

A phenomenological model of coastal scenes based on physical considerations

Jean-Christophe Gonzato and Bertrand Le Saëc

*LaBRI**

351 cours de la libération, 33405 Talence (France)

[gonzato|lesaec]@labri.u-bordeaux.fr

In this article, we present a dynamic version of wave tracing previously introduced in computer graphics by Ts'o and Barsky. An empirical model of water waves based on Gerstner, Biesel models and the Fournier method is also proposed.

Keywords. Ocean, water waves, plunging breakers, wave tracing

1 Introduction

The complexity of natural phenomena has led many researchers to try to produce realistic pictures. One such natural object interesting to model is ocean surface flow. An important use is to predict the wave form inside a port or to determine the best place for a sea wall. Other researchers try to predict the arrival of tsunamis on the coast. The ocean surface is complex to model near the shore. A lot of phenomena take part in this complexity: reflection, refraction, diffraction of waves, breaking waves, whirlpools.

Wave modeling can be split into two categories: complete physical modelisation such in [KM90] [FM96] [CL95] that solves Navier-Stokes equations on the one hand; and an empirical approach [FR86] [Pea86] [TB87] [IZ95] based on the classical wave model of Gerstner and Biesel et al. [Bie52] [Lac65] [AA85]. The former leads to very accurate, but expensive algorithms and the latter has the opposite characteristics.

Our work is dedicated to the geometric modelization of waves. A complete rendering model is currently in development, unavoidable stage to animate scenes realistically. As a result, we will not deal with rendering and animation.

In this paper, first, we propose a geometric model of vertical section of the sea that can be viewed as an improved version of the Fournier and Reeves model [FR86]. It can be manipulated easily and advances the Fournier-Reeves limits of modelisation.

Then, we introduce a new algorithm to control the wave refraction based on Descartes law. In comparison to Ts'o-Barsky algorithm [TB87], our model is more precise near the shoreline and can be also used with more general form of floor (baies and islands for instance).

* Laboratoire Bordelais de Recherche en Informatique (*Université Bordeaux I* and *Centre National de la Recherche Scientifique*). The present work is also supported by the *Conseil Régional d'Aquitaine*.

In this model, waves are described from two views: a side view, taking a vertical section of sea; and a bird's eye view, taking an overhead view of sea. The first one helps us to manage the profile of the waves (general form, breaking, beach arrival). The second one is used to control diffraction, refraction and reflection phenomena.

2 Vertical section of the sea

In this section, we first introduce the main classical wave model developed by Gerstner in 1804. In 1952, Biesel enhance this work to introduce plunging breakers. Then, we present the different previous works made in computer graphics. Finally, our model of waves is introduced.

2.1 Classical wave models

Here, we present the Biesel extension of the Gerstner model. Let us recall the assumptions of this model:

- the atmospheric pressure is constant;
- the ocean is composed of a perfect and homogenous liquid;
- the wind is not blowing anymore;
- there are only plunging waves;
- there are no currents and no friction with the bottom.

The *wavelength* is denoted by λ and the *amplitude* A . We denote by $\gamma = \frac{A}{\lambda}$ the *curvature*, T the period of the wave and h the *depth*. The free surface is the surface of sea with no undulations. The *reference point* is situated in the open sea, the x -axis is directed horizontally to the beach, the y -axis is horizontal and perpendicular to the x -axis, and the z -axis is directed to the top. (Fig. 1)

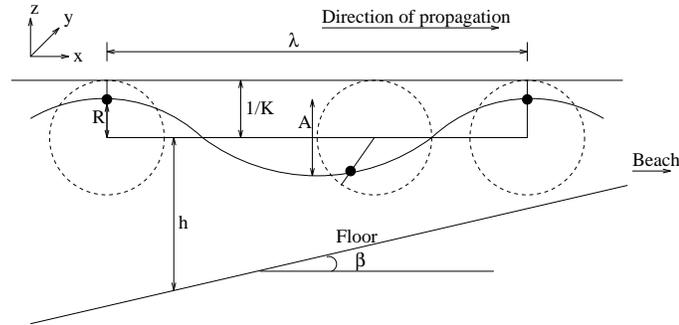


Fig. 1. General definitions.

A first theory was proposed by Gerstner in 1804. It is rigorous for an infinite depth. Each particle of water revolves around a fixed point $M(x_0, z_0)$ describing a circle of radius R . This circle is included in a disc whose radius is $\frac{1}{K}$. K is the wave number, ω is the angular speed.

The (x, z) coordinates of each particle are :

$$\begin{cases} x = x_0 - R_0 e^{Kz_0} \sin(Kx_0 - \omega t) \\ z = z_0 + R_0 e^{Kz_0} \cos(Kx_0 - \omega t) \end{cases} \quad (1)$$

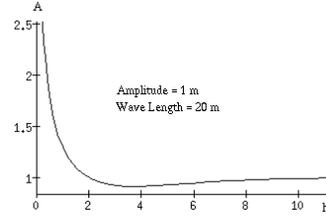
In order to simplify, we set $R = R_0 e^{Kz_0}$. Some data can be determined by physical laws : $\lambda = \frac{2\pi}{K} = \frac{gT^2}{2\pi}$, $A = 2R$, and free surface $z_0 = \frac{\pi A^2}{4\lambda}$. The maximal curvature is obtained when $R = \frac{1}{K}$, so $\gamma_{max} = \frac{A_{max}}{\lambda} = \frac{2}{\lambda} = \frac{1}{\pi} = 0,31$.

Gerstner theory is usable with a great depth, but the influence of the bottom on waves is essential near the shore. When waves arrive on the beach, only the period seems to be invariant [Kin65], the wavelength decreases. All variables marked with an ∞ are used at an infinite depth. In the sequel, h is the depth at the current point x .

$$\frac{\lambda}{\lambda_\infty} = \tanh(Kh) \quad (2)$$

The amplitude decreases progressively to 91% of its open sea value and it increases to the breaking point.

$$\left(\frac{A}{A_\infty}\right)^2 = \frac{1}{\tanh(Kh)\left(1 + \frac{2Kh}{\sinh(2Kh)}\right)} \quad (3)$$



Biesel improves the previous model showing that the circles have to be progressively replaced by ellipses with the major axis oriented along the ground slope when the depth is taken in account. This change entails the formation of breaking waves near the beach.

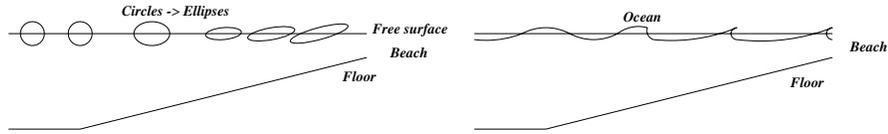


Fig. 2. From Biesel

2.2 Previous work

A first approach uses sinusoidal functions [Pea86]. With this kind of function, we get only one height per point (x, y) of sea. So this point can be stored in a height field, a matrix where are stored the different heights of the sea surface. It is not possible to simulate plunging breakers. The second type of modelization is directly based on Gerstner waves [FR86][IZ95]. Imamiya-Zang propose to use a genetic algorithm to obtain the shape of the waves. Their function simulates a breaking wave, but is limited by a lot of hypotheses. For instance, the slope of the bottom must be constant. The Fournier-Reeves model is the base of our work and will be described in the next part.

Fournier-Reeves model. [FR86] Their waves follow physical laws the most closely. The authors adapt Biesel equations to an approximation easy to compute.

The model is controlled by three factors. A scale factor K_0 determines the influence of the depth on the slope of the ellipses. K_x is an enlargement factor of the ellipse major axis. K_z is a reduction factor of the minor axis. These factors are mainly ranged from 0 to 1. β is the slope of bottom (fig 1).

$$\text{Fournier-Reeves model} \begin{cases} x = x_0 + R \cos(\alpha) S_x \sin(\Phi) + R \sin(\alpha) S_z \cos(\Phi) \\ z = z_0 - R \cos(\alpha) S_z \cos(\Phi) + R \sin(\alpha) S_x \sin(\Phi) \\ S_x = \frac{1}{1 - e^{-K_x h}}, S_z = S_x (1 - e^{-K_z h}) \\ \sin(\alpha) = \sin(\beta) e^{-K_0 h} \\ \Phi = -\omega t + \sum_0^{x_0} K(x) \Delta x \\ K(x) = \frac{K_\infty}{\sqrt{v \tanh(K_\infty h)}} \end{cases} \quad (4)$$

With this model, the terrain cannot have a negative bottom slope because the waves would break on the reverse wave train propagation (fig. 3 : sixth crest). It is sometimes difficult to control the function on its arrival on the beach, because it predicts that the water goes below the ground, which is impossible. This comes from the fact that the major axis tends to ∞ when h goes to 0. So manipulating each scale factor is unavoidable.

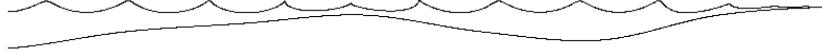


Fig. 3. A limitation of Fournier-Reeves model.

2.3 Our model

We want to create a geometric model with more intuitive parameters, that an average user can easily handle. So, we start our model by computing the general shape of waves using a comprehensive parametric model. This one allows us to combine waves coming from many points of the ocean. We add three new functions named *Stretch*, *Orientation* and *Displacement*. The different parameters governing these ones are easy to manipulate by the user. The Stretch function is used to simulate the acceleration of particles at the crest of the waves. The Orientation and Displacement functions are combined to simulate the influence of gravity. This allows us to simulate plunging breakers.

In the Fournier-Reeves model, which is the base of our model, the major axis of each ellipse tends to ∞ when $h \rightarrow 0$. So the coefficients K_0 , K_x and K_z are necessary to control the general form of the wave near the coast. This becomes a problem when, in the scene, there are many types of varying depths (gentle and steep slopes).

In the Gerstner model, the circles described by the particles of water are restricted to the disc (equivalent to $R \leq \frac{1}{K}$). When $R > \frac{1}{K}$, loops in the wave shape may occur, which does not occur in nature (fig 4.left). An other point is that the major axis of the ellipses becomes greater than the main disc radius. This increase simulates the beginning of plunging waves at medium depth

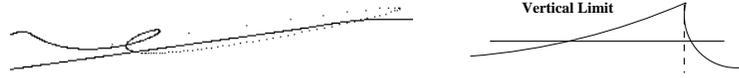


Fig. 4. Left : A problem with Fournier model ; Right : a limitation with our first model.

but, near the shore, the shape wave goes below the ground and the model is inoperative.

Therefore, we propose to limit the major axis of the ellipses to the disc. Since the main disc tends to 0 when h goes to 0, this modification guaranties a predictable arrival of waves on the beach. The limit shape we obtain is presented in fig 4.right.

At this step of our modelisation, two limitations appear: the amplitude of the wave decreases near the shore, and no starting plunging breakers can occur. But, we do not need to modify the parameters (K_0 , K_x and K_z) to adapt the wave shape to the ground definition. We want to manipulate the wave shape by more comprehensive parameters. $K_0 = 0.1$, $K_x = 0.11$ and $K_z = 0.09$ seems to be the best values for our model. They have been chosen empirically after many simulations.

Stretch. In order to reproduce the Biesel laws, we propose to add an enlargement factor to our first model. This factor allows us to stretch progressively waves on the crest in the direction of the major axis while making no modification to the trough of the waves. The maximal stretch length St_{max} is chosen by the user. We use also a factor scale named K_s which determines the influence of the depth on the stretch function.

A parabolic function of phase Φ named $Stretch(\Phi, \lambda)$ is used. The trough of the sea is determined by this equation (by searching the minimum of z in the parametric equation of ellips).

$$\Phi_{min} = atan \left(\frac{\sin(\alpha)S_x}{\cos(\alpha)S_z} \right) \quad (5)$$

$$Stretch(\phi, St_{max}) = \frac{1}{\pi^2}(St_{max}\phi^2 - 2St_{max}\Phi_{min}\phi + St_{max}\Phi_{min}^2). \quad (6)$$

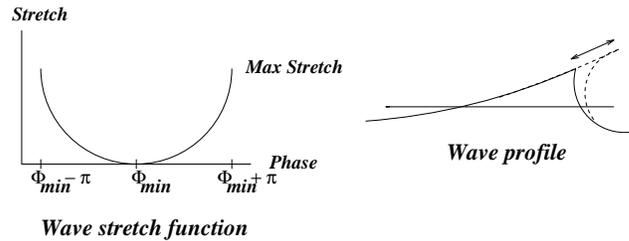


Fig. 5. A stretch function

Plunging waves adaptation. The stretch function is not sufficient for representing real plunging breakers. In fact, due to gravity, water falls down. So we propose to add an orientation function for crests. Our new function modifies the shape on the crest and progressively adds another stretch that decreases in the direction from the angle of the major ellipse axis to the vertical.

This orientation function (named $Orientation(\Phi)$) is combined with a progressively vertical displacement function (named $Displacement(\Phi, A)$). We limit this displacement to A in order to avoid any touching between the crest and the trough of the sea. Moreover, we use a factor scale named K_d which determines the influence of depth on the displacement function.

At the top of the crest, the particle speed is important, but just below it, this velocity is weak, so we have to reconnect to the original wave form more slowly (part 1 of the function). We decide to cut these functions in three empirical parts :

- $\Phi_{min} - \pi \leq \phi < \Phi_{min} - \frac{\pi}{3}$ for the descendant part.
- $\Phi_{min} - \frac{\pi}{3} \leq \phi < \Phi_{min} + \frac{7\pi}{8}$ for the trough part,
- $\Phi_{min} + \frac{7\pi}{8} \leq \phi < \Phi_{min} + \pi$ for the ascendant part.

$$Orientation(\phi, \beta) = \begin{cases} \text{if } (\Phi_{min} - \pi \leq \phi < \Phi_{min} - \frac{\pi}{3}) \Rightarrow \\ \text{the equation of the line passing through} \\ (\Phi_{min} - \pi, -\frac{\pi}{4}) \text{ \& } (\Phi_{min} - \frac{\pi}{3}, \beta) \\ \\ \text{if } (\Phi_{min} - \frac{\pi}{3} \leq \phi < \Phi_{min} + \frac{7\pi}{8}) \Rightarrow \beta \\ \\ \text{if } (\Phi_{min} + \frac{7\pi}{8} \leq \phi < \Phi_{min} + \pi) \Rightarrow \\ \text{the equation of the line passing through} \\ (\Phi_{min} + \frac{7\pi}{8}, \beta) \text{ \& } (\Phi_{min} + \pi, -\frac{\pi}{4}) . \end{cases}$$

$$Displacement(\phi, A) = \begin{cases} \text{if } (\Phi_{min} - \pi \leq \phi < \Phi_{min} - \frac{\pi}{3}) \Rightarrow \\ \text{the equation of the line passing through} \\ (\Phi_{min} - \pi, A) \text{ \& } (\Phi_{min} - \frac{\pi}{3}, 0) \\ \\ \text{if } (\Phi_{min} - \frac{\pi}{3} \leq \phi < \Phi_{min} + \frac{7\pi}{8}) \Rightarrow 0 \\ \\ \text{if } (\Phi_{min} + \frac{7\pi}{8} \leq \phi < \Phi_{min} + \pi) \Rightarrow \\ \text{the equation of the line passing through} \\ (\Phi_{min} + \frac{7\pi}{8}, 0) \text{ \& } (\Phi_{min} + \pi, A) . \end{cases}$$

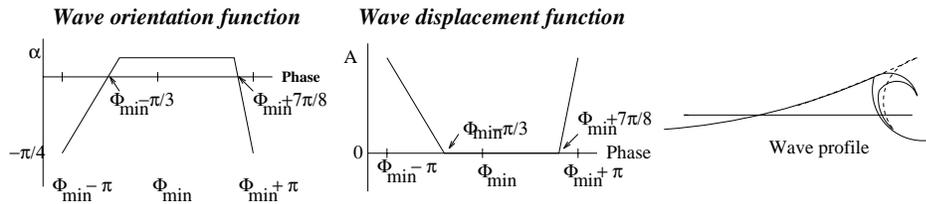


Fig. 6. Plunging breaker.

2.4 Negative slopes.

With Fournier-Reeves model, it is not possible to use negative slopes. Indeed, with negative slopes, the waves break on the reverse wave train propagation. So, in order to avoid this problem, we limit the slope of major axis of ellipse to a positive angle.

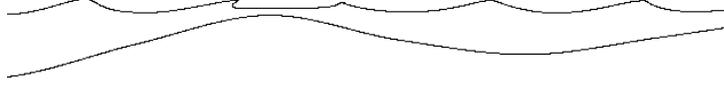


Fig. 7. Wave profile with negative slopes.

2.5 The general form of equations

$$\begin{cases} x = x_0 + R\tau'_\beta S_x \sin(\Phi) + R\tau_\beta S_z \cos(\Phi) + Stretch(\Phi, St_{max})\tau'_\beta e^{-K_s h} \\ \quad + Displacement(\Phi, A) \cos(Orientation(\Phi, \beta))e^{-K_d h} \\ z = z_0 - R\tau'_\beta S_z \cos(\Phi) + R\tau_\beta S_x \sin(\Phi) + Stretch(\Phi, St_{max})\tau_\beta e^{-K_s h} \\ \quad + Displacement(\Phi, A) \sin(Orientation(\Phi, \beta))e^{-K_d h} \end{cases} \quad (7)$$

with

$$\tau_\beta = \sin(\beta)e^{-0.1h}, \quad \tau'_\beta = \sqrt{1 - \tau_\beta} \quad (\beta \text{ is the slope bottom}).$$

$$S_x = \frac{1}{1 - e^{-0.11h}} \quad (\text{Increase of major axis})$$

$$S_z = S_x(1 - e^{-0.09h}) \quad (\text{Decrease of minor axis})$$

$$\Phi = -\omega t + \sum_0^{x_0} K(x) \Delta x .$$

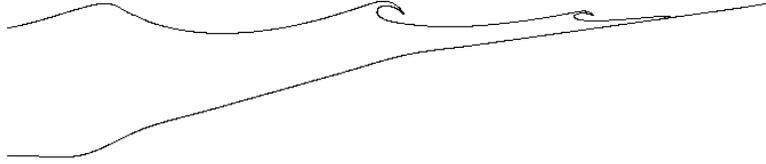


Fig. 8. An example using our model.

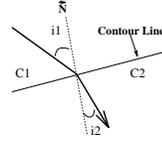
3 Wave trains

3.1 Principles

In order to have a complete description of the waves near the shore, we have to position the wave crest lines. The shape of a crest line depends on the form of ocean floor it passes over. Refraction of waves is the principal phenomenon of the depth effect [Kin65][Lac65]. Waves deflect, like light rays through different media. They tend to align themselves slowly to the contour line and, by extension, to the beach. We can apply the refraction given by Descartes law to wave

trains by using wave orthogonals and the velocity C_1 and C_2 of waves before and after the contour line.

$$\frac{\sin(i_1)}{C_1} = \frac{\sin(i_2)}{C_2} \quad (8)$$



3.2 Previous work

Fournier-Reeves, Peachey. The wave trains built by these models are only formulated using the depth. They only take refraction into consideration in the (x,z) plane. But to model islands and bays, we must include refraction effects in the whole scene i.e. the (x,z) and (x,y) planes.

Ts'o, Imamiya. Wave-refraction calculation was introduced in computer graphics by Ts'o and Barsky [TB87]. Like the ray-tracing algorithm, wave-tracing launches orthogonal wave rays from the open sea along the direction of wave propagation. Descartes's law (eq. 8) is applied to calculate the deflection of the rays. These rays are embedded in uniform grid. The authors approximate the propagation of rays by the well-known Bresenham algorithm. At each point of the grid encountered, the algorithm computes the height of the wave using a 2D model as above. Imamiya follows this work [IZ95].

With this algorithm, it is possible to visualize the refraction factors in bays and along islands. But two problems remain: the first one comes from the fact that 2 consecutive rays can diverge a lot, and so many details can be missed: it is too inaccurate to deal with bays and islands. Second, with the models derived from the Gerstner-Biesel wave model, it is impossible to generate plunging waves (in this case, we often need three values per point on the height field) : the algorithm is inoperative.

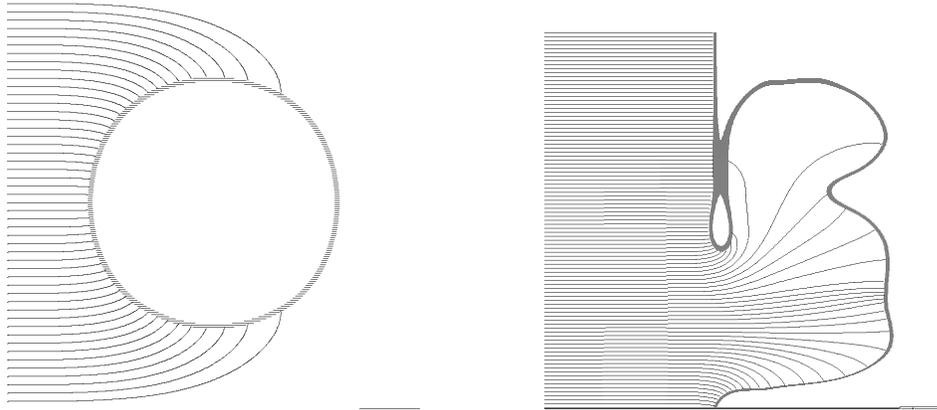


Fig. 9. Ts'o-Barsky wave tracing for an island and a bay

3.3 Our algorithm

Principles The idea of our algorithm is simple: in order to preserve the precision, when two consecutive rays diverge too much, we send a new ray between them.

This could lead to generate an infinite number of rays, each of which can have an infinite length! So, we first have to control the lifetime of rays and their appearances. When these problems are solved, it remains to define clearly the computation of the refraction and how to obtain the tessellation of the sea surface.

At the beginning of the algorithm, we create a list composed of rays that come from the open sea. These rays are separated by a same predefined distance. At each step, they progress incrementally in the scene. We compute their initial depths.

With each ray is associated a flag that can have four values : **DEAD**, **ALIVE**, **FLAT** and **AGROUND**. Their use will be detailed later.

For each ray, during the propagation, the corresponding refraction factor is computed. In order to do that, we need to compute the contour line at the current position.

Contour line calculation. The ground is described by an $m \times n$ matrix named *SEAFLOOR*. To determine the depth at any point of the scene, we use a cubic interpolation. As the bottom is not described by the different contour lines, we have to retrieve them. For each point $M(x, y)$ of the scene, we want to extract three data : the depth, the slope and the contour line. We surround the point M by an equilateral triangle. We approximate the contour line by the line passing through the 2 points on the triangle having a depth value of M .

So we can compute the refraction factor at each step of the wave tracing algorithm using equation 8. The ground is thus decomposed into a scale whole step is the step ray progression. Comparatively, in Ts'o algorithm [TB87], the scene is defined by a set of contour lines given by spline curves. This obliges the author to compute the refraction factor only when an orthogonal wave ray crosses the contour lines (fig. 10). With Ts'o-Barsky method, the distance between two sampling points is not constant. This problem implies a loss of precision.

Ray progression. Each ray progresses along the direction of refraction. We search the contour line passing through the point of calculation. We then compute the new refraction direction and propagate the ray one step. The propagation length is fixed by the user.

When a ray leaves the scene, its flag value is changed to **DEAD** and no further computation is associated with it. When a ray touches the beach its flag value is set to **AGROUND**. In this case, the ray follows the seashore. When it is between two other **AGROUND** rays, its flag become **DEAD**. This kind of ray is only useful for eventually generating new rays.

Amplitude computation. When two rays diverge, the main amplitude of the wave decreases. When two rays converge, amplitude increases. When two rays are distant by L_∞ on the open sea, for a local distance L , the local amplitude is :

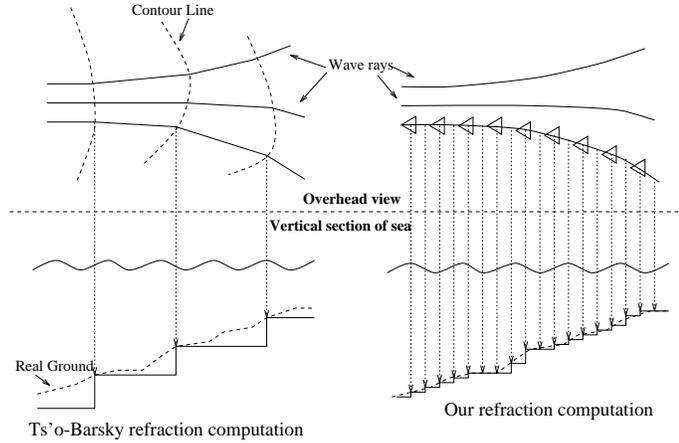


Fig. 10. Computing refraction factors with the Ts'o-Barsky and our model

$$A = A_{\infty} \sqrt{\frac{L}{L_{\infty}}} \quad (9)$$

The value of A_{∞} is in fact $2 \times R_{\infty}$ where R_{∞} is the value of R we use in our model described in the section 2.3. In the same way, A is equal to $2 \times R$.

Over-sampling. A ray is created when the distance between two rays is greater than a threshold value. The new ray is cast. Its current position is computed by interpolation between its two neighboring rays (case 1 fig. 11.Left). When a ray moves away the coast (case 2 fig. 11.Left), our algorithm creates a new ray between the coast and this ray. Since a ray flagged AGROUND, follows the shoreline, we can easily compute the distance between a ray and the coast. The creation of the ray is made by the insertion of an element in the list of rays.

When a ray is flagged DEAD, the corresponding element in the list is removed.

A ray is FLAT when its amplitude is below a given threshold. When two neighbor rays are FLAT, we do not create new rays when the distance is too large. In fact, when rays are FLAT, there are no undulations on the free surface.

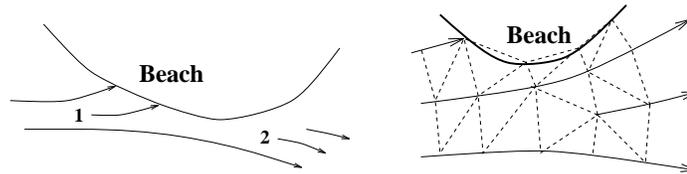


Fig. 11. Left : Two examples of oversampling, Right : Tesselation

Tessellation. After each ray propagation and oversampling, we create two or three facets between each ray position and its last position, depending on the current configuration (fig 11.Right). Facets are created between rays flagged ALIVE, FLAT or AGROUND.

Some crossover may occur between rays. In this case, the two corresponding tessellations are preserved. They will be used for the rendering.

3.4 Results

In figure 9, two different scenes have been introduced. This first one represents a bay and the second an island. In this figure, we apply the Ts'o-Barsky wave tracing.

In figure 12, we apply our dynamic wave tracing on the same scenes. Finally, a bird's eye view of the sea surface is presented in figure 13.

The bay scene is 100 meters long (x axis) and 100 meters large (y axis). We trace one ray per meter ($= 100$ rays) and sample along rays each 0.4 meter. New rays are created if the distance between two rays is greater than 1.2 meter. Our algorithm creates, in this scene, 209 new rays and 39097 facets to define the scene. it runs on an Onyx Silicon Graphics with R10000 processor and takes 2 seconds to compute the entire scene (Dynamic wave tracing and tessellation).

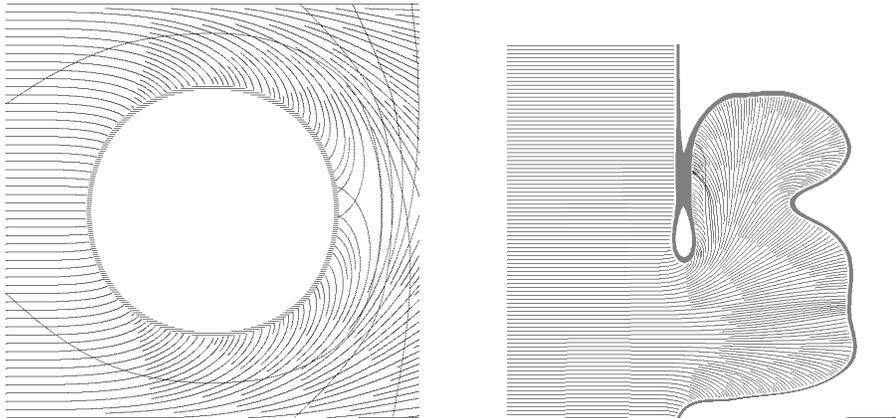


Fig. 12. Dynamic wave tracing for the example of fig. 9

4 Conclusion

In this article, we propose a combination of two algorithms based on physical considerations to modelize ocean scenes. In our model, on the one hand, the manipulation of three intuitive and quasi-independant parameters allows the user a better control on the shape of waves. On the other hand, our *Dynamic Wave Tracing* permits a modelization of more complex scenes than the different previous same type of works.

These tools are integrated in a unique software package (Plates 1-2 see Appendix) and a specific technic of rendering is in the process of elaboration and will be integrated soon.

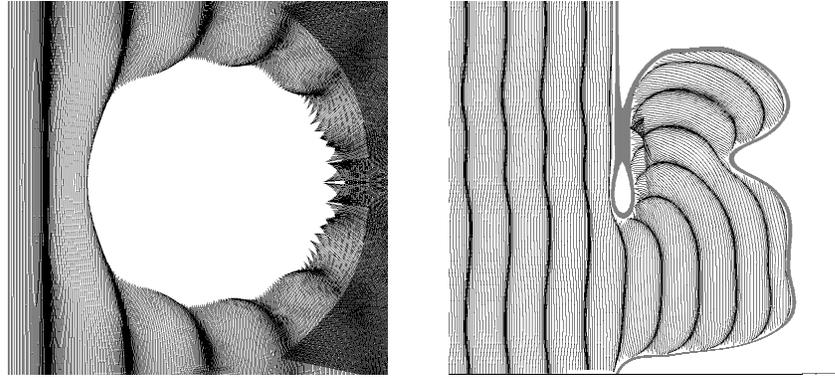


Fig. 13. A bird's eye view of the same scene

In the state of our works, we use Open Inventor for the rendering of our scenes (Plates 3-4 see Appendix).

References

- [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.
- [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.
- [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.
- [FM96] N. Foster and D. Metaxas. Realistic animation of liquids. In *Graphics Interface '96*, 1996.
- [FR86] A. Fournier and W. T. Reeves. A simple model of ocean waves. In *SIGGRAPH'86*, volume 20, pages 75–84, 1986.
- [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.
- [KM90] M. Kass and G. Miller. Rapid, stable fluid dynamics for computer graphics. In *SIGGRAPH'90*, volume 24, 1990.
- [Lac65] H. Lacombes. *Cours d'Océanographie Physique*. Gauthier-Villars, Paris, 1965.
- [Pea86] D. R. Peachey. Modeling waves and surf. In *SIGGRAPH'86*, volume 20, pages 65–74, 1986.
- [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.