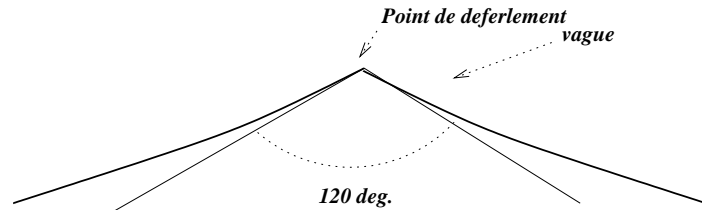


FIG. 15 – Détermination du point de déferlement

1.3.2 Modélisation infographique

Dans cette partie, nous allons nous intéresser aux différents travaux réalisés en infographie pour modéliser l'océan par des équations paramétriques. Malgré des théories approchées de la définition de la surface, l'utilisation des surfaces paramétriques permet d'avoir des générations de modèles 3D en temps réel suffisamment précis. Des techniques classiques de rendu en infographie pourront être rajoutées pour obtenir des images de plus grande qualité.

Modèle sinusoïdal. De loin, la trochoïde du modèle de Gerstner est semblable à une sinusoïde. Cette simplification a été utilisée par Peachey [Pea86] et Ts'o-Barsky [TB87] pour modéliser la surface de l'océan à l'aide de simple fonctions sinusoïdales du type ($\eta = \sin(ax_0 + by_0 - \omega t)$). Ceci permet une modélisation simple à la condition que l'observateur ne soit pas trop proche de la surface à visualiser. La forme de profil de la vague obtenue est alors modélisée par de "vraies fonctions". Ceci qui permet un stockage point par point d'une unique hauteur de vague sur un champ scalaire. D'autre part, cette approche permet de prendre en compte aisément les trains de vagues issus d'ondes de période, d'amplitude et de direction différente. Il suffit de sommer les champs scalaires pour obtenir la surface résultante. Par contre, le déferlement ne peut pas être pris en compte (un champ scalaire ne peut plus être utilisé) et, de plus, il faut vérifier que la somme des amplitudes ne sort pas du domaine autorisé.

Des modèles basés sur les théories de Gerstner et Biesel. Des travaux basés sur les modèles de Gerstner et Biesel ont été utilisés par Fournier-Reeves [FR86] et Imamyia-Zang [IZ95]. Les vagues sont définies par une équation paramétrique qui décrit le trajet des particules d'eau à la surface au cours du temps. Nous rappelons que pour Gerstner, ce trajet est un cercle, et que pour Biesel, c'est une ellipse dont le grand axe s'aligne progressivement avec la pente du sol.

Imamyia et Zang utilisent une équation paramétrique de la forme :

$$\begin{cases} x = x_0 + R \cdot f_1(\alpha) \sin(Kx_0 - \omega t + \beta_1(\alpha)) \\ z = z_0 + R \cdot f_2(\alpha) \cos(Kx_0 - \omega t + \beta_2(\alpha)) + R \cdot f_3(\alpha) \cos(2Kx_0 - 2\omega t) \end{cases} \quad (39)$$

Un algorithme génétique [Esc92] permet de définir les paramètres $f_1, f_2, f_3, \beta_1, \beta_2$ (dépendant de la pente du sol α) des équations de leur modèle afin que la forme de la vague se rapproche au mieux d'une vague déferlante idéale. Cette dernière a pour source une illustration bien connue au Japon (par ex. figure 16).

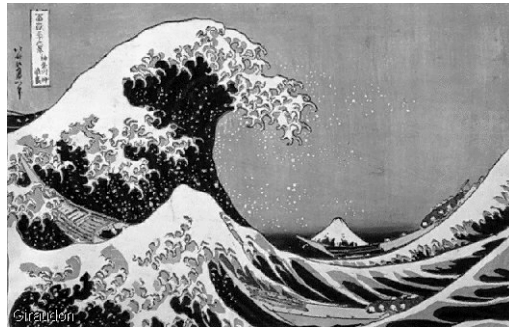


FIG. 16 – Illustration Japonaise

Sous certaines conditions très restrictives, ils obtiennent une forme réaliste de vagues déferlantes. En effet, d'après les auteurs, la pente du sol doit être constante et faible, et la cambrure des vagues très faible. Hélas, le trajet des particules ne correspond plus à un comportement physique correct. En effet, au cours du temps, les particules décrivent un *huit* allongé (fig. 17).

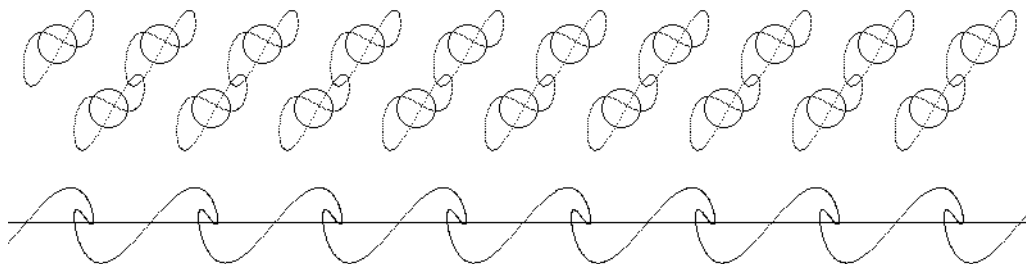


FIG. 17 – Houle et trajet des particules du modèle de Zang

Le modèle de Fournier-Reeves [FR86] est un modèle respectant plus les théories de Gerstner et Biesel. Leur modèle a pour base la théorie de Gerstner et il est ensuite adapté pour "coller" aux travaux de Biesel. Il est la base de notre travail et il sera décrit dans le paragraphe suivant.

1.3.2.1 Le modèle de Fournier-Reeves

Nous rappelons les équations de base de la théorie de Gerstner qu'utilisent Fournier et Reeves :

$$\begin{cases} x = x_0 - R \sin(\Phi) \\ z = z_0 + R \cos(\Phi) \\ \text{avec } \Phi = Kx_0 - \omega t \end{cases} \quad (40)$$

Ce modèle n'est valide qu'à profondeur infinie. De plus, on a vu précédemment que la longueur d'onde ($L = \frac{2\pi}{K}$) varie en fonction de la profondeur h_x par la formule :

$$K_\infty = K \tanh(Kh_x) \quad (41)$$

Afin de tenir compte des variations continues de la profondeur le long de la surface à modéliser, les auteurs ont modifié le calcul de Φ en tenant compte des variations de K depuis le large :

$$\Phi = -\omega t + \int_0^{x_0} K(h(x)) dx \quad (42)$$

avec $K(h(x))$ la valeur de K à la profondeur h au point de coordonnée x .

Pour être appliquée directement dans un modèle numérique, il suffit de discrétiser, par exemple par la méthode de Riemann, cette intégrale. La valeur de Φ devient alors :

$$\Phi = -\omega t + \sum_{i=1}^n K(h(i \frac{x_0}{n})) \frac{x_0}{n} \quad (43)$$

Il faut noter que la valeur de n est telle que le rapport $\frac{x_0}{n}$ reste constant. Ceci devra être vérifié dans la suite de ce chapitre.

Calcul de K en fonction de la profondeur. Ainsi, avec la modification du modèle de Gerstner, les auteurs gardent le caractère continu de la surface de l'océan malgré les variations de profondeurs.

Dans l'état actuel de la présentation du modèle, la longueur d'onde est fonction de la hauteur par $K_\infty = K \tanh(Kh)$. Malheureusement, on veut connaître la valeur de K en tout point et non K_∞ qui est une donnée d'entrée du système : la longueur d'onde du train de vagues prise au large.

Les auteurs effectuent les remarques suivantes :

- lorsque $h \rightarrow 0$, $\tanh(Kh) \rightarrow Kh$ ce qui revient à $K_\infty = K^2 h \Leftrightarrow K = \sqrt{\frac{K_\infty}{h}}$.

– lorsque $h \rightarrow \infty$, $\tanh(Kh) \rightarrow 1$ et donc $K \rightarrow K_\infty$.

Ils proposent alors une valeur approchée à 5% de K : $K(h_x) = \frac{K_\infty}{\sqrt{\tanh(K_\infty h_x)}}$.

La valeur de Φ devient alors :

$$\Phi = -\omega t + \sum_{i=1}^n \frac{K_\infty}{\sqrt{\tanh(K_\infty h(i \frac{x_0}{n}))}} \frac{x_0}{n} \quad (44)$$

Passage au modèle de Biesel. Lorsque la profondeur est trop faible, Biesel propose un modèle transformant progressivement les orbites circulaires des trajectoires des particules en ellipse dont le grand axe tend à s'aligner progressivement avec la pente. Fournier et Reeves proposent une version simplifiée de ceci. Ils évitent alors les paramètres coûteux en temps de calculs et complexes à programmer. Trois coefficients permettant de gérer les effets de la profondeur sur la pente du grand axe (K_0), la longueur du petit axe de l'ellipse (K_z) et enfin la longueur du grand axe (K_x) sont rajoutés. Ces trois paramètres prennent des valeurs comprises entre 0 et 1.

Le modèle final est le suivant :

$$\left\{ \begin{array}{l} x = x_0 + R \cos(\gamma) S_x \sin(\Phi) + R \sin(\gamma) S_z \cos(\Phi) \\ z = z_0 - R \cos(\gamma) S_z \cos(\Phi) + R \sin(\gamma) S_x \sin(\Phi) \\ S_x = \frac{1}{1 - e^{-K_x h}} \\ S_z = S_x (1 - e^{-K_z h}) \\ \sin(\gamma) = \sin(\alpha) e^{-K_0 h} \\ \Phi = -\omega t + \sum_{i=1}^n K(i \frac{x_0}{n}) \frac{x_0}{n} \\ K(x) = \frac{K_\infty}{\sqrt{\tanh(K_\infty h)}} \end{array} \right. \quad (45)$$

où la pente du sol et la pente de l'ellipse sont respectivement α et γ .

Limites du modèle. Le modèle précédent a quelques limites que nous nous efforcerons de contourner dans les paragraphes suivants :

- Les pentes négatives ne sont pas exploitées dans ce modèle. En effet, dans le cas des baïnes, par exemple, les vagues tendent à déferler dans le sens inverse du sens de propagation (troisième crête de la figure 18).
- La manipulation des trois paramètres K_0 , K_x et K_z n'est pas aisée puisqu'ils ne correspondent pas à des paramètres intuitifs. En effet, des valeurs idéales peuvent être trouvées de manière empirique pour un type de sol particulier. Par contre, si, sur une même scène, plusieurs types de fonds marins coexistent, (par exemple, une zone contenant des pentes douces, et des zones avec des pentes fortes), certaines valeurs choisies idéalement pour une zone ne le seront peut-être plus du tout sur la zone voisine.

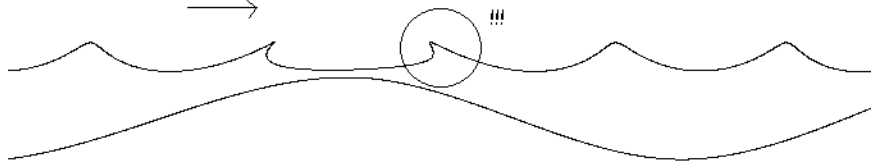


FIG. 18 – Une erreur sur la troisième crête

1.3.3 Notre modèle de houle

1.3.3.1 Quelques modifications du modèle de Fournier-Reeves

Dans l'algorithme de Fournier-Reeves [FR86], le grand axe de l'ellipse tend vers l'infini lorsque la profondeur d'eau devient presque nulle. En effet, la longueur du grand axe de l'ellipse doit être égale à $S_x = \frac{1}{1-e^{-k_x h}}$. Or, dans le modèle de Gerstner, la taille maximale du lieu décrit par les particules à l'intérieur du disque est égal à $R \leq \frac{1}{K}$ (fig. 7) afin de générer une surface correcte. Le rayon du disque ($\frac{1}{K}$) tend vers zéro lorsque la profondeur devient nulle. Dans le cas où la trajectoire des particules quitte le disque, des boucles se produisent à la surface (voir fig. 19). Donc de manière inévitable, dès que la houle atteint une certaine profondeur (à l'approche du rivage), le modèle devient incontrôlable. En effet, suite à notre implémentation du modèle, le trajet théorique des particules d'eau peut être amené à passer sous le niveau du sol (fig. 19).

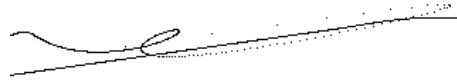


FIG. 19 – Cas limite du modèle de Fournier et Reeves

Aussi, pour “coller” aux limites imposées par le modèle de Gerstner, nous proposons de limiter le trajet des particules d'eau à l'intérieur du disque de rayon $\frac{1}{K}$. Nous calculons, dans un premier temps, la longueur du grand axe ($R \cos(\gamma) S_x$). Nous déduisons la valeur du petit axe S_z à partir du précédent calcul. Nous affectons la valeur limite supérieure du grand axe, si ce dernier dépasse le diamètre du disque, à :

$$S_x = \frac{1}{RK \cos(\gamma)} \quad (46)$$

Ainsi, l'ellipse devient presque plate lorsque la profondeur tend vers zéro. Cette modification nous garantit une arrivée correcte sans boucle ni passage sous le sol. Cette arrivée sur la plage sera ainsi toujours effectuée au niveau de la surface libre de l'océan, soit z_0 . De plus, les trois coefficients K_0 , K_x et K_z peuvent être fixés, car ils permettent de gérer la forme des vagues au moment

où elles échouent sur la plage. Après plusieurs simulations, nous avons choisi $K_0 = 0.1$, $K_x = 0.11$ et $K_z = 0.09$. Des tests ont été effectués sur plusieurs types de fonds (pentes faibles, fortes et combinaison des deux) fabriqués aléatoirement. Un réglage manuel (visuellement satisfaisant) a été effectué pour chaque fond proposé. Les valeurs données précédemment représentent la moyenne des valeurs élue lors de chaque test.

A ce stade de la description, deux problèmes restent à régler :

- Premièrement, l’amplitude des vagues diminue lorsqu’on se rapproche de la plage alors qu’en réalité elle augmente jusqu’au point de déferlement. En effet, comme le rayon du disque tend vers zéro lorsque la profondeur tend elle aussi vers zéro, l’arrivée est effectuée sur la plage avec une amplitude quasi-nulle.
- Deuxièmement, la forme du front de vagues est limitée et ne permet pas de définir des vagues plongeantes (fig 20). En effet, la vitesse des particules de fluide n’est pas suffisamment élevée sur la crête pour simuler les rouleaux et donc générer une vague réaliste. Le front de vagues se limite donc à une parois verticale.

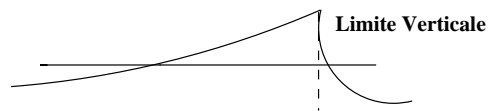


FIG. 20 – Une limite de forme de vague dans notre modèle

Pour compléter notre modèle, nous utilisons trois fonctions empiriques :

- La première permet d’augmenter la vitesse des particules de fluide sur la crête des vagues et donc de l’étirer dans la direction du grand axe de l’ellipse.
- Deux fonctions combinées permettent de simuler l’action de la gravité. La première permet de contrôler la chute des particules et la seconde de définir la direction de cette chute.

Nous aurions pu utiliser la méthode de Stokes pour déterminer le point de déferlement i.e. lorsque l’angle entre la partie montante de la vague et sa partie descendante est supérieure à 120° . Malheureusement, cette méthode est inopérante dans notre cas. En effet, nous devons anticiper le phénomène en étirant la vague avant qu’elle ne commence à déferler. Nous proposons donc de générer le déferlement en fonction de deux paramètres, vus dans les paragraphes suivants, qui donneront plus ou moins d’importance à la profondeur d’eau sur la génération du déferlement.

1.3.3.2 La fonction d’étirement de la vague : *Etirement*

Pour reproduire les théories de Biesel et pour simuler l’accélération des particules sur les crêtes et l’effet local du vent, nous proposons de rajouter à notre premier modèle une fonction d’étirement appelée *Etirement*(Φ). Elle permet d’étirer progressivement la crête dans la direction

du grand axe de l'ellipse. Cette fonction sera contrôlée par deux paramètres : E_{max} définissant la longueur maximale de l'étirement et K_s l'influence de la profondeur sur cet étirement.

Nous avons choisi une parabole pour la fonction $Etirement(\Phi)$ telle qu'il n'y ait aucun étirement dans le creux, et que celui-ci soit maximal sur la crête. Le creux de la vague sera déterminé en recherchant l'ordonnée minimale de l'ellipse par :

$$\Phi_{min} = atan\left(\frac{\sin(\gamma)S_x}{\cos(\gamma)S_z}\right) \quad (47)$$

$$Etirement(\phi, E_{max}) = \frac{1}{\pi^2}(E_{max}\phi^2 - 2E_{max}\Phi_{min}\phi + E_{max}\Phi_{min}^2). \quad (48)$$

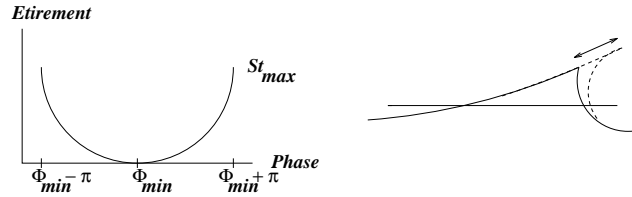


FIG. 21 – a. Fonction Etirement, b. Vue de profil

Pourquoi utilise-t-on une parabole ? D'une part, nous voulons que l'étirement de la vague se fasse de manière progressive et continue du creux vers la crête et vice-versa. Deux demi-droites auraient pu suffir, mais, après quelques tests et observations, nous pensons que l'accélération des particules varie linéairement de la crête vers le creux et inversement. L'utilisation d'une parabole est alors naturelle.

De plus, nous ne voulons pas avoir de modification de la forme de la vague dans le creux. Donc cette parabole passe par le point $(\Phi_{min}, 0)$. Enfin, ceci nous permet de garder un aspect anguleux ou pointu à la crête. L'utilisation d'une fonction sinusoïdale aurait, par exemple, généré une crête arrondie.

1.3.3.3 Les effets de la gravité : Chute & Orientation

La fonction d'étirement n'est pas suffisante pour simuler le phénomène naturel. Il faut prendre en compte les effets de la gravité sur les particules situées sur les crêtes. Nous proposons donc de rajouter une fonction $Orientation(\Phi)$ permettant de basculer progressivement le grand axe de l'ellipse. Cette fonction est combinée à une seconde appelée $Chute(\Phi, A)$ permettant de définir la longueur de la chute des particules dans la direction donnée par la première fonction. Cette chute sera limitée à l'amplitude de la vague. Ainsi, on évitera le croisement entre la crête et le creux. Ces

deux fonctions seront manipulées par un paramètre : K_d qui définit l'influence de la profondeur sur la chute des particules.

Sur le sommet de la crête, le gradient de phase de la fonction décrivant la surface est important. Le gradient de phase de la zone se trouvant juste au dessous de la crête est, au contraire, faible. Donc les deux fonctions ne seront pas symétriques : sur le dessus de la crête, la chute sera très rapide, alors que la reconnexion vers la forme originale (au creux de la vague) s'effectuera beaucoup plus lentement (fig. 22).

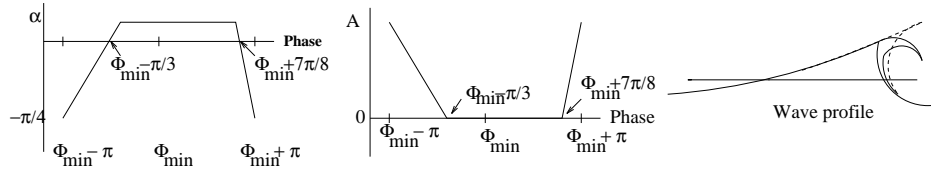


FIG. 22 – a. Fonction Orientation, b. Fonction Displacement, c. Vue de profil

Nous avons choisi de découper ces deux fonctions en trois parties car l'influence de la gravité sur les crêtes ne doit s'effectuer que sur une petite partie de la surface : la crête. Ainsi, la majeure partie de la fonction *Orientation* (resp. *Chute*) est égale aux valeurs prises par les particules dans le modèle original i.e. la pente de l'ellipse γ (resp. *zéro*). Par souci de simplicité, nous avons choisi trois demi-droites.

1.3.3.4 Les pentes négatives

L'algorithme de Fournier et Reeves peut entraîner, dans le cas de pentes négatives, des vagues déferlantes dans le sens inverse à la progression de la vague. En fait, le grand axe de l'ellipse suit la pente du sol et donc se trouve inversé lorsque la pente est négative. Ceci n'est valable qu'à faible profondeur puisqu'à profondeur moyenne le sol n'a que peu d'influence sur la forme des vagues et la pente du grand axe de l'ellipse.

Une première idée serait de limiter l'angle de l'ellipse à un angle positif. Ceci permettrait de ne plus avoir de vagues déferlant dans le sens inverse à la propagation. Cette solution limite donc la vague à une cycloïde. De plus, une vague déferlant au niveau d'une bosse continuera dans le creux.

Pour pouvoir faire déferler une vague dans le sens de propagation, en utilisant les fonctions *Etirement*, *Orientation* et *Chute*, la pente de l'ellipse doit être positive. Nous décidons donc de prendre la valeur absolue de la pente comme référence. Ainsi, une vague, grâce à l'accumulation d'énergie prise sur la bosse, pourra déferler dans le creux. (fig. 23).

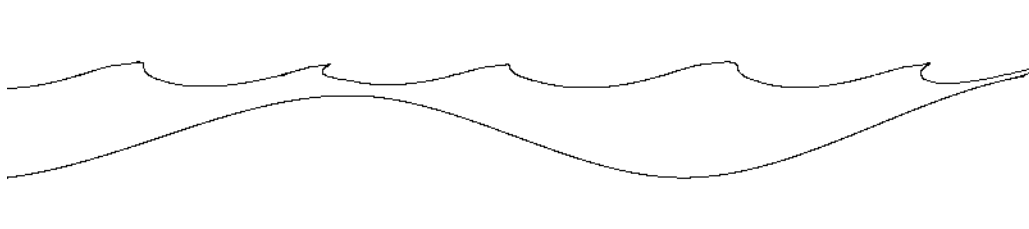


FIG. 23 – Vue de profil d’une onde avec une pente négative. La vague continue sa progression dans le sens du train d’onde

1.3.3.5 Les équations de notre modèle de houle

$$\begin{cases} x = x_0 + \left[R\tau'_\gamma S_x \sin(\Phi) + R\tau_\gamma S_z \cos(\Phi) \right] + Etirement(\Phi, E_{max})\tau'_\gamma e^{-K_s h} \\ \quad + Chute(\Phi, A) \cos(Orientation(\Phi, \gamma)) e^{-K_d h} \\ z = z_0 - \left[R\tau'_\gamma S_z \cos(\Phi) + R\tau_\gamma S_x \sin(\Phi) \right] + Etirement(\Phi, E_{max})\tau_\gamma e^{-K_s h} \\ \quad + Chute(\Phi, A) \sin(Orientation(\Phi, \gamma)) e^{-K_d h} \end{cases} \quad (49)$$

avec

$$\tau_\gamma = |\sin(\alpha)| e^{-0.1h}, \quad \tau'_\gamma = \sqrt{1 - \tau_\gamma} \quad (\gamma \text{ est la pente du sol}).$$

$$S_x = \frac{1}{1 - e^{-0.11h}} \quad (\text{Augmentation du grand axe})$$

$$S_z = S_x(1 - e^{-0.09h}) \quad (\text{Diminution du petit axe})$$

$$\Phi = -\omega t + \sum_{i=1}^n K \left(i \frac{x_0}{n} \right) \frac{x_0}{n}.$$

$$\text{De plus, } S_x \leq \frac{1}{RK \cos(\gamma)}.$$

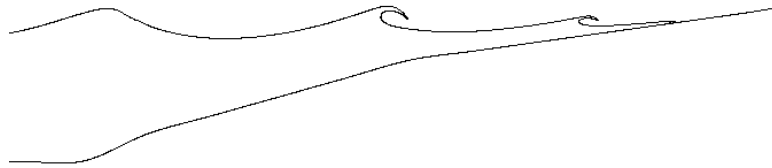


FIG. 24 – Un exemple final

1.3.4 Bilan

Nous venons de proposer un modèle 2D de vagues contrôlé par deux paramètres :

- K_s permettant de modifier l’influence du sol sur l’étirement de la vague sur les crêtes.
- K_d permettant de modifier l’influence de la profondeur sur la simulation de la force de gravité sur les crêtes.

La figure 24 présente un exemple de forme de vagues que l’on peut obtenir avec notre modèle.

Une limite, qui est commune à tous les modèles de vagues modélisés par des équations paramétriques est que les vagues déferlantes ne peuvent pas être modélisées intégralement. En effet, seules les équations de Navier-Stokes permettent de capturer ce phénomène entièrement. Nous nous limitons donc à la génération des vagues plongeantes jusqu'au point de contact entre la crête de la vague et le creux suivant.

Chapitre 2

Influence du sol et de la côte

2.1 Introduction

Après avoir proposé un modèle de la forme des vagues en coupe, nous allons nous intéresser au mouvement global de ces vagues à l'approche des côtes¹ : le déplacement et la déformation des trains de vagues. Un train de vagues peut être défini comme l'ensemble des ondulations provenant d'un même point de l'océan et ayant une même direction initiale de propagation. Une approche efficace pour l'étude de la nature ondulatoire de ceux-ci consiste non pas à considérer le déplacement des crêtes des différentes vagues mais à appréhender la modélisation par le biais des orthogonales à ces crêtes. Cela présente en plus l'intérêt de pouvoir "recoller" avec l'étude en coupe de la forme des vagues que l'on pourra superposer sur ces orthogonales. Ces dernières subissent, entre autres, trois phénomènes : la réfraction, la diffraction et la réflexion. Ces phénomènes sont tout à fait analogues à ceux régissant la propagation de la lumière. Ici, l'indice de réfraction des milieux traversés sera remplacé par la vitesse de propagation de la vague².

Des trois phénomènes précédents, c'est la réfraction qui est la plus manifeste. Il s'agit de la déviation des orthogonales au front de vagues en fonction du relief qui fait varier la vitesse de phase de l'onde. Ainsi, l'évolution du sol agit de manière similaire à des variations continues d'indice de réfraction lors de la propagation d'ondes lumineuses. Du point de vue visuel, c'est la réfraction qui explique la déformation des trains de vagues à l'approche des côtes (fig 25).

Les deux autres phénomènes (réflexion et diffraction) sont essentiels lorsque les trains de vagues rencontrent des obstacles émergés accompagnés d'une rupture brutale de la pente du sol. Ceci est le cas lorsque, par exemple, le train de vagues rencontre une digue à l'entrée d'un port. La

¹Pour cela, nous utiliserons une vue de dessus de l'océan.

²L'indice de réfraction est le rapport entre la vitesse de la lumière dans le milieu traversé et sa vitesse dans le vide. Dans le cas des vagues, il n'y a pas de référent normalisateur comme l'est la vitesse de la lumière dans le vide. Nous utilisons donc la vitesse des vagues au large comme référent

simulation de ces phénomènes devient donc un atout pour une conception d'ouvrages portuaires.

La réflexion est un phénomène simple à envisager. Une vague heurtant violemment un obstacle "rebondit" sur celui-ci. L'obstacle absorbe alors une partie de l'énergie de la vague. La loi régissant la réflexion des orthogonales au front de vagues est identique à celle de la réflexion optique. L'angle de réflexion est égal à l'angle d'incidence. Si l'obstacle est perpendiculaire au plan moyen de l'océan, la réflexion est spéculaire car la dissipation d'énergie dans l'obstacle est négligeable.

La diffraction des vagues est le phénomène visible derrière une digue. Imaginons un train de vagues parallèle à une digue rectiligne. Une partie de celui-ci se réfléchit. De plus, à partir du bout de la digue, on constate l'apparition de vagues situés derrière cet obstacle. Elles ont une forme circulaire. Elles sont en fait issues de la diffraction de la vague incidente née à l'extrémité de la digue aussi appelé musoir.

Après un exposé plus détaillé de ces trois phénomènes, nous décrivons nos travaux conduisant à des algorithmes permettant leur simulation.

2.2 La réfraction des vagues

2.2.1 Modélisation physique

A l'approche d'une côte, les vagues atteignent des eaux peu profondes. Les fronts de vagues sont alors ralentis et leur longueur d'onde décroît. Ceci est dû aux frottements de l'eau sur le fond marin. Ces frottements s'amplifient avec la diminution de la profondeur ce qui explique que, sur un sol non-uniforme, les trains de vagues sont déformés. D'autre part, la réfraction induit aussi des variations d'amplitude de la vague. Lorsque les orthogonales convergent, il y a concentration de masse d'eau et l'amplitude augmente. Lorsqu'elles s'écartent, la masse d'eau est alors répartie sur une plus grande surface de sol et l'amplitude diminue. C'est le phénomène de réfraction (fig. 25) [Kin65][Lac65].

Pour une bonne simulation, l'étude de la réfraction doit être fine. Le relief du sol devra être modélisé de manière précise. De petites irrégularités du fond peuvent avoir des conséquences importantes voire dramatiques comme dans l'anecdote suivante : *"Ainsi, la jetée de Long Beach en Californie, qui avait résisté à l'assaut des vagues durant des années, fut partiellement détruite lors d'une banale tempête en 1930. La jetée fut réparée, mais en 1939, les vagues la rompirent à nouveau. Une étude de la réfraction révéla qu'une houle de longue période, provenant du Sud-Sud-Est et présente seulement durant ces deux tempêtes, avait été concentrée sur la brèche par une petite bosse qui se trouvait, à 80 mètres de profondeur, à plus de 11 km au large. La bosse avait agi comme une lentille, en multipliant la hauteur des vagues par 3.5 environ à l'endroit des*



FIG. 25 – Réfraction des vagues dans une baie

dégâts.”[PLS98]

Pour évaluer la réfraction, les physico-mécaniciens utilisent deux grandes méthodes : Le tracé des plans de vagues développé par Huyghens et le tracé des orthogonales aux fronts de vagues de Johnson, O’Brien et Isaacs [Lac65]. La première méthode positionne graphiquement les crêtes des trains de vagues en fonction des lignes de crêtes précédentes. La seconde utilise une génération de la réfraction à deux échelles des orthogonales au front de vagues et n’est utilisée que pour les grandes étendues.

2.2.1.1 Tracé des plans de vagues

La méthode proposée par Huyghens est la suivante : partant d’une ligne de crête initiale, il propose de retrouver les lignes de crête suivantes. En tout point P_i d’une ligne, le calcul de la vitesse C_i de l’onde en fonction de la profondeur h_i et de la période T est effectué. Le point P'_i situé sur la crête suivante se trouve donc à une distance $C_i T$ de P_i . On peut donc, de proche en proche, tracer la ligne de crête suivante qui correspond à l’enveloppe des circonférences centrées sur la ligne de crête initiale et de rayons $C_i T$ (fig. 26).

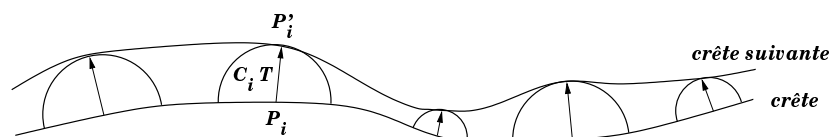


FIG. 26 – Tracé de front d’ondes par la méthode de Huyghens

2.2.1.2 Tracé des orthogonales au front de vagues

Cette méthode permet de contrôler les effets de la réfraction à deux niveaux. Un premier calcul est effectué à grande échelle suivant un plan dit d'approche permettant d'avoir une idée globale de la propagation de la houle depuis le large jusqu'au voisinage de la côte. Un deuxième plan, dit local, peut-être utilisé pour obtenir plus d'informations sur les crêtes de houle obtenues à partir du plan d'approche.

La déviation des orthogonales aux fronts de vagues est fonction de la vitesse de propagation de l'onde et donc fonction de la profondeur (fig 28). La variation de la vitesse est donnée par l'équation 8. La déviation, obtenue à partir des valeurs précédentes, est fonction de la loi de Descartes :

$$\frac{\sin(i_1)}{C_1} = \frac{\sin(i_2)}{C_2} \quad (50)$$

où C_1 et C_2 sont les vitesses de l'onde avant et après la courbe de niveau. i_1 (resp. i_2) est l'angle d'incidence de l'orthogonale sur la courbe de niveau (resp. l'angle de réfraction)

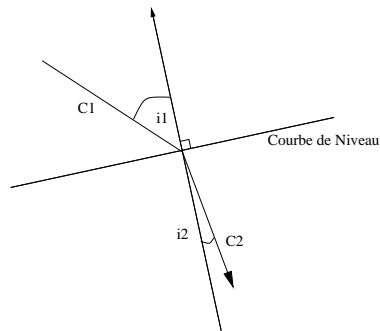


FIG. 27 – Loi de la réfraction

Calcul de l'amplitude des vagues en fonction de la réfraction. La méthode précédente rend possible un calcul aisé de l'amplitude de la houle en fonction de la réfraction en tout point de la scène. L'énergie transmise entre deux orthogonales de distance D_∞ est constante tout au long de leur trajectoire. Si elles ne subissent pas d'autres phénomènes tels que la réflexion ou la diffraction, le volume d'eau compris entre deux orthogonales est, par conséquent, constant (conservation de l'énergie et incompressibilité du fluide). Ceci influence directement l'amplitude qui varie en raison de l'inverse de la racine carrée de la distance entre les orthogonales [TB87] :

$$H_1 = H_\infty \sqrt{\frac{D_\infty}{D_1}} \quad (51)$$

où H_∞ (resp. H_1) l'amplitude de la houle prise au large (resp. au point de calcul).

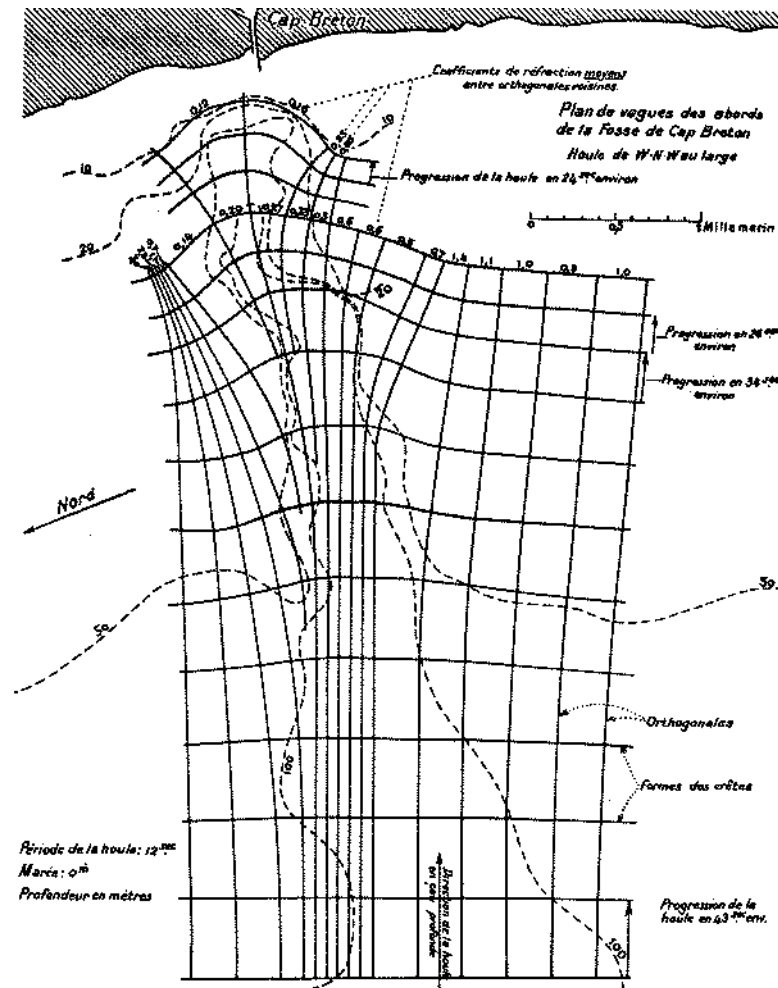


FIG. 28 – Tracé des orthogonales au front de vagues (d'après [Lac65])

2.2.2 Réfraction et infographie

La réfraction de Fournier-Reeves et Peachey. Dans ces deux travaux ([FR86] [Pea86]), la réfraction n'est appréhendée que de manière uni-directionnelle. Ceci implique, par exemple, que le déferlement des vagues se produit toujours dans la même direction, celle des orthogonales situées au large. Cette approche n'est pas suffisante pour modéliser la progression de l'eau dans la figure 30.

Cette figure représente une baie vue de dessus et les orthogonales aux fronts de vagues qui se propagent dans la scène. Premièrement, la connaissance l'état de la surface à l'intérieur de la baie est inconnu. Deuxièmement, les vagues vont toutes s'échouer dans le même sens (horizontale ici). Ceci n'est pas vrai dans la réalité puisque les vagues auront tendances à s'aligner avec le rivage et donc les orthogonales aux fronts de vagues doivent tendre à être perpendiculaire avec la côte.

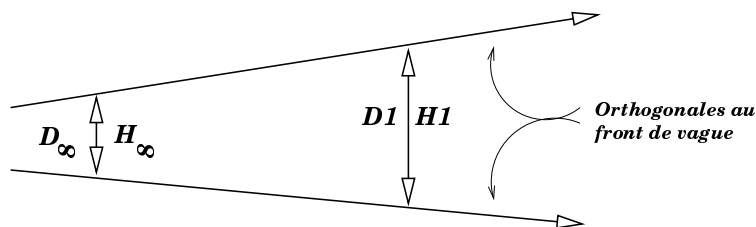


FIG. 29 – Calcul de l’amplitude en fonction de la réfraction

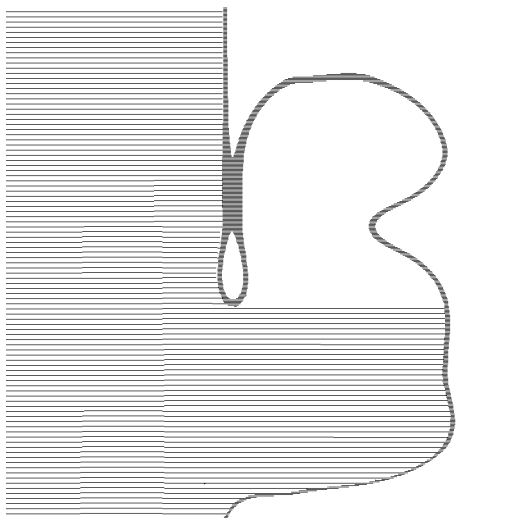


FIG. 30 – Calcul de la réfraction avec le modèle de Fournier-Reeves

La réfraction de Ts’o-Barsky. L’année suivante, Ts’o et Barsky [TB87] utilisent pour la première fois la loi de Descartes pour modéliser la réfraction. Ces travaux ont été repris par Imamiya et Zang en 1995 [IZ95]. Les premiers auteurs ont développé un algorithme qu’ils appellent “*Wave Tracing*”. Le fond marin est modélisé, pour leur méthode, à l’aide de courbe de niveaux sur lequel est posé une grille 2D uniforme. A chaque franchissement de courbe de niveau, les orthogonales aux fronts de vagues sont déviées. La progression des orthogonales dans la grille s’effectue par la méthode de Bresenham et en chaque sommet rencontré, l’amplitude est calculée.

Cette approche comporte trois limites importantes. La première est que la hauteur d’eau est stockée dans un champ scalaire. Ceci interdit les vagues déferlentes nécessitant au moins trois hauteurs d’eau à stocker (fig 31).

D’autre part, la réfraction n’étant calculée que sur les orthogonales au front de vagues et au moment du franchissement des lignes de niveaux, la réfraction peut être alors incomplètement modélisée. C’est le cas en particulier lorsque les trains de vagues heurtent une île ou lorsque l’on veut suivre l’écoulement de l’eau de manière précise à l’intérieur d’une baie (fig. 32).

Finalement, les courbes de niveaux étant décrites par un ensemble de splines indépendantes les

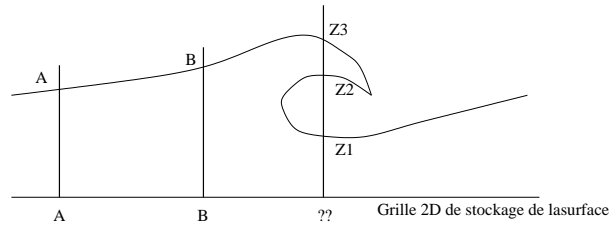


FIG. 31 – Stockage d’une vague déferlente

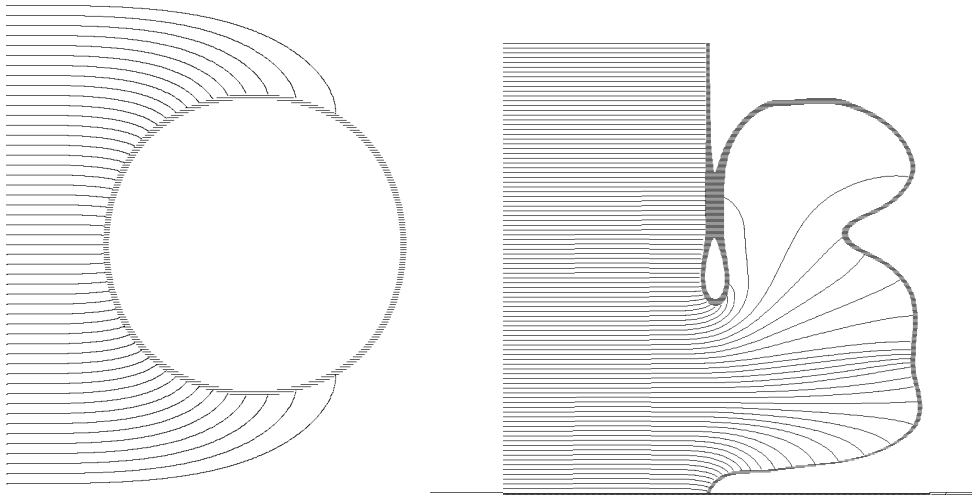


FIG. 32 – Le “Wave-tracing” de Ts’o sur deux exemples : a. une île, b. une baie

unes des autres, il n’est pas possible de calculer de manière efficace la profondeur et la courbe de niveau passant en n’importe quel point de la scène (point M fig. 33), en particulier par les sommets de la grille de stockage.

2.2.3 Notre modèle de Réfraction : Le Lancer de Vagues Adaptatif

2.2.3.1 Principes

Le modèle que nous proposons³ est une version adaptative de “Wave Tracing” de Ts’o et Barsky. Il généralise l’approche utilisée par Johnson, O’Brien et Isaacs [Lac65].

Le principe est simple : dès que deux orthogonales consécutives, que l’on nommera aussi rayons, s’écartent trop l’une de l’autre, nous générons une nouvelle orthogonale entre ces deux dernières. Ceci nous permet de garder une précision (choisie par l’utilisateur) constante sur toute la scène. Pour un bon contrôle de l’algorithme, deux problèmes sont à résoudre. Le premier est le contrôle du nombre de rayons générés. Le second est l’arrêt de la propagation des orthogonales. Ce

³Méthode présentée aussi dans [GS97].

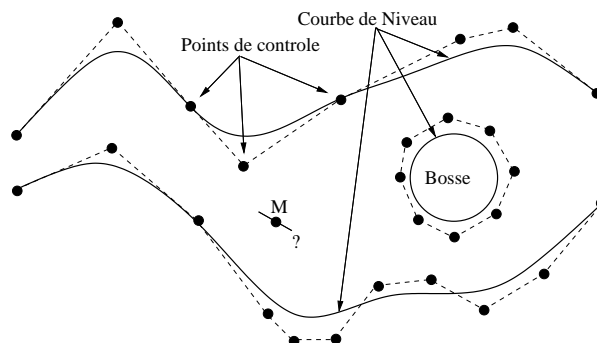


FIG. 33 – Définition du sol par des splines (vue de dessus)

dernier problème peut apparaître, par exemple, autour d'une île où certains rayons peuvent adopter une trajectoire fermée et donc progresser de manière infinie. Nous utilisons alors l'amplitude qui décroît nécessairement avec le temps dans ce cas de figure. Il nous suffit donc de stopper les orthogonales lorsque l'amplitude devient inférieure à un certain seuil. Ceci étant résolu, le nombre de rayons devient borné puisque notre scène est finie, écartant ainsi le premier problème.

2.2.3.2 Le calcul de la profondeur en tout point du fond marin

Pour calculer la déviation de chaque rayon en fonction des phénomènes de réfraction, il faut connaître la configuration du terrain en tout point de la scène. Nous devons donc définir le sol de la manière la plus précise possible.

Bien que cela soit un outil classique en modélisation des terrains, nous ne voulons pas définir le sol à l'aide de courbes splines représentant les différentes courbes de niveaux pour les raisons évoquées précédemment.

Nous décidons de définir notre fond marin par une grille régulière ($M \times N$). Cette grille représente la bathymétrie du sol par une vue de dessus. Chaque noeud de la grille représente une profondeur. Les valeurs de la grille peuvent être obtenues de plusieurs manières : soit par une saisie lourde et fastidieuse en chaque noeud, soit par la génération automatique d'un terrain (un terrain fractal par exemple), ou par des données satellites⁴ etc...

Tout point se trouvant dans une maille de la grille, une profondeur peut être calculée par interpolation. Pour des raisons de compatibilités avec le modèle 2D, nous éliminons l'interpolation linéaire qui génère une courbe C^0 et induit des discontinuités à la surface. Nous utilisons une interpolation en spline-cardinale, générant une courbe C^1 (fig. 34) :

⁴Des données devraient nous être prochainement fournies par l'équipe d'océanographie de l'université de Bordeaux

$$H_M = \frac{1}{2} \{ 2b + (c - a)t + (2a - 5b + 4c - d)t^2 + (3(b - c) - a + d)t^3 \} \quad (52)$$

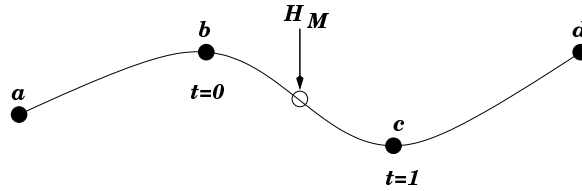


FIG. 34 – Interpolation Spline-Cardinale 2D

L'interpolation spline-cardinale en 3D nécessite les 16 valeurs voisines du point courant d'échantillonnage. Cette fonction est séparable. Le calcul se fait donc en deux passes : tout d'abord, les quatre valeurs voisines situées sur la même ordonnée que le point courant. Enfin, la valeur de la profondeur au point d'échantillonnage est déduite par interpolation avec les quatre dernières valeurs trouvées (fig 35).

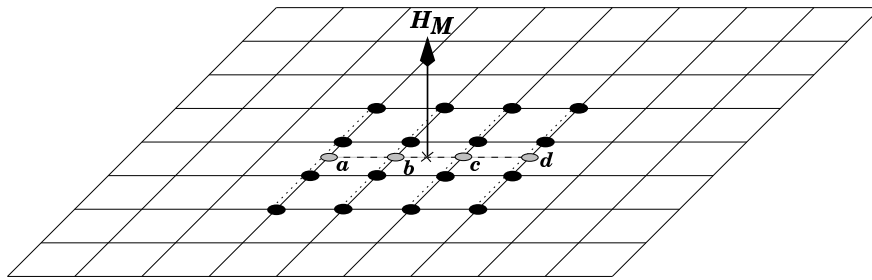


FIG. 35 – Interpolation Cardinale-Spline 3D

2.2.3.3 Calcul de la déviation

Les différentes courbes de niveaux n'étant pas a priori connues, il faut retrouver la configuration locale du terrain en chaque point courant d'échantillonnage pour déterminer la déviation des rayons due à la réfraction.

Pour chaque point M de la scène où la courbe de niveau locale est à déterminer, nous choisissons trois points A, B et C situés à une distance arbitraire δd de celui-ci. Par approximation, la facette décrite par ces trois points, détermine le plan local dans lequel se trouve le point M (fig 36).

Ensuite, nous trions les points A, B et C de telle sorte que seule la profondeur en A soit la seule supérieure à celle de M , ou que la profondeur en A soit la seule inférieure à M . Ainsi, il ne reste plus qu'à trouver pour chacune des arêtes AB et AC les points de profondeur identique à M . Ceci

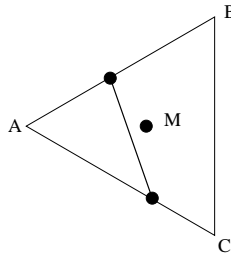


FIG. 36 – Calcul de la courbe de niveau (vue de dessus)

est fait par interpolation linéaire suivant chacun des segments. Le segment de droite passant par ces deux derniers points est considéré comme la courbe de niveau locale passant par M . L'erreur commise est d'autant plus fine que le δd sera choisi petit.

Un cas particulier est possible lorsque M domine ou est dominé par les trois points A, B, ou C ($(h_M > h_A, h_B, h_C)$ ou $(h_M < h_A, h_B, h_C)$). Dans ce cas, la courbe de niveau passant par M est réduite à un unique point. Le rayon n'est alors pas dévié. Pour des raisons techniques, ceci revient à considérer une courbe de niveau fictive perpendiculaire au rayon.

La figure 37.a schématise la méthode de T'so et Barsky pour calculer la déviation des rayons. La figure 37.b schématise notre méthode. Notre approche est un peu plus coûteuse. Par contre elle est beaucoup plus précise puisque T'so et Barsky n'évaluent la déviation des rayons qu'au niveau de chaque courbe de niveau traversée alors que nous le faisons à chaque pas de progression des rayons.

2.2.3.4 Principe général de la progression des rayons

Cette présentation passe nécessairement par une description détaillée de la vie des rayons (naissance, vie et mort) durant le processus de génération de la surface.

Rayons Initiaux. Les rayons sont générés à partir du large. Les rayons progressent en parallèle tant qu'ils ne subissent pas de réfraction (ni d'ailleurs de diffraction et réflexion). A chaque pas de calcul, tous les rayons auront progressé. Ceci est une première différence importante avec le "Wave tracing" de T'so où les rayons progressent indépendamment les uns des autres.

Création de nouveaux rayons. Dès que la distance entre deux rayons est supérieure à un seuil fixé par l'utilisateur, un nouveau rayon est créé. Ce seuil vaut a priori le double de la distance entre deux rayons initiaux mais peut être modifié en fonction des besoins de l'utilisateur. Les propriétés initiales (amplitude, phase, vecteur vitesse, direction de propagation, ...) sont obtenues par interpolation linéaire avec celles de ses deux voisins (cas 1 fig 38).

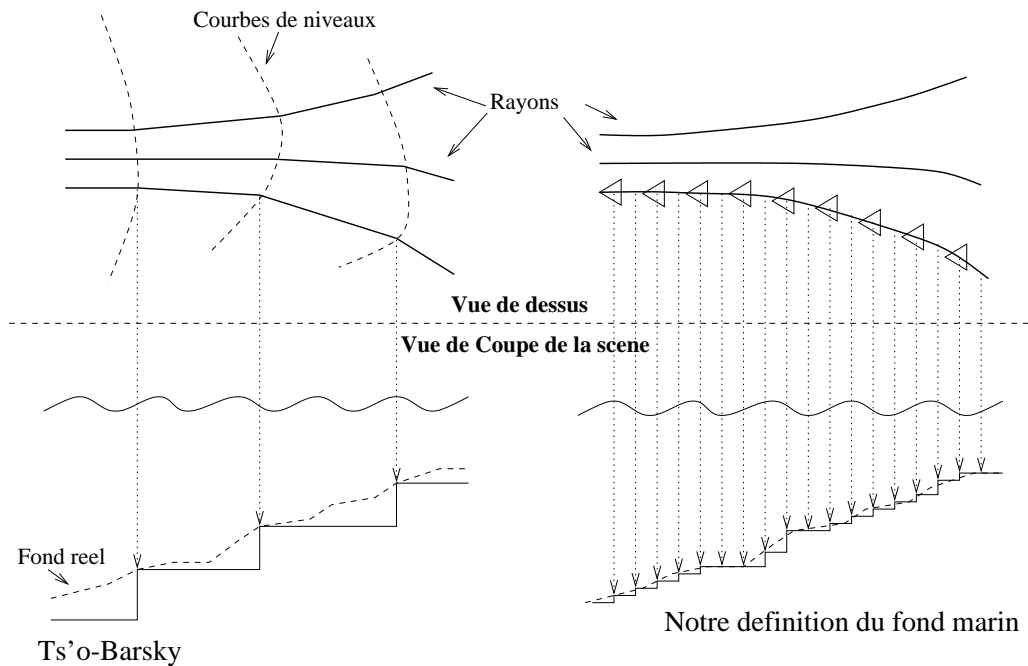


FIG. 37 – Méthode de calcul des facteurs de réfraction

Changements d'état des rayons. Après un pas de progression, plusieurs situations peuvent modifier l'état des rayons. Cette notion d'états, purement algorithmique, correspond à un drapeau pouvant prendre trois valeurs différentes (*DEAD*, *AGROUND* et *FLAT*) expliqués ci-dessous :

- Un rayon sortant de la scène est marqué *DEAD* et plus aucun calculs ne le concernera. Il sera ensuite automatiquement détruit.
- Un rayon touchant la plage devient *AGROUND*. Au lieu d'être détruit ou stoppé, ce rayon suivra le rivage. Etant donné qu'une majorité de rayons va toucher la plage, ceux-ci auront des trajectoires superposées. Evidemment, cette situation n'est pas admissible et lorsque deux rayons voisins sont marqués *AGROUND* et progressent dans la même direction, seul celui couvrant l'autre est conservé. Le rayon couvert est alors détruit après un marquage à *DEAD*.

Le rayon qui arrive sur le bord de la plage pour suivre le rivage dans deux directions possibles : soit celle qui suit le prolongement de la direction d'arrivée du rayon, soit l'opposée. Nous choisissons de faire progresser le rayon dans le même sens que son arrivée. Cette direction se calcule simplement par :

```
VECTEUR quel_cote_de_cote(VECTEUR Vrayon, VECTEUR Cote)
Debut
    Si (Produit_Scalaire(Vrayon, Cote) < 0)
        retourner Cote
```

Sinon

retourner Inverse(Cote)

Fin

où V_{rayon} est le vecteur directeur incident du rayon qui heurte la côte et $Cote$ le vecteur qui suit la courbe de niveau de la plage.

Certains rayons arrivant sur la côte peuvent être déviés par réfraction de telle sorte qu'ils s'éloignent de celle-ci. C'est là qu'interviennent les rayons *AGROUND* puisqu'ils vont permettre la création de nouveaux rayons le long de la côte (fig. 38).

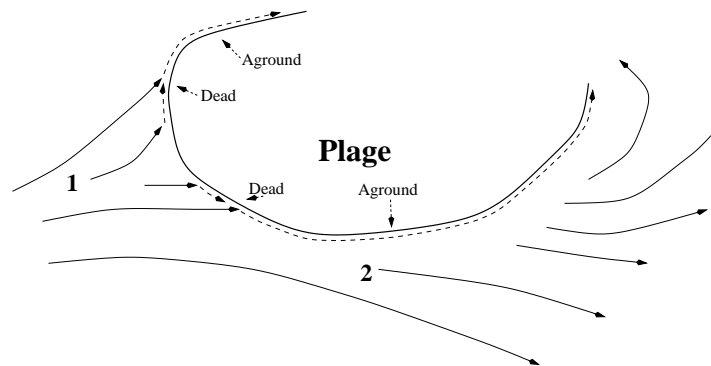


FIG. 38 – Création de nouveaux rayons

- L'amplitude d'un rayon peut devenir non significative en dessous d'une certaine valeur. un tel rayon est alors marqué *FLAT*. Ces rayons ne sont pas détruits car ils peuvent jouer un rôle analogue aux rayons *AGROUND* pour le maintien de la précision lorsqu'ils sont voisins d'un rayon ni *FLAT* ni *AGROUND*. Lorsque deux rayons voisins sont marqués *FLAT*, aucune création n'est effectuée quelle que soit la distance entre ceux-ci. En fait, la petite ondulation créée entre ces deux rayons pourra être facilement reproduite par des techniques classiques de rendu que nous commenterons dans le chapitre suivant.
- Finalement, une gestion particulière du bord de scène doit être effectuée. Pour des raisons similaires aux rayons marqués *AGROUND*, il faut pouvoir maintenir la précision en bord de scène. Ceci s'effectue en créant de nouveaux rayons entre ce bord de scène et le rayon le plus proche de ce point. Les valeurs initiales de ces nouveaux rayons ne peuvent pas être déduites directement de leurs voisines. Nous décidons donc d'y affecter les valeurs de progression du rayon le plus proche (points A et B fig 39).

Fin de la progression. L'algorithme se termine lorsque tous les rayons restant dans la scène sont soit *DEAD*, soit *FLAT*, soit *AGROUND*.

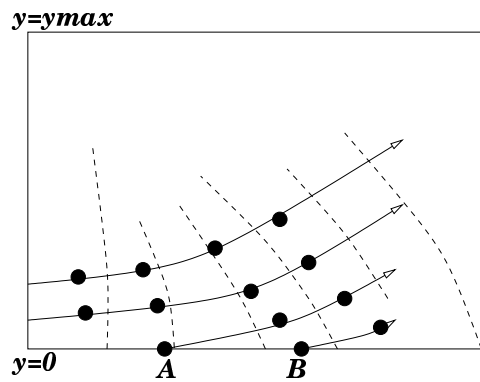


FIG. 39 – Création de nouveaux rayons en bord de scène (vue de dessus)

2.2.3.5 La calcul de l'amplitude

Comme nous l'avons vu, la réfraction influence l'amplitude des vagues. Si deux rayons divergent en raison de la réfraction, leur amplitude diminue et réciproquement, elle augmente lorsqu'ils convergent. Le calcul de l'amplitude de la houle est donc fonction de la distance entre ces deux rayons. On note D_∞ la distance initiale (au large) et D la distance courante. L'amplitude est donnée par la formule :

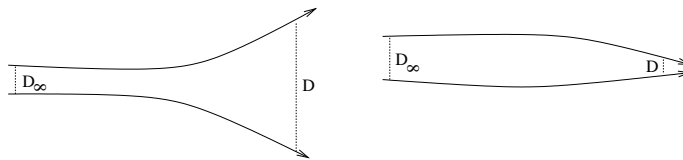


FIG. 40 – Calcul de l'amplitude suite à la réfraction

$$H = H_\infty \sqrt{\frac{D_\infty}{D}} \quad (53)$$

En effet, la caractéristique d'une houle est donnée par :

$$H^2 D = \text{constante} \quad (54)$$

Les rayons progressent pas à pas dans la scène et pour limiter les coûts mémoire, nous ne stockons que la position courante et la précédente de chaque rayon. Il faut donc adapter le calcul de l'amplitude à notre modèle. Ceci est possible car ce calcul peut être effectué localement en comparant les distances entre les rayons à deux pas successifs (D_1 et D_2 fig. 41).

Nous appliquerons donc la loi :

$$H_2 = H_1 \sqrt{\frac{D_1}{D_2}} \quad (55)$$

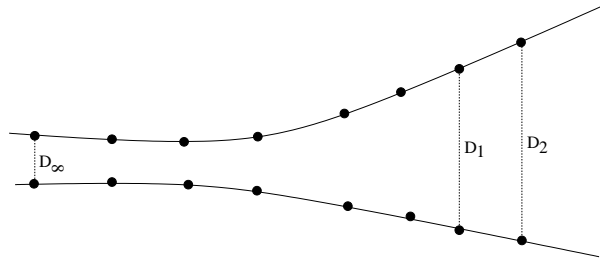


FIG. 41 – Calcul de l'amplitude

où H_1 et H_2 sont les amplitudes de la houle au deux points de calculs.

En effet :

$$\begin{cases} H_2 = H_\infty \sqrt{\frac{D_\infty}{D_2}} \\ H_1 = H_\infty \sqrt{\frac{D_\infty}{D_1}} \end{cases} \Leftrightarrow \frac{H_2}{H_1} = \frac{\sqrt{\frac{D_\infty}{D_2}}}{\sqrt{\frac{D_\infty}{D_1}}} \Leftrightarrow H_2 = H_1 \sqrt{\frac{D_1}{D_2}} \quad (56)$$

Il reste alors le cas à traiter du calcul de l'amplitude lors de la génération d'un nouveau rayon. Dans ce cas, $H_2 = H_3 = H_1 \sqrt{\frac{D_1}{D_2+D_3}}$ (fig 42). Au pas suivant, on utilise la méthode précédemment vue.

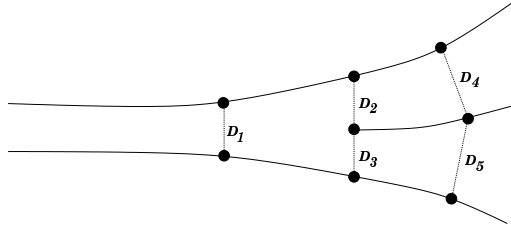


FIG. 42 – Calcul de l'amplitude dans le cas d'un nouveau rayon

Une fois l'amplitude déterminée, le calcul du niveau de la surface par rapport à la surface libre peut être effectué en appliquant le modèle 2D sur chaque rayon (§1.3.3).

2.2.4 Résultats

Nous avons présenté figure 32 deux scènes avec l'algorithme de Ts'o-Barsky. Les mêmes scènes sont générées avec notre algorithme figure 43. Une vue de dessus est proposée figure 44 pour donner un aperçu de l'état de la mer sur ces mêmes scènes. Sur cette dernière figure, nous pouvons remarquer une zone foncée à droite de l'île où des rayons se superposent. La gestion de ce cas sera présenté lors de l'étape de facettisation §2.5. Les parties sombres, sur les figures, représentent les crêtes des vagues. Les différences d'amplitude ne sont pas visibles avec la vue de dessus. Ceci sera imperceptible après la phase de rendu.

La scène de la baie mesure 100 mètres sur 100 mètres. nous avons lancé un rayon tous les mètres, soit 100 rayons au départ et le pas de progression des rayons était de 0,4 mètres. Nous avons créé un nouveau rayon dès que la distance entre ces deux-ci était supérieure à 1,2 mètres. Notre algorithme a créé 209 nouveaux rayons et 39097 facettes. Il s'exécuta pendant environ 2 secondes sur une station de travail Silicon Graphics Onyx, temps de facettisation compris.

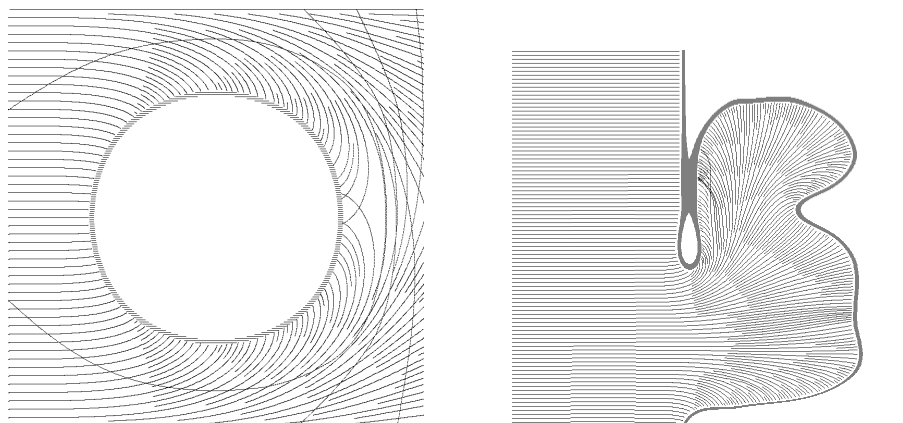


FIG. 43 – Notre algorithme : a. une île, b. une baie

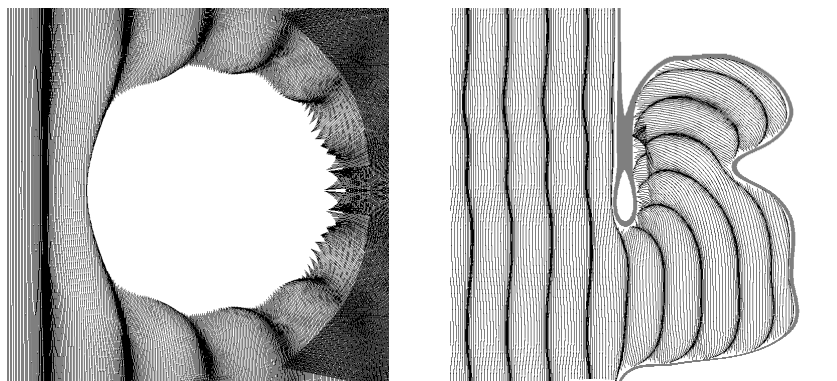


FIG. 44 – Une vue de dessus calculée des mêmes scènes

2.3 La Diffraction des vagues

2.3.1 Un modèle physique

En l'absence de réfraction, la houle se propage de manière rectiligne. Lorsqu'elle heurte une digue, elle est réfléchi. La partie évitant la digue et se trouvant à une distance raisonnable de la digue continue sa propagation rectiligne. Donc sans diffraction, aucune agitation (ou vague) ne

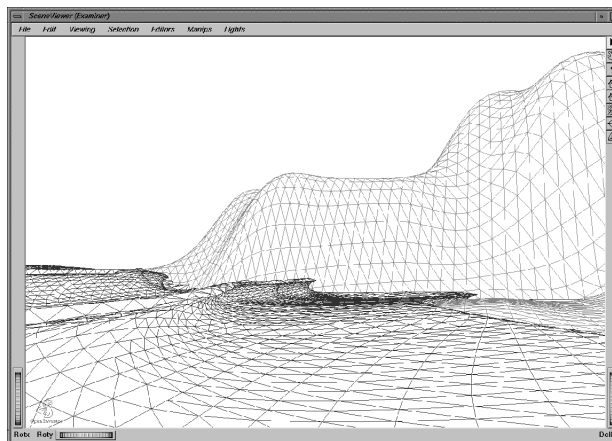


FIG. 45 – Vue 3D facettisée d’une scène avec déferlement

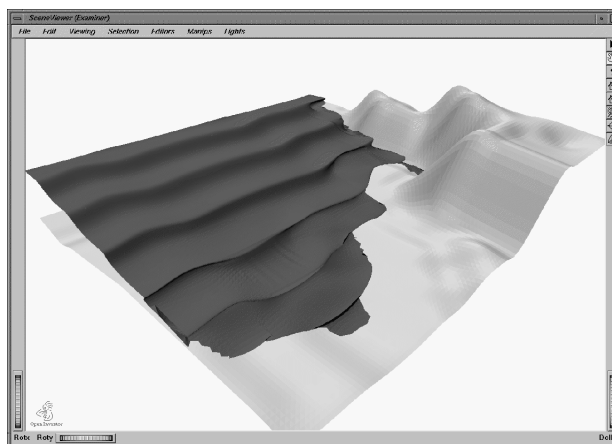


FIG. 46 – Vue 3D d’une scène océane grâce à Open Inventor

devrait être perceptible derrière la digue (fig 47). Or, ce n’est pas le cas. Tout se passe comme si l’extrémité de cette digue appelée musoir était un générateur d’ondes. Ces dernières situées derrière la digue sont de forme circulaire et leur amplitude dépend de celle de la houle incidente et de leur distance au musoir. Dans cette partie, les ondes sont amorties et les crêtes des vagues diffractées prolongent de manière continue celles des trains d’ondes incidents (fig. 48). Donc, la difficulté principale de la modélisation de ces ondes diffractées est le calcul de l’amplitude. Parmi les nombreuses méthodes proposées pour ce calcul dans la littérature [Lac65][Kin65][Bon80], nous ne présentons que celle exposée par Larras [Lac65] ⁵.

Le modèle de Larras est proposé pour des digues semi-infinies et d’épaisseur infinitésimale. Pourtant cette méthode est utilisée en pratique sur le terrain pour calculer l’amplitude derrière les

⁵Malheureusement, nous n’avons pas pu trouver l’article original

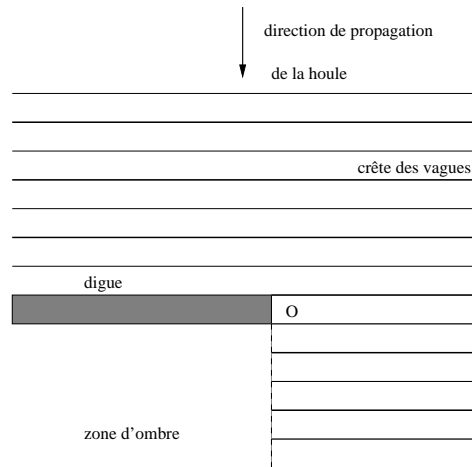


FIG. 47 – Zone d'ombre derrière une digue

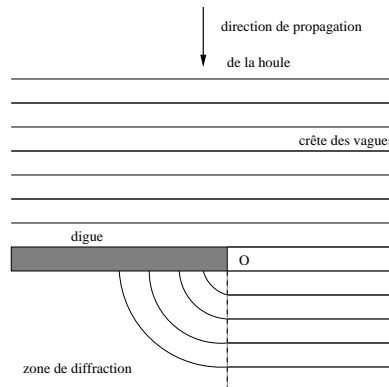


FIG. 48 – Agitation derrière une digue

digues. Lorsque plusieurs musoirs existent sur une digue, ils sont pris en compte indépendamment. L'amplitude en un point est la somme des amplitudes diffractées. L'épaisseur de la digue n'intervient que pour l'identification des musoirs.

Utilisant un système de coordonnées polaires, Larras calcule l'agitation en tout point $P(r, \beta)$ derrière la digue comme une combinaison d'une onde incidente H_i diffractée et d'une onde réfléchie H_r . Ce découpage peut être effectué en supposant le phénomène séparable : H_i est calculée si la digue est supposée parfaitement absorbante (rigide).

Ces deux ondes dépendent respectivement de deux autres coefficients géométriques U_i et U_r . Ces deux coefficients valent :

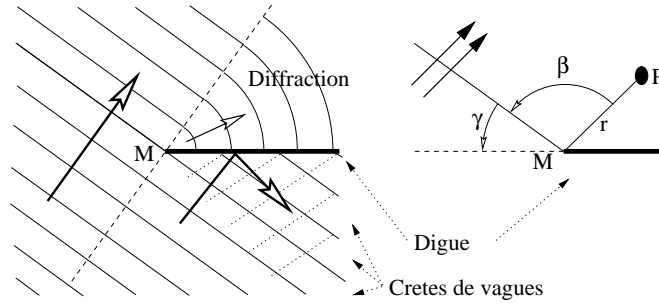


FIG. 49 – Théorie de la diffraction

$$\begin{cases} U_i = 2\sqrt{\pi \frac{r}{L}} \sin\left(\frac{\pi}{4} - \frac{\beta}{2}\right) \\ U_r = 2\sqrt{\pi \frac{r}{L}} \sin\left(\frac{\pi}{4} - \frac{\beta}{2} - \gamma\right) \end{cases} \quad (57)$$

où γ est l'angle entre la crête incidente et la jetée. Pour les points derrière la digue soumis à la diffraction, un des coefficients U_i ou U_r est nécessairement négatif.

Dans [Lac65] où la méthode de Larras est reformulée, l'onde incidente H_i (resp. l'onde réfléchie H_r) est une fonction de U_i (resp U_r) (fig 50.) donnée par les équation de Fresnel :

$$H_i = \left[\left(\int_{U_i}^{\infty} \cos \frac{\pi}{2} r^2 dr \right)^2 + \left(\int_{U_i}^{\infty} \sin \frac{\pi}{2} r^2 dr \right)^2 \right]^{\frac{1}{2}} \quad (58)$$

Finalement, l'amplitude au point P est égal à H_M , l'amplitude au musoir, multipliée par une amplitude, dite relative, H_d calculée par :

$$H_d = \sqrt{H_i^2 + K_r^2 H_r^2 + 2\alpha H_i H_r \cos \Phi} \quad (59)$$

où

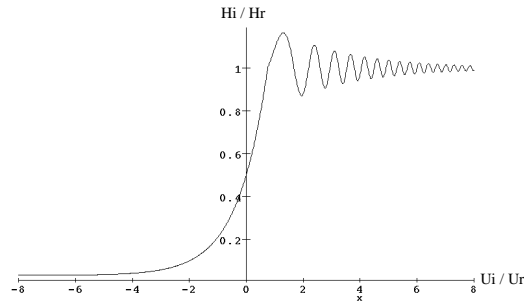
$$\Phi = 4\pi \frac{r}{L} \sin \gamma \cos \beta + \gamma \quad (60)$$

Dans cette formule, K_r est un coefficient de réflexion de la digue dont la valeur est comprise entre 0 et 1. α est la pente de l'obstacle par rapport au plan moyen de l'océan.

2.3.2 Notre modèle et son intégration dans le Lancer de Vagues

A notre connaissance, la diffraction de la houle n'a pas été introduite en infographie. Nous proposons une implémentation⁶ de la méthode de Larras où les fonctions H_i et H_r (fig 50) sont approximées. L'équation 58 est découpée en deux parties : premièrement, une exponentielle croissante est utilisée pour approximer la partie gauche de la courbe ($-\infty < U_i/r < 0.75$). Deuxièmement,

⁶Méthode aussi présentée dans [GS99b].

FIG. 50 – H_i / H_r en fonction de U_i / U_r

une sinusoïde amortie approxime la partie de droite. De plus, l'augmentation de la fréquence de la sinusoïde est traduite par l'adjonction d'une fonction logarithmique dans la description de sa phase.

La fonction H_i (resp. H_r), représentée figure 50 est approximée dans notre modèle par :

$$\left\{ \begin{array}{l} \text{Si } U_i \leq 0.75 \\ H_i(U_i) = 0.03 + 0.47e^{0.96U_i} \\ \text{sinon} \\ H_i(U_i) = 0.014e^{-0.4U_i+3} \times \sin\left(32\ln\left(1 + \frac{U_i}{15}\right)(U_i - 0.75)\right) + 1 \end{array} \right. \quad (61)$$

Ces deux fonctions ont été validées stochastiquement et graphiquement.

2.3.2.1 Définition d'une digue

Notre modèleur permet l'introduction de digues "non nécessairement semi-infinies et d'épaisseur nulle". Un algorithme simple permet de détecter un musoir. Un tel bord de diffraction est inévitablement compris entre deux orthogonales au front de vagues : la première étant réfléchi contre la digue, la seconde évite cette même digue. Une simple recherche entre ces deux points le long de la frontière de la digue permet d'obtenir le musoir. En pratique, nous utilisons des digues parallèles, définies par un parallélépipède, et ainsi, nous connaissons les coordonnées des différents musoirs.

Notre algorithme fonctionnant à l'aide de la propagation des orthogonales au front de vagues, plusieurs musoirs peuvent être traités indépendamment. On supposera, pour limiter l'explosion combinatoire, qu'une onde diffractée ne pourra l'être au plus deux fois sur une même digue.

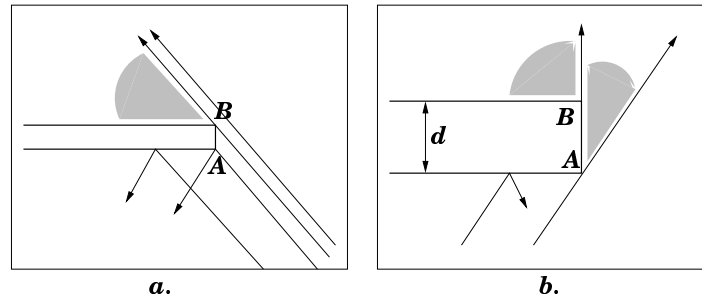


FIG. 51 – Gestion des musoirs

2.3.2.2 Calcul de la diffraction

Dans l’algorithme du lancer de vagues, tous les rayons sont envoyés dans la scène depuis la large. Au fur et à mesure de leur progression, ils sont déviés en fonction de la réfraction. A chaque pas de progression, nous testons si le rayon touche la digue.

Un jeu de drapeaux permet la gestion des rayons ayant touchés la digue qui gèrera en particulier le passage d’un rayon près d’un musoir.

Si un rayon touche la digue (Point 1 fig 52.a), nous levons un drapeau sur chacun de ses deux voisins les plus proches. De proche en proche, nous allons pouvoir déterminer le premier rayon qui passe à côté du musoir. En effet, connaissant les coordonnées des deux musoirs A et B , nous pouvons ainsi détecter rapidement le premier rayon qui a son drapeau de levé et qui passe à côté de la digue.

Une fois ce rayon déterminé, nous allons, par interpolation, déterminer sur le musoir les valeurs essentielles permettant la modélisation de la diffraction :

- Le vecteur directeur de la houle, afin de calculer β et γ
- H l’amplitude de la vague au musoir permettant de calculer l’amplitude en tous points de la zone d’ombre
- Φ la phase de la houle nécessaire pour la génération de la forme de la vague.

Des rayons nommés *DIFFRACTED* sont générés de manière radiale depuis le musoir. Les rayons progressent pas à pas dans la scène. En tout point d’échantillonnage P sur le rayon diffracté, nous calculons l’amplitude relative H_d de la vague. L’amplitude réelle en P prend alors la valeur $H_P = H_d \times H_M$ où H_M est l’amplitude au musoir. H_P est directement appliquée dans l’algorithme 2D (eq. 49) pour le calcul de la surface de l’océan le long du rayon ($R = 2 \times H_P$).

Ainsi, la méthode utilisée ne perturbe pas le Lancer de Vagues Dynamique et la génération de nouveaux rayons diffractés s’effectuera dès que la distance entre orthogonales dépasse un certain seuil.

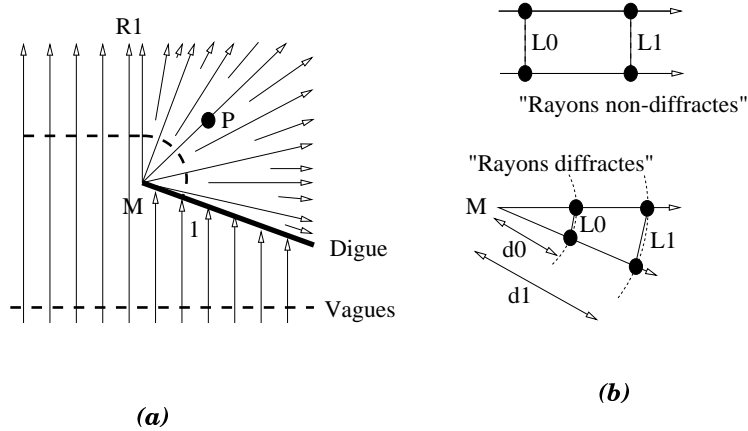


FIG. 52 – Notre méthode pour calculer la diffraction et la réfraction

2.3.2.3 Diffraction et Réfraction simultanément

Naturellement, les rayons diffractés subissent aussi la réfraction. Nous avons vu que, dans le cas standard, le calcul de l'amplitude en fonction de la distance entre deux rayons voisins est donné par :

$$H_1 = H_0 \sqrt{\frac{D_0}{D_1}} \quad (62)$$

où D_0, D_1 (resp. H_0, H_1) sont les distances en deux rayons voisins (resp. l'amplitude) avant et après un pas de progression des fronts d'onde.

La distance entre deux rayons diffractés sans réfraction émis à partir du musoir vérifie l'équation 63 puisque le pas de calcul est constant. De plus, l'amplitude est constante.

$$\frac{D_0}{D_1} = \frac{d_0}{d_1} \quad (63)$$

où d_0 et d_1 sont les distances entre le musoir et les points d'échantillonnages avant et après un pas de progression.

Dans le cas où il y a réfraction, la précédente formule n'est plus satisfaite. Nous proposons donc de pondérer la variation d'amplitude par la formule suivante :

$$H_1 = H_0 \sqrt{\frac{d_1 D_0}{d_0 D_1}} \quad (64)$$

Pour résumer, nous utiliserons la formule générique suivante pour calculer l'amplitude des vagues en fonction de la réfraction et cela pour n'importe quel point de la scène :

$$\left\{ \begin{array}{l} H_1 = H_0 \sqrt{\mu \frac{D_0}{D_1}} \\ \text{avec } \mu = 1 \text{ pour les rayons non diffractés} \\ \text{et } \mu = \frac{d_1}{d_0} \text{ pour les rayons diffractés.} \end{array} \right. \quad (65)$$

2.3.3 Résultats

Les deux figures suivantes illustrent la diffraction par un sol uniforme. Une digue verticale est insérée dans la scène. La figure 53.a présente l'ensemble de l'évolution des rayons et la figure 53.b le train de vagues correspondant. Dans la zone de diffraction derrière la digue, les rayons sont émis suivant un angle constant. Ne connaissant pas forcément l'angle définissant la plage de diffraction, un artefact apparaît lorsqu'on lance le dernier rayon pour échantillonner la zone. Ceci est visible sur la figure 53.a où la génération de nouveaux rayons ne s'effectue pas en même temps que sur le reste de la zone. Par contre, ceci n'a aucune influence sur l'échantillonnage de la surface du train de vagues (fig 53.b).

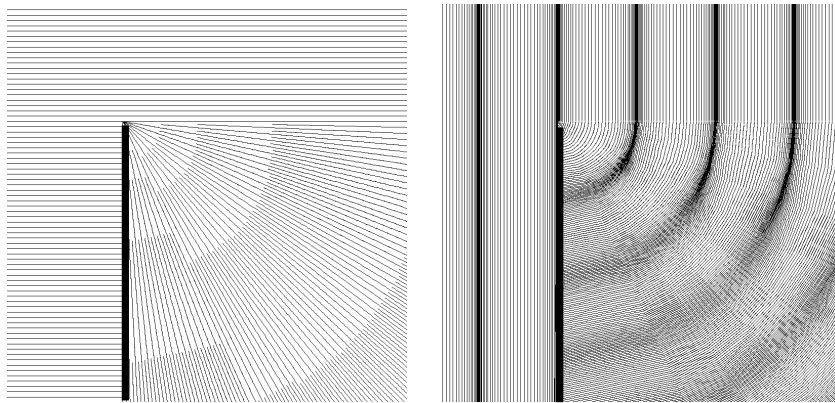


FIG. 53 – a. Les rayons lors du lancer de vagues, b. Les vagues

Cette scène mesure 100 mètres de long et 100 mètres de large ; 70 rayons primaires ont été lancés et ils progressaient par pas de 0.4 mètres. De nouveaux rayons sont créés dès que la distance entre rayons voisins dépassait 1.5 mètres. Notre algorithme a généré 125 rayons diffractés pour définir la zone d'ombre derrière la digue. Ce processus a pris moins de 5 secondes pour générer la scène entièrement (Lancer de vagues, calculs de la diffraction et des facettes) sur une Silicon Graphics O2 avec un processeur R5000.

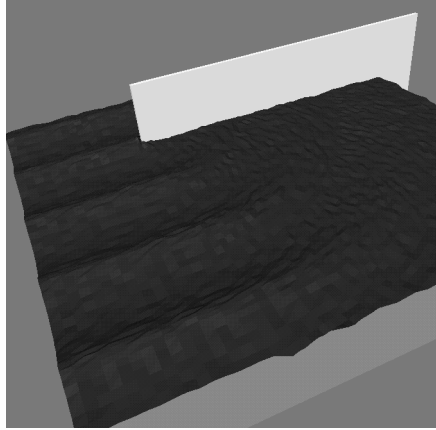


FIG. 54 – Une vue 3D de la diffraction (l’obstacle est perpendiculaire au plan moyen de l’océan)

2.4 La réflexion des vagues

2.4.1 Un peu de physique

La réflexion de la houle est le phénomène qui traduit le “rebond” d’une vague sur un obstacle. Comme la réfraction, la réflexion peut être appréhendée de deux manières : géométrique et énergétique. La nature de l’obstacle où se produit la réflexion influe notablement sur ces deux points de vue.

Lorsque l’onde incidente touche un obstacle, une onde réfléchie est générée obéissant aux lois de Descartes (l’angle d’incidence est égal à l’angle de réflexion). La composée de ces deux ondes s’appelle *houle gaufrée*. Toutefois, lorsque l’angle d’incidence est de 0° , il se crée devant la digue un état d’agitation stationnaire appelé *clapotis*. Dans ce cas, on ne parle plus d’onde gaufrée puisque l’intersection entre l’onde incidente et l’onde réfléchie n’est plus visible. Les trajectoires des particules d’eau ne sont alors plus des cercles ou des ellipses mais des segments verticaux.

Le point de vue énergétique se traduit par une variation de l’amplitude de l’onde incidente qui naturellement diminue au moment de l’impact contre l’obstacle. Cette perte d’énergie dépend d’un coefficient dit de réflexion (K_r) qui varie en fonction de la nature de l’obstacle et de la pente (α) de cet obstacle par rapport à la surface libre. Lorsque la pente est inférieure à 10%, la réflexion devient alors négligeable. La perte d’énergie peut être exprimée par le rapport entre l’amplitude de l’onde incidente juste avant la réflexion et l’amplitude de l’onde juste après l’impact [AA85].

$$\frac{H_r}{H_i} = \frac{2K_r \alpha \sin(\alpha)}{\delta_i \pi^2} \quad (66)$$

Où H_i (resp. H_r) est l’amplitude incidente (resp. réfléchie) et δ_i la cambrure de l’onde incidente.

Lorsque l'onde incidente n'est pas déferlante, la cambrure admet pour valeur limite (Miche 1944 présenté dans [Lac65]) :

$$\delta_i \leq \sqrt{\frac{2\alpha}{\pi} \frac{\sin^2 \alpha}{\pi}} \quad (67)$$

Lorsqu'il y a déferlement, une grande partie de l'énergie de l'onde disparaît. Plusieurs coefficients de réflexions sont fournis dans [AA85] :

- $K_r = 0.9$ à 1 pour les revêtements lisses en béton,
- $K_r = 0.7$ à 0.8 pour les revêtements en maçonnerie,
- $K_r = 0,6$ à $0,7$ pour les enrochements en pierre,
- $K_r = 0,5$ pour les massifs en blocs d'enrochement.

2.4.2 Notre modèle

Juste après la réflexion sur l'obstacle (digue ou rivage), nous utilisons les équations 66 et 67 pour calculer la valeur de l'amplitude de l'onde réfléchie. Par contre, il existe un phénomène non modélisé qui est la modification de la trajectoire circulaire ou elliptique des particules d'eau. En effet, celle-ci est perturbée par la superposition de l'onde incidente et réfléchie. L'onde incidente est légèrement atténuée et l'onde réfléchie amortie. Dans notre modèle, nous avons choisi de ne pas perturber l'onde incidente et d'utiliser une fonction d'amortissement classique pour l'onde réfléchie : $\phi(d) = e^{-\frac{H_i}{H_r}d}$ où d est la distance parcourue par l'onde depuis la réflexion. Le quotient $\frac{H_i}{H_r}$ traduit le fait que plus l'amplitude incidente est proche de l'amplitude réfléchie, plus l'amortissement de l'onde réfléchie est faible. L'amplitude au point $P1$ (fig 55) vaut :

$$H_{P1} = \phi(d) \times H_r \quad (68)$$

Dès que l'amplitude de la vague réfléchie passe en dessous d'un certain seuil, l'onde ne progresse plus.

2.4.2.1 Réflexion sur le musoir

Le musoir est certes un point de diffraction, mais aussi un point de réflexion particulier. En effet, en un tel point il y a une réflexion dans toutes les directions. Ceci se traduit par de petites ondes circulaires réfléchies. Pour modéliser ces fines ondulations partant du musoir, nous proposons de rajouter des rayons de type réfléchis à partir de ce musoir. Une certaine continuité sera alors assurée entre l'onde réfléchie et l'onde transmise. L'amplitude des orthogonales décroîtra en fonction de leur éloignement à la première onde réfléchie. Nous proposons pour cela la formule empirique suivante :

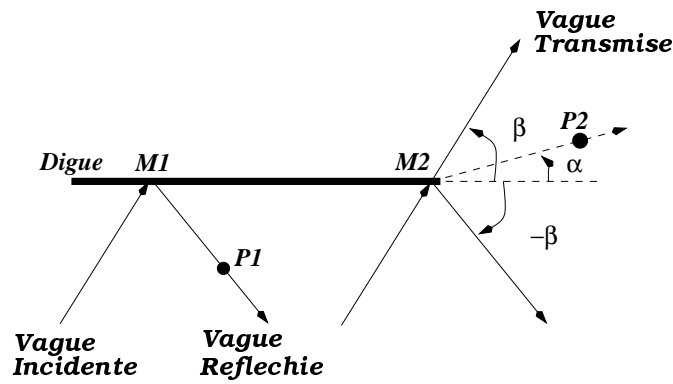


FIG. 55 – Calcul sur le musoir

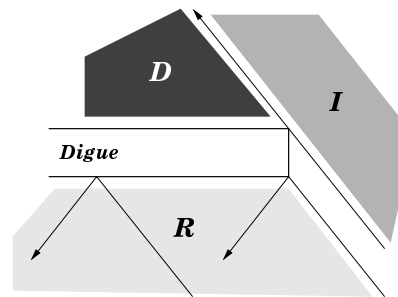


FIG. 56 – Différentes zones de calculs

$$\psi(\alpha) = \cos\left(\frac{\pi(\alpha + \beta)}{4\beta}\right) \quad (69)$$

Cette fonction vaut 1 dans la direction de réflexion et 0 dans la direction de la houle incidente (voir application fig. 57.b).

L'amplitude au point $P2$ vaut alors :

$$H_{P2} = \psi(\alpha) \times \phi(d) \times H_r \quad (70)$$

Où H_r est l'amplitude de l'onde juste après impact au point $M2$.

2.4.3 Résultats

Dans ce paragraphe, nous présentons deux exemples sur la réflexion des vagues (fig. 57). L'image de gauche est une scène composée d'un seul train de vagues réfléchi et diffracté. La vue est produite par un simple algorithme de z-buffer via la bibliothèque Open Inventor. Elle permet de se rendre compte de la trajectoire des vagues. Cette image ne correspond pas à l'aspect que nous obtenons avec notre algorithme où la somme des amplitudes des différentes ondes n'est calculée

que dans la phase d'illumination. Cette image ne présente que les parties d'océan les plus proches de l'observateur. La deuxième image illustre la surface de l'océan pour les vagues réfléchies contre un musoir.

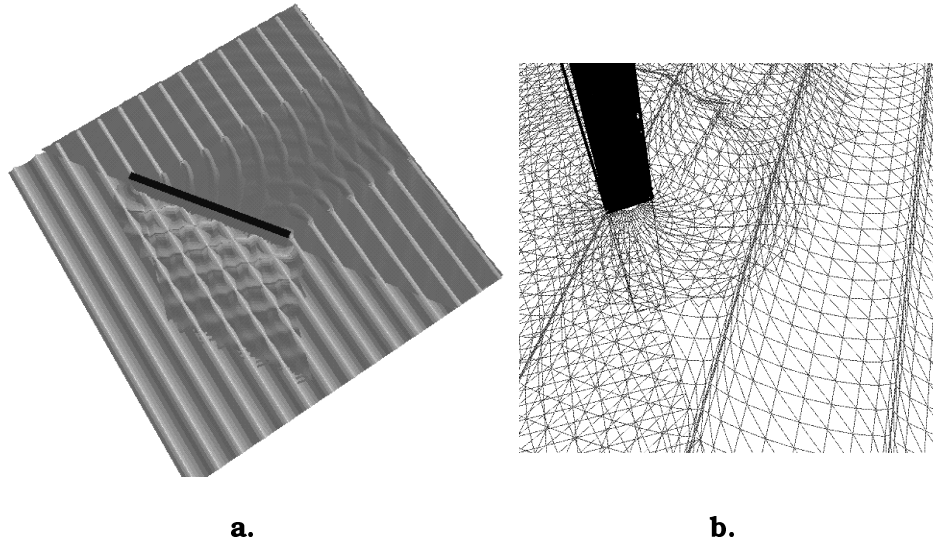


FIG. 57 – a. Une vue Inventor de la réflexion, b. Cas du musoir

2.5 Facettisation et Multi-trains de vagues

Une fois les différents modèles étudiés et adaptés à l'infographie tels que nous l'avons vu dans les paragraphes précédents, une étape de facettisation préalable à l'éclairage de la surface est nécessaire. Nous rappelons que la surface de l'océan ne peut pas être stockée par un simple champ scalaire. Nous décidons donc de stocker la surface dans une nappe de facettes.

Les rayons progressent pas à pas dans la scène. Pour des raisons de coût mémoire, seule la position courante et la position précédente ne sont stockées par rayon au cours de la progression. L'étape de facettisation se fera donc au fur et à mesure de la progression des rayons. Deux facettes triangulaires seront donc créées entre deux rayons (sans marquage particulier ou marquées *FLAT* ou *AGROUND*) à chaque pas de progression (Point 1 fig 58). Les rayons marqués *AGROUND* permettront de générer la surface jusqu'au rivage. Lors de la création de nouveaux rayons, nous générerons trois facettes (point 2 fig 58).

Durant l'étape de progression, plusieurs paquets de rayons peuvent se croiser en fonction des phénomènes de réfraction influant la trajectoire des ondes. Dans ce cas, nous proposons de garder l'ensemble des facettes. La somme des amplitudes des diverses facettes s'effectuera durant la phase de rendu.

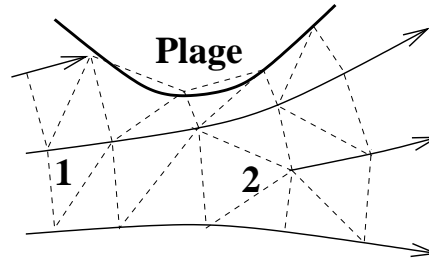


FIG. 58 – Création des facettes de la scène

De plus, la surface de l’océan étant composée de plusieurs trains de vagues ayant des caractéristiques différentes (direction, longueur d’onde, amplitude au large, ...), lors de la phase de facetisation, nous générons donc l’ensemble de facettes pour chacun des trains d’ondes indépendamment les uns des autres. Ainsi, le modèle 3D final est composé d’un ensemble de nappes de facettes représentant chacune un train de vagues. La surface finale réelle correspondant à la composée des divers trains de vagues se fera donc durant la phase d’illumination.

2.6 Récapitulatif sur le Lancer Dynamique de Vague complet

Dans les paragraphes précédents, nous avons détaillé une méthode permettant de modéliser les principaux phénomènes influant la progression des trains de vagues en fonction de la bathymétrie du sol⁷. L’utilisation des orthogonales aux fronts de vagues est le point commun à l’ensemble des modèles pour simuler réfraction, diffraction et réflexion.

Les orthogonales au front de vagues ou rayons progressent en parallèle et pas à pas dans la scène à modéliser. Il sont lancés a priori depuis le large et dès que la distance entre deux rayons consécutifs dépasse un certain seuil, un nouveau rayon est créé au milieu afin de garantir un même échantillonnage de la scène. Du point de vue algorithmique, pour limiter les coûts mémoire, une liste chaînée permet de stocker l’ensemble des rayons à un instant donné de la modélisation. La facetisation est effectuée au fur et à mesure de la progression. Tant que les phénomènes sont limités à la réfraction, la gestion des rayons par une liste est aisée. En effet, dès qu’un nouveau rayon est créé, une nouvelle cellule est rajoutée au bon endroit dans la liste. La notion de voisinage est alors préservée.

Dès que les phénomènes de diffraction et réflexion sont présents, de nouveaux rayons diffractés et réfléchis sont créés. La gestion de voisinage dans la liste n’est plus aussi aisée. En effet, on peut simplement se demander comment insérer de nouvelles cellules, avec une simple liste chaînée, pour garder la notion de voisinage entre le premier rayon diffracté par un musoir (D), le premier

⁷Une version concise du modèle complet peut être vue dans [GS00].

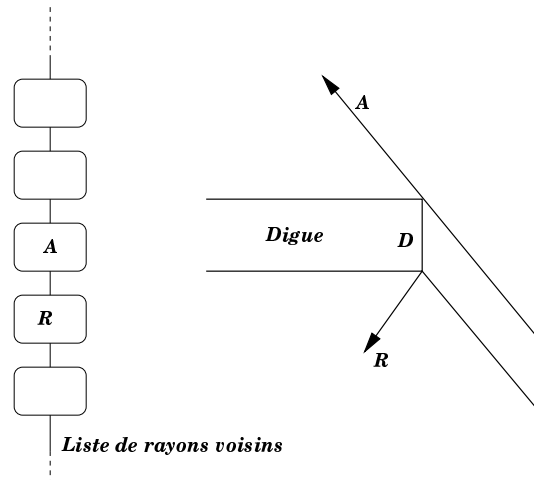


FIG. 59 – Gestion du voisinage des rayons dans la liste

rayon réfléchi (R) et le rayon passant à côté de ce même musoir (A) (fig. 59). Si l'on prend exemple sur le lancer de rayons, lorsque pour un même objet, un rayon transmis et un rayon réfléchi sont envoyés, une structure en arbre est alors utilisée.

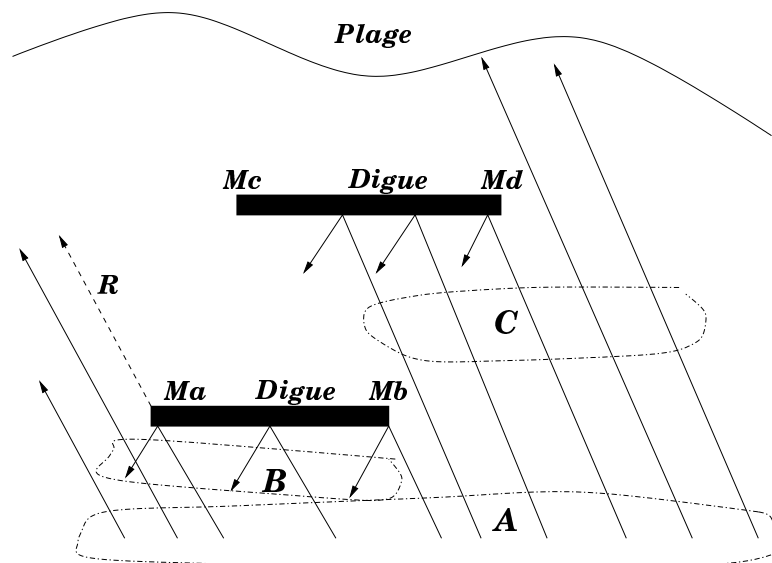


FIG. 60 – Ensemble des rayons se propageant dans une scène

Pour pallier à ce problème de gestion des différents voisins dans la liste, nous décidons de traiter chaque phénomène indépendamment les uns des autres à l'aide d'une file d'attente. Énonçons le principe sur l'exemple suivant (fig 60) :

La scène à modéliser est composée de deux digues finies et la pente du sol sur la côte est inférieure à 10% pour éviter les phénomènes de réflexions des vagues. Un premier train de vagues

(A) est lancé dans la scène. Lorsque celui-ci heurte le premier musoir, nous stockons dans une file le point de diffraction avec les données qui sont nécessaires à la modélisation des rayons derrière la digue. Les rayons réfléchis (B) sont générés par un simple remplacement dans la liste des rayons incidents. Les rayons qui sont passés à côté de la digue (C) continuent leur propagation. Dès qu'ils touchent un autre musoir, de la même manière le point de diffraction est stocké dans la file d'attente. Et ainsi de suite pour les différents musoirs. Dans notre exemple, au passage du premier train de vagues, seuls les musoirs M_a , M_b et M_d seront stockés dans la liste comme points de diffraction à traiter. En effet, le musoir M_c se trouve dans l'ombre de la première digue et les rayons diffractés n'ont pas encore été traités.

Une fois que tous les premiers rayons ont terminé leur progression dans la scène, i.e dès que la liste est vide, nous récupérons le premier élément de la file des phénomènes à traiter, i.e le musoir M_a comme point de diffraction. Nous générons donc les rayons diffractés par ce musoir. Nous pouvons considérer les phénomènes indépendants si les données essentielles à la génération du phénomène mis en attente sont stockés dans la file. Pour un musoir, la direction de la houle incidente ainsi que son amplitude et sa longueur d'onde seront stockées. Afin d'effectuer une reconnexion complète des deux phénomènes, il ne faut pas oublier de lancer un rayon depuis le musoir (rayon R).

Ainsi de suite, nous pouvons générer, à l'aide de cette file, l'ensemble des phénomènes indépendamment les uns des autres. Une fois la file vide, nous pouvons générer la facettisation d'un deuxième train de vagues partant du large.

En conclusion, nous présentons dans un tableau récapitulatif de l'ensemble des principales méthodes infographiques qui existent pour modéliser les divers phénomènes des océans.

Auteur	Modèle 2D	Réfraction	Diffraction	Réflexion	Multi- train de vagues
Fournier-Reeves	X	X (dans la direction du train de vague initial)			X
T'so-Barsky	X	X (Wave Tracing)			X
Peachey	X				X
Nous	X	X	X	X	X

Chapitre 3

Faire la lumière sur l’océan

Dans les deux chapitres précédents, nous avons détaillé un ensemble de méthodes permettant de modéliser les principaux phénomènes liés à la propagation des vagues. Afin de générer des images, les plus proches de la réalité possible, et afin de pouvoir déterminer la couleur¹ de l’océan, nous allons proposer un modèle d’illumination des océans adapté à notre modélisation.

Dans ce chapitre, après une présentation des travaux déjà réalisés en infographie sur l’illumination des scènes océanes, nous proposons notre modèle de rendu. Celui-ci est basé sur un algorithme de lancer de rayons adaptés à nos scènes océanes. En particulier, le résultat de notre modélisation géométrique ne permet pas un calcul direct de l’illumination par lancer de rayons puisque la surface n’est pas explicitement décrite. Les données fournies par notre modéleur sont des empilements de facettes associées à chaque trains de vagues. La hauteur d’eau réelle dépendant des facettes superposées. Aussi, tout d’abord, nous décrivons comment est identifiée la surface réelle de l’eau. Ensuite, nous développerons la méthode utilisée pour améliorer la qualité de nos images en introduisant les vagues capillaires. Ceci n’a pas été fait plus tôt, car cette méthode dépend de la position de l’observateur. Enfin, nous expliciterons le calcul de l’illumination locale basée sur le modèle de Blinn [Bli77], l’approche globale étant fournie par le lancer de rayons².

3.1 Travaux précédents

Modèle de Peachey Peachey [Pea86] gère la géométrie de la surface dans une grille régulière. Pour identifier les points de la scène où l’illumination doit être calculée, l’auteur utilise un algorithme de A-Buffer, version du Z-Buffer minimisant les effets d’aliasing. Après avoir déterminé l’illumination au sommet de chaque facette par une méthode classique, la couleur au point concerné est calculée par interpolation de Gouraud [Gou71]. Une telle approche appliquée sur des facettes de grandes tailles génère des discontinuités qui altèrent, par exemple, les reflets solaires.

¹La lecture du livre de R. Sève traitant de la couleur peut être conseillée [Sèv96]

²Les modèles d’identification de la surface et d’illumination ont été présentés dans [GS99b].

Modèle de Mastin De la même manière que Peachey, Mastin [Ma87] stocke la géométrie de la surface de l'océan dans une grille régulière de hauteurs. De plus, la scène est découpée en un ensemble de boîtes englobantes hiérarchisées en Quad-Tree. Les caractéristiques optiques de l'océan, dans cette méthode, sont identiques en tout point de la scène : le coefficient de transmission est fixé à 30% et le coefficient de réflexion à 70%. La couleur est alors obtenue avec le modèle d'illumination de Phong [Pho75]. Celle-ci dépend donc de l'angle entre la normale à l'océan au point considéré et la direction du soleil (source lumineuse directionnelle). Elle dépend, de plus, de l'angle d'observation. Le principal défaut de cette méthode, admis par Mastin, est la non-utilisation des coefficients de Fresnel qui permet de déterminer les valeurs des coefficients de transmission et de réflexion en fonction de l'angle d'observation. L'auteur utilise un lancer de rayons pour calculer l'illumination globale.

Modèle de Ts'o et Barsky Dans les deux méthodes précédentes, l'eau a globalement un aspect plastique. Pour pallier à cet inconvénient, utilisant aussi un algorithme de Z-buffer, Ts'o et Barsky [TB87] appliquent deux textures sur la surface : une de réfraction et une de réflexion. Le calcul de la couleur d'un pixel, dépendant de la normale à la facette et de l'angle d'incidence, est associé à un point de la texture de réfraction à majorité "bleu-mer" et à un point de la texture de réflexion à majorité "bleu-ciel". Pour calculer les coefficients de transmission et de réflexion, les coefficients de Fresnel sont approximés. La fonction utilisée est dans un premier temps une demi-droite qui se prolonge par une demi-parabole croissante (fig 61).

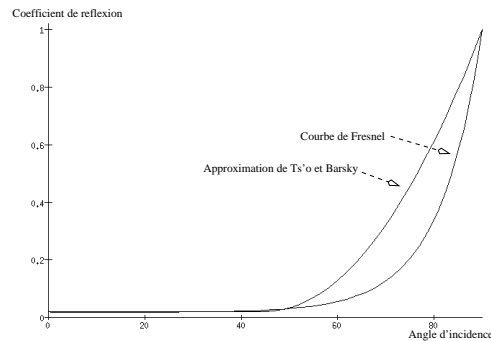


FIG. 61 – Approximation des coefficients de Fresnel par Ts'o et Barsky

Aux vues de ces deux courbes, on peut remarquer simplement que la part d'énergie venant de la réflexion est plus importante que dans la réalité. Ceci n'a que peu d'importance dans le rendu de la scène.

Modèle de Fournier et Reeves A la différence des précédentes méthodes, l'océan est ici géré sur une grille non-uniforme. Plus l'observateur sera proche, plus la précision de la grille sera grande. La couleur de l'océan ne dépend plus uniquement du soleil mais aussi du ciel. Le ciel est modélisé par une texture hémisphérique entourant l'océan. Le rendu est obtenu par le logiciel *Reyes rendering software* utilisant le lancer de rayons. La couleur de l'eau est calculée par la formule suivante :

$$\text{REF} = \text{Constante de reflexion de l'eau,}$$

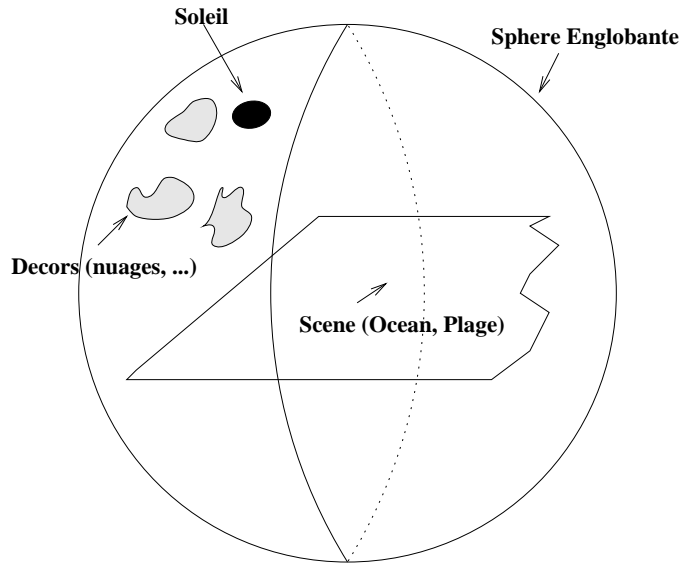


FIG. 62 – Représentation de l'environnement de la scène

`reflcolor` = triplet (R,G,B) correspondant a la couleur de reflexion,
`refl()` = retourne la couleur provement de la carte d'environnement
`watercolor` = triplet (R,G,B) correspondant a la couleur de l'eau
`AMB` = Constante pour l'aspect Ambient de la scene
`ambient()` = retourne la couleur ambiente et son intensite
`DIFF` = Constante pour l'aspect Diffus de l'eau
`diffuse()` = retourne la couleur de la partie diffuse de la partie touchee et son intensite
`SP` = Constante pour la partie Speculaire de l'eau
`specular()` = retourne la couleur de la partie speculaire de la partie touchee et son intensite
`ROUGH` = Constante definissant la limite en dessous de laquelle l'objet est completement speculaire.

*/

```

Couleur_finale = REF * reflcolor * refl(normale, camera) +
                couleurOcean * (AMB * ambient() + DIFF * diffuse(normale) +
                SP * specular(normale, camera, ROUGH));
  
```

Dans les fonctions `refl()` et `specular()` sont inclus le calcul des coefficients de Fresnel.

Les auteurs suggèrent certaines valeurs qui semblent être les plus appropriées à la définition de l'eau :

– REF = 0.8

- AMB = 0.01
- DIFF = 0.1
- SP = 0.5
- ROUGH = 0.02

Les limitations principales de ce modèle proviennent de ce que l'océan n'est pas un objet décrit comme transparent. On ne peut donc pas prendre d'objets sous l'eau. En terme de performance, la figure 13 de leur article [FR86] est une image de taille 2048×1228 contenant 92 724 facettes et à été générée en environ 10 heures sur un *Power 6/32*.

Un produit commercial La firme "Areté Image Software" propose des images de synthèse d'océan de très grande qualité. La surface de leur océan est stockée dans une grille hiérarchique. Le rendu est effectué soit par des produits standards tels Softimage ou Maya. Ils ont, de plus, développé leur propre moteur de rendu nommé RenderWorld. Cette firme est spécialisée dans les effets spéciaux liés au cinéma (Titanic, Waterworld, ...) ou à la publicité. Pour des raisons commerciales évidentes, leurs algorithmes et méthodes ne sont pas diffusés. Nous invitons donc le lecteur à visiter le site <http://www.aretéis.com/> montrant la qualité des images obtenues.

3.2 Identification de la surface

Comme nous l'avons dit la géométrie de notre surface est donnée par empilement de facettes. L'algorithme d'illumination étant basé sur le lancer de rayons, nous devons détecter le point d'intersection d'un rayon d'illumination avec la surface. La technique que nous utilisons est basée sur le *Displacement Mapping* [CR84]. Cette technique est utilisée lorsque les algorithmes classiques d'amélioration de la surface d'un objet (*BRDF* [Sch94] et *Bump mapping* [Bli78]) sont inefficaces. Une comparaison entre les différents algorithmes a été effectué par Becker et Max [BM93]. Le displacement mapping devient indispensable dès que l'objet est observé de manière rasante. Tant que l'on observe un objet perpendiculairement à la surface, les techniques de BRDF et Bump mapping sont entièrement efficaces. Par contre, lorsque l'angle d'incidence est faible, les BRDF et le Bump mapping ne modifient que la couleur de l'objet mais pas la géométrie. Ainsi, pour un objet originellement plat, il apparaîtra toujours aussi plat avec les BRDF et le Bump mapping alors qu'avec le displacement mapping, il sera complètement déformé. Pourtant cette technique a des conséquences sur le calcul d'intersection entre un rayon d'illumination et la surface car cette dernière n'est plus connue explicitement.

Nous utilisons cette méthode de la manière suivante : notre objet, l'océan, est supposé plat (surface libre, $z = 0$). Les déplacements seront obtenus en utilisant les positions relatives des facettes au point considéré pour calculer la surface de l'objet. Nous proposons un algorithme de lancer de rayons de type incrémental permettant d'identifier au plus près le point d'intersection entre la surface de l'océan et le rayon.

Afin d'accélérer la recherche d'intersections entre un rayon et les divers objets de la scène, les différentes facettes de la surface océane sont incluses dans une boîte englobante. En dehors de cette boîte, les algorithmes standards d'intersection seront alors utilisés pour les objets hors de l'eau. La hauteur de la boîte est donnée par :

$$H_{box} = \sum_{t_i} H(t_i) \quad (71)$$

où H_{box} est l'amplitude maximale de la houle i.e. la somme des amplitudes maximales pour chaque train de vagues.

Un cas particulier reste à traiter : le déferlement. Afin de pouvoir effectuer correctement la somme des ondes de chaque train de vagues et ainsi décrire la surface réelle de l'océan, chaque facette sera référencée par le train de vagues qui l'a générée : le numéro du train de vagues par exemple. Un cas est présenté figure 63 où pour effectuer la somme des ondes au niveau du déferlement, nous ajouterons, en suivant la verticale, l'amplitude de la facette 2 à chacune des facettes 1.

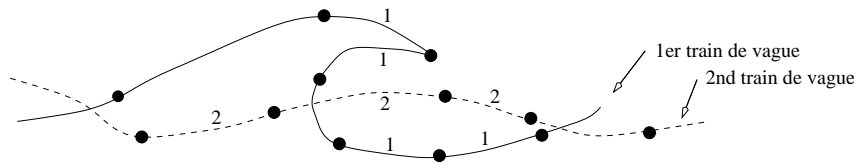


FIG. 63 – Cas particulier du déferlement

Calcul du point d'intersection Pour effectuer le rendu de la scène, nous utilisons un lancer de rayons de type incrémental classique en dehors des boîtes englobant la surface des océans. Ces boîtes englobantes sont des objets transparents non réfractant à part entière. Ils ont une existence géométrique, mais n'ont aucune influence optique sur le rendu.

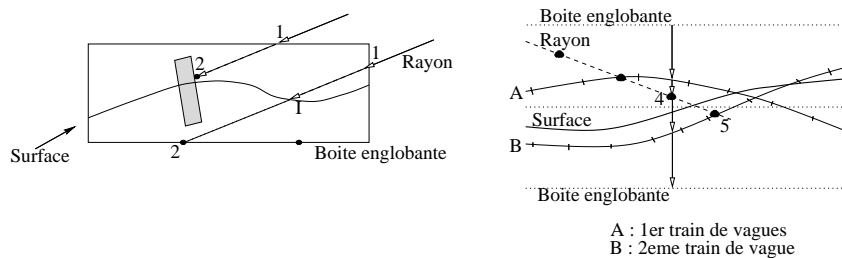


FIG. 64 – Recherche de la surface de l'océan

Lorsqu'un rayon, partant de l'oeil, intersecte une boîte englobante (Points 1. dans la Fig 64), la distance maximale d'exploration de la surface est déterminée (voir figure 64 entre les points 1 et 2). Deux cas peuvent se produire : soit le rayon ne touche aucun objet autre que la surface (celle-ci est alors comprise entre le point d'entrée et le point de sortie de la boîte), soit le rayon intersecte un autre objet présent dans la boîte (la distance maximale est comprise entre le point d'entrée et le point d'intersection de l'objet).

A l'intérieur de la boîte englobante, le rayon progresse pas à pas. Pour des raisons d'efficacité, ce pas va être adaptatif. Ainsi, il dépend de la distance à l'observateur et sera raffiné en fonction de la distance du rayon lumineux à la position réelle de la surface située au dessous de lui. Ce pas est donc évalué à chaque étape.

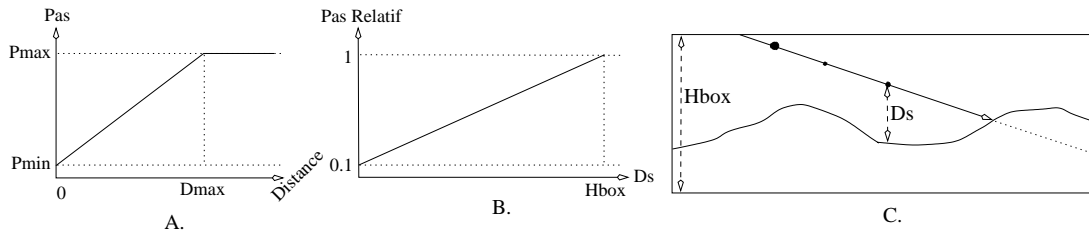


FIG. 65 – Différentes fonctions pour évaluer le pas de progression

Le pas ($Pas(distance)$) est donc une composée de deux fonctions : la première est une fonction F de la distance à l'observateur, la seconde G de la distance Ds à la surface (fig 65) :

$$\begin{aligned}
 Pas(distance) &= F(distance) \times G(Ds) \\
 G(Ds) &= \frac{0.9}{H_{box}} \times Ds + 0.1 \\
 F(distance) &= \frac{P_{max} - P_{min}}{d_{max}} \times distance + P_{min}
 \end{aligned} \tag{72}$$

où P_{min} (resp. P_{max}) le pas minimum (resp. maximum) choisi par l'utilisateur. A partir de la distance d_{max} , le pas sera constant et sera égal à P_{max} .

A chaque étape, la position réelle de la surface est calculée par sommation vectorielle des amplitudes des trains de vagues. Ceci nous permet de savoir si le pas courant est au-dessus ou au-dessous de la surface. Dès que le point courant se trouve sous la surface, une dichotomie est effectuée avec le pas précédent pour identifier la position d'intersection la plus approchée entre le rayon et la surface³. Dans la figure 66.a, la surface n'est pas détectable. Par contre, l'artéfact produit est faible par rapport au pas de progression du rayon. Cette erreur est alors négligeable. Dans la situation 66.b, une intersection est dans tous les cas détectée. Si celle-ci est le point C, comme précédemment, un artéfact est produit mais négligeable au vu du pas. S'il s'agit du point B, la normale dont le calcul est expliqué dans le paragraphe suivant en ce point nous permet⁴ de relancer une dichotomie entre le point 3 et le point B. Enfin, le cas idéal est lorsque le point A est détecté.

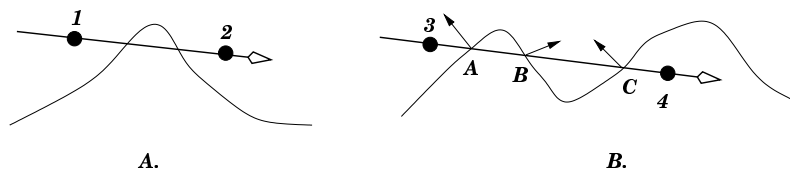


FIG. 66 – Problèmes dans la recherche de la surface

Le calcul de la surface au point courant est obtenu en lançant un rayon vertical depuis le sommet (plafond) de la boîte et passant par le point courant. La hauteur est la somme vectorielle des amplitudes associées aux facettes.

³Ceci est valable lorsque le rayon est au dessus de la surface lors de son entrée dans la boîte englobante. Dans le cas de scènes partiellement sous marines, un calcul dual est effectué.

⁴Ici, la normale au point B n'est pas dans la direction du rayon.

Une fois déterminé le point de la surface intersectée par un rayon, nous avons besoin de la normale en ce point pour effectuer l'illumination. Pour déterminer cette normale, nous calculons les deux vecteurs tangents à la surface en effectuant la somme des plans tangents aux différentes facettes rencontrées lors du calcul de la surface. La normale est alors le produit vectoriel des deux vecteurs déterminant le plan tangent trouvé :

$$N = (1.0, 0.0, -\sum_{i=1}^n \frac{x_i}{z_i}) \wedge (0.0, 1.0, -\sum_{i=1}^n \frac{y_i}{z_i}) \quad (73)$$

où $N_i(x_i, y_i, z_i)$ est la normale au train de vagues i , i -e à la facette i au point d'intersection avec le rayon vertical descendant.

Afin d'éviter tous les problèmes de discontinuités entre les normales des différentes facettes et aussi les normales de la surface réelle de l'océan, l'ensemble des N_i est calculé par interpolation suivant la méthode classique de Phong.

3.3 Amélioration de la géométrie : les vagues capillaires

Les vagues capillaires sont les petites perturbations de la surface de l'océan. Celles-ci sont principalement générées par le souffle du vent sur la surface. Ces petites perturbations que l'on nomme encore vagues, ont un signal de fréquence élevé et une amplitude faible. Elles peuvent être obtenues par un bruit continu⁵.

Entre la longueur d'onde du train de vagues à modéliser et les vagues capillaires, il existe à la surface de l'océan tous les types de longueurs d'ondes. Ces perturbations sont modélisées par un bruit fractal $F(n, x, y, t, k)$ (un exemple avec plusieurs niveaux est donné fig 67). Pour cela nous utilisons une fonction de bruit continu (Bruit(X, Y, T)) qui sera décrite un peu plus loin. La profondeur de récursion de notre bruit dépendra de l'observateur. En effet, plus la surface à observer est éloignée de l'observateur, moins les

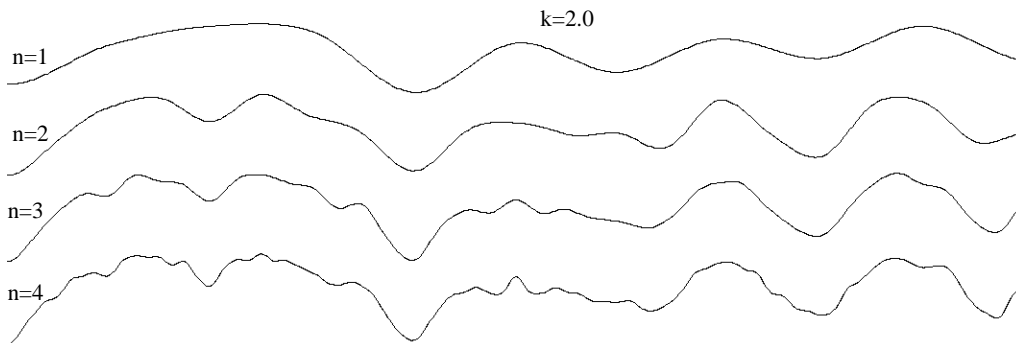


FIG. 67 – Quatre niveaux de bruits fractals différents

détails sont nécessaires. Le niveau de récursion est fixé de manière discontinue en fonction de la distance à l'observateur. Pour éviter les discontinuités à chaque changement d'échelle, on utilise un paramètre D

⁵Mastin [Ma87], proposa un modèle de vague basé sur les transformées de Fourier et l'analyse spectrale de la fonction bruit définissant la surface de l'océan. Cette méthode inclut entre autres les vagues capillaires.

variant de 0 à 1 pour garantir la continuité du signal. Ce paramètre permet l'adjonction progressive des détails d'un niveau à l'autre (fig 68).

Soit la fonction de bruit fractal F :

$$F(n, x, y, t, k) = \sum_{i=1}^n \frac{\text{Bruit}(x, k^i, y, k^i, t)}{k^i} + D \times \frac{\text{Bruit}(x, k^{n+1}, y, k^{n+1}, t)}{k^{n+1}} \quad (74)$$

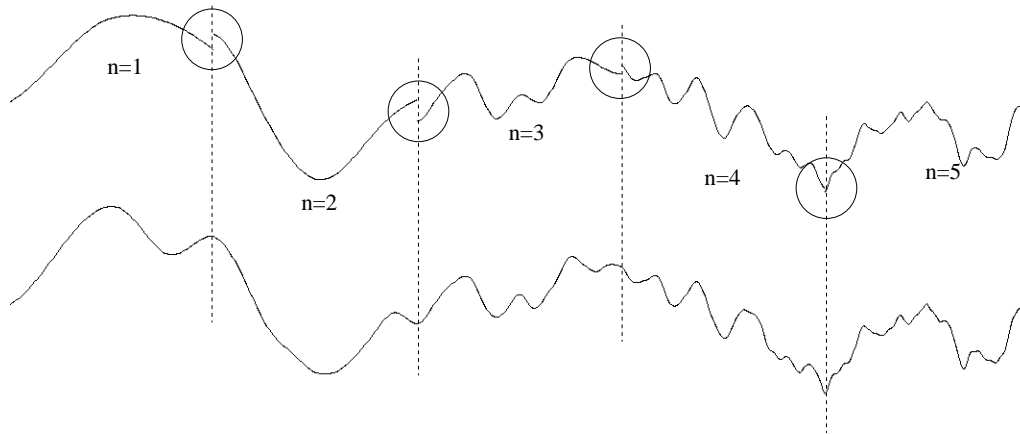


FIG. 68 – Passage continu d'un niveau de bruit au niveau supérieur

Calcul d'un bruit fractal continu : $F(n, x, y, t, k)$ Afin de décrire rapidement le modèle de bruit fractal continu, nous allons nous placer en 2D et calculer $F(n, x, k)$. Le passage à la 3D sera alors immédiat.

Pour créer un bruit continu aléatoire, nous allons construire un vecteur chaîné et circulaire de longueur arbitraire. Dans chaque cellule de ce vecteur, nous allons choisir, au hasard, un nombre J tel que $-Amp < J < Amp$ où Amp désigne l'amplitude maximale désirée. En tout point de l'espace, nous calculons la valeur du bruit par interpolation entre les valeurs choisies. Nous utilisons l'interpolation en spline-cardinale détaillée paragraphe §2.2.3.2.

Soit un algorithme écrit en C :

```
#define SIZE_VECTEUR 20
float NOISE[SIZE_VECTEUR]

float Bruit(float x)
{
    int ix; /* partie entiere de x */
    float dx; /* valeur comprise en 0 et 1 */

    float A, B, C, D; /* valeur des 4 points necessaires
                       a l'interpolation */
```



```

ix = (int)(x);
dx = x - ix;

/* on effectue le modulo pour retrouver l'index dans le vecteur de
bruit */
A = NOISE[ix % SIZE_VECTEUR];
B = NOISE[(ix+1) % SIZE_VECTEUR];
C = NOISE[(ix+2) % SIZE_VECTEUR];
D = NOISE[(ix+3) % SIZE_VECTEUR];
return 0.5*( 2*B + (C-A)*dx + (2*A - 5*B + 4*C -D)*dx*dx + (3*(B-C) \
- A + D)*dx*dx*dx);
}

float F(float x,n,k)
{
int i;
float valeur=0;
float echelle=1;
for (i=0; i<n, i++)
{
valeur += Bruit(x/echelle)*echelle;
echelle /= k;
}
return valeur;
}

```

Pour introduire le temps dans la fonction de bruit (Bruit(X,T)), classiquement, une dimension au vecteur est rajoutée. L'interpolation cardinale-spline nécessite les 16 valeurs voisines du point à calculer. La fonction Bruit en 3D (Bruit(X,Y,T)) nécessite une matrice 3D de bruit et l'interpolation prend les 64 valeurs voisines du point à calculer.

La fonction de bruit présentée jusqu'à présent est isotrope. Or, le vent influe largement sur les vagues capillaires. Il faut donc utiliser un bruit anisotropique et lié à la propagation de la houle. Pour cela, nous utilisons les orthogonales aux fronts de vagues pour étirer dans le bon sens le bruit précédemment décrit.

Résultats de l'algorithme de calcul de la surface de l'océan La figure 69 présente trois cas particuliers de notre algorithme de définition de la surface. La première est une facette unique sur laquelle un bruit fractal a été appliqué. La seconde image présente deux facettes perturbée et leur réflexion dans un miroir. Dans ce cas le miroir est un objet extérieur à la boîte englobante qui définit la surface de l'océan. Finalement, la dernière image montre l'intersection d'une facette perturbée avec un objet situé à la surface.

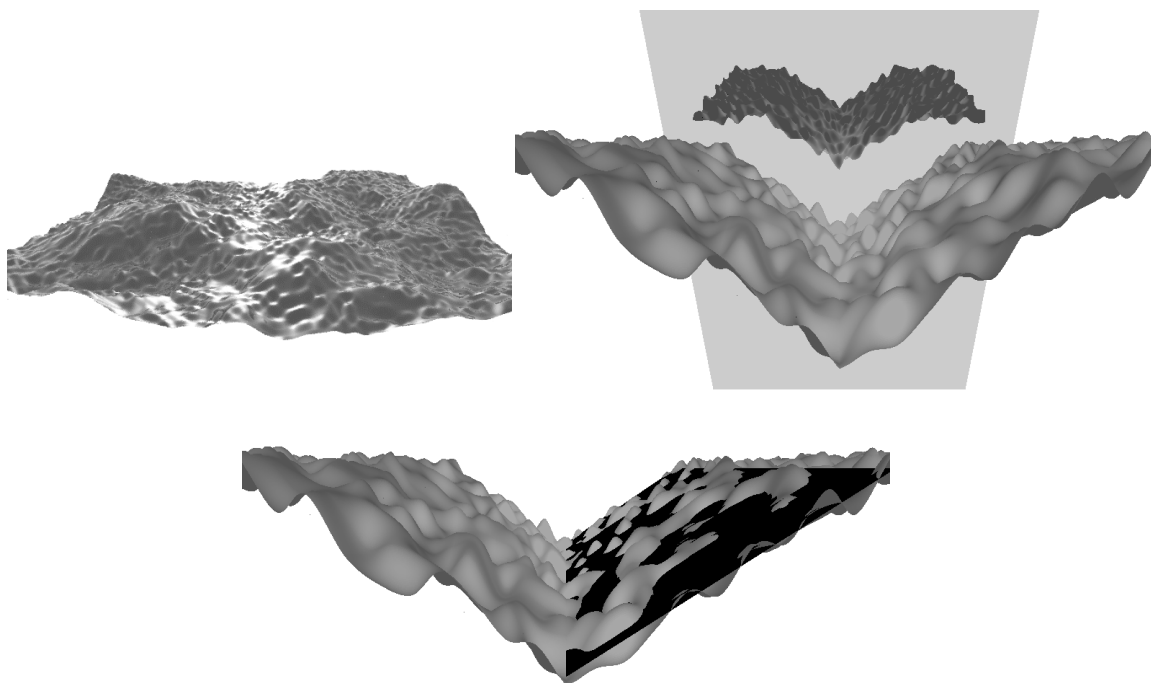


FIG. 69 – Différentes configurations de surface avec des vagues capillaires

3.4 Modèle de rendu

3.4.1 Couleur des océans

Du point de vue optique, la lumière solaire arrivant à l'interface air-eau, se décompose en deux parties : l'une réfléchi, l'autre transmise. L'océan n'étant pas parfaitement transparent, la diffusion et l'absorption de la lumière se produisent. L'absorption consiste en une transformation de la lumière en chaleur sous l'effet joule. Lorsqu'un rayon de lumière heurte une particule, sous l'effet de la diffusion, il est renvoyé dans toutes les directions. Une partie des rayons lumineux est alors redirigée à la surface. Elle est appelée énergie rétro-diffusée. Toutes les particules en suspension dans l'eau (plancton, déchets organiques, etc.) diffusent la lumière.

La lumière blanche est peu à peu absorbée dans l'eau. Les différentes longueurs d'onde du spectre visible ne sont pas absorbées équitablement. En effet, tout d'abord les rouges disparaissent, puis les verts, ... Les composantes bleues sont celles qui traversent la plus grande épaisseur d'eau avant d'être absorbées. Ainsi, l'océan nous paraît globalement bleu. On peut noter toutefois qu'en certains endroits, la présence de plancton est telle que l'océan prend un couleur bleu-verte.

Ivanoff [Iva75] fournit une liste de couleurs d'océan. Ces couleurs dépendent essentiellement de l'ensoleillement. Elles sont données pour une position spécifique du soleil (Azimut) et pour une couverture nuageuse donnée. Ces couleurs ont été obtenues à l'aide d'un spectromètre. Le spectre d'émission de l'océan

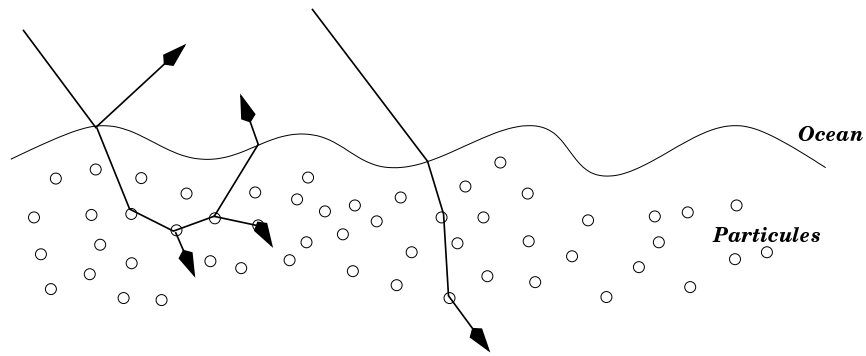


FIG. 70 – Les rayons lumineux dans l’océan

formant toujours un pic dans les bleus, seule la longueur d’onde principale est fournie (en nm) dans le tableau suivant :

Océan	Azimut	Longueur d’onde dominante (nm)
Océan Pacifique	61°	473
Océan Indien	80°	474
Mer Méditerranée	74°	473
Océan Atlantique	70°	483
Mer Baltique	55°	540
Golfe du Panama	34°	487,2
Côte Sénégalaise	<i>Nuageux</i>	512

Les longueurs d’ondes dominantes de ce tableau illustrent à la fois la rétro-diffusion d’énergie et la partie d’énergie réfléchie. Par la suite, nous utiliserons ces valeurs pour l’illumination (notées I_{sea}).

Notre modèle est en première approximation basée sur le modèle de Blinn [Bli77] utilisé dans le lancer de rayons. Hélas, l’utilisation de ce modèle comporte deux limitations majeures dans le contexte des scènes océanes. Tout d’abord, il utilise des coefficients de réflexion et de transmission qui ne dépendent pas de l’angle d’incidence de la lumière ce qui est limitatif pour le rendu des scènes océanes. Nous utiliserons donc explicitement les coefficients de Fresnel. D’autre part, comme nous l’avons vu dans la partie précédente, la couleur des océanes est uniquement donnée en fonction de la longueur d’onde dominante bleue. Ceci est donc un obstacle majeur à l’utilisation d’un modèle RGB. Enfin, cette caractéristique de la couleur est donnée pour une profondeur que l’on peut supposer infinie. Ceci empêcherait même dans le cadre d’un modèle RGB adapté de pouvoir visualiser des objets se trouvant sous la surface de l’eau. Nous utiliserons donc, sous l’eau, un modèle basé sur une approche spectrale. Ainsi, nous pouvons qualifier notre modèle d’illumination de modèle hybride.

Modèle de Blinn pour le lancer de rayons Nous commençons par présenter comment est utilisé le modèle d’illumination local de Blinn dans l’illumination globale de type lancer de rayons. L’intensité I , codée par les trois composantes R,G et B, arrivant à l’oeil de l’observateur est donnée par :

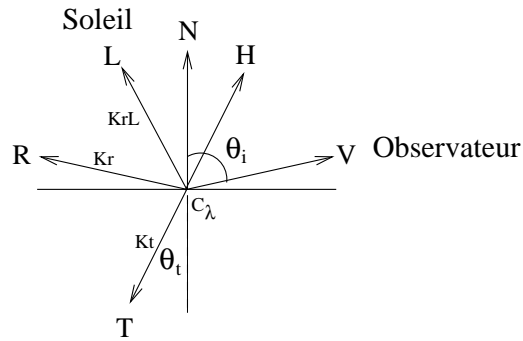


FIG. 71 – Illumination locale

$$I = I_a k_d C_\lambda + K_\alpha I_\alpha (k_d C_\lambda (N \cdot V) + k_s (N \cdot H)^p) + K_r I_r + K_t I_t \quad (75)$$

Voici un bref descriptif de chacun des termes utilisés dans l'équation 75 :

- I_a : Intensité RGB de la lumière ambiante,
- I_r : Intensité RGB de la lumière provenant de la partie réfléchie,
- I_t : Intensité RGB de la lumière provenant de la partie transmise,
- I_α : Intensité RGB de la source lumineuse,
- k_d : Coefficient diffus de l'objet,
- k_s : Coefficient spéculaire de l'objet, notons de plus que $k_d + k_s \leq 1$,
- C_λ : Couleur RGB intrinsèque de l'objet
- K_α : Coefficient d'absorption de l'illumination directe
- K_r : Reflectance de l'objet,
- K_t : Transmittance de l'objet, notons de plus que $K_r + K_t \leq 1$.

3.4.2 Calcul des coefficients de réflexion et de transmission

Lorsqu'un rayon lumineux arrive perpendiculairement à la surface de l'océan, presque toute son énergie est transmise dans l'océan. Par contre, si le rayon lumineux arrive presque parallèle à la surface de l'océan, la plupart de son énergie sera réfléchie et, seule une partie minime sera transmise dans l'eau. Entre ces deux cas extrêmes, la réflectance (K_r) et la transmittance (K_t) varient de manière continue. Ils dépendent de la différence d'indice de réfraction entre l'air ($n = 1$) et l'eau ($n = 1.31$). Les lois de Fresnel permettent de déterminer K_r et K_t par :

$$K_{r1} = \frac{n_1 \cos \theta_i - n_2 \cos \theta_t}{n_1 \cos \theta_i + n_2 \cos \theta_t}$$

$$K_{r2} = \frac{n_2 \cos \theta_i - n_1 \cos \theta_t}{n_2 \cos \theta_i + n_1 \cos \theta_t}$$

$$K_r = \frac{K_{r1}^2 + K_{r2}^2}{2} \quad (76)$$

$$K_t = 1 - K_r$$

où θ_i est l'angle d'incidence du rayon et θ_t son angle de réfraction calculé par la loi de la réfraction de Descartes.

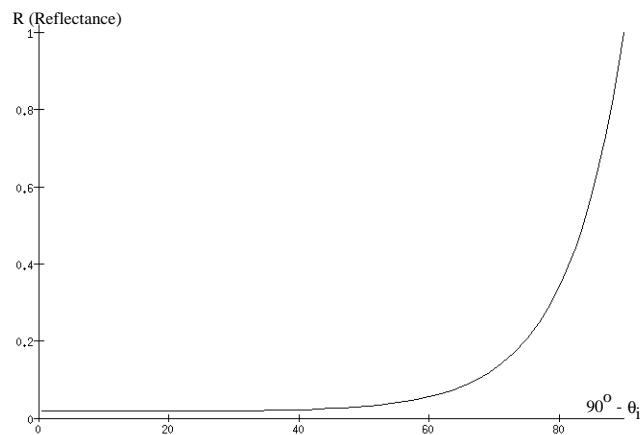


FIG. 72 – Valeur de la réflectance en fonction de l'angle d'incidence du rayon

Le calcul des coefficients de Fresnel va intervenir à deux moments dans le calcul de l'illumination locale. Dans un premier temps, nous calculons la part d'énergie provenant des sources lumineuses (K_{rL} fig 71). Dans un deuxième temps, nous calculons la part d'énergie provenant du fond marin ainsi que la part provenant de la réflexion dépendant de θ_i (K_r et K_t fig 71).

3.4.3 Atténuation de la lumière

Un rayon de lumière se propageant sous l'eau subit à la fois les phénomènes de diffusion et d'absorption au contact des particules en suspensions. La composée de ces deux phénomènes est appelée atténuation de la lumière. Généralement le coefficient d'atténuation K_α est fonction de la distance parcourue dans le milieu et est calculé comme suit dans un milieu homogène et isotrope :

$$K_\alpha = e^{-\alpha d} \quad (77)$$

où α est un coefficient d'atténuation du rayon lumineux et d la distance entre l'objet de la scène et, par exemple, l'observateur.

De manière classique, le rendu d'un objet immergé dans un milieu dit participant, se fait en remplaçant progressivement une partie de sa couleur⁶ par la couleur du milieu. Ceci ne fonctionne pas dans le cas particulier du rendu des objets sous l'eau. En effet, utiliser un seul coefficient d'atténuation par composantes RGB ($\alpha(R), \alpha(G), \alpha(B)$) implique que seules les longueurs d'ondes correspondant aux composantes RGB de l'objet sont atténuées. Dans notre cas, si on utilise un coefficient d'atténuation fort pour les rouges, plus faible pour les verts et enfin très faible pour les bleus, la couleur résultant aura toujours un pic autour de la longueur d'onde correspondant à la composante primaire bleue. La couleur de l'océan n'est pas rendue correctement. Il faut donc utiliser le spectre complet d'absorption de l'eau de mer.

Le graphique suivant représente les différents spectres visible sous l'eau en fonction de la profondeur lorsqu'une lumière blanche heurte la surface et se propage dans les fonds marins. Le spectre visible est compris entre 380 et 780 nm.

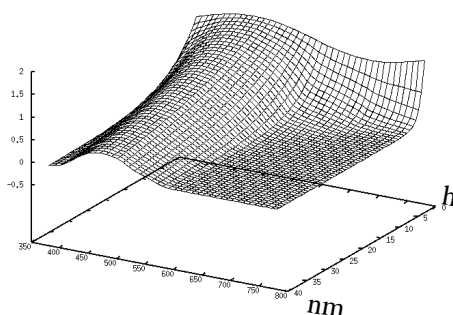


FIG. 73 – Evolution du spectre d'une lumière blanche en fonction de la profondeur

Dans le graphe 73, nous utilisons le spectre complet d'absorption. On remarque rapidement que les bleus (autour de 470 nm) sont moins atténués que les autres longueur d'onde. A une profondeur de 40m, la couleur de l'océan comporte encore une composante bleue.

Il n'existe pas de spectre unique d'absorption de l'océan car il dépend fortement de la nature, de la taille et de la quantité de particules en suspensions. Le spectre d'absorption utilisé (fig 74) est présenté dans [Iva75]. Il définit pour chaque longueur d'onde du spectre visible la valeur de α de l'équation 77.

3.4.4 Le modèle de couleur utilisé

Pour le rendu de l'océan, nous allons utiliser un modèle de couleur hybride utilisant en partie l'espace colorimétrique RGB et en partie le modèle spectral. En effet, pour tous les objets situés au dessus de l'océan, le modèle RGB est suffisant. Pour l'océan, nous utilisons le modèle RGB s'il n'y a pas d'objets sous l'eau. Si un objet sous marin est présent, le modèle spectral nous permet de tenir compte de l'absorption réelle de la lumière sous l'eau.

Pour manipuler les deux modèles dans le même moteur de rendu, nous utilisons des algorithmes de conversions RGB-spectral et spectral-RGB. Le passage du modèle RGB au spectral est proposé par

⁶Elle est alors atténuée.

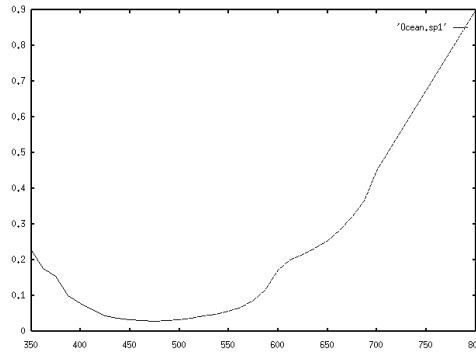


FIG. 74 – Le spectre d’absorption de l’eau de mer.

Glassner[Gla88]. La transformation d’un modèle spectral vers un modèle RGB a été développée par Roy Hall [Hal89]. L’ensemble des composantes spectrales est recodé dans l’espace colorimétrique virtuel XYZ puis dans l’espace réel RGB.

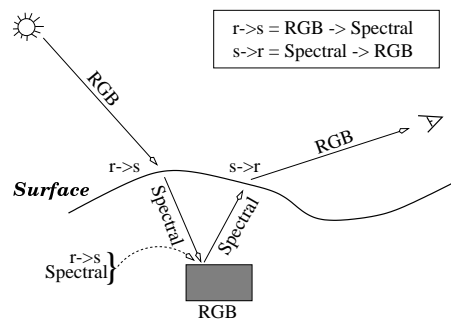


FIG. 75 – Modèles de couleur pour les objets sous l’eau

Si aucun objet ne se trouve sous l’eau, nous choisissons directement la couleur de l’océan donnée par le tableau 3.4.1 qui est converti en RGB avant la phase de rendu. Pour transformer la longueur d’onde fournie par le tableau en couleur RGB, Nous construisons, dans le modèle spectral, une gaussienne centrée autour de cette longueur d’onde. Une largeur de $20nm$ à mi-hauteur de la gaussienne nous semble être empiriquement une bonne valeur.

Par contre, lorsque un objet se trouve sous l’eau, nous devons calculer la couleur réelle de cet objet telle que pourra l’apercevoir l’observateur. Ainsi, dans un premier temps, l’objet est illuminé en calculant la couleur réelle de la source lumineuse touchant l’objet. La couleur RGB de la source lumineuse est donc transformée dans son modèle spectral. L’objet, ayant aussi une couleur définie par les composantes RGB, est transformé dans son modèle spectral. L’illumination locale est effectuée. Finalement, le spectre lumineux revenant à la surface est alors re-converti en RGB (fig 75).

3.5 Résultats

En guise de résultats nous présentons une série d'images couleurs. Les figures 76 et 77 présentent les variations de couleur d'une lumière blanche pénétrant dans l'océan. La figure 76 correspond aux profondeurs allant de 0 à $-15m$ et la figure 77 de 0 à $-50m$. La couleur en fonction de la profondeur est déterminée à partir du spectre complet d'absorption de l'eau donné figure 74.

Les trois images suivantes (78, 80 et 81) présentent trois rendus de l'océan. La première permet de visualiser un recouvrement de plusieurs trains de vagues. La seconde propose une vue générale de jour d'une scène comportant, entre autres, des vagues capillaires. La partie de l'objet sous l'eau est vue par réfraction. Enfin, la troisième image propose une vue générale d'océan de nuit. L'éclipse solaire est une texture qui a été plaquée sur une facette verticale en fond de scène. Une lumière virtuelle a alors été rajoutée devant l'éclipse.

La figure 80 de taille 1024×675 a été générée en 30 minutes sur une Silicon Graphics O2. Nous avons utilisé 8 échantillons par pixels. Le temps de création de l'image 81 de taille 1365×2048 est de l'ordre de 4 heures sur une Silicon Graphics O2. Nous avons utilisé un lancer de rayon adaptatif pour limiter les effets de l'aliasing. Le nombre de rayons par pixel est compris entre 4 et 40.



FIG. 76 – Transmission d'une lumière blanche jusqu'à 15 mètres



FIG. 77 – Transmission d'une lumière blanche jusqu'à 50 mètres



FIG. 78 – Multiples trains de vagues

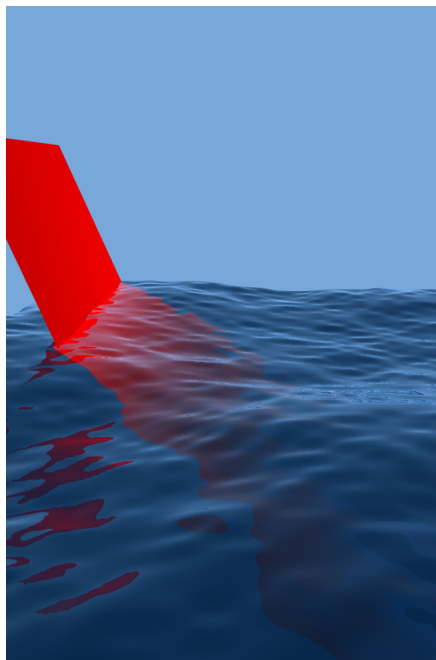


FIG. 79 – Visualisation de l'absorption de la lumière avec un objet plongé dans l'eau

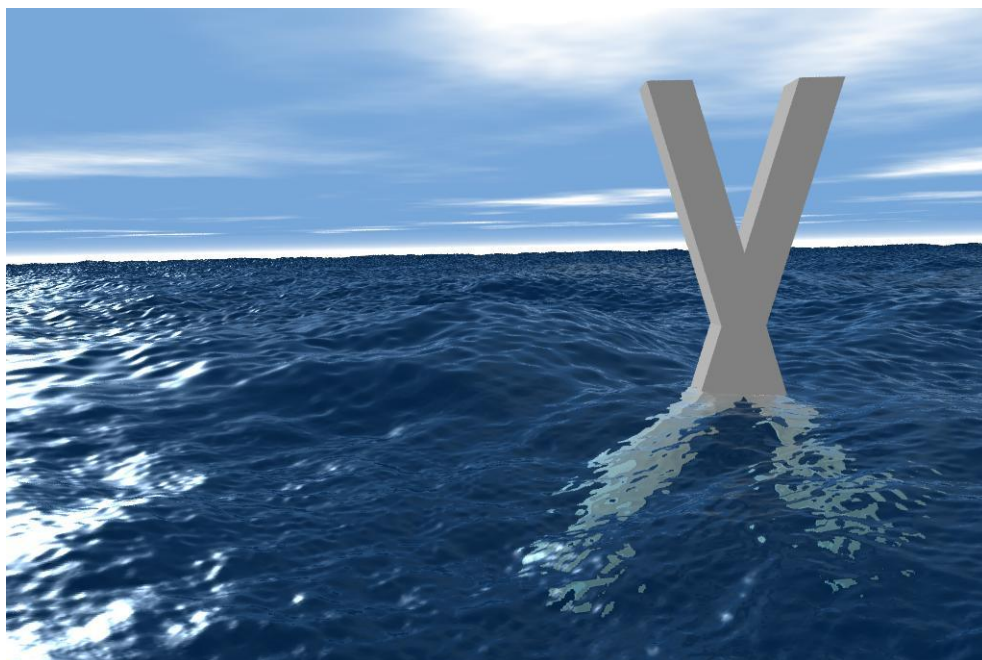


FIG. 80 – Vue générale de l'océan de jour



FIG. 81 – Coucher de soleil sur l'océan

Chapitre 4

Le lancer de faces

La méthode du lancer de vagues dynamique présentée dans le chapitre 2 est un algorithme 2D. Un réflexe naturel est d'en envisager une version 3D de celui-ci. On obtient ainsi un nouveau type de lancer de rayons que nous appellerons le *Lancer de Faces*. Cet algorithme est une méthode de Monte Carlo et utilise une méthode spécifique de stratification du pixel basée sur la géométrie de la scène¹.

4.1 Introduction

Le lancer de rayons fut développé par Whitted [Whi80], méthode présentée par Appel [App68]. Les rayons sont lancés au travers des pixels de l'image à calculer à partir de l'oeil de l'observateur. Ils se propagent dans la scène en suivant le sens inverse du parcours de la lumière. Les rayons ne sont donc pas émis de manière parallèle, mais divergent les uns des autres au cours de leur propagation. Cet algorithme a subi de nombreuses améliorations dont le but est soit de limiter l'aliasing que nous appellerons abusivement aliasing, notion détaillée dans le paragraphe suivant, soit d'augmenter le réalisme des images produites.

Pour définir l'aliasing, plaçons nous dans le cas d'une ligne d'image à calculer. Les intensités lumineuses arrivant sur cette ligne peuvent être considérées comme un signal dont la fréquence n'est pas connu mais peut-être grande. Le théorème de Shannon [Sha49] précise que pour échantillonner ce signal, la fréquence d'échantillonnage doit être supérieure à deux fois la fréquence maximale du signal original. Ceci implique donc que cette fréquence d'échantillonnage, dans notre cas, doit être infinie pour espérer reproduire tous les signaux possibles. Ceci étant impossible, l'échantillon ne permet donc pas de reconstruire le signal d'origine : c'est l'aliasing.

Sur l'image 82.a, l'aliasing correspond d'une part, aux "marches d'escaliers" visibles à la frontière de deux carreaux du damier situés proche de l'observateur et d'autre part à la mauvaise reconstruction du damier situé loin de celui-ci (image du milieu). Dans l'image de droite, les artéfacts apparaissent à une moindre échelle. Elle est dite "anti-aliasée".

Par conséquent, s'il est clair que l'aliasing est inévitable dans notre contexte, on peut espérer le diminuer en augmentant la fréquence d'échantillonnage. Mais dans le cas d'une scène réaliste avec une multitude

¹Deux versions initiales de cet algorithme sont présentées dans [GS98] et [GS99a].

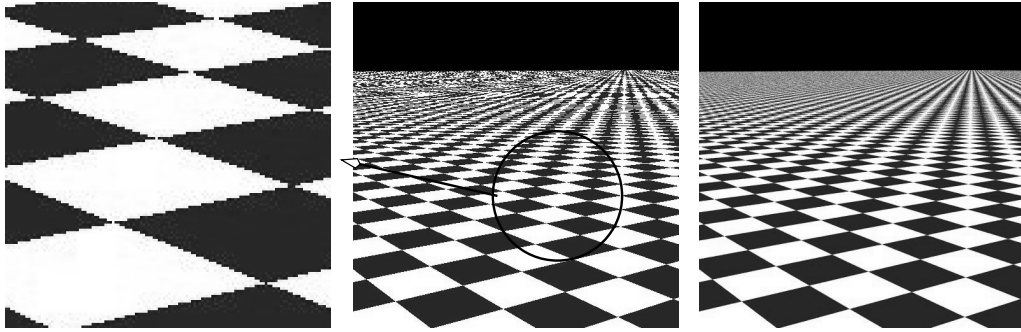


FIG. 82 – Une image "aliasée" et deux images "anti-aliasée" à des niveaux différents.

de détails, il est généralement extrêmement coûteux d'échantillonner à une fréquence telle que l'aliasing peut être considéré comme négligeable. Il faut donc trouver des techniques - dite d'anti-aliasages - permettant d'effectuer un choix judicieux des échantillons du signal pour réduire au maximum l'aliasing tout en limitant les coûts de calculs.

Dans ce type de techniques, le sur-échantillonnage peut avoir lieu à plusieurs endroits de la scène : soit, évidemment à partir de l'oeil de l'observateur et donc du pixel, soit à partir du point d'intersection entre un rayon et un objet. Dans ce dernier cas, le sur-échantillonnage permet de limiter l'aliasing pour les rayons réfléchis, les rayons transmis et enfin pour les rayons permettant de calculer l'ombrage au point d'intersection.

Les méthodes de sur-échantillonnage peuvent être découpées en deux grandes familles :

- Les méthodes utilisant un sur-échantillonnage fixe. Dans le cas du pixel, il est découpé en zones régulières (16×16 par exemple) et un rayon est envoyé au centre de chaque zones.
- Les méthodes utilisant un sur-échantillonnage adaptatif. Le pixel est découpé en un nombre faible de zones (4 par exemple), un rayon est envoyé dans chaque zone. Suivant un certain critère de raffinement, chaque zone pourra être sur-échantillonnée récursivement. Les questions sont alors de savoir quel est le critère de raffinement à utiliser, comment subdiviser la zone et enfin où envoyer le rayon dans la zone subdivisée.

Entre autres, Whitted [Whi80] sur-échantillonne l'image en envoyant plus d'un rayon par pixel en utilisant une subdivision hiérarchique du pixel. Heckbert et Hanrahan [HH84] ont développé le "Beam Tracing" et Amanatides [Ama84] le "Cone Tracing" afin d'éviter l'aspect infinitésimal du rayon (Le terme anglais approprié est le "*Point Tracing*"). Enfin, le lancer de rayons stochastique [RLCC84] a été proposé. Il améliore l'algorithme de Whitted en lançant des rayons aléatoirement dans le pixel. Il évite ainsi de privilégier les composantes verticales et horizontales de l'image. Ces diverses méthodes sont détaillées dans la partie concernant les travaux précédents.

Nous proposons une méthode adaptative basée sur les méthodes de Monte Carlo (voir §4.3) où de nouveaux rayons sont générés au besoin. A l'inverse des méthodes précédentes, ces nouveaux rayons ne démarrent pas tous à partir de l'oeil de l'observateur ou à partir d'un point d'intersection avec un objet. De manière analogue au lancer de vagues vu au chapitre 2, ils sont lancés au point de sur-échantillonnage dans la scène. Notre algorithme qui est donc une généralisation 3D du lancer de vagues doit résoudre un certain

nombre de problèmes :

- Quelle notion de voisinage utiliser pour les rayons ?
- Quel critère utiliser pour conduire le raffinement et comment l'effectuer ?
- Comment contrôler l'explosion combinatoire ?
- Comment calculer la couleur d'un pixel ?

Ceci nous a conduit à interpréter les rayons lancés à partir de l'observateur non plus comme les axes d'angles solides échantillonnant l'espace, mais comme "les coins" d'une facettisation de la pyramide de vue. Notre problème se re-pose en fait en termes de lancer de faces (triangulaires). Lancer une face est alors équivalent à lancer les trois rayons portés par les sommets de la face. Nous disposons ainsi d'un algorithme spécifique de stratification (explications dans §4.3) du pixel. Cette stratification est basée sur la géométrie de la scène.

La notion de voisinage devient alors naturelle. Les rayons sont voisins s'ils sont situés sur une même face. Les critères de raffinement deviennent alors aussi naturels en utilisant les faces. Le premier est un seuil surfacique : si la face devient trop grande, elle est subdivisée. Le second doit subdiviser toute face qui touche partiellement un objet. Enfin, le troisième subdivise une face où de l'aliasing de texture se produit.

De manière analogue au lancer de rayons, l'explosion combinatoire peut être contrôlée arbitrairement en fixant un nombre maximal de subdivisions tolérées. La couleur d'un pixel sera calculée par un moyennage des rayons portés par les sommets de chaque face. L'aire de la face pondèrera le poids de celle-ci dans le pixel.

L'algorithme de lancer de faces va donc capturer les informations sur la scène au cours de la progression des faces. Dès que l'aire d'une face est supérieure à un certain seuil, l'échantillonnage effectué à cet endroit peut ne plus être assez précis. Nous rétablissons un taux d'échantillonnage correct en subdivisant la face. Ceci nous garantit une détection fixe de l'ensemble des objets situés dans la pyramide de vue. Cet algorithme est particulièrement adapté à certains types de scènes :

- Les scènes comportant des objets de mêmes tailles mais situés à des distances très différentes de l'observateur,
- Les scènes comportant un mélange de très gros objets et d'autres très petits. Par exemple, on peut citer une photo où l'on verrait un vol d'oiseaux entre des immeubles.
- Des scènes où un grand nombre de réflexions doivent être prises en compte pour un rendu réaliste.
- Des scènes où des objets semi-speculaires sont présents dans la scène.

Un tel algorithme peut être employé là où les algorithmes classiques de rendu sont inopérants. On pourra citer entre autres :

- *Le rendu des mirages* [BLT90]. Ceux-ci se produisent lorsque des variations brutales de température sont présentes dans la scène. L'indice de réfraction de l'air est fonction de la température. Les rayons de lumière traversant ce milieu sont alors déviés. On peut donc faire une complète analogie entre la réfraction des vagues et la réfraction de la lumière. Ainsi, deux rayons voisins peuvent prendre des chemins très différents à la sortie de la couche de chaleur. La partie des objets située entre ces deux rayons ne sera alors pas rendue.
- *Le rendu des diamants et autres pierres précieuses* [YYS88]. Deux rayons voisins traversant un diamant suivront des chemins complètement différents, en fonction des multiples réflexions, à l'intérieur

de celui-ci. Les lois de dispersion du diamant est parfaitement connue. Par contre, celles des autres gemmes le sont en partie et sont sujettes à de grande variabilités comme la bi-réfringence par exemple. Ainsi, notre algorithme pourra tenir compte efficacement des deux ou trois indices de réfraction décrivant les phénomènes optiques du gemme. On limitera ainsi la divergence des rayons pour avoir un rendu plus précis.

- *L’endoscopie virtuelle.* L’endoscopie virtuelle est une technique qui permet de progresser dans un organe virtuel. Avec l’endoscopie, le chirurgien visualise l’intérieur d’un organe grâce à une fibre optique. Il voit une image mirage de l’organe causée par les multiples réflexions/réfractions de la lumière dans la fibre.
- *Le rendu des objets semi-spéculaires, semi-transparents.* Si un rayon de lumière intersecte ce type d’objet, il y a une multitude de rayons réfléchis ou transmis. Avec notre modèle, nous pouvons lancer une face depuis le point d’intersection dans la direction principale de réflexion ou transmission et ainsi l’aspect flou de l’objet sera rendu.

Dans la suite de ce chapitre, une présentation des méthodes existantes est effectuée. Notre algorithme de Lancer de Faces étant un type de lancer de rayons de Monte Carlo, nous effectuerons une présentation plus précise de cet ensemble de méthodes. Ensuite, nous détaillerons pas à pas notre algorithme.

Finalement, nous présenterons l’application de notre algorithme à la génération des mirages.

4.2 Travaux précédents

Dans cette partie, nous allons détailler les principaux travaux basés sur l’algorithme du Lancer de Rayons classique. Cette liste ne sera pas exhaustive car trop grand est le nombre de travaux traitant du lancer de rayons et de ses applications. Nous nous en tiendrons juste aux méthodes majeures.

Whitted [Whi80] et Painter [PS89] sur-échantillonnent l’image en lançant plusieurs rayons par pixel en utilisant une subdivision régulière hiérarchique du pixel. Le critère de subdivision est le gradient de couleur entre deux zones d’un même pixel. Comme la surface qu’explore un rayon dans la scène augmente avec la distance, les problèmes d’aliasing sont ainsi simplement repoussés. De plus, étant donné que la subdivision est régulière et que le rayon est centré par rapport à la zone, les lignes horizontales et verticales sont ainsi privilégiées. Cet algorithme ne tient pas compte de la structure de la scène.

Heckbert et Hanrahan [HH84] décomposent la pyramide de vue en envoyant des faisceaux de rayons. Ainsi, ils tiennent compte de l’aspect non singulier d’un rayon mais plutôt de l’angle solide qu’il représente. Cette méthode gère donc le problème de l’aliasing en tenant compte de l’aire que représente un pixel. Malheureusement, une nette augmentation de complexité dans les calculs d’intersections est alors introduite. De plus, la scène est limitée à des objets polygonaux. Heckbert présenta des simplifications pour tenir compte des phénomènes de réfraction pour les objets semi-transparents. En effet, l’aspect de faisceau n’est généralement pas préservé après une intersection.

Une version voisine du lancer de faisceaux est le lancer de pinceaux [MSN87]. Des rayons parallèles sont envoyés depuis l’observateur et se propagent dans la scène. Malheureusement, le pouvoir discriminatif d’un rayon diminue toujours avec la distance à l’observateur.

Le lancer de cônes fut développé par Amanatides [Ama84]. La pyramide de vue est alors décomposé

en cônes partant de l'observateur en passant à travers le pixel. Dans cette méthode, les objets de la scène sont limités à des objet polygonaux ou des sphères. Cette méthode permet de tenir compte efficacement de la zone représenté par un rayon. Par contre, on peut signaler que la base du cone (un cercle ou une ellipse) ne permet pas d'effectuer une tessellation correcte de l'image. De plus, les calculs d'intersection sont complexes à gérer. En effet, si un cône heurte partiellement un objet, il faut décomposer la base du cône en sous parties qui progresseront indépendamment dans la scène.

Finalement, les produits commerciaux utilisent généralement le lancer de rayons stochastique en échantillonnant les pixels par les méthodes de Monte Carlo. Cette technique est détaillée dans la partie suivante.

4.3 Lancer de rayons de Monte Carlo

Calculer la couleur d'un pixel revient à faire l'intégrale sur sa surface de l'énergie lumineuse y arrivant. L'évaluation de cette intégrale par méthode de Monte Carlo consiste à choisir une variable aléatoire dont l'espérance mathématique est cette intégrale et de choisir de manière aléatoire des échantillons vérifiant la loi de probabilité de cette variable. La loi des grands nombres indique que la moyenne des échantillons va tendre vers l'espérance mathématique et donc le résultat recherché [HH64].

Pour le lancer de rayons et le calcul de la couleur d'un pixel, la variable aléatoire choisie est le rayon de lumière passant par ce pixel. La loi de probabilité de cette variable est une constante pour l'ensemble des rayons. Classiquement, les rayons sont donc envoyés depuis l'oeil de l'observateur de manière complètement aléatoire ou pseudo-aléatoire dans le pixel. Le calcul de la couleur est l'espérance mathématique de toutes les couleurs des rayons envoyés. Comme le poids de chaque rayon est identique, le calcul de la moyenne des couleurs portées par les rayons du pixel est équivalent au calcul de l'espérance mathématiques.

Ce principe de base a subi de nombreuses évolutions (choix du nombre de rayons par pixel, manière de positionner les rayons sur la surface, choix du critère de raffinement), ce qui explique l'imposante bibliographie dans ce domaine.

Modèle de Schlick. Dans le cadre de cette thèse, nous ne détaillerons que le modèle de Schlick [Sch92] car il allie l'ensemble des techniques permettant de minimiser l'aliasing². Sur la base d'un lancer de rayons, la méthode d'échantillonnage du pixel est de type Monte Carlo. Elle est adaptative (le nombre de rayons est différent pour chaque pixel de l'image) et l'échantillonnage est stratifié. La stratification permet de répartir équitablement l'ensemble des rayons sur une zone du pixel. Pour cela, si N rayons doivent être lancés dans le pixel de surface S , celui-ci est découpé en N zones de surface $\frac{S}{N}$ et un rayon est envoyé par zone. Ainsi, cette technique permet de réduire la variance des méthodes de Monte Carlo et d'accélérer leur convergence. Si un échantillon est envoyé dans chaque strate, la stratification est dite complète.

Chaque pixel de l'image à générer est découpé en quatre sous-pixels de surface égale. La couleur du pixel est alors calculée par moyennage des quatre couleurs retournées. Dans chacun des sous-pixels, un rayon est envoyé.

²Ce modèle est à la fois régulier, irrégulier, stratifié, pondéré, non-corrélé et adaptatif. Nous invitons le lecteur à consulter la thèse de cet auteur pour une définition de chacun de ces termes.

Critère de raffinement. Pour limiter l'aliasing, il est alors possible de passer par une étape de raffinement d'un sous-pixel, i.e. envoyer plusieurs rayons pour ce sous-pixel. Le critère de raffinement choisi est une version modifiée de la notion de *contraste* définie dans [Mit87]. Cette notion possède l'avantage de suivre approximativement la courbe de sensibilité de l'oeil. Le calcul du contraste C s'effectue entre chaque paire de sous-pixels voisins (4-connexes ou 8-connexes) et est égal à :

$$C = \sqrt{C_R^2 + C_G^2 + C_B^2} \quad (78)$$

avec,

$$C_\lambda = \frac{\max(I_\lambda, I'_\lambda) - \min(I_\lambda, I'_\lambda)}{\max(I_\lambda, I'_\lambda) + \min(I_\lambda, I'_\lambda)} \quad (79)$$

où I_λ et I'_λ sont les intensités lumineuses des composantes λ pour les deux sous-pixels.

Méthode de raffinement. Le raffinement doit être effectué tant que le contraste entre deux sous-pixels est supérieur à un certain seuil. Pour cela, dans chaque sous-pixel est stocké à la fois la couleur et son niveau de raffinement. Chaque sous-pixel est comparé au voisin ayant crée le contraste maximal et on choisit de sur-échantillonner le sous-pixel ayant le niveau de raffinement minimal. Passer d'un niveau de raffinement N à un niveau $N + 1$ consiste à envoyer un nouveau rayon dans la scène passant par ce sous-pixel.

Une méthode de stratification du sous-pixel est effectuée suivant une amélioration du modèle de [Shi90] (voir méthode dans [Sch92]). Cette méthode permet une stratification complète du sous-pixel à chaque niveau de raffinement.

Calcul de la couleur d'un sous-pixel après une étape de raffinement Soit N le niveau de raffinement d'un sous-pixel, I_N sa couleur et I_{new} la couleur de retour du nouveau rayon lors de la phase de raffinement. La couleur après sur-échantillonnage se détermine par :

$$I_{N+1} = \frac{N \times I_N + I_{new}}{N + 1} \quad (80)$$

Avec ce calcul, le stockage de l'ensemble des couleurs des rayons du sous-pixel n'est plus nécessaire. Seule la valeur courante du sous-pixel est stockée.

4.4 Le Lancer de Faces

4.4.1 Description générale de l'algorithme

L'algorithme de Lancer de Faces que nous présentons ici, est un modèle de rendu basé sur le lancer de rayons de Monte Carlo. Il est adaptatif et utilise un échantillonnage stratifié original qui prend en compte la géométrie de la scène. Cette stratification limite ainsi l'aliasing géométrique³.

La pyramide de vue est échantillonnée par des faces triangulaires. Pour simplifier l'algorithme, cet échantillonnage sera appliqué sur l'image à générer. Les pixels de l'image sont découpés en faces triangulaires. Le découpage initial est arbitraire. Les faces progressent pas à pas dans la scène et le pas de

³L'aliasing géométrique se produit lorsque une partie des objets visibles de la scènes n'est pas détectée et rendue

progression est calculé en fonction de l'agrandissement de chaque face et de la distance à l'observateur. Les rayons progressent simultanément dans la scène.

Une face est décrite par les 3 rayons passant par chacun de ses sommets. Nous utilisons l'aire de chaque face pour calculer la divergence des rayons dans la scène. Lorsque cette aire devient trop importante, nous la subdivisons en 4 sous-faces en envoyant 3 nouveaux rayons. Ces rayons ont pour origine un point aléatoire sur chaque arête de la face.

D'autres événements influent sur l'histoire d'une face et en particulier, les éventuelles intersections avec les différents objets de la scène :

1. Si les trois sommets de la face touchent un même objet dans le même pas de calcul, la face est réfléchi (resp. réfractée) en fonction des propriétés optiques de l'objet.
2. Si uniquement un de ses trois sommets touche un objet, la facette est automatiquement subdivisée.
3. Si deux sommets de la face intersectent un même objet, nous sommes dans une situation duale à la précédente.
4. Finalement, si deux (resp. trois) sommets touchent deux (resp. trois) objets différents, la face est alors subdivisée.

Dans les trois cas précédents, une gestion particulière des facettes frontières est effectuée.

La progression des faces s'interrompt donc soit lorsqu'une intersection partielle apparaît, soit lorsque le niveau maximal de subdivisions est atteint. L'abandon de la progression d'une face ne termine pas obligatoirement la progression des rayons qui la définissent. Ceux-ci continuent à progresser dans la scène en suivant un algorithme de lancer de rayons classique. L'intersection d'une face avec un objet de la scène est définie comme étant l'intersection des rayons de ses sommets avec cet objet.

Un sommet qui progresse dans la scène appartient à plusieurs faces. Ceci induit un déplacement simultané de toutes les faces qu'il porte. Nous utiliserons une structure de données adaptée pour gérer ces faces.

Le calcul de la couleur d'une face passe par le calcul de la couleur de ses rayons. Leur couleur est calculée de manière usuelle (Phong [Pho75], Cook-Torrance [CT81], ...). La couleur de la face est calculée par moyennage des couleurs des sommets. La couleur d'un pixel dépendra du poids de chaque face. Ce poids est lié au niveau de subdivision de la face.

Après avoir fait un survol rapide de la méthode, nous allons, dans les paragraphes suivants détailler le lancer de faces.

4.4.2 La scène

Pour appliquer d'efficaces lancer de rayons incrémentaux [AW87], les scènes sont découpées en voxels par une grille uniforme 3D. Ceci permet, de plus, d'optimiser le nombre de calculs d'intersections. Par souci de simplicité, chaque pixel est découpé en deux faces triangulaires dont les sommets sont les coins du pixel (fig. 83). La pyramide de vue est donc complètement facettisée.

Par la suite, les croquis illustrant nos explications ne feront apparaître qu'une seule face provenant soit d'une des deux faces primaires, soit d'une subdivision (ex fig 83).

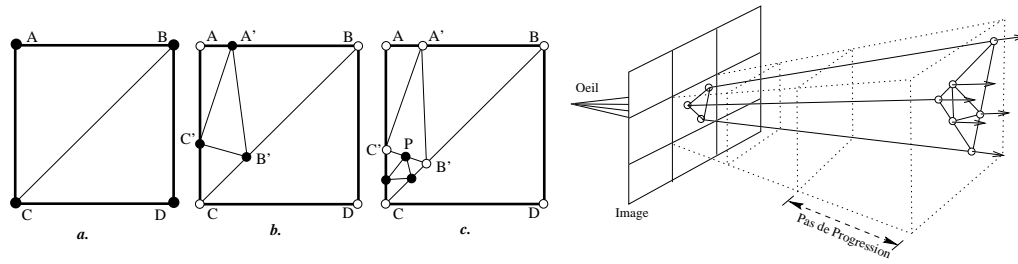


FIG. 83 – Méthode de subdivision des facettes

4.4.3 Pas de progression des faces

Les faces progressent dans la scène pas à pas. Ce pas est prédéterminé. En effet, si la face n'intersecte aucun objet, on connaît à l'avance l'aire après un pas de progression. Ce calcul peut aussi se faire en sens inverse en imposant un pas tel que la surface atteigne la surface maximale avant subdivision. La face est alors subdivisée à chaque pas. Nous rappelons que cette subdivision est un critère pour la détermination des petits objets. Le calcul du pas peut en fait se faire en ne tenant en compte que la longueur des arêtes. Nous appliquerons donc la formule suivante :

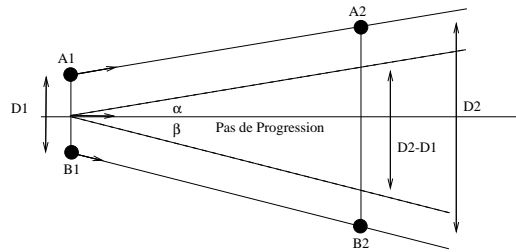


FIG. 84 – Pas de calcul

$$\text{Pas de progression} = \frac{D2 - D1}{\tan(\alpha) + \tan(\beta)} \quad (81)$$

Si un sommet touche un objet entre deux pas de progression, il sera alors stoppé au point d'intersection (fig. 85 croquis du haut). Les deux autres sommets atteindront le pas prédéfini. Dans ce cas, nous effectuons au préalable une subdivision de la face afin de prévenir les phénomènes d'aliasing (fig. 85 croquis du bas). Les traitements effectués sur chacune des faces après intersection seront vus dans la partie §4.4.5. Dans le cas de la figure 85.B, la face réfléchie, ainsi que toutes ses futures subdivisions, auront un pas de progression différent des faces passant à côté de l'objet.

4.4.4 Subdivision d'une face

Lorsque les rayons divergent les uns des autres au cours de leur progression dans la scène, l'aire de chaque face augmente. Lorsqu'elle devient trop grande, la zone explorée par les rayons est trop importante pour garantir une bonne précision et qualité de l'image, la face est alors subdivisée en quatre sous-faces.

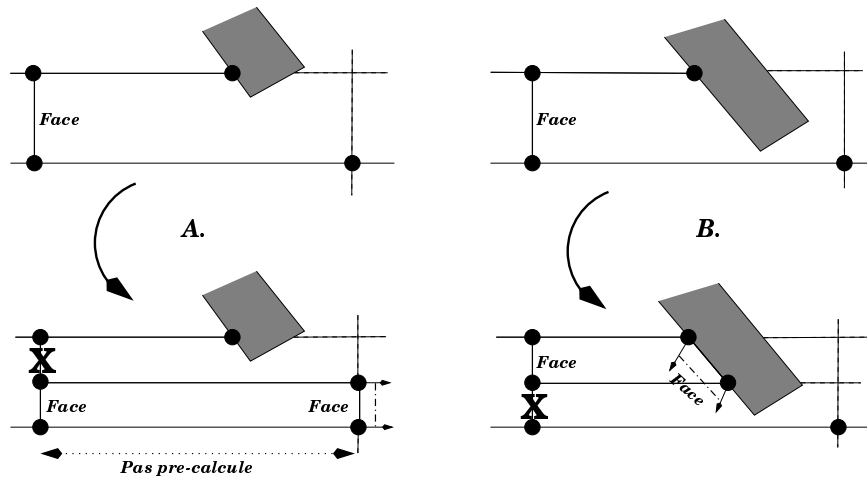


FIG. 85 – Progression avec une intersection

Cette décomposition est obtenue en envoyant trois nouveaux rayons sur chacune de ces arêtes. Le point de départ de chacun d'eux est choisi aléatoirement sur chacune des arêtes de la face. Ainsi, les sommets sont plus proches les uns des autres et nous garantissons une qualité identique de détection des objets de la scène. Une limitation de l'aliasing géométrique est donc faite.

L'aire des faces primaires est de $\frac{\text{Aire_du_Pixel}}{2}$. Cette aire définit la période d'échantillonnage de la scène. Elle nous sert de base à la détermination du seuil pour lequel un sur-échantillonnage est nécessaire. Nous utiliserons donc :

$$\text{Seuil de subdivision} = 4 \times \frac{\text{Aire_du_Pixel}}{2} \quad (82)$$

Comme une face est divisée en quatre sous-faces, le facteur "4" nous permet de garder l'aire initiale de la face mère sur chaque face fille. Ceci donne donc un échantillonnage uniforme. Ce seuil prédéfini peut être modifiable par l'utilisateur en fonction des caractéristiques de la scène. Cette modification s'effectue sur la totalité de la scène.

Nous utilisons un Quad-Tree [Sam84] pour stocker l'ensemble des faces et des liens de parenté qu'elles ont entre elles. Chaque noeud et feuille du Quad-tree correspond à une face en progression dans la scène. Chaque feuille, i.e. chaque face minimale, est reliée aux trois sommets qui la composent.

Des liens supplémentaires entre les faces et certains sommets progressant dans la scène sont rajoutés. En effet, sur exemple de la figure 83.b, la face F (resp. G) est formée par les sommets $A'B'C'$ (resp. $C'B'C$). Lorsque G est subdivisée (fig 83.c), un lien entre F et P est rajouté dans la structure de données. Ainsi, si F doit être subdivisée plus tard dans la scène, il sera inutile de choisir un nouveau sommet sur l'arête $B'C'$. On prendra directement le sommet P déjà créé.

Lorsque une face est réfléchi (après intersection avec un objet), les caractéristiques du sous-arbre de cette face sont indépendantes des autres parts du Quad-Tree. Ceci permet d'assurer une progression simultanée et en parallèle de chaque groupe différent de faces.

Afin d'éviter une explosion combinatoire des calculs et du stockage, nous choisissons naturellement de

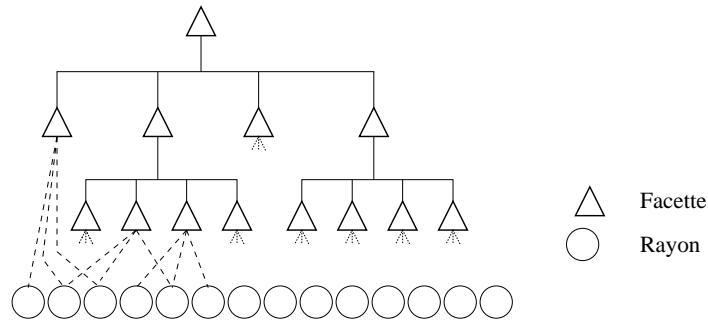


FIG. 86 – Un exemple de quadtree

limiter le nombre de subdivisions de chaque face en fixant une profondeur maximale du Quad-Tree. L'utilisateur va fixer cette profondeur pendant la phase de description des divers paramètres de son image (taille, caméra, ...). Lorsque le niveau maximal de subdivision est atteint, la gestion des faces devient alors inutile. Les sommets progressent alors indépendamment les uns des autres dans la scène en suivant l'algorithme de lancer de rayons classique.

4.4.5 Intersections avec des objets

Dans cette partie, les divers traitements effectués lorsqu'une face touche un ou plusieurs objets sont présentés. Evidemment, si la face ne touche aucun objet, elle poursuit sa progression dans la scène.

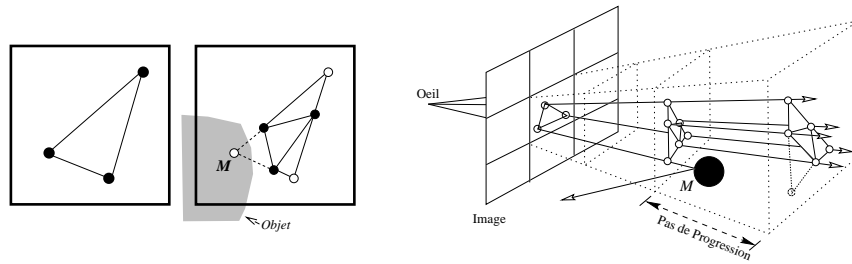


FIG. 87 – Intersection avec un objet spéculaire

Lors d'une intersection entre une face et un ou plusieurs objets, quatre cas dépendant soit du nombre d'objets touchés, soit du nombre de sommets qui touchent ce ou ces objets, doivent être traités. Cette intersection pourra mener à la subdivision de la face (fig 85).

- Si les trois sommets d'une face touchent un même objet, une face réfléchie (resp. réfractée) va être générée en utilisant les rayons réfléchis (resp. réfractés) des sommets de la face. Cette ou ces deux nouvelles faces vont être substituées à la précédente.
- Si un seul sommet touche un objet, un rayon réfracté peut être créé en fonction des propriétés optiques de l'objet rencontré. Dans ce cas, le sommet réfracté sera substitué au sommet incident et la face poursuivra son chemin dans la scène. Par contre, si un sommet réfracté n'est pas créé, la face sera d'abord subdivisée. En effet, le but de notre algorithme est à la fois de tenir compte des objets

dans la scène mais aussi de limiter l'aliasing sur la surface frontière. La face qui arrive sur l'objet n'a pas encore atteint son aire limite. Ceci signifie donc qu'elle répond bien aux critères de précision d'échantillonnage demandé. Par contre, il est évident qu'à la frontière d'un objet les différences de couleurs de deux rayons voisins (un touchant l'objet et pas l'autre) sont généralement très importantes. Aussi, l'échantillonnage doit être plus précis. Ainsi, nous sur-échantillons la face une fois. Dans l'exemple 88, les faces BPD , DPE et EPC sont subdivisées pour une meilleure détection des contours de l'objet. L'algorithme détruit automatiquement les faces frontières de la structure de données i.e. les faces contenant le point P . De la même manière, si, par la suite, la face ABC doit être subdivisée, seul AGH continuera sa progression dans la scène.

Si, de plus, un rayon réfléchi est créé au point P , il progresse seul dans la scène par l'algorithme classique de lancer de rayons de Whitted.

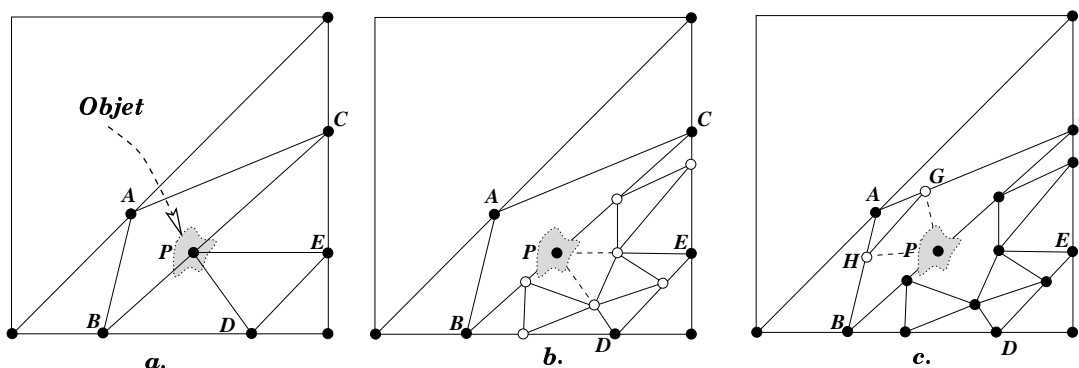


FIG. 88 – Subdivision après une intersection

- Si deux des trois sommets d'une face touchent un même objet, l'algorithme suit le même processus que celui expliqué dans le point précédent.
- Si deux ou trois sommets de la face touchent des objets différents, dans ce cas, il faut effectuer une première subdivision pour garantir une bonne détection des contours des différents objets. Les nouvelles faces sur le contour sont alors détruites. Pour détruire une face, nous supprimons sa représentation dans le Quad-Tree. Les sommets de la face détruite progressent soit avec les faces voisines soit seuls.

Pourquoi une seule subdivision pour la détection des contours ? Si la face arrive contre l'objet tel quel, soit elle est déjà à son niveau de récursion maximal et donc l'objet est trop petit pour être détecté correctement par l'oeil de l'observateur, soit l'aire de la face est en dessous de l'aire maximale choisie. Dans ce dernier cas, l'objet est à une telle distance qu'il pourra être observé et rendu avec une précision suffisante. Donc, simplement, pour limiter l'aliasing géométrique, nous décidons de subdiviser une fois de plus les faces.

Il reste à régler un cas limite présenté sur la figure 89. Si deux sommets d'une face touchent deux portions d'un même objet ayant une arête vive, premièrement, l'arête ne sera pas rendue et deuxièmement, la face réfléchie ne rendra pas compte correctement de la réalité. Le problème concernant la non-génération de l'arête est négligeable puisque les détails liés à l'arête sont inférieurs à la face, cette dernière étant notre

critère de précision pour la génération de l'image. Le second problème se résout simplement en imposant une limite à l'angle entre les deux directions de propagation de sommets réfléchis. En effet, si le produit scalaire des deux vecteurs directeurs est inférieur à zéro, nous supprimons la face reliant les sommets incriminés.

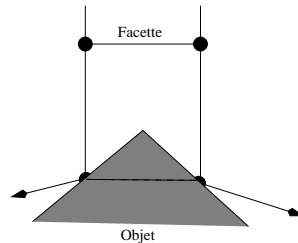


FIG. 89 – Un cas limite

4.4.6 Le calcul des composantes chromatiques d'un pixel

La couleur d'un pixel est la représentation du flux lumineux inclu dans l'angle solide contenu dans le pixel. Classiquement, cet angle solide est représenté par un rayon placé sur son axe central (fig 90.A). D'un autre point de vue, les rayons peuvent être situés à la frontière de l'angle solide. La couleur sera alors une moyenne de la couleur des rayons (fig. 90.B).

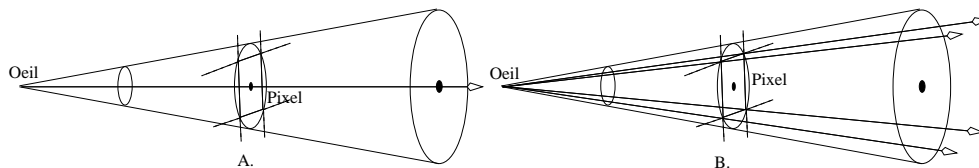


FIG. 90 – Rayons et angles solides

Dans notre algorithme, le calcul de la couleur d'un pixel est effective lorsque tous la couleur de chaque rayon a été déterminée.

Nous présentons au préalable le calcul de la couleur pour quelques travaux précédents.

Calcul des poids des rayons pour l'algorithme de Whitted. L'algorithme de Whitted effectue une subdivision hiérarchique des pixels, en les découpant en 4 à chaque subdivision (fig 91).

Chaque rayon suit l'axe médian de l'angle solide. L'aire représentée par chaque rayon peut être facilement calculée par le niveau de récursion auquel elle a été créée. Au niveau 1, les quatre rayons ont un poids de $\frac{1}{4}$ chacun. Les rayons créés au niveau 2 auront un poids de $\frac{1}{16}$. Le poids d'un rayon vaut donc :

$$\%(R) = \frac{1}{4^{N_R}} \quad (83)$$

où $\%(R)$ est le poids du rayon R et N_R son niveau de subdivision.

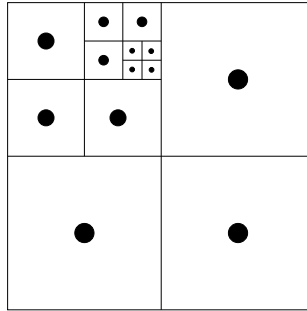


FIG. 91 – Subdivision hiérarchique d'un pixel

La couleur du pixel est donc égal à :

$$\text{Couleur du pixel} = \sum R_i \times \% (R_i) \quad (84)$$

Calcul de la couleur du pixel avec le Lancer de Rayons de Monte Carlo. La méthode qui est généralement employée pour évaluer la couleur finale d'un pixel lorsqu'on utilise un algorithme de lancer de rayons de Monte Carlo est la suivante : les rayons étant envoyés de manière aléatoire dans le pixel, les lois de la statistique montrent facilement que tous les rayons ont le même poids. La couleur finale du pixel est donc la moyenne des couleurs des rayons passant par celui-ci. Le poids de chaque rayon est donc de :

$$\% (R) = \frac{1}{\text{nombre de rayons}} \quad (85)$$

On utilise l'équation 84 pour déterminer la valeur des composantes chromatiques du pixel.

Notre calcul de couleur. Le calcul de la couleur du pixel passe par deux étapes : premièrement, la détermination de la couleur de chaque face qui l'échantillonne. La couleur d'une face est alors donnée par la moyenne des couleurs des trois rayons qui la composent. Deuxièmement, le poids de chaque face doit être déterminé. Le critère réflexe est la surface de la face. Ainsi, nous stockons sur chacune d'entre-elles, au cours de leur propagation, leur poids. Le pixel ayant une surface normalisée à 1, le poids des deux faces primaires est égal à $\frac{1}{2}$. Lors d'une subdivision de face, nous calculons le poids de chacune des quatre nouvelles faces en fonction de leur aire.

Ainsi le poids d'une face G dérivée de sa face mère F est égal à :

$$\% (G) = \% (F) \times \frac{S(G)}{S(F)} \quad (86)$$

où $\% (G)$ et $\% (F)$ sont les poids des faces G et F . L'aire de chaque face au moment de la subdivision est donnée par la fonction $S(\dots)$.

La couleur du pixel est alors calculée par :

$$\text{Couleur du Pixel} = \sum F_i \% (F_i) \quad (87)$$

$$\text{avec } F_i = \frac{S1_{F_i} + S2_{F_i} + S3_{F_i}}{3}$$

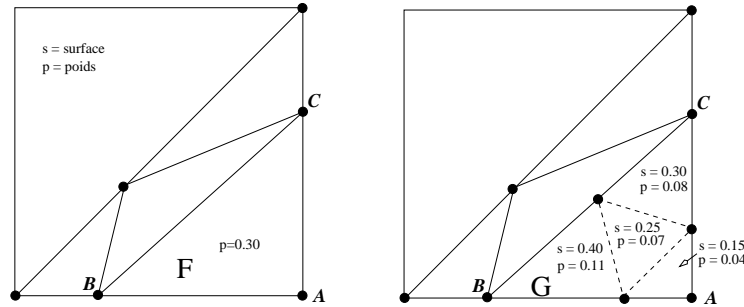


FIG. 92 – Calcul du poids d'une face

où F_i représente la couleur de la face i , $\% (F_i)$ le poids de cette face et enfin $S1, S2$ et $S3$ les trois sommets de la face F_i .

4.4.7 Gestion macroscopique des calculs

A ce stade de la présentation, un problème reste à soulever. Les quatre sommets définissant les deux faces primaires sont situés sur les coins du pixel. Un sommet primaire est donc commun à quatre pixels. L'ensemble des faces de l'image ne peut être envoyé dans la scène en parallèle pour des raisons évidentes d'explosion combinatoire et de coût mémoire⁴. Si les pixels sont calculés indépendamment les uns des autres, le calcul du trajet de chaque sommet primaire sera effectué quatre fois. Ceci non plus n'est pas envisageable.

Nous utilisons donc un système de rendu spécifique de l'image. En effet, nous traitons un paquet de pixels en parallèle en utilisant un cache correspondant à une ligne de l'image ou une partie de la ligne. Les sommets sont, alors, traités lignes par lignes. Les chemins des sommets primaires situés dans les deux coins inférieurs du pixel sont stockés dans le cache. Ils ne seront pas alors re-calculés lors de l'évaluation des pixels de la ligne d'en dessous.

Afin de limiter la taille mémoire du cache, celui-ci aura une taille d'au plus une ligne de l'image. Dans le cas où la taille du cache est inférieure à la taille de l'image (partie grise de l'image 93), l'image sera générée colonne par colonne. La largeur de la colonne aura la taille du cache. A l'intérieur de cette colonne, l'image sera calculée ligne après ligne. Seul le trajet des sommets situés sur la bordure du cache sera alors calculé deux fois.

4.4.8 Points forts de la méthode

Gestion de l'aliasing géométrique. L'aliasing géométrique se produit lorsque, au moment du rendu de la scène, tous les objets visibles ne sont pas détectés. Ceci concerne aussi les surfaces frontières des objets.

Notre méthode de stratification du pixel, basée sur la propagation des faces dans la scène, assure une détection de l'ensemble des objets de la pyramide de vue avec un même pas d'échantillonnage local. Elle

⁴Si on prend une image 1024×1024 et si on atteint 80 faces par pixel, il faudra stocker $8 \cdot 10^7$ faces.

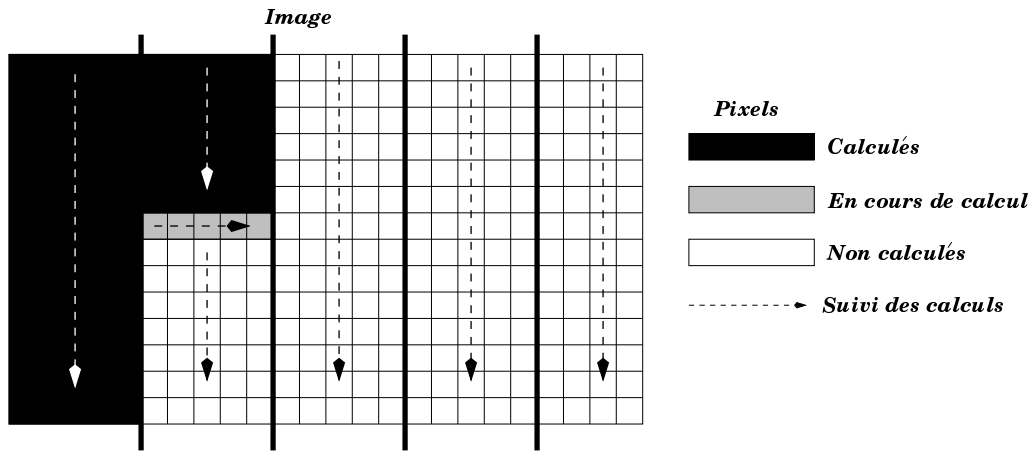


FIG. 93 – Génération d'une image

minimise donc l'aliasing géométrique.

Cet aliasing est d'autant plus minimisé qu'un sur-échantillonnage des faces est effectué lorsque celles-ci touchent les frontières des objets.

Gestion de l'aliasing de texture. Le lancer de faces est un algorithme permettant de limiter l'aliasing de texture. Pour cela, lorsqu'une face touche un objet texturé, sa couleur n'est alors pas la moyenne des couleurs des trois rayons qu'elle porte mais la moyenne de l'ensemble des pixels de la partie de texture touchée. Pour cela la méthode de Crow [Cro84], détaillée par la suite, est utilisée. L'image bitmap utilisée pour la texture de l'objet n'est stockée telle quelle. En chaque point (i, j) de la texture, on stocke la somme des couleurs des points de l'image compris entre un coin de l'image et ce point (i, j) . Ainsi la valeur V d'un point (i, j) de la texture se calcule par :

$$V(i, j) = \sum_{k=0}^i \sum_{l=0}^j C(k, l) \quad (88)$$

où $C(k, l)$ est la couleur du pixel (k, l) de la texture.

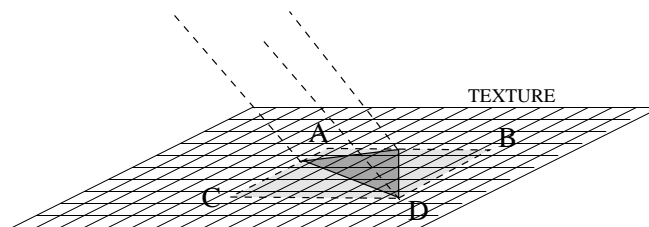


FIG. 94 – Gestion de l'aliasing de texture

Nous décidons d'approximer la couleur de la face par la couleur du rectangle englobant celle-ci ($ABCD$ fig. 94). La couleur du rectangle englobant est égal à :

$$\text{Couleur de la face} = \frac{V_D + V_A - V_B - V_C}{\text{Aire du rectangle}} \quad (89)$$

Rendu des objets semi-spéculaires, semi-transparents. Dans les images (fig. 95) tirées de [Arv95], nous présentons un objet semi-reflechissant et un objet semi-transparent. Lorsqu'un rayon de lumière in-

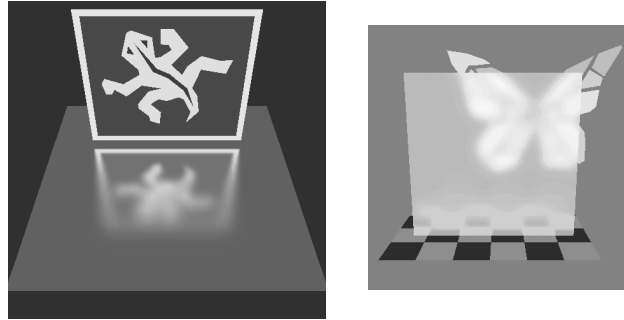


FIG. 95 – Exemple d'objets semi-spéculaire et semi-transparent (tiré de [Arv95])

tersecte un de ces objets et pour tenir compte de la non-régularité de la surface, il faut renvoyer un grand nombre de rayons autour de la direction principale de réflexion. Dans les algorithmes standard et pour limiter l'explosion combinatoire, le nombre de rayons renvoyé est souvent très limité.

Notre algorithme a l'avantage de pouvoir renvoyer depuis le point d'intersection une unique face dans la scène (fig. 96). Au cours de sa progression elle pourra être subdivisée en suivant les techniques exposées plus tôt.

Dans le cas où une face complète touche un objet semi-spéculaire ou semi-transparent, nous renvoyons un ensemble de faces connexes permettant de couvrir les différentes propagations des rayons réfléchis. Pour cela, une face simple est renvoyée par rayon incident et une reconnexion des trois faces est effectuée en utilisant un rayon positionné entre ces trois faces (voir exemple fig. 97).

4.5 Discussion

En l'état actuel, la méthode présentée ici est en cours de développement et d'implémentation. Seuls la gestion des faces, leurs subdivisions et le calcul de la couleur ont été effectués. On peut ainsi, pour le moment, tester l'algorithme sur les phénomènes d'aliasing géométrique.

Pour montrer la viabilité du projet, nous allons comparer le Lancer de Faces (LF) au Lancer de Rayons de Monte Carlo par le biais de l'algorithme de Schlick (LRMC) présenté §4.3. Pour cela, nous devons disposer, d'une part, d'une scène test contenant un grand nombre d'objets permettant de générer de l'aliasing. La scène est donc composée d'un ensemble de facettes quadrangulaires de couleur uniforme et de taille identique. Elles sont orientées de manière aléatoire et disposées à des distances différentes de l'observateur.

D'autre part, nous devons pouvoir comparer les deux méthodes en terme de nombre maximum de rayons envoyés par pixels. Ce nombre est un bon critère de qualité d'une image. Il faut donc trouver une

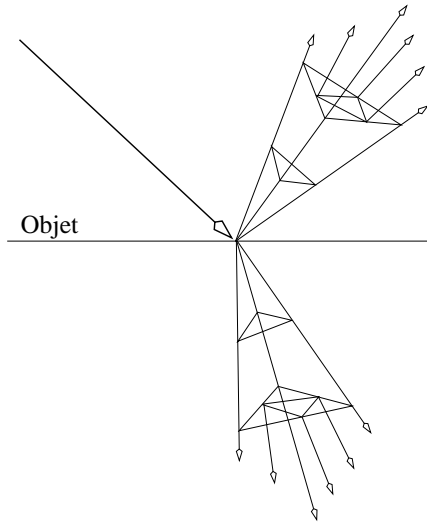


FIG. 96 – Rendu des objets semi-spéculaires

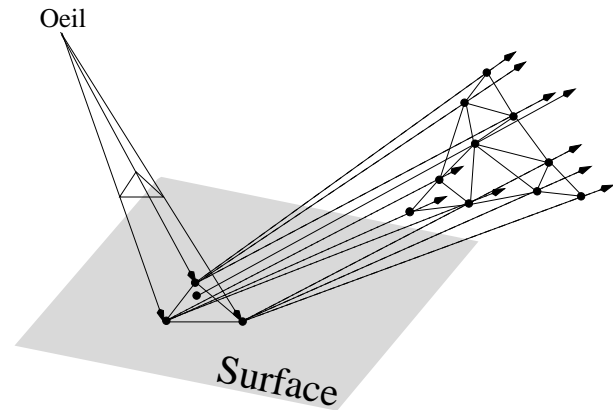


FIG. 97 – Une face réfléchie par un objet semi-spéculaire

relation entre le niveau de subdivision de l'ensemble des faces et le niveau de raffinement des sous-pixels du lancer de rayons de Monte Carlo. Ce calcul est présenté dans la partie suivante.

4.5.1 Relation entre niveau de subdivision (LF) et niveau de raffinement (LRMC)

En fixant un nombre de subdivision maximal pour notre méthode, nous désirons connaître le niveau de raffinement maximal pour le LRMC. Pour cela, Nous allons calculer dans un premier temps le nombre maximal de rayons envoyé pour un niveau de subdivision N_{LF} donné. A partir de ce nombre trouvé, il est alors possible de déduire facilement le nombre de raffinements maximum que l'on utilisera avec la méthode de Schlick.

Le calcul du nombre de rayons en fonction d'un niveau de subdivision du LF se résoud simplement en théorie des graphes :

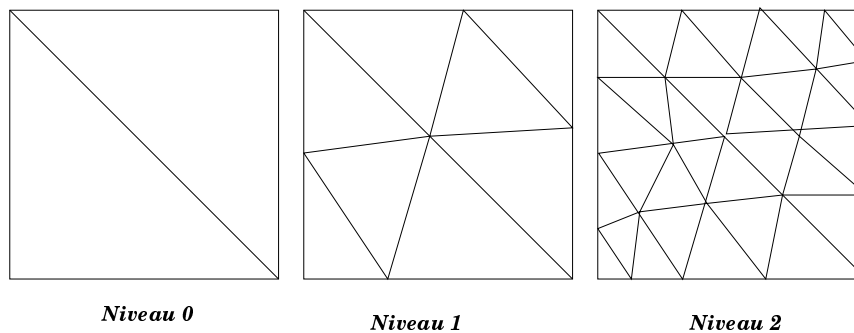


FIG. 98 – Le graphe à trois niveaux de subdivisions

Le graphe formé par les faces est planaire. Nous allons considérer les calculs pour un graphe où chaque

face a été subdivisée complètement. La figure 98 montre un graphe de niveau 2. Il existe une relation simple reliant le nombre de faces du graphe à son nombre d'arêtes. Soit F_N (resp. A_N et S_N) le nombre de faces (resp. le nombre d'arêtes et le nombre de sommets) au niveau N .

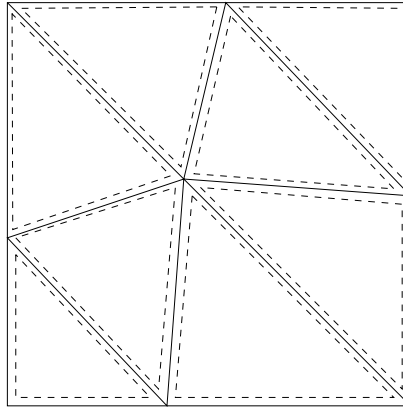


FIG. 99 – Une relation entre le nombre de faces et d'arêtes dans le graphe

De manière préliminaire, le nombre de faces pour un niveau de subdivision donné peut être calculé facilement car toutes les faces sont subdivisées en 4 :

$$F_N = 2 \times 4^N \quad (90)$$

Le nombre d'arêtes reliées à la face extérieure du graphe est égal à 4×2^N au niveau de subdivisions N . De plus, sachant qu'une face contient 3 arêtes, le calcul de $3F_N$ permet de comptabiliser l'ensemble des arêtes du graphe deux fois, y compris celles reliées à la face extérieure au graphe.

La relation entre le nombre de faces et le nombre d'arêtes est donc :

$$3F_N = 2A_N - 4 \times 2^N \quad (91)$$

On peut donc en déduire A_N :

$$A_N = 2^{N+1} + 3 \times 4^N \quad (92)$$

Le calcul du nombre de sommets du graphe de niveau N (ce qui revient à dire le nombre de rayons à envoyer avec le LRMC), est basé sur la remarque suivante : lorsque l'on passe d'un niveau de subdivision au suivant, un nouveau sommet (ou rayon) est rajouté sur chaque arête du graphe. Soit la relation :

$$S_N = S_{N-1} + A_{N-1} \quad (93)$$

La valeur directe de S_N se déduit par :

$$\left\{ \begin{array}{l} S_N - S_{N-1} = A_{N-1} \\ \vdots \\ S_1 - S_0 = A_0 \end{array} \right. \quad (94)$$

Avec $S_0 = 4$ et $A_0 = 5$.

On obtient :

$$S_N = \sum_{i=0}^{i=N-1} A_i = 2 \sum_{i=0}^{i=N-1} 2^i + 3 \sum_{i=0}^{i=N-1} 4^i \quad (95)$$

Finalement, le nombre de sommets dans le graphe, donc le nombre de rayons, pour une subdivision de niveau N est égal à :

$$S_N = 4^N + 2^{N+1} + 1 \quad (96)$$

Pour calculer maintenant le niveau maximal de raffinement en fonction de S_N , nous faisons l'hypothèse que dans chaque sous-pixel d'un pixel, la probabilité d'atteindre le niveau maximal de raffinement est identique. Aussi, le pixel étant divisé en quatre sous-pixel, le niveau de raffinement maximal N_{LRMC} pour une niveau de subdivision N_{LF} sera de :

$$N_{LRMC} = \frac{S_N}{4} \quad (97)$$

A titre d'exemple, si on choisit un niveau maximal de subdivision des faces égal à $N_{LF} = 3$. Le lancer de faces pourra ainsi envoyer jusqu'à $S_3 = 81$ rayons par pixel. Nous limiterons donc le niveau de raffinement du Lancer de Rayons de Monte Carlo à $N_{LRMC} = 20$ raffinements.

4.5.2 Résultats du comparatif

Nous avons comparé les deux méthodes sur le même type de scène. Seul le nombre d'objet varie. Dans la figure 100.A, nous avons disposé 200 objets aléatoirement dans la scène alors que la figure 100.B seulement 100 objets ont été créés.

Nous utilisons le même noyau logiciel pour les deux méthodes de rendu (LF et LRMC), i.e. le même calcul d'intersection, la même déclaration de scène, les mêmes méthodes d'accélération. Les seules différences logicielles concernent donc la gestion des faces et le calcul de la couleur finale du pixel.

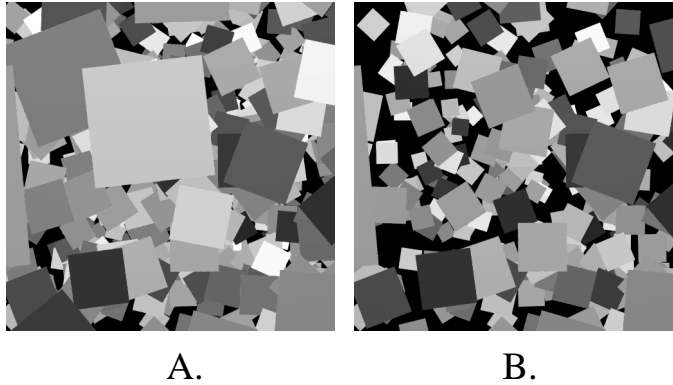


FIG. 100 – Images résultats produites avec le lancer de faces

Nous limitons le niveau de subdivision du lancer de faces à 3. Ceci implique (vu précédemment) une limite de 20 raffinements pour la méthode de Monte Carlo.

Les tests ont été réalisés sur un PC Pentium II 300 Mhz. La taille de chaque image générée est de 512×512 pixels.

Pour une qualité d'image identique, nous obtenons les résultats suivants :

Image	LRMC			LF		
	Rayons par par image	Rayons par par pixel	Temps de calcul	Rayons par par image	Rayons par par pixel	Temps de calcul
100.A	1467644	5.59	26s	1399358	5.34	30s
100.B	1491066	5.68	23s	1752262	6.68	27s

Nous pouvons remarquer, aux vues des résultats que pour une image de même qualité visuelle, les temps de calculs ainsi que le nombre moyen de rayons par pixel pour les deux méthodes sont quasiment similaires. Ils sont tout de même en défaveur de notre méthode. La perte de temps engendrée est due essentiellement à la gestion des facettes et les nombreuses subdivisions effectuées sur les surface frontières aux objets.

Pour l'image 100.B, le nombre de rayons moyen par pixel est supérieur au LRMC ce qui entraîne une perte de temps inévitable (fond noir). Ceci s'explique par le plus grand nombre de "trous" dans la scène. Les faces qui passent par ces trous se subdivisent tout au long de leur progression pour intersecter d'éventuels petits objets. Ayant une connaissance préalable de la scène, tout le travail de subdivision dans les "trous" peut sembler inutile mais il est indispensable lorsqu'une telle connaissance n'est pas possible.

Notre algorithme est configurable en fonction des connaissances qu'a l'utilisateur de sa scène. Il peut soit limiter le nombre maximal de subdivisions, soit configurer la distance maximale de parcours des faces dans la gestion. Une fois cette distance dépassée, la gestion des faces est abandonnée et les rayons progresse alors dans la scène par l'algorithme classique de Lancer de rayons.

4.6 Application à la simulation des mirages

En la présence de brusques changements de température, l'indice de réfraction de l'air est modifiée. Les variation d'indice de réfraction sont alors continus et bruités en fonction des différentes couches de températures. Les rayons lumineux qui traversent une telle zone sont alors déviés par les lois de la réfraction. Ce phénomène optique est nommé *mirage*. L'ouvrage de R. Greenler référence l'ensemble des phénomènes lumineux dûs à la réfraction de la lumière [Gre80].

Notre algorithme de Lancer de Vagues est tout à fait adapté à la génération des mirages. En effet, à la sortie de la couche de chaleur, deux rayons voisins peuvent prendre des directions de propagation très différentes. ils peuvent ainsi créer un aliasing géométrique important en passant côté d'une partie de la scène. Nous appliquons donc ici le lancer de vagues à la génération des mirages.

L'oeil humain a l'habitude que la lumière se propage en ligne droite. Dans le cas des mirages, une illusion d'optique est créée et l'observateur a l'impression que des miroirs ont été insérés dans la scène.

Il existe trois types de mirages :

Les *mirages* dit "*inférieurs*" (fig 101) se forment lorsque des couches de températures élevées se trouvent à la surface du sol. L'indice de l'air diminue et les rayons lumineux se rapprochant du sol sont



FIG. 101 – Un mirage inférieur

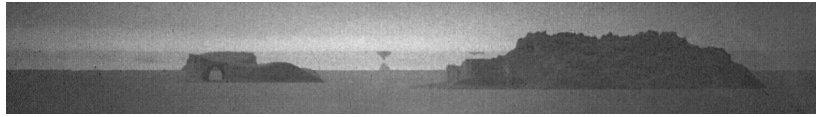


FIG. 102 – Un mirage supérieur

alors déviés vers le ciel créant une illusion de miroir au sol. Les *mirages "supérieurs"* (fig 102) se produisent lorsque une couche d'air chaud est située en altitude du sol. Les rayons dirigés vers le ciel sont alors "retournés" vers le sol. Ceci donne l'impression que le paysage est inversé dans le ciel. Les "*Fata Morgana*" sont une combinaison multiple des deux types de mirages précédents.

4.6.1 Précédents travaux

Les travaux concernant les mirages sont peu nombreux. A notre connaissance, seule deux techniques ont été mises au point :

- Dans [BTL90],[BLT90], une méthode qui utilise lancer de rayons classique, permet la création de mirages. Dans les scènes, un objet virtuel formé d'un ensemble de strates parallèles et transparentes, mais d'indice de réfraction différents, modélise le champ de température. Ce champ de température est aussi nommé *milieu réfringent*. Les rayons progressant dans la scène sont alors déviés à l'intérieur de cet objet (fig. 104).
- Pour K. Musgrave [Mus90], la réflexion interne totale du rayon lumineux est le phénomène fondamental pour la simulation des mirages. Donc pour générer un mirage, seule cette réflexion interne a besoin d'être prise en compte. L'image générée ne représentera pas le phénomène exact, mais sera tout aussi réaliste pour la synthèse d'images.

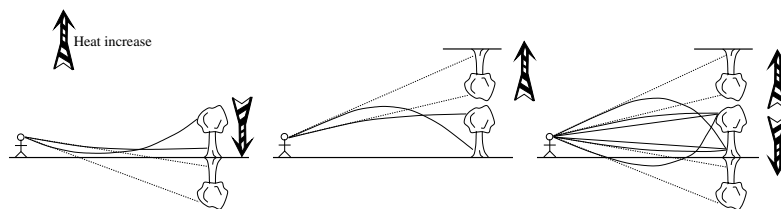


FIG. 103 – a. Mirage inférieur, b. Mirage supérieur, c. Fata Morgana

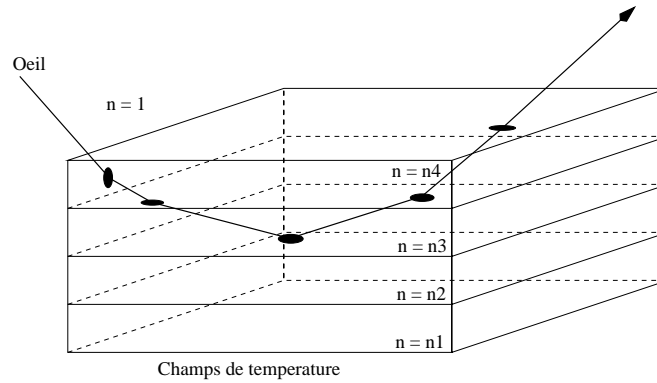


FIG. 104 – Parcours de la lumière dans le modèle de Berger

4.6.2 Définition des milieux réfringents

Premiers modèles. Deux types de modèle pour définir un objet dont l'indice de réfraction varie continuellement peuvent être utilisés. Le premier consiste à reprendre celui de Berger et al. . Le milieu est alors représenté par un objet constitué de couches d'indice constant (fig 105.a). Ce modèle a le réel avantage de ne nécessiter que très peu de temps de calculs pour connaître l'indice de l'objet en un point donné (car les couches restent à peu près parallèles). De plus, la réfraction, dépendant de la normale à la surface isothermique devient très facile à calculer. Par contre, le modèle par strate ne permet pas de gérer facilement des variations d'indice arbitraires.

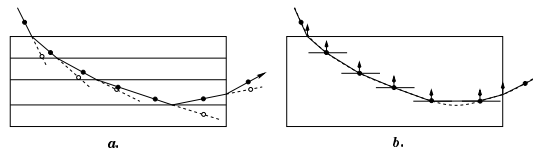


FIG. 105 – Deux méthodes pour définir un milieu d'indice continu

Le deuxième utilise des fonctions paramétriques pour décrire le champs de température. C'est ce choix que nous avons fait car il est plus réaliste. Nous disposons d'une fonction $N(x, y, z, t)$ donnant l'indice de réfraction du milieu réfringent. La difficulté majeure est le calcul de la normale à la surface isotherme en tout point du milieu. Pour cela, une version 3D de l'algorithme vu §2.2.3.3 pour le calcul des courbes de niveau du sol pour la réfraction des vagues, peut être appliquée. Le principe en est le suivant : en tout point M du milieu, nous tirons au hasard quatre points (A, B, C, D) autour de M . Nous pouvons ainsi déterminer les trois points (E, F, G) sur les arête du tétraèdre ayant la même température que M . La normale N au plan local passant pas ces trois points est une bonne approximation de la normale à la surface isotherme passant par M .

Calcul de l'indice de réfraction en fonction de la température de l'air. L'indice de réfraction N de l'air varie avec la pression P exprimée en millibars et la température T exprimée en Kelvins, suivant

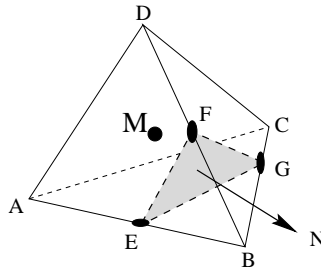


FIG. 106 – Calcul de la normale à la surface isotherme

l'approximation [EKG77][EFV82] :

$$N - 1 = 8 \times 10^{-5} \times \frac{P}{T} \quad (98)$$

Notre définition du champ de température En l'absence de tout autre phénomène, sur un sol plat et uniforme, la température varie seulement en fonction de l'altitude en formant des couches uniformes parallèles au sol. Afin d'obtenir différents types de mirages, il suffit de modifier le gradient de température en fonction du sol.

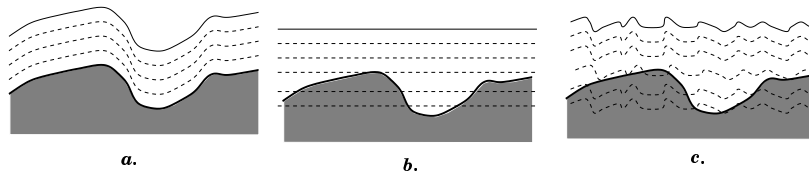


FIG. 107 – Différents champs de températures

Dans la réalité, le sol n'est jamais parfaitement plat. La chaleur s'accumule dans les creux. Au lieu d'utiliser une fonction de température dépendant de la distance au sol (fig. 107.a), nous prenons une fonction dépendant de l'altitude (fig. 107.b). Ainsi, la température sera plus forte dans les creux que dans les bosses.

Améliorations. L'air, dans une scène réelle, n'est pas composée de couches aussi planes. Afin d'avoir un rendu plus proche de la "réalité", nous allons modifier localement les couches de température.

Pour cela, deux méthodes sont à notre disposition :

- Pour des raisons de rapidité de calcul de la normale à la couche isotherme, nous gardons volontairement les couches de température planes. Par contre, à chaque étape où le vecteur réfracté est calculé, nous appliquons une fonction de perturbation de la normale ou *Bump mapping* [BM93] (fig 107.c).
- La méthode précédente ne peut pas modéliser toutes les formes de couches d'air comme, par exemple, les poches de températures. La technique que nous utilisons est donc de rajouter un bruit fractal 3D sur les couches de températures. Le milieu réfringent est donc plus réaliste mais demande d'utiliser la méthode vue précédemment pour calculer la normale à la surface isotherme - cette normale servant au calcul de la réfraction.

4.6.3 Le milieu réfringent et le Lancer de Faces

Dans l'algorithme initial du Lancer de Faces, le pas de progression des rayons est déterminé de manière automatique. En effet, la lumière est supposée se propager en ligne droite dans un milieu homogène et isotrope. En ce qui concerne les mirages, ce n'est pas le cas. L'utilisateur va donc définir un pas de progression dans le milieu réfringent. Celui-ci sera adaptatif et dépendra de l'indice de réfraction de l'air. En effet, plus l'indice de réfraction est faible et inférieur à 1, plus la déviation est importante et plus le pas de progression doit être fin.

L'utilisateur définira en fait deux pas de progression des faces dans le milieu réfringent : un pas maximum (Pas_{max}) et un pas minimum (Pas_{min}). Le pas courant dans le milieu sera donné par interpolation entre ces deux pas prédéfinis par l'utilisateur.

4.6.4 Résultats



FIG. 108 – Un exemple de Fata Morgana

Nous présentons, en guise de résultat, deux images. La première (fig. 108) est la représentation d'un Fata Morgana (mélange de mirage supérieur et inférieur). Le texte est placé exactement au centre de l'image. Nous ne voyons ici que les différents mirages mélangés.

La seconde image 109 représente un mirage devant la tour Eiffel. Nous avons utilisé l'algorithme de Bump mapping pour perturber localement la couche de température. Une photo de la tour Eiffel a été plaquée sur un facette verticale en fond de scène. Le temps de génération de l'image (800×800) 109 est de l'ordre de 5 minutes sur un Pentium II 300 mhz. Nous avons limité le nombre de subdivisions de chaque face à 3.

4.7 Conclusion

Nous avons présenté dans ce chapitre un algorithme de rendu basé sur des méthodes de Monte Carlo appelé Lancer de Faces. Il permet, entre autres, un échantillonnage identique en tout point de la scène et ainsi il limite les problèmes d'"aliasing" géométriques. La pyramide de vue est découpée en faces triangulaires qui sont envoyées dans la scène. Dès qu'une face devient trop importante, elle est subdivisée. Les rayons portés par les sommets des faces ne sont pas lancés a priori et systématiquement à partir de l'observateur mais aux points où cela est nécessaire. Ceci fait l'originalité de notre méthode par rapport aux lancer de rayons classiques.

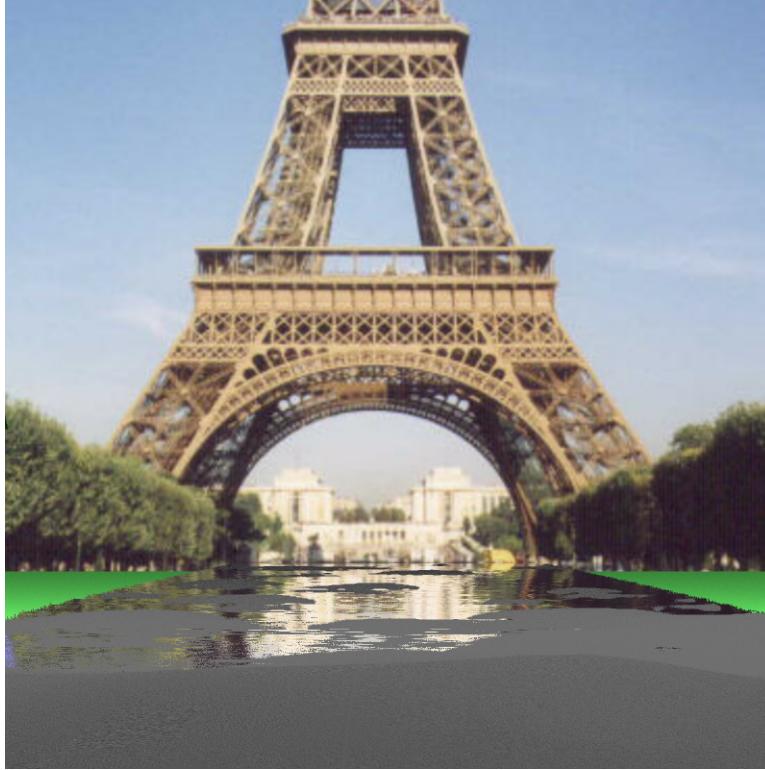


FIG. 109 – Mirage sous la tour Eiffel

Ce travail n'est pas complètement achevé et ni implémenté dans son intégralité mais une première comparaison avec un lancer de rayons de Monte Carlo adaptatif permet de montrer que les résultats sont similaires (autant en terme de temps de calcul que de précision de l'image générée).

Enfin, cet algorithme a été appliqué à la génération des mirages où un exemple a été présenté.

Conclusion

La modélisation des océans, qui est une infime partie des phénomènes naturels, est réalisée en deux grandes étapes : premièrement, la modélisation géométrique est nécessaire pour définir l'interface air-eau. Cette modélisation utilise des données physiques d'entrées qui correspondent autant aux caractéristiques internes de l'océan dans la scène (amplitude et longueur d'onde des vagues, direction et vitesse de propagation, ...) qu'aux facteurs extérieurs (bathymétrie des fonds marins, vitesse du vent, ...). Deuxièmement, un éclairage ou "illumination" de la surface de l'océan permet de générer des images dites réalistes.

Pour modéliser un océan, il faut regarder le phénomène sous deux points de vue : une vue de coupe permettant une définition correcte de la forme des vagues et une vue de dessus pour modéliser la propagation des ondes dans la scène. Pour chacun des aspects, nous avons effectué un choix de modèle tiré de la physique que nous avons adapté aux exigences de l'infographiste.

Pour définir la forme des vagues - chapitre 1 -, nous avons construit un modèle complet tiré des théories physiques de *Gerstner* et *Biesel* et des travaux en infographie de *Fournier* et *Reeves*. Cette méthode, comparée aux précédentes approches effectuées dans le domaine, permet une modélisation plus intuitive de la forme des vagues. Entre autres, il assure une arrivée correcte de la houle sur le rivage. Il peut être utilisé dans les baines. Finalement, trois paramètres empiriques sont utilisés pour définir le déferlement des vagues. Ceci s'effectue par l'ajout d'une fonction simulant l'accélération des particules sur les crêtes des vagues et d'une autre fonction simulant la gravité.

Le déplacement et la propagation des trains de vagues - chapitre 2 - sont principalement influencés par la bathymétrie des fonds marins. Les trois phénomènes mis en jeu sont la *réfraction*, la *diffraction* et la *réflexion* des trains de vagues. La variation de vitesse de l'onde en fonction de la profondeur d'eau génère la réfraction. La réflexion apparaît lorsque la houle heurte un obstacle et la diffraction est générée derrière ces obstacles.

Après avoir effectué une étude des différentes théories issues de la physique, nous avons proposé un modèle complet utilisant les orthogonales aux fronts de vagues pour générer le déplacement de l'onde dans la scène. Il est de plus adaptatif en échantillonnant de manière équivalente la totalité de la surface. Le principe est simple : les orthogonales sont envoyées en parallèle dans la scène et dès que la distance entre deux d'entre elles voisines est supérieure à un certain seuil, nous en générons une nouvelle au milieu. Ce modèle complet est appelé *Lancer de Vagues adaptatif*.

La géométrie de la surface se déduit alors de la combinaison du lancer de vagues dynamique et du modèle générant la forme des vagues. Ayant généré des vagues déferlantes, nous ne pouvons pas stocker la surface composée de plusieurs trains de vagues de manière classique dans un champ scalaire. La géométrie est composée d'un ensemble de facettes indépendantes. Chaque ensemble définit un train de vagues. Nous avons donc développé une méthode - chapitre 3 - basée sur le *displacement mapping* pour reconstruire et identifier, durant le processus de rendu, la surface réelle de l'océan.

Une fois la géométrie reconstruite, l'étape d'illumination est alors nécessaire - chapitre 3. Elle a, de plus, une part importante dans la qualité des images obtenues. L'illumination globale est basée sur un lancer de rayons incrémental classique. Par contre, nous avons développé un modèle hybride d'illumination locale. En effet, il utilise le modèle de couleur RGB pour le rendu de tous les objets situés hors de l'eau et le modèle spectral pour le rendu des fonds marins. Ceci permet de tenir compte de l'absorption réelle de la lumière par l'eau. Nous utilisons pour cela un spectre complet d'absorption de l'eau de mer. De plus, les coefficients de Fresnel sont approximés pour déterminer les parts d'énergies provenant des fonds marins et des réflexions.

Dans cette première partie de mon travail de thèse concernant la modélisation des vagues, je pense que l'ensemble des "modèles déterministes" a été étudié et des apports certains ont été présentés. La notion de "modèle déterministe" englobe l'ensemble des phénomènes, agissant sur les vagues et leur propagation, où toutes les particules d'eau peuvent être placées de façon déterminée dans la scène. Ces apports ouvrent la voie à plusieurs perspectives : la modélisation des phénomènes "non déterministes".

J'évoquerai, par exemple, la modélisation complète du déferlement par des systèmes particuliers. En effet, le modèle de forme de vague présenté dans cette thèse ne permet pas une génération complète du déferlement - phénomène non encore entièrement résolu par les physiciens. Un système particulier pourrait tout à fait modéliser le chaos de gouttes d'eau à la surface d'une vague déferlante en ce qui concerne les vues de près. Pour une vue de loin d'une vague déferlante, une texture 3D adaptée et plaquée à la surface est envisageable. Ces deux types de méthodes (système particulière et texture) pourraient être adaptés à toutes les générations d'écumes et d'embruns.

Le modèle géométrique ne simule pas la propagation de la vague sur la plage et l'infiltration de l'eau dans le sable lors du reflux. La modélisation physique de ce phénomène me paraît assez complexe à résoudre. Un certain nombre de thèses en cours traitent de ce seul sujet. Par contre, une modélisation empirique semble plus envisageable où les phénomènes de réfraction des vagues seraient abordés à une échelle plus fine.

De plus, je suis tout à fait conscient qu'utiliser le modèle d'illumination de *Blinn* en y ajoutant spectre et coefficients de Fresnel peut paraître absurde. Pour des raisons de temps, nous n'avons pas pu implémenter des modèles plus récents où l'erreur commise par rapport à une certaine réalité serait moins grande. Ceci ne changera nullement les principes posés chapitre 3.

Finalement, la turbidité de l'eau en bord de plage est à prendre en compte et à simuler. En effet, sur le rivage, le fluide se compose d'un mélange d'eau, d'air, de sel et surtout de particules de sable et de poussières. L'application pourrait s'effectuer en illuminant un delta-cube d'eau et en utilisant ce cube comme texture de la couleur de l'eau en bord de rivage. Cette idée pourrait aussi s'appliquer au rendu de la mousse sur la plage provenant de l'écume. L'application pourrait effectuer le rendu d'un delta-cube de mousse et l'utiliser comme texture fractale sur la scène.

Des applications de notre approche concernant la modélisation de scènes océanes sont de plus envisagées. On pourrait modéliser l'écoulement de l'eau dans le bassin d'Arcachon ou dans la baie du Mont St Michel. J'ai pris des contacts avec la société "Areté Image Software" pour une possible collaboration. Cette collaboration porterait plus sur la partie géométrie des océans et entre autres le déferlement. Enfin, une collaboration avec les océanologues de l'université de Bordeaux est à envisager (URA CNRS 197, Département de Géologie et Océanographie).

Une deuxième partie de ce travail de thèse a concerné - chapitre 4 - la présentation d'un algorithme de rendu basé sur le lancer de rayons. Cet algorithme, nommé *Lancer de Faces*, se range dans la catégorie des méthodes de Monte Carlo. Son originalité est d'utiliser une stratification du pixel originale en découpant la pyramide de vue en faces triangulaires. Le raffinement de l'évaluation de la couleur du pixel s'effectue en fonction de la géométrie de la scène. Le principe est de garder une notion de voisinage entre les rayons lancés dans la scène et ainsi limiter l'aliasing géométrique par un sur-échantillonnage des faces si la surface de celles-ci devient trop importante. Une application à la simulation des mirages à été présentée.

Le lancer de faces est un projet en cours de développement où des travaux restent à valider. Il reste à finaliser le rendu des surfaces semi-spéculaires et la gestion de l'aliasing de textures. Plusieurs applications pourraient être envisagées ; notamment la simulation de l'endoscopie virtuelle et le rendu des pierres précieuses.

Tout au long de ce travail de thèse, j'ai essayé de suivre un même fil conducteur. La première étape concernait une étude bibliographique des phénomènes physiques. Dans un deuxième temps, j'ai effectué la même recherche mais dans le domaine de l'infographie afin de connaître les principales techniques d'optimisation des équations physiques pour suivre les contraintes de l'infographiste. Une troisième étape concerne la modélisation d'un premier algorithme sur papier puis son implémentation. Tout implémentation fait surgir des problèmes non encore vus sur papier. Ainsi, un va-et vient entre modélisation et implémentation a été effectué pour trouver des techniques inédites et rapides. Le résultat obtenu est un modèleur complet de vagues, un moteur de rendu dédié à l'illumination des scènes océanes. Enfin, une nouvelle technique de lancer de rayons a été implémentée.

Bibliographie

- [AA85] C. Aristaghes and P. Aristaghes. *Théories de la houle, houle réelle, propagation de la houle*. Service technique central des ports maritimes et voies navigables, 1985.
- [Ama84] J. Amanatides. Ray tracing with cones. In *Siggraph'84*, pages 129–135, 1984.
- [App68] A. Appel. Some techniques for shading machine renderings of solids. In *SJCC*, 37–45 1968.
- [Arv95] J. Arvo. Applications of irradiance tensors to the simulation of non-lambertian phenomena. In *Siggraph'95*, 1995.
- [AW87] J. Amantides and A. Woo. A fast voxel traversal algorithm for ray tracing. In *Proceedings of Computer Graphics International*, pages 149–155, 1987.
- [Bie52] F. Biesel. Study of wave propagation in water of gradually varying depth. In *Gravity Waves*, pages 243–253. U.S. National Bureau of Standards Circular 521, 1952.
- [Bli77] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *Siggraph'77*, pages 192–198, 1977.
- [Bli78] J. F. Blinn. Simulation of wrinkled surface. In *Siggraph'78*, volume 12(3), pages 286–292, 1978.
- [BLT90] Marc Berger, Nancy Levit, and Terry Trout. Rendering mirages and other atmospheric phenomena. In *Eurographics'90*, pages 459–68, September 1990.
- [BM93] B. G. Becker and N. L. Max. Smooth transitions between bump rendering algorithms. In *SIGGRAPH'93*, pages 183–190, 1993.
- [Bon80] R. Bonnefille. *Cours d'hydraulique maritime*. Masson, 1980.
- [BTL90] Marc Berger, Terry Trout, and Nancy Levit. Ray tracing mirages. In *IEEE Computer Graphics and Applications*, pages 36–41, May 1990.
- [CL95] J. Chen and N. Lobo. Toward interactive-rate simulation of fluids with moving obstacles using navier-stokes equations. In *Graphical Models and Image Processing*, pages 107–116, March 1995.
- [Com90] R. Comolet. *Mécanique expérimentale des fluides*. Masson, 1990.
- [CR84] Cook and L. Robert. Shade trees. In *Siggraph'84*, volume 18(3), pages 223–231, 1984.
- [Cro84] F. C. Crow. Summed-area tables for textures mapping. In *CGA*, volume 1, pages 12–24, 1984.

- [CT81] R. Cook and K. Torrance. A reflectance model for computer graphics. In *Siggraph'81*, pages 307–316, 1981.
- [EFV82] L. Lazzeri E. Fabri, G. Fiorio and P. Violino. Mirage in the laboratory. In *American J. Physics*, volume 50(6), pages 517–528, 1982.
- [EKG77] K. Thyagarajan E. Khular and A. Ghatak. A note on mirage formation. In *American J. Physics*, volume 45(1), pages 90–92, 1977.
- [Esc92] C. Escazut. Les algorithmes génétiques. Technical report, Université de Nice-Sophia Antipolis, 1992.
- [Fle91] C. A. J. Fletcher. *Computational techniques for fluid dynamics*, volume 2. Springer series in computational techniques, 1991.
- [FM96] N. Foster and D. Metaxas. Realistic animation of liquids. In *Graphics Interface '96*, 1996.
- [FM97] N. Foster and D. Metaxas. Controlling fluid animation. *CGI'97*, pages 178–188, 1997.
- [FR86] A. Fournier and W. T. Reeves. A simple model of ocean waves. In *SIGGRAPH'86*, volume 20, pages 75–84, 1986.
- [Ger09] F. J. Gerstner. Theorie der wellen. (*reprint*) *Ann des Physik*, 32 :412–440, (1802) 1809.
- [Gla88] A. S. Glassner. How to derive a spectrum form an rgb triplet. *Computer Graphics and Applications*, 8(3) :60–70, 1988.
- [Gou71] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, 20(6) :623–628, 1971.
- [Gre80] R. Greenler. *Rainbows, Halos, and Glories*. Cambridge University Press, 1980.
- [GS97] J. C. Gonzato and B. Le Saëc. A phenomenological model of coastal scenes based on physical considerations. In Springer-Verlag, editor, *8th Eurographics Workshop on Computer Animation and Simulation*, pages 137–148, 1997.
- [GS98] J. C. Gonzato and B. Le Saëc. Lancer dynamique de "facette" d'illumination - application à la simulation des mirages. In *AFIG'98*, pages 153–162, 1998.
- [GS99a] J. C. Gonzato and B. Le Saëc. Face tracing. Technical Report 1220-99, LaBRI, 1999.
- [GS99b] J. C. Gonzato and B. Le Saëc. On modeling and rendering ocean scenes (diffraction, surface tracking and illumination). In *WSCG'99*, pages 93–101, 1999.
- [GS00] J. C. Gonzato and B. Le Saëc. On modeling and rendering ocean scenes. *Journal of Visualisation and Computer Simulation - A paraître*, début 2000.
- [Hal89] R. Hall. *Illumination and Color for Computer Generate Imagery*. Springer-Verlag, 1989.
- [HH64] J. M. Hammerley and D. C. Handscomb. *Monte Carlo methods*. Wiley, 1964.
- [HH84] P. S. Heckbert and P. Hanrahan. Beam tracing polygonal objects. In *Siggraph'84*, pages 119–127, 1984.
- [Iva75] A. Ivanoff. *Introduction à l'Océanographie*, volume II. Vuibert, Paris, 1975.

- [IZ95] A. Imamiya and D. Zhang. Modelling breaking ocean waves, influence of floor and refraction. In *Pacific Graphics 95*, 1995.
- [Kin65] B. Kinsman. *Wind Waves, their generation and propagation on the ocean surface*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1965.
- [Lac65] H. Lacombes. *Cours d'Océanographie Physique*. Gauthier-Villars, Paris, 1965.
- [Ma87] G. A. Mastin and al. Fourier synthesis of ocean scenes. *IEEE Computer Graphics and Applications*, pages 16–23, March 1987.
- [MAZ98] S. Michelin, D. Arquès, E. Felgines and K. Zampieri. Convection thermique en régime turbulent, synthèse d'images et animation de la chaleur. In *Proceedings AFIG'98*, pages 143–152, 1998.
- [Mit87] D. P. Mitchell. Generating antialiased images at low sampling densities. In *Siggraph'87*, pages 65–72, 1987.
- [MSN87] T. Takahashi M. Shinya and S. Naito. Principles and applications of pencil tracing. In *Siggraph'87*, pages 45–54, 1987.
- [Mus90] F. Kenton Musgrave. A note on ray tracing mirages. In *IEEE Computer Graphics and Applications*, pages 10–12, November 1990.
- [Pea86] D. R. Peachey. Modeling waves and surf. In *SIGGRAPH'86*, volume 20, pages 65–74, 1986.
- [Pho75] B. T. Phong. Illumination for computer generated pictures. *CACM*, 18 :311–317, 1975.
- [PLS98] Les vagues. *Dossier Pour la Science*, Octobre 1998.
- [PS89] J. Painter and K. Sloan. Antialiased ray tracing by adaptative progressive refinement. In *Siggraph'89*, pages 281–288, 1989.
- [Ran63] W. J. M. Rankine. On the exact form of waves near the surface of deep water. *Phil. Trans. Roy. Soc. A*, 153 :127–138, 1863.
- [RLCC84] T. Porter R. L. Cook and L. Carpenter. Distributed ray tracing. In *Siggraph'84*, pages 165–174, 1984.
- [Sèv96] Robert Sève. *Physique de la couleur - de l'apparence colorée à la technique colorimétrique*. MASSON, 1996.
- [Sam84] H. Samet. The quadtree and related hierarchical data structures. In *Computing Surveys*, volume 16(2), pages 187–260, 1984.
- [Sch92] C. Schlick. Divers éléments pour une synthèse d'images réalistes. Master's thesis, Université Bordeaux I, 1992.
- [Sch94] C. Schlick. An inexpensive brdf model for physically-based rendering. In *Eurographics'94*, volume 13, pages 233–246, 1994.
- [Sha49] C. E. Shannon. Communications in the presence of noise. In *IRE*, volume 37, pages 10–21, 1949.
- [Shi90] P. Shirley. *Physically based lighting calculations for computer graphics*. PhD thesis, University of Illinois at Urbana Champaign, 1990.

- [TB87] P. Y. Ts'o and B. A. Barsky. Modeling and rendering waves : Wave-tracing using beta-splines and reflective and refractive texture mapping. In *ACM Transactions on Graphics*, volume 6, pages 191–214, July 1987.
- [Whi80] J. T. Whitted. An improved illumination model for shaded display. In *CACM*, pages 343–349, 1980.
- [YYS88] N. Inamoto Y. Yuan, T. L. Kunii and L. Sun. Gemstone fire : Adaptative dispersive ray tracing of polyhedrons. *The Visual Computer*, pages 259–270, 1988.

Résumé

Modélisation des Scènes Océaniques

De nos jours, l'informatique graphique joue un rôle de plus en plus important. Elle permet, entre autres, d'essayer de comprendre le monde qui nous entoure en y reproduisant les principaux phénomènes naturels (feu, nuages, optiques des métaux, ...). L'objet de cette thèse est la modélisation géométrique et optique des scènes océaniques. Plus précisément, nous voulons déterminer la surface des océans (hauteur, longueur d'onde et direction des vagues) en fonction de différentes contraintes imposées par une scène réelle (bathymétrie, digues, ports, ...). Pour imiter au mieux la réalité, nos travaux sont basés sur différentes théories physiques.

Nous proposons tout d'abord un algorithme pour la modélisation géométrique de la forme des vagues, déferlantes ou non, tant au large qu'à l'approche des côtes, basé sur des équations paramétriques. La prise en compte de la réfraction de la houle sous l'influence du sol, de la réflexion contre les obstacles et de la diffraction derrière les digues, sont proposées dans un nouvel algorithme appelé Lancer Dynamique de Vagues. Celui-ci permet, entre autres, d'échantillonner, de manière uniforme, l'ensemble de la scène. Enfin, un algorithme d'illumination spécifique alliant modèle spectral et espaces colorimétriques standards est présenté.

L'algorithme utilisé pour la réfraction de la houle admet une généralisation tri-dimensionnelle appelée Lancer de Faces. Celle-ci est un nouveau type de lancer de rayons (basé sur les méthodes de Monte Carlo) puissant pour le contrôle des différents types d'aliassages classiques en images de synthèse (aliassage géométriques, de texture, ...).

Mots clefs : *Informatique graphique, Phénomènes naturels, Réfraction, Diffraction, Réflexion, Lancer Dynamique de Vagues, Océan, Aliassage.*

Abstract

Oceanic Scenes Modelisation

Nowadays, computer graphics plays a crucial role. Among other things, it allows us to get a better knowledge of our world by simulating the main natural phenomena (fire, clouds, metal optics, ...). In this Ph.D. thesis, the aim is to modelise geometry and optics of the oceanic surfaces. Furthermore, according to the different constraints of a real scene (bathymetry, breakwaters, ...), we want to determine some of characteristics of water waves (height, wavelength, direction of propagation, ...). To mimic a real oceanic scene landscape as accurately as possible, our research is based on several physics theories.

First, we propose a new algorithm dedicated to the modelisation of wave shapes (breaking and non-breaking waves) in the open sea and near coasts. This algorithm is based on parametric equations. Wave refraction (function of the bathymetry), wave reflection (on obstacles) and wave diffraction (behind breakwaters) are modelised by a new algorithm called Dynamic Wave Tracing. Among other things, it allows to sample the entire scene with the same accuracy. Finally, a illumination algorithm is developed to render the oceanic surface. It combines spectral methods with standard color spaces.

The algorithm used to generate wave refraction is generalised to 3-D. This method, called Face Tracing, is a new type of Monte Carlo ray tracing. It efficiently controls different types of aliasing about computer images (geometric aliasing, texture aliasing, ...).

Keywords : *Computer graphics, Natural phenomena, Refraction, Diffraction, Reflection, Dynamic Wave Tracing, Ocean, Aliasing.*