Bachelier Céline
Master 1

07/01/2007

Projet de modélisation de l'Unité d'Enseignement 7Projet d'analyse et de modélisation

Université de la Méditerranée
Centre d'océanographie de Marseille
Master de sciences de la mer et de l'environnement

Année 2006/2007

Sommaire

Modélisation de la Méditerranée Nord Occidentale

I. Présentation des outils de ROMS

Dans un premier temps, nous devons télécharger sur internet les outils qui seront nécessaire pour faire notre modélisation. Nous allons donc avoir besoin des librairies et scripts suivant :

- Aforc_NCEP (Centre National de Prédiction Environnementale) : utilisé pour la prédiction du temps dans le monde (hydrologie, météorologie, alertes, tornades ouragans, …). Nous utiliserons aussi Forecast_tools pour ce genre d'analyses.
- Aforc_QuikSCAT et QuikSCAT_clim : prennent leurs données du satellite quikSCAT. Ces derniers nous donnent les vecteurs de vent dans tous les océans et leurs climatologies.
- Air_sea : concerne ce qui ce passe à l'interface océan-atmosphère (l'albédo et autres mesures physiques).
- COADS05: fournit également des données physiques, mais aussi la climatologie, …
- Compile: sert pour la compilation de ROMS.
- Diagnostic_tools : Ce sont des scripts matlab pour les animations et les analyses statistiques du modèle, pour vérifier sa validité.
- Mask: utilisé pour masquer certaines surfaces de la grille de notre modèle afin de faciliter la modélisation (lissage des côtes).
- M_map: nous donne la carte du monde.
- Netcdf_g77, netcdf_ifc : Ce sont des librairies en fortran pour compiler Linux avec g77 et ifc.
- Netcdf_matlab, Mex_60, Mexnc, Netcdf_*86-84, Opendap-tools, sont des librairies.
- Preprocessing_tools : Scripts de matlab qui vont nous permettre d'avoir quelques graphiques de simulation.
- Roms_Agrif : le code source en fortran.
- Run : c'est l'endroit ou va se dérouler notre modélisation. Nous devons toujours nous trouver là pour n'importe quelle exécution.

- SeaWIFS : prend ses données du satellite du même nom. Il mesure la concentration de phytoplancton dans l'océan et la traduit par différentes couleurs.
- SST_pathfinder (Several Sea Temperature) : nous donne quelques températures de surface à très haute résolution.
- Tides : localise les différentes marées, tout comme TPXO6 et TPXO7.
- Topo : concerne la topographie globale (bathymétrie, altimétrie).
- Visualisation_tools : Ce sont des scripts matlab permettant de visualiser graphiquement le modèle une fois ce dernier terminé.
- WOA2001 et WOA2005 (World Ocean Atlas) : consiste à faire une climatologie du monde avec des propriétés in situ.

II. Mise en place et préparation du modèle

Une fois les outils de Roms téléchargés, afin de pouvoir les utiliser, nous les avons décompacter et désarchiver (avec les commandes 'gunzip' et 'tar', en fonction dans le terminal).

Nous nous plaçons dans le dossier Run (dans matlab et dans le terminal), pour toute exécution de scripts ou autre. C'est à cet endroit que les données principale ce trouve pour faire les simulations.

Nous allons nous apercevoir qu'en exécutant jobcomp, pour tester la compilation, cela ne fonctionnera pas. Jobcomp utilise le compilateur 'ifort' au lieu de'g77'.
Nous ferons donc la modification nécessaire.

Par suite, nous irons dans matlab pour exécuter le script 'start', afin de vérifier si tous les chemins pour accéder aux outils de Roms sont bons et présents.
Mais encore une fois il nous apparaîtra un message d'erreur nous disant qu'il manque des chemins, et qu'il ne trouve pas loaddap et mexcdf. Donc à la fin du script 'startup.m', il faudra rajouter :

```
Addpath([mypath,'mex60'])
Addpath([mypath,'Opendap_tools/FEDORA'])
Disp('It is done')
```

Afin de ne pas exécuter 'start' à chaque fois que l'on ouvre matlab, il faudra renommer 'start.m' en 'startup.m'. Et nous ouvrirons matlab à partir du terminal après s'être placé dans 'Run'.

Après plusieurs simulations d'un mois nous allons faire une simulation sur plusieurs années, dans notre cas 5 années complètes.

III. Comment faire fonctionner notre modélisation ?

1. Conditions initiales

Il faudra personnaliser notre modélisation, modifier quelques scripts afin de l'adapter à notre cas, ici la Méditerranée Nord Occidentale.

Dans 'romstools_param.m', nous allons modifier :
- le titre, la position (latitude minimale et maximale : 35° à 46°Nord, longitude minimale et maximale : -2° à 20°)
- la résolution de la grille (nous prendrons ¼, c'est-à-dire que chaque degré correspond à 4 mailles de grille)
- les frontières ouvertes (ouvertes à l'Ouest et au Sud et fermées à l'Est et au Nord)
- la profondeur minimale (hmin =10)
- le nombre de niveau vertical de profondeur que nous laisserons à 32 (coordonnées vertical s, qui suit la topographie).

Remarque : nous avons trouvé un petit bogue avec 'Edit mask', elle nous rétrécie la carte, c'est pour cela que nos coordonnées de position sont un peu larges.

Dans 'param.h', il faut créer notre grille :

```
If defined NWMED
    Parameter (LLm0 = 75,    MMm0 = 58,    N = 32)
Endif
```

Et de ce fait dans 'cppdefs.h', il faut modifier un peu la programmation. En effet, nous avions précédemment défini un autre modèle (Benguela). Il faut également redéfinir les frontières ouvertes et fermées.

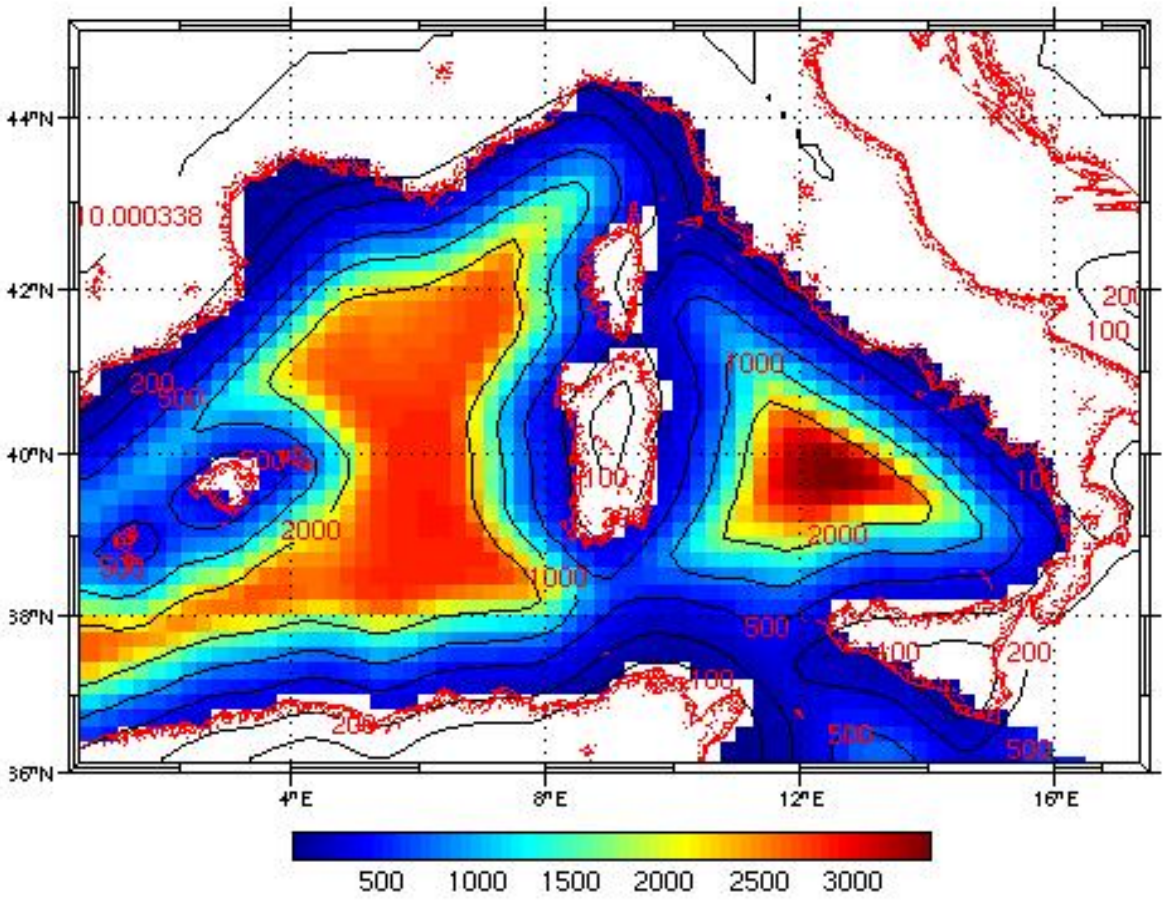
```
Undef     BENGUELA
Define    NWMED
```

2. Quelques simulations

Nous pouvons à présent tester quelques graphiques et commencer à mettre en place le modèle.

Dans matlab, nous allons pouvoir exécuter 'make_grid', ce qui va nous donner la grille du modèle (une grille d'Arakawa de type C). Nous allons pouvoir masquer les zones de terre ou de mer qui ne nous intéresse pas ou bien qui permettrons de faciliter la modélisation.
Cf. http://www.com.univ-mrs.fr/~b302189/

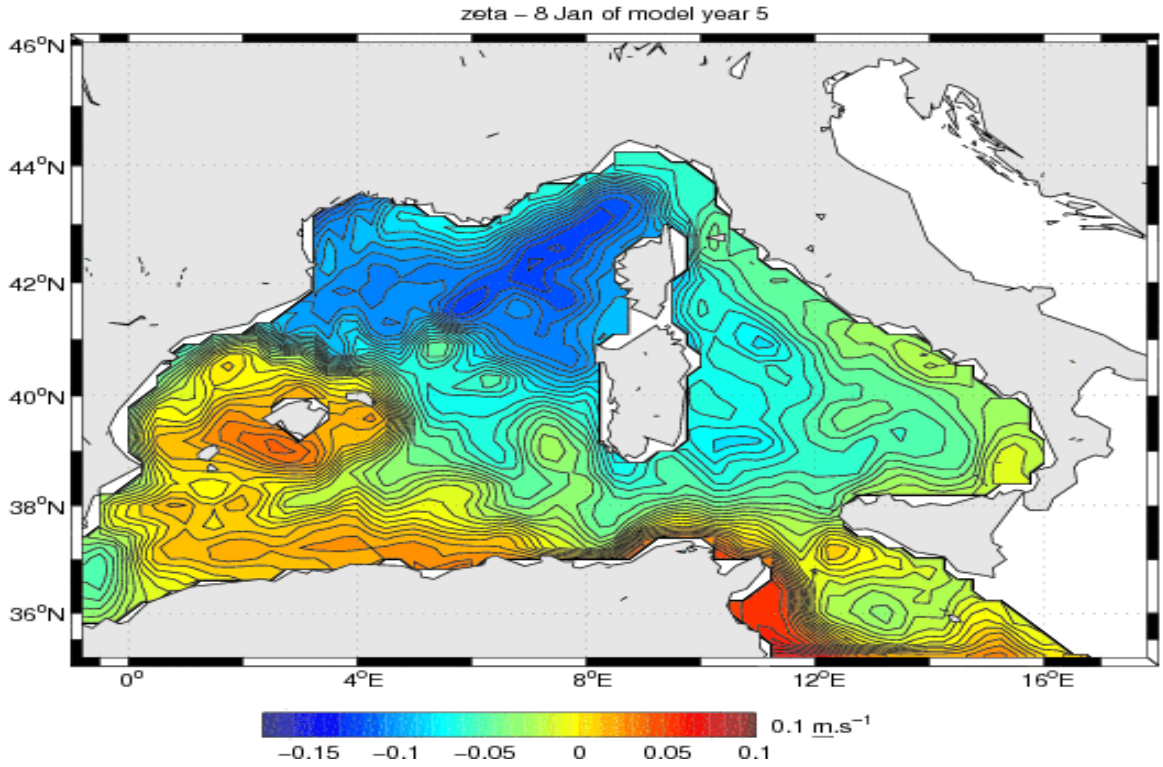
Après cette manipulation nous obtenons la bathymétrie de notre modèle.



Nous pouvons remarquer les lissages des côtes que nous avons fait afin de faciliter le déroulement de la modélisation.

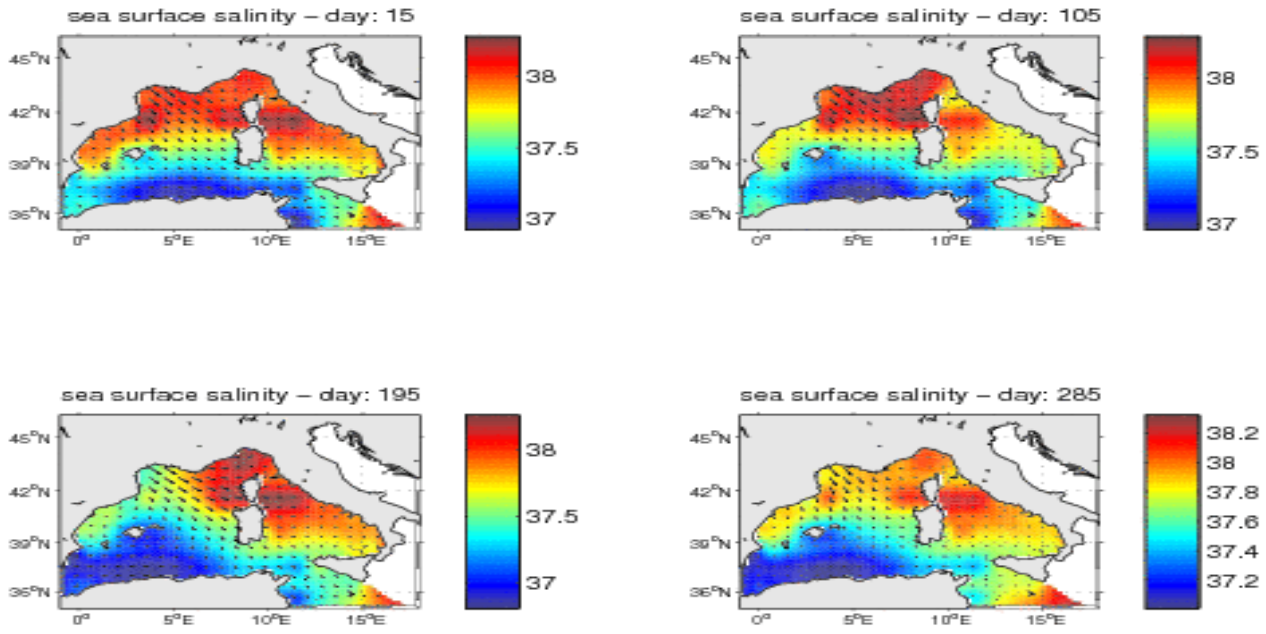
Par la suite, nous pourrons exécuter ‘make_forcing’, ‘make_clim’ et ‘make_tides’ (en cas de marée, mais ici nous négligerons ce phénomène). Ces derniers programmes nous donnerons des graphiques et des profils.

Pour gagner du temps puisque nous faisons une simulation de plusieurs mois, nous allons pouvoir exécuter dans matlab ‘ad_config’. Nous obtiendrons ainsi tous les graphiques fournis par les commandes ‘make_forcing’ et ‘make_clim’. Exemple : voir figures ci-dessous.

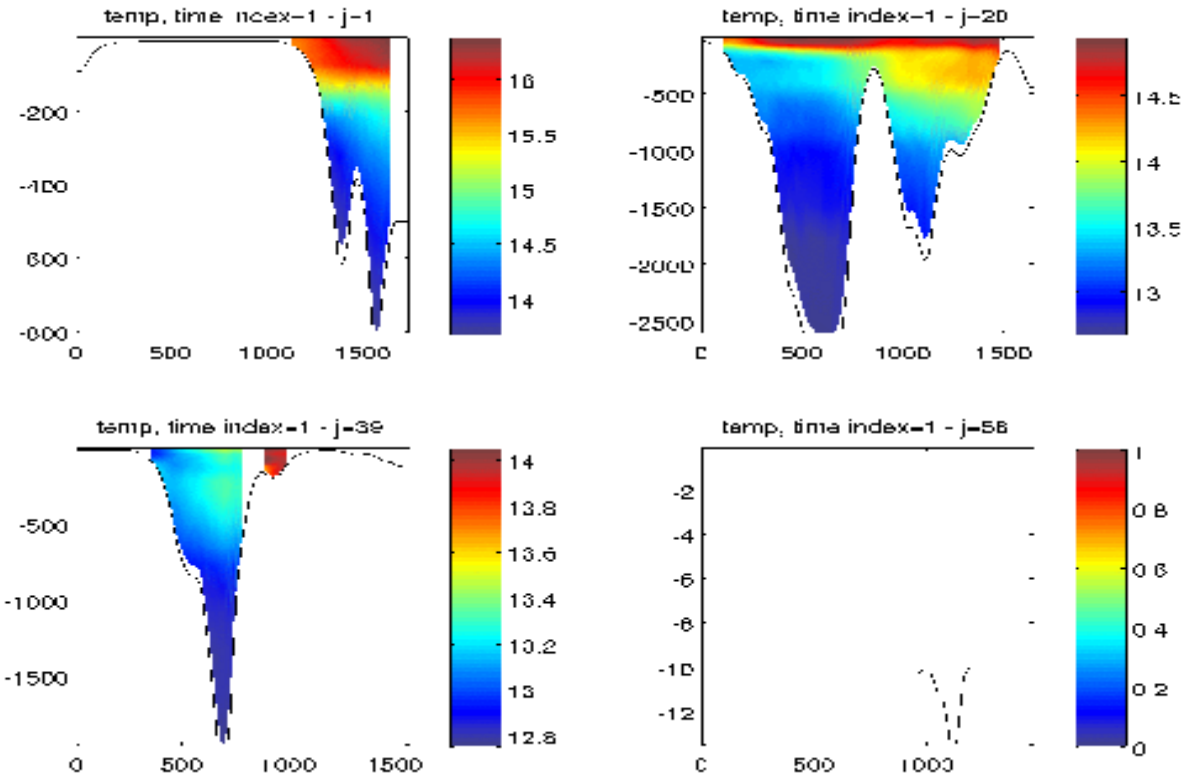


L'élévation de du 8 janvier de la 5ème année

Dans ce cas, nous remarquons une élévation majoritairement plus importante au Sud du bassin.
Elle varie de - 0.15 à 0.1 m/s.



Salinité de surface sur quatre jours (le 15ème, 105ème, 195ème, et 285ème



Profil de température à différente latitude

3. Finalisation du modèle

Nous pouvons à présent compiler, en retournant dans le terminal et en exécutant jobcomp.

Les dernières modifications à apporter se trouvent dans le fichier 'roms.in', utilisé pour les petites simulations, et dans 'roms_inter.in', utilisé pour les plus importantes (ici plusieurs années).

Nous devons faire un certain nombre de calcul : le pas de temps, la durée de la simulation et le temps au bout duquel va se faire l'archivage.

Calculons le pas de temps :

$$\Delta te \leq \frac{1}{2 * (g*h)^{1/2}} * \frac{1}{\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)^{1/2}} \quad \text{c'est à dire} \quad \Delta te \leq 37.39 \text{ secondes}$$

Afin de faciliter les calculs nous prendrons Δte = 36 s.

$\Delta ti \sim 60 * \Delta te$ donc $\Delta ti = 2160$ *secondes (pour la suite* $\Delta ti = \Delta t$ *)*

Nous pouvons maintenant déterminer la durée de simulation :

$$Durée = NTIMES * \Delta t$$
$$= \frac{86400 * 30}{2160} = 1200\ s$$

Calculons NWRT, qui correspond au temps au bout duquel les données vont être archivées.

$NWRT = \frac{2 * 86400}{\Delta t} = 80\ s$ si l'on veut garder les données tous les 2 jours

Nous fixerons ensuite NTDFAST à 60. C'est un autre paramètre du modèle qui représente le nombre de fois que tourne une boucle barotrope à l'intérieur d'une boucle barocline.

Une fois ces paramètres calculés, nous modifions 'run_roms.csh' (script shell) pour ajouter le pas de temps, le nombre de jour et le nombre de mois et d'année. Nous ajouterons ensuite dans 'compute' :

```
Nice +6 ./$CODFILE
```

Ceci permet de créer une acceptabilité de l'ordinateur.

En exécutant 'roms_inter.in' dans le terminal, et grâce à ce dernier script shell, les simulations vont être sauvegardé dans 'roms_avg_Y1M1.nc' (donnée moyennée, par exemple), et 'roms_rst.nc' sera créé et correspondra à la condition initiale du prochain mois.

Dans le but de laisser travailler l'ordinateur, et éteindre notre session, il faudra exécuter la commande suivante dans le terminal :

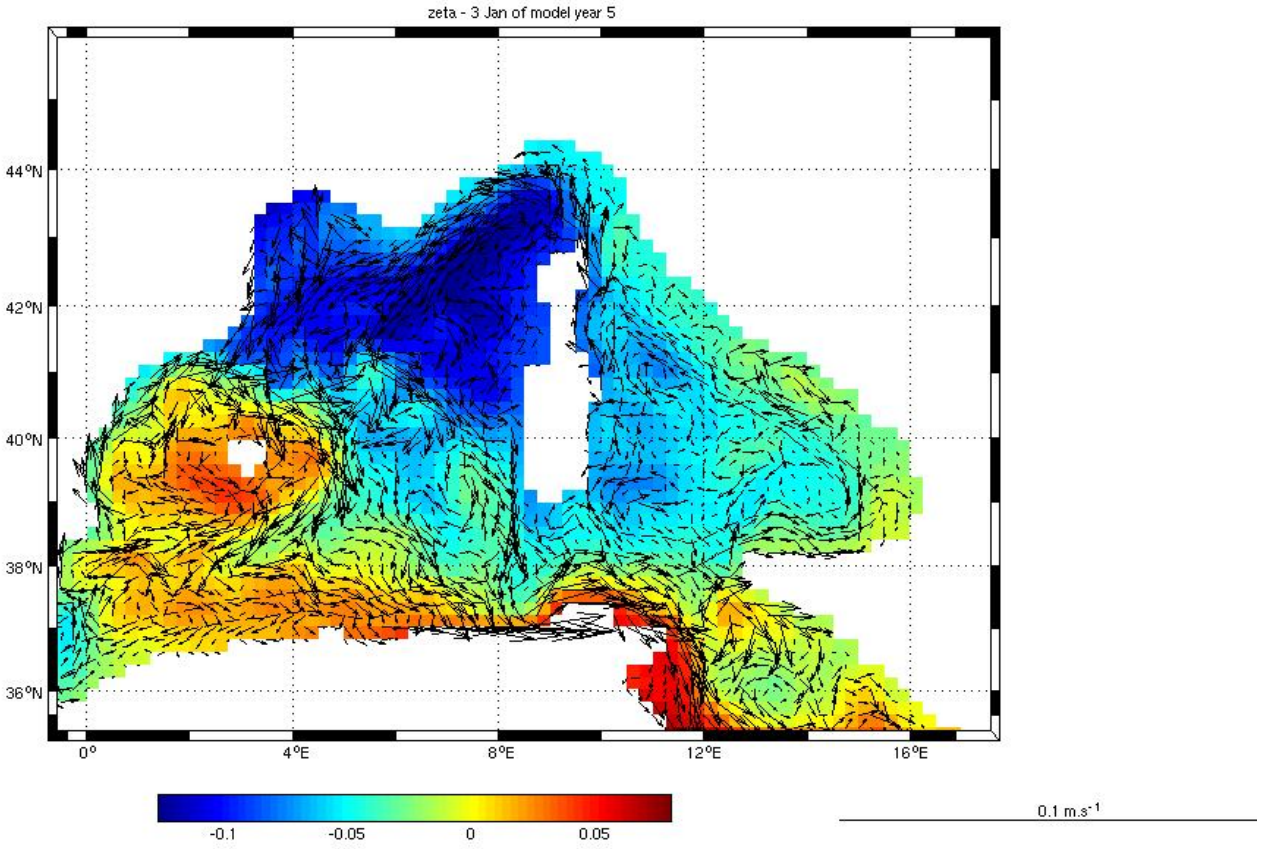
```
nohup ./run_roms.csh > exp.out&
```

Après quelques jours, nous pouvons désormais avoir une simulation de notre modèle grâce à la commande 'roms_gui' utilisable dans matlab.

Nous obtiendrons les élévations, les températures, les vecteurs de vitesses et leurs intensités, les salinités et d'autres paramètres.

Mais il ne faut pas oublier de vérifier notre modèle, d'en déduire les artéfacts, ainsi que les bons fonctionnements qui retrace la réalité.

Ci-dessous la simulation du 3 janvier de la dernière année :

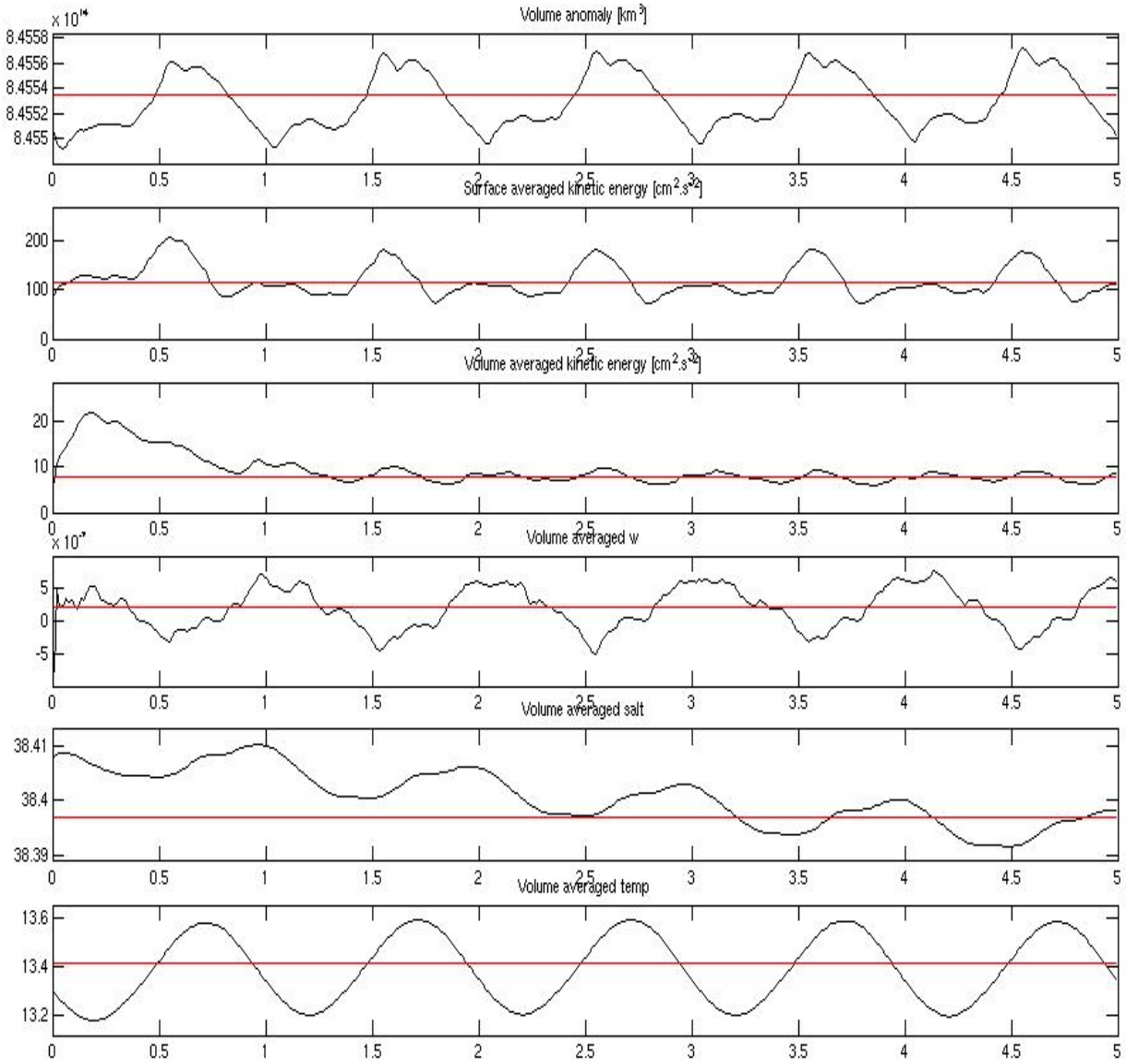


Elévation et champs de vitesse

IV. Diagnostics et discussions

Nous allons vérifier si le modèle à bien fonctionner.
Nous nous servirons du dossier 'diagnostic_tools', et de la commande 'roms_diags'.
L'ordinateur calculera des grandeurs dans le but de faire le diagnostic du modèle (énergie cinétique, volume total, salinité, température, l'anomalie de volume).
Ces données sont conservées dans le fichier 'roms.mat'.
Nous pouvons alors dessiner ces valeurs en exécutant 'plot_diags'.

Cf. graphique ci-dessous.



Le modèle se met en place relativement rapidement, environ 6 mois pour l'anomalie de volume et l'énergie cinétique moyennée en volume et en surface, environ 1 an pour le volume plus profond. Ceci est sans doute dû au fait que l'une des conditions initiales dit qu'à partir de –1000 mètres il n'y a plus de mouvement.

Nous remarquons également que le modèle respecte bien la saisonnalité (oscillations saisonnières).

A contrario, la salinité pose un petit problème (souvent représenté dans les modèles), elle diminue au fur et à mesure des années.
Ceci peut être expliqué par les eaux siciliennes moins salées que la normale pour compenser la salinité des eaux atlantiques.

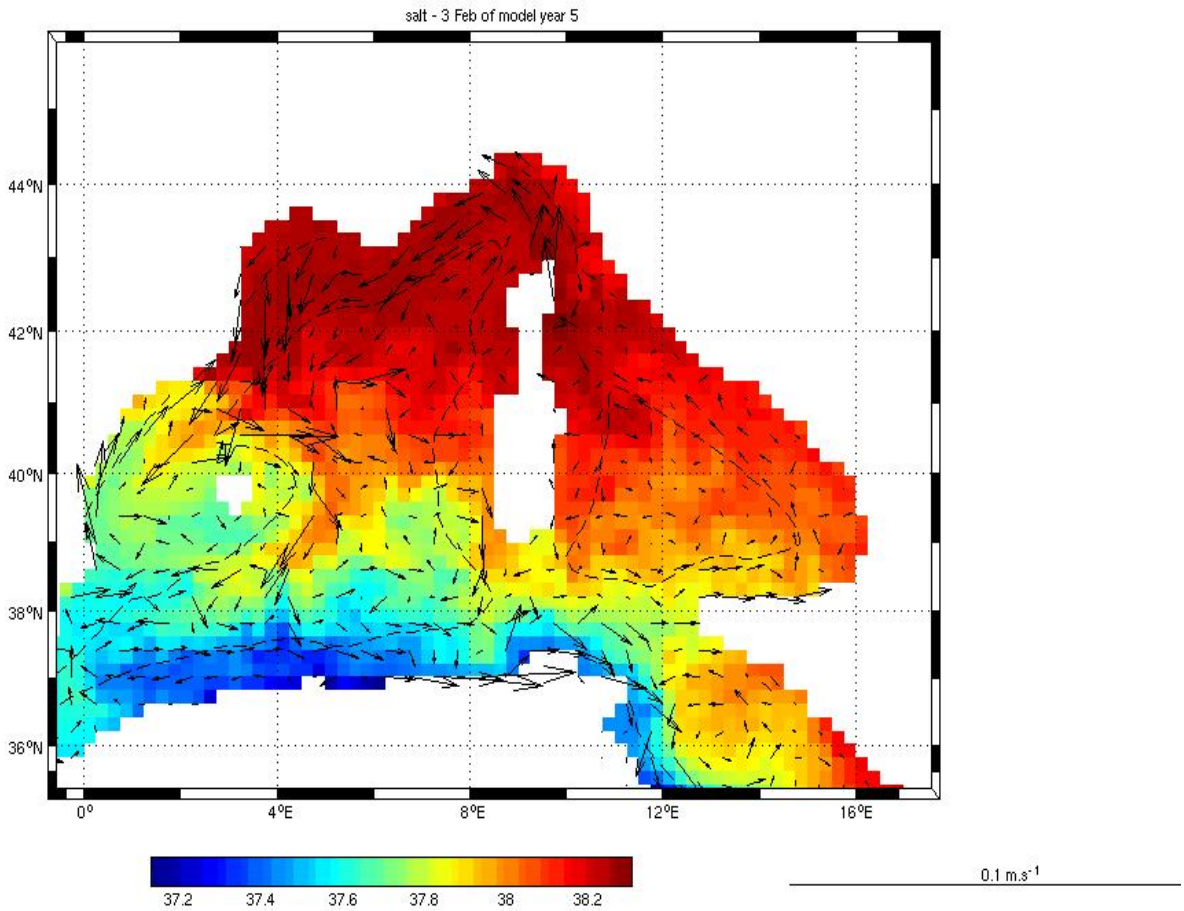
1. Comparaison de notre modèle

Comparons à présent notre modèle à d'autres modélisations de la méditerranée Nord Occidentale qui ont été faites : celles de M. A. Ahumada et A. Cruzado, et celles de Millot et Taupier.

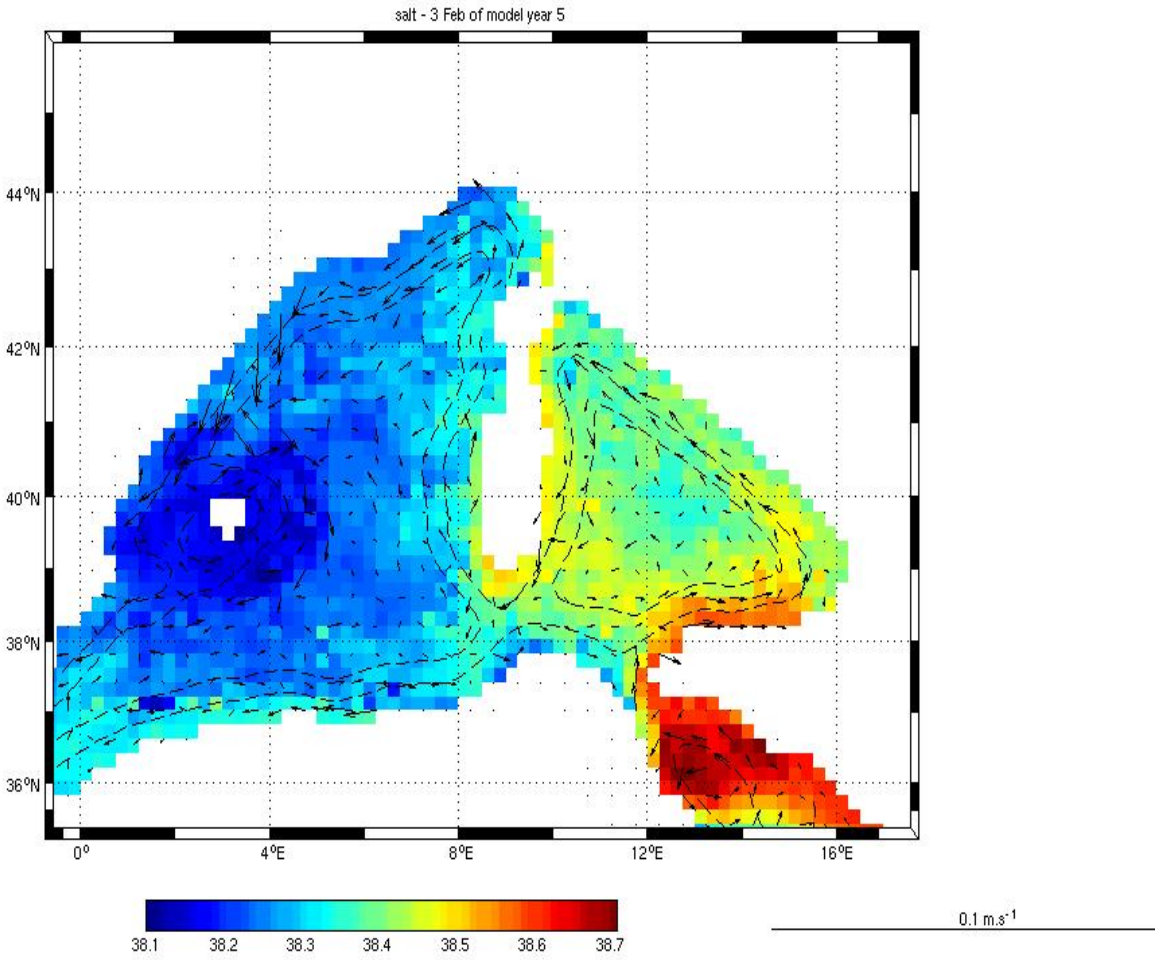
La modélisation d'Ahumada et Cruzado concerne uniquement la Méditerrnée Nord Occidental jusqu'en dessous de la Sardaigne. Nous élargirons donc nos comparaisons avec le modèle de Millot et Taupier.

La salinité :

Pour commencer, nous pouvons remarquer que la salinité augmente avec la profondeur. Les eaux plus salées étant plus dense elles sont donc plus profondes.



Salinité et champs de vitesse à -30 mètres de profondeur en février de la dernière année de simulation

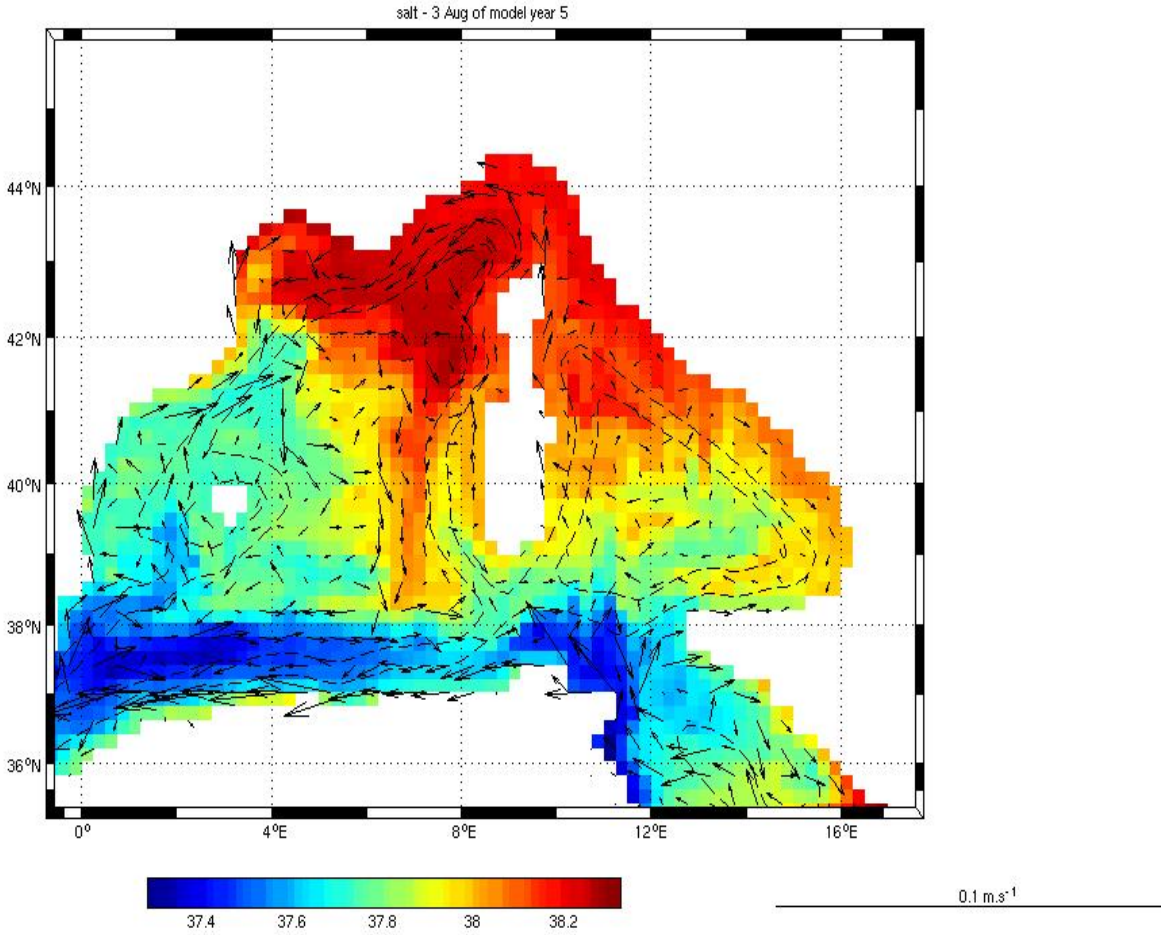


Salinité et champs de vitesse à -160 mètres de profondeur en février de la dernière année de simulation

La salinité est plus importante en surface au Nord de ce bassin. Ce phénomène correspond à celui observé chez Ahumada et Cruzado. Par contre en profondeur chaque partie de la Méditerranée à sa propre salinité de part l'écoulement des eaux et de leurs provenances.

Nous pouvons mettre également en évidence le fait que la mer Tyrrhénienne soit plus salée que le cœur de la Méditerranée nord occidentale. Nous pouvons peu être l'expliquer par le fait que ce soit une mer assez fermée.
De même que les eaux siciliennes vont être encore plus salées. Elles se situent plus au Sud, donc il peut y avoir plus d'évaporation.
Les salinités sont relativement en accord avec celle d'Ahumada et Cruzado. Exemple des données du mois de février à -30 mètres de profondeur : elles varient environ de 37.1 à 38. Et à -160 mètres de profondeur : elles varient environ de 37.7 à 38.4 (celles-ci correspondent un peu moins).

Nous aurions tendance à pensée que l'été la salinité devrait augmenter de part l'intensité solaire et évaporatoire croissante dans cette période.
Mais nous ne notons pas d'augmentation sur notre modèle (plutôt une l'inverse).



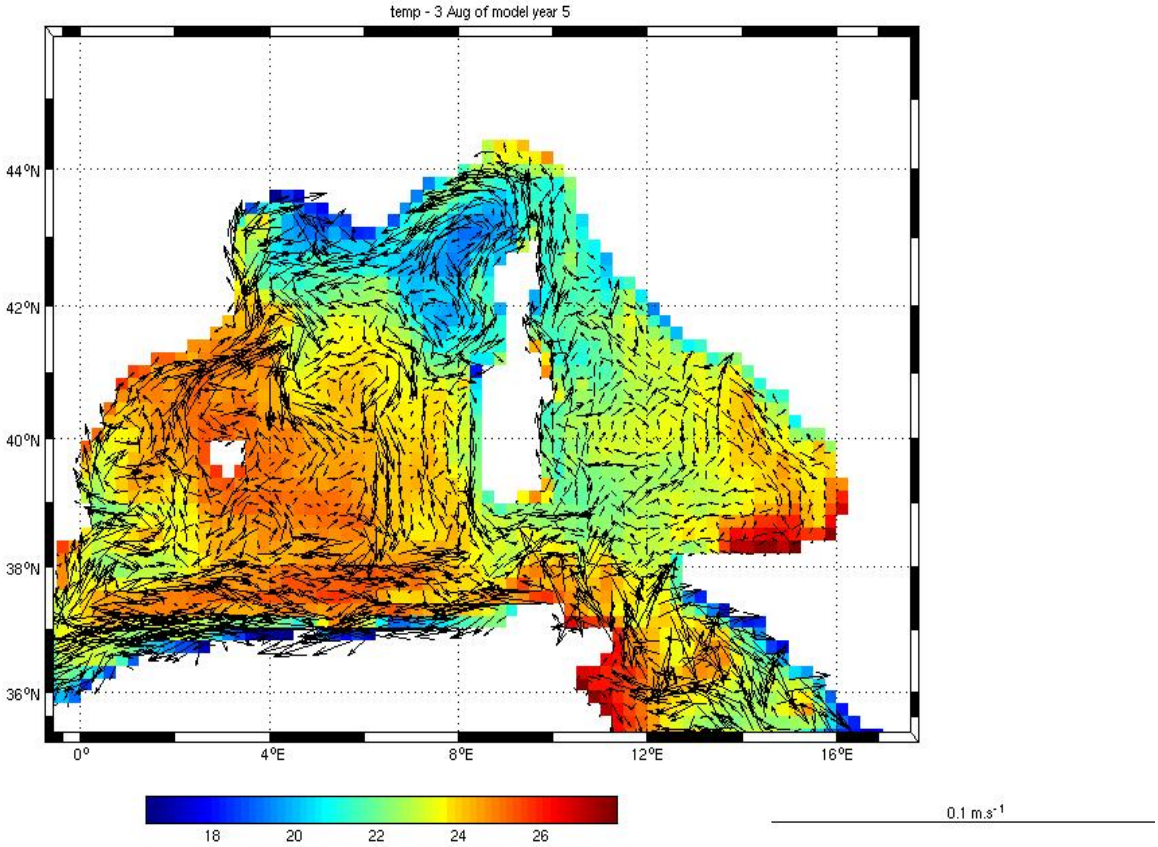
Salinité et champs de vitesse à -30 mètres de profondeur en Aout de la dernière année de simulation

Les données d'Ahumada et Cruzado, nous montrent une évolution de la salinité. En Aout la valeur de 37.8 (en jaune) en surface va dominer le nord du bassin, tandis qu'en profondeur cela va être de l'ordre des 38.3 (en orange). Par contre en novembre les salinités moins fortes vont envahir le bassin.
PS : mon modèle me pose un problème sur ce sujet, car la saisonnalité est bien représentée sur le 'plot_diags'.

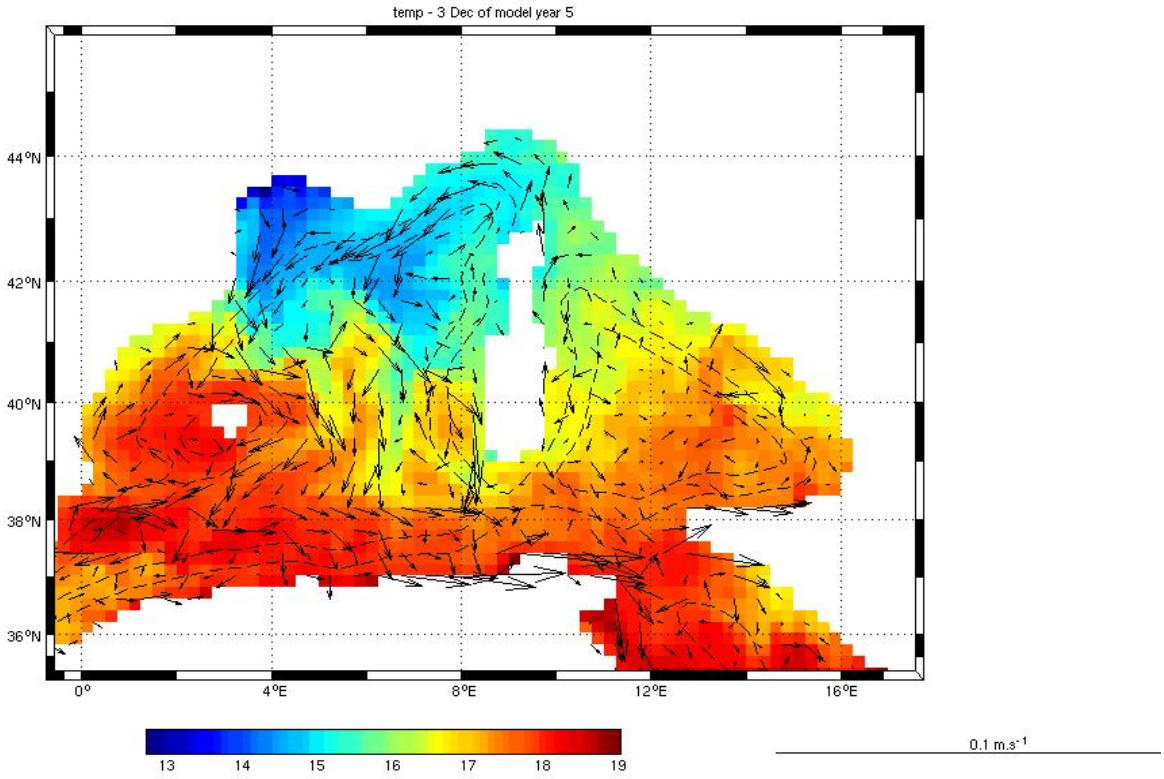
<u>La température</u> :

Concernant la température de surface, en hiver elle est de l'ordre de 13 à 19° C. Ce qui correspondrait plutôt bien au modèle de Millot et Taupier.
Les estivale sont de l'ordre de 17 à 28° C.
De plus elle est globalement plus importante au Sud.

Ce paramètre nous confirme la saisonnalité.



Température et champs de vitesse du mois d'aout et décembre de la dernière année de simulation



Les champs de vitesse :

D'après toutes nos simulations précédentes, nous remarquons que le sens global du courant est le même que celui des deux autres modèles.
Nous pouvons y voir : le courant Liguro Provençal Catalan provenant des eaux siciliennes, qui poursuit sont chemin par un large gyre cyclonique en revenant au dessus de la Corse, ainsi que le courant Corse qui contourne l'île de l'Est vers l'Ouest à une certaines profondeur, les Eaux Atlantiques Intermédiaires qui rentrent par le détroit de Gibraltar et longe les côtes Arabes.
Nous pouvons également distinguer plusieurs gyres, moins visibles que sur les autres modèles (celui d'Ahumada et Cruzado étant de plus haute résolution que le notre) : les gyres anticycloniques prés des côtes africaines.
Par contre nous n'apercevons pas ceux de la mer Tyrrhénienne présents sur les autres modèles.

Le modèle d'Ahumada et Cruzado, montre que le courant descend des deux côtés de la Corse à des profondeurs comprises entre 0 et environ 350m.
Puis à -400 mètres il descend du côté droit de l'île et remonte sur son côté gauche. Ce qui confirme nos simulations.

Par contre ce qui ce passe à gauche des îles Baléares et également du côté Est (mer Tyrrhénienne et eaux siciliennes) de notre modèle n'est pas clair et constitue surement un artéfact.

2. Conclusion

Pour conclure ce projet, nous allons pouvoir dire que notre modèle retrace bien les circulations et paramètres physiques réels et saisonniers dans l'ensemble.
Nous retrouvons le large gyre cyclonique au Nord du bassin (courant Liguro provençal Catalan), une circulation instable sur les pentes de fonds.
L'intensité du vent va être déterminante dans l'apparition de certains tourbillons comme celui de la mer Tyrrhénienne par exemple.
Cependant nous avons remarqué quelques disfonctionnement : la salinité saisonnière, certains champs de vitesse ainsi que la température élevée tout prés des côtes Sud de la Sicile, c'est un peu étrange.