

Projet C++ - Simulation numérique de surfaces d'océans

Naoto Lucas, Gaeng Jean-Baptiste

May 22, 2018

1 Présentation des résultats obtenus et de l'implémentation du projet

1.1 Compilation du projet, documentation et tests

Le projet se compile avec la ligne de commande suivante à la racine du projet.

```
make
```

N.B : Pour lancer une simulation de Gerstner ou de Philips, il est nécessaire de décommenter au préalable la ligne "define Modele-Gerstner" ou "define Modele-Philips" du fichier main.cxx.

La documentation est générée à l'aide de :

```
make doc
```

Le script de test du projet est lancé avec la commande :

```
make test
```

Nous avons utilisé la bibliothèque Catch2 afin d'écrire les tests, en raison de sa facilité d'utilisation (un seul .h à importer).

1.2 Résultats obtenus

Voici les résultats graphiques que nous obtenons pour les simulations des ondes de Gerstner et de Philips :

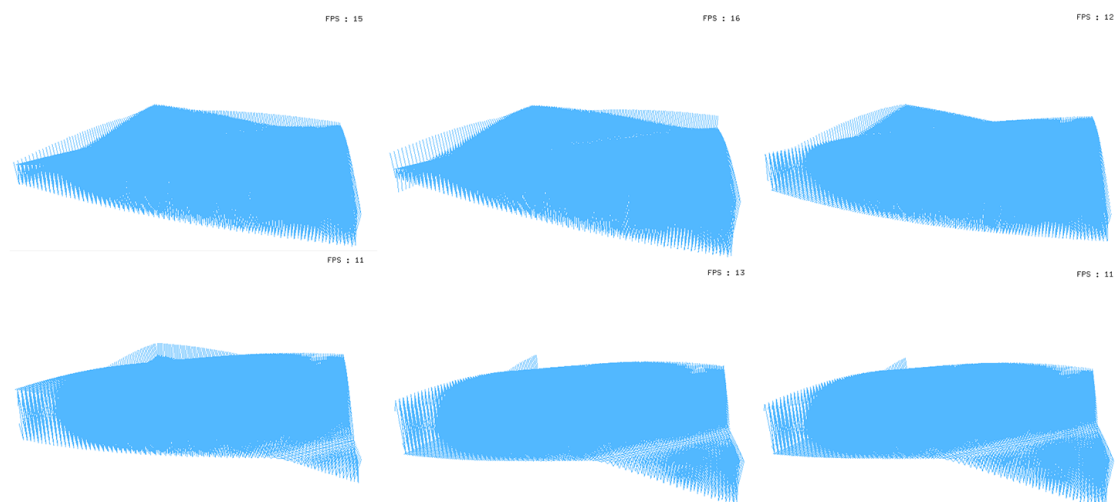


Figure 1: Simulation d'une grande vague suivant le modèle de Gerstner

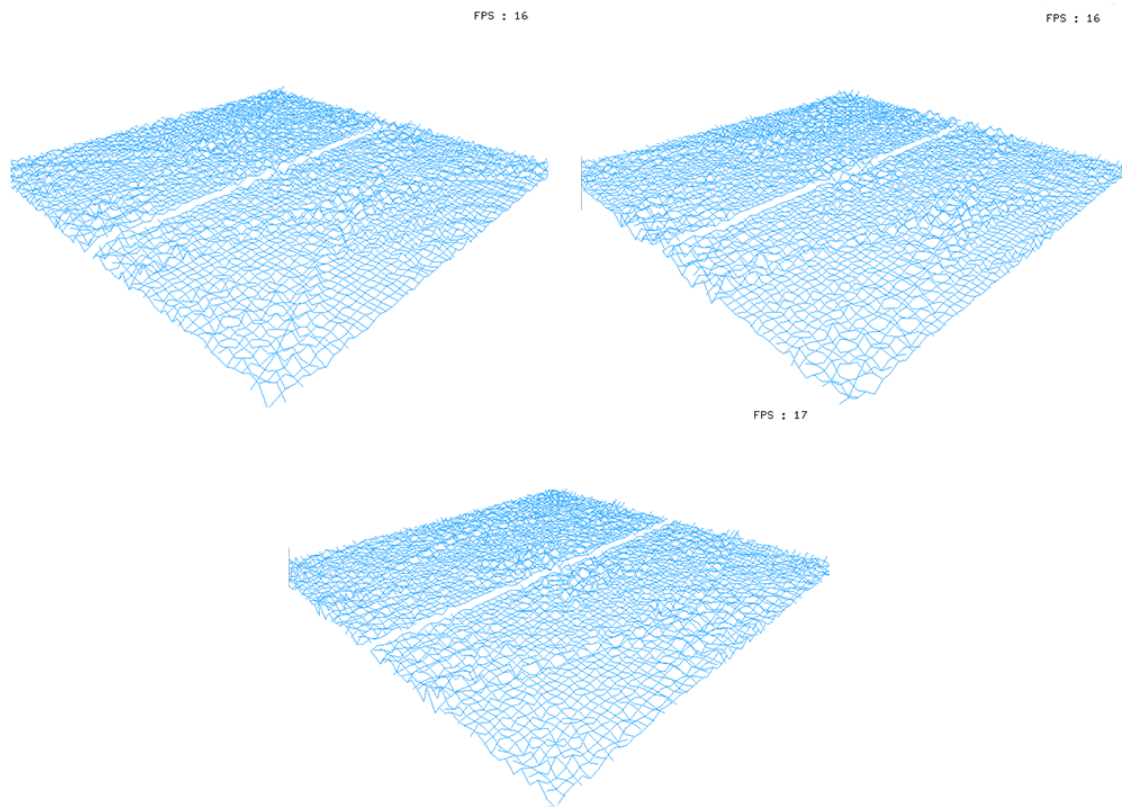


Figure 2: Simulation d'une onde selon le modèle de Philips

Toutes les fonctions liées au modèle de Philips ont été implémentées, mais comme on peut l'observer il reste encore à ajuster les paramètres pour obtenir une onde à l'allure satisfaisante. Il est aussi possible que nous ayons fait une erreur dans l'une des formules du modèle ce qui pourrait expliquer l'allure générale de l'onde, même si nous voyons qu'il s'agit bien d'une onde se déplaçant au fil du temps.

Pour calculer la hauteur de la vague avec le modèle de Philips, il fallait pouvoir accéder à L_x et L_y , et posséder une table de classe `Height` pour stocker le résultat de la FFT inverse avant l'appel de l'accessor `(x, y, t)`. En effet, il fallait garder le polymorphisme mais la hauteur dans le modèle de Philips se calculait d'un coup, nous le calculons donc pour chaque temps une fois, et l'accessor accède à une valeur de hauteur précalculée.