

Contrat Été 2023

# CARNET DE BORD, UNIVERSITÉ MCGILL

RÉALISÉ DANS LE CADRE  
D'UN PROJET POUR

ISMER–UQAR

22/09/2023

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   | <b>TODO</b> Téléchargement de Wavewatch III – <i>&lt;2023-09-18 Mon&gt;</i>                            | 2 |
| 1.1 | Télécharger et construire le modèle . . . . .  | 2 |
| 1.2 | Débuggage NetCDF et OpenMPI . . . . .  | 3 |
| 2   | <b>DONE</b> Faire fonctionner le modèle standard – <i>&lt;2023-09-19 Tue&gt;</i>                       | 3 |
| 2.1 | Re-Construire la grille, la mapsta et les forçages – <i>&lt;2023-09-19 Tue&gt;</i> . . . . .           | 3 |
| 3   | <b>DONE</b> Faire fonctionner le modèle en MPI – <i>&lt;2023-09-21 Thu&gt;</i>                         | 4 |
| 3.1 | Déterminer les bonnes switches pour la compilation MPI – <i>&lt;2023-09-20 Wed&gt;</i> . . . . .       | 4 |
| 3.2 | Compiler le modèle en MPI – <i>&lt;2023-09-21 Thu&gt;</i> . . . . .                                    | 4 |
| 3.3 | Installer les compilateurs ifort et mpiifort – <i>&lt;2023-09-21 Thu&gt;</i> . . . . .                 | 4 |
| 4   | <b>DONE</b> Codage et test du couplage MPI – <i>&lt;2023-09-21 Thu&gt;</i>                             | 5 |
| 4.1 | Reconstruire la fonction de couplage du modèle shallow water – <i>&lt;2023-09-21 Thu&gt;</i> . . . . . | 5 |
| 4.2 | Remettre à jour les sous-routines de Wavewatch III – <i>&lt;2023-09-22 Fri&gt;</i> . . . . .           | 5 |
| 5   | <b>DONE</b> Rencontre avec Louis-Philippe et David – <i>&lt;2023-09-22 Fri&gt;</i>                     | 6 |
| 5.1 | Création d’un « set-up » de référence du modèle « shallow water » . . . . .                            | 6 |
| 5.2 | Retour sur la stratification . . . . .   | 6 |
| 5.3 | Type de grille du modèle Wavewatch III . . . . .   | 6 |
| 5.4 | Varia et compute Canada . . . . .  | 6 |

# 1 TODO Téléchargement de Wavewatch III – <2023-09-18 Mon>

## 1.1 Télécharger et construire le modèle

Avant tout, un courriel m'avait été envoyé avec un lien pour télécharger le modèle à l'adresse <https://polar.ncep.noaa.gov/waves/wavewatch/distribution/> en 2019. Une fois téléchargée, on peut développer l'archive **tar** dans un nouveau dossier (*ww3* dans notre cas),

```
mkdir ww3
tar -xvf wwatch3.v5.16.tar.gz -C ww3
cd ww3
```

On se retrouve avec une multitude de fichiers **tar** éparpillés au long du dossier *ww3*. Pour développer les multiples archives **tar** restantes, on utilise le script bash d'installation fourni par l'équipe de Wavewatch,

```
chmod +x install_ww3_tar
./install_ww3_tar
```

Tout au long de l'installation, quelques questions seront posées sur les compilateurs désirés. Personnellement, j'utilise les compilateurs *gcc* et *gfortran* par habitude. Par la suite, on s'assure que tout est en ordre à l'aide du script *w3-setup* cachée dans le dossier *ww3/bin*. Le script *w3-setup* crée l'environnement Wavewatch contenue dans le dossier courant sous le nom de *wwatch3.env*. De plus, cette commande nous permet de choisir

1. le répertoire du dossier *ww3* (*home/charlesedouard/Desktop/Travail/ww3*);
2. le fichier de compilation associé à notre compilateur (*ww3/bin/comp.gfortran*);
3. ainsi que les *switches* nécessaires à la compilation du modèle (*ww3/bin/switch-lizotte*).

Dans mon cas, j'ai mes propres *switches*, c'est pourquoi – dans la prochaine commande – on cite *-s lizotte* en référence au fichier *switch-lizotte*. Plus d'informations là dessus à venir. On peut effectuer cette commande depuis le dossier *ww3*,

```
./bin/w3_setup /home/charlesedouard/Desktop/Travail/ww3 -s lizotte -c gfortran
```

Comme on utilise NetCDF4, il est impératif de créer deux variables d'environnement, soient *WWATCH3-NETCDF* et *NETCDF-CONFIG*,

```
export WWATCH3_NETCDF=NC4
export NETCDF_CONFIG=/usr/lib/x86_64-linux-gnu
```

\*N.B. Pour trouver le répertoire de NetCDF4, on utilise **nc-config**,

```
which nc-config
```

Il serait pertinent d'ajouter ces variables d'environnement à notre *.bashrc* et d'ajouter *bin* et *exe* à notre *PATH*. Donc, dans le fichier *.bashrc*, on ajoute

```
export WWATCH3_NETCDF=NC4
export NETCDF_CONFIG=/usr/bin/nc-config
PATH=$PATH:/home/charlesedouard/Desktop/Travail/ww3/bin
PATH=$PATH:/home/charlesedouard/Desktop/Travail/ww3/exe
export PATH
```

Là, nous serions rendus l'étape de la compilation du modèle. La compilation s'effectue à l'aide de la commande *w3-make*, par contre il faut revenir sur les *switches* du modèle. J'ai mes propres *switches* qui s'assurent que tout fonctionne bien en sortant de la boîte (*switch-lizotte*). L'intérieur du fichier se lie comme suit

```
F90 NOGRB NOPA LRB4 NC4 TRKNC SHRD PR3 UQ FLX2 LN1 ST2 NL1 BTO ICO ISO
REF1 DBO TRO BSO XXO WNT1 WNXO CRT0 CRX0 00 01 02
```

Pour vérifier si les *switches* sont bonnes, on prend la commande `sort-switch <switches-file> -r`. La commande `-h` fonctionne pour avoir de l'aide,

```
sort_switch switch_lizotte -r
```

Finalement, on applique le script `w3-make`,

```
w3_make
```

La commande `w3-make` s'applique aussi sur les sous-routines individuelles de Wavewatch,

```
w3_make ww3_grid ww3_shell <...>
```

Et tout devrait bien compiler !

## 1.2 Débuggage NetCDF et OpenMPI

À ce point, il est possible que la compilation n'ait pas lieu car bien des bibliothèques NetCDF ne sont toujours pas installées. Si c'est le cas, on peut sudo-installer la commande **nc-config** à l'aide de la première commande et NetCDF-fortran à l'aide de la seconde, comme proposé dans une réponse produite sur [Stack Overflow](#).

```
sudo apt install libnetcdf-dev
sudo apt install libnetcdff-dev
```

Pour voir si tout est en ordre, on vérifie la commande **nc-config** à l'aide des commandes

```
nc-config --version
nc-config --all
```

Il se peut que la bibliothèque contenant **mpifort** n'est toujours pas installée sur notre ordinateur. Il existe beaucoup de moyen de l'installer, mais **cmake** fait des siennes en cherchant un ancien répertoire MPI qui n'existe plus. La [solution](#) proposée par un utilisateur GitHub est donnée par

```
# sudo apt install openmpi-bin
# sudo apt-get install mpich
sudo apt-get install libopenmpi-dev
```

## 2 DONE Faire fonctionner le modèle standard – <2023-09-19 Tue>

### 2.1 Re-Construire la grille, la mapsta et les forçages – <2023-09-19 Tue>

Heureusement, j'avais sauvegardé beaucoup de matériel avant l'annihilation de Mingan, dont quelques fonctions Python qui construisent les champs, soient

```
⇒ build-grids.py;
⇒ build-current.py;
⇒ build-wind.py.
```

J'en ai aussi profité pour les remettre à jour avec NetCDF4. Quelques problème d'orientation des variables et dimensions sont apparus. Il semble que je gérais très mal ce problème lors de maîtrise, mais le code est maintenant béton. Mentionnons que les NetCDF doivent impérativement avoir des coordonnées dans l'ordre

```
⇒ time (temps);
```

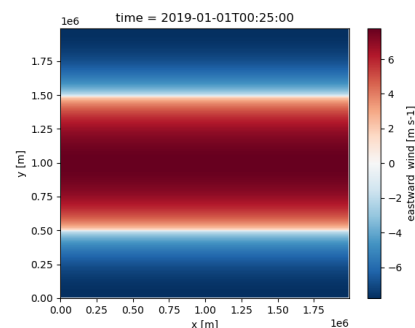


FIGURE 1 – *Output du vent sorti d'une rapide run de Wavewatch. Les axes sont bien ordonnés donc les input sont réglementaires.*

- ⇒  $y$  (latitude);
- ⇒  $x$  (longitude).

et que cet ordre doit **aussi** se retrouver dans le vecteur à l'intérieur du NetCDF et/ou du **xarray.Dataset**. Après ces efforts, le modèle semble bien rouler sans problème. Ce mardi le modèle était finalement bien installé avec toutes les librairies nécessaires, aujourd'hui les fonctions produisant les forçages nécessaires au couplage sont créés. Il ne reste donc qu'à vérifier si le modèle prend tout et après on peut se lancer dans le couplage.

Ce mercredi midi, le modèle fonctionne à merveille.

### 3 DONE Faire fonctionner le modèle en MPI – <2023-09-21 Thu>

#### 3.1 Déterminer les bonnes switches pour la compilation MPI – <2023-09-20 Wed>

Maintenant que le modèle *standalone* fonctionne efficacement, on doit le lancer en MPI. Il faudra d'abord changer les *switches*. Essentiellement, les *switches* permettent au script **w3 make** de prendre les fichiers *ftn* et de les retirer des lignes de code inutiles. Ce faisant, le script les transforme en fichiers compilable de type *.o* (objets) et les met dans le dossier *ww3/obj*. Par la suite, le script de compilation choisi (initialement *comp.gfortran*) transforme ces objets en exécutables qu'on peut retrouver dans le dossier *exe*. Les *switches* contrôlent donc tout, au final.

Ainsi, lors de la compilation standard de Wavewatch 3 (voir section 2.1), nous avons actionné les *switches* suivantes :

```
F90 NOGRB NOPA LRB4 NC4 *SHRD* PR3 UQ FLX2 LN1 ST2 NL1 BTO DBO TRO BSO ICO ISO REF1 XXO WNT1
WNXO CRT0 CRX0 TRKNC 00 01 02
```

Par contre, le [workshop de la NOAA](#) disponible en ligne insiste modifier deux *switches*, soient **DIST** et **MPI** pour remplacer la *switch* **SHRD**. Notre fichier de *switches* devrait donc ressembler à

```
F90 NOGRB NOPA LRB4 NC4 *DIST MPI* PR3 UQ FLX2 LN1 ST2 NL1 BTO DBO TRO BSO ICO ISO REF1 XXO
WNT1 WNXO CRT0 CRX0 TRKNC 00 01 02
```

#### 3.2 Compiler le modèle en MPI – <2023-09-21 Thu>

Ensuite, il faut le compiler. Normalement, pour utiliser MPI, on change **gfortran** par **mpif90** lors de la compilation. Par contre, ça ne semble pas marcher du tout. J'ai testé plusieurs types de compilations, tels que *mpifort* et *mpif90*, ce qui a pris beaucoup de temps. Après un peu de recherche, il semble que certaines sous-routines de Wavewatch 3 ne voulaient pas être compilées à cause d'une minuscule ligne d'appel MPI en Fortran. Par exemple, les sous-routines

- ⇒ *w3iorsmd.ftn*;
- ⇒ *w3iosfmd.ftn*;
- ⇒ *w3waves.ftn*

posaient problème. Mentionnons que la ligne problématique était

```
INCLUDE "mpif.h"
```

et que la plupart des compilateurs *mpifort* et *gfortran* peuvent avoir de la misère avec ça. On a donc du se tourner vers un autre compilateur, soit *ifort*.

#### 3.3 Installer les compilateurs ifort et mpiifort – <2023-09-21 Thu>

Avant tout, faire ça pour purger les débris liés à une précédente suite Intel,

```
sudo apt autoremove intel-hpckit
sudo apt update
```

Ensuite, on peut utiliser le [script d'un utilisateur d'un forum Fortran](#) pour installer *ifort*, *icc*, *icpc* et *mpi*,

```
sudo apt install \
    intel-oneapi-compiler-dpcpp-cpp-and-cpp-classic \
    intel-oneapi-compiler-fortran \
    intel-oneapi-mpi-devel
```

Si ça marche pas au final, on peut faire ça, mais c'est 9Go, c'est la [solution proposée par le site de Intel](#). En somme, ça installe toute la [suite Intel HPC toolkit](#), soit

```
sudo apt install intel-hpckit
```

Après, il faut ajouter *ifort* et les autres options de compilation au *PATH*, comme proposé dans le [Getting Started on Linux](#). En fait, il serait préférable d'ajouter ça à notre *.bashrc*,

```
source /opt/intel/oneapi/setvars.sh
```

On retourne dans *ww3/bin* et on tente de recompiler avec nos *switches* MPI, donc

```
w3_setup /home/charlesedouard/Desktop/Travail/ww3 -c Intel -s mpilizotte
...
w3_make ww3_shel
```

Et tout fonctionne! Le problème venait effectivement du compilateur. Plusieurs utilisateurs mentionnaient que *mpif90* a souvent des difficultés à jouer avec plusieurs versions de Fortran et de MPI. Ça devait être ce qui faisait dérailler la compilation du modèle avec *gfortran*. Par conséquent, nous ferons tout les compilations nécessaires à l'aide de *ifort* et *mpiifort* dès maintenant.

## 4 DONE Codage et test du couplage MPI – <2023-09-21 Thu>

### 4.1 Reconstruire la fonction de couplage du modèle shallow water – <2023-09-21 Thu>

Heureusement, la reconstruction des sous-routines de couplage ne m'a demandé que peu d'effort. Comme les frontières du modèle n'étaient plus périodique, il a été possible de retirer bien des sous-routines et variables superflues. Entre autres,

1. il a fallu remettre à jour les quantités MPI et les frontières *free slip* et *no normal flow*.
2. et retirer tous les mécanismes de points fantômes qu'on utilisait précédemment pour combler les lacunes du modèle périodique. Je suis assez heureux que ça soit parti, ça encombrait de beaucoup la sous-routine de couplage du modèle *shallow water*.

La sous-routine de couplage est extrêmement plus propre et moins encombrée. C'est un ménage du printemps réussi.

### 4.2 Remettre à jour les sous-routines de Wavewatch III – <2023-09-22 Fri>

Maintenant, il faut que les deux modèles se parlent, donc il faut

- Modifier la fonction de couplage du modèle */shallow water/1* pour qu'il communique pleinement avec Wavewatch III;
- Compiler Wavewatch avec les anciennes fonction modifiées lors de la maîtrise;
- Remettre à jour le *ww3 shel* pour que les dates soient bonnes;
- Recréer un *launcher*.

## 5 DONE Rencontre avec Louis-Philippe et David – <2023-09-22 Fri>

### 5.1 Création d'un « set-up » de référence du modèle « shallow water »

La rencontre a portée sur de nombreux points pertinents, mais avant tout il faudrait créer une expérience qui serait comparable dans tous les cas à  $n$  couches. Pour l'essentiel, on veut coupler le modèle Wavewatch avec un modèle *shallow water* à  $n$  couches, mais il faudrait que les modèles soient semblables pour pouvoir comparer l'effet des ondes sur chacun des *set-up*. Ce *set-up* de référence serait caractérisé par une fonction de courant barotrope (moyennée dans le temps) similaire pour tout nombre  $n$  de couches. Le but principal est de conserver l'énergie. Il sera impératif de jouer avec la valeur du paramètre  $r$  contrôlant le frottement au fond de la dernière couche. Le paramètre  $r$  se retrouve dans l'expression,

$$\frac{\partial u}{\partial t} = -ru. \quad (5.1)$$

Selon Louis-Philippe, la relation devrait être linéaire avec l'épaisseur de la dernière couche, mais ça reste à voir.

Set up de référence avec Shallow water.

⇒ 2 à 10 couches : et il faudrait avoir la même fonction de courant barotrope moyennée dans le temps, pour conserver l'énergie. Comment ajuster le bottom drag ? Ça devrait être linéaire selon LP.  $du/dt = -ru$

### 5.2 Retour sur la stratification

Pour ce qui est de la stratification, Louis-Philippe propose d'appliquer une exponentielle décroissante pour la densité avec un *decay scale* ( $\chi$ ) de 1000 m. Bref une stratification qui aurait la forme,

$$\rho(z) = \exp\{z/\chi\} \quad \text{où} \quad \chi = 1000 [m]. \quad (5.2)$$

### 5.3 Type de grille du modèle Wavewatch III

Cette semaine j'ai passé un peu de temps pour m'assurer que le modèle Wavewatch III est construit sur une grille Arakawa-A. Lors de la rencontre, nous avons tous vérifié et n'avons pas vu rien qui contredisait cette affirmation. L'indice principal qui me convainc en ce qui attrait à la grille de Wavewatch, ce sont les *output*. En effet, les *output* sont de type NetCDF et ils sont bien cordés sur une grille Arakawa-A avec les mêmes positions en  $x$  et  $y$ . Nous avons statué que c'était le cas et qu'il faudrait donc réaliser une interpolation linéaire ou une moyenne, ce qui a déjà été codé ce jeudi.

### 5.4 Varia et compute Canada

En parallèle, s'assurer que le modèle roule bien.

Louis-Philippe rappelle que lorsque tout sera en orde, on pourra envoyer le code à Compute Canada. « Ça c'est mon sponsor 'kqu-192-02' » – Louis-Philippe Nadeau.