

Contrat Été 2023

RAPPORT HEBDOMADAIRE – MCGILL

RÉALISÉ DANS LE CADRE
D'UN PROJET POUR

ISMER-UQAR

17/03/2023

Table des matières

0.1	Le modèle est en état de fonctionner [2/2]	2
0.1.1	Mise en contexte	2
0.1.2	DONE Régler le problème des <i>oldfile</i>	2
0.1.3	DONE Régler le bordel des <i>compile model</i> et recompiler efficacement	2
0.2	Modifications à apporter au modèle <i>shallow water</i> cette semaine [1/1]	3
0.2.1	DONE Retirer la <i>slab layer</i>	3

0.1 Le modèle est en état de fonctionner [2/2]

0.1.1 Mise en contexte

Avant de débiter les changements au modèle *shallow water*, assurons nous

1. qu'il est compilable ;
2. qu'il peut rouler sans erreur sur toutes les machines (donc qu'on n'utilise pas des routine Fortran trop vieilles, entre autre).

Au cours de ma maîtrise, quelques petits problèmes conceptuels et techniques s'étaient déjà accumulés au fil du code. Nous allons donc régler ces problèmes-là et s'assurer que tout marche, avant tout.

0.1.2 DONE Régler le problème des *oldfile*

Comme on avait couplé le modèle *shallow water* avec Wavewatch III dans le passé, les *restart files* devaient avoir une forme légèrement différente que dans la version de Tianze. Principalement, il fallait ajouter la dérive de Stokes du dernier pas de temps **à l'intérieur du restart file**. À l'époque, la solution que j'avais trouvée était de rajouter une *switch* nommée **oldfile** dans le fichier de paramètres, *parameter.f90*. Ceci avait comme problématique la coexistence de deux types de *restart files* un peu partout. Ça augmentait malheureusement le risque d'erreurs à chaque fois qu'on relançait des expériences, mais ça m'évitait de devoir relancer mes plus longues expériences – qui contenaient souvent des résultats importants. C'était une solution temporaire qui s'est malheureusement éternisée jusqu'à la fin de la maîtrise. **Je me met une note** qu'on devrait toujours avoir les mêmes type de *restart files* pour éviter d'avoir **beaucoup trop** de *switches*.

Comme solution, couplé ou non, les *restart files* vont maintenant contenir la dérive de Stokes et le dernier τ à la surface de l'eau. Et puis, la *switch oldfile* a tout simplement été retirée, maintenant qu'on a supprimé ces vieilles données-là.

0.1.3 DONE Régler le bordel des *compile model* et recompiler efficacement

Un lecteur avisé de mon mémoire aurait remarqué la présence de 3 modèles différents. Le tout avec de nombreux fichiers de paramètres associés à chacun. C'est pourquoi, lors de ma maîtrise, j'ai du codé plusieurs compilateurs avec plusieurs dossiers et plusieurs listes différentes de paramètres. Avec le temps, le nombre de compilateurs a grandit de manière exponentielle dans mes dossiers, il a donc fallu faire un gros ménage juste pour faire fonctionner le modèle *shallow water* sur mon portable. Comme les différentes librairies Fortran ne sont pas les même sur mon ordinateur, il a fallu reconstruire un compilateur. Mais ça fonctionne maintenant à la perfection !

Par précision, il a fallu installer les librairies à l'aide des commandes

```
sudo apt-get install libfftw3-dev libfftw3-doc
```

et pour trouver le *path*,

```
dpkg -L libfftw3-bin
```

ce qui nous permet d'ajouter

```
fftw_path=/usr/share/doc/libfftw3-bin/
```

à notre *makefile* (*compile model* dans notre cas). Pour l'instant, la compilation fonctionne ; j'ai lancé un test pour la nuit ce <2023-03-14 Tue>.

Pour que le modèle roule bien, mentionnons qu'il a besoin de certains dossiers et fichiers, soit

⇒ tous les fichiers texte associés aux nombres d'onde ;

- ⇒ les dossiers *data* et *specdata* vides pour les données ;
- ⇒ un dossier *rst* vide pour les *restart files*.

J’ai donc créé un fichier vide prêt à partir.

0.2 Modifications à apporter au modèle *shallow water* cette semaine [1/1]

0.2.1 DONE Retirer la *slab layer*

Comme on veut un modèle à plusieurs couches, la *slab layer* ne sert plus vraiment à rien. Aussi, comme on veut nettoyer le code, retirer la *slab layer* est probablement la première chose à faire. Déjà, Ça va nous débarrasser de quelques sous-routines superflues et beaucoup de variables maintenant inutiles.

J’ai ajouté la quantité **UStokes** et **VStokes** dans les sous-routines *p corrections.f90* et *rhs.f90* (avec un `top(k)`), de sorte que ça apparaisse dans le *RHS* de la conservation de la masse et pour le gradient de pression à la surface. Bref, cette quantité couplée a simplement remplacé *Uek* et *Vek* dans les deux même sous-routines. **UStokes** et **VStokes** sont définis comme nulles dans la sous-routine *initialize.f90* et ne sont pas mises à jour sans l’activation de *switch* (COU .eq. .true.), qui permet le couplage avec le modèle de vagues.

J’ai aussi profité de cette occasion pour alléger les lignes de codes liées au τ_o à la surface. Avant, j’avais deux *switches* pour choisir si le transfert de momentum de l’atmosphère devait être sur la première couche, la couche d’Ekman ou couplée avec Wavewatch III. Bref, c’était un bordel et j’en ai profité pour faire un gros ménage dans tous ces *switches* là. Ça pourrait expliquer pourquoi tout ça me prend un peu plus de temps que prévu, disons. . .

Finalement, toutes les sous-routines *dump etc.f90* – que Tianze avait rajoutées – ont pris beaucoup de temps à réorganiser. Dans chacune d’elles, il y avait un *Uek* qui était couplé avec d’autres quantités qui servaient de diagnostique, etc. Dans le même ordre d’idée, le courant est maintenant exporté avec la même **shape** que tous les autres champs. Avant, le courant de **toutes les couches** était exporté dans le même fichier avec le préfixe *u_o* (pour le différentiel de *Uek*). Maintenant, ce sera *u₁*, *u₂*, *u₃*, etc. De cette manière, ma fonction Python (**readata.py**) qui ouvre les *output* du modèle *shallow water* pour les transformer en fichier **NetCDF** va seulement ouvrir des fichiers de même taille. C’est quelque chose que je n’avais pas entrepris pendant ma maîtrise, car je n’ai simplement pas eu le temps pendant ma rédaction. Je suis bien heureux que ça soit enfin fait. Essentiellement, le modèle commence à être bien plus *clean*, pour reprendre les mots de David.

Toujours vers la fin de la semaine, le modèle semble encore *explorer*, comme la fameuse ligne *thickness too small* aime si bien nous le rappeler. Ce vendredi, j’ai investigué et il semble que ce soit parce que le transfert de momentum depuis l’atmosphère est défini comme

$$\tau_0 = \frac{\tau_{atm}}{\rho_o}, \quad (0.2.1)$$

ce qui voulait dire que j’envoyais des vents de l’ordre de 1000 m/s au modèle. Pour information, on lance la même expérience que (Chen et al, 2021), mais avec seulement deux couches, ce qui pourrait s’apparenter à si $H_{Ek} = H_1$.

Le test lancé pendant la fin de semaine devrait nous encourager (Voir figure 1).

TABLE 1 – Valeur des différents paramètres de l'expérience de Chen et al (2021).

Paramètres	Symbole	Valeur
Taille du domaine	$L_x = L_y$	2000 km
Pas de temps	Δt	300 s
Paramètre de Coriolis	f	$7 \times 10^{-5} \text{ s}^{-1}$
Amplitude du vent	τ_{atm}	0.1 N m^{-2}
Coefficient de viscosité biharmonique	A_{bh}	$\text{dx}^4 \times 10^{-5} \text{ s}^{-1}$
Coefficient de frottement au fond	r_{drag}	10^{-7} s^{-1}
Coefficient dissipation du Laplacien inverse	r_{InvLap}	$(2\pi/L_y)^2 \times 10^{-6} \text{ s}^{-1}$
Épaisseur de la couche supérieure	H_1	1000 m
Épaisseur de la couche de fond	H_2	3000 m
Vitesse des ondes internes de gravité	c_{bc}	2 ms^{-1}
Switch d'activation d'Ekman	δ_{BF}	1

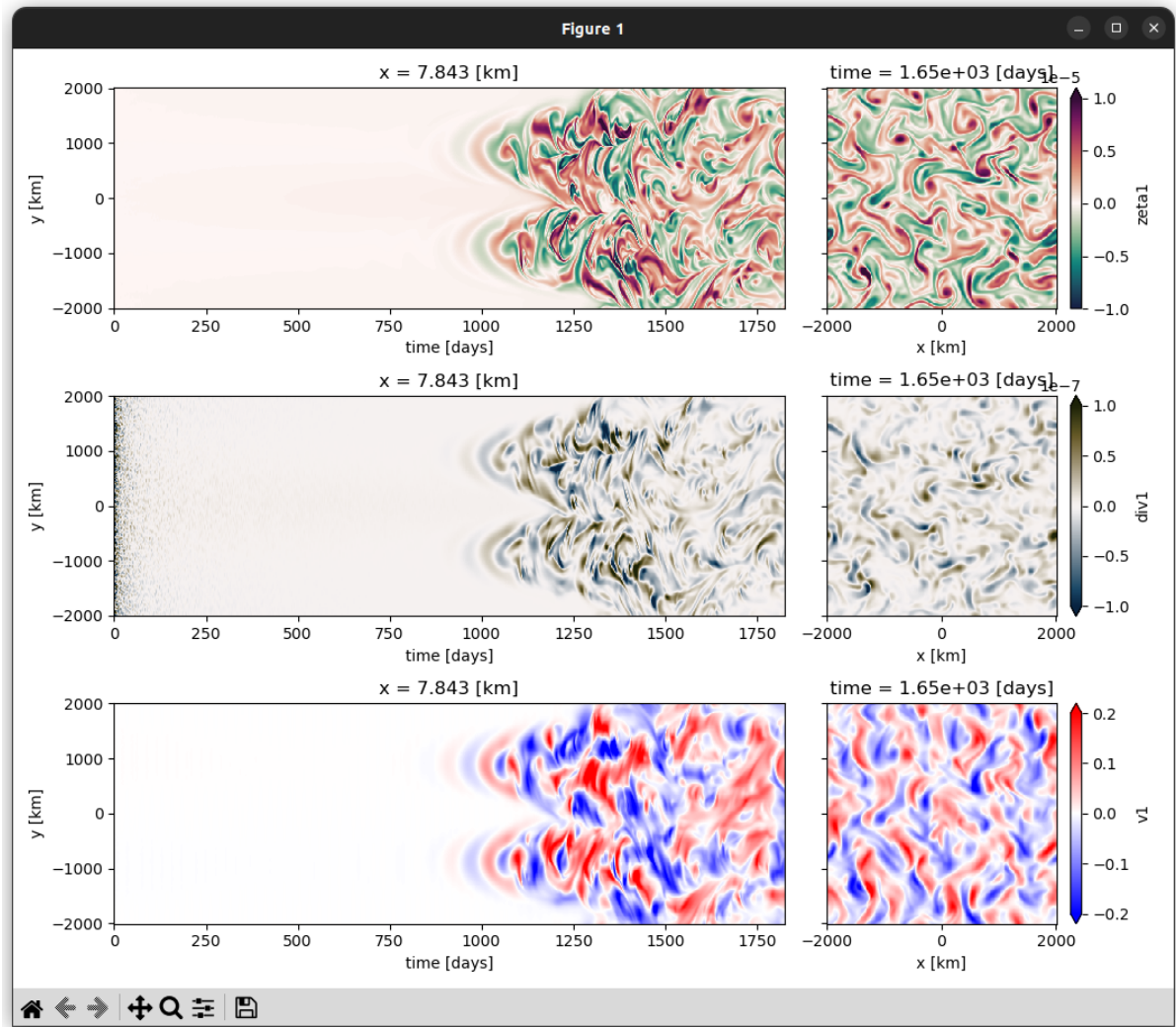


FIGURE 1 – Résultat du test de qualité du modèle. En ordre, diagramme de Hovmoller de la divergence, du rotationnel et du courant dans la première couche.