

Contrat Été 2023

CARNET DE BORD, UNIVERSITÉ MCGILL

RÉALISÉ DANS LE CADRE
D'UN PROJET POUR

ISMER–UQAR

06/10/2023

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

Table des matières

0.1	DONE	Le problème des ondes de Kelvin au bord – <i><2023-10-03 Tue></i>	2
0.1.1		Partial slip condition	2
0.1.2		Changer l’hyperviscosité par une viscosité – <i><2023-10-04 Wed></i>	2
0.1.3		Changer le profil de vent – <i><2023-10-04 Wed></i>	2
0.2	DONE	Compute Canada – <i><2023-10-05 Thu></i>	3
0.2.1		Introduction	3
0.2.2		Importation des modules sur Compute Canada	3
0.2.3		Script bash de type SLURM – <i><2023-10-10 Tue></i>	3
0.2.4		Compilation des modèles numériques en MPI – <i><2023-10-10 Tue></i>	4
0.3	DONE	Nouveau schéma pour l’introduction de la dérive de Stokes dans les équations du mouvement – <i><2023-10-06 Fri></i>	4

0.1 DONE Le problème des ondes de Kelvin au bord – <2023-10-03 Tue>

Depuis quelques temps, toutes les simulations numériques non-couplées plantaient. Lorsque nous analysions les *output*, il était évident que quelque chose se passait aux bords. L'épaisseur de la couche de surface atteignait son minimum au côtes nord et sud, come si une onde de Kelvin beaucoup trop forte s'y était développée.

0.1.1 Partial slip condition

David eu la brillante idée de vouloir « ralentir » l'onde – et donc diminuer son amplitude – en changeant sa vitesse aux frontières. Pour se faire, nous avons choisi de modifier la *free slip condition* pour quelque chose qui retirerait du momentum aux frontières, donc une *partial slip condition*. Aux frontières, la dérivée serait plutôt exprimée selon

$$\left(\frac{\partial v}{\partial x} + \alpha v \right) \Big|_{x=0} = 0, \quad (0.1.1)$$

ce qui se traduit en différence finie par

$$\left(\frac{v(1) - v(0)}{\Delta x} \right) + \alpha \left(\frac{v(1) + v(0)}{2} \right) = 0. \quad (0.1.2)$$

Et puis le paramètre α devrait représenter l'inverse d'une longueur caractéristique L_c , de sorte que

$$\alpha = \frac{1}{L_c}, \quad \text{tel que} \quad L_c \sim 5 \text{ km}. \quad (0.1.3)$$

Mais malgré tout, ça n'a pas sembler fonctionner. En effet, avec le temps, le courant et sa dérivée se stabilisent et l'effet devient quasiment similaire à un *free slip* selon David. De plus, notre longueur caractéristique L_c ne peut pas être si grande que ça, ce qui fait en sorte que quelques points plus loin que la frontière et on est revenu à une condition quasi *free slip*. Donc, malgré que la *switch* est présente dans le fichier de paramètres, nous avons abandonné ce chemin.

0.1.2 Changer l'hyperviscosité par une viscosité – <2023-10-04 Wed>

David soupçonnait qu'il y avait un problème relié à l'hyperviscosité dans le modèle. Il aurait été possible que la manière qu'on avait implémenté l'hyperviscosité ne fonctionne pas bien avec les bords – étant donné que ça met en jeu un Laplacien double, etc. C'est pourquoi David m'a proposé de la changer pour une viscosité qui dépend du Laplacien directement. On aurait même pu mettre les deux, au final. La viscosité peut en fait être exprimée mathématiquement par

$$\text{Viscosité} = Ah_2 \nabla^2 u - Ah_4 \nabla^4 u. \quad (0.1.4)$$

Bien que nous ayons implémenté une viscosité au lieu d'une hyperviscosité, le résultat était toujours le même. Les couches du modèle atteignaient des épaisseurs nulles après quelques pas de temps aux frontières nord et sud.

0.1.3 Changer le profil de vent – <2023-10-04 Wed>

Après avoir fait pas mal de recherche sur les ondes de Kelvin dans le [1], David est arrivé à la conclusion que ce n'étaient pas des ondes de Kelvin, mais plutôt de l'*upwelling* induit par le courant de bord. Ce courant était trop fort à cause du vent, donc on est passé d'un vent

$$\tau = -\tau_0 \cdot \cos\left(\frac{2\pi y}{L_y}\right), \quad (0.1.5)$$

à

$$\tau = \tau_0 \cdot \left(1 - \cos\left(\frac{2\pi y}{L_y}\right) \right). \quad (0.1.6)$$

On a aussi diminué la valeur de τ_0 de moitié et tout fonctionne mieux. Le modèle produit maintenant des instabilités barocliniques sans problème.

0.2 DONE Compute Canada – <2023-10-05 Thu>

0.2.1 Introduction

Pour se connecter,

```
ssh -Y celiz2@beluga.computecanada.ca
```

Le mot de passe est laissé à la discrétion du lecteur de ce rapport. En résumé, nous travaillons sur la grappe de calcul de l'ETS, soit Beluga en l'honneur de la baleine qui est en train de crever dans le Saint-Laurent.

0.2.2 Importation des modules sur Compute Canada

Pour faire fonctionner nos modèles, on a besoin d'une couple de modules. Au même titre que sur la grappe de calcul Mingan, il est toujours possible de voir les modules listés à l'aide de la commande

```
module list
```

Et normalement, il est possible de voir les modules disponibles à l'aide de la commande

```
module avail
```

Par contre, il y a tellement de modules qu'on est fortement encouragé d'utiliser le modteur de recherche *spider* pour voir si un module existe. Par exemple,

```
module spider <module name>
```

Pour le modèle *shallow water*, on a besoin de deux modules, soient

```
# Shallow water model stuff
module load openblas
module load flexiblas
```

qui nous permettent d'avoir *BLAS* et *LAPACK*. Pour ce qui est de *fishpack*, tout s'est installé sans problème et les bibliothèques sont dans le répertoire local. Tandis que pour *Wavewatch III*, il faut absolument ajouter ces commandes à notre *.bashrc*, soient

```
# Wavewatch III stuff
module load netcdf-fortran
export WWATCH3_NETCDF=NC4
export NETCDF_CONFIG=/cvmfs/soft.computecanada.ca/easybuild/software/2020/avx512/Compiler/intel2020/netc
nc-config
PATH=$PATH:$HOME/projects/def-lpnadeau/celiz2/wavewatch3/bin
PATH=$PATH:$HOME/projects/def-lpnadeau/celiz2/wavewatch3/exe
export PATH
```

0.2.3 Script bash de type SLURM – <2023-10-10 Tue>

Pour faire fonctionner les module MPI sur compute Canada, il faut utiliser *srun*, mais la commande *mpirun* fonctionne aussi.

0.2.4 Compilation des modèles numériques en MPI – <2023-10-10 Tue>

Un résumé rapide du travail effectué pour que ça *marche* entre vendredi et mardi :

1. Le modèle *shallow water* doit **absolument** être compilé avec *gfortran*, *fishpack* n'a malheureusement pas été programé pour fonctionner avec *ifort* ;
2. Il semble que les deux modèles doivent **impérativement** être compilés avec le même compilateur pour que MPI fonctionne. Je n'avais initialement pas ce problème sur mon ordinateur personnel, mais il semble que ça soit nécessaire sur Béluga. On pourrait peut-être trouver un moyen définissant des variables globales de MPI, mais je préfère ne pas m'embarquer là-dedans. Ça serait une perte de temps monumentale.
3. La sous-routine *ww3 prnc* du modèle Wavewatch (soit la routine qui transforme les fichiers NetCDF en input pour Wavewatch) ne peut pas être compilée avec *gfortran*. Pour vrai, j'ai tout essayer. C'est comme si – malgré l'importation des modules nécessaires (*netcdf*, *netcdf-fortran*, *netcdf-mpi-fortran*, etc) – la fonction **USE MPI** dans la sous-routine *ww3 prnc* est incapable de compiler.

Comme solution, je compile donc toutes les sous-routines qui ne sont pas le *shel* avec *ifort* et je compile le *shel* en MPI avec *gfortran*. C'est malheureusement le seul moyen que j'ai trouvé.

Dans le meilleur des mondes, j'aurais aimé que tout se compile soit avec *gfortran* ou avec *ifort*, mais il semble que ça doit en être autrement. Je voulais mettre ça ici, parce que ça m'a pris beaucoup de temps pour *fuck around and find out* tout ça.

0.3 DONE Nouveau schéma pour l'introduction de la dérive de Stokes dans les équations du mouvement – <2023-10-06 Fri>

Initialement, les équations du mouvement d'un système Boussinesq sont exprimées selon l'expression

$$\frac{\partial \mathbf{u}}{\partial t} = (f + \zeta) \hat{\mathbf{k}} \times \mathbf{u} = -\nabla B + \mathbf{D} + \frac{\tau_a}{\rho_o H}, \quad (0.3.1)$$

où la fonction de Bernoulli (B) est exprimée par $B = p/\rho_o + \mathbf{u}^2/2$.

Dans leur résumé, SUZUKI et FOX-KEMPER [2] définissent la dérive de Stokes \mathbf{u}_S comme une contribution lagrangienne à notre écoulement, de sorte qu'on peut décrire ce courant lagrangien \mathbf{u}_L par

$$\mathbf{u}_L = \mathbf{u} + \mathbf{u}_S. \quad (0.3.2)$$

En somme,

- ⇒ Ce courant lagrangien \mathbf{u}_L se substitue dans les termes d'advection, de la même manière qu'un référentiel en mouvement ;
- ⇒ Les termes de Stokes-Coriolis, Craik-Leibovic et la nouvelle fonction de Bernoulli découlent donc directement cette au référentiel en mouvement.

Lorsqu'on ajoute cette contribution lagrangienne à notre courant, l'expression 0.3.1 devient plutôt

$$\frac{\partial \mathbf{u}}{\partial t} = (f + \zeta) \hat{\mathbf{k}} \times (\underbrace{\mathbf{u}}_{\text{Courant Lagrangien}} + \underbrace{\mathbf{u}_S}_{\text{B.-Stokes}}) = -\underbrace{\nabla B_S}_{\text{B.-Stokes}} + \underbrace{\mathbf{D}}_{\text{Contr. des Vagues}} + \frac{\tau_o}{\rho_o H}. \quad (0.3.3)$$

où la nouvelle fonction de Bernoulli qui tient compte de la dérive de Stokes est donnée par

$$B_S = B + \mathbf{u} \cdot \mathbf{u}_S + \mathbf{u}_S^2/2. \quad (0.3.4)$$

Bibliographie

- [1] G. K. VALLIS, *Atmospheric and Oceanic Fluid Dynamic : Fundamentals and Large-scale Circulation*, Second Edition. The Edinburg Building, Cambridge CB2 2RU, UK : Cambridge University Press, 2006.
- [2] N. SUZUKI et B. FOX-KEMPER, « Understanding Stokes forces in the wave-averaged equations », *Journal of Geophysical Research : Oceans*, t. 121, n° 5, p. 3579-3596, 2016.