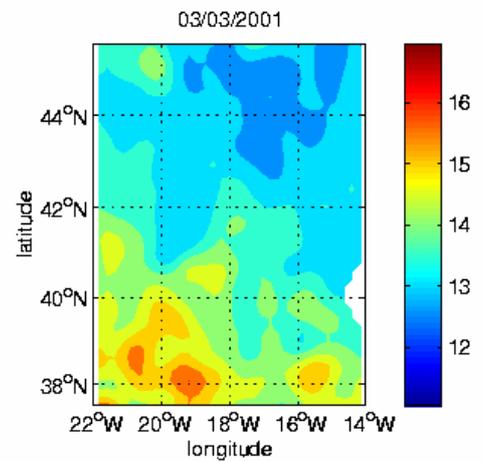
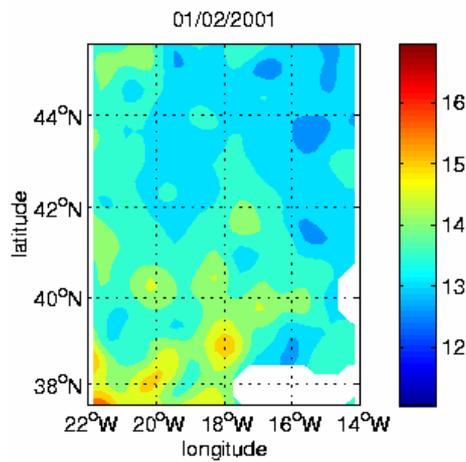


Inversion par filtre de Kalman

Version 2.4

Documentation utilisateur



Historique du document

Numéro de version du document	Date	Description
Version 1.0	Avril 2003	Documentation de la version 1.5 du modèle inverse avec filtre de Kalman
Version 2.0	Janvier 2004	Documentation de la version 2.0 du modèle inverse avec filtre de Kalman
Version 2.1	Mars 2004	KA V2.1
Version 2.4	Décembre 2004	KA V2.4

SOMMAIRE

1. Introduction	5
2. Rappel sur la méthode	6
2.1. Les données	6
2.1.1. Les profils.....	6
2.1.2. Les mesures eulériennes :.....	6
2.1.3. Les mesures lagrangiennes.....	6
2.1.4. Les données issues d'une analyse	6
2.2. Le champ à estimer	7
2.3. Principe du filtre de Kalman:	7
2.3.1. Etape d'analyse:	7
2.3.2. Etape de prévision:	7
3. Préparation des données	8
3.1. Fichiers associés à la décomposition verticale	8
3.1.1. Fichier moyen.....	8
3.1.2. Fichier des modes.....	9
3.1.3. Fichier de pondération.....	9
3.1.4. Fichiers dérivés	10
3.2. Fichiers de données	10
3.2.1. Profils de température et de salinité	11
3.2.2. Séries temporelles de température et de salinité en un point	11
3.2.3. Séries temporelles de données de courant en un point.....	11
3.2.4. Données de courant fournies sur une grille.....	12
3.2.5. Données de courant issues de l'ADCP de coque	14
3.2.6. Données vitesses et température issues de flotteurs.....	14
3.2.7. Données de température de surface.....	14
4. L'inversion	16
4.1. Organisation du programme	16
4.1.1. Initialisation.....	16
4.1.2. Boucle sur les jours	17
4.2. Arborescence recommandée	18
4.3. Le fichier de configuration de l'inversion (xxx-cfg.asc)	18
4.4. Les fichiers en sortie de l'inversion	25
4.4.1. Les listing de contrôle	26
4.4.2. Le fichier statistique	26
4.4.3. Le fichier 'données'	26
4.4.4. Les fichiers de résultats dans l'espace spectral	27
4.4.5. Les fichiers des champs dans l'espace physique.....	27
4.4.6. Les fichiers des cartes d'erreur aux niveaux témoins	27
4.4.7. Le fichier de reprise	27
4.4.8. Le fichier matrice de covariance	28
4.5. Lancement des programmes	28
4.5.1. Généralités.....	28

4.5.2.	Inversion.....	30
4.6.	Améliorations prévues	31
5.	Analyse des résultats	32
5.1.	Partie réservée	32
5.1.1.	Initialisation des fichiers	33
5.1.2.	Création de fichiers NetCDF	33
5.1.3.	Création d'animation.....	34
5.2.	Partie publique	35
5.2.1.	Initialisation des fichiers	35
5.2.2.	Tracé des profils moyens et modes	36
5.2.3.	Statistiques de l'inversion	36
5.2.4.	Visualisations des champs.....	36
5.2.5.	Comparaison aux données.....	37
5.2.6.	diagnostics.....	38
5.2.7.	Exemples de tracés	39
5.3.	Améliorations prévues	44
6.	Annexes.....	47
6.1.	Format NetCDF des fichiers de données.....	47
6.1.1.	Fichier Profileur	47
6.1.2.	Format des fichiers NetCDF flotteurs	49
6.1.3.	Format des fichiers NetCDF mouillages TS	50
6.1.4.	Format des fichiers NetCDF mouillages UV	51
6.1.5.	Format des fichiers NetCDF MADCP	52
6.1.6.	Format des fichiers NetCDF soprane	53
6.1.7.	Format des fichiers NetCDF ADCP de coque	55
6.1.8.	Format des fichiers NetCDF température de surface	57
6.2.	Exemple d'arborescence pour l'inversion	58

1. Introduction

Ce document constitue le manuel utilisateur pour le logiciel d'inversion avec filtre de kalman. Il :

- précise les fichiers en entrée/sortie de l'inversion
- indique comment lancer le programme d'inversion proprement dit (*inkalnl_mp_V24*)
- explique comment visualiser et exploiter les résultats issus de ce programme.

Pour plus d'information quant au principe mis en œuvre par le programme d'inversion, l'utilisateur peut se référer à la documentation 'Analyses de champs tridimensionnels : Description des méthodes mises en œuvre'.

Le document est associé à la version 2.4 du logiciel développée en novembre 2004. Par rapport à la version précédente, la principale différence réside dans le fait que le calcul des paramètres (température, salinité, ...etc) ne s'effectue plus sur le calculateur nymphéa. Ces paramètres sont désormais calculés sur station unix/linux via le progiciel matlab.

2. Rappel sur la méthode

Lors de campagnes d'observation intensives, telles que POMME par exemple, on met en oeuvre un grand nombre d'instruments de mesures, afin de suivre l'évolution des variables caractéristiques et d'en capturer les différentes échelles. On dispose en parallèle de mesures d'opportunité (satellites, expériences voisines, océanographie opérationnelle). Se pose alors la question de produire une synthèse de ces données qui rende plus faciles les calculs de bilan et l'interprétation des jeux de données pris individuellement. Les méthodes d'estimation optimale permettent la combinaison d'un ensemble d'informations de nature différente telles que :

- Les observations
- La relation entre ces observations et le champ à estimer (matrice d'observation)
- Les connaissances statistiques sur les champ à estimer et les données (matrices de covariance)
- Une connaissance a priori du champ, qui peut impliquer une équation d'évolution plus ou moins complexe.

2.1. Les données

Les données peuvent être réparties en quatre grandes catégories :

2.1.1. Les profils

- Profils de température et salinité (CTD, XBT, XCTD, profileurs dérivants)
- Profils de courant suivant la route navire (VMADCP)

2.1.2. Les mesures eulériennes :

- Courantomètres ponctuels
- Courantomètres Doppler
- Capteurs de température
- Capteurs de salinité

2.1.3. Les mesures lagrangiennes

- Bouées de surface
- Flotteurs profonds

2.1.4. Les données issues d'une analyse

- Champs de courant issus d'une analyse de l'altimétrie
- Température de surface

2.2. Le champ à estimer

Dans la terminologie de l'estimation, le champ à estimer est appelé variable d'état. Nous avons choisi d'estimer les trois variables: température, salinité et courant. Les champs tridimensionnels associés à chacune d'elles sont projetés sur des modes verticaux. Le courant est supposé géostrophique, il est relié à la densité et dérive d'une fonction courant. La composante horizontale des champs est projetée sur un nombre fini de composantes de Fourier. Cette décomposition présente les avantages suivants :

- Un filtrage naturel des petites échelles par la troncature à la fréquence de coupure
- La prise en compte de chaque donnée à sa position exacte sans interpolation
- Le calcul simple et précis, car effectué dans l'espace de Fourier, des quantités dérivées telles que la vorticité ou les gradients de température.

2.3. Principe du filtre de Kalman:

2.3.1. Etape d'analyse:

Chaque observation (y) est reliée aux variables du champ par une relation linéaire. Un terme d'erreur représente à la fois l'erreur de mesure et l'erreur due à la représentation simplifiée :

La connaissance d'une ébauche (x^f) permet d'extraire des données la partie nouvelle, appelée innovation (d). La solution (x^a), et la matrice covariances correspondante (P^a) sont alors donnée par les formules suivantes :

$$\begin{aligned}x_i^a &= x_i^f + K_i d_i \\P_i^a &= (I - K_i H_i) P_i^f \\d_i &= y_i^0 - H_i x_i^f \\K_i &= P_i^f H_i^T (H_i P_i^f H_i^T + R_i)^{-1}\end{aligned}$$

2.3.2. Etape de prévision:

Lorsque l'on analyse le champ à différents pas de temps, l'analyse précédente, accompagnée de la matrice de covariance permettent de définir l'ébauche si l'on dispose de la matrice d'évolution du champ (M) :

$$\begin{aligned}x_i^f &= M_{i-1} x_{i-1}^a \\P_i^f &= M_{i-1} P_{i-1}^a M_{i-1}^T + Q_{i-1}\end{aligned}$$

3. Préparation des données

Le programme d'inversion avec filtre de kalman requiert 2 types de fichiers de données: un premier type décrit la décomposition verticale : profil moyen et modes, le second type contient l'ensemble des observations, regroupées par catégorie. La préparation de ces données est effectuée sur station de travail UNIX/LINUX, avec un ensemble de scripts Matlab.

3.1. Fichiers associés à la décomposition verticale

Les fichiers décrivant la décomposition verticale en modes sont:

- Le fichier moyen
- Les fichiers des modes
- Le fichier de pondération
- Le fichier moyen dérivé
- Les fichiers des dérivées des modes

Tous ces fichiers préparés pour les analyses POMME se trouvent dans le répertoire :

KA_MODES = '/home1/doelana/POMME/KALMAN/modes/reference'

Ils ont été construits à partir d'un ensemble de mesures CTD de la région. Les profils utilisés sont regroupés dans un fichier binaire matlab :

\$(KA_MODES)/profils_utiles.mat .

Une relation T-S de probabilité maximale a été calculée à chaque niveau à partir de cette base de données, elle est conservée dans le fichiers :

\$(KA_MODES)/TS_prob.mat .

3.1.1. Fichier moyen

L'utilisateur doit créer un fichier contenant les profils moyens des paramètres température, salinité, densité, ...etc. Pour ce faire, il sélectionne un ensemble de profils dans la région à inverser et en déduit la moyenne. Le fichier résultat doit avoir l'extension '**.dat**' et la structure suivante :

- La 1ère ligne contient le nombre de ligne du fichier
- Chaque ligne suivante comprend :
 - la profondeur (en m)
 - la vitesse moyenne du son (m/s)
 - la température moyenne (°)
 - la salinité moyenne
 - l'anomalie de densité in-situ moyenne
 - la température potentielle moyenne
 - la fonction $\alpha(z)$, opposée du rapport des compressibilités thermiques et halines, qui permet de convertir un mode de température en un mode de salinité neutre
 - la fonction de compressibilité haline ($d\rho/ds$)
 - la fréquence de Brunt-Vaisala moyenne (N^2)
 - N/f moyen

Pour l'inversion proprement dite, seules les informations de profondeur, de température, de salinité et de $\alpha(z)$ sont utilisées. Pour générer le fichier moyen, on peut s'inspirer du script matlab : *hyd_bsv_moy.m* .

3.1.2. Fichier des modes

L'inversion étant basée sur les modes verticaux en température, salinité et fonction courant, l'utilisateur doit préalablement constituer sa propre base de modes. Quand on parle de modes verticaux, il s'agit en fait de modes verticaux d'anomalie (de température, de salinité, ...etc.). Si on choisit d'utiliser des modes empiriques, l'utilisateur doit avoir à disposition :

- le fichier moyen précédemment présenté
- un ensemble de profils de température sur la région qui l'intéresse (région/zone d'inversion)

L'inversion nécessite 3 fichiers des modes :

- un fichier des modes de température
- un fichier des modes de salinité
- un fichier des modes de la fonction courant

Chaque fichier a le format suivant :

- 1 ligne comprenant le nombre de ligne de données du fichier, chaque ligne étant associée à une profondeur du fichier moyen
- 1 ligne précisant le nombre de colonne du fichier, chaque colonne correspondant à un mode
- 1 ligne indiquant la norme des modes. Ceux-ci doivent être normés (donc norme égale à 1)
- les lignes restantes contiennent la valeur de chacun des modes

Les modes sont nécessairement fournis aux mêmes niveaux que le profil moyen.

Pour créer ses modes, l'utilisateur peut s'inspirer du script matlab : *hyd_bsv_eof.m* qui calcule des modes empiriques. Ce script permet :

- d'effectuer une pondération par profil (certains profils peuvent avoir un poids plus ou moins important par rapport aux autres)
- d'effectuer une pondération en fonction de l'épaisseur des couches
- d'effectuer une pondération par variance du niveau

L'utilisateur doit déterminer le nombre de modes verticaux qu'il souhaite conserver. Pour ce faire, il peut s'appuyer sur le graphique représentant le pourcentage d'énergie totale cumulé en fonction du nombre de modes.

Il est aussi possible d'utiliser d'autres types de modes (dynamiques, ...etc.).

3.1.3. Fichier de pondération

Le fichier de pondération définit le produit scalaire pour lequel les modes sont orthogonaux. Il permet de prendre en compte dans l'inversion les profils de température sous forme de modes. La projection d'un profil de température $T_i(z)$ sur le mode k est définie par : $\mu_{ki} = (T_i(z), w_k^2(z)L_k^T(z))$ où $(..)$ désigne le produit scalaire canonique.

Ce fichier est créé par le script de calcul des modes (*hyd_bsv_eof.m*) . Le format de ce fichier est le suivant :

- 1 ligne pour chaque profondeur du fichier moyen. Chaque ligne comporte :
 - le coefficient de pondération utilisé par le calcul de détermination des modes pour la température

- le coefficient de pondération utilisé par le calcul de détermination des modes pour la salinité
- les valeurs propres pour la température
- les valeurs propres pour la salinité
- l'écart-type pour la température
- l'écart-type pour la salinité

A l'heure actuelle, seules les données de ce fichier associées à la température sont utilisées (cf. 3.2.1).

3.1.4. Fichiers dérivés

L'utilisateur doit créer le fichier de dérivées verticales correspondant aux profils moyens et aux modes. Le format des fichiers des dérivées est identique au format des fichiers présentés précédemment. Pour les créer, l'utilisateur peut s'inspirer du script matlab : *hyd_bsv_eofderiv.m*. A noter que ces fichiers ne sont plus utilisés mais le code les lit quand même.

3.2. Fichiers de données

Les données pouvant être prises en compte par l'inversion avec filtre de kalman sont de 7 types :

1. Profils de température et ou de salinité
2. Séries temporelles de température et ou de salinité en un point (mouillage)
3. Séries temporelles de vitesses de courant (U et V) en un point courantomètre sur mouillage et/ou M-ADCP)
4. Vitesses de courant (U et V) issues de données d'ADCP de coque
5. Vitesses de courant (U et V) fournies sur une grille (soprane)
6. Températures de surface fournies sur une grille
7. Données de vitesses et température issues de flotteurs

Pour l'inversion, excepté pour les données de soprane et de température de surface, les données ont été filtrées sur 48h et moyennées sur 24h de façon à obtenir une donnée par position/par jour/par profondeur.

Tous les fichiers de données qui ont permis de générer les fichiers NetCDF utiles à l'inversion sont référencés sous le répertoire :

KA_DATA = /home1/doelanaival/POMME/KALMAN/data.V2.

Ce répertoire comprend un sous-répertoire par type de données :

- profileurs
- mouillages
- ADCP de coque
- soprane
- température de surface
- flotteurs

Chacun de ces sous-répertoires contient :

- un répertoire associé aux données originales
- un répertoire associé aux fichiers de données NetCDF quotidiens utiles à l'inversion

- un répertoire associé aux programmes permettant de créer les fichiers NetCDF en entrée de l'inversion à partir des fichiers de données originales
- un répertoire associé aux tracés

Pour pouvoir utiliser les fichiers NetCDF dans l'inversion, il est impératif que les fichiers soient regroupés par type de donnée et par année (cf. 4.3 + annexes).

3.2.1. Profils de température et de salinité

Les données de température et de salinité prises en compte par le modèle inverse avec filtre de kalman sont des données issues de tout type de profileur (CTD, XBT, flotteurs, ...etc). Les données brutes profileurs sont fournies en temps réel par le centre de données Coriolis et mises à disposition 1 fois par semaine. Il y a 1 fichier par type de profil. Ces données brutes sont interpolées sur la grille régulière : $[[0 :5 :400] [410 :10 :2000] [2020 :20 :6600]]$. Pour plus de détails sur ces fichiers, l'utilisateur peut se référer à la documentation « Coriolis Data-Center – External Interfaces ». A partir de ces fichiers NetCDF interpolés, l'utilisateur doit créer un fichier par jour et par type de profileur. Les fichiers ainsi créés comprennent les données de température et, le cas échéant, les données de salinité. A noter que pour certains profileurs (PF), lorsque la salinité était absente, on a déterminé une pseudo-salinité basée sur la relation T-S probable du fichier *TS_prog.mat* .

Les fichiers ainsi créés seront en entrée du modèle inverse.

La nomenclature de ces fichiers est : **XY_profil-yaaaammmdjj.nc** avec :

- XY : type de profileur
 - CT : CTD
 - TE : Tesac
 - PF : flotteur
 - ... etc.
- aaaa : année
- mm : mois
- jj : jour

Ainsi, le fichier PF_profil-y2001m02d14.nc est le fichier NetCDF associé aux données flotteur pour le 14/02/2001.

Il est à noter que les profils de température peuvent être pris par l'inversion sous 2 formes différentes :

- 1 fois sous forme de couche
- 1 fois sous forme de modes pour les profils atteignant une certaine profondeur fixée par l'utilisateur

Les profils de salinité sont pris en compte uniquement sous forme de couche.

3.2.2. Séries temporelles de température et de salinité en un point

Ces données de température et de salinité sont issues de mouillage. L'inversion nécessite la création d'un fichier NetCDF par jour regroupant les données de température et salinité de tous les mouillages associés à la région à inverser.

Le 25/11/2003, un nouveau jeu de données des mouillages POMME a été récupéré ; les fichiers NetCDF ont été régénérés. Lorsque la température est disponible et la salinité absente, une pseudo-salinité a été calculée.

3.2.3. Séries temporelles de données de courant en un point

Ces données de vitesses de courant sont issues de mouillage. Elles peuvent provenir soit de plusieurs courantomètres à mesures ponctuelles, type Anderaa, soit de profileurs (ADCP

Mouillage). L'inversion requiert la création d'un fichier NetCDF par jour regroupant les données de vitesses de tous les mouillages relatifs à la région à inverser. Néanmoins, on sépare les données issues de mouillage des données issues d'ADCP mouillages. Les fichiers NetCDF utiles à l'inversion ont été créés suite à la récupération le 25/11/2003 d'un nouveau jeu de données.

3.2.4. Données de courant fournies sur une grille

Le modèle inverse peut prendre en compte des données de courant (U, V) à un niveau sur une grille. Ce sont typiquement les données issues de l'assimilation de l'altimétrie. Pour les analyses Pomme, nous avons utilisé le premier niveau des analyses SOPRANE.

SOPRANE est un système de prévision océanique de l'état tridimensionnel de l'océan en Atlantique Nord-Est. Il est basé sur l'assimilation de données altimétriques dans un modèle quasi-géostrophique d'océan. Chaque semaine, SOPRANE délivre des analyses de la situation océanique et des prévisions jusqu'à 2 semaines. Ci-après, un exemple de la fonction courant à 100m fournie par SOPRANE

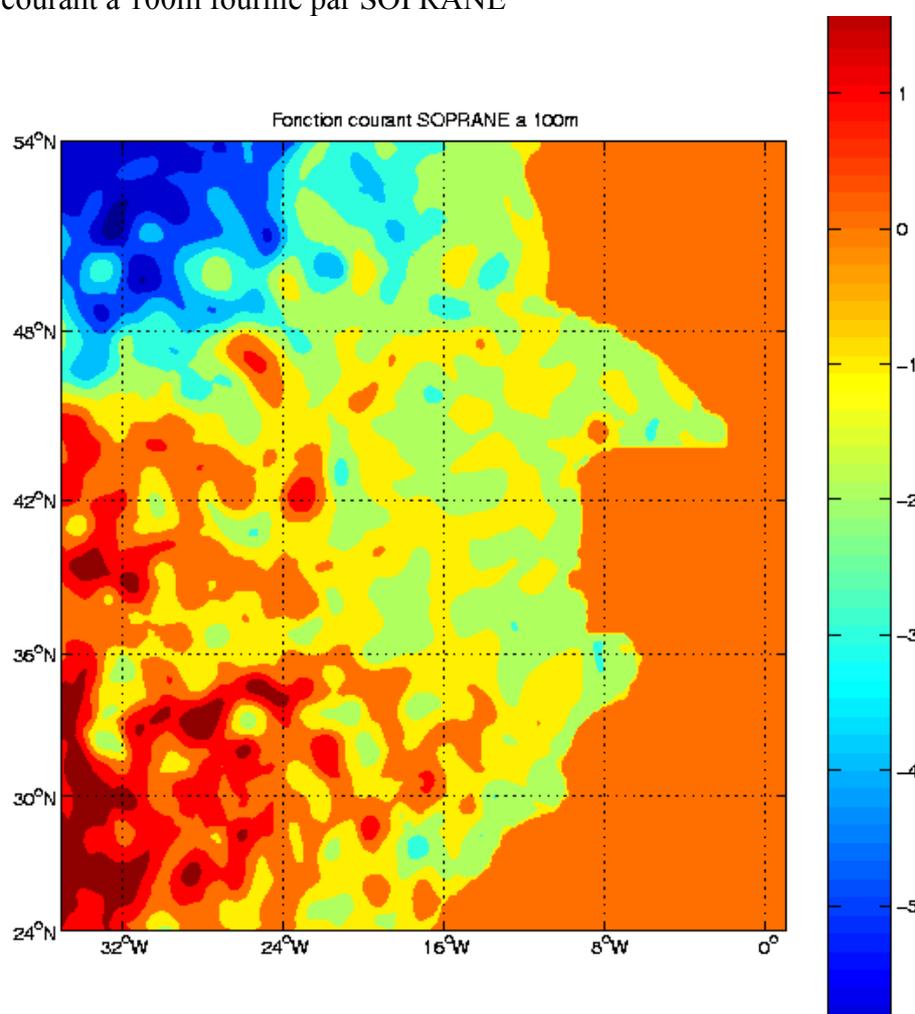


Figure 1 : Fonction courant fournie par Soprane (échelle en $10^{-4} \text{ m}^2 \text{ s}^{-1}$).

Pour créer les fichiers NetCDF SOPRANE qui seront utilisés dans le modèle inverse avec filtre de kalman, on part de :

- fichiers NetCDF fournis par SOPRANE comprenant :
 - Longitude, Latitude. La résolution horizontale de SOPRANE est de $1/10^{\text{ième}}$ de degré.

- Pour chacun des points de la grille définis par ces longitudes et latitudes, le champ de courant (en m^2/s) pour les profondeurs :
 - 100m
 - 325m
 - 550m
 - 750m
 - 1000m
 - 1325m
 - 1775m
 - 2375m
 - 3200m
 - 4600m
- fichiers ASCII (MSLA) de l'altimétrie associés comportant :
 - l'anomalie de la hauteur de la mer (en mm)
 - l'erreur correspondante (en %)
 La résolution horizontale de ce fichier est de $\frac{1}{4}$ de degré.

A partir de ces données, on crée un fichier **NetCDF** par jour de données SOPRANE (1 fichier par semaine) contenant :

- Longitude
- Latitude
- Vitesse horizontale U (en m/s)
- Vitesse méridienne V (en m/s)
- Erreur associée (en %) (issue de l'altimétrie)
- Profondeur associée

Les vitesses U et V sont données par :

$$U = -\partial\Psi / \partial y \text{ et } V = +\partial\Psi / \partial x$$

Elles sont calculées en chaque point de la grille SOPRANE comme des dérivées centrées.

A chaque point de données, on associe une erreur de vitesse déduite de l'erreur MSLA (donnée en %) selon le tableau ci-après.

Erreur MSLA (%)	Erreur de vitesse (cm/s)
0-5	2.5
5-10	3.0
10-20	3.5
20-30	4.0
30-40	4.5
40-50	5.0
>50	10.0

A chaque point SOPRANE, on affecte l'erreur MSLA géographiquement la plus proche. Au pire, on effectue donc une erreur de $\frac{1}{4}$ de degré (résolution des fichiers ASCII MSLA).

La création de ces fichiers s'effectue via le script matlab : **soprane_02.m** .

La nomenclature des fichiers résultat est : **sopr02_yaaaammdjj.nc** avec :

- aaaa : année
- mm : mois

- jj : jour

Ainsi, le fichier `sopr02_y2000m02d14.nc` est le fichier Netcdf associé aux données soprane pour le 14/02/2000.

En l'état actuel, **l'inversion ne prend en compte que les données soprane relatives au premier niveau, c'est-à-dire au niveau 100m dans notre cas.**

3.2.5. Données de courant issues de l'ADCP de coque

Les données de courant issues de l'ADCP de coque permettent d'insérer dans le modèle inverse des profils de données de courant.

En effet, les ADCPs de coque fournissent des profils de courant allant de la surface à une certaine profondeur (800m pour un ADCP 75 Khz) sur toute la trajectoire du navire où l'ADCP est installé. Au LPO, ces données sont traitées via le logiciel CASCADE (cf. « CASCADE : Logiciel de traitement des données d'ADCP de Coque Version 3.0 : DRO/LPO 02-03 »).

Les données d'ADCP de coque prises en compte par le modèle inverse sont issues d'un fichier campagne ADCP NetCDF généré par CASCADE. Ce fichier comprend les vitesses U et V (en cm/s) pour la région traitée, lissée horizontalement et verticalement ($\frac{1}{4}$, $\frac{1}{2}$, $\frac{1}{4}$).

Ces données ont ensuite subi 3 traitements.

1. Création d'un fichier filtré (*ka_vamdcpl_filt.m*) : Les vitesses sont moyennées par bins espacés de 10 km, les données situées de part et d'autre à 2*10km sont prises en compte. Les données différant de plus de 2.7 écart types sont éliminées. L'erreur est ré-évaluée.
2. Nettoyage de données aberrantes à partir d'une liste noire (*ka_vamdcpl_clean.m*)
3. Création des fichiers quotidiens multicampagnes (*ka_vamdcpl_daily.m*) . Les données sont mises sur niveaux standards (pas = 20 m), un filtrage vertical est appliqué (butterworth d'ordre 2, fréquence de coupure : $\frac{1}{4}$)

A partir de ces données, on crée un fichier ADCP NetCDF par jour de données contenant :

- Z : profondeur (en m)
- Longitude
- Latitude
- Date
- Campagne d'origine
- Vitesse zonale U (en m/s)
- Vitesse méridienne V (en m/s)
- Erreur (m/s)

3.2.6. Données vitesses et température issues de flotteurs

Toutes les données flotteurs ont été récupérées. Pour les flotteurs SEASCAN, dans les fichiers originaux, la pression a été estimée à une constante avec une précision de +/- 100m. Aussi, les fichiers NetCDF quotidiens créés pour l'inversion ne contiennent pas la température associée à ces flotteurs. Les fichiers créés quotidiennement comportent les données de température et de vitesses associés à tous les flotteurs relatifs à la région à inverser.

3.2.7. Données de température de surface

On a récupéré les données de température de surface de G. Caniaux. Pour l'inversion, on garde 1 point sur 2 en longitude et latitude. Pour ne pas exploser le nombre de données à prendre en compte par l'inversion pour un jour donné, on crée 2 fichiers de température de

surface par semaine; aucun des 2 ne correspondant à un jour pour lequel les données soprane sont disponibles. On force l'erreur à 0.02 degré. On crée donc :

- 1 fichier NetCDF de température de surface pour chaque jour suivant un jour associé à des données soprane
- 1 fichier NetCDF de température de surface 4 jours après

4. L'inversion

Le programme d'inversion est mis en œuvre sur la machine du pôle de calcul Brestois (NYMPHEA).

4.1. Organisation du programme

```
program inkaln1
```

4.1.1. Initialisation

ini1 : Définition du run

call read_rundef () : Définit les dates de début et fin d'analyse et le fichier de reprise.

call invini () : Définit la décomposition verticale et horizontale, les covariances spatiales et temporelle, et construit la liste des données disponibles. 14 types de données sont prises en compte.

```
txt_data(1) = 'Profils temperature - couches  '
txt_data(2) = 'Profils salinité      - couches  '
txt_data(3) = 'Profils temperature - modes    '
txt_data(4) = 'Profils salinité      - modes    '
txt_data(5) = 'Champs UV un niveau          '
txt_data(6) = 'Profils VM-ADCP UV           '
txt_data(7) = 'Profils M-ADCP UV           '
txt_data(8) = 'Temperature mouillages      '
txt_data(9) = 'Salinité mouillages         '
txt_data(10) = 'UV mouillages              '
txt_data(11) = 'Temperature flotteurs      '
txt_data(12) = 'UV flotteurs               '
txt_data(13) = 'Temperature de surface     '
txt_data(14) = 'Salinité (pseudo) de surface '
```

ini2 : Définition de l'état climatologique et de l'état initial

call lit_reprise ()

Ini3 : Préparation des calculs d'erreur on-line

La matrice de covariance d'erreur qui apparaît dans les formule est définie dans l'espace de Fourier. Pour contruire la carte d'erreur il faut repasser dans l'espace physique. Etant donnés, la taille de la matrice et la lourdeur du calcul, la matrice n'est pas sauvée et la calcul de la carte d'erreur est effectué 'on-line', à deux niveaux choisis (ztrac1 = 100m, ztrac2 = 1000 m), pour les variables température, salinité et fonction courant. La fonction suivant initialise ces calculs :

call errini ()

4.1.2. Boucle sur les jours

loop1: Construction de l'état a-priori

call kalmpr(): Construit le vecteur d'état a priori et matrice de covariance: **X_f** , **P_f**

loop2 : Inventaire des données du jour

Un ensemble de sous-programmes construisent les blocs de la matrice d'observation, et la portion des vecteurs données et erreur :

```
H_obs_i(1:nb_dat(i),1:mxcoej), data_vec(), data_err(), (i = 1, 14)
  nb_dat(1) = nprof_T
  nb_dat(2) = nprof_S
  nb_dat(3) = nprof_Tctd
  nb_dat(4) = nprof_Sctd
  nb_dat(5) = nprof_soprane
  nb_dat(6) = nprof_adcp
  nb_dat(7) = nprof_m_adcp
  nb_dat(8) = nprof_m_T
  nb_dat(9) = nprof_m_S
  nb_dat(10) = nprof_m_UV
  nb_dat(11) = nprof_float_T
  nb_dat(12) = nprof_float_UV
  nb_dat(13) = nprof_tsurf
  nb_dat(14) = nprof_ssurf
```

loop3 : Construction de la matrice H_obs et du vecteur données

call cbr_cal(): Place les différents blocs pour construire la matrice H

loop4 : Calcul de la solution inverse (espace de Fourier)

call invgl_sol () : Calcule l'innovation, la solution et le résidu

loop5 : Calcul de matrice de covariance Cpps

call invgl_err () : Calcule la resolution (K*H), et la covariance Pa

loop6 : Calcul des champs dans l'espace physique

call calcul_phi_derivees(): Calcule la composante horizontale des champs dans l'espace physiques ainsi que leurs dérivées premières, seconde et troisièmes.

loop7 : Ecriture des résultats

call sav_dat () : Sauve les données, innovation, erreur et résidus

call savvar (xjour): Sauve la diagonale de la matrice de covariance

call sav_phi () : Sauve la composante horizontale des champs

loop8 : Calculs optionnels

Tous les 7 jours, 3 champs, 2 niveaux :

```

if (cal_err .and. idat .eq. idat_cal_err) then
  call errcal ()
endif

```

Exceptionnel : sauve les matrices d'erreur complètes pour 1 jour donné

```

if (ermat .and. (idat .eq. num_inv_sav)) then
  call sav_covar ()
endif

```

Sauve le dernier jour pour reprise:

```

if (idat .eq. ndatinv) then
  call sav_reprise ()
endif

```

4.2. Arborescence recommandée

Pour le bon déroulement de l'inversion, il est recommandé de créer un répertoire de travail et d'y définir les sous-répertoires suivants:

- Un sous-répertoire '**config**' qui comprendra les fichiers de configuration d'extension '**-cfg.asc**' en entrée de l'inversion
- Un sous-répertoire '**modes**' contenant les divers modes ainsi que les profils moyens à utiliser pour l'inversion. Le nom de ce répertoire se retrouve dans le fichier de configuration pour la détermination du fichier moyen et des fichiers des modes.
- Un sous-répertoire '**data**' comprenant un sous-répertoire par type de données. Chacun de ces sous-répertoires doit contenir un répertoire par année contenant les fichiers NetCDF quotidiens relatifs aux données disponibles. Ces répertoires sont référencés dans le fichier de configuration dans la partie 'données'.
- Un sous-répertoire '**resu**' où les fichiers en sortie de l'inversion (et de calima) seront générés.

4.3. Le fichier de configuration de l'inversion (xxx-cfg.asc)

C'est un fichier ASCII d'extension '**-cfg.asc**' dont seules les lignes suivant une ligne ***data*** sont prises en compte. Néanmoins, l'outil de lecture ne testant pas de mots clé, l'ordre des informations à entrer, indiqué dans l'exemple suivant, doit être respecté.

```

=====
Fichier de configuration pour l'inversion KA-V2.0
RUN de test sans contrainte modes
=====

```

Dans ce fichier seules sont prises en compte les lignes qui suivent un ***data***

Attention:

l'ordre d'entree des parametres et le nombre par ligne doit etre respecte,
la lecture fortran permet d'ajouter des commentaires en fin de ligne.

```

=====

```

```

*****
*
*      Contrôle des sauvegardes et impressions :      *
*
*****

c * controle sortie listing (de 1 a 5, 1 impression minimale, 5 maximale)

*data* listng
2

c * Sauvegarde de la matrice de covariance des erreurs
c * -----
c * Compte tenu de sa taille, une seule matrice est sauvegardee
c *   ermat = .true. on sauve
c *   num_inv_a_sauver : Numero d'inversion auquel sauvegarder la matrice
c *   de covariance des erreurs

*data* ermat // num_inv_a_sauver
.false.
1

c * Sauvegarde des cartes d'erreur aux niveaux test:
c * -----
c * Ce calcul prend beaucoup de temps. L'utilisateur choisi donc:
c *   - s'il souhaite ou non la calculer (oui = .true.).
c *   - Si .true., le premier jour de calcul (relatif) et le pas
c *   ex: 1 7 : Le calcul de l'erreur s'effectue pour le premier jour
c *   d'inversion puis au jour 8, 15, ...etc.

*data* calcul erreur
.true.
  4 7
-100.0 -1000.0

*****
*
*      Periode a inverser:      *
*
*****

c * Jours selectionnes et points de reprise
c * -----
c * dat_ref: date de reference (généralement le 1 janvier de l'année de
c *   debut des analyses).
c *   Convention: Le 1 janvier de l'annee de reference devient le jour 0
c *   (comme dans beaucoup d'instruments: couranto, tomo ..)

*data* dat_ref // jrdeb, jrfin
01/01/2000

c * Fichier de climatologie
c * -----
c * j_climato:
c * nom_climato: nom du fichier si (j_climato > 0)
c * Le champ climatologique sert a rappeler la prediction.
c * Pour le moment le champ climatologique est lu comme un jour particulier
c * du fichier climato et reste le meme pour toute les pas de temps.
c * Evolution prevue:

```

```

c * Prise en compte d'une climatologie avec cycle annuel.

*data* j_climato
-9999
nom_climato

*****
*
*      Definition de la décomposition verticale:      *
*
*****

c * fichier moyen
c * fichier base verticale: (_T, _S, _F, _P)
c * nmbcl_temp, nmbcl_sali: nombre de modes de T, et de S_active

*data*      /DECVERT/
modes/ANE_pom3_moy
modes/ANE_pom3_eof_Z20
8 6

*****
*
*      Definition des covariances horizontales:      *
*      et des temps de memoire associées aux modes  *
*
*****

* nmod: nombre total de modes, y compris le barotrope
*      le programme verifie sa coherence avec : nmbcl_temp+nmbcl_sali+1
* tmod1: amplitude du premier mode de temperature
*      les modes de T sont normes et adimensionnels, tmod1 est donc
*      homogene a une temperature.
* (ech_hor_km(im), rapmod(im), xmmod(im), im = 1, nmod)
* ech_hor_km: echelle horizontale du mode im en km
*      cvhr(1,im) = ech_hor_km(im)*1000.0d0      = Ech_hor
* rapmod : dimension du mode im, l'amplitude du mode im sera donnee par:
*      cvhr(2,im) = tmod1*rapmod(im)/rapmod(2) = Ampli
*      Ech_hor et Ampli servent au calcul des variances a-priori par
*      nombre d'ondes.
*      tmod1 et rapmod(im) peuvent etre choisis en se basant
*      sur les valeurs propres du calcul des EOF
* xmmod(im): temps de memoire du mode im en jours
*      alpmem = exp(-(delta/xmmod(im))**2) ou dlta = datinv(idat) - xjmem

*data* /COVHOR/
15      (nmod)
5      (tmod1 = ampli mode T-1)
60. 0.15  12.0      (bt-0) 1 ech_hor_km(im), rapmod(im), xmmod(im)
60. 1.00  12.0      (bcl1) 2
60. 0.50  12.0      (bcl2) 3
60. 0.30  12.0      (bcl3) 4
60. 0.20  12.0      (bcl4) 5
60. 0.10  12.0      (bcl5) 6
60. 0.10  12.0      (bcl6) 7
60. 0.10  12.0      (bcl7) 8
60. 0.10  12.0      (bcl8) 9
60. 0.30  12.0      (bcl8+1) 10
60. 0.20  12.0      (bcl8+2) 11
60. 0.10  12.0      (bcl4+3) 12
60. 0.05  12.0      (bcl4+4) 13
60. 0.05  12.0      (bcl4+5) 14

```

```

c * Mise en oeuvre d'une prediction (filtre de Kalman)
c * -----
c *   kalman = .true. le filtre est actif
c *   pourcent = la matrice de covariance des erreurs de prediction (cpqq)
c *             est construite comme un pourcentage de la matrice de covariance
c *             de la climatologie (ccli): cpqq = (pourcent/100)**2 * ccli

*data* kalman // pourcent
.true.
3.0

*****
*
*   Definition de la décomposition horizontale
*
*****
c * xxlon0, ylat0
c * xrslidf  xxlobdf, xlindf
c * fact
c *
c * xxlon0, ylat0 : point origine (degres decimaux)
c * xrslidf : resolution demandee (km)
c * xxlobdf: taille boite d'observation (km) - c'est à dire la zone que
c *          l'on souhaite analyser
c * xlindf : taille boite d'inversion (km) - le champ est periodique sur
c *          la boite d'inversion (décomposition specttrale)
c * fact:     facteur multiplicatif de l'echelle de covariance
c *          sert à déterminer la zone de recherche des données.
c *          xlon_min = xlon0 + (xlobs/2) - fact*ech_hor_km(1)
c *          xlon_max = xlon0 + (xlobs/2) + fact*ech_hor_km(1)
c *
c * Le nombre d'harmoniques en x et en y prises en compte est calcule
comme
c * le plus grand nombre pair
c *   ninv <= min((xlin_df - 1.0)/xres_df, ninv_mx)
c * La resolution est ensuite ajustée et xlinv et xlobs sont recalculés.
c * Le nombre d'onde de troncature adimensionnel est ensuite defini comme:
c *   ktrnc = ninv/2 - 1
c *   L_onde min = xlinv/ktrnc - soit 2*xresol
c *

*data*
-18.0 41.5
40.0 1040.0 1280.0
1

```

```

*****
DONNEES
*****

*****
*
*   Donnees de type profil de T et/ou S
*
*****
c * Présence données flotteurs profilants:

*data* nb_nomprov/ (nom_lst_prov)
  5
data/BA
data/CT
data/PF
data/TE
data/XB

c * Présence données Analyse objective:

*data* nb_AO/ (nom_lst_prov)
  0

*   Quantification des erreurs
*   -----
*   ermod_prof_T_pct, ermod_prof_S_pct:
*   Erreur modele sur donnees T et S en couche (due a la projection)
*   exprimee en % de la RMS du niveau
*   er_coef_pct
*   Erreur sur les coefs des modes exprimee en % de l'amplitude a
piori des modes
*   profmin :
*   Profondeur minimale des profils pour les decrire en mode
*   (si 0 ==> pas de passage en mode).
*   (si >0 ==> passage en mode pour les profils > a profmin).

*data*
15 10      (ermod_prof_T_pct, ermod_prof_S_pct)
25        (er_coef_pct)          ! Contrainte des modes
1800

*   Moyennes sur des couches
*   -----
*   4 tranches sont definies par le pas (en m) et les indices
*   idef_ts(1:4,1:3)
*   ex: (i*10,i=0,4), (i*20,i=6,15), (i*50,i=7,30), (i*100,i=16,20)//
*data*
10  0  4
20  3 15
20 16 75
50 31 40

```

```

*****
*
*      Donnees SOPRANE
*
*****
! nb_soprane (0 ou 1)
! rep_soprane
! nom_soprane
! errmod_sop_U : erreur modele soprane sur U (en m/s)
! errmod_sop_V : erreur modele soprane sur V (en m/s)

*data* /SOPRANE/
1
data/soprane/
sopr02_
0.03 0.02

*****
*
*      Donnees VMADCP
*
*****
! nb_adcp (0 ou 1)
! nom_adcp
! errmod_adcp : erreur modele ADCP (en m/s)

*data* /VMADCP/
1
data/pom_adcp/
0.005

*****
*
*      Donnees M-ADCP
*
*****
! nb_m_adcp (0 ou 1)
! nom_madcp
! errmod_m_adcp : erreur modele ADCP (en m/s)

*data* /MADCP/
1
data/POM_m_ADCP/
0.005

*****
*
*      Donnees M-T T mouillage
*
*****
! nb_m_T (0 ou 1)
! nom_mT
! errmod_m_T : erreur modele T (en % de la valeur de T)

*data* /MTMP/
1
data/POM_m_TS/
10

```

```

*****
*
*      Donnees M-S S mouillage
*
*****
! nb_m_S      (0 ou 1)
! nom_mS
! errmod_m_S : erreur modele S  (en % de la valeur de S)

*data* /MSAL/
1
data/POM_m_TS/
5

*****
*
*      Donnees M-UV UV mouillage
*
*****
! nb_m_UV      (0 ou 1)
! nom_mUV
! epais_ch_mUV : E = epaisseur_muv (en m), on affectera la valeur
!                de mUV a la profondeur P a la couche P-E/2:P+E/2
! errmod_m_UV  : erreur modele UV  (en m/s)

*data* /MUV/
1
data/POM_m_UV/
40
0.005

*****
*
*      Donnees Temperature Flotteur
*
*****
! nb_float_T    (0 ou 1)
! nom_floatT
! errmod_float_T : erreur modele T  (en % de la valeur de T)

*data* /FLTMP/
1
data/POM_float/
10

*****
*
*      Donnees UV Flotteur
*
*****
! nb_float_UV    (0 ou 1)
! nom_floatUV
! epais_ch_floatUV : E = epaisseur_muv (en m), on affectera la valeur
!                de mUV a la profondeur P a la couche P-E/2:P+E/2
! errmod_float_UV : erreur modele UV  (en m/s)

*data* /FLUV/
1
data/POM_float/
60
0.01

```

```

*****
*
* Donnees Temperature de Surface.
*
*****
! nb_tsurf      (0 ou 1)
! rep_tsurf
! errmod_tsurf_T : erreur modele T (en % de la valeur de T)

*data* /TSURF/
1
data/POM_Tsurf/
5

*****
*
* Contrainte limite de vitesse a un niveau
*
*****
! nb_niv_ulim   (0 ou 1)
! z_ulim, err_ulim

*data* /ULIM/
1
2000 0.05

```

4.4. Les fichiers en sortie de l'inversion

Les fichiers générés par l'inversion se situent sous le sous-répertoire '**resu**' du répertoire de lancement. Ils sont de 2 types :

- ASCII
- Binaire 'unformatted': Il est à noter que le programme d'inversion est écrit en fortran90. Lors de l'écriture d'un enregistrement de données dans le fichier binaire, ce langage rajoute un enregistrement 'particulier' avant et après l'enregistrement de données. Ceci est à prendre en compte lors de la lecture sous matlab des fichiers binaires issus de l'inversion.

c Fichiers en sortie:

```

c -----
      exres   = '_ct1.lst'   ! fichier listing compte-rendu de resultats
      exctr   = '_ct2.lst'   ! fichier listing de verification du run
      expar   = '_da1.asc'   ! fichier parametres du run et statistiques
      exdat   = '_da2.bin'   ! Fichier données: position, valeur et résidus
      exinv   = '_fin.bin'   ! Fichiers espace de Fourier coefficients
                          ! et erreurs

      excer   = '_fer.bin'   !
      exphi   = '_phy.bin'   ! Fichiers espace physique champs horizontaux
      exier   = '_ie1.bin'   !
      exier2  = '_ie2.bin'   !
      excov   = '_mco.bin'   ! Matrice de covariance Cp_pr, Cp_ps
                          ! (pour test seulement)

```

Les fichiers résultats de l'inversion sont de 6 types présentés ci-après. Dans la suite, le nom du fichier résultat est donné pour un nom de fichier **nom_inv**.

4.4.1. Les listing de contrôle

- **nom_inv_ct1.lst** : Ce fichier ASCII contient un récapitulatif des paramètres de l'inversion ainsi que, pour chaque jour d'inversion :
 - des informations sur les résultats de l'inversion pour chaque type de données (Température en couche ou en CTD, Salinité en couche, VMADCP, Soprane, données mouillage)
 - erreur a priori
 - innovation
 - résidu et résidu sur bruit a posteriori
- **nom_inv_ct2.lst** : Ce fichier ASCII est un fichier de contrôle et comprend notamment :
 - la température et salinité moyenne
 - l'intégrale des modes sur les diverses couches définies par l'utilisateur via le fichier **_cfg.asc**.
 - des valeurs de la matrice cpps
 - des valeurs de la matrice dd
 - une indication sur le conditionnement
 - les mesures de temps passés dans diverses étapes de l'inversion

4.4.2. Le fichier statistique

- **nom_inv_da1.asc** : Ce fichier ASCII comprend
Les paramètres définissant la configuration :
 - Fichiers moyens et modes
 - Définition de la zone et résolution, nombre de modes
 - Amplitudes et échelles a-priori
 - Erreurs modèles ajoutées aux données

Et pour chaque jour d'inversion :

- le jour d'inversion
- le nombre de données prises en compte par type de données
- L'énergie pour chacun des modes
- La valeurs RMS des résidus

4.4.3. Le fichier 'données'

- **nom_inv_da2.bin**
Ce fichier binaire comprend, pour chaque jour de l'inversion et pour chaque type de données (Température en couche, Salinité en couche, Température en CTD, Salinité en CTD, Soprane, VMADCP) :
 - un texte indiquant le type de données en cours
 - le nombre de données associées prises en compte pour l'inversion
 - La position des données
 - les données
 - les erreurs a priori
 - les innovations associées

- les résidus associés

4.4.4. Les fichiers de résultats dans l'espace spectral

Ces fichiers binaires contiennent les informations sur les champs horizontaux dans l'espace spectral pour chaque jour d'inversion.

- **nom_inv_fin.bin** : Ce fichier contient :
 - un entête comportant :
 - la dimension du nombre d'ondes
 - le nombre de jours de l'inversion
 - le nombre de modes
 - le nombre d'ondes
 - le nombre de coefficients de Fourier
 - le nombre de coefficients de Fourier pour la zone d'observation
 - le décalage en X et Y de la fenêtre d'observation par rapport à la fenêtre d'inversion
 - les nombres d'ondes adimensionnels composante Est et composante Nord et les indices associés.

Cet entête sera utilisée pour le calcul de paramètre.

 - les coefficients de Fourier des champs horizontaux pour chaque mode.
- **nom_inv_fer.bin** : Matrice d'erreur

4.4.5. Les fichiers des champs dans l'espace physique

Ces fichiers contiennent les champs horizontaux pour chaque mode dans l'espace physique sur la zone d'observation.

- **nom_inv_phi.bin** : Ce fichier binaire comprend pour chaque jour de l'inversion et pour chaque mode :
 - le champ horizontal
 - toutes les dérivées premières, secondes et troisièmes de ce champ

4.4.6. Les fichiers des cartes d'erreur aux niveaux témoins

Ces fichiers comportent les erreurs sur : température, salinité et fonction courant, calculés dans l'espace physique et sommés sur les modes aux niveaux témoins (zerr1 et zerr2 du fichier de configuration).

- **nom_inv_ie1.bin** : Ce fichier binaire comprend les erreurs des champs Température, Salinité et Courant au premier niveau témoin à chaque pas de temps.
- **nom_inv_ie2.bin** : Ce fichier binaire comprend les erreurs des champs Température, Salinité et Courant au second niveau témoin à chaque pas de temps.

4.4.7. Le fichier de reprise

- **nom_inv.reprise** : Ce fichier binaire comprend les composantes de Fourier et la matrice cpps (a posteriori) pour le dernier jour de l'inversion. Ce fichier permet d'effectuer une inversion ultérieure à partir de ce fichier. Ainsi, l'utilisateur peut découper une grande inversion en petites si celle-ci nécessite plus de temps CPU que le temps CPU autorisé dans les batchs de nymphéa.

L'utilisateur peut lancer une première inversion du jour X1 au jour X2, puis une seconde inversion du jour (X2+1) au jour X3 en partant du fichier reprise de la première inversion. Ceci est équivalent à une inversion du jour X1 au jour X3

4.4.8. Le fichier matrice de covariance

- **nom_inv_mco.bin** : Ce fichier binaire comprend la matrice cpr complète et les nombres d'ondes pour le dernier jour d'inversion. La sauvegarde de ce fichier est optionnelle (car la matrice à sauver est très grosse).

4.5. Lancement des programmes

4.5.1. Généralités

L'exécutable lié à l'inversion avec filtre de kalman, situé sous le répertoire /home2/nympha/chemon/inv_kalman/KA_pomme/KA_V2.4 se nomme :

- inkalnl_mp_V24

La taille mémoire utile à l'exécutable inkalnl_mp_V24 dépend des paramètres présentés ci-dessous avec leur valeur actuelle. Ces paramètres sont définis dans le fichier invparam.h.

```
c =====
c Definition de tous les parametres du programme d'inversion
c =====

      implicit real*8      (a-h, o-z)
      implicit integer*4  (i-m)

c  lurstd = unite logique d'entree standard
c  luwstd = unite logique de sortie standard

      parameter (lurstd = 5, luwstd = 6)

c Parametres inversion
c -----
      parameter (
&   mxmodeT = 8      ! nombre max de modes de Temperature
& , mxmodeS = 6      ! nombre max de modes de salinite active
& , mxjour  = 366    ! nombre maximum de jours d'inversion
& , mxfou   = 18     ! nombre maximum d'harmoniques dans une direction
& , maxx_res = 2    ! facteur multiplicatif de la resolution
                        ! (pour la reconstruction dans l'espace physique)
& , mxcvhr  = 5      ! nombre de parametres caracterisant une covariance
                        ! horizontale
&           )

c parametres deduits des parametres inversion
c -----
c mxonds  = nombre max d'ondes compte tenu de la troncature circulaire
c mxfft   = nombre de composantes de fourier par direction
c mxcoef  = nombre de coefficients de Fourier (sin cos et constante)
c mxmbcl  = nombre de modes T+S,
c mxmode  = nombre total de modes verticaux

      parameter (
&   mxonds = mxfou*(2 + mxfou*1.5)
& , mxfft  = 2*mxfou
& , mxcoef = 2*mxonds + 1
& , mxmbcl = mxmodeT + mxmodeS
& , mxmode = mxmbcl + 1
&           )
```

```

!   Données:
!   -----
!   parameter (
&   mx_typ_dat = 14      ! nombre max. de types de données
&   , mxz_ref = 250     ! nombre max. de niveau des modes

!   Profils T et S:
&   , mxrepprov = 5     ! nombre max de repertoire de fichiers type
!   profil.
&   , mx_nb_prlu = 300  ! nombre max de profils (T ou S) lus par
!   fichiers
&   , mxz_dat = 260     ! nombre max de points dans les profils de
!   donnees
!   T ou S
&   , mx_nb_prof = 200  ! nombre max de profils (T ou S) utilises par
!   inversion
&   , mxz_ch_prof = 100 ! nombre max de couches sur lesquels sont
!   moyennes les profils de T ou S avant utilisation

!   Profils U,V de VMADCP:
&   , mx_pr_vmadcpx = 200 ! nombre max de profils VMADCP par jour.
&   , mxz_ch_adcp = 50   ! nombre max de couches VM-ADCP

!   Mouillages MADCP:
&   , mxrepmadcpx = 10   ! nombre max de Mouillages avec MADCP par jour.
&   , mxz_ch_m_adcp = 30 ! nombre max de couches M-ADCP

!   Mouillages T, S et courants:
&   , mxrepmUV= 20      ! nombre max de Mouillage UV.
&   , mxrepmT = 20      ! nombre max de Mouillage T.
&   , mxrepmS = mxrepmT ! nombre max de Mouillage S.
&   , mxminst = 15      ! nombre max d'instrument par mouillage
!   entre 0 et 2000m
&   , mxz_ch_m_UV = 130 ! nombre max de couches sur lesquelles sont
!   moyennes les profils vitesses issues de
!   mouillages avant utilisation

!   Flotteurs:
&   , mxifloat = 60     ! nombre max de flotteurs/jour
&   , mxz_ch_float_UV = 130 ! nombre max de couches sur lesquelles sont
!   moyennes les profils vitesses issues de
!   flotteurs avant utilisation

!   SOPRANE:
&   , mxlat_sop = 158   ! nombre max de points en latitude dans
!   les fichiers soprane complets
&   , mxlon_sop = 168   ! nombre max de points en longitude dans les
!   fichiers soprane complets
&   , istep_sop_lon = 4 ! facteur de sous echantillonnage des points
!   soprane
&   , istep_sop_lat = 4 ! facteur de sous echantillonnage des points
!   soprane

&   , mxlat_sop_s = mxlat_sop/istep_sop_lat
!   nombre max de points en latitude
!   dans les fichiers soprane reduits
&   , mxlon_sop_s = mxlon_sop/istep_sop_lon
!   nombre max de points en longitude
!   dans les fichiers soprane reduits

!   Temperatures de surface:
&   , mxlat_tsurf = 65  ! Nombre max de latitude donnees de temperature
!   de surface
&   , mxlon_tsurf = 50  ! Nombre max de longitude donnees de temperature

```

&) !de surface

4.5.2. Inversion

Pour réaliser une inversion, il faut :

- se connecter sur le calculateur scientifique de l'ifremer **nymphaea**
- L'utilisateur trouvera des informations sur ce calculateur sur l'intranet Ifremer à l'adresse : <http://w3.ifremer.fr/intranic/Assistance/Guides/nymphaea/intro.php>

L'interactif sur nymphaea étant limité à 1 heure sur un processeur (soit 15 minutes si on utilise 4 processeurs), il est nécessaire de lancer le programme d'inversion en batch. Le programme est à lancer sur la queue **bigmem** comme suit :

- **setenv OMP_NUM_THREADS 4**
- **bsub -N -o output -n 4 -R "span[ptile=4]" -q bigmem script_mp**

A priori, il faut que script_mp soit exécutable.

L'option '-n' permet d'indiquer le nombre de processeurs désirés.

Dans le cas où l'utilisateur doit lancer 2 inversions à suivre (avec la seconde inversion nécessitant le fichier de reprise de la première inversion), il faut lancer les 2 jobs ainsi :

- **bsub -J nom1 -N -o output1 -n 4 -R "span[ptile=4]" -q bigmem script_mp**
- **bsub -w 'done(nom1)' -N -o output2 -R "span[ptile=4]" -n 4 -q bigmem script_mp2**

Il est à noter l'intérêt de la queue batch **bigfast**. Elle permet de lancer sur un nœud de 32 Go un exécutable pour une durée maximale de 4 heures (soit 1 heure si on utilise les 4 processeurs). Cette queue est prioritaire par rapport à bigmem et peut se révéler très utile en période de tests et/ou de débogage pour lancer une inversion sur 1 à 2 jours.

Un exemple de script est fourni ci-après :

```
#!/bin/csh
cd /home2/nymphaea/fgaillar/inv_kalman/runs/pomme/run30
/home2/nymphaea/fgaillar/inv_kalman/versions/KA_V2.1/inkaln1_mp_V2 <
./batch/pom33_a.in
```

Le fichier **batch/pom33_a.in** doit contenir :

1. le nom de la racine des fichiers résultant de l'inversion
2. le nom du fichier de configuration (sans l'extension '_cfg')
3. les jours de début et de fin
4. le jour de reprise (0 = pas de reprise)
5. le nom du fichier de reprise

soit par exemple :

```
pom33_a
pom33
 256 325          % jrdeb, jrfin
 0              % j_reprise
resu/?????
```

4.6. Améliorations prévues

La version actuelle (V2.4) du programme d'inversion avec filtre de kalman n'est pas dans sa version définitive. Notamment, il est prévu de :

1. 'Nettoyer' le code :
 - enlever tout ce qui est relatif à calima et fichiers dérivés
 - adopter les notation 'Ide et al ' 'vecteur X, matrices P' dans les noms de variables
 - passer en f90
2. vérifier le calcul de la projection sur les modes de salinité
3. Ajouter des contraintes de conservation de la chaleur
4. Ajouter une contrainte de stabilité verticale, voire de forçage de surface dans l'étape de prévision

5. Analyse des résultats

L'exploitation des résultats issus de l'inversion s'effectue via l'utilitaire **inv_ana** écrit sous matlab. Cet outil a été développé au fur et à mesure des besoins. Il n'est pas encore finalisé et devra être amélioré du point de vue écriture et interface. Néanmoins, il est fonctionnel et offre à l'utilisateur différents types de tracés afin de visualiser et valider les résultats de ses inversions. La dernière version de cet utilitaire date de Novembre 2004.

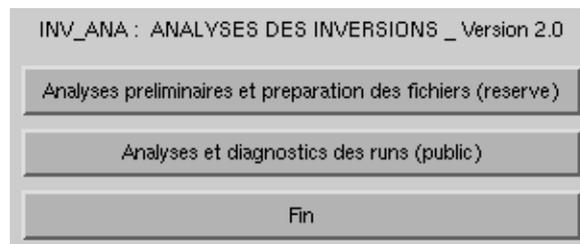
Afin de pouvoir exécuter **inv_ana**, l'utilisateur doit préalablement insérer le chemin **/home/doelanaval/chemon/inv_kalman/invana_V2.1** dans sa variable **MATLABPATH**. Sous matlab, il peut alors taper : **inv_ana**.

inv_ana est découpé en 2 parties :

- 1 partie réservée
- 1 partie publique

La partie réservée travaille sur les fichiers résultats de l'inversion stockés sur le calculateur nymphéa. Elle est accessible uniquement à F. Gaillard et C. Kermabon.

La partie publique travaille sur les fichiers issus de la partie réservée et est accessible à tous.



5.1. Partie réservée

Ce sous-menu permet notamment de :

- valider le résultat de l'inversion via la visualisation rapide :
 - de statistiques :
 - nombre de données prises en compte par l'inversion
 - résidus de l'inversion
 - de tracés :
 - tracé des modes
- générer les fichiers NetCDF utiles à la partie publique
- créer des animations pour la température, la salinité et la fonction courant à des niveaux définis.

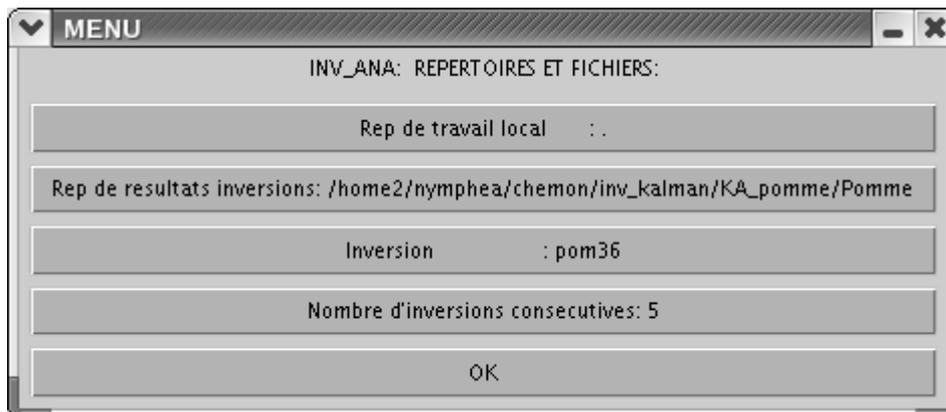


5.1.1. Initialisation des fichiers

Cette interface permet à l'utilisateur de définir l'inversion sur laquelle il va travailler.

L'utilisateur doit indiquer :

- Le répertoire de travail. Les fichiers de données seront générés sous le sous-répertoire *data* et les tracés sous le sous-répertoire *plot*.
- Le répertoire père du sous-répertoire *resu* où sont stockés les fichiers issus de l'inversion (généralement sur le calculateur nymphéa)
- Le nom de l'inversion (racine des fichiers à traiter)
- Le nombre de runs consécutifs à prendre en compte. Les batchs sont limités en temps sur le calculateur nymphéa. Une inversion peut alors être constituée de plusieurs runs (par exemple, un run du jour 1 au jour 70 puis un second run du jour 71 au jour 140). Dans ce cas, les runs doivent obligatoirement être nommés *nomin_a* pour le run1, *nominv_b* pour le second, ...etc.

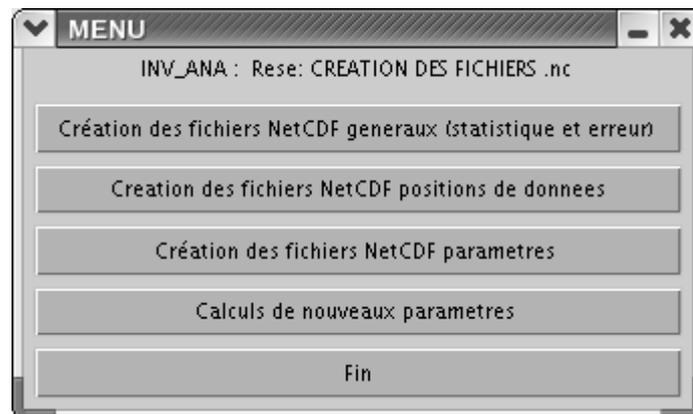


5.1.2. Création de fichiers NetCDF

Cette partie permet de :

- Créer les fichiers dits 'généraux' :
 - un fichier d'extension '*_stat.nc*' qui mémorise les statistiques (nombre de données, résidus, énergie des modes, ...etc)

- un fichier d'extension '*_erreur.nc*' qui contient les cartes d'erreur (erreur brute et pourcentage d'erreur) à deux niveaux.
- Créer un fichier par type de données, excepté pour les données soprane (car les positions des données soprane sont connues et identiques pour chaque jour). Ce fichier comprend pour chaque jour d'inversion des informations sur les données prises en compte par l'inversion :
 - La position des données
 - La valeur de la données
 - La valeur de l'innovation, du bruit et du résidu
- Créer des fichiers paramètre: l'utilisateur peut générer :
 - La température, ses dérivées horizontales et la salinité
 - Les vitesses (U,V), la fonction courant et la vorticité relative.
- Calculer d'autres paramètres à partir des paramètres de base (température, salinité, vitesses, pression). On peut ainsi calculer :
 - La température, salinité, ...etc lissée
 - La température potentielle
 - La densité in-situ
 - La densité potentielle à la surface
 - La densité potentielle à 1000m



5.1.3. Création d'animation

2 sous-menus permettent à l'utilisateur de créer des animations pour la température, la salinité et la fonction courant. Selon le sous-menu choisi, sur l'animation, on fait apparaître ou non la trajectoire des flotteurs.

5.2. Partie publique

Cette interface permet à l'utilisateur de :

- tracer les profils et modes utilisés pour l'inversion
- tracer les statistiques
- visualiser les champs (de température, de salinité, ...etc.)
- comparer les résultats de l'inversion avec les données
- déterminer d'autres paramètres via la sous-partie 'diagnostics'



5.2.1. Initialisation des fichiers

Cette partie permet à l'utilisateur de préciser :

- le répertoire de travail : Les fichiers de données et les tracés seront respectivement générés sous les sous-répertoires *data* et *plot*.
- le répertoire père du sous-répertoire *data* où se situent les fichiers à visualiser et/ou exploiter (il s'agit des fichiers créés par la partie réservée).
- le nom de l'inversion (racine des fichiers à prendre en compte)



5.2.2. Tracé des profils moyens et modes

Ce sous-menu permet de visualiser les profils moyens (température, salinité) et les modes utilisés pour l'inversion en cours. On trace également le profil moyen de pression qui a permis de déterminer les paramètres tels que la température potentielle.

5.2.3. Statistiques de l'inversion

Ce sous-menu permet de visualiser :

- l'énergie des modes
- le nombre de données par jour et par type
- le résidu sur bruit par jour et par type de données

5.2.4. Visualisations des champs

Cette interface permet de :

- tracer la carte d'un champ (de température, de salinité, ...etc.) pour une période donnée avec un pas défini pour une profondeur définie
- tracer la carte des champs de température, de salinité et de fonction courant pour un jour donné et pour une profondeur définie
- tracer des sections verticales de champ
 - selon une latitude
 - selon une longitude
 - en fonction du temps pour une position donnée
- tracer les champs de vitesse ou de gradients de température horizontaux sous forme de vecteurs
- tracer le diagramme Theta-S de l'inversion.



Certaines de ces visualisations font appel aux interfaces suivantes :

demande_info

Jour debut JJ/MM/AAAA	<input type="text"/>	Nombre figure/page	1 <input type="text"/>
Jour fin JJ/MM/AAAA	<input type="text"/>	Generation d'un postscript par figure	Oui <input type="text"/>
Pas	<input type="text"/>	Masque	Non <input type="text"/>
Profondeur (m)	<input type="text"/>		

OK

demande_info_sec

Type de section	zonale <input type="text"/>	Latitude	<input type="text"/>
Jour debut JJ/MM/AAAA	<input type="text"/>	Longitude	<input type="text"/>
Jour fin JJ/MM/AAAA	<input type="text"/>	Nombre de figures/pages	1 <input type="text"/>
Pas	<input type="text"/>	1 postscript/figure	Oui <input type="text"/>
Profondeur min (m)	<input type="text"/>	Masque	Non <input type="text"/>
Profondeur max (m)	<input type="text"/>	% erreur	<input type="text"/>
		Nom MLD	<input type="text"/>

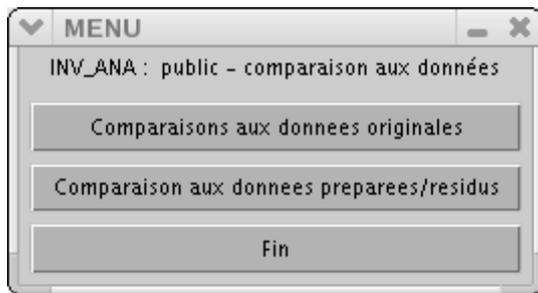
OK

5.2.5. Comparaison aux données

Cette partie permet de :

- Comparer le résultat de l'inversion avec les données originales (données soprane et données de température de surface)
- Comparer le résultat de l'inversion avec les données prises en compte par l'inversion (données préparées pour le calcul de l'inversion à partir des données originales). Ceci peut

être effectué pour les données eulériennes et les profils de température, salinité, ADCP de coque et ADCP mouillage.



5.2.6. diagnostics

Cette partie est divisée en 4 parties :

- 1 partie relative à la ***couche de mélange*** qui comprend :
 - 1 sous-partie 'calcul' qui permet de :
 - déterminer la couche de mélange pour une période de travail définie. Le résultat de ce calcul est mémorisé dans un fichier NetCDF d'extension '_MLD_datedebut_datefin.nc'.
Pour chaque jour et chaque point de l'inversion, on définit la profondeur de la couche de mélange (MLD) par :
 - la première profondeur au dessous de P telle que la température de MLD diffère de la température de P de plus de DT degré. P et DT sont entrés par l'utilisateur.
 - A chaque valeur de la couche de mélange est associé un flag initialisé à 0.
 - flagguer la couche de mélange en fonction de la carte d'erreur de température (brute ou en pourcentage)
 - Pour ce faire, l'utilisateur indique 2 valeurs :
 - L'erreur acceptable (E1) permettant de valider la couche de mélange à bon
 - L'erreur acceptable (E2) permettant de valider la couche de mélange à douteux
 - Ainsi, la couche de mélange est flagguée à 1 (bon) si l'erreur sur la température au niveau témoin (erreur générée par le programme d'inversion) est inférieure ou égale à E1. La couche de mélange est flagguée à 2 (douteux) si l'erreur sur la température au niveau témoin est comprise entre E1 et E2. La couche de mélange est flagguée à 3 (mauvais) si l'erreur sur la température au niveau témoin est supérieure à E2.
 - lisser la couche de mélange
 - calculer la température et densité potentielle moyenne de la couche de mélange
 - 1 sous-partie 'visualisation' permettant de :
 - tracer les cartes de couche de mélange
 - tracer la série temporelle de la couche de mélange pour une bande de latitude
 - tracer la série temporelle de la couche de mélange pour une position définie
 - étudier la série temporelle de la MLD.

- 1 partie relative aux *champs moyens* qui comprend :
 - 1 sous-partie ‘calcul’ qui permet de calculer la moyenne pour une période définie de la température, de la salinité, des composantes U et V de la vitesse, des gradients horizontaux de température, des termes turbulents de la vitesse.
 - 1 sous-partie ‘visualisation’ qui permet de tracer tous ces paramètres
- 1 partie relative aux *transports de chaleur* comportant :
 - 1 sous-partie ‘calcul’ pour déterminer le flux de chaleur entre 2 profondeurs
 - 1 sous-partie ‘visualisation’ permettant de tracer ces paramètres
- 1 partie ‘*autres*’ permettant de calculer :
 - la fréquence de Brunt-Vaisala pour un jour ou une position
 - le module du gradient de température (permettant éventuellement de détecter les tourbillons)



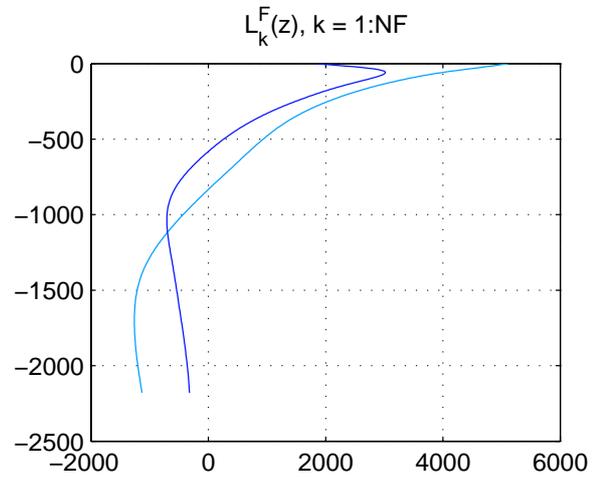
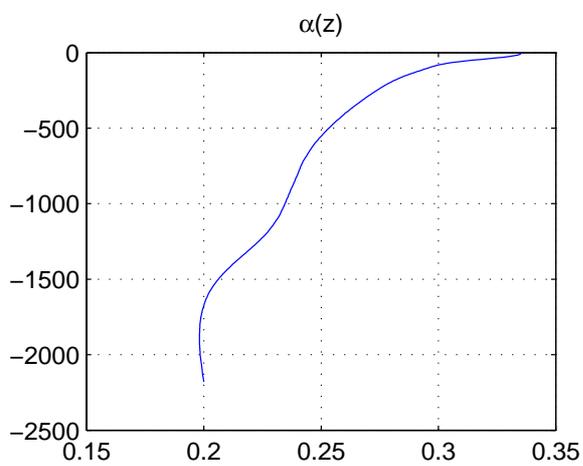
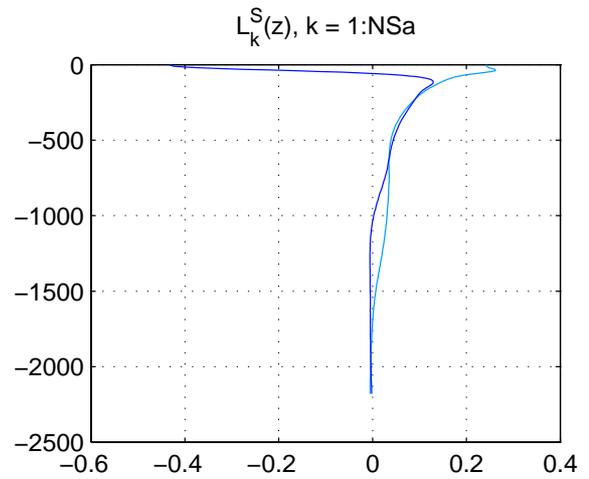
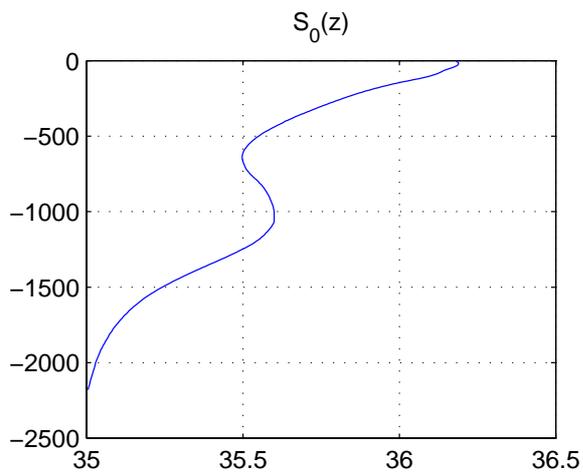
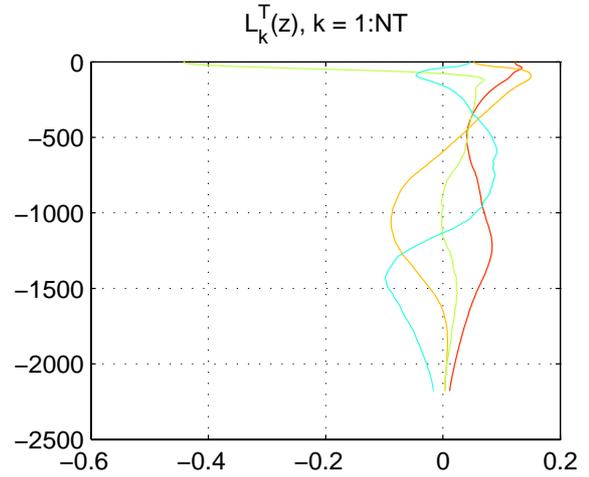
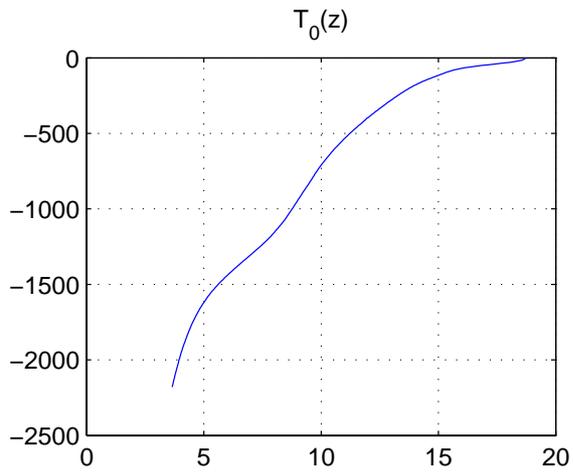
5.2.7. Exemples de tracés

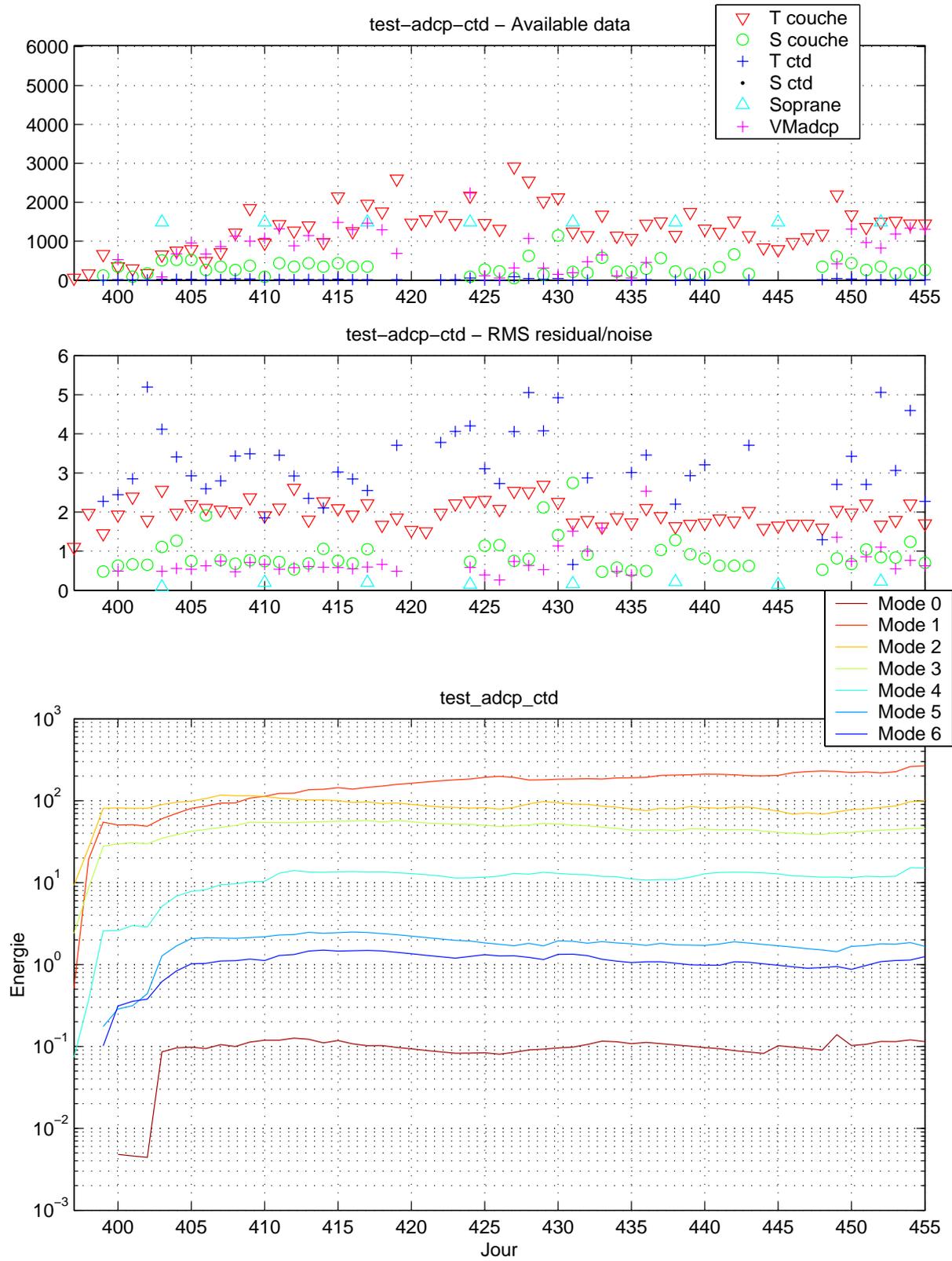
Pour des exemples de tracés, l'utilisateur peut se référer à la documentation : ‘Inversion avec filtre de kalman : Validation du modèle’ dont tous les tracés ont été générés via cet utilitaire.

Néanmoins, on trouvera ci-après :

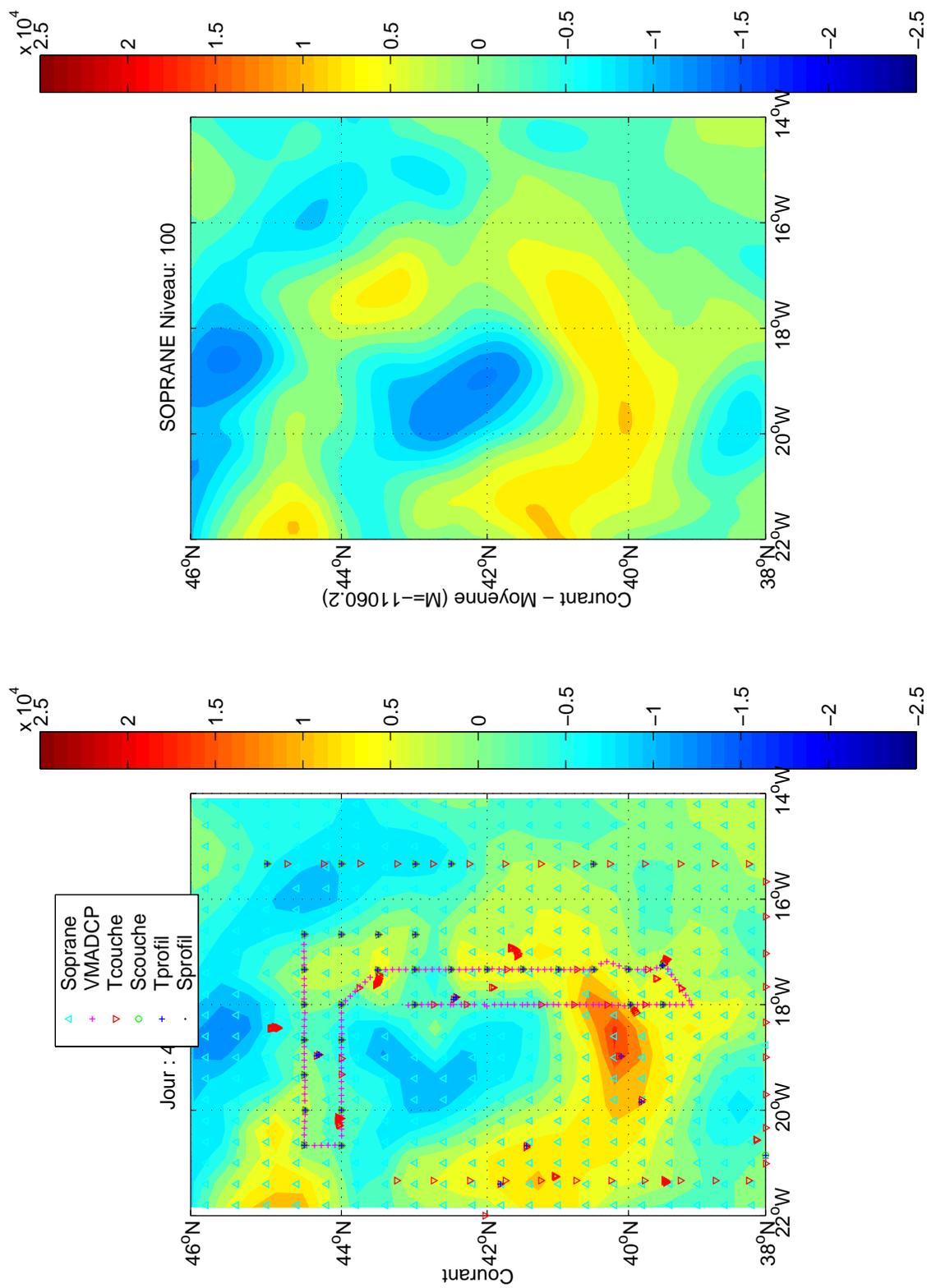
- Un exemple de tracé associé :
 - Au profil de température moyen
 - Au profil de salinité moyen
 - Aux divers modes de température, salinité et courant
- Un exemple de tracé associé aux statistiques générales de l'inversion
- Un exemple de comparaison entre les données originales et le résultat de l'inversion pour :
 - Les données de courant SOPRANE
 - Les données de température

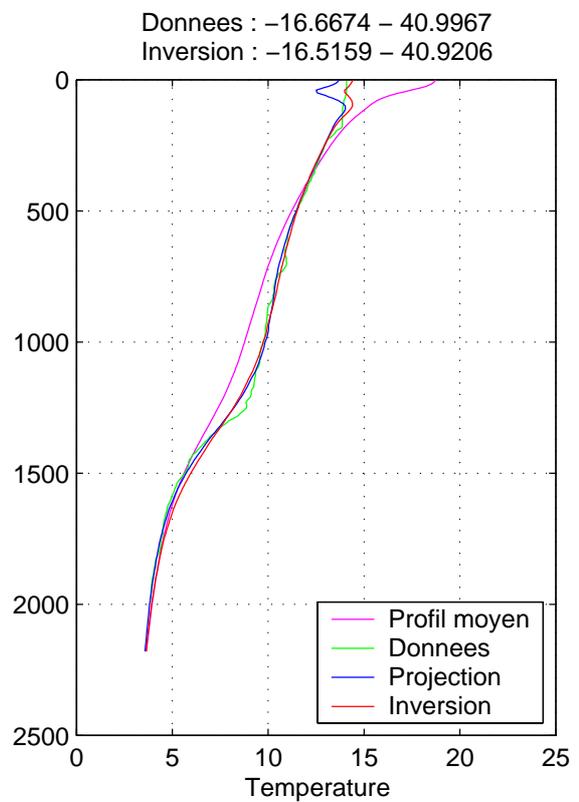
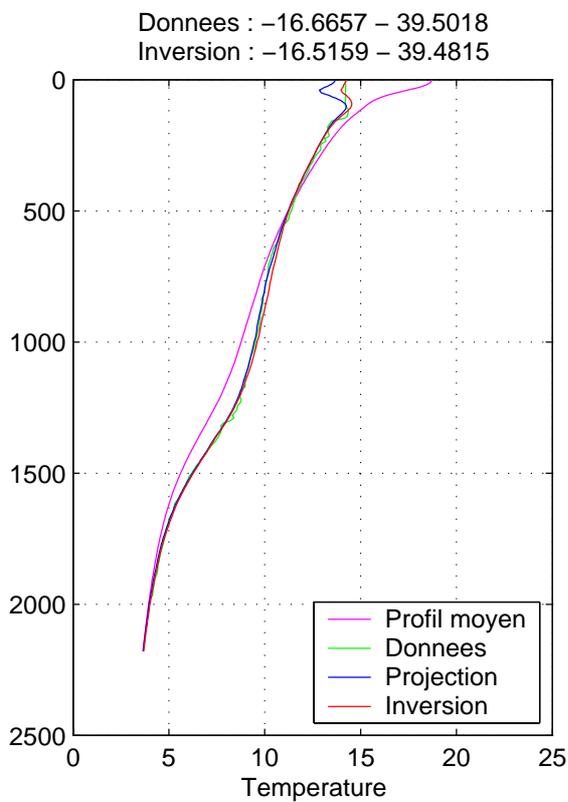
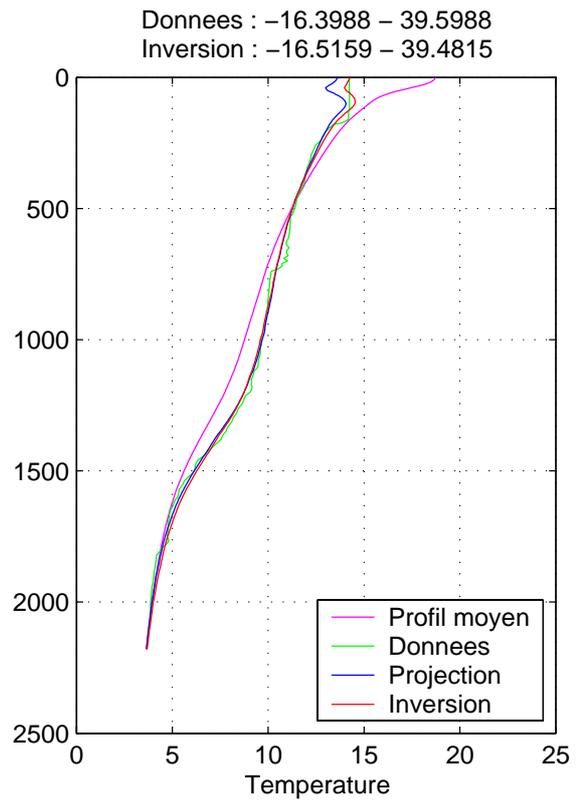
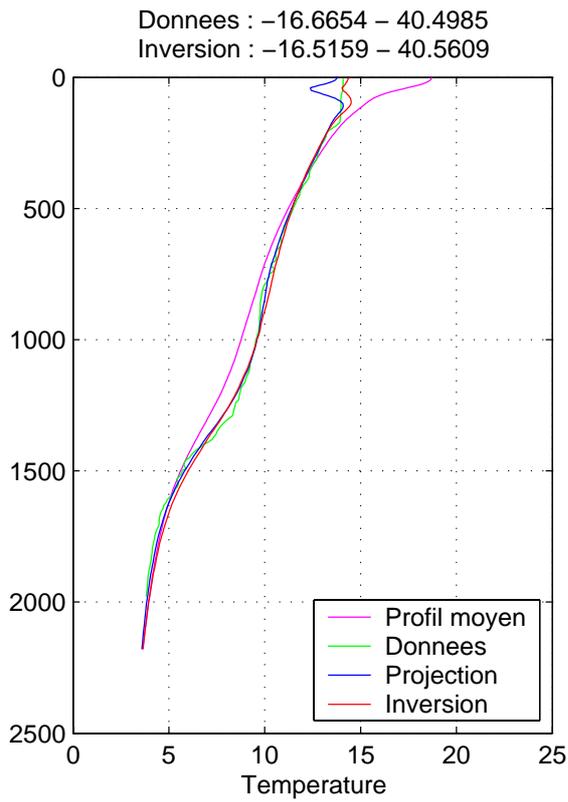
test_adcp_ctd





test_adcp_ctd





5.3. Améliorations prévues

Outre pour les interfaces, **inv_ana** doit aussi être modifié d'un point de vue fonctionnel. Notamment, il est prévu de :

- Générer automatiquement le fichier moyen de pression à partir des fichiers température et salinité NetCDF créés dans la partie réservée.

6. Annexes

6.1. Format NetCDF des fichiers de données

6.1.1. Fichier Profileur

```
netcdf PF_profil-y2000m09d26 {
dimensions:
    mN_PROF = 12 ;
    mN_ZLEV = 250 ;
variables:
    float JULD(mN_PROF) ;
        JULD:long_name = "Julian Days relative to Reference_Date_Time"
;
        JULD:units = "Julian Days" ;
        JULD:Fillvalue = -9999.f ;
        JULD:Conventions = "Julian Days relative to
Reference_Date_Time" ;
    float LONGITUDE(mN_PROF) ;
        LONGITUDE:long_name = "Longitude" ;
        LONGITUDE:units = "degrees" ;
        LONGITUDE:Fillvalue = -9999.f ;
    float LATITUDE(mN_PROF) ;
        LATITUDE:long_name = "Latitude" ;
        LATITUDE:units = "degrees" ;
        LATITUDE:Fillvalue = -9999.f ;
    float DEPH(mN_PROF, mN_ZLEV) ;
        DEPH:long_name = "Depth below sea surface" ;
        DEPH:units = "metres" ;
        DEPH:Fillvalue = -9999.f ;
    float TEMP(mN_PROF, mN_ZLEV) ;
        TEMP:long_name = "Temperature in situ T90" ;
        TEMP:units = "degres Celsius" ;
        TEMP:Fillvalue = -9999.f ;
    float Error_TEMP(mN_PROF, mN_ZLEV) ;
        Error_TEMP:long_name = "Error on Temperature in situ T90" ;
        Error_TEMP:units = "degres Celsius" ;
        Error_TEMP:Fillvalue = -9999.f ;
    float PSAL(mN_PROF, mN_ZLEV) ;
        PSAL:long_name = "Salinity" ;
        PSAL:units = "psu" ;
        PSAL:Fillvalue = -9999.f ;
    float Error_PSAL(mN_PROF, mN_ZLEV) ;
        Error_PSAL:long_name = "Error on Salinity" ;
        Error_PSAL:units = "psu" ;
        Error_PSAL:Fillvalue = -9999.f ;

// global attributes:
    :Last_update = "20-Aug-2003" ;
    :Experiment_name = "POMME" ;
    :File_contents = "TE:Cleaned+Pseudosal" ;
    :Reference_date_time = "01/01/1950 00:00:00" ;
    :Start_date = "26-Sep-2000 04:04:41" ;
    :Stop_date = "26-Sep-2000 15:45:00" ;
```

```
:South_latitude = 38.9780006408691 ;  
:North_latitude = 43.7999992370605 ;  
:West_longitude = -20.6720008850098 ;  
:East_longitude = -17.9389991760254 ;  
:Comments = "Created by program ka_dataprep_profTS.m" ;  
}
```

6.1.2. Format des fichiers NetCDF flotteurs

```
netcdf POM_float-y2000m09d20 {
dimensions:
    mN_FLOAT = UNLIMITED ; // (13 currently)
    mN_NAME = 9 ;
variables:
    char FLOAT_NAME(mN_FLOAT, mN_NAME) ;
        FLOAT_NAME:long_name = "Float Name" ;
    float JULD(mN_FLOAT) ;
        JULD:long_name = "Julian days relative to Reference date time"
;
        JULD:convention = "Julian days relative to Reference date time"
;
        JULD:units = "Julian Days" ;
        JULD:Fillvalue = -9999.f ;
    float LONGITUDE(mN_FLOAT) ;
        LONGITUDE:long_name = "Longitude" ;
        LONGITUDE:units = "Degrees" ;
        LONGITUDE:Fillvalue = -9999.f ;
    float LATITUDE(mN_FLOAT) ;
        LATITUDE:long_name = "Latitude" ;
        LATITUDE:units = "Degrees" ;
        LATITUDE:Fillvalue = -9999.f ;
    float DEPTH(mN_FLOAT) ;
        DEPTH:long_name = "Depth" ;
        DEPTH:units = "metre" ;
        DEPTH:Fillvalue = -9999.f ;
    float TEMP(mN_FLOAT) ;
        TEMP:long_name = "Temperature" ;
        TEMP:units = "degres Celsius" ;
        TEMP:Fillvalue = -9999.f ;
    float Error_TEMP(mN_FLOAT) ;
        Error_TEMP:long_name = "Error on Temperature" ;
        Error_TEMP:units = "degres Celsius" ;
        Error_TEMP:Fillvalue = -9999.f ;
    float U(mN_FLOAT) ;
        U:long_name = "Zonal speed" ;
        U:units = "cm/s" ;
        U:Fillvalue = -9999.f ;
    float Error_U(mN_FLOAT) ;
        Error_U:long_name = "Error on Zonal speed" ;
        Error_U:units = "cm/s" ;
        Error_U:Fillvalue = -9999.f ;
    float V(mN_FLOAT) ;
        V:long_name = "Meridional speed" ;
        V:units = "cm/s" ;
        V:Fillvalue = -9999.f ;
    float Error_V(mN_FLOAT) ;
        Error_V:long_name = "Error on Meridional speed" ;
        Error_V:units = "cm/s" ;
        Error_V:Fillvalue = -9999.f ;

// global attributes:
    :Last_update = "09-Jan-2004 14:12:59" ;
    :Experiment_name = "POMME" ;
    :File_contents = "Float Data : T, U, V" ;
    :Reference_date_time = "01/01/1950 00:00:00" ;
    :Comments = "Created by cree_fic_flotteur.m" ;
}
```

6.1.3. Format des fichiers NetCDF mouillages TS

```
netcdf POM_m_TS-y2000m11d23 {
dimensions:
    mN_ZLEV = 13 ;
    mN_PROF = UNLIMITED ; // (4 currently)
    mN_Mnam = 2 ;
variables:
    char MOOR_NAME(mN_PROF, mN_Mnam) ;
        MOOR_NAME:long_name = "Mooring name" ;
    float JULD(mN_PROF) ;
        JULD:long_name = "Julian Days relative to Reference_Date_Time"
;
        JULD:units = "Julian Days" ;
        JULD:Fillvalue = -9999.f ;
        JULD:Conventions = "Julian Days relative to
Reference_Date_Time" ;
    float LONGITUDE(mN_PROF) ;
        LONGITUDE:long_name = "Longitude" ;
        LONGITUDE:units = "degrees" ;
        LONGITUDE:Fillvalue = -9999.f ;
    float LATITUDE(mN_PROF) ;
        LATITUDE:long_name = "Latitude" ;
        LATITUDE:units = "degrees" ;
        LATITUDE:Fillvalue = -9999.f ;
    float DEPH(mN_PROF, mN_ZLEV) ;
        DEPH:long_name = "Depth below sea surface" ;
        DEPH:units = "metres" ;
        DEPH:Fillvalue = -9999.f ;
    float TEMP(mN_PROF, mN_ZLEV) ;
        TEMP:long_name = "Temperature in situ T90" ;
        TEMP:units = "degres Celsius" ;
        TEMP:Fillvalue = -9999.f ;
    float Error_TEMP(mN_PROF, mN_ZLEV) ;
        Error_TEMP:long_name = "Error on Temperature in situ T90" ;
        Error_TEMP:units = "degres Celsius" ;
        Error_TEMP:Fillvalue = -9999.f ;
    float PSAL(mN_PROF, mN_ZLEV) ;
        PSAL:long_name = "Salinity" ;
        PSAL:units = "psu" ;
        PSAL:Fillvalue = -9999.f ;
    float Error_PSAL(mN_PROF, mN_ZLEV) ;
        Error_PSAL:long_name = "Error on Salinity" ;
        Error_PSAL:units = "psu" ;
        Error_PSAL:Fillvalue = -9999.f ;
    float Flag_PSAL(mN_PROF, mN_ZLEV) ;
        Flag_PSAL:long_name = "Flag on Salinity. 1:measured ok; 2:
pseudosal" ;
        Flag_PSAL:units = "psu" ;
        Flag_PSAL:Fillvalue = -9999.f ;

// global attributes:
    :Last_update = "12-Dec-2003" ;
    :Experiment_name = "POMME" ;
    :File_contents = "Mooring: T+S+Pseudosal" ;
    :Reference_date_time = "01/01/1950 00:00:00" ;
    :Start_date = "23/11/2000" ;
    :Stop_date = "23/11/2000" ;
    :South_latitude = 38.997167 ;
    :North_latitude = 44.998333 ;
    :West_longitude = -20.003333 ;
    :East_longitude = -15.27 ;
    :Comments = "Created by program cree_netcdf_temp.m and
cree_fic_inv.m" ;
```

6.1.4. Format des fichiers NetCDF mouillages UV

```
netcdf POM_m_UV-y2000m10d18 {
dimensions:
    mN_ZLEV = 13 ;
    mN_PROF = UNLIMITED ; // (5 currently)
    mN_Mnam = 2 ;
variables:
    char MOOR_NAME(mN_PROF, mN_Mnam) ;
        MOOR_NAME:long_name = "Mooring name" ;
    float JULD(mN_PROF) ;
        JULD:long_name = "Julian Days relative to Reference_Date_Time"
;
        JULD:units = "Julian Days" ;
        JULD:Fillvalue = -9999.f ;
        JULD:Conventions = "Julian Days relative to
Reference_Date_Time" ;
    float LONGITUDE(mN_PROF) ;
        LONGITUDE:long_name = "Longitude" ;
        LONGITUDE:units = "degrees" ;
        LONGITUDE:Fillvalue = -9999.f ;
    float LATITUDE(mN_PROF) ;
        LATITUDE:long_name = "Latitude" ;
        LATITUDE:units = "degrees" ;
        LATITUDE:Fillvalue = -9999.f ;
    float DEPH(mN_PROF, mN_ZLEV) ;
        DEPH:long_name = "Depth below sea surface" ;
        DEPH:units = "metres" ;
        DEPH:Fillvalue = -9999.f ;
    float U(mN_PROF, mN_ZLEV) ;
        U:long_name = "East composant speed" ;
        U:units = "cm/s" ;
        U:Fillvalue = -9999.f ;
    float Error_U(mN_PROF, mN_ZLEV) ;
        Error_U:long_name = "Error on U" ;
        Error_U:units = "cm/s" ;
        Error_U:Fillvalue = -9999.f ;
    float V(mN_PROF, mN_ZLEV) ;
        V:long_name = "North composant speed" ;
        V:units = "cm/s" ;
        V:Fillvalue = -9999.f ;
    float Error_V(mN_PROF, mN_ZLEV) ;
        Error_V:long_name = "Error V" ;
        Error_V:units = "cm/s" ;
        Error_V:Fillvalue = -9999.f ;

// global attributes:
    :Last_update = "12-Dec-2003" ;
    :Experiment_name = "POMME" ;
    :File_contents = "Mooring: U, V" ;
    :Reference_date_time = "01/01/1950 00:00:00" ;
    :Start_date = "18/10/2000" ;
    :Stop_date = "18/10/2000" ;
    :South_latitude = 38.997167 ;
    :North_latitude = 44.998333 ;
    :West_longitude = -20.003333 ;
    :East_longitude = -15.27 ;
    :Comments = "Created by program cree_netcdf_UV.m and
cree_fic_inv.m" ;
}
```

6.1.5. Format des fichiers NetCDF MADCP

```
netcdf POM_m_ADCP-y2000m11d08 {
dimensions:
    mN_ZLEV = 30 ;
    mN_PROF = UNLIMITED ; // (2 currently)
    mN_Mnam = 2 ;
variables:
    char MOOR_NAME(mN_PROF, mN_Mnam) ;
        MOOR_NAME:long_name = "Mooring name" ;
    float JULD(mN_PROF) ;
        JULD:long_name = "Julian Days relative to Reference_Date_Time"
;
        JULD:units = "Julian Days" ;
        JULD:Fillvalue = -9999.f ;
        JULD:Conventions = "Julian Days relative to
Reference_Date_Time" ;
    float LONGITUDE(mN_PROF) ;
        LONGITUDE:long_name = "Longitude" ;
        LONGITUDE:units = "degrees" ;
        LONGITUDE:Fillvalue = -9999.f ;
    float LATITUDE(mN_PROF) ;
        LATITUDE:long_name = "Latitude" ;
        LATITUDE:units = "degrees" ;
        LATITUDE:Fillvalue = -9999.f ;
    float DEPH(mN_PROF, mN_ZLEV) ;
        DEPH:long_name = "Depth below sea surface" ;
        DEPH:units = "metres" ;
        DEPH:Fillvalue = -9999.f ;
    float U(mN_PROF, mN_ZLEV) ;
        U:long_name = "East composant speed" ;
        U:units = "cm/s" ;
        U:Fillvalue = -9999.f ;
    float Error_U(mN_PROF, mN_ZLEV) ;
        Error_U:long_name = "Error on U" ;
        Error_U:units = "cm/s" ;
        Error_U:Fillvalue = -9999.f ;
    float V(mN_PROF, mN_ZLEV) ;
        V:long_name = "North composant speed" ;
        V:units = "cm/s" ;
        V:Fillvalue = -9999.f ;
    float Error_V(mN_PROF, mN_ZLEV) ;
        Error_V:long_name = "Error V" ;
        Error_V:units = "cm/s" ;
        Error_V:Fillvalue = -9999.f ;

// global attributes:
    :Last_update = "05-Jan-2004" ;
    :Experiment_name = "POMME" ;
    :File_contents = "Mooring: U, V" ;
    :Reference_date_time = "01/01/1950 00:00:00" ;
    :Start_date = "08/11/2000" ;
    :Stop_date = "08/11/2000" ;
    :South_latitude = 38.9972 ;
    :North_latitude = 44.9983 ;
    :West_longitude = -18.0125 ;
    :East_longitude = -17.991 ;
    :Comments = "Created by program cree_netcdf_UV.m and
cree_fic_inv.m" ;}
```

6.1.6. Format des fichiers NetCDF soprane

```
netcdf sopr02_y2000m03d22 {
dimensions:
    N_CODE = 4 ;
    N_DATE_TIME = 19 ;
    N_STRING4 = 4 ;
    N_STRING16 = 16 ;
    N_COOR = 3 ;
    N_DATE_JUL1 = 1 ;
    Level = 10 ;
    Latitude = 158 ;
    Longitude = 168 ;
    N_PARAM = 40 ;
    N_DATA = UNLIMITED ; // (0 currently)
variables:
    float Estimated_julian(N_DATE_JUL1) ;
        Estimated_julian:long_name = "Julian day of estimation" ;
        Estimated_julian:Conventions = "Julian day relative to
Reference_date_time" ;
        Estimated_julian:_FillValue = -9999.f ;
    float Start_julian(N_DATE_JUL1) ;
        Start_julian:long_name = "Julian day of first data" ;
        Start_julian:Conventions = "Julian day relative to
Reference_date_time" ;
        Start_julian:_FillValue = -9999.f ;
    float Stop_julian(N_DATE_JUL1) ;
        Stop_julian:long_name = "Julian day of last data" ;
        Stop_julian:Conventions = "Julian day relative to
Reference_date_time" ;
        Stop_julian:_FillValue = -9999.f ;
    float Level(Level) ;
        Level:long_name = "Level" ;
        Level:units = "meters" ;
        Level:_FillValue = -9999.f ;
        Level:valid_min = 0.f ;
        Level:valid_max = 7000.f ;
    float Latitude(Latitude) ;
        Latitude:long_name = "Latitude of zone" ;
        Latitude:units = "degrees_north" ;
        Latitude:_FillValue = -9999.f ;
        Latitude:valid_min = -90.f ;
        Latitude:valid_max = 90.f ;
    float Longitude(Longitude) ;
        Longitude:long_name = "Longitude of zone" ;
        Longitude:units = "degrees_east" ;
        Longitude:_FillValue = -9999.f ;
        Longitude:valid_min = -180.f ;
        Longitude:valid_max = 180.f ;
    float Fact_error(Level) ;
        Fact_error:long_name = "Covariance Error Factor " ;
        Fact_error:_FillValue = -9999.f ;
    int Flag_180 ;
        Flag_180:long_name = "Does the area cross the 180 meridian?
(e.g. Pacific Ocean)" ;
        Flag_180:Conventions = "1 : 180 crossed, 0 : 180 not crossed" ;
    char PARAMETERS(N_PARAM, N_CODE) ;
        PARAMETERS:long_name = "List of parameters" ;
        PARAMETERS:Conventions = "GF3 code of the indexed parameter
among (DEPH, PRES, PSAL, TEMP + extended codes)" ;
        PARAMETERS:_FillValue = "" ;
    float PSIA(Level, Latitude, Longitude) ;
        PSIA:long_name = "Streamfunction" ;
        PSIA:_FillValue = -9.999e+08f ;
```

```

        PSIA:units = "m^2/s" ;
        PSIA:valid_min = -9.999e+07f ;
        PSIA:valid_max = 9.999e+07f ;
float Error_MSLA(Latitude, Longitude) ;
        Error_MSLA:long_name = "Error on MSLA " ;
        Error_MSLA:_FillValue = -9999.f ;
        Error_MSLA:units = "%" ;
        Error_MSLA:valid_min = 0.f ;
        Error_MSLA:valid_max = 100.f ;
float UGEO(Level, Latitude, Longitude) ;
        UGEO:long_name = "Geostrophic velocity - East component" ;
        UGEO:_FillValue = -9999.f ;
        UGEO:units = "m/s" ;
        UGEO:valid_min = -10.f ;
        UGEO:valid_max = 10.f ;
float VGEO(Level, Latitude, Longitude) ;
        VGEO:long_name = "Geostrophic velocity - North component" ;
        VGEO:_FillValue = -9999.f ;
        VGEO:units = "m/s" ;
        VGEO:valid_min = -10.f ;
        VGEO:valid_max = 10.f ;
float Error_VGEO(Latitude, Longitude) ;
        Error_VGEO:long_name = "Error on geostrophic velocity" ;
        Error_VGEO:_FillValue = -9999.f ;
        Error_VGEO:units = "m/s" ;
        Error_VGEO:valid_min = -10.f ;
        Error_VGEO:valid_max = 10.f ;

// global attributes:
        :Title = "SOPRANE rejeu all levels" ;
        :Project_name = "POMME" ;
        :Experiment_name = "SOPRANE" ;
        :Experiment_description = "SOPRANE analyzed fields during POMME" ;
" ;
        :Data_provider = "PSI: M. Assenbaum, UV: LPO" ;
        :File_contents = "analyzed fields model+altimetry" ;
        :File_content_description = "Vitesses derivees de la fonction
courant SOPRANE sur grille C ##Erreur_uv : deduite de l\'erreur MSLA selon
le tableau suivant ##0-5->2.5,5-10->3.0,10-20->3.5,20-30->4.0,30-40-
>4.5,40-50->5.0,>50->10.0" ;
        :Geographical_area = "North-East Atlantic" ;
        :South_latitude = 33.f ;
        :North_latitude = 48.7f ;
        :West_longitude = -25.f ;
        :East_longitude = -8.3f ;
        :Coord_system = "GEOGRAPHICAL-WGS84" ;
        :Analysis_period = "years 2000-2001" ;
        :Reference_date_time = "01/01/1950 00:00:00" ;
        :Estimated_date_time = "22/03/2000 00:00:00" ;
        :Start_date_time = "15/03/2000 00:00:00" ;
        :Stop_date_time = "21/03/2000 00:00:00" ;
        :Reference_parameter = "DEPH" ;
        :Data_types = " " ;
        :Input_data_files = " " ;
" ;
        :Software = "matlab: soprane.m" ;
        :Last_update = "02-Dec-2003 14:45:16" ;
}

```

6.1.7. Format des fichiers NetCDF ADCP de coque

```
netcdf pom_adcp_y2000m09d19 {
dimensions:
    N_pts = 1 ;
    N_deph = 40 ;
    N_camp = 9 ;
    N_char = 10 ;
variables:
    float JULD(N_pts) ;
        JULD:long_name = "Julian day of profiles relative to
Reference_date_time" ;
        JULD:units = "julian days" ;
        JULD:Conventions = "Relative julian days" ;
        JULD:_FillValue = -9999.f ;
    float LATITUDE(N_pts) ;
        LATITUDE:long_name = "Latitude of profiles" ;
        LATITUDE:units = "degrees_north" ;
        LATITUDE:_FillValue = -9999.f ;
        LATITUDE:valid_min = -90.f ;
        LATITUDE:valid_max = 90.f ;
    float LONGITUDE(N_pts) ;
        LONGITUDE:long_name = "Longitude of profiles" ;
        LONGITUDE:units = "degrees_east" ;
        LONGITUDE:_FillValue = -9999.f ;
        LONGITUDE:valid_min = -180.f ;
        LONGITUDE:valid_max = 180.f ;
    float DEPH(N_deph) ;
        DEPH:long_name = "Depth" ;
        DEPH:units = "meter" ;
        DEPH:_FillValue = -9999.f ;
        DEPH:valid_min = 0.f ;
        DEPH:valid_max = 2000.f ;
    char CRUISE_NAME(N_camp, N_char) ;
        CRUISE_NAME:long_name = "Name of the cruise to which Num_cruise
relates" ;
    float VEL_U(N_pts, N_deph) ;
        VEL_U:long_name = "Zonal or Eastward velocity" ;
        VEL_U:_FillValue = -9999.f ;
        VEL_U:units = "m/s" ;
        VEL_U:valid_min = -20.f ;
        VEL_U:valid_max = 20.f ;
    float VEL_V(N_pts, N_deph) ;
        VEL_V:long_name = "Meridional or Northward velocity" ;
        VEL_V:_FillValue = -9999.f ;
        VEL_V:units = "m/s" ;
        VEL_V:valid_min = -20.f ;
        VEL_V:valid_max = 20.f ;
    float VEL_ERR(N_pts, N_deph) ;
        VEL_ERR:long_name = "Error on Eastward or Northward velocity" ;
        VEL_ERR:_FillValue = -9999.f ;
        VEL_ERR:units = "m/s" ;
        VEL_ERR:valid_min = -20.f ;
        VEL_ERR:valid_max = 20.f ;
    short Num_cruise(N_pts) ;
        Num_cruise:long_name = "Cruise number in CRUISE_NAME list" ;

// global attributes:
    :Last_update = "20-Jan-2004 15:17:34" ;
    :Experiment_name = "all cruises" ;
    :Project_name = "POMME" ;
    :Experiment_description = " " ;
    :Geographical_area = "North-East Atlantic" ;
    :Data_type = "dayly ADCP data for inversion" ;
    :Data_provider = "F. Gaillard" ;
```

```
}  
:Version = "Version 2" ;  
:Reference_date_time = "01/01/1950 00:00:00" ;  
:Coord_system = "GEOGRAPHICAL-WGS84" ;
```

6.1.8. Format des fichiers NetCDF température de surface

```
netcdf POM_Tsurf-y2000m10d12 {
dimensions:
    LON = 100 ;
    LAT = 130 ;
    Cte = 1 ;
variables:
    float DEPH(Cte) ;
        DEPH:long_name = "Depth" ;
        DEPH:units = "m" ;
    float LON(LON) ;
        LON:long_name = "Longitude" ;
        LON:units = "degrees_east" ;
    float LAT(LAT) ;
        LAT:long_name = "Latitude" ;
        LAT:units = "degrees_north" ;
    float SST(LAT, LON) ;
        SST:long_name = "Temperature" ;
        SST:units = "degrees_Celsius" ;
    float Error_SST(LAT, LON) ;
        Error_SST:long_name = "Error on Temperature" ;
        Error_SST:units = "degrees_Celsius" ;
}
```

6.2. Exemple d'arborescence pour l'inversion

