

Contrat Été 2024

# RAPPORT HEBDOMADAIRE

RÉALISÉ DANS LE CADRE  
D'UN PROJET POUR

ISMER-UQAR

25/10/2024

Rédaction  
Charles-Édouard Lizotte  
[charles-edouard.lizotte@uqar.ca](mailto:charles-edouard.lizotte@uqar.ca)  
ISMER-UQAR  
Police d'écriture : **CMU Serif Roman**

# Table des matières

- 1 Point frontières de Wavewatch 2
  - 1.1 Ce qui a été fait jusqu'à maintenant . . . . . 2
  - 1.2 Module Wavespectra . . . . . 2
    - 1.2.1 Spectre de Jonswap . . . . . 2
    - 1.2.2 Spreading (Étalement dans les fréquences) . . . . . 3

## 1. Point frontières de Wavewatch

### 1.1. Ce qui a été fait jusqu'à maintenant

Grossièrement, la configuration est terminée, donc on a des forçage, une carte, etc. Tout ce qu'il reste à faire c'est de créer un spectre permanent qui sera toujours actif aux points frontière. Le gros problème auquel j'ai du faire face la semaine dernière a été de s'assurer que Wavewatch puisse prendre les données entrantes aux points frontières. Donc, cette tâche se déclinait en deux parties :

- **Créer un spectre de vagues sous forme de fichiers NetCDF.** Ça ne m'a pas pris beaucoup de temps, même s'il y a eu ajustements à faire pour s'ajuster à des **conventions** sur les données sortantes et entrantes dans les modèles de vagues.
- **S'assurer que Wavewatch puisse s'accaparer ces données dans son fonctionnement.** Malheureusement, ça ne marche toujours pas, car il semble y avoir des problèmes de compatibilité entre le langage Fortran et Python.

Donc, nous en sommes là. J'ai communiqué avec Jeremy dans le but d'obtenir des fichiers NetCDF entrantes pour comprendre d'où venait mon erreur. J'attends une réponse de sa part.

Entre temps, il existe un module Python qui permet de créer et lire des spectres provenant des plus gros modèles de vagues (SWAN, WW3, etc.) nommé **wavespectra**. Il permet aussi de créer des fonctions de *spreading*, des Jonswap et des Pierson-Moskowitz. J'étais réticent à l'utiliser la semaine dernière, car je croyais être capable de solver le problème de compatibilité, mais je vais mettre mon énergie ailleurs.

### 1.2. Module Wavespectra

#### 1.2.1 Spectre de Jonswap

Eliot avait ça :

TABLE 1 – Tableau tiré de la maîtrise d'Éliot Bimuth.

Description de la variable	Symbole	Valeur	Unités
Taille de la grille	$L_x$	5	km
Taille des points de grille	$\Delta x$	500	m
Nombre de points de grille	$n_x$	10	–
Épaisseur des floes	$h$	0.5	m
Diamètre moyen des floes	$\langle D \rangle$	200	m
Période du maximum spectral	$T_p$	6	s
Fréquence du maximum spectral	$f_p$	1/6	s <sup>-1</sup>
Hauteur significative des vagues	$H_s$	1	m
Minimum de fréquence du modèle	$f_{min}$	1/20	s <sup>-1</sup>
Maximum de fréquence du modèle	$f_{max}$	1/2.5	s <sup>-1</sup>
Nombre de fréquences du modèle	$n_f$	61	–

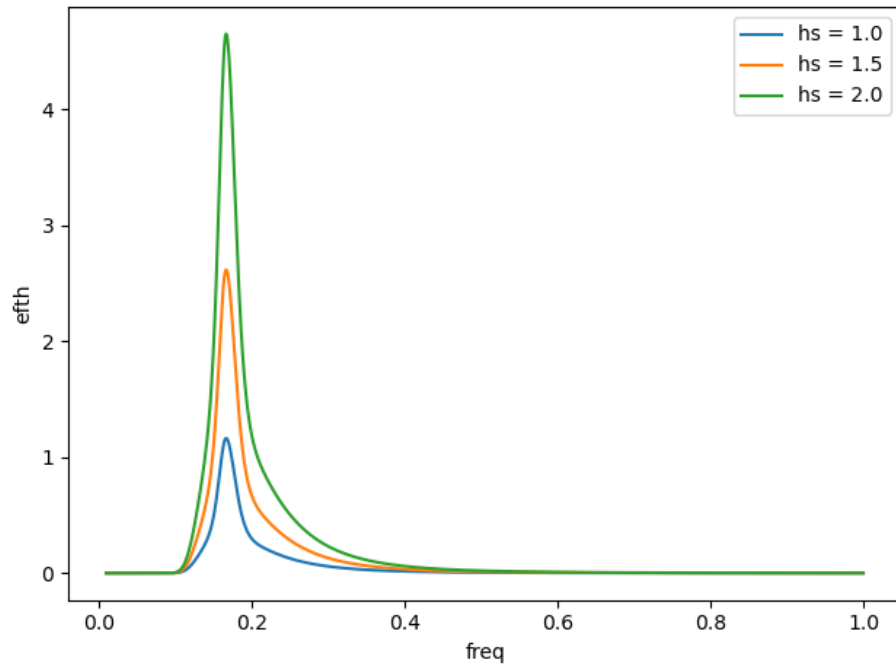
Bien que Wavewatch propose lui-même une fonction Jonswap comme condition initiale, il ne permet pas vraiment de l'appliquer aux points de conditions frontières. Par contre, le module Python **wavespectra** le permet. Le module **wavespectra** prend l'équation de Hasselmann, donc

$$E(f) = \alpha g^2 (2\pi)^{-4} f^{-5} \exp\left(\frac{5}{4} \left(\frac{f}{f_p}\right)^{-4}\right) \cdot \gamma^{[\dots]} \quad (1.1)$$

avec les classique de **hasselmann1973measurements**, soit

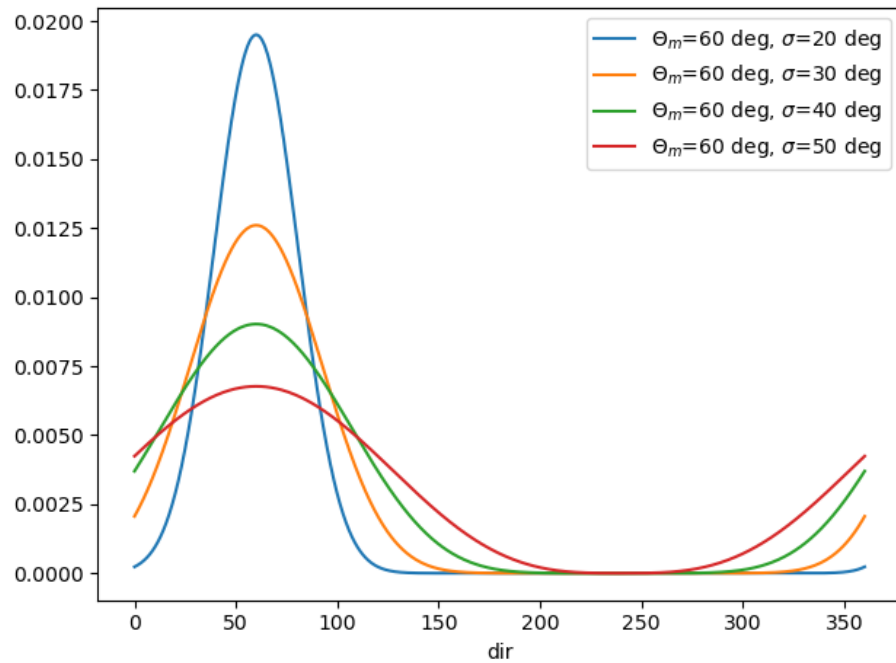
$$\alpha = 0.0081 \quad \sigma_A = 0.07 \quad \sigma_B = 0.09 \quad \gamma = 3.3 \quad (1.2)$$

Le module mentionne que  $4\sqrt{m_0} = h_s$ , donc qu'il y ait un rapport avec la hauteur significative des vagues. Ça donne ceci, donc ça marche :



### 1.2.2 Spreading (Étalement dans les fréquences)

Avec la fonction tirée de l'article de Cartwright (1963), on obtient un étalement directionnel de la forme (voir le [module en ligne](#)) :



Pour aucune raison, nous prendre un  $\sigma$  de 30 degrés. Grossièrement, la fonction d'étalement dans les fréquences

devrait avoir la forme :

$$G(\theta, f) = F(s) \cos^2 \frac{1}{2}(\theta - \theta_m), \quad (1.3)$$

et on a un spectre résultant :

