

# A new efficient wave model for maritime training simulator

J. M. Cieutat<sup>†</sup>  
J. C. Gonzato<sup>°</sup>  
P. Guitton<sup>°</sup>

<sup>†</sup>LIPSI-ESTIA  
Technopole Izarbel, B.P.215  
F-64102 Bayonne Cedex  
France

j.cieutat@estia.fr

<sup>°</sup>LaBRI\*  
351 cours de la libération  
F-33405 Talence  
France

gonzato@labri.fr  
guitton@labri.fr

## Abstract

The market of maritime training simulators is as important as the market of aeronautical training simulators. Many maritime accidents are caused by piloting errors in critical situations. Such situations are usually due to the environment (waves and stream). Hence, the main challenge of the maritime simulator is to simulate this environment as accurately as possible and more precisely to simulate ocean waves. Nowadays, two kinds of wave simulation theories are presented. Firstly, accurate models are created by physicists. In computer graphics, researchers put forward some more simple physically-based models in order to produce static photo-realistic images. This paper sets out to describe a new efficient real-time model of wave propagation and shows its integration in a real maritime simulator.

**Keywords:** Virtual Reality, Training Simulation, Ocean Waves

## 1 Introduction

There are a great many of maritime training simulators worldwide. This market is as old as the market of aeronautical training simulation but with the following difference. Due to cultural aspects for a long time, maritime learning preferred to limit the use of maritime training simulators to the use of radar simulators only. They progressively changes their outlook towards a more global use of training simulators. In consequence, designing a ship simulator today is becoming the same formidable challenge as designing a flight simulator in the past.

A trainee is required to have a total feeling of being on a real ship, with complete use of all of the instruments needed by navigation like a steering wheel and dashboard, or a radar and sonar, and in addition software of marine

charts visualization. With virtual images, the pilot navigates and fishes in real conditions, in keeping with the behavior of the ships of other trainees, in a region which is familiar to him and where all the conditions of the environment are recreated (seaside, changes in climate, states and depth of sea, species of fish, ...). The challenge has thereby become the same as for a flight simulator : how to create a complete animation and to keep at the same time a great efficiency of virtual rendering of the scene ?

Describing a training simulator in all its aspects could be utopian, the focus of this paper is on the influence of the environment ship and its navigation. The main phenomena which lead a pilot to critical situations are the ocean waves and stream (wind is not discussed in the paper). It is noticeable that major maritime accidents are caused by piloting errors during these critical conditions. The main challenge on a maritime training simulator is to reproduce ocean waves (shape, propagation, ...) as near as possible to the real phenomenon.

In this paper, we aim to deal with this problem : we put forward a new wave model which enables real-time rendering and we show the integration in an existing maritime simulator. In the first section, we present some training simulators. Then, we describe wave models, from purely physical to physically realistic. And, we study how to integrate our wave model to a real simulator.

## 2 The market of maritime simulation

There are two majors companies in this market : NordControl [2] and Transas [3]. The former is a Norwegian company with more than 500 simulator deliveries and more over 25 years of accumulated experience. The latter is a russian company which is taking over a great part of this market thanks to a low-price strategy.

These two companies offer a large range of maritime training simulators. Nevertheless, Nordcontrol company is more well-known for its imposing ship's bridges constituted both of real instruments and virtual renderings of

\*Laboratoire Bordelais de Recherche en Informatique (*Université Bordeaux I, E.N.S.E.I.R.B. and C.N.R.S.*). The present work is also supported by the *Conseil Régional d'Aquitaine*.

scenes while Transas company is more well-known for smaller equipment easy to install in a personal computer environment.

## 2.1 Our simulator : LearnSea



Figure 1: A classroom of future maritime pilots

LearnSea is a maritime simulator developed by the ESTIA [1]. It provides Computer Assisted Interactive Teaching for Navigation and Fishing.

Devised by teachers to meet the needs of initial training, this teaching tool has been designed for students of maritime schools to better assimilate theoretical and practical training within the seafaring professions, to ensure improved school results and better vocational training. The configuration of this tool enables a single teacher from a control post to supervise simultaneously up to six pairs of students working on a simulated navigation bridge (Fig. 1).



Figure 2: A view of the virtual ship approaching a port.

The teaching post of LearnSea comprises various assistance software tools for the administration and follow-up of students, the preparation of simulation exercises with multiple choice questionnaires, and the supervision of the student in training.

There are categories of exercises on the training of maritime signals, the carrying out of manoeuvres, practical navigation, the use of radar, practical techniques of fishing and other numerous categories. There are also levels of graded difficulty on the concentration of ships, sea conditions and stream strength, wind strength and visibility conditions ( rain, fog, ... ).

The ship to be piloted in navigation may be a cargo ship (Fig. 2), oil tanker or even a tug. In fishing the ship may be a trawler or a seiner, and the teacher can input up to 50 other ships riding at anchor and up to 20 other ships in movement.

LearnSea was designed on the principle of active teaching which can be interpreted as follows :

Firstly, the teacher works on a supervision screen which receives in real time the basic indicators concerning ship behaviour of the six simulation bridges with several assistance software tools at his disposal for supervision (Fig. 1). He is also aided by a control system on each bridge informing the student that a manoeuvre has to be carried out, but without giving the solution.

In case of doubt, the teacher can transfer data to the student's post at a given time. In this way, he can connect on to the images of a bridge using his own master screen, decide whether to interrupt the simulation or to re-continue the simulation. He can also change the configuration of the simulation exercise according to the level of each student by introducing new ships, causing breakdowns, shutting down a navigational or fishing instrument, changing the state of the sea or weather forecast.

## 3 Fast simulation of Ocean Waves

Many models exist which can generate ocean scene images. Physicists put forward very accurate models. These theories are usually complex to implement and the result is produced after a lengthy computation time. In computer graphics (or natural phenomena), the different theories presented are easier to implement and the computed images are photo-realistic. Unfortunately, there is no model which can be directly applied to a real-time simulator.

The aim therefore of this work is to create a new physically-based model of ocean waves sufficiently rapid to be integrated into a real-time training simulator. This implies the simplification of physical equations, to carry out some pre-computations and finally to separate complex calculus from the ones found directly.

This section deals with the method used to simulate ocean waves on a real-time maritime simulator. The phenomenon must be modelled as near as possible to the reality. A simulator implies the computation of the wave propagation in a compatible time with virtual reality constraints i.e. at least, the simulation must be fluid.

First, we introduce quickly the main types of physical ocean wave modelling. Second, the previous works in natural phenomena modelling are introduced. Finally, our method proposes to separate the wave simulation into two

parts : in the open sea – where the sea bed has no influences on the wave propagation – and near the coast. The limit is done by the depth. The sea bed has an influence on wave propagation when depth is less than half a wavelength.

### 3.1 Introduction to physical modelling

The most recent theories [18, 6] use a finite element decomposition of Navier-Stokes equations which are usable for all types of fluids dynamic. The different phenomena modifying the wave propagation (refraction, diffraction, reflection, breaking waves, ...) are directly simulated by the Navier-Stokes equations. The implementation of these theories is usually difficult and simulating a large scene entails a long computation time. Logically, Navier-Stokes could not be applied in real-time simulators because of the cost of computation. Therefore, since the next section describes the state of art of wave simulation in computer graphics, we do not intend to present the works based on Navier-Stokes equations (see. [7, 13, 5]).

The second type of theory aims to decompose the wave propagation in two parts : a profile view to model the waveshape and a bird's view for the refraction, diffraction and reflection. The waveshape is modelled by semi-empirical parametric equation. The physicist Gerstner, in 1804, put forward a first theory [9, 16] which is rigorous for 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 disk whose radius is  $\frac{1}{K}$ .  $K$  is the wavenumber and  $\omega$  the angular speed. The  $(x, z)$  coordinates of each particle are (Fig. 3) :

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

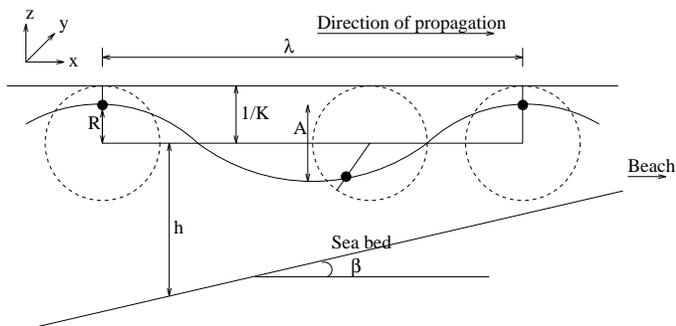


Figure 3: General definition of wave

The Gerstner theory is usable at great depth i.e. when the depth is greater than half a wavelength. In shallow water, the circles described by the particles of water become progressively ellipses with the major axes oriented along the sea bed slope. This change entails the formation of breaking waves near the beach. This theory was put forward by Biesel [4].

The refraction phenomenon is classically modelled by Snell-Descartes laws. It is detailed in the next section. For further details of various physical theories, refer to [14].

### 3.2 Previous works in natural phenomena modelling

Few works have been done on the wave modelling theme. As far as we know, no article treats the problem of how to produce wave propagation as near as possible to the real phenomena in a real-time simulator. All of them are devoted to photo-realistic wave generation and produce animations in a non-applicable time for a common maritime simulator. Let us introduce the main works in the domain.

The first and well-known theory was put forward by Fournier and Reeves in 1986 [8]. The authors enhanced the Gerstner and Biesel models by simplifying the various parametric equations. Their method does not take in account refraction, reflection and diffraction phenomena which are common in a port for instance. A similar model – based on the Z-Buffer algorithm for the wave rendering – is put forward by Peachey [15] in 1986. T'so and Barsky [17], in 1987, aim to simulate waveshape by a simple sinusoid and axe their research on refraction modelling according to the varying ground. The method, called wave-tracing, consists in simulating wave refraction as light refraction. Instead of computing the modification of wave crest lines, it is easier to compute the direction of propagation (with orthogonal to wave crest). The velocity of wave decreases when the depth decreases. The direction of propagation of each orthogonal is computed by applying the Snell-Descartes law according to the velocity.

In the next two sections, models are presented, adapted from our first method developed to generate photo-realistic images of ocean both of the open sea and coastline [10, 11, 12].

### 3.3 Simulation near coastline

**Dynamic wave tracing introduction.** In order to have complete description of waves near the shore, we have to position the wave crest lines. The shape of the crest line depends on the form of the ocean sea bed it passes over. In [10] an algorithm called dynamic wave tracing provides a geometrical definition of coastal scene with refraction is described in detail. A brief introduction follows in the next paragraphs.

The surface of the ocean near the coast is computed by tracing rays – orthogonal to wave crest – from the open sea to the beach. First, in order to simplify, only one main direction of propagation is assigned to waves. Step by step, the refraction deflection is calculated. The refraction is computed by the change of the wave velocity according to the depth. In order to preserve the precision, we send a new ray between two consecutive rays when they diverge too much (Figure 4). The waveshape is computed along these rays by an improved version of the Fournier-Reeves

wave [8]. The model of waveshape is done in equation 2. The details of this model are done in [10]. Rapidly, our waveshape model controls efficiently the arrival of wave on the beach and simulate the beginning of breakers phenomenon. The system coordinate used is presented in figure 8. A tessellation is done between each rays.

The waveshape model is :

$$\begin{cases} x = x_0 + R\tau'_\beta S_x \sin(\Phi) + R\tau_\beta S_z \cos(\Phi) \\ z = z_0 - R\tau_\beta S_z \cos(\Phi) + R\tau'_\beta S_x \sin(\Phi) \end{cases} \quad (2)$$

with

$$\tau_\beta = \sin(\beta)e^{-0.1h}, \quad \tau'_\beta = \sqrt{1 - \tau_\beta}$$

$$S_x = \frac{1}{1 - e^{-0.11h}}$$

$$S_z = S_x(1 - e^{-0.09h})$$

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

$$K(x) = \frac{K_\infty}{\sqrt{\tanh(K_\infty h(x))}}$$

where  $\beta$  is the slope bottom,  $\tau_\beta$  and  $\tau'_\beta$  the slope of the ellipse,  $S_x$  the increase factor of the major axis,  $S_z$  the decrease factor of the minor axis and  $K_\infty$  the wave number at the open sea.

The time of computation – refraction computation and global tessellation – is about 5 s on a Silicon Graphics O2 for our static test scene. This scene is 100 m long and 100 m wide and we sample along rays every 0.4 m. New rays are cast when the distance between two consecutive rays is greater than 1.0 m.

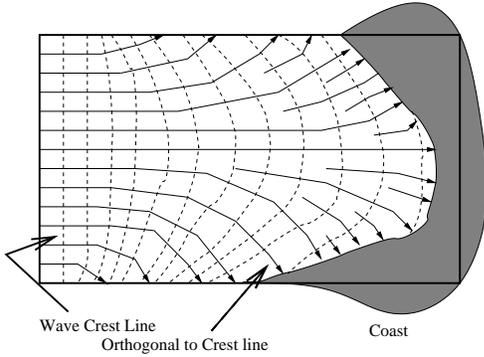


Figure 4: The wave crest lines near the coast

**Adaptation to real-time maritime simulator.** In order to adapt this model to real-time maritime simulator software, we have to pre-compute ray paths and waveshape. The first limitation in our modelling is to use only one wave-train. Thus, we have only one refraction ray path to compute per point of the surface of the ocean near the coast.

The coast and the coastal waters influenced by refraction are cut into user-defined regions (Figure 5). For each region, a computation of ray paths is done and the tessellation is stored in a file. The different bounding boxes composing the shore are not necessary parallel and can overlap

the same part of a scene. The different files are first composed of the description of each point of the tessellated surface and second, by a classic list of vertices defining the different faces like in a Inventor file. The data stored per point of the ocean surface tessellation are :

- $x_0, z_0$  the position of the particle if there is no undulations,
- $V_x, V_y$  define the direction of propagation i.e. the director vector of the ray passing over  $x_0, z_0$ ,
- $\phi = \sum_0^{x_0} K(x)\Delta x$  . the non-temporal dependent phase,
- $A = R\tau'_\beta S_x, A' = R\tau_\beta S_z, B = R\tau'_\beta S_z$  and  $B' = R\tau_\beta S_x$  the four coefficients defining the ellipse.

In order to compute  $\phi$  for each tessellated point of the different boxes defining the surface near the coast, a virtual linear wave crest is placed on the scene (Figure 5).  $\phi$  is fixed to the value zero as start point for each wave ray-path computation.

For each frame of the real-time animation we have to compute the new time-dependent position  $M(x_M, y_M, z_M)$  of each point of the surface by :

$$\begin{cases} x_M = x_0 + (A \sin(\phi - \omega t) + A' \cos(\phi - \omega t)) \times V_x \\ y_M = y_0 + (A \sin(\phi - \omega t) + A' \cos(\phi - \omega t)) \times V_y \\ z_M = z_0 - B \cos(\phi - \omega t) + B' \sin(\phi - \omega t) \end{cases} \quad (3)$$

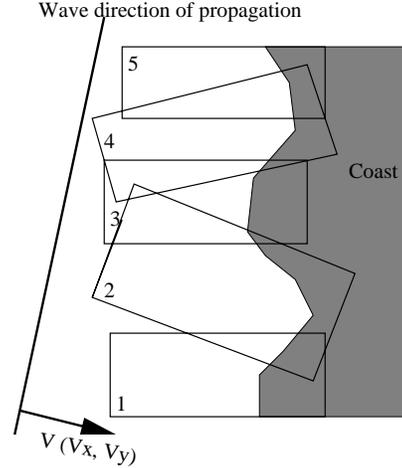


Figure 5: Separate pre-computations of wave propagation near the coast

### 3.4 Ship in the Open Sea

In order to have a complete description of wave propagation and shape near a ship navigation in the open sea, all previous work – dynamic wave tracing – made on refraction computation is unnecessary. The physical model of Gerstner is rigorous for great depths. It is assumed that,

for our application, a based-Gerstner model is applicable for depths greater than half a wavelength.

Our previous model (equation 2) is based on the Gerstner [9] and Biesel [4] models. Looking more closely at these parametric equations, it can be noticed (when depth increases) that :

- $\tau_\beta$  tends to zero,
- $\tau'_\beta$  tends to one,
- $S_x$  tends to one,
- $S_z$  tends to one.

In the open sea, the equation 2 is equal to :

$$\begin{cases} x = x_0 + R \sin(K_\infty x_0 - \omega t) \\ z = z_0 - R \cos(K_\infty x_0 - \omega t) \end{cases} \quad (4)$$

Where  $K_\infty$  is the wave number at the open sea.

Classically, the rendering of the whole ocean surface is approximated by the main visible part of sea from the ship used for the training simulation. A fixed hierarchical grid is created around the ship and follows it during its ocean navigation. Outside the grid, the ocean surface is approximated by a flat facet. At each frame during simulation, the position of the vertices of the grid is re-computed using equation 4.

In order to apply equation 4,  $x_0$  has to be determined by the fixed virtual linear crest line – noted in the following VCL – (Figure 5).  $x_0$  is equal to the distance between the point of computation and the wave crest fixed line (Figure 6)

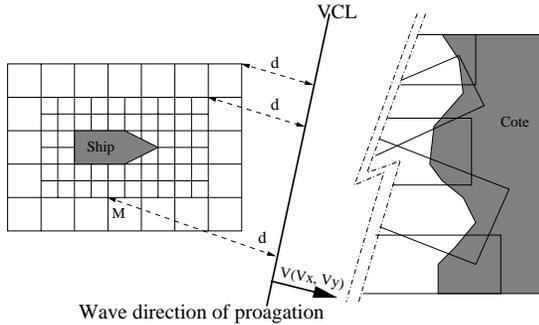


Figure 6: Wave computation around the ship

For each frame of the real-time animation the new time-dependent position  $M(x_M, y_M, z_M)$  of each point of the surface is computed by :

$$\begin{cases} d = \text{distance between VCL and original } M(x_0, y_0, z_0) \\ x_M = x_0 + (R \sin(K_\infty d - \omega t)) \times V_x \\ y_M = y_0 + (R \sin(K_\infty d - \omega t)) \times V_y \\ z_M = z_0 - R \cos(K_\infty d - \omega t) \end{cases} \quad (5)$$

### 3.5 Co-habitation of the two models

With the two previous models, we can simply limit the computation of the wave simulation. Indeed, for real-time simulation, when the ship is situated in the open sea, only the local grid fixed around the ship has to be computed (section 3.4). The sea bed does not influence the wave propagation.

When the ship approaches near a pre-computed zone, the corresponding file is automatically loaded and the animation appears instantaneously (section 3.3). The frame rate decreases due to the amount of computation.

When the ship navigates in a pre-computed zone, the fixed grid around the ship wave modelling is abandoned in order to limit computations. Although, more than one zone file could be loaded. Indeed, the number of loaded pre-computed file depends on the position of the ship and where it is facing.

Finally, it can be noticed that there is no problem of junction between a fixed pre-computed file describing a zone of surface near the shore and the grid around the ship passing next to this same zone. As the original equations and original crest line are the same, the surfaces are identical.

## 4 Ship point of view

Specific nautical terms, used in the following section, are introduced on the figure 7.

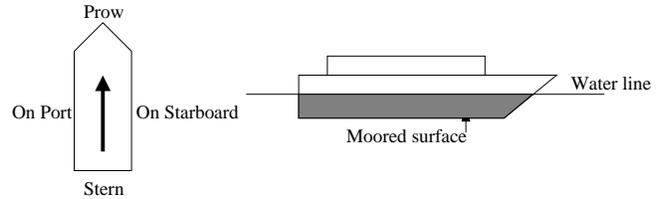


Figure 7: nautical vocabulary.

A great number of forces influencing a ship's movement over the ocean surface have to be taken into consideration to compute at each frame its new position and its new transversal and longitudinal inclinations. For instance, relating to our study, stream can simply be simulated on the propagation of the ship. Supposing the stream is a force with a constant value, we can directly add the velocity of this actor to the velocity of the ship during its course.

However, it is not as simple to compute swell influences on a ship as stream influence. Therefore, in order to produce a comprehensible explanation, the most visible effects of the swell on the ship are introduced first, in the following section.

One of the main effects is a vertical translation along Z-axis while the second effects are two rotations following Y-axis and X-axis respectively called pitching and rolling in nautical domain (Fig. 9). The ship translation in the Z direction is due to the nearest wave. When the amplitude

and the length of this wave are great enough, the ship is raised first to then go down after the wave top. When a wave approaches the front of the ship, its prow is raised while its stern is goes down to become the contrary after the wave top (pitching). When a wave approaches on port the ship is bent on starboard to become the contrary after the wave top (rolling).

To compute the visible effects of the swell on the ship, we need to extract some informations from the nearest wave hitting the ship first. These computations are detailed in the next section. We assume that the ship is a single point which is the center of gravity (Fig. 8).

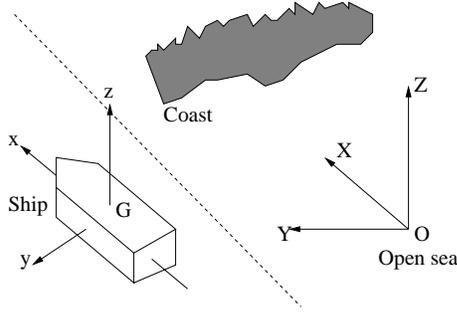


Figure 8: Coordinate system.

Starting from these preliminary computations, the following section presents an overview of the different possibilities we have to integrate the new efficient real-time model of wave propagation in our simulator. These possibilities are graded by levels of increasing complexity with the knowledge that the higher a level is, the more physically accurate the results are.

#### 4.1 Computations of swell characteristics at the ship position

##### 4.1.1 Computations of swell characteristics when the ship is in the open sea

**Sea height at the ship position.** The sea height at the ship position  $S(S_x, S_y, S_z)$  is simply determined by  $T_Z = -R \cos(K_\infty d - \omega t)$  where  $d$  is the distance between VCL and the ship position  $S$ .

**Inclination angles of the swell.** To compute inclination angles of the swell transversally and longitudinally to the ship, we need to know the slope of the nearest wave where it intersects the ship first. This slope - the tangent plane of the surface - is classically computed by the derived function of the equation 5.

One snapshot of the ship is presented Fig. 10.

##### 4.1.2 Computations of swell characteristics when the ship is near the shore

The computation of the surface is not dependent on the position of the ship as we have seen previously in the open

sea. The ship navigates on a pre-computed surface (refer to section 3.3). In our Dynamic Wave Tracing model, all rays - orthogonal to wave crests - are independent of each other. The physical wave shape computation is done along the ray. The surface is extrapolated by a tessellation between each consecutive ray.

In order to compute inclination angles of the swell, we cannot directly use the equation 2 because the ship can be positioned between two rays. The physical description of the surface is only done by the face under the ship.

Thereby, to compute sea height at the ship position, we must directly use the elevation of the face located under the ship. In the same way, to compute the slope of the surface hitting the ship, we must directly use the normal of this face. Some explanations are given in the following.

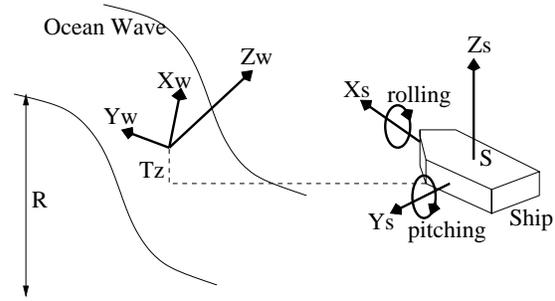


Figure 9: Inclination angles of the swell.

In order to compute the inclination angles of the swell transversally and longitudinally to the ship, we use the two local axis systems :  $(X_s, Y_s, Z_s)$  axis system for the ship and  $(X_w, Y_w, Z_w)$  axis system for the wave or the face (Fig. 9).

Before any computation, the plane  $(X_s, Y_s)$  of the ship is the plane of the sea without swell and at the end of computations, the  $Z_s$  axis must be aligned to the  $Z_w$  axis.

Hence, the first formula which enables to place  $Z_s$  in the plane  $(Y_w, Z_w)$  of the wave is the inclination angle of the swell longitudinally to the ship. This angle, which is a rotation following the  $Y_s$  is determined by :

$$\theta = -\arccos\left(\frac{(\vec{Z}_w - (\vec{Z}_w \cdot \vec{Y}_s) \cdot \vec{Y}_s) \cdot \vec{Z}_s}{\|\vec{Z}_w - (\vec{Z}_w \cdot \vec{Y}_s) \cdot \vec{Y}_s\|}\right) \quad (6)$$

Following that, the other formula which ultimately enables to align the  $Z_s$  axis with the  $Z_w$  axis is the inclination angle of the swell transversally to the ship. This angle, which is a rotation following the  $X_s$  axis, is equal to  $\mu = -\arccos(\vec{Z}_w \cdot \vec{Z}_s)$ .

#### 4.2 An overview of different levels of integration

##### 4.2.1 Geometric results

The first approach is geometric only. Starting from the sea height under the ship, inclination angles of the swell transversally and longitudinally to the ship (section 4.1), we directly deduce translation in the  $Z$  direction, pitching

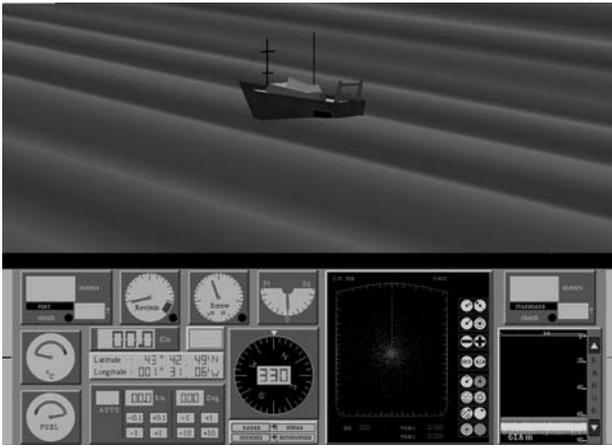


Figure 10: A ship in open sea

and rolling of the ship. Movements of ships and waves are similar. These kinds of results cannot be used for a training simulator because they are unrealistic.

#### 4.2.2 Mechanical results with static hypothesis

Static hypothesis means that inclination forces on the ship are slow and progressive. In this case, the ship reaches an equilibrium position with no angular speed. In our simulator, the equilibrium position is computed for each frame.

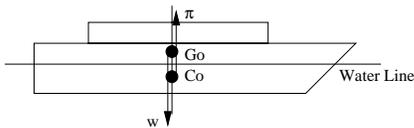


Figure 11: Water line of a ship

In the following, we explain why movements of ships and waves are not synchronous. We then present consequences when ship is moving forward.

When a ship is still on a sea with no wave, its water line is invariant due to the equilibrium between the forces facing each other (refer to the Fig. 11 where the ship weight  $w$  (applied on the gravity center  $Go$ ) and the Archimède's vertical push  $\pi$  (applied on the center of the hull under water  $Co$ ) are illustrated). Its moored surface is the main parameter which have to be taken into account to study ship behavior. In this special configuration, it is invariant.

What happens when characteristics of the ship environment change due to the swell now ?

Let us assume the ship is always still, its moored surface is not invariant when the sea height under this ship is changing. Because of inertia, when a ship is raised or goes down due to the swell, the equilibrium position is not immediately reached and its moored surface is changing.

In the same way, ship inclinations are not synchronous with waves inclinations.

On Fig. 12, we can look at the change of the gravity center and the center of the hull under water when ship is

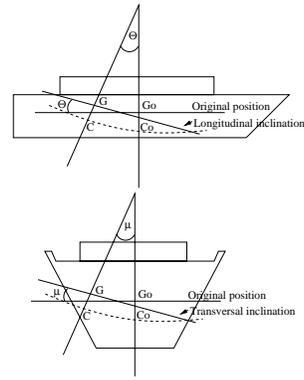


Figure 12: Longitudinal and Transversal stability of a ship

inclined longitudinally and transversally. This problem can be compared with pendulum swinging.

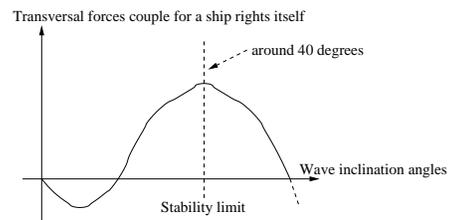


Figure 13: Transversal stability curve

The maximum of stability is reached when the ship center of gravity and the center of its hull under water are on the same vertical (refer to the Fig 11). Whatever the wave inclination is, the ship will directed towards this position. Because of the twice couples of forces which permits to a ship to right itself, its inclinations are not synchronous with waves inclinations. We can deduce that the moored surface of a ship is not invariant when its environment is changing due to the swell.

In order to propose a realistic training simulator, we saw that it is important to integrate in our computations ship inertia and couples of forces which permits to right a ship. For example, a couple of forces can be deduced from the corresponding curve delivered any time a ship is built (refer to Fig. 13).

We have just seen that inclinations of ship and waves are not synchronous (moored surface varies). Thereby, delimited to our study (wind for example is not discussed here, ...), when a ship is moving forward, hydrodynamic resistance is depending on its forward movements (its speed, its gyration) but also on the variations of moored surface due to the swell.

Hydrodynamic resistance to a moving forward direction is computed starting from the general law of hydrodynamic. The equation of this law is :

$$F = \frac{\rho c S V^2}{2} \quad (7)$$

Where  $\rho$  is the voluminal mass of water,  $c$  a coefficient

depending on the number of Reynolds,  $S$  the surface perpendicular to a moving forward direction and  $V$  the speed of a ship following this direction (perpendicular to the surface).

At the end, each hydrodynamic resistance creates a couple of forces defined by :

$$C = Fd \quad (8)$$

Where  $F$  is the hydrodynamic resistance and  $d$  the lever arm.

The general law of hydrodynamic is very appropriate since it takes into account moored surface of a ship. Variations of surfaces under water due to the swell must be integrated now.

#### 4.2.3 Mechanical results with dynamic hypothesis

Dynamic hypothesis is more accurate. It means that inclination forces on the ship can be quick and unprogressive. In this case, since the ship reaches an equilibrium position with angular speeds, the accurate position is beyond the equilibrium position. In our simulator, this difference is not yet taken into account.

## 5 Conclusion

In this article, we aim to simulate ocean environment (stream and ocean waves) on a real maritime simulator. Conscious of the main phenomena modifying the behaviour of the ship propagation, future pilots can learn how to react in critical situations. We put forward a new real-time algorithm to simulate ocean wave propagation. In order to limit calculations, the model proposes to cut ocean surface into two parts. The first is the open sea where the sea bed has no effects to the waves. Physical equation can be directly applied. The second, near the coast, needs a pre-computation of the phenomena modifying the wave propagation (refraction, ...). In the last part of this article, we present how the waves modify the behaviour of a ship. Finally, some plates of the simulator are presented (as on board : sonar, radar and sounder views ; as external views).

## References

- [1] <http://www.estia.fr>.
- [2] <http://www.kongsberg.com>.
- [3] <http://www.transas.com>.
- [4] 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.
- [5] 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.
- [6] C. A. J. Fletcher. *Computational techniques for fluid dynamics*, volume 2. Springer series in computational techniques, 1991.
- [7] N. Foster and D. Metaxas. Realistic animation of liquids. In *Graphics Interface '96*, 1996.
- [8] A. Fournier and W. T. Reeves. A simple model of ocean waves. In *SIGGRAPH'86*, volume 20, pages 75–84, 1986.
- [9] F. J. Gerstner. Theorie der wellen. (*reprint*) *Ann des Physik*, 32:412–440, (1802) 1809.
- [10] 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.
- [11] J. C. Gonzato and B. Le Saëc. On modelling and rendering ocean scenes. *Journal of Visualization and Computer Animation*, 11(1):27–37, 2000.
- [12] J.C. Gonzato. *Modélisation des scènes océaniques*. PhD thesis, LaBRI, Université Bordeaux I, Décembre 1999.
- [13] M. Kass and G. Miller. Rapid, stable fluid dynamics for computer graphics. In *SIGGRAPH'90*, volume 24, 1990.
- [14] B. Kinsman. *Wind Waves, their generation and propagation on the ocean surface*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1965.
- [15] D. R. Peachey. Modeling waves and surf. In *SIGGRAPH'86*, volume 20, pages 65–74, 1986.
- [16] 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.
- [17] 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.
- [18] Y. Xu and al. Physically based simulation of water currents and waves. *Computer & Graphics*, 21:277–280, 1997.

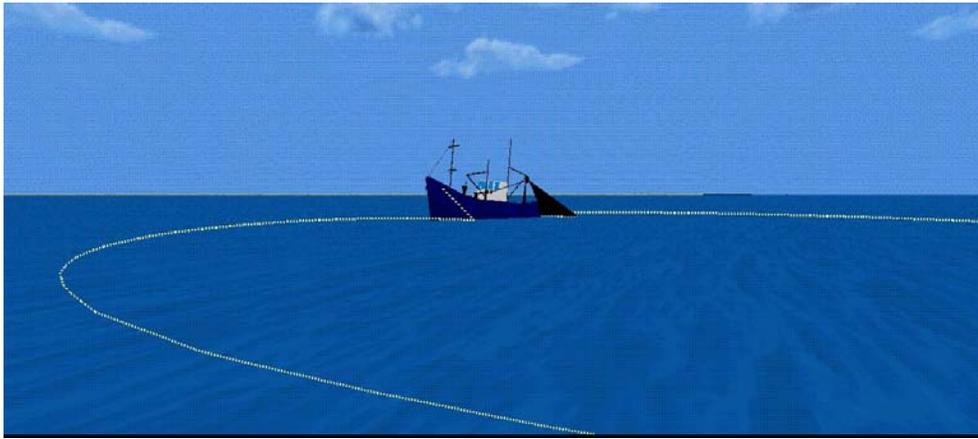


Plate 1: Fishing with seine



Plate 2: View from the bridge



Plate 3: The virtual sonar



Plate 4: The virtual radar