

# Simulação Visual de Ondas Oceânicas em Tempo Real Usando a GPU

Alex Salgado    Aura Conci    Esteban Clua

Universidade Federal Fluminense, Niterói, Brasil

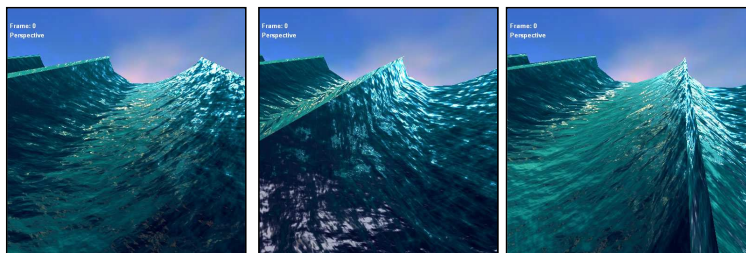


Figura 1: Simulação de ondas em tempo real usando equações oceanográficas

## Abstract

Modeling natural phenomena is one of the most challenging tasks in computer graphic. The ocean wave simulation is included in this challenge. With the GPU graphical processing, it is possible to use advanced rendering techniques with great realism in real-time applications. This work simulates the ocean wave behavior processing all geometric computation and rendering in GPU. The shape is defined by Gerstner's equation where the water particles movement is simulated. It is possible to consider both the representations: deep water and shallow water. The method also simulates the breaking waves near the shore and the wave refraction, which is the topology deep sea influence on its movement. The implementation of this work uses combinations of advanced real-time rendering techniques of tangent-space reflective bump mapping and environment mapping to Fresnel reflection and HDR. As illustrated in session 6, this technique creates a very realistic waves animation and adjusted scenario.

**Keywords:** GPU, real-time rendering, ocean waves, water, breaking waves, natural phenomena, vertex and fragment programming

### Authors' contact:

{asalgado,aconci,esteban}@ic.uff.br

## 1. Introdução

Elementos da natureza (terrenos, nuvens, água, árvores, etc) são objetos complexos de serem modelados devido à sua natureza fractal de geometria. Por outro lado, são de fundamental importância na criação de cenários virtuais, tanto para jogos com para aplicações de simulação e visualização. A natureza por si só já é complexa, mas hoje é possível gerar imagens impressionantes, graças à modelagem de fenômenos naturais utilizando alguns modelos físicos e estatísticos. Entre esses, pode-se citar a simulação de líquidos, fogo e gás (Fedkiw, Stam e Jensen,

2001)(Foster e Fedkiw, 2001). No entanto, criar e renderizar águas é uma das tarefas mais onerosas da computação gráfica. Uma renderização realística da água requer que a incidência da luz do sol e a iluminação do céu estejam corretas, que a modelagem da superfície da água esteja real e que o transporte de luz no corpo da água e o seu movimento seja captado corretamente.

Até pouco tempo os jogos tratavam a água como uma superfície plana, limitando-se a aplicar uma textura de modo a assemelhar-se com a cor da água, não sendo consideradas as suas propriedades físicas.

A maioria dos trabalhos considera a modelagem geométrica das ondas através de duas formas de modelagem: física ou empírica. No modelo físico, adota-se a equação de Navier-Stokes, que é baseada na dinâmica dos fluidos para representar o movimento da água através do tempo (Kass e Miller, 1990) (Foster e Metaxas, 1996) (Chen e Lobo, 1995). Devido a seu grande custo de processamento, este modelo ainda é inviável para processamento em tempo real. A abordagem com base totalmente física requerida para estudos científicos é bem diferente da abordagem para jogos em termos de precisão e fidelidade dos cálculos. A outra forma de modelagem é através dos modelos empíricos, os mais conhecidos são de Fournier e Reeves (1986), Peachey (1986), Ts'o e Barsky (1987), Imamiya e Zhang (1995). Estes modelos são baseados no modelo clássico de ondas de Gerstner (Kinsman, 1965) em que a busca para a representação da fidelidade visual é maior do que a fidelidade aos fenômenos físicos.

O objetivo deste trabalho é modelar e visualizar a superfície das águas do oceano em tempo real na GPU, representando inclusive efeitos complexos como a influência do vento, do relevo do fundo do mar e o efeito de ondas se quebrando. Para isto será desenvolvido um modelo de representação da geometria e um sistema de visualização de ondas, através da aproximação empírica. Este sistema poderá

ser usado em jogos, simuladores ou em aplicações que necessitem representar oceanos em tempo real.

Neste artigo será utilizado a GPU (Graphic Processing Unit) para o processamento dos vértices e dos fragmentos. A equação da superfície trocóiide (uma generalização de uma ciclóide) é usada como base para a modelagem da forma da onda de acordo com o trabalho realizado por Fournier e Reeves (1986). A animação e a renderização é feita utilizando um modelo de sombreamento (shader) programado na linguagem Cg da NVIDIA, sendo aplicadas técnicas de reflexão por bump mapping no espaço da tangente, environment mapping, reflexão de Fresnel (Wloka, 2002) e HDR (High Dynamic Range).

## 2. Trabalhos relacionados

Tem havido grande evolução das técnicas de computação gráfica na modelagem realística da água, bem como na sua visualização e animação (Iglesias, 2004). A água, por ser um fluido, pode mover-se em direções e caminhos complexos. A variável “tempo” deve ser incluída nas equações para garantir o movimento do fluido e movimentação adequada.

O termo realismo possui diferentes significados na computação gráfica e depende exclusivamente do objetivo a que se deseja alcançar. Enquanto numa aplicação de visualização científica o modelo de iluminação de cena não é a prioridade, mas sim a precisão dos comportamentos, num jogo espera-se uma iluminação real, mas não uma física precisa. Tendo isto em conta alguns trabalhos recentes abordam diferentes níveis de realismo na simulação de águas (realismo físico e foto-realismo) (Adabala e Manobar, 2002) (Ferwerda, 1999).

A primeira tentativa de visualizar ondas da água utilizou a técnica de bump mapping (Blinn, 1978). Esta técnica também é usada neste trabalho. Este método permite obter superfícies com rugosidades realistas através da perturbação da normal da superfície modelada, mas não permite a criação de ondas grandes.

Fishman e Schachter (1980) introduziram a técnica de “height field” e posteriormente Max (1981) desenvolveu outro método visando o realismo na simulação do oceano, principalmente a visualização da textura do mar quando a câmera está próxima da superfície. Seu modelo hidrodinâmico foi baseado na idéia de que um modelo de onda é representado por uma solução aproximada (válido apenas para ondas de pequenas amplitudes) em que a velocidade da onda  $v$  é proporcional a raiz quadrada do comprimento de onda  $L$ .

$$v = \sqrt{\frac{gL}{2\pi}} = \sqrt{\frac{g}{k}} \quad (\text{Eq 1})$$

onde  $k=2\pi/L$  - é chamado o número da onda (o espacial análogo da frequência) e  $g$  é a força da gravidade.

Max também usou a primeira aproximação linear do modelo de Stokes (uma série de Fourier infinita a qual se assemelha as ondas trocoidas quando a série é usada com termos além das de terceira ordem).

Entre 1986 e 1988, com a preocupação da interação entre sólido e líquido, foram desenvolvidos técnicas que simulam refração e colisões. Dentre estes trabalhos, destaca-se o sistema de partículas de Reeves (1983) usados na simulação de espumas nas ondas. Peachey (1986) considerou que a superfície pudesse ser modelada usando “height field” e também conseguiu grande realismo em seu trabalho. Fournier e Reeves (1986) é um clássico sempre referenciado nos trabalhos atuais, baseado no modelo de Gerstner-Rankine proposto na oceanografia séculos atrás (Gerstner, 1809).

No modelo de Fournier-Reeves, os autores assumem que a partícula da água descreve uma órbita estacionária circular ou elíptica. Este é o modelo geométrico adotado neste trabalho, como será discutido na seção 4). Formas de ondas realistas e outros efeitos necessários, tais como os relacionados à profundidade (refração e quebras) e o vento, podem ser reproduzidos variando-se alguns parâmetros da equação da órbita deste modelo. Para controlar o formato do oceano Fournier e Reeves (1986) introduziram alguns “trens de ondas”, i.e. grupos de ondas compartilhando as mesmas características básicas (altura, período e amplitude de onda) e a mesma fase original. No trabalho Fournier e Reeves (1986) também foram gerados efeitos de spray e espuma.

Até o final de 1988, alguns fenômenos físicos relacionados à água que ainda não havia sido explorados, começaram a ser considerados. A maior questão era a descrição exata da dinâmica de fluidos. Por outro lado, alguns efeitos interessantes não haviam sido considerados até então, tais como: simulação de ondas refletidas, a interação entre luz e água, a análise caótica, etc. Para superar essas limitações, vários novos modelos foram criados para simular a dinâmica de fluidos. A princípio, estes novos modelos podem ser agrupados em dois grupos de diferentes abordagens: Dinâmica de fluidos baseada na interação de um número grande de partículas e dinâmica de fluidos descrita pela solução de um conjunto de equações diferenciais parciais.

Os trabalhos relacionados à primeira abordagem foram os trabalhos de Sims (1990) e Tonnesen (1991) onde os autores estudaram as forças de atração e repulsão entre as partículas para simular vários graus de viscosidade do fluido e o estado da mudança da matéria tal como o derretimento.

A segunda abordagem procura resolver diretamente um sistema de equações diferenciais parciais (PDE) descrevendo a dinâmica de fluidos (Kass e Miller, 1990). Esta técnica cria resultados realistas em termos de física, mas devido a uma simulação muito refinada

da dinâmica de fluídos requer um cálculo computacional do movimento dentro de um determinado volume de controle. Seguindo esta abordagem de pesquisa, os trabalhos desenvolvidos foram simplificando estes cálculos tornando sua utilização viável computacionalmente.

Um modelo bastante realístico pôde ser obtido usando a equação de Navier-Stokes, a mais detalhada de todos os modelos de fluídos. Devido a esta característica e algumas simplificações dos termos, a equação de Navier-Stokes tem sido amplamente utilizada na computação gráfica para simulação do movimento da água (Chen e Lobo, 1995) (Chen, Lobo, Hughes e Moshell, 1997) (Foster e Metaxas, 2000) (Witting, 1999).

O que se tem visto é que o movimento da água é um fenômeno bastante complexo e variável, incluindo efeitos difíceis de serem analisados. A descrição exata destes efeitos é normalmente caracterizada pela área da física denominada dinâmica de fluídos ou fluidodinâmica computacional (CFD). Recentemente vêm se destacando nesta área pelo seu grande realismo e aplicações em efeitos especiais de cinema os trabalhos de Fedkiw, Stam e Jensen (2001), Geiger, Leo, Ramussen, Losasso e Fedkiw (2006), Losasso, Fedkiw e Osher (2006) e Irving, Guendelman, Losasso e Fedkiw (2006). Finalmente, seguindo a linha de Fournier e Reeves (1986) e Tessendorf (1999), pode-se citar o trabalho de Gonzato e Le Saec (2000) que também tentam simplificar a física e se preocupar com a visualização realística da cena.

### 3. Simulação física de oceanos

O sistema proposto para a simulação da onda na GPU é composto de 4 estágios: geração da onda, modelagem da superfície, computação óptica e renderização da água. A figura 2 mostra estes estágios e suas conexões. Os dois primeiros estágios serão realizados dentro da programação de vértice. A geração da geometria da onda será definida utilizando-se a equação de Gerstner (Tessendorf, 1999) com trajetória circular para simular a geometria em águas profundas e trajetória elíptica para simular a geometria em águas rasas. Como a proposta do trabalho é a construção de um shader na forma de efeito, a modelagem da superfície será feita de uma forma muito simples: será aplicado um shader sobre uma superfície geometricamente simples. No caso deste trabalho, será considerado um retângulo no plano XZ.

Esta é uma das vantagens da solução proposta sob o ponto de vista de reutilização e ganho em produtividade, pois pode ser facilmente incluído em qualquer game engine, apenas referenciando-se a um conjunto de shaders. Sobre o modelo, será aplicado o bump mapping para dar o efeito de rugosidade na geometria da superfície peculiar as micro ondas que se formam no oceano. Para os dois últimos estágios será

desenvolvido um programa de fragmentos. A computação da óptica irá calcular a iluminação, reflexão de Fresnel e HDR. No último estágio, unindo os três anteriores e a contribuição da cor da água, finalmente será renderizado o oceano.

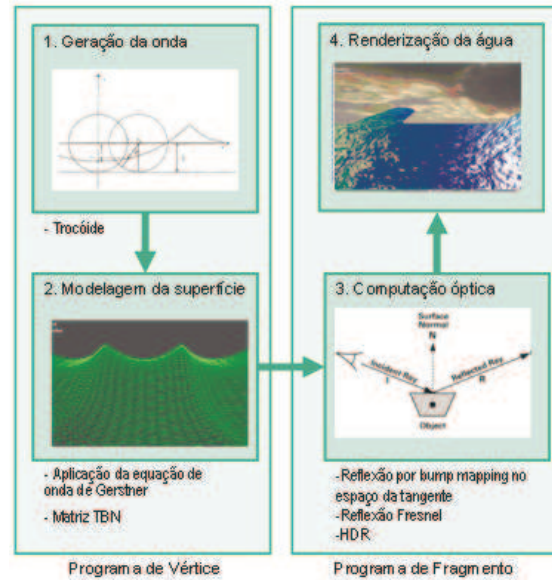


Figura 2 – Estágios do sistema de simulação

#### 3.1 O modelo básico da onda

Usa-se aqui o modelo apresentado por Fournier e Reeves (1986). O movimento dos fluidos geralmente é descrito por duas formulações: Euleriana ou Lagrangiana. A formulação Euleriana é mais adequada na hidrodinâmica e ao estudo das ondas, especialmente para o desenvolvimento de modelos estocásticos na análise do mar. Considera-se um ponto  $(x, y, z)$  e procura-se responder questões sobre as características do fluído neste ponto em função do tempo, como por exemplo, a velocidade:

$$U = f(x, y, z, t) \quad (\text{Eq. 2})$$

A formulação Lagrangiana, a princípio é mais apropriada para modelagem gráfica por tratar o oceano como sendo uma primitiva geométrica. Ela descreve a trajetória de um ponto  $(x_0, y_0, z_0)$  dado por uma posição de referência. Isto pode ser visto como a trajetória de uma partícula. Por exemplo, pode-se saber a velocidade no tempo  $t$ :

$$V_x = f_x(x_0, y_0, z_0, t), \quad V_y = f_y(x_0, y_0, z_0, t), \quad V_z = f_z(x_0, y_0, z_0, t) \quad (\text{Eq. 3})$$

#### 3.2 Ondas de Gerstner

Para uma simulação efetiva do oceano, deve-se controlar a agudez (steepness) da onda. A onda perfeita tem a forma de uma senoide – como uma onda num lago calmo. Entretanto, para simular o oceano é necessário criar cristas com picos afinados e calhas

arredondadas. Para representar esta onda com mais realismo será utilizado o modelo de onda de Gerstner. O modelo físico descreve a superfície em termos de movimentos de pontos individuais na superfície.

Será considerado que uma partícula descreve um movimento circular a partir de sua posição de repouso. O plano XZ é o plano do mar em repouso e Y representa a coordenada de altura ao plano da superfície do mar. Considerando o movimento no plano XY, a equação de Gerstner simplificada será o sistema:

$$\begin{aligned} x &= x_0 + r \sin(kx_0 - \omega t) \\ y &= y_0 - r \cos(kx_0 - \omega t) \end{aligned} \quad (\text{Eq. 4})$$

Onde:

$H = 2r$  é a altura da onda;

$k = 2\pi/L$  é o número de onda;

$L = 2\pi/k = gT^2/2\pi$  é o comprimento de onda;

$T = 2\pi/\omega$  é o período;

$c = L/T = \omega/k$  é a celeridade da onda (velocidade da fase) ou seja, a velocidade de viagem da crista.

Observando-se a equação 4 como uma equação paramétrica percebe-se que se trata de uma trocóiide, uma generalização da cicloíde. Esta equação representa a curva descrita por um ponto  $P$  que tem distância  $r$  do centro de um círculo de raio  $1/k$  que se move circularmente sobre uma linha de distância  $1/k$  sob o eixo X (figura 3). Portanto, amplitude  $A = r$ ,  $x_0$  e  $y_0$  são as coordenadas iniciais do mar em repouso,  $\omega$  é a frequência,  $t$  o tempo e  $k$  o número da onda.

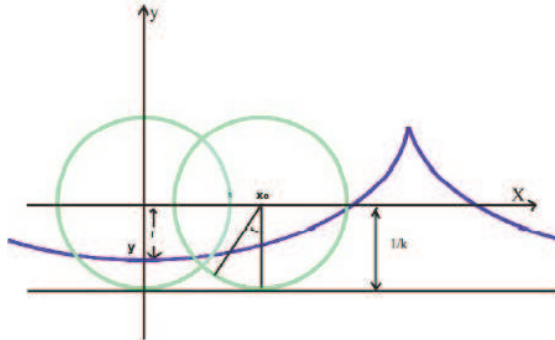


Figura 3 – Movimento de uma partícula na trocóiide

Assumindo  $\Phi = kx_0 - \omega t$  como sendo a fase da onda, pode-se reescrever a equação como:

$$\begin{aligned} x &= x_0 + r \sin(\Phi) \\ y &= y_0 - r \cos(\Phi) \end{aligned} \quad (\text{Eq. 5})$$

Com este modelo básico é possível chegar as formas desejadas para uma cena realista. Por exemplo, alterando-se  $kr$ , obtêm-se várias formas de onda como se pode ver na figura 4.

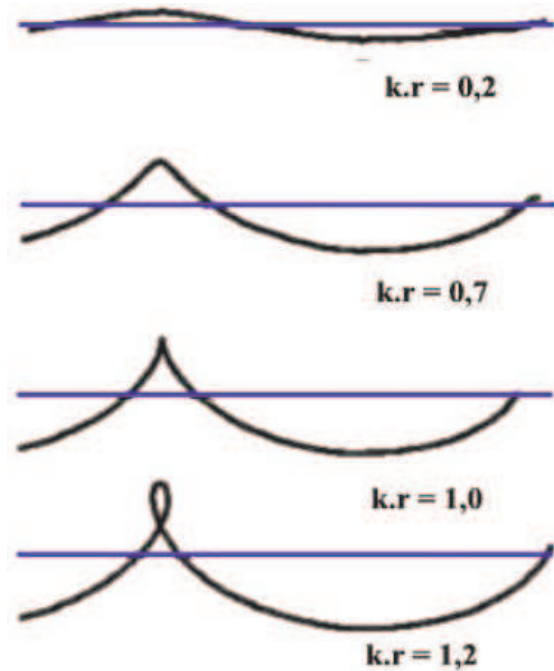


Figura 4 – Formas de onda variando o parâmetro  $kr$ .

### 3.3. Consideração dos estágios do sistema

A partir deste modelo básico, será definida a forma da onda bem como seu sombreamento. Desta forma, foi criado um um programa de vértice e um programa de fragmento. O programa de vértice definido implementa a equação de onda de Gerstner com variações dos atributos para simular a influência do fundo na formação da onda como a refração, influência do vento e quebra. O programa de vértice também calcula a matriz para o cálculo da iluminação. O programa de fragmento utiliza as informações recebidas do programa de vértice para calcular a iluminação e definição das cores finais.

## 4. Implementando a equação de Gerstner na GPU

Este trabalho utiliza a programação em GPU para implementar a simulação de onda. Esta sessão descreve cada passo necessário para obter a forma de onda final desejada.

Como o movimento de cada partícula da água é definido por um traçado circular (figura 5), com o emprego do shader é possível aplicar a equação 4 em cada vértice da figura que define a superfície do mar.



Figura 5 – Posição de uma partícula de água com o movimento da onda

Utilizando a linguagem Cg, este cálculo será efetuado no programa de vértice, que executará os seguintes passos:

1. Receber como entrada um polígono retangular representando a superfície do oceano no plano XZ;
2. Transformar os vértices do polígono utilizando a equação 4;
3. Transformar as coordenadas atuais do espaço do objeto projetado;
4. Enviar dados do vértice transformado para o processador de fragmentos.

#### 4.1. Acrescentando a influência do vento

Uma vez definido a forma básica da onda serão acrescentados alguns efeitos especiais. Um deles é o efeito do vento sobre as ondas, fazendo com que elas sofram uma inclinação na parte superior da onda na mesma direção do vento.

Para acrescentar tal efeito, será adicionado mais um controle no ângulo de fase da equação de Gerstner:

$$\Phi = kx_0 - \omega t - \lambda \Delta y \Delta t \quad (\text{Eq. 6})$$

Onde:

$\lambda$  é uma constante de proporcionalidade do vento

Pela expressão é possível constatar que a partícula será mais acelerada no topo e mais desacelerada na base da onda gerando uma projeção na forma da onda em direção à frente de onda.

#### 4.2. Refração e influência do fundo do mar sobre as ondas

Quando as ondas se propagam de águas profundas para águas rasas (ao se aproximar da costa, por exemplo), as características da onda mudam assim que ela começa a sofrer a influência do solo, mantendo-se constante apenas o período. A velocidade da onda diminui com a diminuição da profundidade. Assim que as ondas passam a sentir o fundo, um fenômeno chamado refração pode ocorrer. Quando as ondas entram na zona de transição (profunda para rasa), a parte da onda em águas profundas move-se mais rapidamente que a onda em águas rasas. Como mostrado na figura 6, essa diminuição na velocidade da fase da onda pode ser percebido numa visão aérea do mar. A medida que a onda sente o fundo do mar e sua velocidade e seu comprimento de onda diminui, a crista da onda tende a se alinhar com a costa marítima. De acordo com a equação:

$$L = cT \quad (\text{Eq. 7})$$

Onde L é o comprimento de onda; c a celeridade da onda; T o período.

Na zona de transição entre águas profundas e águas rasas, a celeridade (c) da onda (m/s) é calculada pela equação:

$$c^2 = \frac{g}{k} \tanh(kh) \quad (\text{Eq. 8})$$

Onde  $k = 2\pi/L$  é o número de onda; g é a aceleração da gravidade e h a profundidade da água.

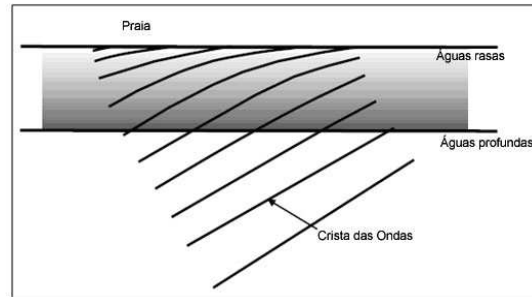


Figura 6 – Refração da onda ao se aproximar da costa (águas rasas).

Em águas profundas, onde a profundidade da água é maior que a metade do comprimento de onda,  $k.h$  é um valor grande de modo que  $\tanh(k.h)$  é aproximadamente igual a 1. Portanto, a celeridade em m/s pode ser escrito como:

$$c^2 = \frac{g}{k} = \frac{gL}{2\pi} \quad (\text{Eq. 9})$$

ou com um produto de celeridades,

$$c * c = \frac{gL}{2\pi}$$

e substituindo  $c = L/T$ , tem-se:

$$c * \frac{L}{T} = \frac{gL}{2\pi}$$

simplificando L,

$$c = \frac{gT}{2\pi}$$

ou seja, supondo  $g = 9,8 \text{ m/s}^2$ ,

$c = T * 1,56$  é a celeridade da onda em águas profundas.

Em águas rasas, onde a profundidade da água (h) é menor que 1/20 do comprimento de onda,  $t.h$  é um valor pequeno, logo  $\tanh(t.h)$  é aproximadamente igual a  $t.h$ . Substituindo  $\tanh(t.h)$  por  $t.h$  na equação 8 e simplificando k, tem-se:

$$c^2 = \frac{g}{k} (kh) = gh \quad (\text{Eq. 10})$$

Extraindo a raiz quadrada, obtém-se:

$c = \sqrt{gh}$ , como sendo a celeridade da onda em águas rasas. A figura 7 mostra a variação das ondas representada nas equações 9 e 10.

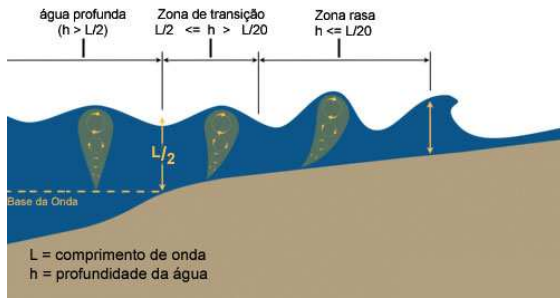


Figura 7 – Transição e variação das ondas

### 4.3. O comprimento de onda de acordo com a profundidade

A diminuição da profundidade da água também altera o comprimento de onda, sendo que o período permanece constante (Kinsman, 1965). Se chamarmos de  $k_\infty$  o número de onda em uma profundidade infinita, uma boa aproximação para o número de onda  $k$  na profundidade  $h$  é dada por:

$$k \tanh(kh) = k_\infty \quad (\text{Eq. 11})$$

Quando  $x \rightarrow 0$ , então  $\tanh(x) \rightarrow x$ . Logo, em águas rasas, onde a profundidade é pequena, a relação fica:

$$k^2 h = k_\infty \quad \text{ou} \quad k = \sqrt{\frac{k_\infty}{h}}$$

Quando  $x \rightarrow \infty$ , então  $\tanh(x) \rightarrow 1$ , logo  $k \rightarrow k_\infty$ . Uma vez que  $k \cdot h = 2\pi h/L$ , com uma relação  $h/L$  de  $1/2$  obtém-se um argumento de  $\pi$  para a tangente hiperbólica o que é praticamente igual a 1. Por esta razão, a “profundidade” tem relacionamento com o comprimento de onda e significa a relação  $h/L$  ser maior que  $1/2$  como citado anteriormente na figura 7. Assim, uma boa aproximação para a equação 11 é:

$$k = \frac{k_\infty}{\sqrt{\tanh(k_\infty h)}} \quad (\text{Eq. 12})$$

### 4.4. A celeridade em relação à profundidade

Uma vez afetado o comprimento de onda, a celeridade da onda também é afetada como mostrado no item anterior, o que significa:  $c/c_\infty = k_\infty/k$  e a onda é refratada assim como a velocidade sofre diminuição. De fato, pode-se aplicar a lei de Snell Descartes (USP-Educar, 2006) para calcular o ângulo que a frente de onda faz ao partir de uma profundidade infinita e entrar numa profundidade  $h$ .

$$\frac{\sin(\theta_h)}{\sin(\theta_\infty)} = \frac{c_h}{c_\infty} \quad (\text{Eq. 13})$$

A figura 8 mostra que quando a onda emitida por A' se desloca até B em um intervalo de tempo  $t$ , a onda emitida por A, neste mesmo intervalo de tempo, sofre

um deslocamento menor até B', considerando que  $v_2 < v_1$ .

Sendo:  $A'B = v_1 \cdot t$  e  $AB' = v_2 \cdot t$

Obtem-se:

$$\frac{A'B}{AB'} = \frac{v_1}{v_2}$$

Da geometria da figura 7, tem-se:

$$\sin \theta_1 = \frac{A'B}{AB} \quad \sin \theta_2 = \frac{AB'}{AB} \quad (\text{Eq. 14})$$

Dividindo as duas equações, obtém-se:

$$\frac{\sin \theta_1}{\sin \theta_2} = \frac{A'B}{AB'} = \frac{v_1}{v_2}$$

Como  $n_1 = v_1 / v_0$  e  $n_2 = v_2 / v_0$ , substituindo na equação anterior, obtém-se a expressão da lei de Snell Descartes:

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{n_2}{n_1}$$

O efeito de profundidade não pode ser calculado considerando-se apenas informações locais, o atraso da fase que é introduzido é acumulativo. Sendo  $k$  é uma função de profundidade, que por sua vez é função de  $x_0$ . Assumindo  $\Phi = 0$  para  $x_0 = 0$ , e que a constante de proporcionalidade  $\lambda = 0$ , a equação da fase agora será:

$$\Phi = -\omega t + \int_0^{x_0} k(x) dx \quad (\text{Eq. 15})$$

onde

$$k(x) = \frac{k_\infty}{\sqrt{\tanh(k_\infty h(x))}}$$

Desta forma será possível simular o efeito de refração da onda ao aproximar-se da costa e receber influência do fundo do mar.

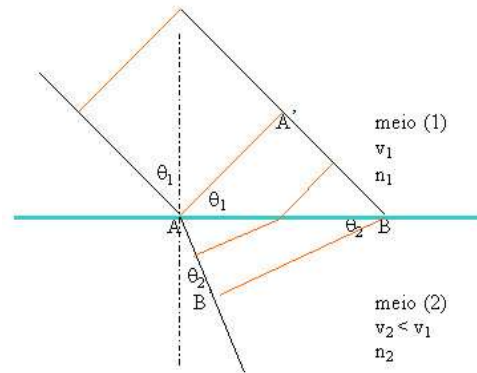


Figura 8 – Frente de onda na refração de acordo com a lei de Snell.

### 4.5. Quebra da onda na costa

As teorias clássicas dizem que a medida que a onda se aproxima da costa, sua trajetória passa a ser elíptica ao invés da circular em águas profundas. Biesel (1952) propôs um modelo em que o eixo maior da elipse se

alinhe com a inclinação do fundo do mar até que a profundidade se torne igual a zero (figura 9 e 10).



Figura 9 – Como a profundidade afeta a órbita



Figura 10 – Como a profundidade afeta a forma da onda

Para adaptar a programação em GPU será utilizada a forma simplificada de Fournier-Reeves (1986) levando em consideração o custo computacional:

$$\begin{aligned} x &= x_0 + r \cos \alpha \cdot S_x \cdot \sin \Phi + \sin \alpha \cdot S_z \cdot \cos \Phi \\ y &= y_0 - r \cos \alpha \cdot S_z \cdot \cos \Phi + \sin \alpha \cdot S_x \cdot \sin \Phi \end{aligned} \quad (\text{Eq. 16})$$

Onde  $\Phi$  é a fase;  $\text{sen} \alpha = \text{sen} \gamma e^{-k_0 h}$ ,  $\gamma$  é a inclinação do fundo do mar em direção a trajetória da onda;  $S_x = 1 / (1 - e^{-kxh})$  é o incremento do eixo maior da elipse;  $S_y = S_x (1 - e^{-kyh})$  é o decremento do eixo menor da elipse;  $K_0$  determina a influência da profundidade na inclinação na elipse;  $K_x$  é um fator de alargamento do eixo maior da elipse;  $K_y$  é um fator de redução do eixo menor da elipse;  $r$  é o raio do disco.

É importante notar que  $S_x \rightarrow \infty$  quando a profundidade  $h \rightarrow 0$ .

Com essa mudança da equação do movimento é possível obter um melhor realismo na forma da onda quando ela se quebra aproximando-se da costa e recebe a influência do fundo do mar.

#### 4.6. Correção do modelo de Fournier-Reeves

O modelo apresentado anteriormente apresenta duas limitações. A primeira delas é que o modelo não permite que o fundo do mar tenha inclinações negativas, pois as ondas quebram-se na direção reversa da propagação da onda. Como pode ser visto na figura 11, esta limitação torna-se muito irreal. Já que uma possibilidade de fundo irregular com depressões e saliências é muito comum na natureza, especialmente em regiões rochosas ou em águas muito agitadas.



Figura 11 – Limitação do modelo de Fournier-Reeves.

Para resolver este problema usando o modelo de Fournier-Reeves, é necessário limitar a inclinação do eixo maior da elipse para um ângulo sempre positivo, considerado zero as inclinações negativas.

O segundo problema vem do fato que no modelo de Gerstner, os círculos descritos pelas partículas de água são restritos ao raio do disco, isto é,  $r \leq 1/k$ . Quando  $r$

$> 1/k$  podem ocorrer laços na equação que representa a forma da onda que não ocorre na natureza como mostrado na figura 12. Neste caso, a solução consiste em fazer com que o eixo maior da elipse receba um valor maior que o disco de raio  $r$ . Esta ação simula o aparecimento da quebra da onda em profundidade média, mas ao aproximar-se da costa, o modelo torna-se inoperante, pois são formados laços na forma da onda. Para este problema, a solução utilizada neste trabalho foi limitar o eixo maior da elipse ao raio  $r$  do disco adquirindo a segunda forma mostrada na figura 12.



Figura 12 – Correção de laços na formação da onda.

Para melhorar a aparência da geometria da queda da onda, aproximando-se das ondas em espiral (quebras mergulhando), Gonzato e Le Saec (1997) propuseram uma alteração na fase da equação de Fournier-Reeves esticando e torcendo a crista da onda como mostra a figura 13. Gonzato adicionou três novas funções chamadas Strech, Orientation e Displacement. A função Strech é usada para simular a aceleração da partícula na crista da onda. As funções Orientation e Displacement são combinadas para simular a influência da força da gravidade.



Figura 13 – Perfil da onda de Gonzato e Le Saec (1997)

#### 4.7. A função Strech

Para manter a forma inicial da onda de Biesel, foi adicionado como parâmetro da função, um fator para esticar a crista da onda em direção ao eixo maior da elipse ao passo que nenhuma modificação é realizada na calha da onda. O parâmetro  $St_{max}$  define este tamanho máximo de alargamento. Também é criado um parâmetro de escala  $K_s$  para determinar a influência da profundidade na função Strech.

Uma função parabólica chamada  $Stretch(\Phi, \lambda)$  de fase  $\Phi$  é usada. O fundo da onda é obtido através do valor mínimo de  $y$  na equação paramétrica da elipse da seguinte forma:

$$\Phi_{\min} = a \tan \left( \frac{\sin(\alpha) S_x}{\cos(\alpha) S_y} \right)$$

E a função Strech é definida como:

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

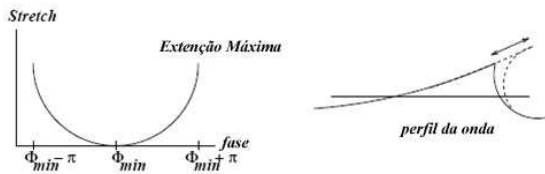


Figura 14 – Função Stretch

#### 4.8. Quebrando a onda

Em complemento a função Stretch, é necessário alterar a forma da onda para representar a força da gravidade fazendo com que a mesma se quebre. Para isso, é criado uma função de orientação da crista da onda. Esta função altera a forma da onda e progressivamente adiciona um novo estiramento que decreta na direção do eixo maior da elipse até a posição vertical.

Esta função de orientação é combinada com uma função de deslocamento progressivo. Este valor é limitado a  $2r$  de modo que a não permitir a colisão da crista com o fundo da onda. Além disso, um fator de escala chamado  $K_d$  é utilizado para determinar a influência da profundidade na função de deslocamento. No topo da crista da onda, a velocidade é importante, mas logo abaixo dela a velocidade é bem menor. Deste modo, esta função é dividida em três partes de forma empírica:

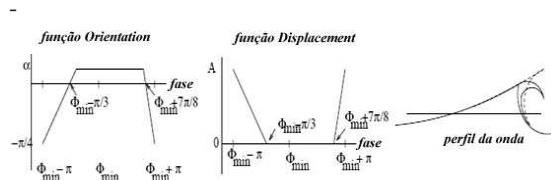


Figura 15 – Quebra da onda

Acrescentando a função Stretch, a função Displacement e a função Orientation, a equação da onda se transforma em:

$$\begin{cases} x = x_0 + R\tau_\beta S_x \sin(\Phi) + R\tau_\beta S_y \cos(\Phi) + \text{Stretch}(\Phi, St_{\max})\tau_\beta e^{-K_d h} \\ \quad + \text{Displacement}(\Phi, 2r)\cos(\text{Orientatia}(\Phi, \beta))e^{-K_d h} \\ y = y_0 - R\tau_\beta S_y \cos(\Phi) + R\tau_\beta S_x \sin(\Phi) + \text{Stretch}(\Phi, St_{\max})\tau_\beta e^{-K_d h} \\ \quad + \text{Displacement}(\Phi, 2r)\sin(\text{Orientatia}(\Phi, \beta))e^{-K_d h} \end{cases}$$

Onde:

$\tau_\beta = \sin(\beta)e^{-0.1h}$ ,  $\tau_\beta' = \sqrt{1 - \tau_\beta}$ , sendo  $\beta$  é a inclinação do fundo;

$S_x = \frac{1}{1 - e^{-0.11h}}$ , é o incremento do eixo maior;

$S_y = S_x(1 - e^{-0.09h})$ , é o decremento do eixo menor;

#### 5. Visualização

O capítulo anterior definiu a modelagem geométrica da onda. Como a água pode apresentar-se com uma aparência diferente dependendo do contexto da cena, é importante definir algumas categorias de efeitos de água. No caso deste trabalho, foram utilizadas técnicas para que a água assemelhe-se a aparência do oceano,

mas não houve contribuição inédita neste tema, sendo que as técnicas aplicadas foram reflexão por environment mapping, efeito Fresnel e HDR.

#### 6. Implementação e resultados

O objetivo deste trabalho consiste em criar uma simulação realística da superfície das ondas da superfície do mar. Para criar uma ferramenta produtiva e reutilizável, foi criado um efeito (arquivo.fx) de modo a simplificar a sua implementação tanto para um programador quanto para um artista 3D.

Utilizou-se o FX Composer 1.8 que é uma IDE para desenvolvimento de shader da NVIDIA® na linguagem Cg (Shader 2.0), e também foi testado aplicando-se o efeito dentro de um software de renderização 3D proprietário, o 3D MAX® da Discreet®. Todas as fotos apresentadas na figura 16 são retiradas da aplicação sendo executada em tempo-real.

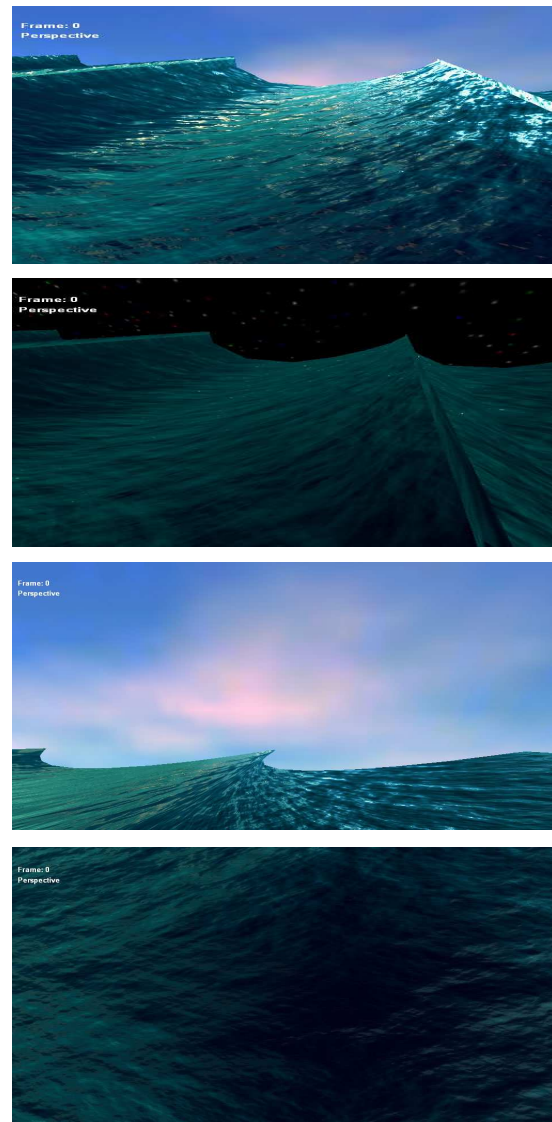


Figura 16 – Resultados em tempo-real



Para visualizar a cena, foi utilizado um computador AMD Athlon XP 1800 com 512 RAM com uma placa de vídeo NVIDIA FX 5200 Shader 2.0.

O shader foi programado de forma que alguns parâmetros pudessem ser configurados pelo usuário. Estes parâmetros definem o comportamento visual e físico da onda do mar. De forma empírica e visual foram configurados valores para estes parâmetros de forma a atingir o objetivo final de uma cena realista. Manipulando-se essas propriedades, é possível alterar a granularidade das ondas capilares; alterar a velocidade da partícula da água, alterar a contribuição da reflexão de Fresnel e HDR no cálculo da iluminação; criar ambientes noturnos, diurnos, sol, estrelas através do cubemap; alterar o período, a frequência e raio da onda e controlar a contribuição do vento.

## 7. Conclusão

Foi proposto um sistema de visualização tempo real para simulação de ondas oceânicas. Para alcançar um nível de simulação que caracterizasse uma aplicação em tempo real, todo o processamento e programação deste trabalho foi direcionado para a arquitetura e o hardware gráfico da GPU.

Foi abordado o comportamento das ondas em águas profundas, águas rasas com a influência do fundo do mar sobre a forma destas ondas. Foi possível simular a modelagem geométrica baseada nas leis físicas e algumas configurações empíricas com atenção para as ondas se quebrando na costa oceânica. Simulou-se também a refração das ondas quando esta sente o fundo do mar e perde velocidade de fase de acordo com a lei de Snell Descartes, alterando as propriedades da onda tais como o comprimento de onda e velocidade, a medida que se aproxima da costa.

Os resultados obtidos foram satisfatórios e convincentes quanto a visualização da superfície de um oceano com formação de ondas. Para aumentar o desempenho da renderização, pode-se acrescentar algoritmos de níveis de detalhe (LOD) para que os polígonos distantes da visão possam ser renderizados com menos detalhes (Bustamante e Celes, 2002).

Para representar a topologia do fundo do mar que foi utilizado nas equações da onda, poderia-se utilizar um mapa de altura que armazenasse estas informações. Assim o programa de vértice poderia ler este mapa e interpretar os valores tornando mais refinado a visualização. Entretanto, este recurso só é possível com o perfil do Shader 3.0 (Gerasimov, Fernando e Green, 2004) utilizando o recurso de Vertex Texture. Neste trabalho não foi gerado o efeito de espuma e bolhas na superfície das ondas. O que pode ser gerado utilizando a física de sistema de partículas e autômatos celulares. Pode-se também gerar o fenômeno de caustics que ocorre na superfície da água (Fernando, 2004).

A crista da onda ficou com uma aparência relativamente regular e reta que não existe na natureza da onda. Seria interessante criar irregularidades e ruídos na formação dessas cristas gerando padrões aleatórios utilizando a função noise.

## 8. Bibliografia

(Adabala e Manobar, 2002) N. Adabala, S. Manohar, Techniques for realistic visualization of fluids: a survey, *Comput. Graph. Forum* volume 21 (1), pp. 65-81, 2002.

(Biesel, 1952) F. Biesel, Study of wave propagation in water of gradually varying Depth, *Gravity Waves*, pp. 243-253, U.S. National Bureau of Standards Circular 521, 1952.

(Blinn, 1978) J.F. Blinn, Simulation of wrinkled surfaces, *Proceedings of SIGGRAPH'78*, *Comput. Graph.* volume 12 (3), pp. 286-292. 1978.

(Chen e Lobo, 1995) J. Chen, N. Lobo, Toward interactive-rate simulation of fluids with moving obstacles using navier-stokes equations, *Graphical models and Image Processing*, pp.107-116, (março 1995).

(Chen, Lobo, Hughes e Moshell, 1997) J.X. Chen, N.V. Lobo, C.E. Hughes, J.M. Moshell, Real-time fluid simulation in a dynamic virtual environment, *IEEE Comput.Graph. Appl.*, pp.52-61, (maio-junho 1997).

(Fedkiw, Stam e Jensen, 2001) R. Fedkiw, J. Stam., H. W. Jensen. Visual Simulation of Smoke, *Proceedings of SIGGRAPH 2001*, pp. 15-22, 2001.

(Ferwerda, 1999) J.A. Ferwerda, Three varieties of realism in computer graphics, *Cornell Workshop on Rendering, Perception and Measurement*, 1999.

(Fernando, 2004) R. Fernando. GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics, Addison-Wesley Pub, 2004.

(Fishman e Schachter, 1980) B. Fishman, B. Schachter, Computer display of height fields, *Comput. Graph.* 5 pp. 53-60, 1980.

(Fournier e Reeves, 1986) A. Fournier, W. T. Reeves, A simple model of ocean waves, *SIGGRAPH'86*, volume 20, pp. 75-84, 1986.

(Foster e Metaxas, 2000) N. Foster, D. Metaxas, Modeling water for computer animation, *Commun. ACM* volume 43 (7), pp. 60-67, 2000.

(Foster e Fedkiw, 2001) N. Foster, R. Fedkiw, Practical Animation of Liquids, *Proceedings of SIGGRAPH 2001*, pp. 23-30, 2001.

- (Geiger, Leo, Ramussen, Losasso e Fedkiw, 2006) Geiger, W., Leo, M., Rasmussen, N., Losasso, F. and Fedkiw, R., So Real It'll Make You Wet, SIGGRAPH 2006 Sketches and Applications, 2006.
- (Gerasimov, Fernando e Green, 2004) P. Gerasimov, R. Fernando, S. Green, Shader Model 3.0: Using Vertex Texture, NVidia Corporation, 2004.
- (Gerstner, 1809) F.J. Gerstner, Theorie der wellen, Ann. der Physik 32, pp. 412-440, 1809.
- (Gonzato, Le Saec, 2000) J.C. Gonzato, B. Le Saec, On Modeling and Rendering Ocean Scenes, Journal of Visualisation and Computer Simulation, 11, pp. 27-37, 2000.
- (Irving, Guendelman, Losasso e Fedkiw, 2006) Irving, G., Guendelman, E., Losasso, F. and Fedkiw, R., Efficient Simulation of Large Bodies of Water by Coupling Two and Three Dimensional Techniques, SIGGRAPH 2006, ACM TOG 25, pp. 805-811, 2006.
- (Imamiya e Zhang, 1995) A. Imamiya, D. Zhang, Modelling breaking ocean waves, influence of floor and refraction, Pacific Graphics 95, 1995.
- (Iglesias, 2004) A. Iglesias, Computer graphics for water modeling and rendering: a survey, Future Generation Computer System, volume 20 (8), pp. 1355-1374, (novembro 2004).
- (Kass e Miller, 1990) M. Kass, G. Miller, Rapid, stable fluid dynamics for computer graphics, ACM Press, New York, NY, USA, 1990.
- (Kinsman, 1965) B. Kinsman, Wind Waves: Their Generation and Propagation on the Ocean Surface, Prentice Hall, 1965.
- (Peachey, 1986) D. R. Peachey, Modeling waves and surf, SIGGRAPH '86, volume 20, pp. 65-70, 1986.
- (Reeves, 1983) W.T. Reeves, Particle systems—a technique for modeling a class of fuzzy objects, Proceedings of SIGGRAPH '83, Comput. Graph. Volume 17 (3) pp. 359–376, 1983.
- (Sims, 1990) K. Sims, Particle animation and rendering using data parallel computation, Proceedings of SIGGRAPH '90, Comput. Graph. volume 24 (4), pp. 405–413, 1990.
- (Ts'ó e Barsky, 1987) P. Y. Ts'ó, B. A. Barsky, Modeling and rendering waves: Wave-tracing using beta-splines and reflective and refractive texture mapping, ACM Transactions on Graphics, volume 6 pp. 191-214, (Julho 1987).
- (Tonnesen, 1991) D. Tonnesen, Modeling liquids and solids using thermal particles, Proceedings of Graphics Interface '91, pp. 255–262, 1991.
- (Tessendorf, 1999) J. Tessendorf, Simulating ocean water, SIGGRAPH '99 Course Notes, 1999.
- (Witting, 1999) P. Witting, Computational fluid dynamics in a traditional animation environment, in: Proceedings of SIGGRAPH '99, pp. 129–136, 1999.
- (Wloka, 2002) M. Wloka, Technical report: Fresnel Reflexion, NVIDIA Corporation, (Junho 2002).