Natural Water Simulation

Team Wavy

School of Data and Computer Science(SYSU)

2018 年 7 月 1 日

Outline

Outline

Some main references

- A lecture note of ocean waves simulation of NTU

Some main references

- A lecture note of ocean waves simulation of NTU
- Haibo Ma. Research and application of real-time wave simulation algorithm [D]. Shenzhen University, 2017.

Some main references

- A lecture note of ocean waves simulation of NTU
- Haibo Ma. Research and application of real-time wave simulation algorithm [D]. Shenzhen University, 2017.
- Jeschke S, Wojtan C. Water wave packets[J]. Acm Transactions on Graphics, 2017, 36(4):1-12.

Some main references

- A lecture note of ocean waves simulation of NTU
- Haibo Ma. Research and application of real-time wave simulation algorithm [D]. Shenzhen University, 2017.
- Jeschke S, Wojtan C. Water wave packets[J]. Acm Transactions on Graphics, 2017, 36(4):1-12.
- Fuding Fan. Real time rendering algorithm for virtual ocean multi scene elements [D]. Yanshan University, 2016.

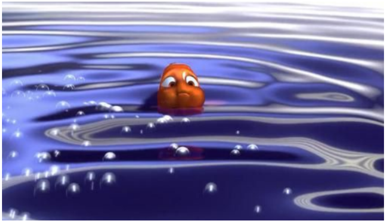
Some main references

- A lecture note of ocean waves simulation of NTU
- Haibo Ma. Research and application of real-time wave simulation algorithm [D]. Shenzhen University, 2017.
- Jeschke S, Wojtan C. Water wave packets[J]. Acm Transactions on Graphics, 2017, 36(4):1-12.
- Fuding Fan. Real time rendering algorithm for virtual ocean multi scene elements [D]. Yanshan University, 2016.
- Stefan J, Tomas S, Matthias M.F, Nuttapong C, Miles M, Chris W. Water Surface Wavelets[J]. ACM Transactions on Graphics, 2017, 37(4):1-13.

Applications

It's just everywhere!

- Ads



The first film

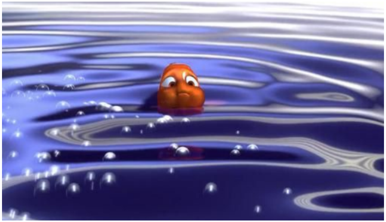
The second film

Finding Nemo(海底总动员)

Applications

It's just everywhere!

- Ads
- Cartoon



The first film

The second film

Finding Nemo(海底总动员)

Applications

- Computer games(Lifelike effect makes us absorbed!)

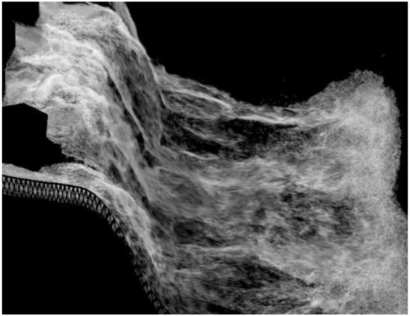
2D games?

3D games

Water simulation in games

Applications

- Research and industrial design



Flood model for dam design

HOW to simulate?

Quite difficult to deal with the **volatile** material...

Three mainstream approaches:

HOW to simulate?

Quite difficult to deal with the **volatile** material...

Three mainstream approaches:

- Texture-based method : minimize the calculation, wildly used in real time rendering

HOW to simulate?

Quite difficult to deal with the **volatile** material...

Three mainstream approaches:

- Texture-based method : minimize the calculation, wildly used in real time rendering
    - Blinn, 1978, Bump Mapping

HOW to simulate?

Quite difficult to deal with the **volatile** material...

Three mainstream approaches:

- Texture-based method : minimize the calculation, wildly used in real time rendering
  - Blinn, 1978, Bump Mapping
- Construction-based method : more mathematicall

HOW to simulate?

Quite difficult to deal with the **volatile** material...

Three mainstream approaches:

- Texture-based method : minimize the calculation, wildly used in real time rendering
  - Blinn, 1978, Bump Mapping
- Construction-based method : more mathematicall
  - Cosine function superposition algorithm

HOW to simulate?

Quite difficult to deal with the **volatile** material...

Three mainstream approaches:

- Texture-based method : minimize the calculation, wildly used in real time rendering
  - Blinn, 1978, Bump Mapping
- Construction-based method : more mathematicall
  - Cosine function superposition algorithm
  - Gerstner Wave

HOW to simulate?

Quite difficult to deal with the **volatile** material...

Three mainstream approaches:

- Texture-based method : minimize the calculation, wildly used in real time rendering
  - Blinn, 1978, Bump Mapping
- Construction-based method : more mathematicall
  - Cosine function superposition algorithm
  - Gerstner Wave
  - B-spline

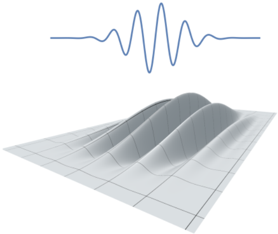
HOW to simulate?

Quite difficult to deal with the **volatile** material...

HOW to simulate?

Quite difficult to deal with the **volatile** material...

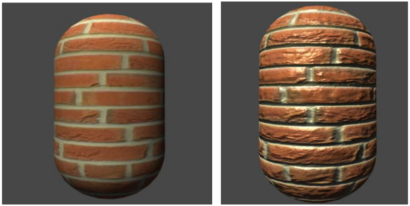
- Based on physics models : Realistic and Lifelike(pick it!✓)



Texture-based method : Bump Mapping

"FAKE" wave!!

Normal texture mapping *does not change the shape of objects*, but produces uneven effects.

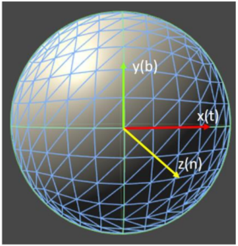


Normal texture mapping vs. Bump mapping

Texture-based method : Bump Mapping

How to store the normal map?

- In model space

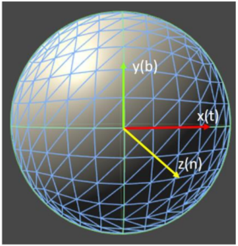


The tangent space of the vertices of a model

Texture-based method : Bump Mapping

How to store the normal map?

- In model space
- In tangent space(切线空间)



The tangent space of the vertices of a model

Texture-based method : Bump Mapping

To apply bump mapping in wave simulation, we should store

- Normal vector of the vertex

Steps:

Texture-based method : Bump Mapping

To apply bump mapping in wave simulation, we should store

- Normal vector of the vertex
- Tangent vector of the vertex

Steps:

Texture-based method : Bump Mapping

To apply bump mapping in wave simulation, we should store

- Normal vector of the vertex
- Tangent vector of the vertex
- Binormal vector of the vetex(副法线向量)

Steps:

Texture-based method : Bump Mapping

To apply bump mapping in wave simulation, we should store

- Normal vector of the vertex
- Tangent vector of the vertex
- Binormal vector of the vetex(副法线向量)

Steps:

1. 在片元着色器中根据UV坐标对法线纹理进行采样，并还原真正的法线信息

Texture-based method : Bump Mapping

To apply bump mapping in wave simulation, we should store

- Normal vector of the vertex
- Tangent vector of the vertex
- Binormal vector of the vetex(副法线向量)

Steps:

1. 在片元着色器中根据UV坐标对法线纹理进行采样，并还原真正的法线信息
2. 利用矩阵M将光源方向以及视线方向从世界空间转换到切线空间，或者利用矩阵M的逆转置矩阵将法线从切线空间转换到世界空间

Texture-based method : Bump Mapping

To apply bump mapping in wave simulation, we should store

- Normal vector of the vertex
- Tangent vector of the vertex
- Binormal vector of the vetex(副法线向量)

Steps:

1. 在片元着色器中根据UV坐标对法线纹理进行采样，并还原真正的法线信息
2. 利用矩阵M将光源方向以及视线方向从世界空间转换到切线空间，或者利用矩阵M的逆转置矩阵将法线从切线空间转换到世界空间
3. 根据光照方程计算该片元的颜色

Texture-based method：Bump Mapping

To apply bump mapping in wave simulation, we should store

- Normal vector of the vertex
- Tangent vector of the vertex
- Binormal vector of the vetex(副法线向量)

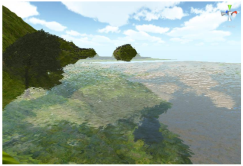
Steps:

1. 在片元着色器中根据UV坐标对法线纹理进行采样，并还原真正的法线信息
2. 利用矩阵M将光源方向以及视线方向从世界空间转换到切线空间，或者利用矩阵M的逆转置矩阵将法线从切线空间转换到世界空间
3. 根据光照方程计算该片元的颜色
4. 更新每个点的UV坐标，回到步骤1

Texture-based method : Bump Mapping

The core issue of the algorithm: Constantly update the UV coordinates, then map the texture and calculate the light, so that the water appears to be fluctuating.

- Just **deceive** your eyes!

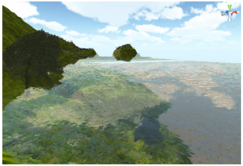


Applying bump mapping in water simulation

Texture-based method : Bump Mapping

The core issue of the algorithm: Constantly update the UV coordinates, then map the texture and calculate the light, so that the water appears to be fluctuating.

- Just **deceive** your eyes!
- **Minumize** the calculation, effective in many cases.



Applying bump mapping in water simulation

Construction-based method

- Cosine function superposition algorithm(余弦函数叠加法)

Construction-based method

- Cosine function superposition algorithm(余弦函数叠加法)
- Gerstner wave

Construction-based method

- Cosine function superposition algorithm(余弦函数叠加法)
- Gerstner wave
- Fast Fourier transform(FFT)

Construction-based method

- Cosine function superposition algorithm(余弦函数叠加法)
- Gerstner wave
- Fast Fourier transform(FFT)
- ...

Cosine function superposition algorithm(余弦函数叠加法)

Two main factors that influence the calculation of Fourier transform of the wave

- Gravity : g

And if the wavelength is long enough, the gravity plays the major role in the calculation.

Cosine function superposition algorithm(余弦函数叠加法)

Two main factors that influence the calculation of Fourier transform of the wave

- Gravity : g
- Surface tension(表面张力) : λ

And if the wavelength is long enough, the gravity plays the major role in the calculation.

Cosine function superposition algorithm(余弦函数叠加法)

Using a nonlinear partial differential equation(NPDE) to describe the movement of the water surface, then we can got a approximate solution of the function

$$\mathrm{c} = \sqrt{\frac{g\lambda}{2\pi}} = \sqrt{\frac{g}{u}}$$

where $u = \frac{2\pi}{\lambda}$ is the number of the waves, then we can describe the height of water surface as follow

$$\mathrm{z}(x, t) = \mathrm{a}\cos u(x - ct) = \mathrm{a}\cos(ux - \omega t)$$

Cosine function superposition algorithm(余弦函数叠加法)

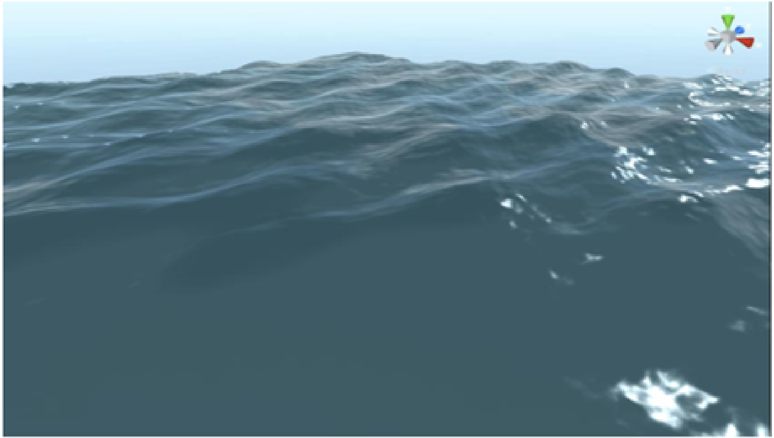
a is the amplitude(振幅), $/omega = uc$ is the angular velocity(角速度), in the 2-dimensional coordinate system

$$z(x, y, t) = -h + a\cos(lx + my - \omega t)$$

h is the sea level height, since different waves that are independent

$$f(x, y, t) = -h + \sum_{i=1}^{n} a_i \cos(l_i x + m_i - \omega_i t)$$

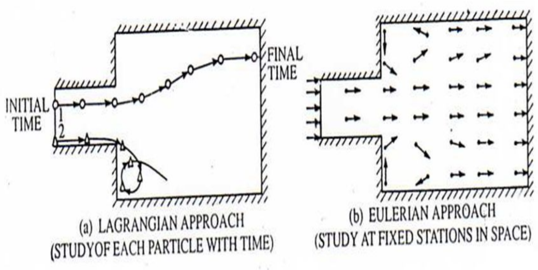
Cosine function superposition algorithm(余弦函数叠加法)



Based on physics models

Algorithms based on physics models can be divided into two categories

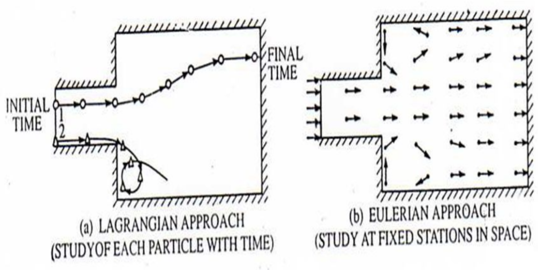
- Eulerian approache(grid-based)



Based on physics models

Algorithms based on physics models can be divided into two categories

- Eulerian approache(grid-based)
- Lagrangian approache(based on particle system)



Navier-Stokes(N-S) equation

$$\underbrace{\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right)}_{1} = \underbrace{-\nabla p}_{2} + \underbrace{\nabla\cdot\left(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - \frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)}_{3} + \underbrace{\mathbf{F}}_{4}$$

Drawbacks

- Usually has no analytic solution

Navier-Stokes(N-S) equation

$$\underbrace{\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)}_{1}=\underbrace{-\nabla p}_{2}+\underbrace{\nabla\cdot\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)}_{3}+\underbrace{\mathbf{F}}_{4}$$

Drawbacks

- Usually has no analytic solution
- Exponential time complexity !

Navier-Stokes(N-S) equation

To optimize...

- Make restriction to the conditions

Navier-Stokes(N-S) equation

To optimize...

- Make restriction to the conditions
  - Hypothesis of small amplitude

Navier-Stokes(N-S) equation

To optimize...

- Make restriction to the conditions
  - Hypothesis of small amplitude
  - Hypothesis of shallow water[Kass and Miller 1990]

Navier-Stokes(N-S) equation

To optimize...

- Make restriction to the conditions
  - Hypothesis of small amplitude
  - Hypothesis of shallow water[Kass and Miller 1990]
  - Hypothesis of infinite depth of water[Mastin et al. 1987; Tessendorf 2004b]

Navier-Stokes(N-S) equation

To optimize...

- Make restriction to the conditions
  - Hypothesis of small amplitude
  - Hypothesis of shallow water[Kass and Miller 1990]
  - Hypothesis of infinite depth of water[Mastin et al. 1987; Tessendorf 2004b]
  - Ignore the solid boundary[Fournier and Reeves 1986]

Navier-Stokes(N-S) equation

To optimize...

- Make restriction to the conditions
  - Hypothesis of small amplitude
  - Hypothesis of shallow water[Kass and Miller 1990]
  - Hypothesis of infinite depth of water[Mastin et al. 1987; Tessendorf 2004b]
  - Ignore the solid boundary[Fournier and Reeves 1986]
- Expore better numerical calculation Methods

Navier-Stokes(N-S) equation

To optimize...

- Make restriction to the conditions
  - Hypothesis of small amplitude
  - Hypothesis of shallow water[Kass and Miller 1990]
  - Hypothesis of infinite depth of water[Mastin et al. 1987; Tessendorf 2004b]
  - Ignore the solid boundary[Fournier and Reeves 1986]
- Expore better numerical calculation Methods
  - (Canabal et al. 2016)

Navier-Stokes(N-S) equation

To optimize...

- Make restriction to the conditions
  - Hypothesis of small amplitude
  - Hypothesis of shallow water[Kass and Miller 1990]
  - Hypothesis of infinite depth of water[Mastin et al. 1987; Tessendorf 2004b]
  - Ignore the solid boundary[Fournier and Reeves 1986]
- Expore better numerical calculation Methods
  - (Canabal et al. 2016)
  - (Tessendorf 2004a)

Navier-Stokes(N-S) equation

To optimize...

- Make restriction to the conditions
  - Hypothesis of small amplitude
  - Hypothesis of shallow water[Kass and Miller 1990]
  - Hypothesis of infinite depth of water[Mastin et al. 1987; Tessendorf 2004b]
  - Ignore the solid boundary[Fournier and Reeves 1986]
- Expore better numerical calculation Methods
  - (Canabal et al. 2016)
  - (Tessendorf 2004a)
- More creative...

Navier-Stokes(N-S) equation

To optimize...

- Make restriction to the conditions
  - Hypothesis of small amplitude
  - Hypothesis of shallow water[Kass and Miller 1990]
  - Hypothesis of infinite depth of water[Mastin et al. 1987; Tessendorf 2004b]
  - Ignore the solid boundary[Fournier and Reeves 1986]
- Expore better numerical calculation Methods
  - (Canabal et al. 2016)
  - (Tessendorf 2004a)
- More creative...
  - Take the single wave as a whole research object in Lagrangian approache (instead of a single wave particle) [Yuksel et al 2007]

Outline

Overview of the article

- Using the method 1 and 3 mentioned above

Overview of the article

- Using the method 1 and 3 mentioned above
- Some novel work

Overview of the article

- Using the method 1 and 3 mentioned above
- Some novel work
  - Introducing the concept of wave packet

Overview of the article

- Using the method 1 and 3 mentioned above
- Some novel work
  - Introducing the concept of wave packet
  - Consider 5 visual verisimilitudes(视觉效果), including dispersion, refraction, reflection, diffraction and dissipation of water waves

Overview of the article

- Using the method 1 and 3 mentioned above
- Some novel work
  - Introducing the concept of wave packet
  - Consider 5 visual verisimilitudes(视觉效果), including dispersion, refraction, reflection, diffraction and dissipation of water waves
  - Parallelism and optimization

Overview of the article

- Using the method 1 and 3 mentioned above
- Some novel work
  - Introducing the concept of wave packet
  - Consider 5 visual verisimilitudes(视觉效果), including dispersion, refraction, reflection, diffraction and dissipation of water waves
  - Parallelism and optimization
  - Providing methods of adjusting parameters

Some preparations!

phase velocity(相速度) c_p

$3x10^9 m/s$, differentiated with 'group' velocity!

Some preparations!

phase velocity(相速度) c_p

$3x10^9 m/s$, differentiated with 'group' velocity!

wavenumber(波数) k

$k = \frac{2\pi}{\lambda} = \frac{2\pi f}{v}$, a descriptor of wave properties.

Some preparations!

wave packets(波包)

一个石块扔进池塘引起的波纹不是一个波而是一组波，我们把这组波称为波包(Wave packets)，其中每个波称为子波.

Some preparations!

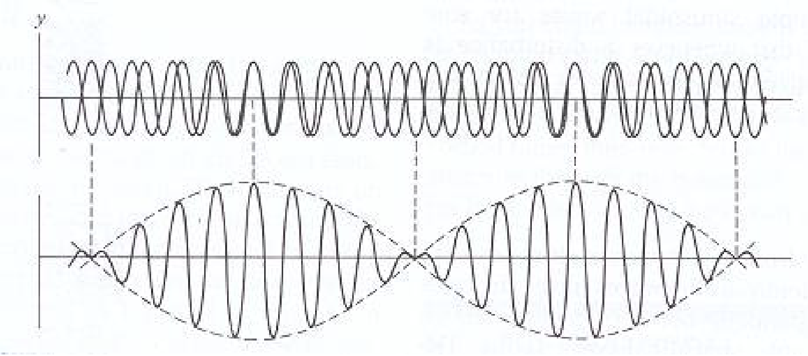
wave packets(波包)

一个石块扔进池塘引起的波纹不是一个波而是一组波，我们把这组波称为波包(Wave packets)，其中每个波称为子波.

group velocity(群速度)

用函数图像显示出来，下图中上面一行的就是两个子波，下面的则是他们形成的波包。整个波包移动的速度就是群速度.

Some preparations!



Some preparations!

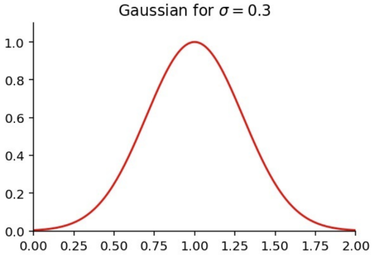
群速度的定义式如下，它等于某个子波的相速度加上k乘以相速度对k的偏导

$$c_g = \frac{d\omega}{dk} = \frac{dkv}{dk} = c_p + k\frac{\partial c_p}{\partial k}$$

A personal analysis of this paper

How can we demonstrate **wave packts**?

- Using Gauss Kernel Distribution(GKD, 高斯核分布) e^{-x^2} to represent a wave packet. x is the space coordinates while on the surface of the water



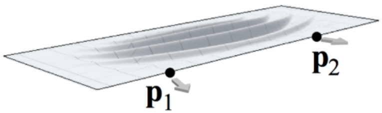
GKD function

A personal analysis of this paper

Represent the wave packets...

- Amplitude(振幅) : a_j

水面上，一个波包初始表示为一个前进方向长为3倍代表波长(Wavelength)，正切方向长为6倍波长的矩形切片.



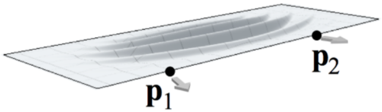
Represent a wave packet

A personal analysis of this paper

Represent the wave packets...

- Amplitude(振幅) : a_j
- Wavenumber(波数) : k_j

水面上，一个波包初始表示为一个前进方向长为3倍代表波长(Wavelength)，正切方向长为6倍波长的矩形切片.



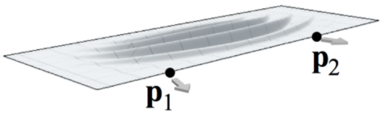
Represent a wave packet

A personal analysis of this paper

Represent the wave packets...

- Amplitude(振幅) : a_j
- Wavenumber(波数) : k_j
- Wavelength(波长) : λ_j

水面上，一个波包初始表示为一个前进方向长为3倍代表波长(Wavelength)，正切方向长为6倍波长的矩形切片.



Represent a wave packet

A personal analysis of this paper

波包运动过程中，两个点控制点的位置使用欧拉前项法迭代更新，下一时刻的位置等于前一时刻位置乘以波包的群速度

$$\mathbf{p}(t+\Delta t) := \mathbf{p}(t) + \Delta t \mathbf{c}_g(\mathbf{p}(t), k_j)$$

A personal analysis of this paper

积分形式：积分变量是波数k，被积函数第一项$a(k)$代表一个波数为k的子波的振幅。直观上可以理解：点x处的水面高度是所有子波的振幅加权求和.

$$\eta(x,t) = \int_{-\infty}^{\infty} a(k) \cos\left(kx - \omega(k,h)t\right) dk.$$

A personal analysis of this paper

最后，假设在时刻t，水面点x重叠了N个波包，那么该点的高度就是这N个波包分别使用Airy波理论算出的水面高度之和. 公式如下: 通过该公式可以方便地计算出水面上任意点的高度，从而绘制出水面.

$$\eta(\hat{\mathbf{x}}, t) \approx \sum_{j=1}^{N} a_j \phi(\hat{\mathbf{x}}) \cos\left(\mathbf{k}_j \cdot (\hat{\mathbf{x}} - \mathbf{X}_{pg})\right),$$

More clearly?

A personal analysis of this paper

How about dispersion(色散), refraction(折射), reflection(反射), diffraction(衍射) and energy loss(能量损失) of water waves?

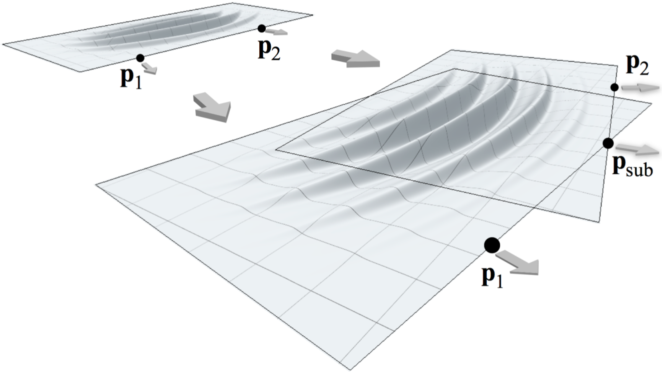
dispersion(色散)

色散是指不同波长的波作为一个波包在传播中分散的现象.

论文中使用两个控制点表示波包方向的做法自然地给出了实现这一现象的方法: 由于控制点P1P2与群速度不共线，因此当P1和P2不断更新迭代后其距离会不断拉大，达到3倍代表波长时就从几何中点P_{sub}处会分离，它们的几何中点P_{sub}，分别与P_1，P_2连成两个新的波包的前进方向的边.

Interesting!

A personal analysis of this paper

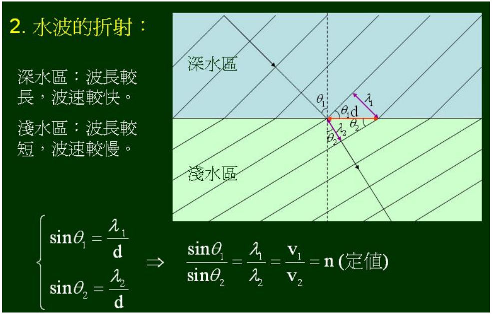


波包“分蘖”

A personal analysis of this paper

refraction(折射)

折射是由于在水面波通过不同深度的水域时，其群速度会发生改变，由惠更斯定理(Huygens theorem)可算出折射后的速度方向.



A personal analysis of this paper

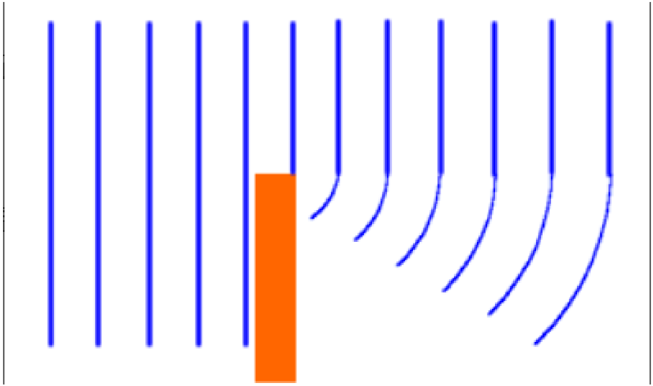
The paper just used the normal reflection theorem to simulate reflection, quite reasonable.

diffraction(衍射)

波的衍射是指波遇到障碍物后偏离原来的运动方向的现象.

In this paper, the method of diffraction is to show the effect of "sticking" on obstacles(粘附障碍物) after wave meets obstacles.

A personal analysis of this paper



A personal analysis of this paper

Let's focus on the energy loss now!
由于水的黏着性等因素，水面波会发生能量耗散，并使得振幅随时间降低，最终降为0。这一过程可以用下面这个对振幅的迭代方程表示：

$$a(t+\Delta t) = a(t)\exp(-(2\nu k^2 + \nu k\,(\omega(k)/2\nu)^{\frac{1}{2}}\,/2)\,\alpha\Delta t),$$

其中α被称为阻尼系数，α越大，水波耗散地越快

A personal analysis of this paper

Something other things： How do authors accelerate the calculation and make optimization?

- 程序GPU加速，并行地计算出水面网格上各点的高度

A personal analysis of this paper

Something other things： How do authors accelerate the calculation and make optimization?

- 程序GPU加速，并行地计算出水面网格上各点的高度
- 波包设置上限，通过动态调整阻尼系数使播报数量稳定在上限附近

A personal analysis of this paper

Last but not least...
How to generate different form of water waves by adjusting the speed of wave sources?
波源指引发波的物体，例如一个移动中的小船，它某一时刻的运动引发的波的波数可以由开尔文定理给出：公式中v是这一时刻小船的速度，θ是波包群速度方向和小船运动方向的夹角.

$$k = \frac{g}{||\mathbf{v}||^2 \cos^2 \theta},$$

A personal analysis of this paper

For example, adjust v, we can simulate the different effect of a swimming duck and a working ship.

Outline

Demo

Now is the demo time!

Group member

李新锐 15323032	孙一言 16337216
王锦鹏 16337232	王永锋 16337237
颜彬 16337269	韦博耀 16337242

Thank you

Thank you for listening!