

Rapport TPs Modélisation et Programmation C++

Baudet Gladys & Romero Quentin

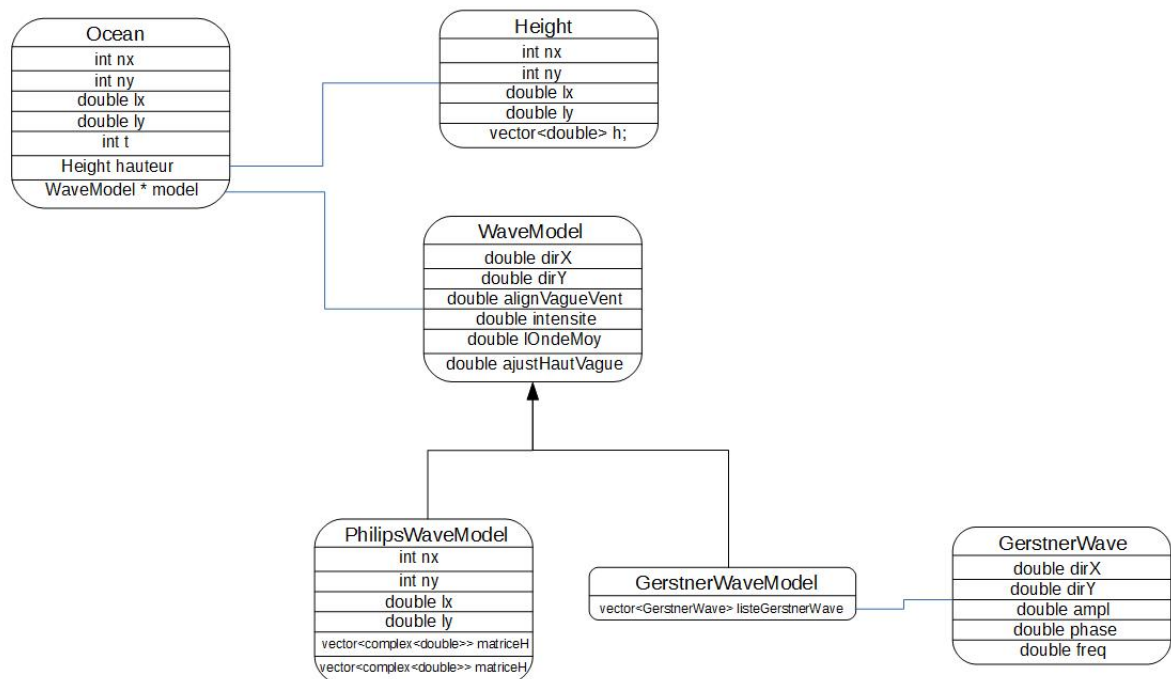
May 7, 2017

1 Choix d'implémentation

Par manque de temps, au lieu d'utiliser des templates dans Dvector pour pouvoir utiliser des complexes et des doubles, nous avons utilisé la STL avec le type `vector< complex<double> >`.

La documentation est faite sous Doxygen.

Le diagramme suivant décrit les différentes classes ainsi que leurs relations :



2 Modèle de Gerstner

Le modèle de Gerstner propose une modélisation d'océan avec plusieurs ondes de Gerstner superposées. Une telle onde est créée avec différents paramètres, dont les effets sur le rendu sont les suivants :

- `dirX`, `dirY` : direction du vent ; lorsque ces paramètres augmentent, les vagues sont plus rapprochées et leur vitesse de propagation augmente.
- `alignVagueVent` : Lorsque ce paramètre diminue ($\ll 1$), les vagues se déplacent toutes selon la même direction, de façon plus coordonnée. Lorsqu'il augmente ($\gg 1$), les différentes vagues de Gerstner n'ont plus de coordination.

- intensité : on obtient une mer plate lorsque ce paramètre est petit, et des ondes d'amplitude plus grande lorsqu'il augmente.
- ajustHauteurVague : plus ce paramètre est grand, plus les vagues sont hautes.

3 Modèle de Phillips

3.1 FFT et iFFT

Description des choix, défauts de cache

Nous avons tout d'abord implémenté une version 1D de la FFT, que l'on utilise ensuite pour calculer la FFT2D. Pour cela, on calcule la FFT1D sur les lignes de la matrice, puis on transpose la matrice résultat pour ré-appliquer la FFT1D sur les lignes du tableau. On récupère alors la FFT du tableau.

Pour ne considérer que des sous-vecteurs du vecteur considéré comme un tableau, on prend en paramètres un indice de début et un indice de fin du sous-tableau sur lequel appliquer la transformation. On fait de même pour l'IFFT, en passant par une version 1D et en l'appliquant au cas 2D.

La façon dont est écrite la transposition ne fonctionne que pour des matrices carrées. De plus, l'implémentation de la FFT ne fonctionne que pour des matrices de côté une puissance de 2.

Cette implémentation est peu performante : lors de son utilisation, nous ne pouvons gérer un tableau plus grand que 16x16, et encore, on est à 3 FPS maximum. Au delà, à 32x32, le calcul prend trop de temps. Ce nombre réduit de points ne nous permet pas de voir si le rendu ressemble à une mer, ni l'effet des différents paramètres.

3.2 Implémentation

Séparation des calculs : matrices de base et calculs pour tout t

Dès la création d'une instance de la classe PhilipsWaveModel, on appelle une fonction init, qui calcule les facteurs de Philips ainsi que les fréquences ω pour tous les k, que l'on stocke dans deux matrices différentes. A chaque calcul de \tilde{h} , pour chaque x,y,t, on réutilise les valeurs contenues dans les matrices connues.