

# Simulação de ondas e oceano



**Antoniel Magalhães  
Luis Felipe**



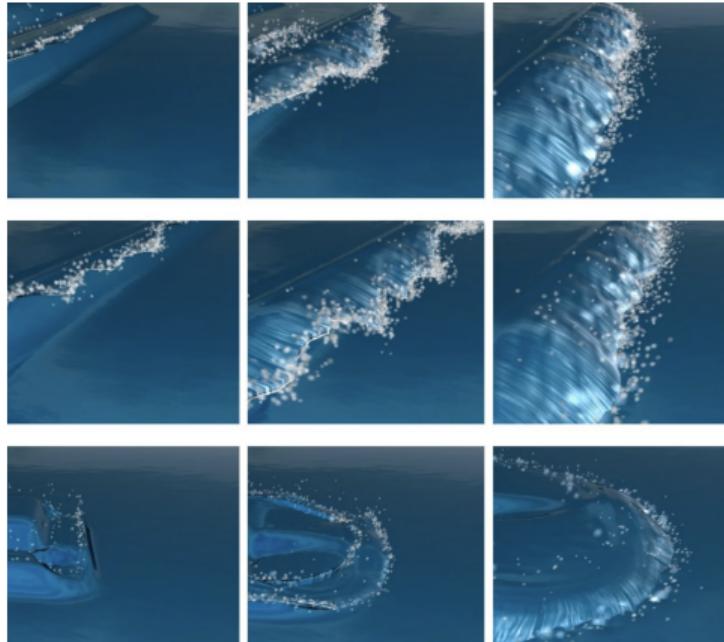
# Agenda

1. Introdução
2. Imagens dos Artigos de Referência
3. Computação Atual de Ondas
4. Técnicas Avançadas de Simulação
5. Referências

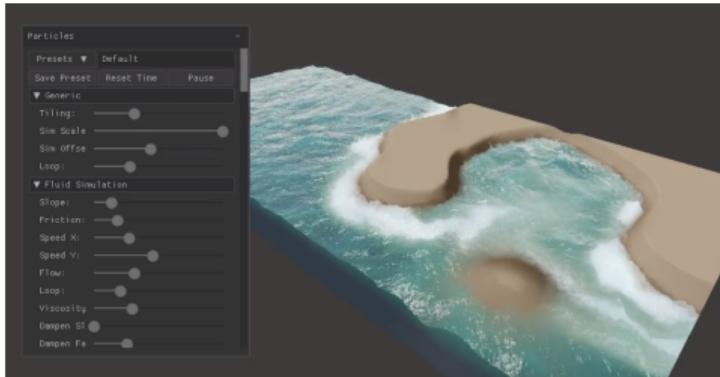
# Introdução

- A simulação de ondas e oceano é uma área de estudo que combina física, matemática e computação para modelar o comportamento das ondas no mar.
- Este campo é crucial para:
  - Jogos e aplicações interativas
  - Efeitos visuais em filmes e animações
  - Visualização científica
  - Simuladores de treinamento marítimo

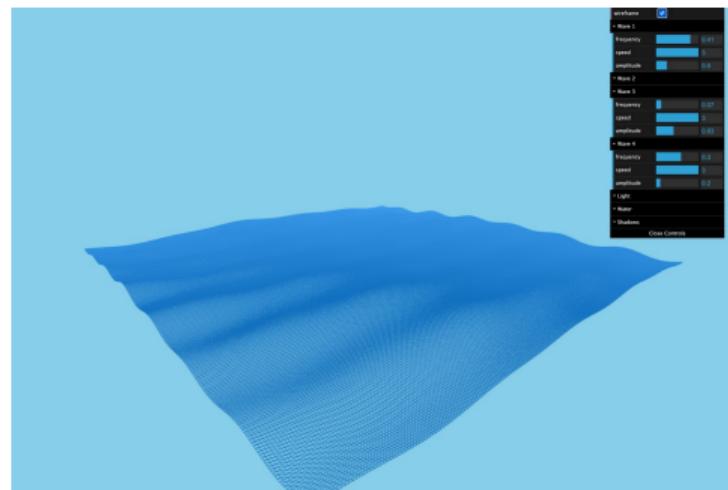
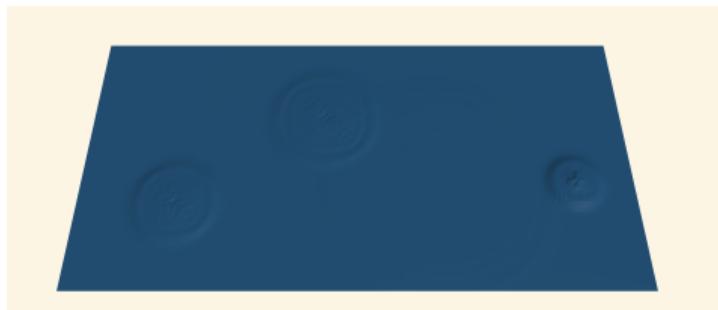
# Imagens dos Artigos de Referência



# Computação Atual de Ondas



# Demonstração da Simulação



# Survey: Estado da Arte em Simulação de Oceanos

- **Simulação de Águas Profundas:**

- Abordagens **Espaciais**: funções periódicas
- Abordagens **Espectrais**: transformada de Fourier a partir de espectros de ondas
- Abordagens **Híbridas**: combinação dos métodos espaciais e espectrais

- **Simulação de Águas Rasas:**

- Métodos **Eulerianos**: baseados nas equações de Navier-Stokes (NSE)
- Métodos **Lagrangianos**: por exemplo, Smoothed Particle Hydrodynamics (SPH)
- Métodos **Híbridos**: integração de abordagens Eulerianas e Lagrangianas

# Terrain and Water Rendering with Hardware Tessellation (2010)

- **O que é Tesselação?**

- Técnica que subdivide a geometria na GPU em tempo real
- Gera níveis de detalhe adaptativos sem sobrecarregar a CPU

- **Aplicações:**

- **Terrenos:** Renderização de superfícies extensas com detalhes variáveis
- **Oceanos:** Simulação de ondas com movimento físico e reflexões Fresnel

- **Relação com o Survey:**

- Complementa métodos de geração de ondas (funções periódicas, Fourier)
- Utiliza tesselação para gerar geometrias detalhadas de ondas

- **Resultados:**

- Melhor desempenho e realismo em cenas complexas

# Função Procedural para Geração de Ondas

- **Objetivo:** Criar uma superfície ondulada realista aplicando deslocamentos a cada vértice.
- **Fórmula do Deslocamento:**

$$s(p, a, \lambda, k, t) = a \begin{bmatrix} -\cos(\phi) \\ \sin(\phi) \end{bmatrix} \quad \text{com} \quad \phi = \frac{2\pi}{\lambda}(p \cdot k + v t)$$

- **Combinação de Ondas:**

$$p_{\Sigma} = p + \sum_{i=0}^n s(p, a_i, \lambda_i, k_i, t)$$

- **Onde:**

- $p$  é a posição original do vértice,  $a$  é a amplitude da onda,  $\lambda$  é o comprimento de onda,  $k$  é a direção da onda (normalizada),  $v$  é a velocidade da onda e  $t$  é o tempo.

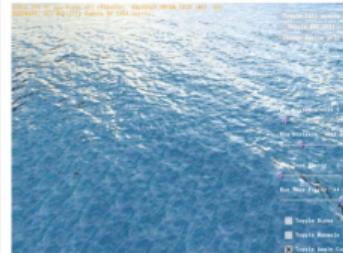
- **Processo:**

- Aplicada a cada vértice da malha subdividida pela tesselação.
- Executada no **Domain Shader**, onde os novos vértices são posicionados.

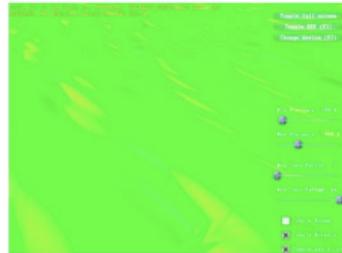
- **Relação com o Survey:**

- Aqui, a tesselação gera dinamicamente os vértices, e a função procedural os transforma em uma geometria ondulada.

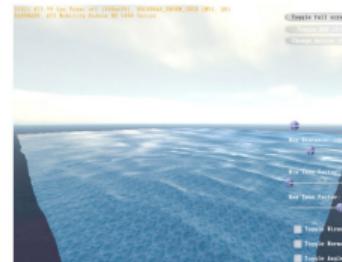
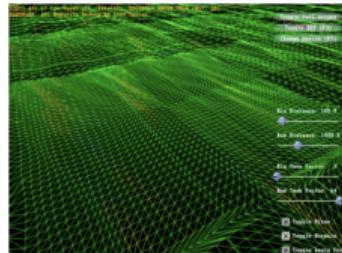
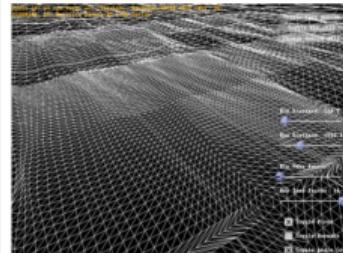
# Demonstração



(a) Ocean



(b) Ocean normals



(a) Ocean normals



(b) Wired ocean normals

# Real-time Breaking Waves for Shallow Water Simulations

- **Objetivo:** Aprimorar simulações de águas rasas com o efeito de ondas que quebram, utilizando um modelo de campo de altura eficiente para aplicações em tempo real.
- **Desafio:** Simulações 3D de fluidos são computacionalmente caras para aplicações interativas.
- **Solução:** Combinação de simulação de campo de altura (2D) com partículas para representar ondas que quebram.

# Geração de Ondas Quebrando

- **Detecção:** Identificação de regiões com frentes de ondas íngremes onde a inclinação da altura do fluido é maior que um limiar e a velocidade do fluido se opõe ao gradiente da altura.
- **Critério de Detecção:**  $|\nabla H(x)| > t_H$  e  $\nabla H(x) \cdot u(x) < 0$ , onde  $t_H = p_H g \Delta t / \Delta x$ .
- **Construção de Linhas de Onda:** Criação de linhas de pontos conectados ao longo da frente da onda.
- As linhas são advectadas usando a velocidade da onda e projetadas ao longo do gradiente de altura.
- **Geração de Patches de Fluidos:** Criação de patches de partículas conectadas ao longo da linha da onda para representar a onda quebrando.
- A velocidade das partículas do patch é calculada com base na velocidade da linha da onda e na energia potencial.

# Demonstração

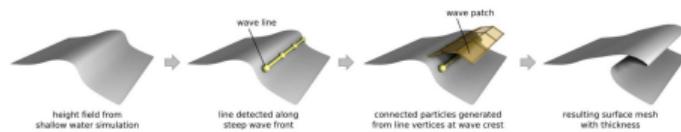


Figure 2. Here an overview of our wave simulation approach can be seen.

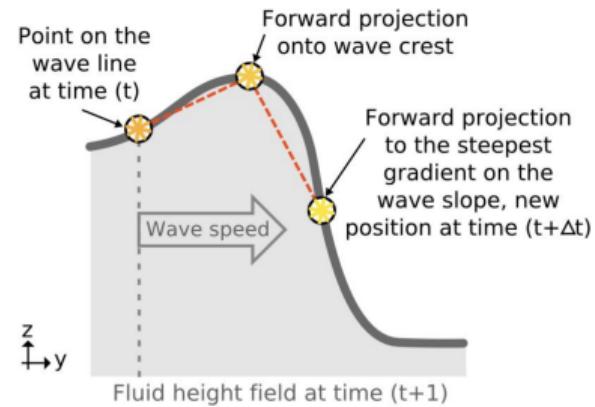


Figure 4. This picture shows a side view of the wave line vertex advection.

# Resultados e Desempenho

- Simulações em tempo real de ondas quebrando com taxas de quadros de 40 a 75 fps em hardware comum:
  - Intel Core 2 Duo 2.13 GHz e Nvidia Geforce 7950 GPU.
- O modelo permite a criação de ondas em diversas configurações e condições.
- A simulação de fluidos representa cerca de 80% do tempo de execução, enquanto a renderização e o overhead do motor gráfico consomem os 20% restantes.
- Aproximadamente metade do tempo de simulação é gasto na simulação de águas rasas, e um quarto é gasto no algoritmo de simulação de ondas.

# Referências

- [1] Xavier Bonaventura. *Terrain and Ocean Rendering with Hardware Tessellation*, pages 83–94. AK Peters/CRC Press, 2018.
- [2] Emmanuelle Darles, Benoît Crespin, Djamchid Ghazanfarpour, and Jean-Christophe Gonzato. A survey of ocean simulation and rendering techniques in computer graphics. *Computer Graphics Forum*, 30(1):43–60, 2011.
- [3] Nils Thürey, Matthias Müller-Fischer, Simon Schirm, and Markus Gross. Real-time breaking waves for shallow water simulations. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)*, pages 39–46, 2007.