

Auteurs:

P. Le Bot (DOPS/LPO)

C. Kermabon (DOPS/LPO)

P. Lherminier (DOPS/LPO)

F. Gaillard (DOPS/LPO)

3 Février 2011 – Rapport OPS/LPO 11-01

**Chaine Automatisée de Suivi des
Courantomètres Acoustiques
Doppler Embarqués.**

**CASCADE V6.1 : Logiciel de validation et de
visualisation des mesures ADCP de coque**

Documentation utilisateur et maintenance

SOMMAIRE

1	INTRODUCTION GENERALE	1
2	PREPARATION DE L'EXPLOITATION DES DONNEES .	3
2.1	Configuration minimum requise	3
2.1.1	- Données de départ.....	3
2.1.2	- Pour les navires équipé de TRANSECT	3
2.1.3	- Pour les navires équipé de VMDAS	4
2.2	Arborescence de travail recommandée	4
2.3	Arborescence de développement	5
3	EXPLOITATION DES DONNEES	6
3.1	Conversion de fichiers VMDAS :	7
3.2	Configuration.....	9
3.3	Validation Correction.....	10
3.3.1	- Définition de la couche de référence.....	10
3.3.2	- Ajout de la bathymétrie	11
3.3.3	- Changement de navigation	12
3.3.4	- Changement de cap.....	13
3.3.5	- Nettoyage.....	14
3.3.6	- Correction de Désalignement/Assiette/Amplitude	15
3.3.7	- Ajout de la marée	17
3.3.8	- Invalidation de données entre 2 dates	18
3.3.9	- Invalidation de données entre 2 ensembles.....	19
3.3.10	- Matérialisation des périodes sans mesure.....	20
3.3.11	- Filtrage	21
3.4	Analyse Fichiers campagnes	22
3.4.1	- Tracé de la dérive	23
3.4.2	- Tracé 1D	24
3.4.3	- Tracé 2D	25
3.4.4	- Comparaison.....	26
3.4.5	- Créer un fichier section	29
3.4.5	- Créer un fichier station	30
3.5	Analyse Fichiers Exploitation	31
3.5.1	- Section	31
3.5.2	- Station	35
3.6	Aide.....	37
3.6.1	- Information flags.....	37
3.6.2	- Informations campagne.....	37
4	REFERENCES	39
5	ANNEXES	40

TABLE DES ILLUSTRATIONS

Figure 1 – Interface CASCADE EXPLOITATION.....	6
Figure 2 – Interface conversion de fichiers VMDAS.....	8
Figure 3 - Différence de temps entre les ensembles.....	8
Figure 4 – Fenêtre de configuration fichier et répertoires	10
Figure 5 – Validation / corrections (1) – couche de référence.....	11
Figure 6 – Ecran ajout de la bathymétrie	12
Figure 7 – Ecran changement de navigation.....	12
Figure 8 – Ecran changement de cap	13
Figure 9 – <i>Ecran nettoyage</i>	15
Figure 10 – Ecran de correction de Désalignement/Assiette/Amplitude	16
Figure 11 – Ajout de la marée	17
Figure 12 – Ecran invalidation de données entre 2 dates	18
Figure 13 – Ecran invalidation de données entre 2 ensembles	19
Figure 14 – Ecran de matérialisation des périodes sans mesure	20
Figure 15 – Ecran du filtrage	21
Figure 16 – Ecran analyse fichiers campagne	22
Figure 17 – Fenêtre tracé de la dérive	23
Figure 18 – Ecran de Tracé 1D	24
Figure 19 – Ecran de Tracé 2D	25
Figure 20 – Ecran Comparaison Uadcp/Unavire.....	26
Figure 21 – Ecran Comparaison Route/Station.....	27
Figure 22 – Ecran Comparaison du cap Unavire/Ubottom.....	28
Figure 23 – Ecran Créer un fichier section.....	29
Figure 24 – Ecran Créer un fichier station.....	30
Figure 25 – Ecran d’exploitation d’un fichier section	31
Figure 26 – Ecran Contourage pour les sections	32
Figure 27 – Tracé de vecteurs pour les sections	33
Figure 28 – Tracé 1D pour les sections	34
Figure 29 – Tracé 2D pour les sections	34
Figure 30 – Ecran d’exploitation d’un fichier station.....	35
Figure 31 – Ecran des profils moyens pour une station	36
Figure 32 – Informations flags.....	37
Figure 33 – Ecran informations générales.	38

TABLE DES ANNEXES

Annexe I. Exemple de fichiers section et station ASCII	40
Annexe II. Tableau descriptif du format OceanSite.....	41
Annexe III. Format de fichier OceanSite	44
Annexe IV. Format fichier TRINAV.....	51
Annexe V. Format fichier cap externe	52
Annexe VI. Exemple de fichier NetCDF section.....	53
Annexe VII. Exemple de fichier NetCDF station.....	55
Annexe VIII Exemple de fichier NetCDF de Bathymétrie.....	57
Annexe IX. Tracé de conversion de fichier VMDAS.....	58
Annexe X. Tracé de contrôle de l'ajout de la bathymétrie.....	59
Annexe XI. Changement de navigation.....	60
Annexe XII. Changement de cap	61
Annexe XIII. Tracé de nettoyage.....	62
Annexe XIV. Matérialisation des périodes sans mesure	63
Annexe XV. Filtrage	64
Annexe XVI. Tracé de la dérive.....	65
Annexe XVII. Tracé 1D.....	66
Annexe XVIII. Tracé 2D.....	67
Annexe XIX. Comparaison	69
Annexe XX. Exploitation Section – Contourage	74
Annexe XXI. Exploitation Section – Vecteur	75
Annexe XXII. Exploitation Station – Vecteur	76
Annexe XXIII. Rapport des dépendances entre fonctions.....	78

1 Introduction générale

CASCADE est un ensemble de programmes matlab initié en 1998 au LPO pour ses besoins de recherche dans le traitement et l'analyse des mesures ADCP (Acoustic Current Doppler Profiler) de coque (Kermabon et Gaillard, 2001). Ce logiciel a depuis été mis en œuvre dans un mode opérationnel par le centre de données SISMER et est utilisé dans différents organismes tel que l'IRD et l'intitut Shirshov de Moscou. Il fait l'objet de mises à jour régulières afin de prendre en compte les remarques utilisateur ainsi que les suggestions d'amélioration.

Les données sont souvent réparties en quatre grandes catégories caractérisant leur niveau de traitement (ref : Rapport GTDN, 2002). Il en va de même des données ADCP. On distingue :

- Le niveau 0 : Ce sont les données brutes fournies par les différents systèmes d'acquisition, l'ADCP d'une part, le système de navigation/capteurs d'attitude d'autre part. Les courants sont la traduction directe de la mesure Doppler dans la direction des faisceaux.
- Le niveau 1 : Il s'agit des données traitées : fichiers 'Processed' de Transect, fichier 'Campagne' de CASCADE, fichiers .LTA ou .STA de VMDAS. Les courants sont des courants absolus ou relatifs, exprimés dans le repère géographique et géoréférencés. ils sont accompagnés des mesures annexes permettant d'évaluer ces données (amplitude de l'écho reçu, estimation de l'erreur, mesures d'attitude et de vitesse navire, vitesse sur le fond).
- Le niveau 2 : Ces données sont validées, corrigées, affectées d'une information de qualité ou de précision¹. C'est le produit standard le plus largement diffusé.
- Le niveau 3 : Les données de synthèse, préparées à partir du niveau 2 et dans lequel l'échantillonnage a été modifié.

Initialement, CASCADE comprenait 2 parties :

- Une partie traitement : Cette partie permet de générer des données de niveau 1 à partir des données ADCP de niveau 0. Actuellement, cette partie n'est nécessaire que pour des données ADCP acquises sous le logiciel d'acquisition RDI Transect qui ne permet pas de bénéficier en temps réel des meilleures données externes possibles (cap, roulis, tangage, ...etc). Le logiciel d'acquisition RDI VMDAS n'ayant pas cet handicap et se généralisant sur les navires, la partie traitement est amenée à disparaître à très court terme. La version en cours est la version 5.4.
- Une partie exploitation

qui fournit les outils nécessaires à la production de données de **niveau 2 et 3** à partir des données de **niveau 1**.

Le document présent est associé à la partie EXPLOITATION de CASCADE, version 6.1.

Dans son état actuel, CASCADE EXPLOITATION accepte les données produites par CASCADE traitement ou par VMDAS.

¹ Dans ce document, les anglicismes "flag", flagage et "flaguer" font référence à l'information de qualité associée à chaque donnée.

Un premier ensemble d'opérations de l'exploitation permet de compléter, valider et filtrer les données de courant fournies en entrée, sans en modifier l'échantillonnage. En particulier, il est possible de :

- Définir des indicateurs de qualité,
- De comparer les mesures de courant à la vitesse du navire
- De compenser un éventuel désalignement des transducteurs
- D'ajouter les données de bathymétrie le long de la route
- De calculer les courants corrigés de la marée barotrope
- De filtrer les données

Ces données de **niveau 2**, bien qu'ayant demandé une intervention humaine et sa part d'expertise, sont un produit standard qui sera utilisé par une large communauté. Ces fichiers sont le point de départ des analyses scientifiques ou opérationnelles.

Un second ensemble d'opérations de l'exploitation permet d'identifier des phases différentes dans la campagne en définissant des sections, l'abscisse n'est plus alors le temps mais la distance le long de la section, ou des stations dont l'échantillonnage temporel peut être redéfini. La construction de ces nouveaux fichiers fait intervenir une moyenne spatiale ou temporelle. On obtient alors des données de **niveau 3**. La définition de ces données dépend beaucoup de la finalité, chaque utilisateur produira ses propres fichiers.

CASCADE EXPLOITATION propose les outils de visualisations et d'analyse du contenu du fichier correspondant aux niveaux 2 et 3 de données.

2 Préparation de l'exploitation des données

2.1 Configuration minimum requise

Pour utiliser la partie EXPLOITATION de CASCADE Version 6.1, il faut disposer de :

- matlab version minimale 2008b.
- les toolbox, matlab m_map et timeplt accessibles via internet :
 - <http://www.eos.ubc.ca/~rich/map.html> pour m_map
attention de charger également les lignes de côtes haute résolution GSHHS : <http://www.ngdc.noaa.gov/mgg/shorelines/data/gshhs/> (fichier *.b) et les installer dans le répertoire m_map/private/.
La version m_map1.4c est livrée avec CASCADE et fonctionne bien. La dernière version (m_map1.4e) pose quelques problèmes.
 - <http://woodshole.er.usgs.gov/operations/sea-mat/> pour timeplt. Cette toolbox est également livrée avec CASCADE car elle a été modifiée pour être compatible avec les dernières versions de matlab (2010b).
- de fichiers de bathymétrie. Par défaut, le logiciel propose 4 choix de bathymétrie à charger séparément:
 - etopo2 : bathymétrie créée par l'équipe MERCATOR avec une résolution de 2 minutes
 - « Bathy6min » : bathymétrie créée à partir du fichier etopo2, en moyennant puis décimant tous les 3 points le fichier etopo2.
 - « Bathy18min » : fichier de bathymétrie à 18 minutes, extrait du fichier etopo2.
 - gebco : bathymétrie issue du CDRM GEBCO avec une résolution de 1 minute.

Si l'utilisateur le souhaite, il peut créer son propre fichier de bathymétrie. Celui-ci doit être au format NetCDF et contenir les variables (cf. Annexe VIII) :

- *latitude* de dimension *latitude*
- *longitude* de dimension *longitude*
- *z* (bathymétrie négative en m) de dimension (*latitude*, *longitude*)
- De fichiers de marée. Nous proposons d'utiliser les fichiers de modèle de marée générés par le modèle inverse de marée, basé sur les données satellites Topex/Poséidon, de G. Egbert et L. Erofeeva, qui sont disponibles sur internet (voir section 3.3.7).

2.1.1 - Données de départ

Pour lancer la partie EXPLOITATION de CASCADE Version 6.1, l'utilisateur doit disposer de fichiers d'entrée. Ces fichiers diffèrent selon le logiciel d'acquisition des ADCPs de coque.

2.1.2 - Pour les navires équipé de TRANSECT

Les données brutes issues de TRANSECT doivent avoir préalablement été traitées par la partie traitement de CASCADE Version 5.4. Le traitement a produit un fichier campagne appelé <nom_fic>.nc. L'utilisateur doit convertir ce fichier NetCDF en un fichier NetCDF à la norme Oceansite, conforme à la partie EXPLOITATION. Cette conversion peut se faire via la fonction disponible avec CASCADE : *conv_old_new*. Le nom du fichier ainsi converti (qui devient le fichier de travail) a la nomenclature suivante :

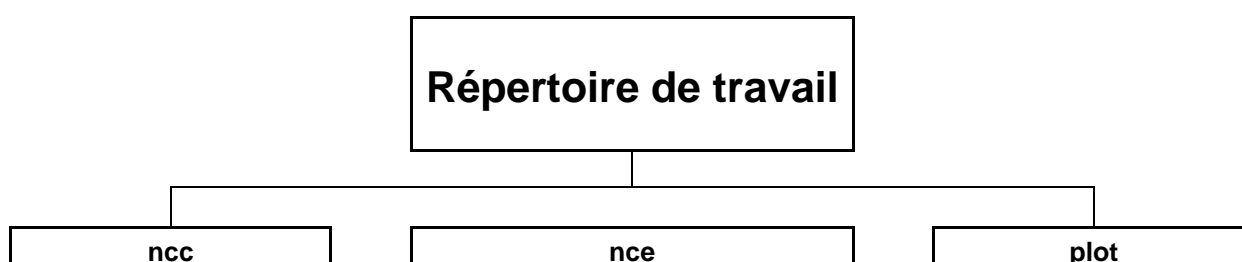
<nom_fic_osite>.nc

2.1.3 - Pour les navires équipés de VMDAS

L'utilisateur doit avoir à disposition tous les fichiers *.LTA (ou *.STA) de la campagne. Il est alors nécessaire de les concaténer dans le répertoire de travail, dans l'**ordre chronologique** (via la commande cat, par exemple, sous linux). L'utilisateur indique le fichier ainsi concaténé en entrée. Un fichier au format NetCDF à la norme OceanSite est créé et devient le fichier de travail. Un tracé de trajectoire permet de vérifier que toutes les données ont bien été prises en compte (Annexe IX)

2.2 Arborescence de travail recommandée

Il est recommandé d'opter pour l'arborescence de travail suivante :



Le répertoire de travail comprend :

- Les fichiers *.LTA (ou *.STA) pour les données acquises via le logiciel VMDAS. Ces fichiers peuvent être aussi dans un sous-répertoire à part.
- Les fichiers ASCII définissant les stations (*_sta.list) et sections (*_sec.list) de la campagne
- trois sous-répertoires :
 - ncc : contient :
 - le fichier campagne au format NetCDF généré par la partie traitement de CASCADE Version 5.4 (pour les données acquises via le logiciel TRANSECT)
 - le fichier de travail (au format NetCDF à la norme OceanSite)
 - les divers fichiers campagne générés lors de l'exploitation des données.
 - nce : contient les fichiers NetCDF associés aux sections et stations de la campagne.
 - plot : comporte tous les tracés créés par l'exploitation.

Si ces 3 sous-répertoires n'existent pas dans le répertoire de travail, CASCADE les crée automatiquement.

Le répertoire de travail comprend également le fichier conf_exploit.mat. Ce fichier est généré par CASCADE et mémorise les divers paramètres saisis par l'utilisateur. Dans un premier temps, il contient des valeurs par défaut. Ensuite, il comprend les valeurs saisies par l'utilisateur dans la session CASCADE précédente.

2.3 Arborescence de développement

- **/exploitation** : Ce répertoire est le « répertoire père » du logiciel CASCADE.
- **/exploit** : Ce répertoire contient toutes les fonctions de calcul propres à CASCADE.
- **/ihm** : Contient les fonction plus particulièrement dédiées à l'interface utilisateur ainsi que les logos.
- **/tools** : répertoire comprenant des fonctions générales utilisées par CASCADE (calcul du jour julien, de la moyenne sans prise en compte des NaN, ..etc).
- **/bathymetrie** : répertoire utilisé par défaut pour lire les fichiers de bathymétrie. De base, il contient 4 fichiers de bathymétrie (cf. 2.1). L'utilisateur peut y ajouter ses propres fichiers de bathymétrie (attention au format à respecter). Si l'utilisateur le souhaite, il peut aussi définir, via l'interface, un autre répertoire à partir duquel CASCADE ira chercher les fichiers de bathymétrie.
- **/tide** : répertoire contenant les fonctions utiles à l'ajout de la marée ainsi qu'un sous-répertoire model comprenant les 2 fichiers de marée par défaut (cf. 2.1). A noter que l'utilisateur peut définir, via l'interface, son propre répertoire de marée dans lequel il devra donc déposer ses fichiers de marée.

3 Exploitation des données

Pour lancer CASCADE, sous matlab, l'utilisateur doit tout d'abord modifier le path de matlab afin de pouvoir accéder aux fonctions matlab développées pour CASCADE.

Soit CHEMIN_CASCADE le répertoire comprenant le logiciel CASCADE EXPLOITATION, ceci se fait via la commande matlab : **addpath(genpath(CHEMIN_CASCADE))**

Pour lancer CASCADE, sous matlab, l'utilisateur doit ensuite se placer dans son répertoire de travail et lancer l'application via:

- **ce** pour lancer la version française
- **ce('en')** pour lancer la version anglaise

Pour optimiser les performances liées à l'utilisation de la mémoire, il est possible de lancer matlab sans l'environnement java (matlab -nojvm).

L'interface se présente de la façon suivante :

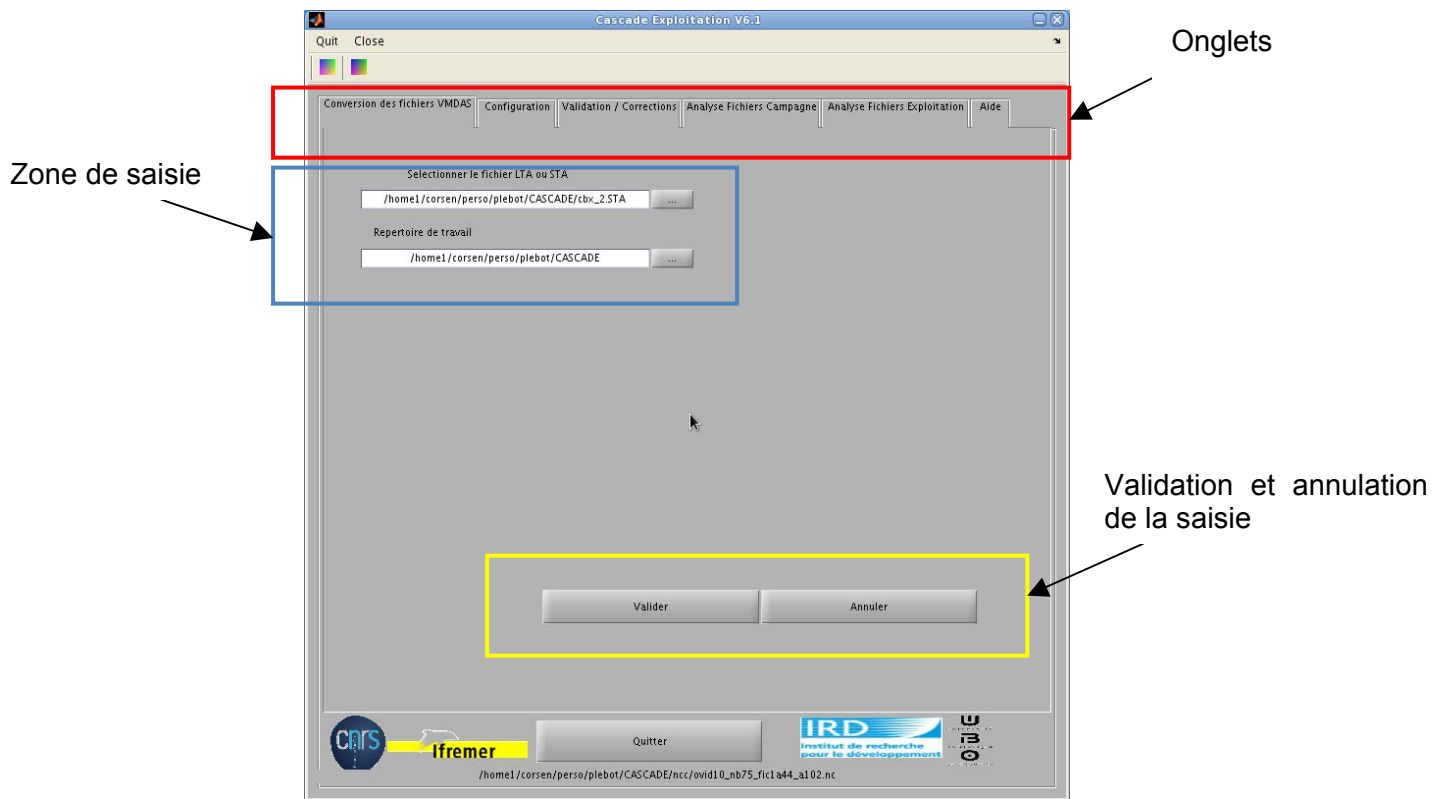


Figure 1 – Interface CASCADE EXPLOITATION

Si les textes se chevauchent dans l'interface (suivant la taille de l'écran), l'utilisateur peut jouer sur la propriété DefaultUicontrolFontSize via la commande, sous matlab :

`set(0,'DefaultUicontrolFontSize',7)` : l'utilisateur ajustera la taille (7) en fonction de son écran.

Dans la partie supérieure de la fenêtre, les onglets décrits ci-dessous permettent de passer d'une section à une autre.

1. Onglet « Conversion des fichiers VMDAS »

Ce choix permet la conversion des fichiers issus du logiciel d'acquisition VMDAS. L'utilisateur indique son répertoire de travail, où les fichiers générés par CASCADE seront écrits, ainsi qu'un fichier LTA ou STA concaténé ; celui-ci est converti au format NetCDF à la convention OceanSite (*format en annexe II*).

2. Onglet « Configuration »

Cette étape permet de configurer l'environnement de travail, de sélectionner le répertoire de bathymétrie et de marée, ainsi que le fichier de campagne (format NetCdf, osite) à exploiter.

3. Onglet « Validation/Correction

Cette étape travaille à partir du fichier NetCDF à la norme OceanSite. Elle permet de modifier le fichier campagne existant (ajout de flag de qualité sur les données, ajout de données extérieures (marée, bathymétrie)) et/ou d'en créer de nouveaux (générés via une correction de l'attitude du navire, un filtrage, ...etc).

4. Onglet « Analyse Fichiers Campagne »

Cette étape sert à visualiser les variables des fichiers campagne NetCDF OceanSite ainsi qu'à la création des fichiers d'exploitation (sections, stations). Ces derniers sont mémorisés dans le sous-répertoire nce. Les fichiers campagne ne sont pas modifiés à cette étape.

5. Onglet « Analyse Fichiers Exploitation »

Cette étape travaille sur les fichiers NetCDF (sections et/ou stations) créés au moment de l'analyse des fichiers campagne. Elle permet de visualiser les variables de ces fichiers

6. Onglet « Aide »

Cet onglet permet à l'utilisateur d'accéder à tout moment lors du traitement à diverses informations concernant les flags de qualité ou encore, à diverses informations concernant le fichier de campagne (traitements appliqués, % données bonnes,...).

CASCADE EXPLOITATION démarré, l'interface s'ouvre sur la fenêtre « configuration ».

Remarque : Si dans le répertoire d'où est lancé le logiciel, aucun fichier conf_exploit.mat n'est trouvé, CASCADE le crée automatiquement avec des valeurs par défaut. Il est à noter que les valeurs par défaut proposées à l'utilisateur sont correctes pour la majorité des campagnes. L'utilisateur doit néanmoins les modifier à son gré suivant sa connaissance de la région des données.

Les valeurs sont sauvegardées pour chaque session, aussi, au prochain lancement du logiciel dans le répertoire de travail, l'utilisateur retrouve les valeurs qu'il a saisies dans la session précédente.

3.1 Conversion de fichiers VMDAS :

L'utilisateur doit saisir :

- le fichier <nom_fic.LTA> (ou <nom_fic.STA>) (issu de la concaténation chronologique des fichiers *LTA (ou *STA) de la campagne)
- répertoire de travail

En sortie, un fichier NetCDF au format OceanSite dont la nomenclature est : <nom_fic_osite>.nc est généré dans le sous-répertoire ncc du répertoire de travail. Il devient le fichier de travail. (cf. *Annexe III*).

Au cours de cette étape, les vitesses absolues du courant sont calculées à partir des vitesses relatives du fichier et des vitesses navire estimées à partir des positions et dates de début et fin de chaque ensemble.

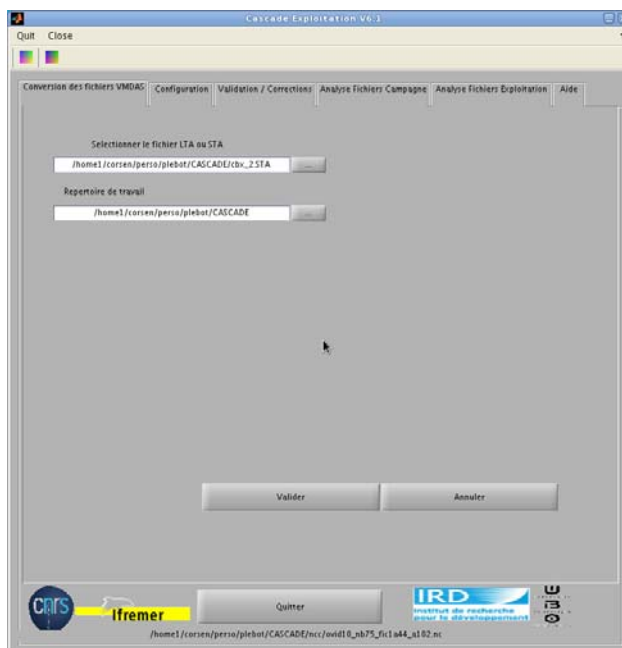


Figure 2 – Interface conversion de fichiers VMDAS

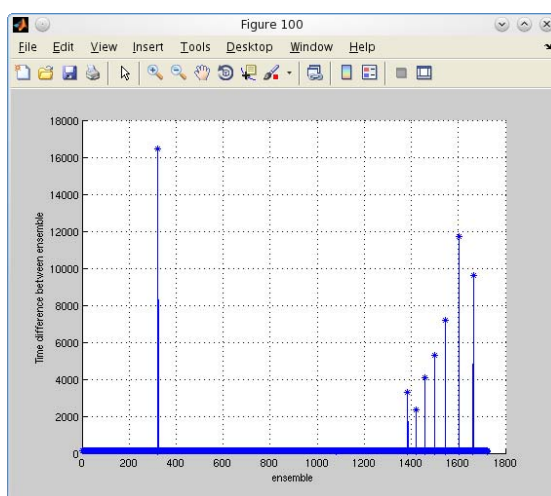


Figure 3 - Différence de temps entre les ensembles

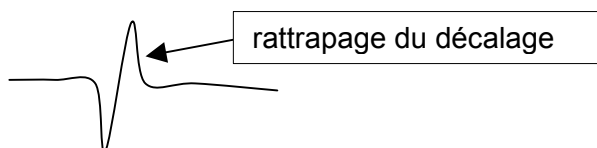
La "figure 100" présentée ci-dessus représente la différence de temps entre les ensembles. Comment interpréter cette figure:

Les pics positifs indiquent un arrêt de l'ADCP sur une durée indiquée en ordonné (durée en seconde). Ces arrêts peuvent correspondre aux stations hydrographiques.

Si des valeurs négatives apparaissent sur ce tracé, cela indique un décalage de l'horloge (un retour en arrière). La lecture des points suivants permet de déterminer si il s'agit d'un saut ponctuel suivi d'un recalage de l'horloge ou d'un décalage répercuté sur les données suivantes.

Dans le cas d'un rattrapage du décalage de l'horloge, un pic positif doit suivre la valeur négative.

Schématisation d'un décalage suivi du recalage de l'horloge.



Notez qu'une erreur de ce type est automatiquement corrigée si elle ne concerne qu'un seul point (date mise en Fill_Value) et n'apparaîtra pas sur la figure.

Dans le cas d'un retour en arrière de l'horloge sans recalage de celle-ci, la valeur négative n'est pas suivi du rattrapage (pic positif); l'utilisateur est alors invité à contrôler ces données afin de déterminer la raison de ce retour en arrière dans le temps (ordre des fichiers STA, changement d'heure,...).

Schématisation d'un décalage sans recalage de l'horloge.



3.2 Configuration

Cette fenêtre est celle qui est lancée au démarrage du logiciel CASCADE.

Pour accéder à cette interface au cours d'une session, l'utilisateur cliquera sur l'onglet « configuration » .

Cette fenêtre de configuration permet à l'utilisateur de sélectionner le nom de son répertoire de travail, le nom du fichier qu'il souhaite traiter (typiquement, le fichier issu de l'étape de conversion ou le fichier généré par CASCADE traitement converti au format OceanSite via la fonction conv_old_new), les répertoires dans lesquels se trouvent ses fichiers de bathymétrie et de marée.

Cette interface permet également de renseigner des paramètres dans le fichier Netcdf tels que le nom de la campagne à traiter et le nom du navire sur lequel celle-ci a été réalisée (par défaut, ces paramètres sont positionnés à « Unknown »).

Pour les sorties graphiques, deux formats sont proposés à l'utilisateur, postscript (extension PS) ou Portable Network Graphic (extension PNG). Cet extension est notée <ext> dans la suite du document.

Une fois l'ensemble des champs renseigné, l'utilisateur doit cliquer sur valider pour que ces choix soient pris en compte, le logiciel passe alors automatiquement à l'onglet suivant « validation/corrections ».

Une fois validé, le nom du fichier courant s'affiche dans le bas de la fenêtre.

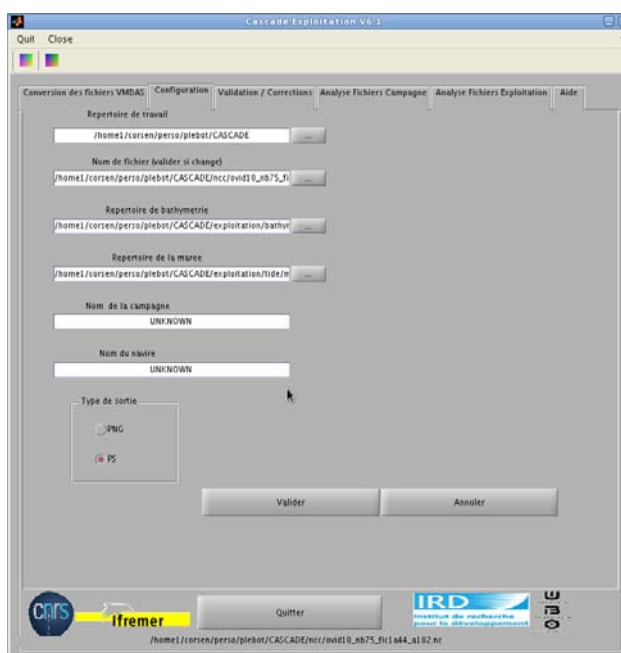


Figure 4 – Fenêtre de configuration fichier et répertoires

A chaque étape du traitement, tant que le bouton « Valider » n'a pas été sélectionné, il est possible de revenir au paramètre par défaut en cliquant sur « Annuler ».

Pour garder une compatibilité avec les version antérieures de Cascade, un test est effectué sur les anciens fichiers de campagne pour vérifier l'existence de la variable FLAG_MIN_CORR (seuil de corrélation). Si cette variable n'existe pas dans le fichier (fichier antérieur à la V6.1 de Cascade), l'utilisateur est averti de la création de ce nouveau flag dans le fichier de campagne.

NB : Les étapes suivantes ne peuvent pas être effectuées sur un format autre que Ocean Site. C'est pourquoi l'utilisateur est averti par un message d'alerte si le nom du fichier Netcdf n'est pas au format oceanSite,.

3.3 Validation Correction

L'écran Validation/Correction est montré Figure 5. Par défaut, la fenêtre « définition de la couche de référence » est sélectionnée.

Les paragraphes suivants décrivent en détails les différents choix possibles du tableau de bord (cf : fig.4) pour valider et corriger les données. Pour identifier rapidement le module en cours, dans le tableau de bord, le bouton de l'opération concernée s'affiche en rouge dans l'interface.

3.3.1 - Définition de la couche de référence

La couche de référence sert de base à la définition de certains critères (nettoyage, comparaison route-station).

Elle est définie par les numéros de cellule ("bin") MIN et MAX entre lesquelles les données sont, à priori, considérées comme bonnes et cohérentes entre elles.

Il ne faut pas prendre en compte la couche de surface et les cellules les plus profondes. Les cellules 3 à 8 sont généralement appropriées.

Il faut saisir les valeurs et cliquer sur « Valider ».

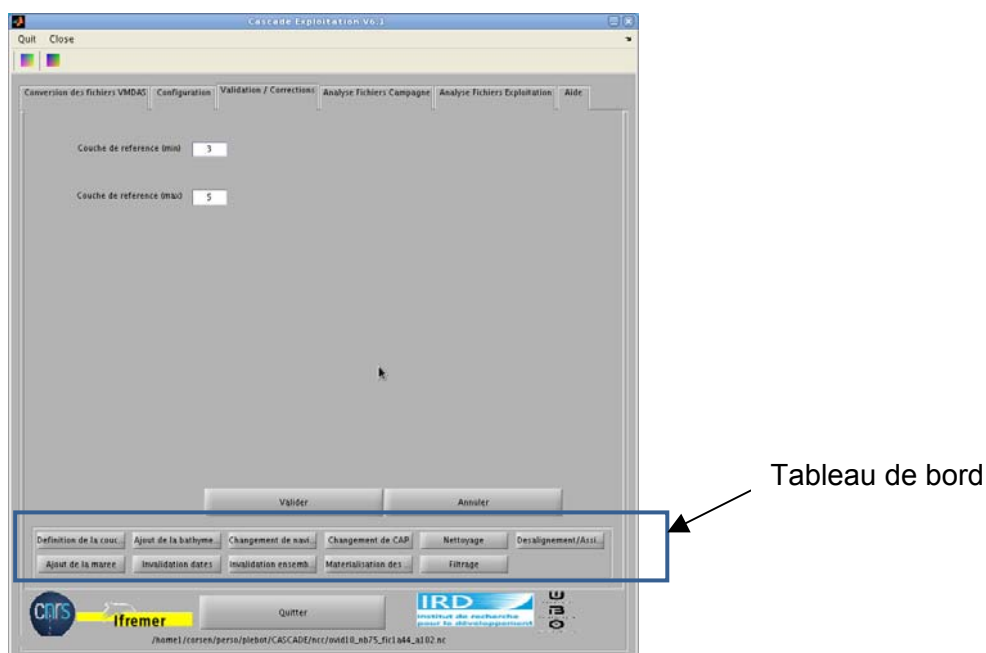


Figure 5 – Validation / corrections (1) – couche de référence

3.3.2 - Ajout de la bathymétrie

Cette interface permet de choisir le type de bathymétrie via une liste déroulante.

Par défaut les bathymétries suivantes sont proposées :

- Gebco (résolution de 1 minute)
- Etopo2 (résolution 2 minutes)
- Bathy6min (résolution 6 minutes)
- Bathy18min (résolution 18 minutes)
- l'utilisateur peut ajouter ses propres fichiers NetCDF de bathymétrie (cf. 2.1) et les placer dans le répertoire de bathymétrie. Ces nouveaux fichiers apparaîtront alors dans la liste déroulante.

Plot de contrôle : Cette option permet à l'opérateur de visionner la carte de la campagne avec le tracé de la bathymétrie (*Annexe X*).

Contourage : Si cette option est cochée, sur le plot de contrôle, la bathymétrie est représentée par des lignes de contourage (contourage plein sinon).

Cette étape associe une bathymétrie à la trajectoire de la campagne ADCP.

Une variable BATHY est ajoutée dans le fichier de travail. Celle-ci pourra servir à la détection du fond lors du nettoyage.

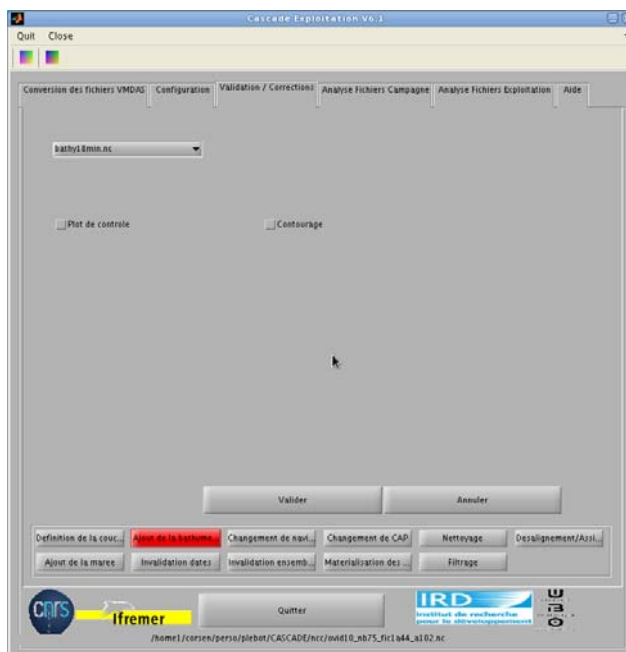


Figure 6 – Ecran ajout de la bathymétrie

3.3.3 - Changement de navigation

Cette interface permet de recalculer les vitesses absolues à partir d'une autre navigation. Par défaut, CASCADE calcule les vitesses absolues du courant à partir des vitesses de navires issues des positions et dates de début et de fin de chaque ensemble, navigation notée GPS.

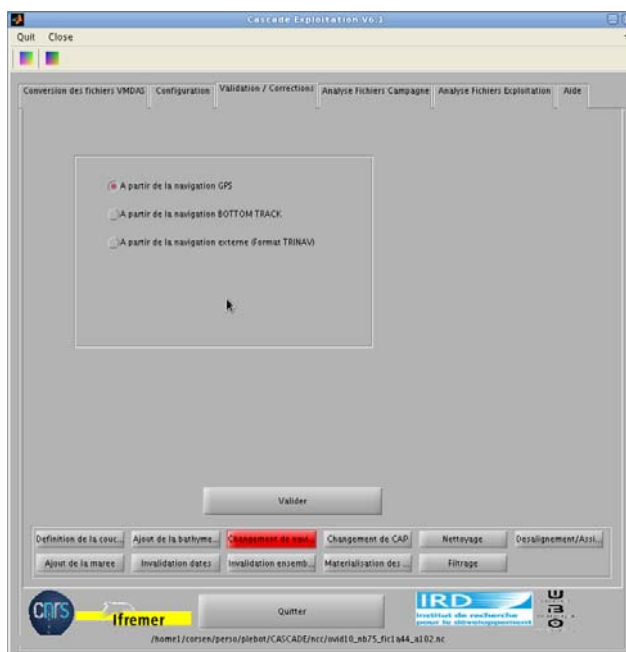


Figure 7 – Ecran changement de navigation

L'utilisateur peut recalculer les vitesses absolues du courant à partir des vitesses du fond (issues du bottom-tracking), si disponible, du fichier à traiter ou à partir d'un fichier externe au format TRINAV qu'il peut lui même créer à partir d'autres sources de navigation dont il

dispose (Annexe IV). Dans ce cas, les variables U_EXTERNAL et V_EXTERNAL sont alors ajoutées dans le fichier campagne

Sélectionner « A partir de la navigation externe... » pour faire apparaître le champ de saisie du nom de fichier.

Les vitesses absolues du courant avant et après prise en compte du changement de navigation sont visualisées.

Remarque : Les RMS sont associés aux vitesses absolues calculées avec la vitesse navire ET le cap GPS (calculés dans CASCADE traitement). Ces variables ne sont pas modifiées lorsque l'on recalcule des vitesses absolues à partir d'une autre navigation. Les RMS ne correspondent plus alors à ce que l'on a dans le fichier (Annexe XI).

3.3.4 - Changement de cap

A cette étape, il est possible de recalculer la vitesse absolue à partir d'un cap autre que celui utilisé par le logiciel d'acquisition, noté GPS. L'utilisateur peut recalculer les vitesses à partir des caps G1 et G2 du fichier si disponible. Il a également la possibilité de les recalculer à partir d'un fichier de cap externe qu'il doit lui même créé (format en Annexe V) à partir des diverses sources de cap dont il peut bénéficier. La variable HDG_EXTERNAL est alors ajoutée dans le fichier campagne. Les vitesses absolues du courant avant et après la prise en compte de ce changement de cap sont visualisées. Un plot de comparaison entre le cap externe et le cap externe interpolé aux dates ADCP est également tracé.

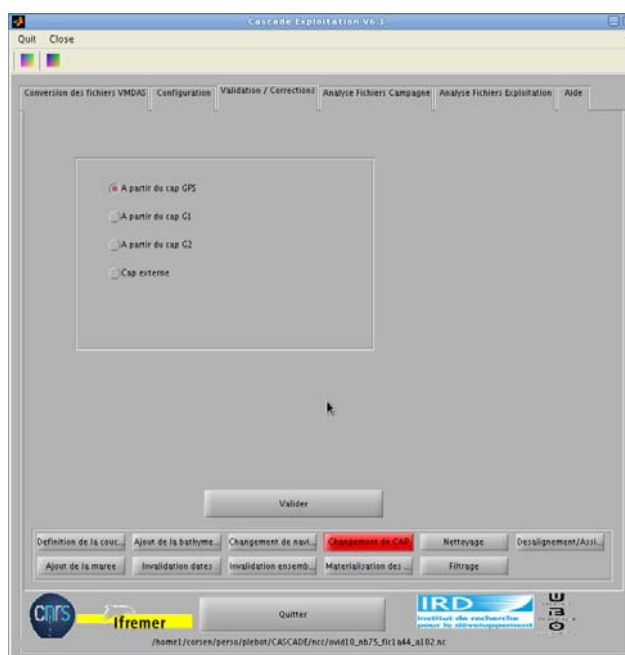


Figure 8 – Ecran changement de cap

Remarque : Les RMS sont associés aux vitesses absolues calculées avec la vitesse navire ET le cap GPS (calculés dans CASCADE traitement). Ces variables ne sont pas modifiées lorsque l'on recalcule des vitesses absolues à partir d'un autre cap. Les RMS ne correspondent plus alors à ce que l'on a dans le fichier (Annexe XII).

3.3.5 - Nettoyage

Le nettoyage permet d'associer un flag de qualité aux données de vitesse du courant en fonction de certains critères. Chaque critère positionne les flags sans tenir compte de leurs valeurs précédentes.

Les valeurs des flags affectées aux données sont les suivantes :

Valeur	Explication – Commentaires	Variable associée
1	Données bonnes	
2	Données douteuses (données relatives aux cellules dont l'une des composantes horizontales (U ou V) diffère trop des 5 voisins horizontaux et verticaux, ou points isolés). Lorsque plus de 50% de la couche de référence est flaguée incorrecte (à 2 ou plus), tout le profil sous le premier point douteux est flagué à 2.	Vdifflim fact_cis cis_max_u
3	Filtre médian sur N_fl3 ensembles au-delà de X_fl3 écarts-types.	N_fl3 X_fl3
4	Cellules dont l'une des composantes horizontales a un cisaillement vertical différentiel $> X$ (s^{-1}). L'histogramme des cisaillements tracé en début de nettoyage permet de déterminer la valeur X, qui diffère considérablement d'une région à l'autre de l'océan.	cis_max
5	Cellules dont l'erreur de vitesse du courant $> X$ cm /s	w_max
6	Cellules dont l'une des vitesses absolues horizontales (U ou V) $> X$ m/s	v_max
7	Données absentes	
8	Détection du fond par le bottom-tracking ou la bathymétrie selon le choix de l'utilisateur.	
9	Données invalidées entre 2 dates ou entre 2 ensembles par l'utilisateur	

En gras, dans le tableau, sont indiquées les valeurs à définir par l'utilisateur.

Ce programme remplit le tableau des flags (donc écrase tout flagage antérieur).

Le nettoyage est effectué en trois étapes :

1. Une première fonction (Net_vit_flag4a8) positionne les flags des plus mauvaises données (du flag 4 au flag 8).
2. Les données jugées valides à l'issue de ce premier nettoyage sont ensuite comparées, niveau par niveau aux **N_fl3** données voisines, de part et d'autre (Net_vitesse). Un critère basé sur l'écart median (**E_med**) à la médiane (**med**) est appliqué : les données en dehors de $[med - X_fl3 * E_med, med + X_fl3 * E_med]$ sont flaguées à 3.
3. La dernière partie (Net_vit_flag2) repère les points douteux qui pourraient rester (flags 2) en considérant les profils. Elle procède de la façon suivante :
 - La moyenne des 11 profils encadrant le profil considéré est effectuée, elle est suivie d'une moyenne glissante verticale sur 5 points. Ceci fournit le profil moyen de l'environnement. On calcule ensuite le cisaillement vertical de ce profil moyen et on en retient le maximum **cis_max_u**. L'écart maximal **vdifmax** autorisé entre le profil considéré et le profil moyen est calculé comme : $\max([fact_cis * cis_max_u, vdifflim])$. La fonction est actuellement réglée avec des valeurs par défaut :


```
vdifflim = 30 (cm/s);
fact_cis = 4;
```
 - Ensuite on flague aussi à 2 les singulets et doublets isolés.

Deux modes de nettoyage sont proposés à l'utilisateur, un mode standard et un mode expert permettant de définir plus ou moins de paramètres.

Attention, les défauts de nettoyage sont sévères. Ils ont été ajustés pour les NB de l'Atalante et de la Thalassa (40 ensembles, 2.5 écarts-type). Pour le BB on doit réduire à 20 ensembles et 3 écarts-type sinon de bonnes données sont éliminées en zone équatoriale.

2 modes : standard et expert

Mode standard

Mode expert

Figure 9 – Ecran nettoyage

Quand les valeurs sont saisies, cliquer sur « valider »

Différents graphes s'affichent à l'écran, pendant que le nettoyage s'effectue :

- Un graphe représentant le Nettoyage MEDIAN de U, et le Nettoyage MEDIAN de V (en vert = U/V = bonnes données, en bleu = médiane, en rouge = ce qui a été flagué à 3 (éliminé), en noir = mesures de dispersion autour de la médiane). Ce graphe s'affiche cellule par cellule.
- Si des interférences sont détectées, un graphe représentant les interférences en fonction du nombre d'ensembles s'ouvre. Les données affichées en rouge sont les données enlevées.
- Un graphe représentant l'écho intensité en fonction du nombre d'ensembles avec et sans interférence.

Le premier tracé (histogramme de cisaillements) est sauvegardé sous le nom de fichier suivant : `<nom_fic>_histo_cisaill.<ext>`. Certains des tracés générés sont sauvegardés sous : `<nom_fic>_f4_8.<ext>` (Annexe XIII)

Quand le nettoyage est terminé, un bilan est affiché dans la fenêtre MATLAB. Dans ce bilan, il faut vérifier que le nombre d'erreurs détectées semble réaliste. Ces valeurs sont sauvegardées et peuvent être visualisées en allant dans le menu : Aide → Infos Campagne

A l'issue du Nettoyage, toutes les valeurs mesurées ont un flag qualité.

3.3.6 - Correction de Désalignement/Assiette/Amplitude

Lorsque la moyenne de la vitesse verticale (pour les ADCPs autre que les OceanSurveyor) calculée sur toutes les bonnes données (flaguées à 1) est trop élevée ($>1\text{cm/s}$) ou quand la vitesse perpendiculaire du courant semble corrélée à la vitesse du navire (voir 3.4.4 et 3.6.2), il peut être nécessaire de passer par cette étape. Celle-ci permet de corriger les vitesses du courant d'une éventuelle erreur sur l'angle de l'ADCP par rapport à l'axe du navire ou sur l'assiette du navire (tangage) ou encore d'une erreur sur le facteur d'amplitude. L'écran correspondant à cette étape est le suivant :

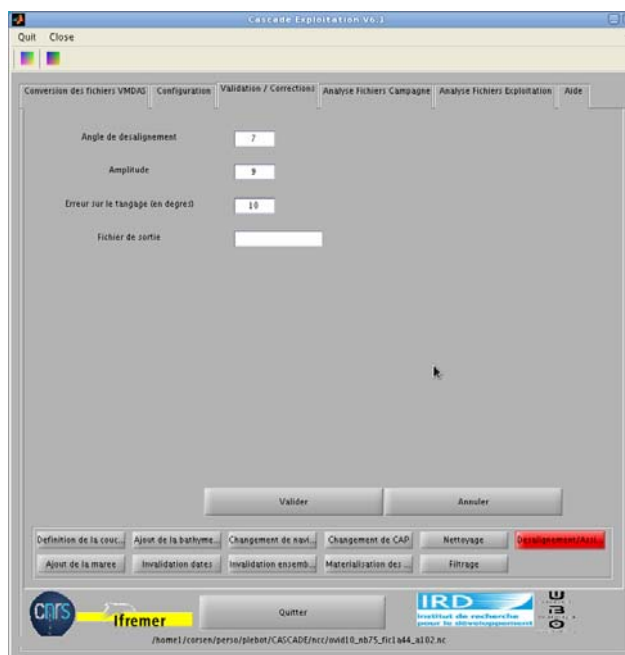


Figure 10 – Ecran de correction de Désalignement/Assiette/Amplitude

Lorsque l'on valide, la correction est lancée, et la nouvelle valeur de la moyenne de la vitesse verticale est donnée.

Un nouveau fichier campagne au format NetCDF est créé dans le répertoire ncc : `<nouveau_nom_fic>.nc` ; c'est ce fichier qui doit désormais être utilisé pour le reste de l'exploitation, mais il faut le sélectionner dans l'onglet de Configuration.

Remarques :

- ❖ *Le désalignement agit sur la vitesse perpendiculaire du courant.*
- ❖ *L'amplitude agit sur la vitesse parallèle du courant.*
- ❖ *L'erreur sur le tangage agit sur la vitesse verticale du courant, mais aussi sur la vitesse horizontale. Il n'est pas conseillé de changer ce paramètre pour les Ocean Surveyor, car la vitesse verticale moyenne ne permet pas de valider le changement.*

3.3.7 - Ajout de la marée

Ce module permet d'ajouter les variables correspondant aux composantes de la vitesse, corrigée de la marée barotrope. Cette étape doit être réalisée sur le fichier nettoyé, avant filtrage.

Sont ajoutées dans le fichier de travail les variables :

- vitesses horizontales de la marée (U_TIDE et V_TIDE)
- transports de marée (TU_TIDE et TV_TIDE)
- vitesses du courant corrigées de la marée (UVEL_ADCP_CORTIDE et VVEL_ADCP_CORTIDE)

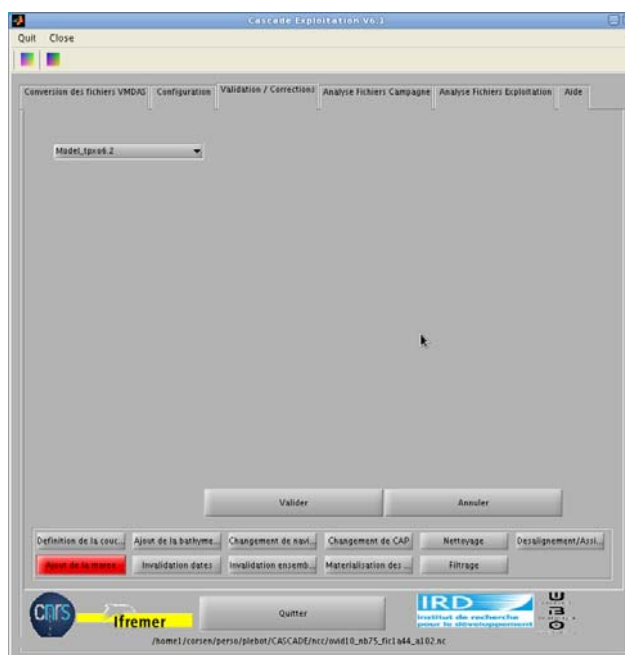


Figure 11 – Ajout de la marée

Les programmes nécessaires ont été récupérés sur le site de l'Oregon State University et sont déjà implémentés dans Cascade (avec leur autorisation). Le fichier binaire de données (tpxo.tar.Z ; ne pas prendre le fichier NetCDF) ainsi que les références à citer si on l'utilise sont disponibles sur internet :

<http://volkov.oce.orst.edu/tides/global.html>

Nous utilisons la solution « GLOBAL INVERSE SOLUTION TPXO » définie sur la région 90.125S-90.125N, 0.125E-360.125E, avec une résolution de $\frac{1}{4}$ degré et qui comprend 10 composantes : M2, S2, N2, K2, K1, O1, P1, Q1, Mf, Mm, estimées à partir des mesures TOPEX/Poseidon et Jason. Attention à corriger le chemin des fichiers dans Model_tpxo.X.X.

Il faut être prudent concernant l'évaluation de la marée près des côtes (dans les zones peu profondes). En effet, le modèle de marée est basée sur les transports et détermine les vitesses de marée à partir de la bathymétrie du modèle de marée dont la résolution n'est pas suffisante pour ces zones.

Contrairement à la bathymétrie, l'utilisateur peut pas placer ses propres fichiers de marée puisque le format requis est binaire. La marée TPXO pouvant ne pas convenir, il est envisagé (pour une prochaine version) de créer un outil permettant d'exporter les dates et positions en ascii, et d'importer les variables U_TIDE et V_TIDE correspondantes (calculées en externe) dans le fichier campagne.

3.3.8 - Invalidation de données entre 2 dates

La visualisation des tracés 2D (cf. 3.4.2) a pu permettre de mettre en évidence des zones où les données doivent être invalidées car incorrectes. Cette étape permet d'invalider ces données.

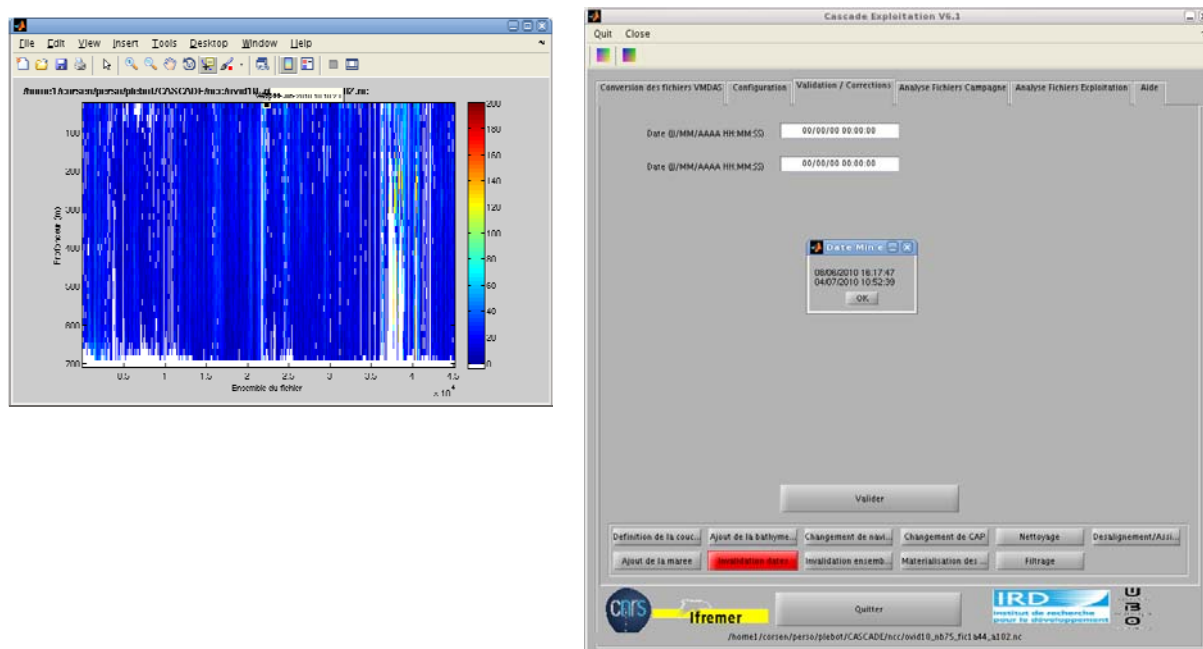



Figure 12 – Ecran invalidation de données entre 2 dates

Pour invalider les données (et leur associer un flag de qualité 9), il faut procéder de la façon suivante :

- Le graphe des vitesses horizontales du courant (associées aux flags 1 et 2) en fonction des numéros d'ensembles s'affiche, en plaçant le pointeur de la souris sur le tracé, les dates et l'heure correspondant à l'ensemble sélectionné s'affiche sur la figure. Il est possible de zoomer jusqu'à pouvoir cliquer sur le ou les numéros d'ensembles à supprimer, dans ce cas, la date ne s'affiche plus sur le tracé, pour la faire réapparaître il suffit de cliquer sur le bouton « Data Cursor Mode »  dans la barre menu de la figure.
 - Quand le zoom est OK, se positionner dans la fenêtre principale de CASCADE et saisir la date et l'heure de l'ensemble au bon format (jj/mm/aaaa hh:mm:ss).
 - Les deux opérations précédentes sont à reproduire pour sélectionner le dernier ensemble à invalider.
- Après vérification de ces dates, il faut cliquer sur Valider pour invalider les données entre ces 2 dates, un message de confirmation d'invalidation apparaît. **Attention, si les dates sont fausses et que l'utilisateur les a validées, il faut recommencer tout le traitement à partir du nettoyage des données afin de re-valider ces dates.** Pour sortir sans invalider de données, il faut sélectionner une autre opération dans le tableau de bord ou changer d'onglet suivant l'étape désirée.
 - L'opération est répétée si plusieurs ensembles de données doivent être supprimés.

3.3.9 - Invalidation de données entre 2 ensembles



Figure 13 – Ecran invalidation de données entre 2 ensembles

On rentre les ensembles de début et de fin de la période que l'on souhaite invalider puis on clique sur Valider, un message de confirmation d'invalidation apparaît. Les données associées à cette période ont un flag de qualité de 9. **Attention, si erreur dans la saisie des ensembles puis validation, il faut recommencer tout le traitement à partir du nettoyage des données afin de re-valider ces ensembles.**

3.3.10 - Matérialisation des périodes sans mesure

Il s'agit ici de traiter le cas où les mesures se sont arrêtées pendant un certain laps de temps au cours de la campagne. Ce module va insérer des enregistrements vides (ne contenant que la date) dans le fichier campagne, afin que visuellement, sur les tracés, on détecte les périodes trop importantes sans données. Sans cela, sur les contourages de matlab, ces périodes n'apparaissent pas clairement. La matérialisation des périodes sans mesure a pour effet que les trous trop importants sont représentés par une zone blanche dans les tracés matlab (contourage).

L' écran suivant s'affiche :

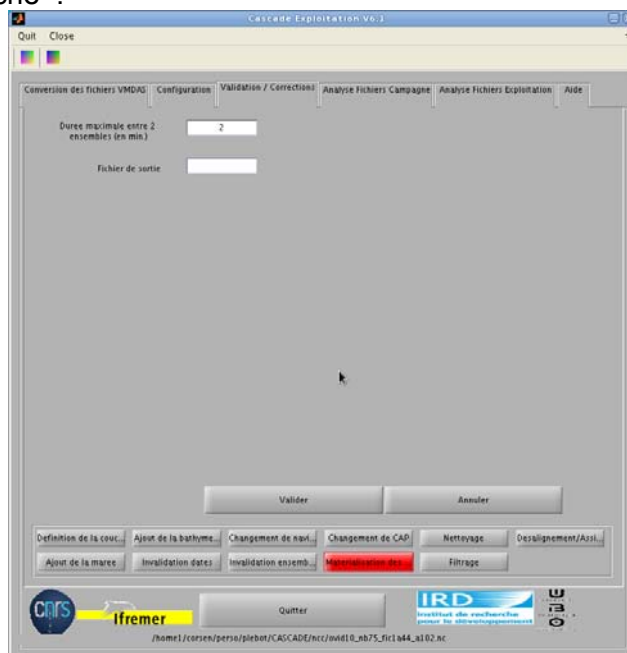


Figure 14 – Ecran de matérialisation des périodes sans mesure

L'utilisateur indique la durée maximale entre 2 données qu'il autorise ainsi que le nom du fichier de sortie. C'est ce fichier qui doit ensuite être choisi pour la suite de l'exploitation via l'onglet de Configuration. (Annexe XIV)

3.3.11 - Filtrage

Cette option permet un filtrage linéaire ($1/4[\text{valeur précédente}] + 1/2[\text{valeur courante}] + 1/4[\text{valeur suivante}]$) des vitesses et des flags, en fonction des flags de qualité, selon la verticale (profondeur) et/ou l'horizontale (temps). Les trous ainsi comblés (d'une longueur maximale de 2) sont flagués à 2. L'écran correspondant est le suivant :

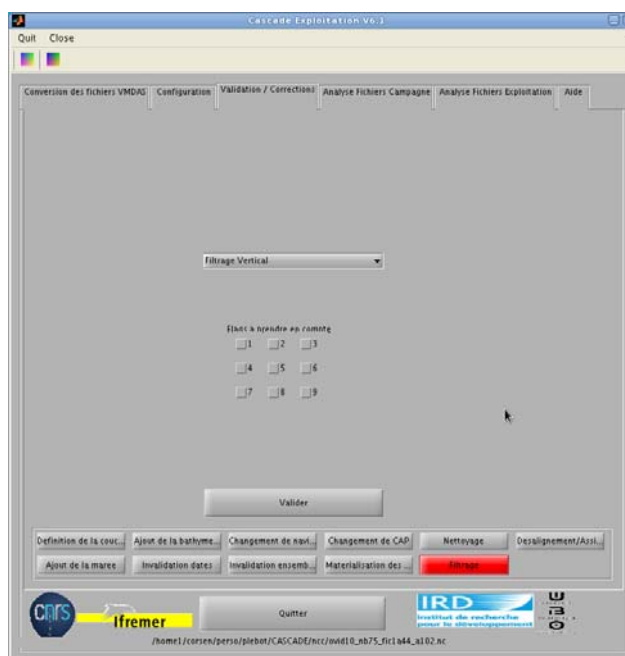


Figure 15 – Ecran du filtrage

Il faut choisir le type de filtrage (Horizontal (fonction du temps) et/ou vertical (fonction de la profondeur)), et les flags à prendre en compte lors de ce filtrage.. Le filtrage se lance lorsque que l'utilisateur appuie sur le bouton Valider. Deux graphes 2D s'affichent à l'écran :

- Les valeurs des flags et des composantes de la vitesse horizontale (U,V) et verticale (W) avant filtrage
- Les valeurs des flags et des composantes de la vitesse horizontale (U,V) et verticale (W après filtrage)

Un nouveau fichier campagne au format NetCDF est créé dans le sous-répertoire ncc :

`<nom_fic>_f< type_filtrage ><numéro_flag>.nc`

Le f signifie que le fichier est filtré.

Le type de filtrage est soit :

- h pour horizontal
- v pour vertical
- vh pour filtrage vertical et horizontal

Le numéro_flag indique la liste des flags pris en compte pour le filtrage. C'est le fichier ainsi généré qui devra être utilisé dans la suite de l'exploitation.

Cette étape génère de plus un fichier plot dont le nom se compose ainsi :

`<nom_fic>_filtr_<type_filtrage>.<ext> (Annexe XV)`

3.4 Analyse Fichiers campagnes

Cette étape permet d'analyser le fichier campagne et de créer les fichiers NetCDF section et station.

Lorsque l'onglet «Analyse Fichiers campagnes» est sélectionné, la fenêtre suivante s'affiche :

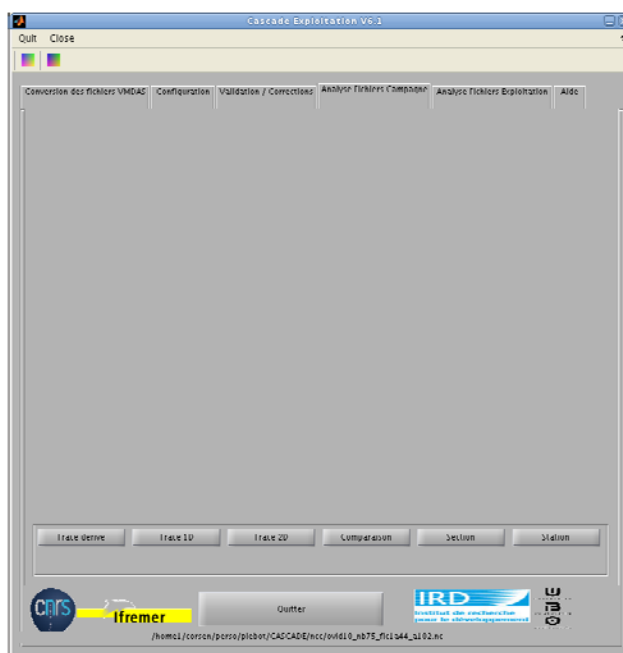


Figure 16 – Ecran analyse fichiers campagne

3.4.1 - Tracé de la dérive

Cette étape génère un fichier plot: <nom_fic>_derive.<ext> créé dans le sous-répertoire plot (Annexe XVI).

Le fait de créer un polynôme ne change rien dans le fichier campagne. Cette étape est juste graphique. Elle permet simplement de visualiser la dérive horloge de l'ADCP au cours de la campagne.

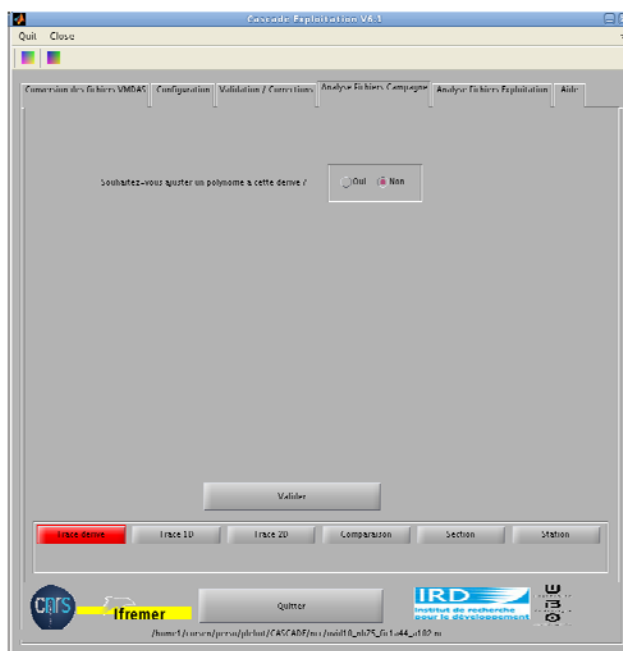


Figure 17 – Fenêtre tracé de la dérive

3.4.2 - Tracé 1D

Ce choix permet de tracer les variables 1D du fichier. Ces variables sont tracées en fonction du temps ou en fonction des ensembles. Selon le choix de l'utilisateur.

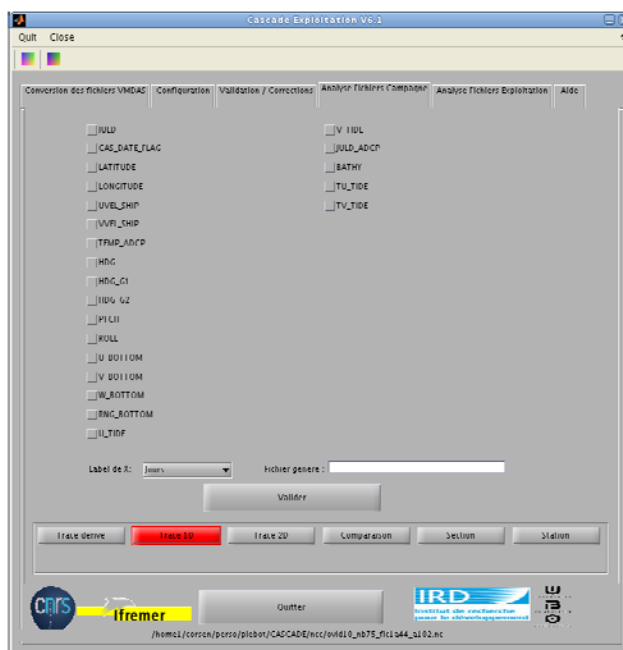


Figure 18 – Ecran de Tracé 1D

Les tracés sont générés dans un fichier dont le nom se compose ainsi : *<nom_fic>_trace1D.<ext>* (Annexe XVII)

Le nom n'est pas fixe, il doit être saisi par l'utilisateur.

3.4.3 - Tracé 2D

Ce choix permet de tracer les variables 2D du fichier, en fonction des ensembles et de la profondeur ou des cellules.

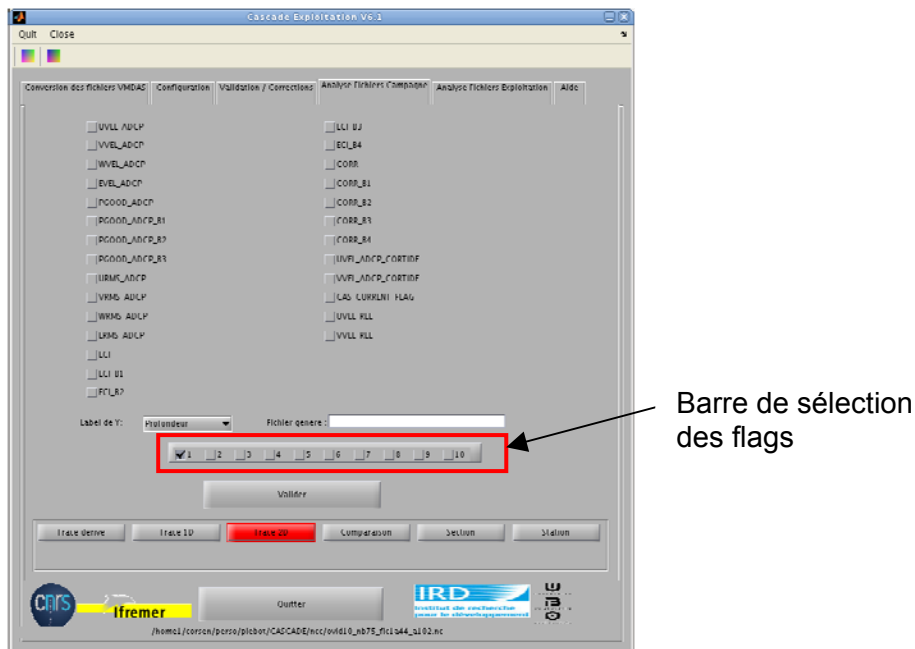


Figure 19 – Ecran de Tracé 2D

Attention: la variable CAS_CURRENT_FLAG (la valeur des flags) ne peut pas être cochée seule, mais elle fera toujours l'objet d'une figure à part si elle est sélectionnée.

On peut zoomer sur ces graphes (Zoom In et Zoom out dans le menu Tools, ou directement dans la barre d'outils du graphe) et changer les couleurs ou les limites de l'échelle de couleur directement dans la barre d'outil du graphe (menu Edit/colormap) mais il faut alors réenregistrer la figure soi-même.

C'est également sur les graphes des composantes du courant que l'on peut repérer des vitesses incorrectes (traits verticaux d'une couleur tranchant sur le fond) ou incorrectement invalidées. En visualisant les flags en parallèle, l'utilisateur peut juger s'il doit être plus ou moins sévère pour les divers critères de nettoyage. Si aucun des critères du nettoyage ne peut invalider des données visiblement incorrectes, il doit alors noter les numéros d'ensembles associés afin de les invalider par dates ou ensembles.

L'utilisateur sélectionnera les flags qu'il souhaite tracer en cochant les numéros correspondants. Pour sélectionner l'ensemble des flags, il suffit de cliquer sur la barre de flags.

Les tracés (Annexe XVIII) sont sauves dans un fichier dont le nom se compose ainsi :

<nom_fic>_trace2D_1.<ext>

<nom_fic>_trace2D_2.<ext> si CAS_CURRENT_FLAG est cochée.

Le nom <nom_fic> n'est pas fixe, il doit être saisi par l'utilisateur

3.4.4 - Comparaison

- **Uadcq / Unavire**

Le but de cette étape est de vérifier que la vitesse du courant et celle du navire ne sont pas corrélées. Si une corrélation nette existe, c'est qu'il y a un problème sur les mesures du courant.

L'écran correspondant est le suivant :

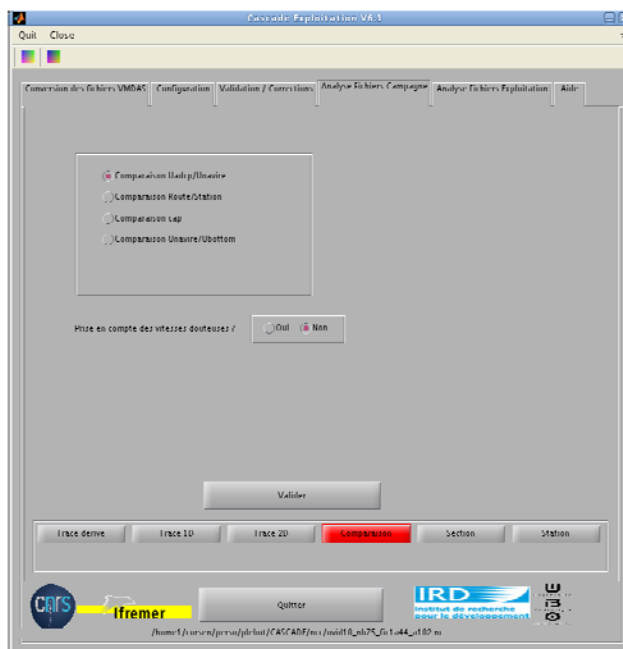


Figure 20 – Ecran Comparaison Uadcq/Unavire

Lors de la comparaison, il est possible de prendre en compte les vitesses douteuses.

« Valider » lance la comparaison, et trois graphes s'affichent :

- Un graphe sur les composantes de la vitesse du navire
- Un graphe sur les composantes de la vitesse du courant moyennée sur la couche de référence
- Un graphe superposant les composantes de la vitesse du navire et celles de la vitesse du courant (avec un facteur d'échelle pour la vitesse du navire).

L'étude de ces graphes permet de mettre en évidence une éventuelle corrélation entre les 2 vitesses que l'utilisateur peut alors corriger via l'application d'un désalignement de l'ADCP (cf. 3.3.6).

Les graphes sont sauvegardés automatiquement dans le sous-répertoire plot, <nom_fic>_cmp_Uadcq_Vnav.<.ext> (Annexe XIX).

- **Route/Station**

On compare les données de vitesse de courant en route avec les données de vitesse de courant en station. Les vitesses sont calculées sur la couche de référence.

Les vitesses absolues moyennées sur la couche de référence en station doivent s'approcher des vitesses absolues en route encadrant la station. Si cela n'est pas le cas, cela induit un éventuel problème de désalignement ou d'amplitude.

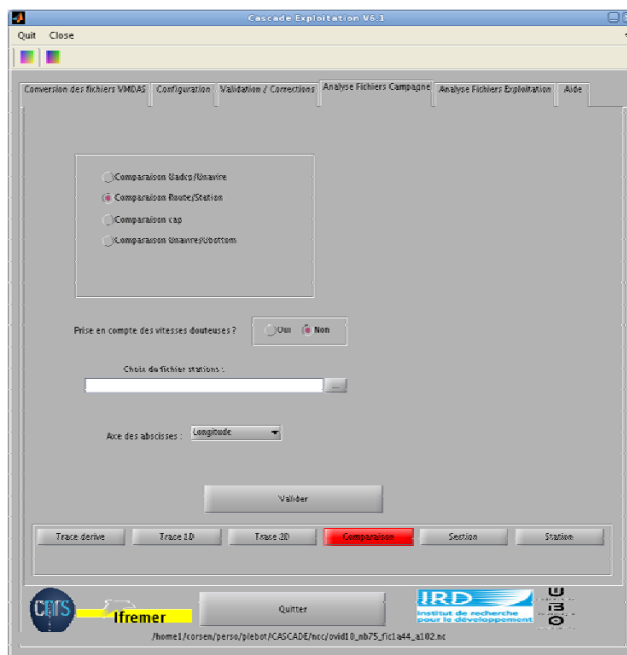


Figure 21 – Ecran Comparaison Route/Station

L'utilisateur entre le nom du fichier ASCII station à considérer (*Annexe XIX-1*). Il indique s'il désire considérer les données douteuses ou non ainsi que l'axe des abscisses (longitude ou latitude selon la route du navire) (*Annexe XIX-2*).

Les graphes générés sont sauvegardés dans le sous-répertoire plot : `<nom_fic>_cmp_sta_route*.<ext>`

On peut les modifier si besoin avant de cliquer sur OK.

- **Cap**

On compare deux caps parmi une liste proposée. Cette liste est générée à partir du fichier campagne. On recherche dans celui-ci toutes les variables liées au cap et on propose de pouvoir les comparer. (*Annexe XIX-3*)

Les graphes générés sont sauvegardés dans le sous-répertoire plot : `<nom_fic>_comparaison_cap.<ext>`.

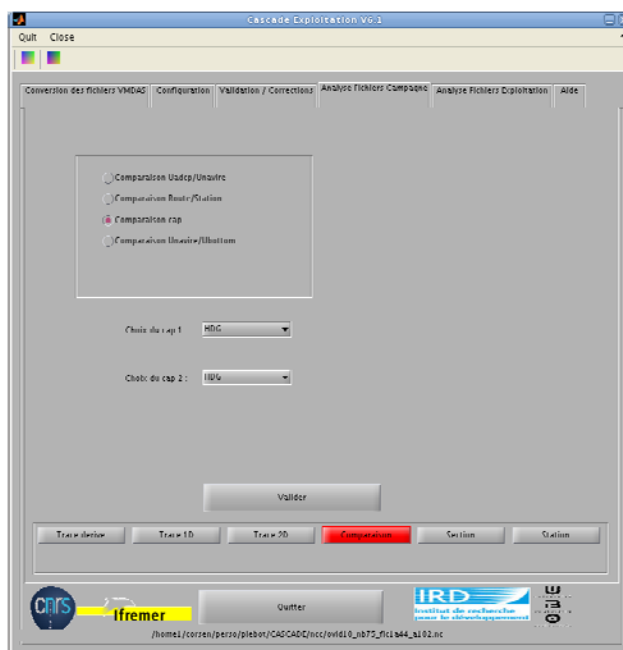


Figure 22 – Ecran Comparaison du cap Unavire/Ubottom

On compare le module et angle des vitesse navire et fond. 2 graphes sont générés :

Le module de la vitesse fond en fonction du module de la vitesse navire. Si aucun problème d'amplitude n'existe, alors les données doivent se regrouper autour de la droite $x=y$. Les vitesses fond et navire doivent être comparables. Si tel n'est pas le cas, l'utilisateur peut y remédier via l'application d'une amplitude (cf. 3.3.6), >1 si les vitesses navire sont plus grandes que les vitesses fond, et inversement. Si une telle différence existe, elle est généralement confirmée par une corrélation entre vitesse courant parallèle à la route et vitesse navire dans les informations sur le fichier campagne (voir 3.6.2).

La différence de direction entre la vitesse fond et la vitesse navire. L'éventuel angle de désalignement à appliquer est estimé pour les moments où le navire ne change pas de direction. Cet angle est représenté par un intervalle de confiance sur le graphe. Si l'angle moyen est significatif, on doit retrouver une corrélation entre vitesse courant perpendiculaire à la route et vitesse navire dans les informations sur le fichier campagne (voir 3.6.2).

En fonction de ces graphes, l'utilisateur peut corriger les vitesses absolues du courant via l'application d'une amplitude et/ou d'un désalignement (cf.3.3.6)

3.4.5 - Créer un fichier section

Cette étape permet de créer un fichier NetCDF section (*Annexe VI*). Une section correspond à une route cohérente du navire. Lors d'une campagne, s'il y a des changements de directions importants du navire, ou des changements d'environnement, on peut créer plusieurs sections pour représenter les données.

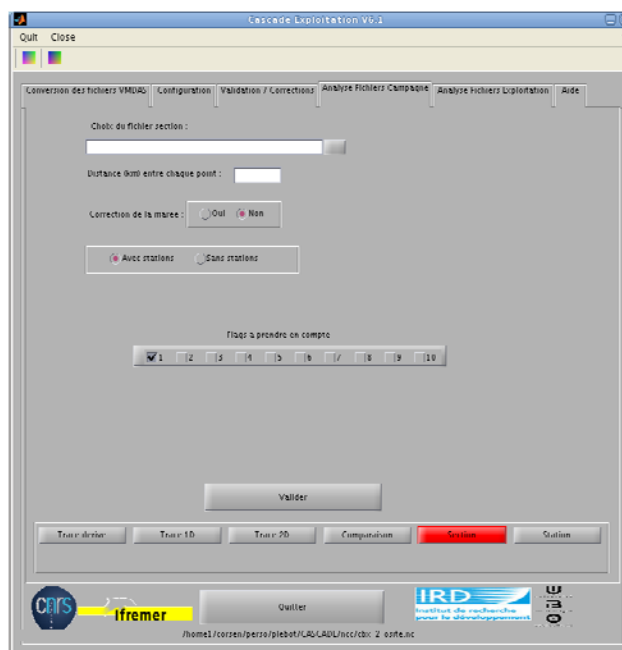


Figure 23 – Ecran Créer un fichier section

Dans le champ « Fichier Section » il faut mettre le nom du fichier ASCII $\langle nom_fic \rangle_sec.list$ que l'utilisateur doit préalablement créé manuellement (*Annexe I*). On moyenne les données sur X kilomètres, X étant fixé par l'utilisateur. On décide de prendre les données corrigés ou non de la marée. On peut garder ou pas les données associées aux stations.

La nomenclature du fichier créé est la suivante :

$$\langle nom_fic \rangle_sec_X \begin{matrix} /m & s \\ /x & x \end{matrix} _ \langle flag \rangle .nc$$

X représente le nombre de kilomètres entre chaque point.

/m prise en compte de la marée : elle est retirée du signal

/x non prise en compte de la marée

/s stations incluses

/x stations enlevées (non prise en compte dans la moyenne sur X kilomètres)

<flag> indique les flags pris en compte lors de l'étape

Le fichier est créé dans le sous-répertoire nce.

Exemple : *ovide_osite_sec_02ms1.nc* pour un fichier moyenné sur 2 km incluant les stations et pour lequel la marée a été retirée. Ce fichier a été généré à partir de données correctes (flaguées à 1).

3.4.5 - Créer un fichier station

Cette étape permet de créer le fichier NetCDF station pour la campagne (*Annexe VII*). Une station correspond à un lieu et une durée où le navire reste sur place (par rapport au fond) pour une opération particulière (mise à l'eau d'instruments généralement).

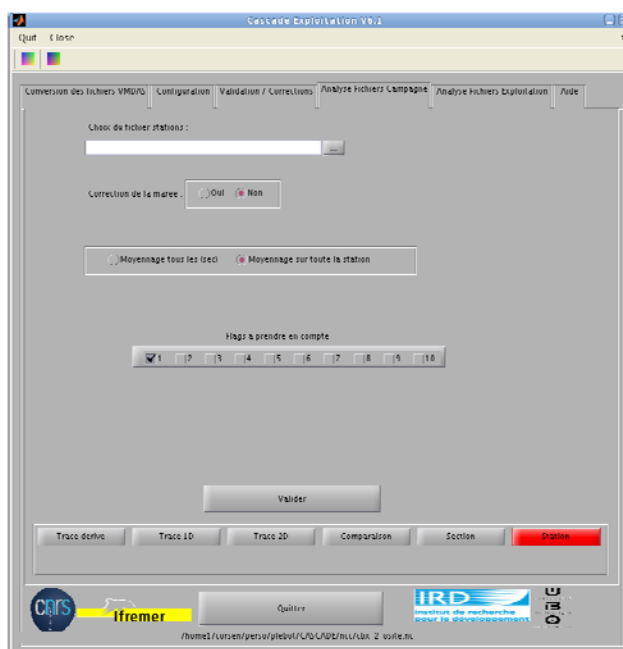


Figure 24 – Ecran Créer un fichier station

Dans le champ « Fichier Station » il faut mettre le nom du fichier ASCII *<nom_fic>_sta.list* que l'utilisateur doit préalablement créé manuellement (*Annexe I*). L'utilisateur a le choix de prendre les données corrigées ou non de la marée (si la marée a été ajoutée au fichier campagne). On peut moyennner sur un temps défini ou bien sur toute la station. On détermine les flags à prendre en compte (1-2 en général) puis on clique sur « Valider » pour commencer la création.

La nomenclature du fichier créé est la suivante :

$$\langle \text{nom_fic} \rangle_sta_X \begin{array}{l} /m \\ /x \end{array} _ \langle \text{flag} \rangle .nc$$

X représente le nombre de secondes qui a servi pour moyennner, si on a moyennné sur toute la station alors X = 99999.

/m prise en compte de la marée : elle est retirée du signal

/x non prise en compte de la marée

<flag> indique les flags prise en compte lors de l'étape

Le fichier est créé dans le sous-répertoire nce.

Exemple : *ovide_osite_sta_00120x1.nc* pour un fichier dans lequel les stations sont moyennnées sur 2 minutes et la marée non retirée du signal. Ce fichier a été généré à partir de données correctes (flaguées à 1) ;

3.5 Analyse Fichiers Exploitation

3.5.1 - Section

Cette étape permet de visualiser les variables du fichier NetCDF section (*Annexe VI*) et de tracer les vitesses du courant associées sous diverses formes.

4 choix sont possibles : le contournage, les tracés de vecteurs, les tracés 1D et 2D.

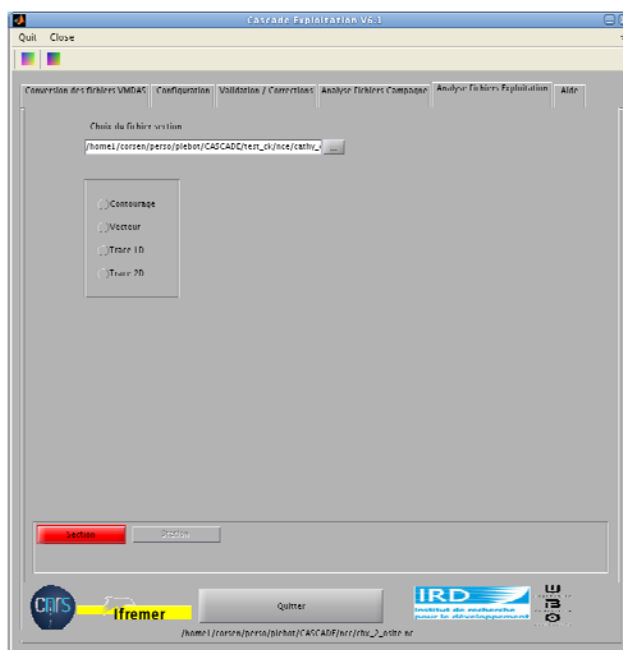


Figure 25 – Ecran d'exploitation d'un fichier section

3.5.1.1 - Contourage

L'option contourage permet de contourer les vitesses absolues du courant horizontales et, optionnellement, la composante verticale.

Lorsque l'option contourage est choisie, l'écran suivant s'affiche :

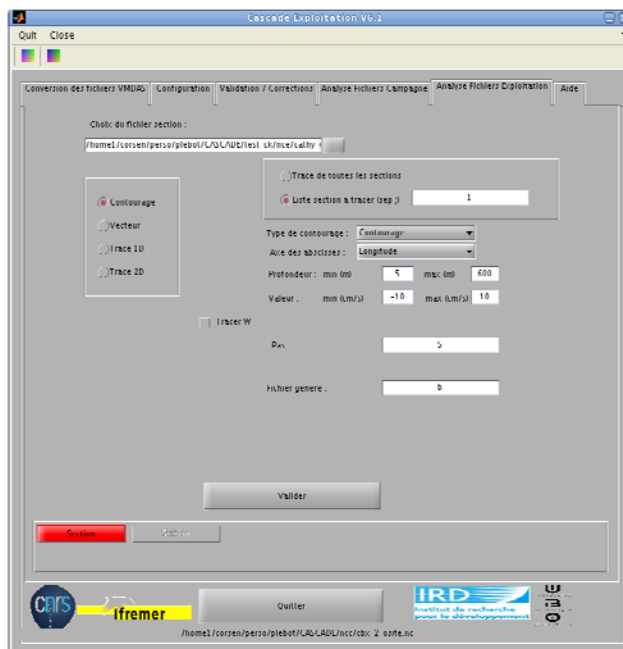


Figure 26 – Ecran Contourage pour les sections

Il est possible de tracer toutes les sections ou bien de choisir une ou plusieurs section(s) en particulier. Dans le cas où l'utilisateur désire ne tracer que des sections spécifiques, il suffit de rentrer la liste des sections en les séparant par des points virgule.

Ensuite, il faut choisir :

- le type de contourage : image ou contourage
- l'axe des abscisses : latitude ou longitude.
- Les profondeurs, en m, minimales et maximales du tracé (positives)
- Les valeurs minimales et maximales du contourage des vitesses
- Le tracé de la vitesse verticale ou non
- Le pas utilisé par le contourage
- Le nom du fichier tracé qui sera sauvegardé dans le sous-répertoire plot.

Une fois tous les champs renseignés, il faut cliquer sur « Valider » pour lancer le tracé de contourage. Le tracé de W est en option, il suffit de sélectionner tracer W pour que celui-ci s'effectue. Un exemple est fourni en *Annexe XX*.

3.5.1.2 - Vecteur

Lorsque l'option vecteur est choisie, l'écran suivant s'affiche :

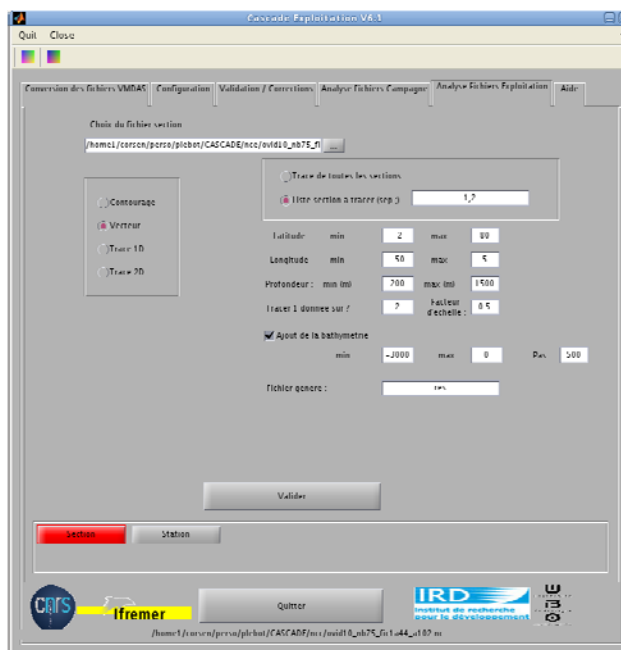


Figure 27 – Tracé de vecteurs pour les sections

Il est possible de tracer toutes les sections ou bien de choisir une ou plusieurs section(s) en particulier. Dans le cas où l'utilisateur désire des sections spécifiques, il suffit de rentrer la liste des sections en les séparant par des points virgule.

L'utilisateur doit également préciser :

- Les bornes géographiques du tracé
- Les profondeurs (positives) à prendre en compte pour calculer la moyenne des vitesses absolues du courant
- « Tracer 1 donnée sur X » permet de ne tracer qu'un vecteur sur X. Cela rend le tracé plus clair.
- Le facteur d'échelle permet de jouer sur l'échelle des vecteurs. On joue sur ce paramètre afin de régler la longueur des flèches associées au vecteur vitesse.
- « Ajout bathymétrie » : cette option permet de réaliser la figure avec ou sans bathymétrie. Si le bouton est coché, l'utilisateur saisit les valeurs de contourage de la bathymétrie (valeurs minimales et maximales et le pas). Attention : les profondeurs sont à saisir en négatif.
- Le nom du fichier, sauvegarde du tracé, qui sera généré dans le sous-répertoire plot.

Une fois tous les champs renseignés, il suffit de cliquer sur « Valider » pour lancer le tracé de vecteurs..

Plusieurs figures s'affichent (*Annexe XXI*) :

- Le tracé des sections
- Le nombre de cellules en fonction de la profondeur, ce graphe peut être utile au moment de la définition des couches à tracer, si le nombre de cellules n'est pas significatif sur une tranche d'eau, il faut éventuellement revoir les profondeurs minimale et maximale.
- Les tracés de vecteurs pour chaque section.

3.5.1.3 - Tracé 1D :

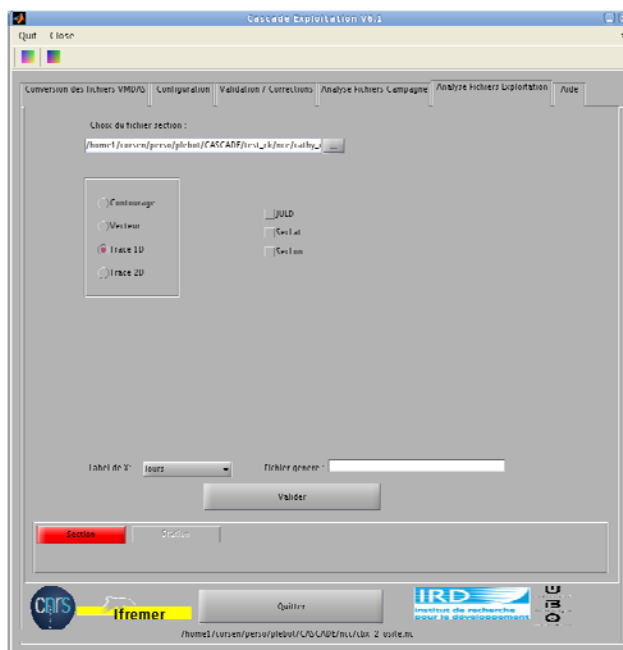


Figure 28 – Tracé 1D pour les sections

Cette étape réalise les mêmes tracés que le tracé 1D dans l'analyse du fichier campagne. Elle permet donc de visualiser les variables 1D (fonction du temps ou des ensembles) du fichier NetCDF section.

3.5.1.4 - Tracé 2D

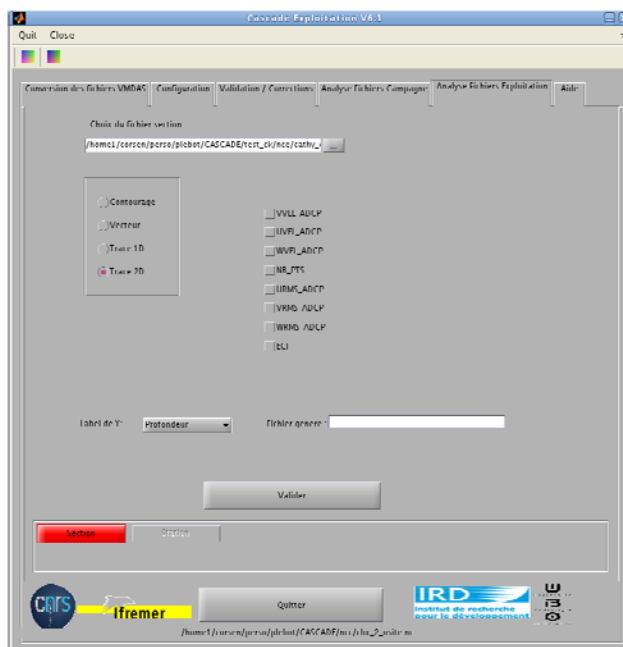


Figure 29 – Tracé 2D pour les sections

Cette étape réalise les mêmes tracés que le tracé 2D dans l'analyse du fichier campagne. Elle permet donc de visualiser sous forme de contourage les variables 2D (fonction des ensembles et de la profondeur ou des cellules) du fichier NetCDF section.

3.5.2 - Station

Cette étape permet de visualiser les différentes variables du fichier NetCDF station (*format en Annexe VII*) sous diverses formes.

6 choix sont possibles : le contourage, les tracés de vecteurs, les profils (moyens et débuts-fin) et les tracé 1D et 2D.

3.5.2.1 - Vecteur

Lorsque l'option vecteur est choisie, l'écran suivant s'affiche :

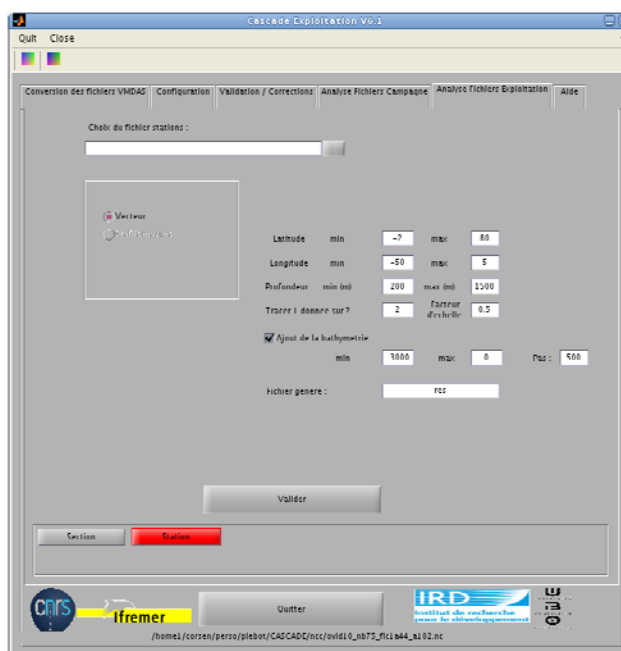


Figure 30 – Ecran d'exploitation d'un fichier station

L'utilisateur renseigne :

- Les bornes géographiques du tracé
- Les profondeurs (positives) à prendre en compte pour calculer la moyenne des vitesses absolues du courant
- « Tracer 1 donnée sur X » permet de ne tracer qu'un vecteur sur X. Cela rend le tracé plus clair.
- Le facteur d'échelle permet de jouer sur l'échelle des vecteurs. On joue sur ce paramètre afin de régler la longueur des flèches associées au vecteur vitesse.
- « Ajout bathymétrie » : cette option permet de réaliser la figure avec ou sans bathymétrie. Si le bouton est coché, l'utilisateur saisit les valeurs de contourage de la bathymétrie (valeurs minimales et maximales et le pas). Attention : les profondeurs sont à saisir en négatif).
- Le nom du fichier, sauvegarde du tracé, qui sera généré dans le sous-répertoire plot.

Une fois tous les champs renseignés, l'utilisateur clique sur 'Valider' pour lancer le tracé. Plusieurs figures s'affichent (*Annexe XXII*) :

- La carte de localisation de la station

- Le nombre de cellules en fonction de la profondeur, ce graphe peut être utile au moment de la définition des couches à tracer, si le nombre de cellule n'est pas significatif sur une tranche d'eau, il faut éventuellement revoir les profondeurs minimale et maximale.
- Les tracés de vecteurs pour chaque station.

3.5.2.2 - Profils moyens

Cette étape permet de tracer le(s) profil(s) moyen(s) des vitesses pour toutes les stations.

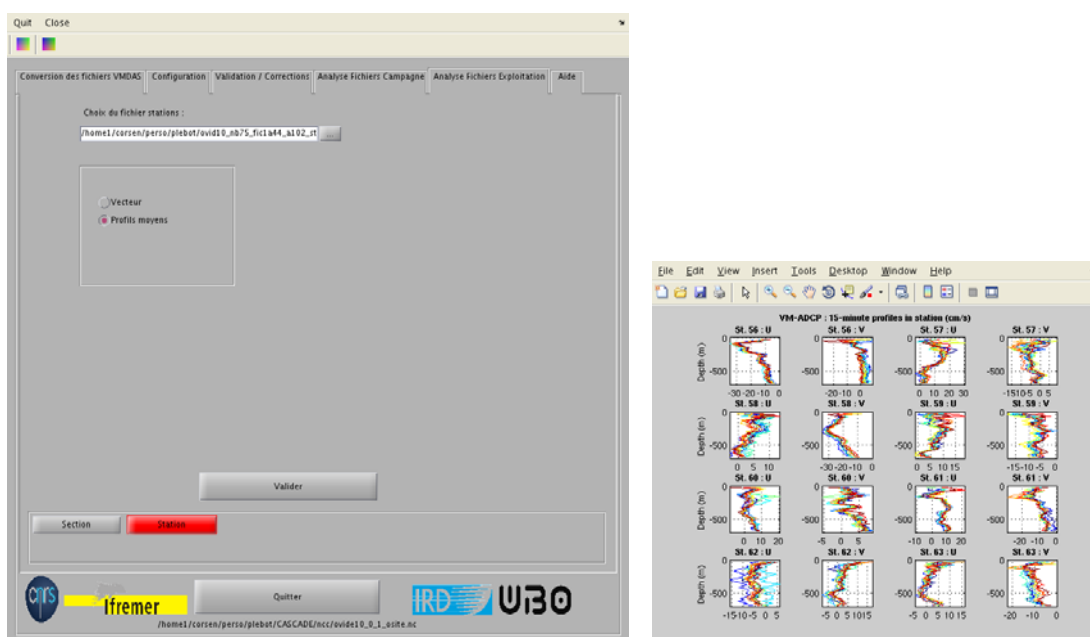


Figure 31 – Ecran des profils moyens pour une station

3.6 Aide

3.6.1 - Information flags

Cette fonction renseigne, à tout moment l'utilisateur, sur la définition des flags du fichier de données en cours.

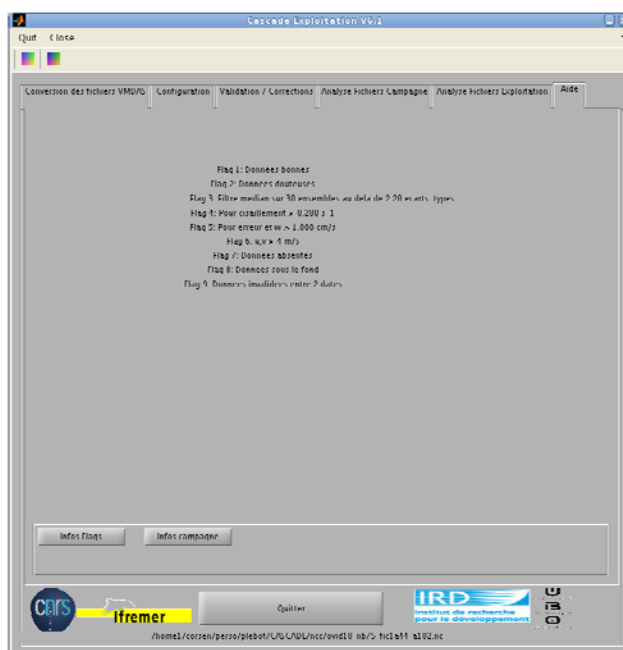


Figure 32 – Informations flags

3.6.2 - Informations campagne

Le lancement de cette commande entraîne la création d'un fichier *ncc/<nom_fic>.txt* récapitulant toutes les informations liées aux flags, au nombre de données flaguées, à la vitesse verticale moyenne, à la corrélation entre vitesse navire et vitesse du courant. On y trouve aussi l'amplitude et le désalignement appliqués ainsi que l'état du fichier (marée prise en compte ou non, bathymétrie ajoutée ou non, ...etc) .

Ces informations sont également affichées à l'écran :

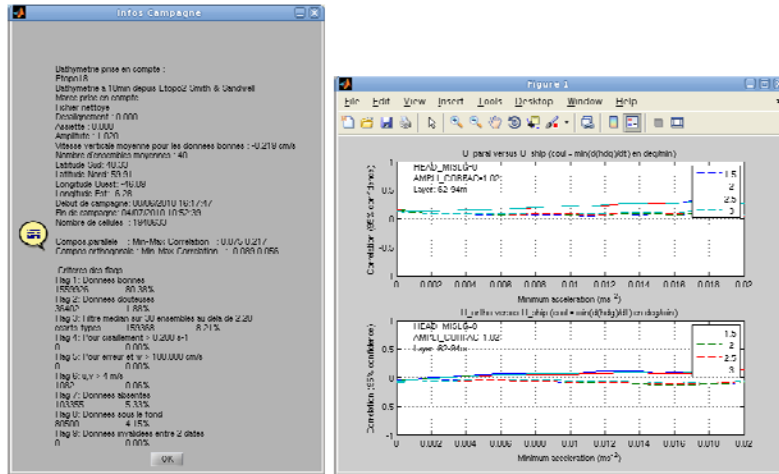


Figure 33 – Ecran informations générales.

La vitesse verticale moyenne, pour des ADCPs NarrowBand ou Broadband, est importante : sa valeur doit être proche de zéro (inférieure à 1 cm/s). Sinon, l'utilisateur doit y remédier via l'application d'une éventuelle correction d'assiette. A noter que pour les ADCPs OceanSurveyor, la vitesse verticale n'est pas représentative et sa moyenne ne donne aucune information quant à la validité des données.

Des informations concernant les corrélations entre la vitesse du navire et les composantes des vitesses absolues du courant parallèle et orthogonale à la vitesse du navire sont données en bas de l'écran : l'idéal est que les deux valeurs de chacune des corrélations soient petites et centrées (autour de zéro) pour des accélérations de plus de 0.01 ms⁻². La corrélation des composantes avec la vitesse du navire traduit en effet un problème de désalignement ou d'amplitude.

- Si la corrélation avec la composante parallèle est significativement différente de 0 => Problème d'amplitude. Il faut généralement diminuer l'amplitude si la corrélation est positive.
- Si la corrélation avec la composante orthogonale est significativement différente de 0 => Problème de désalignement. Il faut généralement diminuer le désalignement si la corrélation est positive.

Un graphe associé aux corrélations est tracé. Il est sauvegardé dans le sous-répertoire plot avec le nom `corrcoef_*.<ext>`.

En conséquence, l'utilisateur décide ou non de corriger les données via l'application d'un facteur d'amplitude et d'un désalignement différents de 1 et 0 respectivement.

4 Références

Marchalot, C., J. P. Berthomé, J. Bertrand, A. Cressard, C. Edy, F. Gaillard, R. Le Suavé, P. Viollette : Groupe de Travail Données Navires. Rapport final de la phase 0. R.Int.TMSI/IDM/02-017

Kermabon, C. et F. Gaillard, Janvier 2001 : CASCADE : logiciel de traitement des données ADCP de coque. Documentation maintenance - utilisateur (LPO-IFREMER).

Izenic Y, C. Kermabon, F. Gaillard, P, Lherminier : CASCADE 4.4 : logiciel de traitement et d'analyse des mesures ADCP de coque

Format OceanSite, oceansites-user-manual-v0.6.doc, T.Carval

5 Annexes

Annexe I. Exemple de fichiers section et station ASCII

Il doit avoir l'extension *sec.list* (exemple: *ovidesecc.list*) pour les sections.

Pour chaque section, il doit contenir la date et l'heure de début et la date et l'heure de fin.

Exemple de fichier section ASCII

```
1 03/01/2004 16:30:00 06/01/2004 08:10:00
2 06/01/2004 08:20:00 08/01/2004 14:38:00
3 08/01/2004 14:38:00 13/01/2004 06:40:00
```

Date début section

Date fin section

Et l'extension *_sta.list* (exemple: *ovide_sta.list*) pour les stations.

Pour chaque station, il doit contenir la date et l'heure de début et la date et l'heure de fin.

Exemple de fichier station ASCII

```
1 03/01/2004 16:30:00 00/01/2004 18:10:00
2 13/01/2004 00:38:00 13/01/2004 06:40:00
```

Date début station

Date fin station

Comment créer ces fichiers ?

Pour créer ces fichiers, l'utilisateur peut utiliser les plots latitude-longitude (tracé 1D) avec l'outil de datation puis entrer à la main les valeurs voulues dans un fichier selon les formats ci-dessus.

Annexe II. Tableau descriptif du format OceanSite

DIMENSION
Code
N_DATE_TIME
N_LEVEL
DATE_TIME
CONST1
CONST2
STRING32
STRINGFILT
STRINGFILTUSES

VARIABLES	
Code	Long name
REFERENCE_DATE_TIME	Date of reference for Julian days
JULD	Julian date relative to REFERENCE DATE TIME
DATE_TIME_UTC	ASCII gregorian date and time
LONGITUDE	Longitude of each location
LATITUDE	Latitude of each location
UVEL_SHIP	Eastward ship velocity
VVEL_SHIP	Northward ship velocity
DEPH	Depth of bin center
TEMP_ADCP	ADCP Transducer Temperature
HDG	Ship heading
HDG_G1	Ship heading G1
HDG_G2	Ship heading G2
PTCH	Ship pitch
ROLL	Ship roll
UVEL_ADCP	Eastward absolute ADCP current velocity
VVEL_ADCP	Northward absolute ADCP current velocity
WVEL_ADCP	Upward absolute ADCP current velocity
EVEL_ADCP	Absolute ADCP current velocity error
PGOOD_ADCP	Percent of good data before ensemble averaging
URMS_ADCP	Root mean square Eastward velocity
VRMS_ADCP	Root mean square Northward velocity
WRMS_ADCP	Root mean square Upward velocity
ERMS_ADCP	Root mean square velocity error
ECI	Mean echo intensity
ECI_B1	Echo intensity - Beam 1
ECI_B2	Echo intensity - Beam 2
ECI_B3	Echo intensity - Beam 3
ECI_B4	Echo intensity - Beam 4
CORR	Mean correlation
CORR_B1	Correlation - Beam 1
CORR_B2	Correlation - Beam 2

CORR_B3	Correlation - Beam 3
CORR_B4	Correlation - Beam 4
U_BOTTOM	Bottom track Eastward velocity
V_BOTTOM	Bottom track Northward velocity
W_BOTTOM	Bottom track Upward velocity
RNG_BOTTOM	Bottom range
U_TIDE	Eastward tide velocity
V_TIDE	Northward tide velocity
UVEL_ADCP_CORTIDE	Eastward absolute velocity corrected for tide
VVEL_ADCP_CORTIDE	Northward absolute velocity corrected for tide
WMEAN_DIAG	Mean vertical velocity (flag 1)
MINCORR_PARA_DIAG	Min correlation with ship velocity - Parallel
MAXCORR_PARA_DIAG	Max correlation with ship velocity - Parallel
MINCORR_ORTHO_DIAG	Min correlation with ship velocity - Transverse
MAXCORR_ORTHO_DIAG	Max correlation with ship velocity - Transverse
BATHY	Bathymetry
TX_FREQUENCY	ADCP transmitter frequency
SCALE_FACTOR	Scale factor
BEAM_ANGLE	Beam angle/vertical
ADCP_ANGLE	ADCP angle/ship axis
BIN_LENGTH	Bin length
MIDDLE_BIN1_DEPTH	Depth of first bin center
XOFF	
CORR_PR	
NB_ENS_AVE	Number of averaged ping per ensemble
HEAD_MISLG	Heading misalignment
PITCH_MISLG	Pitch misalignment
AMPLI_CORFAC	Correction factor on velocity amplitude
REF_LAYER_ILIM	Reference layer limit index
FLAG2_HALF_WINDOW	Flag 2 half width of window
FLAG2_SCF_MED_DEV	Flag 2 scaling factor median deviation
FLAG3_MAX_DEV	Flag 3 maximum deviation relative to mean profile
FLAG3_SCF_VSHEAR	Flag 3 scaling factor on vertical shear
FLAG4_MAX_VSHEAR	Flag 4 maximum vertical shear
FLAG5_MAX_WVEL	Flag 5 maximum horizontal velocity
FLAG6_INTERF	Interference threshold
FLAG6_MAX_VVEL	Flag 6 maximum vertical velocity
FLAG8_BOTTOM	Flag 8 bottom detection
FILT_TYPE	Type of filtering
FILT_FLAGS	List of flags used before filtering
JULD_ADCP	Julian date ADCP relative to REFERENCE_DATE_TIME
CAS_CURRENT_FLAG	Flag on ADCP current (U,V,W)

ATTRIBUTS

Code
DATE_CREATION
SOFTWARE
ADCP_CONSTRUCTOR
ADCP_TYPE
CONVENTIONS
CRUISE_NAME
PLATFORM_NUMBER (WMO)
PLATFORM_NAME
DATE_UPDATE
DATA_TYPE
FORMAT_VERSION
NAVIGATION_REFERENCE

Annexe III. Format de fichier OceanSite

```

netcdf ovide2010_osite_fhv {
dimensions:
    CONST1 = 1 ;
    N_DATE_TIME = 1724 ;
    N_LEVEL = 45 ;
    DATE_TIME = 14 ;
    CONST2 = 2 ;
    STRING32 = 32 ;
    STRINGFILT = 30 ;
    STRINGFILTUSES = 1 ;
variables:
    float TX_FREQUENCY(CONST1) ;
        TX_FREQUENCY:long_name = "ADCP transmitter frequency" ;
        TX_FREQUENCY:units = "KiloHz" ;
        TX_FREQUENCY:_FillValue = -999999.f ;
    float SCALE_FACTOR(CONST1) ;
        SCALE_FACTOR:long_name = "Scale factor" ;
        SCALE_FACTOR:_FillValue = -999999.f ;
    float BEAM_ANGLE(CONST1) ;
        BEAM_ANGLE:long_name = "Beam Angle/vertical" ;
        BEAM_ANGLE:units = "degrees" ;
        BEAM_ANGLE:_FillValue = -999999.f ;
    float ADCP_ANGLE(CONST1) ;
        ADCP_ANGLE:long_name = "ADCP Angle/ship axis" ;
        ADCP_ANGLE:units = "degrees" ;
        ADCP_ANGLE:_FillValue = -999999.f ;
    float BIN_LENGTH(CONST1) ;
        BIN_LENGTH:long_name = "Bin Length" ;
        BIN_LENGTH:units = "meter" ;
        BIN_LENGTH:_FillValue = -999999.f ;
    float MIDDLE_BIN1_DEPTH(CONST1) ;
        MIDDLE_BIN1_DEPTH:long_name = "Depth of first bin center" ;
        MIDDLE_BIN1_DEPTH:units = "meters" ;
        MIDDLE_BIN1_DEPTH:_FillValue = -999999.f ;
    float NB_ENS_AVE(CONST1) ;
        NB_ENS_AVE:long_name = "Number of averaged pings per ensemble"
;
        NB_ENS_AVE:_FillValue = -999999.f ;
    float HEAD_MISLG(CONST1) ;
        HEAD_MISLG:long_name = "Heading Misalignment" ;
        HEAD_MISLG:units = "degrees" ;
        HEAD_MISLG:_FillValue = -999999.f ;
    float PITCH_MISLG(CONST1) ;
        PITCH_MISLG:long_name = "Pitch Misalignment" ;
        PITCH_MISLG:units = "degrees" ;
        PITCH_MISLG:_FillValue = -999999.f ;
    float AMPLI_CORFAC(CONST1) ;
        AMPLI_CORFAC:long_name = "Correction factor on velocity
amplitude" ;
        AMPLI_CORFAC:_FillValue = -999999.f ;
    float XOFF(CONST1) ;
        XOFF:long_name = "Transducer Depth" ;
    float CORR_PR(CONST1) ;
        CORR_PR:long_name = "=1 : pitch/roll used, 0 otherwise" ;
    char REFERENCE_DATE_TIME(DATE_TIME) ;
        REFERENCE_DATE_TIME:long_name = "Date of reference julian day"
;
        REFERENCE_DATE_TIME:conventions = "YYYYMMDDHHMMSS" ;

```

```

REFERENCE_DATE_TIME:_FillValue = " " ;
double JULD(N_DATE_TIME) ;
JULD:long_name = "Julian day relative to REFERENCE_DATE_TIME"
;
JULD:_FillValue = -999999. ;
double JULD_ADCP(N_DATE_TIME) ;
JULD_ADCP:long_name = "ADCP Julian day relative to
REFERENCE_DATE_TIME" ;
JULD_ADCP:_FillValue = -999999. ;
double JULD_j1(N_DATE_TIME) ;
JULD_j1:long_name = "Begin Ensemble Julian day relative to
REFERENCE_DATE_TIME" ;
JULD_j1:_FillValue = -999999. ;
double JULD_j2(N_DATE_TIME) ;
JULD_j2:long_name = "End Ensemble Julian day relative to
REFERENCE_DATE_TIME" ;
JULD_j2:_FillValue = -999999. ;
char DATE_TIME.UTC(N_DATE_TIME, DATE_TIME) ;
DATE_TIME.UTC:long_name = "ASCII gregorian date and time" ;
DATE_TIME.UTC:convention = "YYYYMMDDHHMMSS" ;
DATE_TIME.UTC:_FillValue = " " ;
float CAS_DATE_FLAG(N_DATE_TIME) ;
CAS_DATE_FLAG:long_name = "Flag on date" ;
CAS_DATE_FLAG:_FillValue = -999999.f ;
float LATITUDE(N_DATE_TIME) ;
LATITUDE:long_name = "Latitude of each location" ;
LATITUDE:units = "degree_north" ;
LATITUDE:valid_min = -90. ;
LATITUDE:valid_max = 90. ;
LATITUDE:_FillValue = -999999.f ;
float LONGITUDE(N_DATE_TIME) ;
LONGITUDE:long_name = "Longitude of each location" ;
LONGITUDE:units = "degree_north" ;
LONGITUDE:valid_min = -180. ;
LONGITUDE:valid_max = 180. ;
LONGITUDE:_FillValue = -999999.f ;
float UVEL_SHIP(N_DATE_TIME) ;
UVEL_SHIP:long_name = "Eastward ship velocity" ;
UVEL_SHIP:units = "meter per second" ;
UVEL_SHIP:valid_min = -20. ;
UVEL_SHIP:valid_max = 20. ;
UVEL_SHIP:_FillValue = -999999.f ;
float VVEL_SHIP(N_DATE_TIME) ;
VVEL_SHIP:long_name = "Northward ship velocity" ;
VVEL_SHIP:units = "meter per second" ;
VVEL_SHIP:valid_min = -20. ;
VVEL_SHIP:valid_max = 20. ;
VVEL_SHIP:_FillValue = -999999.f ;
float DEPH(N_LEVEL) ;
DEPH:long_name = "Depth of bin center" ;
DEPH:units = "meter" ;
DEPH:valid_min = -12000. ;
DEPH:valid_max = 0. ;
DEPH:_FillValue = -999999.f ;
float TEMP_ADCP(N_DATE_TIME) ;
TEMP_ADCP:long_name = "ADCP transducer temperature" ;
TEMP_ADCP:units = "Degree_Celsius" ;
TEMP_ADCP:valid_min = -5. ;
TEMP_ADCP:valid_max = 45. ;
TEMP_ADCP:_FillValue = -999999.f ;
float HDG(N_DATE_TIME) ;

```

```

    HDG:long_name = "Ship Heading" ;
    HDG:units = "Degree" ;
    HDG:valid_min = -360. ;
    HDG:valid_max = 360. ;
    HDG:_FillValue = -999999.f ;
float HDG_G1(N_DATE_TIME) ;
    HDG_G1:long_name = "Ship Heading G1" ;
    HDG_G1:units = "Degree" ;
    HDG_G1:valid_min = -360. ;
    HDG_G1:valid_max = 360. ;
    HDG_G1:_FillValue = -999999.f ;
float HDG_G2(N_DATE_TIME) ;
    HDG_G2:long_name = "Ship Heading G2" ;
    HDG_G2:units = "Degree" ;
    HDG_G2:valid_min = -360. ;
    HDG_G2:valid_max = 360. ;
    HDG_G2:_FillValue = -999999.f ;
float PTCH(N_DATE_TIME) ;
    PTCH:long_name = "Ship Pitch" ;
    PTCH:units = "Degree" ;
    PTCH:valid_min = -360. ;
    PTCH:valid_max = 360. ;
    PTCH:_FillValue = -999999.f ;
float ROLL(N_DATE_TIME) ;
    ROLL:long_name = "Ship Roll" ;
    ROLL:units = "Degree" ;
    ROLL:valid_min = -360. ;
    ROLL:valid_max = 360. ;
    ROLL:_FillValue = -999999.f ;
float U_BOTTOM(N_DATE_TIME) ;
    U_BOTTOM:long_name = "Bottom Track Eastward velocity" ;
    U_BOTTOM:units = "meter per second" ;
    U_BOTTOM:valid_min = -20. ;
    U_BOTTOM:valid_max = 20. ;
    U_BOTTOM:_FillValue = -999999.f ;
float V_BOTTOM(N_DATE_TIME) ;
    V_BOTTOM:long_name = "Bottom Track Northward velocity" ;
    V_BOTTOM:units = "meter per second" ;
    V_BOTTOM:valid_min = -20. ;
    V_BOTTOM:valid_max = 20. ;
    V_BOTTOM:_FillValue = -999999.f ;
float W_BOTTOM(N_DATE_TIME) ;
    W_BOTTOM:long_name = "Bottom Track Vertical velocity" ;
    W_BOTTOM:units = "meter per second" ;
    W_BOTTOM:valid_min = -20. ;
    W_BOTTOM:valid_max = 20. ;
    W_BOTTOM:_FillValue = -999999.f ;
float RNG_BOTTOM(N_DATE_TIME) ;
    RNG_BOTTOM:long_name = "Bottom Range" ;
    RNG_BOTTOM:units = "meter" ;
    RNG_BOTTOM:_FillValue = -999999.f ;
float UVEL_ADCP(N_DATE_TIME, N_LEVEL) ;
    UVEL_ADCP:long_name = "Eastward absolute ADCP current
velocity" ;
    UVEL_ADCP:units = "meter per second" ;
    UVEL_ADCP:valid_min = -20. ;
    UVEL_ADCP:valid_max = 20. ;
    UVEL_ADCP:_FillValue = -999999.f ;
float VVEL_ADCP(N_DATE_TIME, N_LEVEL) ;
    VVEL_ADCP:long_name = "Northward absolute ADCP current
velocity" ;

```

```

    VVEL_ADCP:units = "meter per second" ;
    VVEL_ADCP:valid_min = -20. ;
    VVEL_ADCP:valid_max = 20. ;
    VVEL_ADCP:_FillValue = -999999.f ;
float WVEL_ADCP(N_DATE_TIME, N_LEVEL) ;
    WVEL_ADCP:long_name = "Vertical absolute ADCP current
velocity" ;
    WVEL_ADCP:units = "meter per second" ;
    WVEL_ADCP:valid_min = -20. ;
    WVEL_ADCP:valid_max = 20. ;
    WVEL_ADCP:_FillValue = -999999.f ;
float EVEL_ADCP(N_DATE_TIME, N_LEVEL) ;
    EVEL_ADCP:long_name = "absolute ADCP current velocity error" ;
    EVEL_ADCP:units = "meter per second" ;
    EVEL_ADCP:valid_min = -20. ;
    EVEL_ADCP:valid_max = 20. ;
    EVEL_ADCP:_FillValue = -999999.f ;
float UVEL_REL(N_DATE_TIME, N_LEVEL) ;
    UVEL_REL:long_name = "Eastward relative ADCP current velocity"
;
    UVEL_REL:units = "meter per second" ;
    UVEL_REL:valid_min = -20. ;
    UVEL_REL:valid_max = 20. ;
    UVEL_REL:_FillValue = -999999.f ;
float VVEL_REL(N_DATE_TIME, N_LEVEL) ;
    VVEL_REL:long_name = "Northward relative ADCP current
velocity" ;
    VVEL_REL:units = "meter per second" ;
    VVEL_REL:valid_min = -20. ;
    VVEL_REL:valid_max = 20. ;
    VVEL_REL:_FillValue = -999999.f ;
float PGOOD_ADCP(N_DATE_TIME, N_LEVEL) ;
    PGOOD_ADCP:long_name = "% of good data with 4 beams" ;
    PGOOD_ADCP:units = "percent" ;
    PGOOD_ADCP:valid_min = 0. ;
    PGOOD_ADCP:valid_max = 100. ;
    PGOOD_ADCP:_FillValue = -999999.f ;
float PGOOD_ADCP_B2(N_DATE_TIME, N_LEVEL) ;
    PGOOD_ADCP_B2:long_name = "% of good data with 3 beams" ;
    PGOOD_ADCP_B2:units = "percent" ;
    PGOOD_ADCP_B2:valid_min = 0. ;
    PGOOD_ADCP_B2:valid_max = 100. ;
    PGOOD_ADCP_B2:_FillValue = -999999.f ;
float PGOOD_ADCP_B3(N_DATE_TIME, N_LEVEL) ;
    PGOOD_ADCP_B3:long_name = "% of good data rejected because of
EW" ;
    PGOOD_ADCP_B3:units = "percent" ;
    PGOOD_ADCP_B3:valid_min = 0. ;
    PGOOD_ADCP_B3:valid_max = 100. ;
    PGOOD_ADCP_B3:_FillValue = -999999.f ;
float PGOOD_ADCP_B4(N_DATE_TIME, N_LEVEL) ;
    PGOOD_ADCP_B4:long_name = "% of data with 2 bad beams" ;
    PGOOD_ADCP_B4:units = "percent" ;
    PGOOD_ADCP_B4:valid_min = 0. ;
    PGOOD_ADCP_B4:valid_max = 100. ;
    PGOOD_ADCP_B4:_FillValue = -999999.f ;
float ECI(N_DATE_TIME, N_LEVEL) ;
    ECI:long_name = "Mean echo Intensity" ;
    ECI:units = "count" ;
    ECI:_FillValue = -999999.f ;
float ECI_B1(N_DATE_TIME, N_LEVEL) ;

```

```

    ECI_B1:long_name = "Mean echo Intensity for beam 1" ;
    ECI_B1:units = "count" ;
    ECI_B1:_FillValue = -999999.f ;
float ECI_B2(N_DATE_TIME, N_LEVEL) ;
    ECI_B2:long_name = "Mean echo Intensity for beam 2" ;
    ECI_B2:units = "count" ;
    ECI_B2:_FillValue = -999999.f ;
float ECI_B3(N_DATE_TIME, N_LEVEL) ;
    ECI_B3:long_name = "Mean echo Intensity for beam 3" ;
    ECI_B3:units = "count" ;
    ECI_B3:_FillValue = -999999.f ;
float ECI_B4(N_DATE_TIME, N_LEVEL) ;
    ECI_B4:long_name = "Mean echo Intensity for beam 4" ;
    ECI_B4:units = "count" ;
    ECI_B4:_FillValue = -999999.f ;
float CORR(N_DATE_TIME, N_LEVEL) ;
    CORR:long_name = "Correlation" ;
    CORR:units = "count" ;
    CORR:_FillValue = -999999.f ;
float CORR_B1(N_DATE_TIME, N_LEVEL) ;
    CORR_B1:long_name = "Correlation for beam 1" ;
    CORR_B1:units = "count" ;
    CORR_B1:_FillValue = -999999.f ;
float CORR_B2(N_DATE_TIME, N_LEVEL) ;
    CORR_B2:long_name = "Correlation for beam 2" ;
    CORR_B2:units = "count" ;
    CORR_B2:_FillValue = -999999.f ;
float CORR_B3(N_DATE_TIME, N_LEVEL) ;
    CORR_B3:long_name = "Correlation for beam 3" ;
    CORR_B3:units = "count" ;
    CORR_B3:_FillValue = -999999.f ;
float CORR_B4(N_DATE_TIME, N_LEVEL) ;
    CORR_B4:long_name = "Correlation for beam 4" ;
    CORR_B4:units = "count" ;
    CORR_B4:_FillValue = -999999.f ;

```

***** Variables ajoutées ou modifiées lors de l'exploitation *****

```

int REF_LAYER_ILIM(CONST2) ;
    REF_LAYER_ILIM:long_name = "Reference Layer Limit Index" ;
    REF_LAYER_ILIM:_FillValue = -999999 ;

float BATHY(N_DATE_TIME) ;
    BATHY:units = "meter" ;
    BATHY:comment = "Bathymetrie a 6min depuis Etopo2 Smith &
Sandwell" ;
    BATHY:long_name = "Etopo6" ;
    BATHY:_FillValue = -999999.f ;
    BATHY:valid_min = -12000. ;
    BATHY:valid_max = 0. ;
    BATHY:file =
"/home1/corsen/perso/plebot/CASCADE/exploitation/bathymetrie/bathy6min.
nc" ;

```

Def. Couche de
référence

Ajout de la
Bathymétrie

----- NETTOYAGE -----

```

float CAS_CURRENT_FLAG(N_DATE_TIME, N_LEVEL) ;
    CAS_CURRENT_FLAG:long_name = "Flag on ADCP velocity" ;
    CAS_CURRENT_FLAG:valid_min = 0. ;

```

Nettoyage

```

CAS_CURRENT_FLAG:valid_max = 9. ;
CAS_CURRENT_FLAG:_FillValue = -999999.f ;

int FLAG3_HALF_WINDOW(CONST1) ;
  FLAG3_HALF_WINDOW:long_name = "Flag 3 half width of window" ;
  FLAG3_HALF_WINDOW:_FillValue = -999999 ;
float FLAG3_SCF_MED_DEV(CONST1) ;
  FLAG3_SCF_MED_DEV:long_name = "Flag 3 scaling factor median
deviation" ;
  FLAG3_SCF_MED_DEV:_FillValue = -999999.f ;
float FLAG5_MAX_WVEL(CONST1) ;
  FLAG5_MAX_WVEL:long_name = "Flag 5 maximum horizontal
velocity" ;
  FLAG5_MAX_WVEL:units = "meter per second" ;
  FLAG5_MAX_WVEL:_FillValue = -999999.f ;
float FLAG4_MAX_VSHEAR(CONST1) ;
  FLAG4_MAX_VSHEAR:long_name = "Flag 4 maximum vertical shear" ;
  FLAG4_MAX_VSHEAR:units = "meter per second" ;
  FLAG4_MAX_VSHEAR:_FillValue = -999999.f ;
int FLAG8_BOTTOM(CONST1) ;
  FLAG8_BOTTOM:long_name = "Flag 8 bottom detection" ;
  FLAG8_BOTTOM:comments = "0 = no detection; 1 = ADCP bottom
range; 2 = external bathymetry" ;
  FLAG8_BOTTOM:_FillValue = -999999 ;
float FLAG2_SCF_VSHEAR(CONST1) ;
  FLAG2_SCF_VSHEAR:long_name = "Flag 2 scaling factor on
vertical shear" ;
  FLAG2_SCF_VSHEAR:_FillValue = -999999.f ;
float FLAG2_MAX_DEV(CONST1) ;
  FLAG2_MAX_DEV:long_name = "Flag 2 maximum deviation relative
to mean profile" ;
  FLAG2_MAX_DEV:_FillValue = -999999.f ;
float FLAG6_MAX_VVEL(CONST1) ;
  FLAG6_MAX_VVEL:long_name = "flag - maximum vertical velocity"
;
  FLAG6_MAX_VVEL:units = "meter per second" ;
  FLAG6_MAX_VVEL:_FillValue = -999999.f ;
float FLAG6_INTERF(CONST1) ;
  FLAG6_INTERF:long_name = "Interference threshold" ;
  FLAG6_INTERF:_FillValue = -999999.f ;

```

----- Ajout de la marée -----

```

float U_TIDE(N_DATE_TIME) ;
  U_TIDE:units = "meter per second" ;
  U_TIDE:type_tide = "Model_tpxo6.2" ;
  U_TIDE:long_name = "Eastward tide Velocity" ;
  U_TIDE:_FillValue = -999999.f ;
  U_TIDE:valid_min = -20. ;
  U_TIDE:valid_max = 20. ;
float V_TIDE(N_DATE_TIME) ;
  V_TIDE:units = "meter per second" ;
  V_TIDE:type_tide = "Model_tpxo6.2" ;
  V_TIDE:long_name = "Northward tide Velocity" ;
  V_TIDE:_FillValue = -999999.f ;
  V_TIDE:valid_min = -20. ;
  V_TIDE:valid_max = 20. ;
float UVEL_ADCP_CORTIDE(N_DATE_TIME, N_LEVEL) ;
  UVEL_ADCP_CORTIDE:units = "meter per second" ;
  UVEL_ADCP_CORTIDE:long_name = "Eastward absolute velocity
corrected for tide" ;

```

Ajout de la marée

Ajout de la marée

```

    UVEL_ADCP_CORTIDE:_FillValue = -999999.f ;
    UVEL_ADCP_CORTIDE:valid_min = -20. ;
    UVEL_ADCP_CORTIDE:valid_max = 20. ;
float VVEL_ADCP_CORTIDE(N_DATE_TIME, N_LEVEL) ;
    VVEL_ADCP_CORTIDE:units = "meter per second" ;
    VVEL_ADCP_CORTIDE:long_name = "Northward absolute velocity
corrected for tide" ;
    VVEL_ADCP_CORTIDE:_FillValue = -999999.f ;
    VVEL_ADCP_CORTIDE:valid_min = -20. ;
    VVEL_ADCP_CORTIDE:valid_max = 20. ;

```

```

float TU_TIDE(N_DATE_TIME) ;
    TU_TIDE:units = "meter^2 per second" ;
    TU_TIDE:long_name = "Eastward tide Transport" ;
    TU_TIDE:_FillValue = -999999.f ;
    TU_TIDE:valid_min = -20. ;
    TU_TIDE:valid_max = 20. ;
float TV_TIDE(N_DATE_TIME) ;
    TV_TIDE:units = "meter^2 per second" ;
    TV_TIDE:long_name = "Northward tide Transport" ;
    TV_TIDE:_FillValue = -999999.f ;
    TV_TIDE:valid_min = -20. ;
    TV_TIDE:valid_max = 20. ;

```

----- Fin ajout marrée -----

```

float HDG_EXTERNAL(N_DATE_TIME) ;
    HDG_EXTERNAL:long_name = "Ship external heading" ;
    HDG_EXTERNAL:units = "degree" ;
    HDG_EXTERNAL:valid_min = 0. ;
    HDG_EXTERNAL:valid_max = 360. ;
    HDG_EXTERNAL:_FillValue = -999999.f ;
char FILT_TYPE(STRINGFILT) ;
    FILT_TYPE:long_name = "Type of filtering" ;
double FILT_FLAGS(STRINGFILTUSES) ;
    FILT_FLAGS:long_name = "List of flags used before filtering" ;

// global attributes:
    :DATE_CREATION = "26-Nov-2010" ;
    :SOFTWARE = "CascadeVersion 6.1" ;
    :ADCP_CONSTRUCTOR = "RDI" ;
    :ADCP_TYPE = "Ocean Surveyor" ;
    :CONVENTIONS = "OceanSite dictionary" ;
    :CRUISE_NAME = "ovide_2010" ;
    :PLATFORM_NAME = "THALASSA" ;
    :DATE_UPDATE = "26-Nov-2010" ;
    :DATE_TYPE = "SADCP" ;
    :FORMAT_VERSION = "1.0" ;
    :NAVIGATION_REFERENCE = "GPS Navigation" ;
    :HEADING_REFERENCE = "EXTERNAL Heading" ;
    :PLATFORM_NUMBER = "FNFP" ;
}

```


Annexe IV. Format fichier TRINAV

```

netcdf ovidenavi {
dimensions:
    dim_trin = 230976 ;
    cte = 1 ;

// global attributes:
    :CREATION_DATE = "17-Mar-2005" ;
    :INST_TYPE = "TRINAV" ;
    :PROG_CMNT1 = "Converted to netCDF via MATLAB by Cdf_trin_c" ;

variables:
    float Lfilt_TRIN ;
        Lfilt_TRIN:units = "seconds" ;
        Lfilt_TRIN:long_name = "Largeur du filtre" ;
        Lfilt_TRIN:_FillValue = -999999.f ;
        Lfilt_TRIN:valid_range = 0.f, 3600.f ;
    double T_TRIN(dim_trin) ;
        T_TRIN:units = "decimal days" ;
        T_TRIN:long_name = "JULIAN DAYS TRINAV" ;
        T_TRIN:_FillValue = -999999. ;
        T_TRIN:valid_range = 2451545., 2461545. ;
    double Lat_TRIN(dim_trin) ;
        Lat_TRIN:units = "decimal Degrees" ;
        Lat_TRIN:long_name = "LATITUDE TRINAV" ;
        Lat_TRIN:_FillValue = -999999. ;
        Lat_TRIN:valid_range = -90., 90. ;
    double Lon_TRIN(dim_trin) ;
        Lon_TRIN:units = "decimal Degrees" ;
        Lon_TRIN:long_name = "LONGITUDE TRINAV" ;
        Lon_TRIN:_FillValue = -999999. ;
        Lon_TRIN:valid_range = -180., 180. ;
    float VitU_TRIN(dim_trin) ;
        VitU_TRIN:units = "m/s" ;
        VitU_TRIN:long_name = "Vitesse navire filtre (U)" ;
        VitU_TRIN:_FillValue = -999999.f ;
        VitU_TRIN:valid_range = -15.f, 15.f ;
    float VitV_TRIN(dim_trin) ;
        VitV_TRIN:units = "m/s" ;
        VitV_TRIN:long_name = "Vitesse navire filtre (V)" ;
        VitV_TRIN:_FillValue = -999999.f ;
        VitV_TRIN:valid_range = -15.f, 15.f ;
    float AccX_TRIN(dim_trin) ;
        AccX_TRIN:units = "m/s*2" ;
        AccX_TRIN:long_name = "Vitesse navire filtre (x)" ;
        AccX_TRIN:_FillValue = -999999.f ;
        AccX_TRIN:valid_range = -0.1f, 0.1f ;
    float AccY_TRIN(dim_trin) ;
        AccY_TRIN:units = "m/s*2" ;
        AccY_TRIN:long_name = "Vitesse navire filtre (y)" ;
        AccY_TRIN:_FillValue = -999999.f ;
        AccY_TRIN:valid_range = -0.1f, 0.1f ;
}

```

Annexe V. Format fichier cap externe

```
netcdf ovide_new_cap {
dimensions:
    N_DATE_TIME = 10987 ;
variables:
    double JULD_EXT(N_DATE_TIME) ;
        JULD_EXT:long_name = "Julian days relative to REFERENCE_DATE_TIME"
;
        JULD_EXT:_FillValue = -9999999. ;
    float HDG_EXT(N_DATE_TIME) ;
        HDG_EXT:long_name = "Ship heading" ;
        HDG_EXT:units = "degree" ;
        HDG_EXT:valid_min = -360.f ;
        HDG_EXT:valid_max = 360.f ;
        HDG_EXT:_FillValue = -9999999.f ;
}
```

Les dates sont les dates en jours juliens.

Annexe VI. Exemple de fichier NetCDF section

```
netcdf ovide_osite_sec_02ms1 {
dimensions:
    N_DATE_TIME = 2496 ;
    N_LEVEL = 50 ;
    CONST1 = 1 ;
    nbre_section = 3 ;
    DATE_TIME = 14 ;
variables:
    short INDICE(nbre_section) ;
        INDICE:units = "indice" ;
        INDICE:long_name = "indice" ;
    char REFERENCE_DATE_TIME(DATE_TIME) ;
        REFERENCE_DATE_TIME:long_name = "Date of reference for Julian
days" ;
        REFERENCE_DATE_TIME:convention = "YYYYMMDDHHMISS" ;
        REFERENCE_DATE_TIME:_FillValue = " " ;
    double JULD(N_DATE_TIME) ;
        JULD:long_name = "Julian days relative to REFERENCE_DATE_TIME" ;
        JULD:_FillValue = -999999. ;
    float SecLat(N_DATE_TIME) ;
        SecLat:units = "degree" ;
        SecLat:long_name = "Latitude" ;
    float SecLon(N_DATE_TIME) ;
        SecLon:units = "degree" ;
        SecLon:long_name = "Longitude" ;
    float DEPH(N_LEVEL) ;
        DEPH:long_name = "Depth of bin center" ;
        DEPH:units = "meter" ;
        DEPH:valid_min = -12000.f ;
        DEPH:valid_max = 0.f ;
        DEPH:_FillValue = -999999.f ;
    float VVEL_ADCP(N_DATE_TIME, N_LEVEL) ;
        VVEL_ADCP:long_name = "Northward absolute ADCP current velocity" ;
        VVEL_ADCP:units = "meter per second" ;
        VVEL_ADCP:valid_min = -20.f ;
        VVEL_ADCP:valid_max = 20.f ;
        VVEL_ADCP:_FillValue = -999999.f ;
    float UVEL_ADCP(N_DATE_TIME, N_LEVEL) ;
        UVEL_ADCP:long_name = "Eastward absolute ADCP current velocity" ;
        UVEL_ADCP:units = "meter per second" ;
        UVEL_ADCP:valid_min = -20.f ;
        UVEL_ADCP:valid_max = 20.f ;
        UVEL_ADCP:_FillValue = -999999.f ;
    float WVEL_ADCP(N_DATE_TIME, N_LEVEL) ;
        WVEL_ADCP:long_name = "Upward absolute ADCP current velocity" ;
        WVEL_ADCP:units = "meter per second" ;
        WVEL_ADCP:valid_min = -20.f ;
        WVEL_ADCP:valid_max = 20.f ;
        WVEL_ADCP:_FillValue = -999999.f ;
    short NB_PTS(N_DATE_TIME, N_LEVEL) ;
        NB_PTS:long_name = "Number of points used for averaging" ;
    float URMS_ADCP(N_DATE_TIME, N_LEVEL) ;
        URMS_ADCP:long_name = "Root mean square Eastward velocity" ;
        URMS_ADCP:units = "meter per second" ;
        URMS_ADCP:valid_min = -20.f ;
        URMS_ADCP:valid_max = 20.f ;
        URMS_ADCP:_FillValue = -999999.f ;
    float VRMS_ADCP(N_DATE_TIME, N_LEVEL) ;
        VRMS_ADCP:long_name = "Root mean square Northward velocity" ;
        VRMS_ADCP:units = "meter per second" ;
```

```
VRMS_ADCP:valid_min = -20.f ;
VRMS_ADCP:valid_max = 20.f ;
VRMS_ADCP:_FillValue = -999999.f ;
float WRMS_ADCP(N_DATE_TIME, N_LEVEL) ;
WRMS_ADCP:long_name = "Root mean square Upward velocity" ;
WRMS_ADCP:units = "meter per second" ;
WRMS_ADCP:valid_min = -20.f ;
WRMS_ADCP:valid_max = 20.f ;
WRMS_ADCP:_FillValue = -999999.f ;
float ECI(N_DATE_TIME, N_LEVEL) ;
ECI:long_name = "Mean echo intensity" ;
ECI:units = "count" ;
ECI:_FillValue = -999999.f ;

// global attributes:
:DATE_CREATION = "25-Nov-2004" ;
:INST_TYPE = "RD Instruments ADCP" ;
:INST_MODEL = "narrow band" ;
:PROG_CMNT1 = "Section file" ;
:delta_distance = 2. ;
:type_vitesses = "Vitesses corrigees de la maree" ;
:avec_ou_sans_station = "Stations prises en compte" ;
:flag_vitesses = "1" ;
}
```

Annexe VII. Exemple de fichier NetCDF station

```

netcdf ovide_osite_sta_00120x1 {
dimensions:
    N_DATE_TIME = 337 ;
    N_LEVEL = 50 ;
    CONST1 = 1 ;
    nbre_station = 9 ;
    DATE_TIME = 14 ;
variables:
    short INDICE(nbre_station) ;
        INDICE:units = "indice" ;
        INDICE:long_name = "indice" ;
    double NUMERO(nbre_station) ;
        NUMERO:long_name = "numero de station" ;
    char REFERENCE_DATE_TIME(DATE_TIME) ;
        REFERENCE_DATE_TIME:long_name = "Date of reference for Julian
days" ;
        REFERENCE_DATE_TIME:convention = "YYYYMMDDHHMISS" ;
        REFERENCE_DATE_TIME:_FillValue = " " ;
    double JULD(N_DATE_TIME) ;
        JULD:long_name = "Julian days relative to REFERENCE_DATE_TIME" ;
        JULD:_FillValue = -999999. ;
    float StaLat(N_DATE_TIME) ;
        StaLat:units = "degree" ;
        StaLat:long_name = "Latitude" ;
    float StaLon(N_DATE_TIME) ;
        StaLon:units = "degree" ;
        StaLon:long_name = "Longitude" ;
    float DEPH(N_LEVEL) ;
        DEPH:long_name = "Depth of bin center" ;
        DEPH:units = "meter" ;
        DEPH:valid_min = -12000.f ;
        DEPH:valid_max = 0.f ;
        DEPH:_FillValue = -999999.f ;
    float VVEL_ADCP(N_DATE_TIME, N_LEVEL) ;
        VVEL_ADCP:long_name = "Northward absolute ADCP current velocity" ;
        VVEL_ADCP:units = "meter per second" ;
        VVEL_ADCP:valid_min = -20.f ;
        VVEL_ADCP:valid_max = 20.f ;
        VVEL_ADCP:_FillValue = -999999.f ;
    float UVEL_ADCP(N_DATE_TIME, N_LEVEL) ;
        UVEL_ADCP:long_name = "Eastward absolute ADCP current velocity" ;
        UVEL_ADCP:units = "meter per second" ;
        UVEL_ADCP:valid_min = -20.f ;
        UVEL_ADCP:valid_max = 20.f ;
        UVEL_ADCP:_FillValue = -999999.f ;
    float WVEL_ADCP(N_DATE_TIME, N_LEVEL) ;
        WVEL_ADCP:long_name = "Upward absolute ADCP current velocity" ;
        WVEL_ADCP:units = "meter per second" ;
        WVEL_ADCP:valid_min = -20.f ;
        WVEL_ADCP:valid_max = 20.f ;
        WVEL_ADCP:_FillValue = -999999.f ;
    short NB_PTS(N_DATE_TIME, N_LEVEL) ;
        NB_PTS:long_name = "Number of points used for averaging" ;
    float URMS_ADCP(N_DATE_TIME, N_LEVEL) ;
        URMS_ADCP:long_name = "Root mean square Eastward velocity" ;
        URMS_ADCP:units = "meter per second" ;
        URMS_ADCP:valid_min = -20.f ;
        URMS_ADCP:valid_max = 20.f ;
        URMS_ADCP:_FillValue = -999999.f ;
    float VRMS_ADCP(N_DATE_TIME, N_LEVEL) ;
        VRMS_ADCP:long_name = "Root mean square Northward velocity" ;

```

```
VRMS_ADCP:units = "meter per second" ;
VRMS_ADCP:valid_min = -20.f ;
VRMS_ADCP:valid_max = 20.f ;
VRMS_ADCP:_FillValue = -999999.f ;
float WRMS_ADCP(N_DATE_TIME, N_LEVEL) ;
  WRMS_ADCP:long_name = "Root mean square Upward velocity" ;
  WRMS_ADCP:units = "meter per second" ;
  WRMS_ADCP:valid_min = -20.f ;
  WRMS_ADCP:valid_max = 20.f ;
  WRMS_ADCP:_FillValue = -999999.f ;
float ECI(N_DATE_TIME, N_LEVEL) ;
  ECI:long_name = "Mean echo intensity" ;
  ECI:units = "count" ;
  ECI:_FillValue = -999999.f ;

// global attributes:
:DATE_CREATION = "25-Nov-2004" ;
:INST_TYPE = "RD Instruments ADCP" ;
:INST_MODEL = "narrow band" ;
:PROG_CMNT1 = "Station file" ;
:duree_moyenne = 120. ;
:type_vitesses = "Vitesses non corrigees de la maree" ;
:flag_vitesses = "1" ;
}
```

Annexe VIII Exemple de fichier NetCDF de Bathymétrie.

```

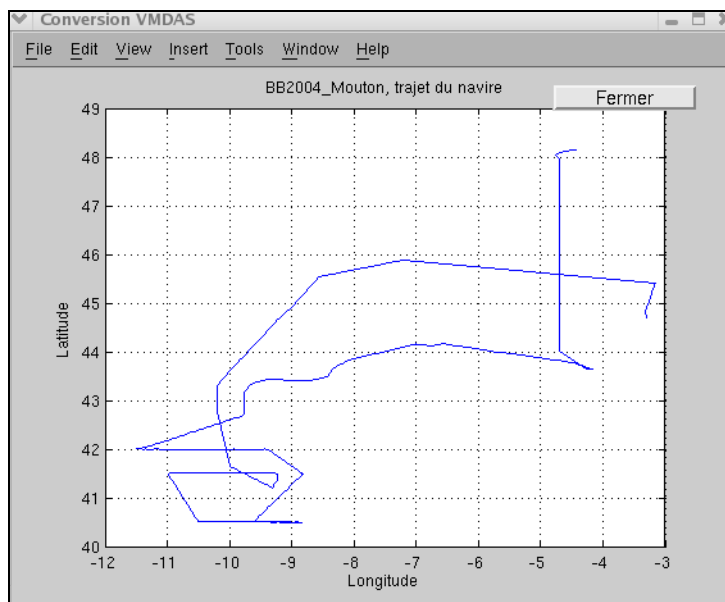
netcdf bathy6min {
dimensions:
    latitude = 1800 ;
    longitude = 3600 ;
variables:
    short z(latitude, longitude) ;
        z:long_name = "etopo6" ;
        z:units = "m" ;
        z:valid_min = -10357.6666666667 ;
        z:valid_max = 0. ;
    float latitude(latitude) ;
        latitude:long_name = "latitude" ;
        latitude:units = "degree_north" ;
        latitude:valid_min = -89.966667175293 ;
        latitude:valid_max = 89.9333343505859 ;
    float longitude(longitude) ;
        longitude:long_name = "longitude" ;
        longitude:units = "degree_east" ;
        longitude:valid_min = -180. ;
        longitude:valid_max = 179.899993896484 ;

// global attributes:
    :bathy_model = "Etopo6" ;
    :history = "Bathymetrie a 6min depuis Etopo2 Smith & Sandwell"
;
    :Conventions = "COARDS" ;
    :Producer_agency = "Ifremer" ;
    :Creation_time = "21-Oct-2010" ;
    :North_latitude = 89.9333343505859 ;
    :South_latitude = -89.966667175293 ;
    :West_longitude = -180. ;
    :East_longitude = 179.899993896484 ;
}

```

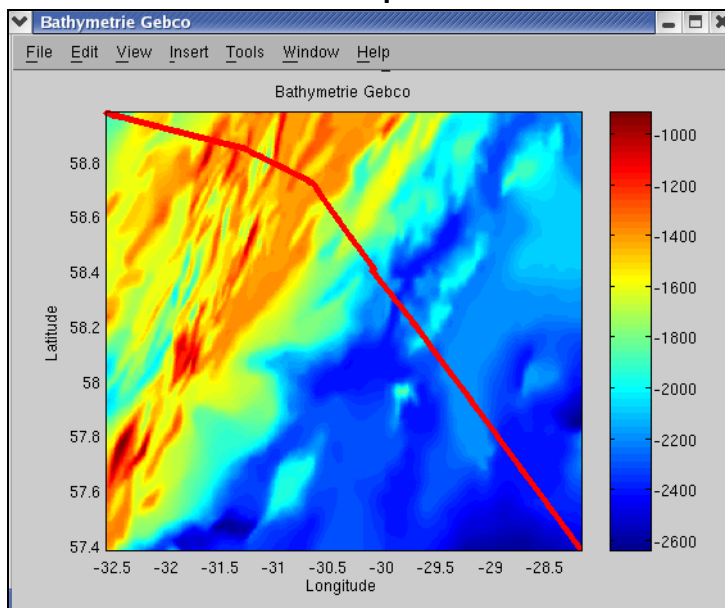
Annexe IX. Tracé de conversion de fichier VMDAS

Lorsque l'on réalise cette étape, les fichiers LTA (ou STA) sont convertis au format NetCDF OceanSite, cette étape génère un tracé permettant de visualiser globalement la campagne.

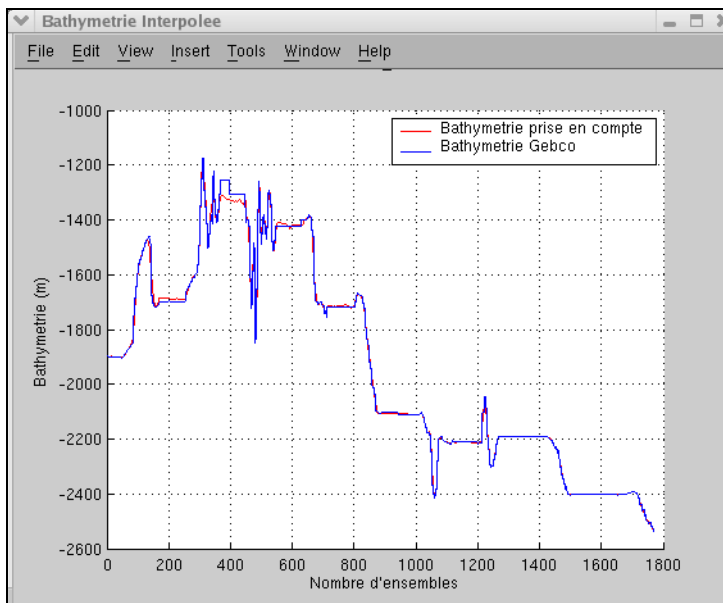


Annexe X. Tracé de contrôle de l'ajout de la bathymétrie

- 1 -

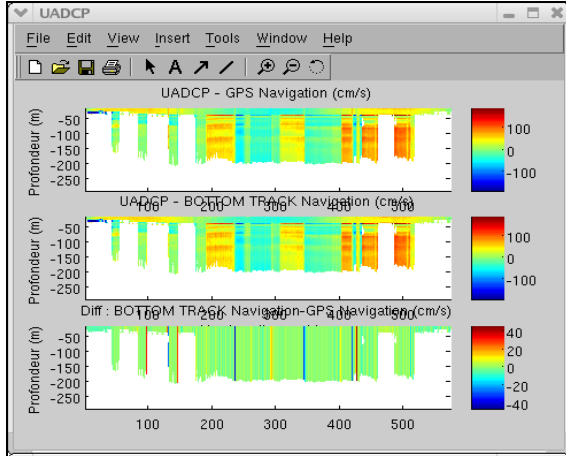


- 2 -

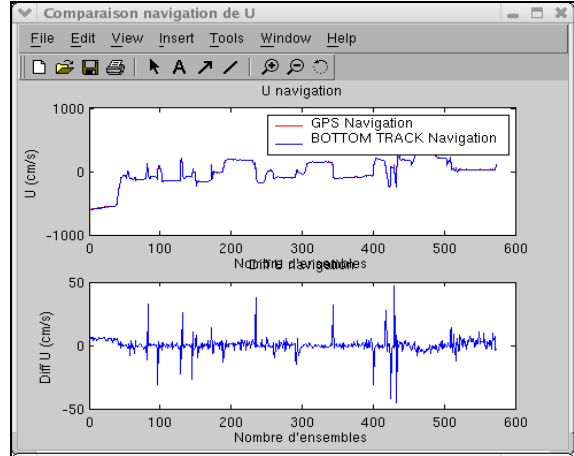


Annexe XI. Changement de navigation

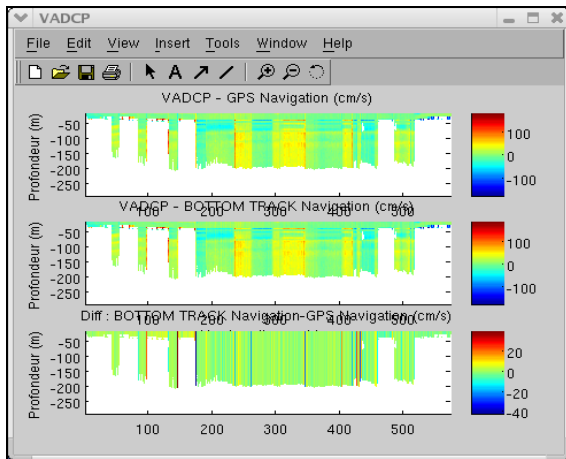
- 1 -



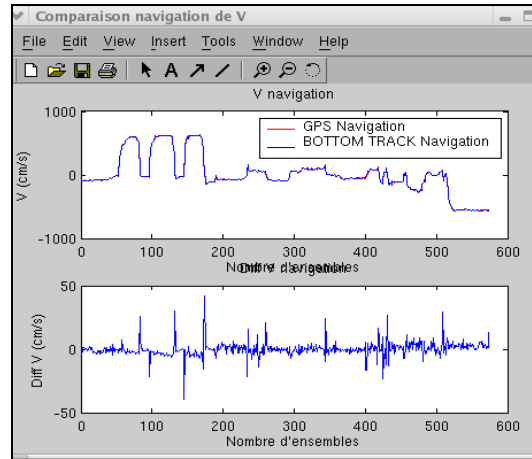
- 3 -



- 2 -

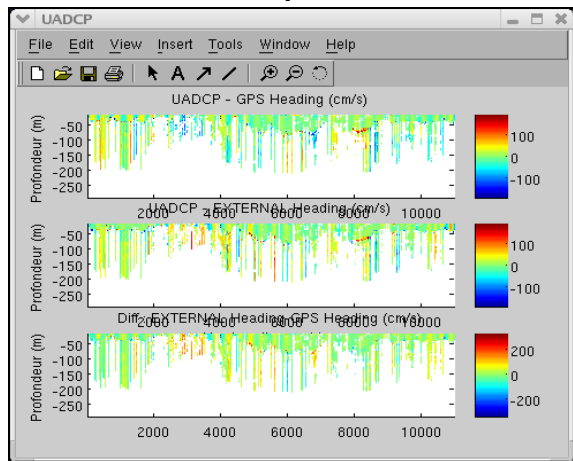


- 4 -

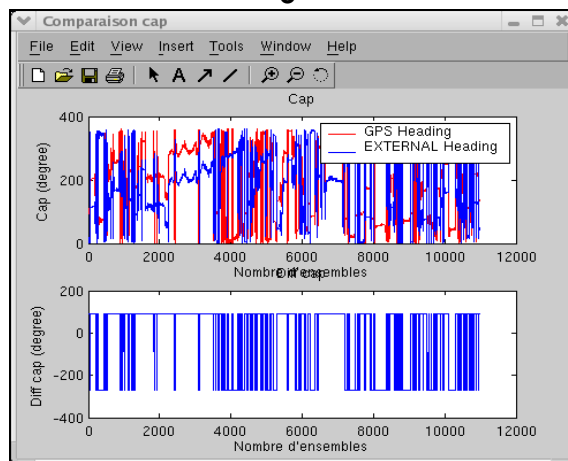


Annexe XII. Changement de cap

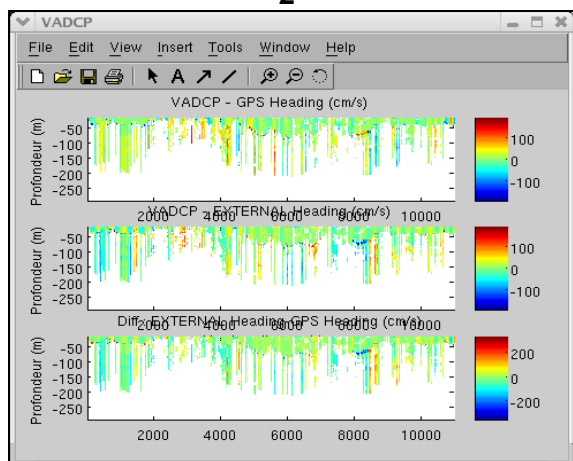
- 1 -



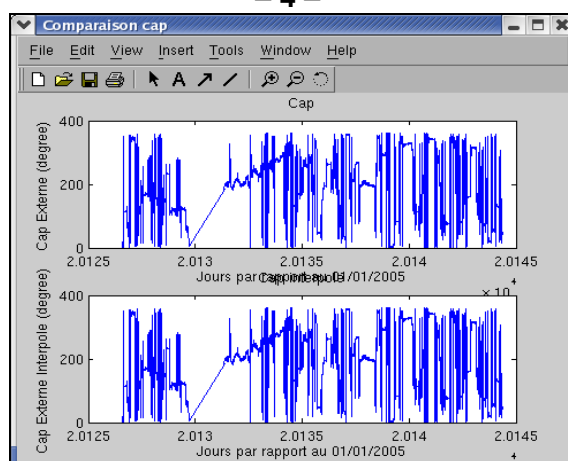
- 3 -



- 2 -

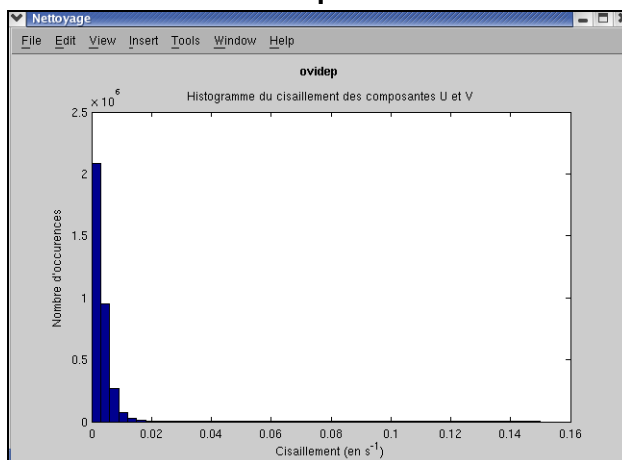


- 4 -

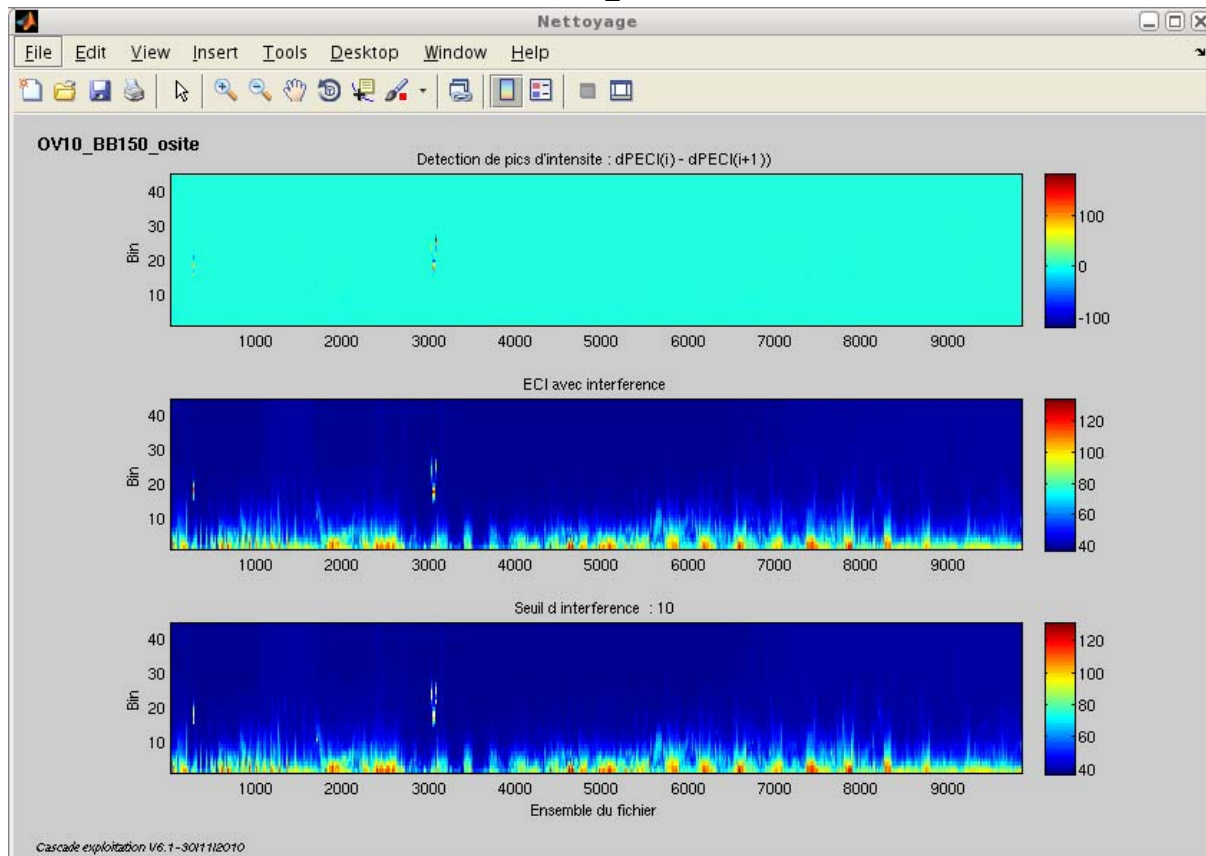


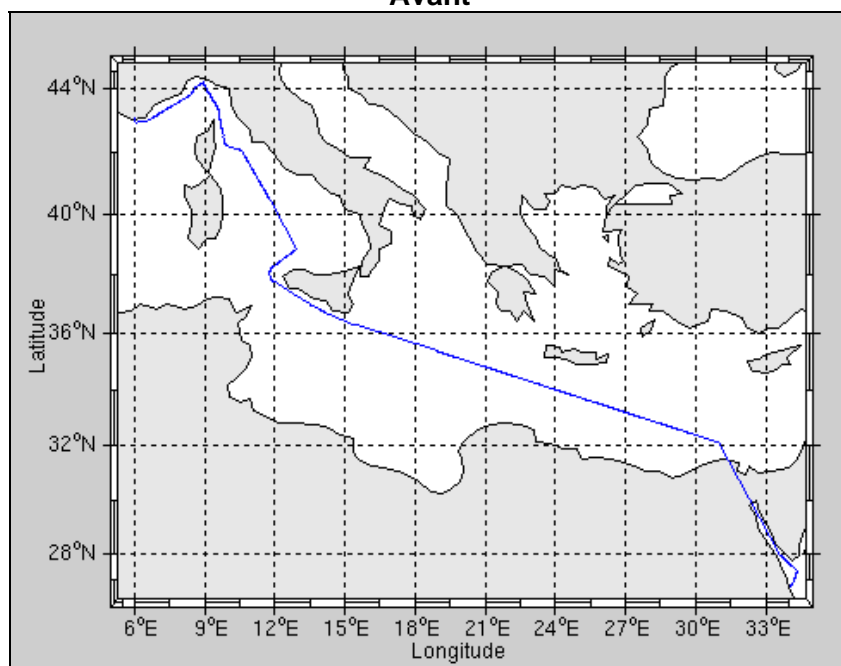
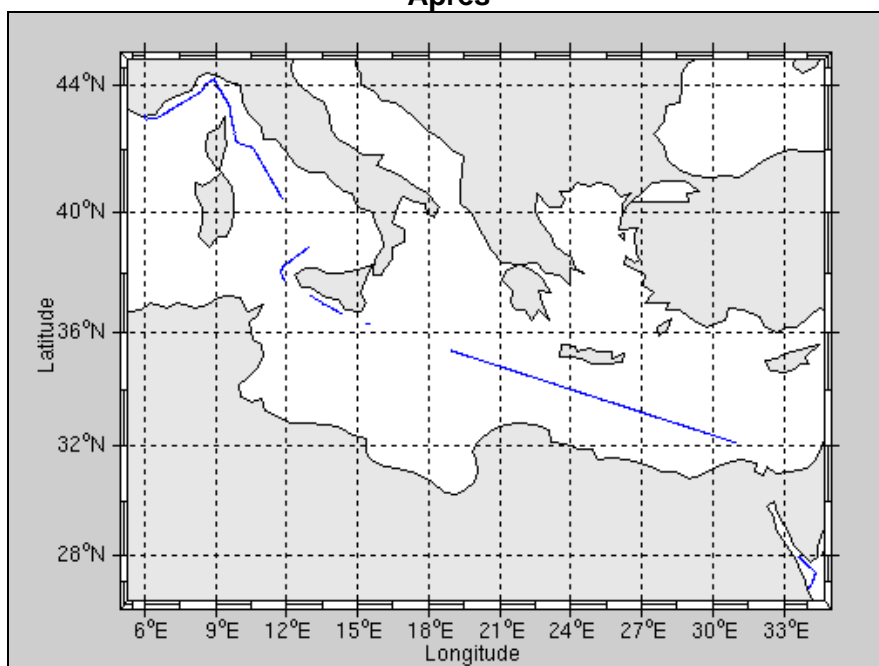
Annexe XIII. Tracé de nettoyage

- 1 -



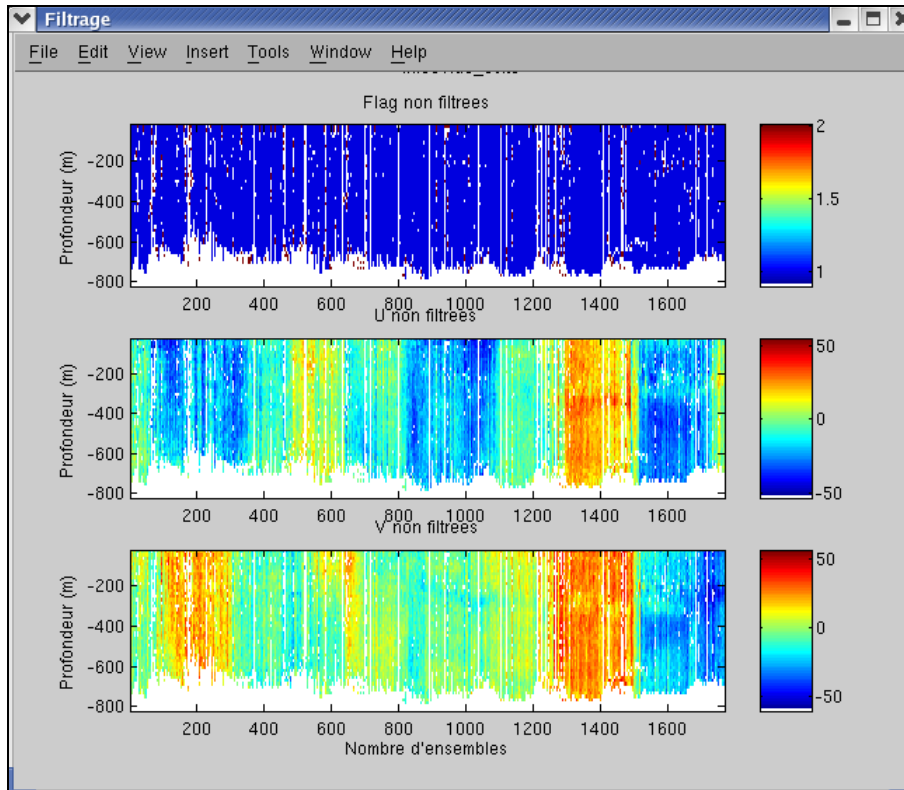
- 2 -



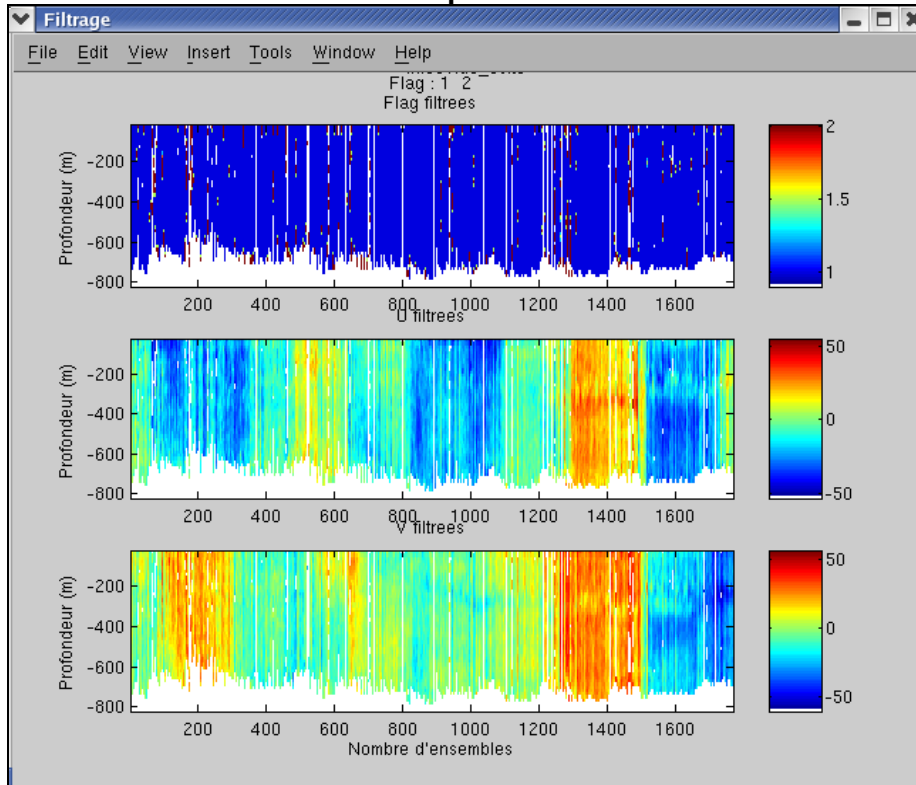
Annexe XIV. Matérialisation des périodes sans mesure**– Avant –****– Après –**

Annexe XV. Filtrage

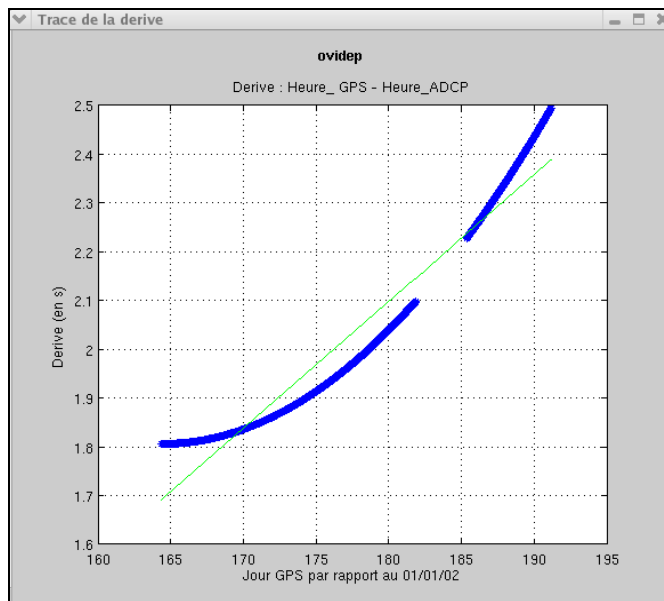
– Avant –



– Après –

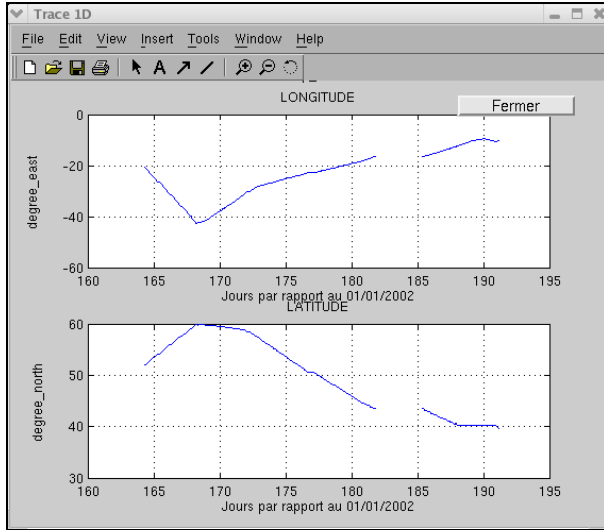


Annexe XVI. Tracé de la dérive

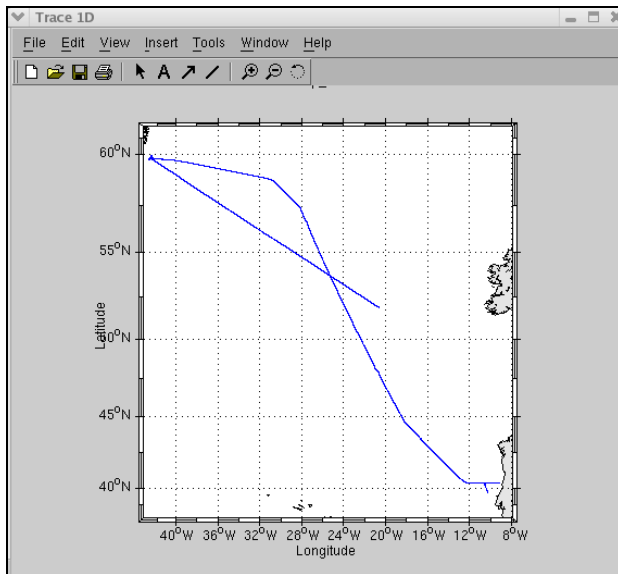


Annexe XVII. Tracé 1D

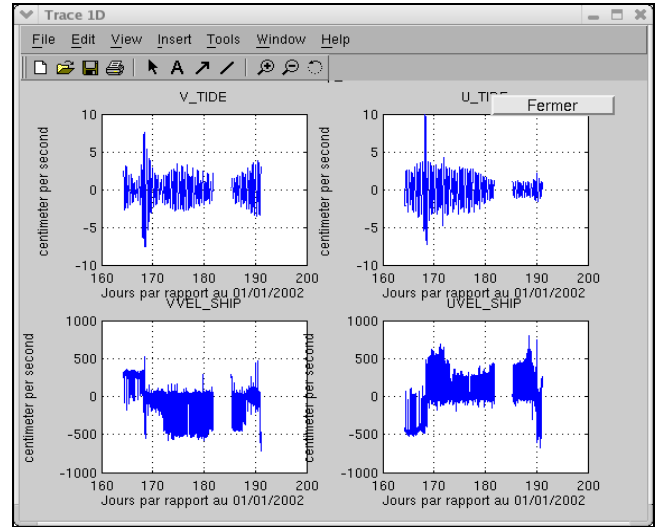
- 1 -



- 2 -

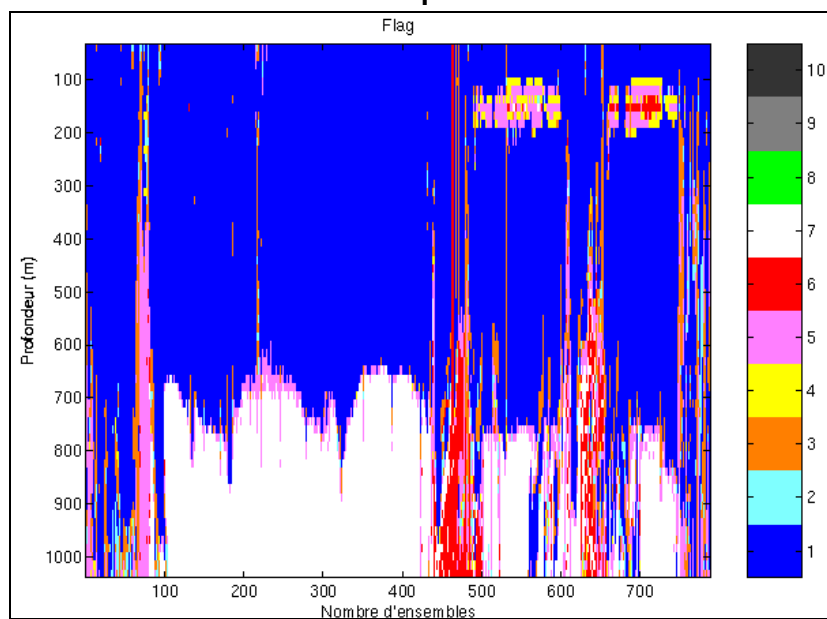


- 3 -

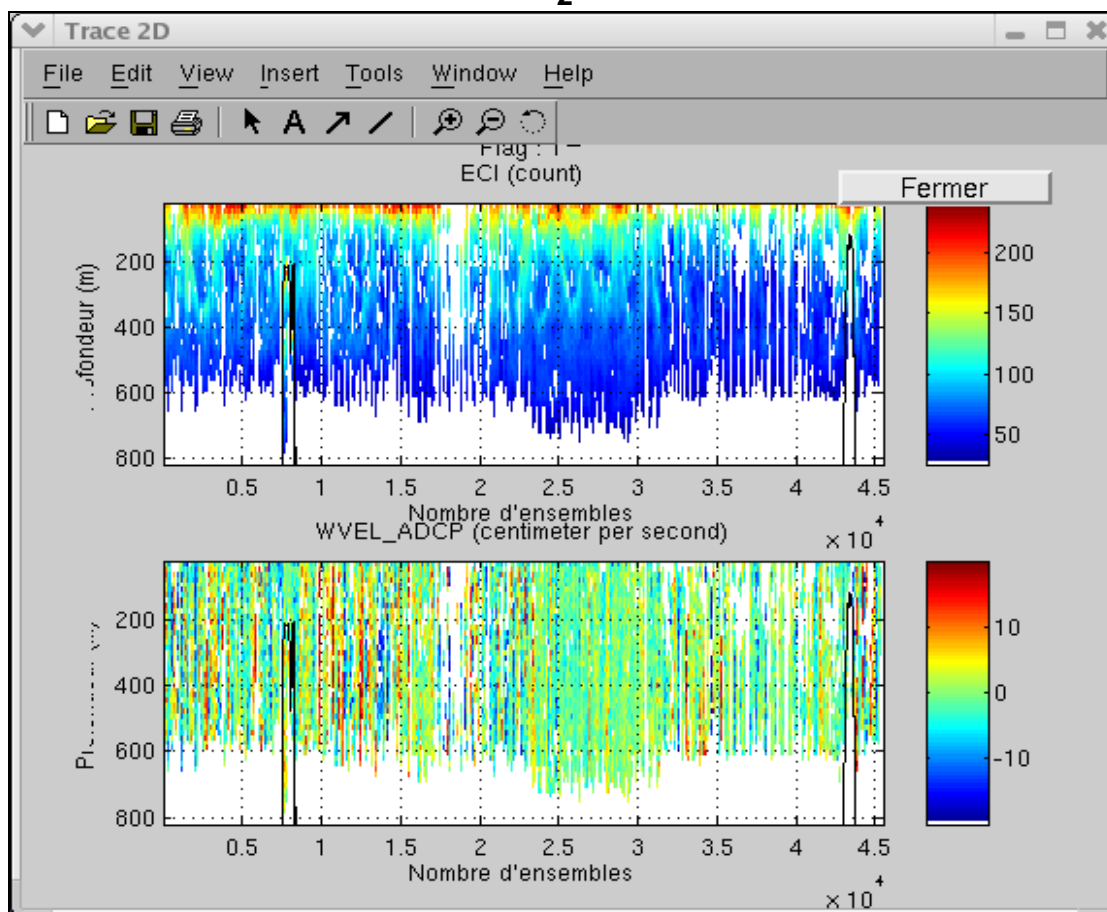


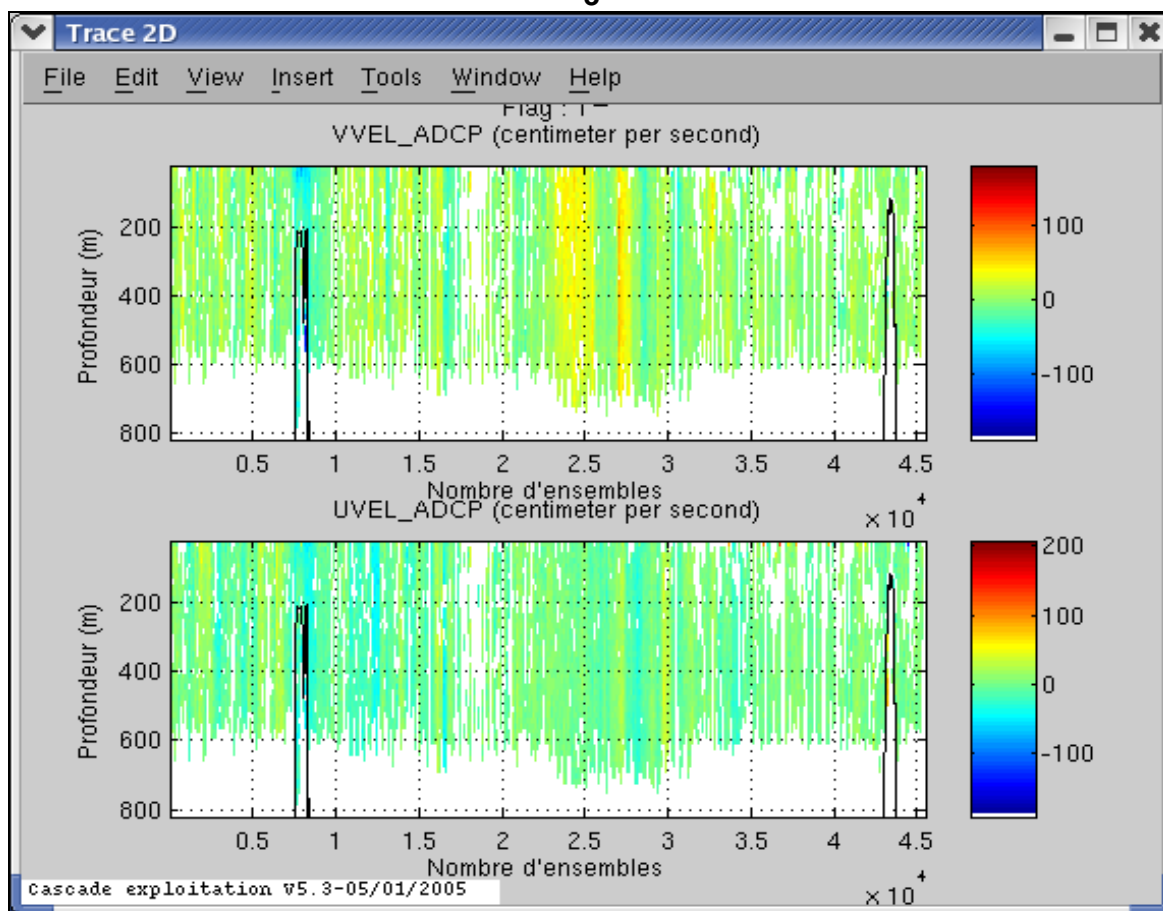
Annexe XVIII. Tracé 2D

- 1 -



- 2 -

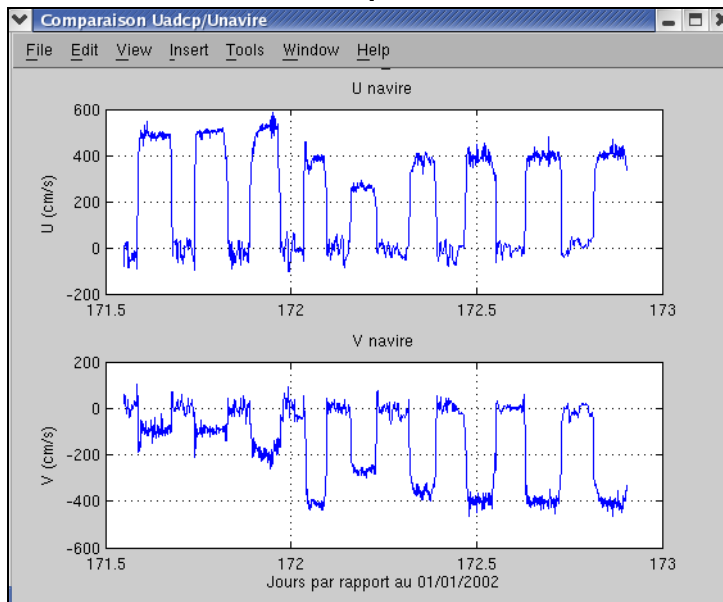




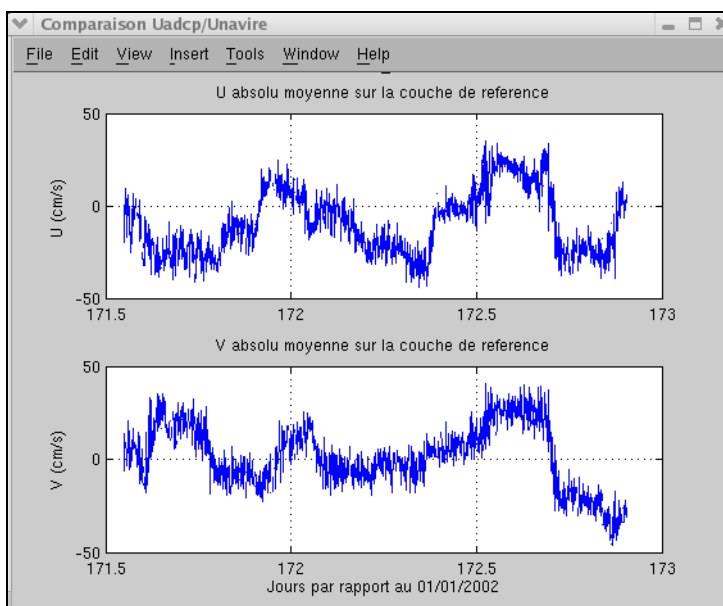
Annexe XIX. Comparaison

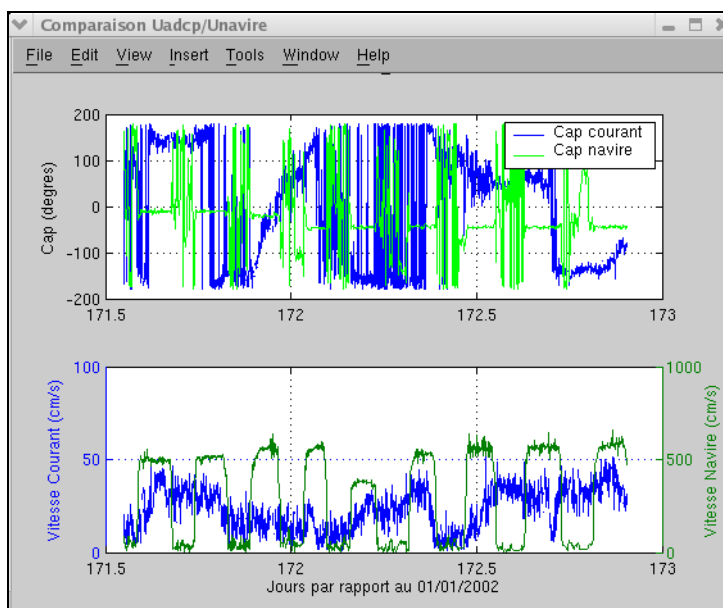
1. Uadcp/Unavire

- 1 -



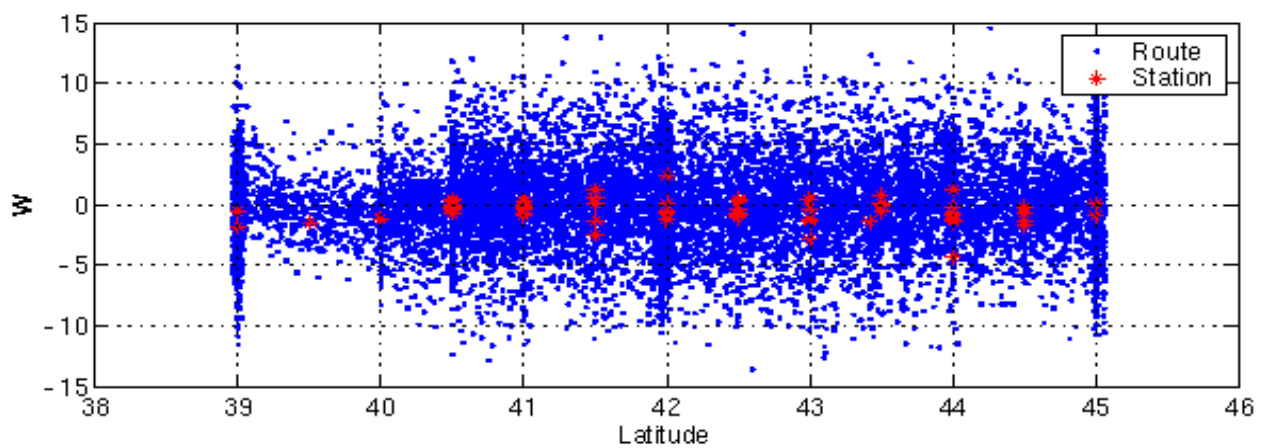
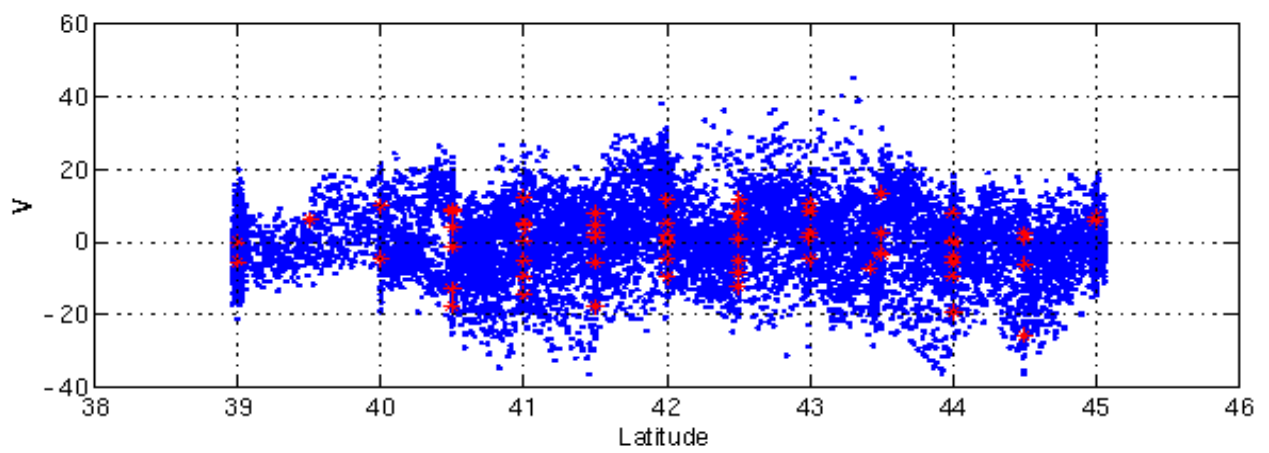
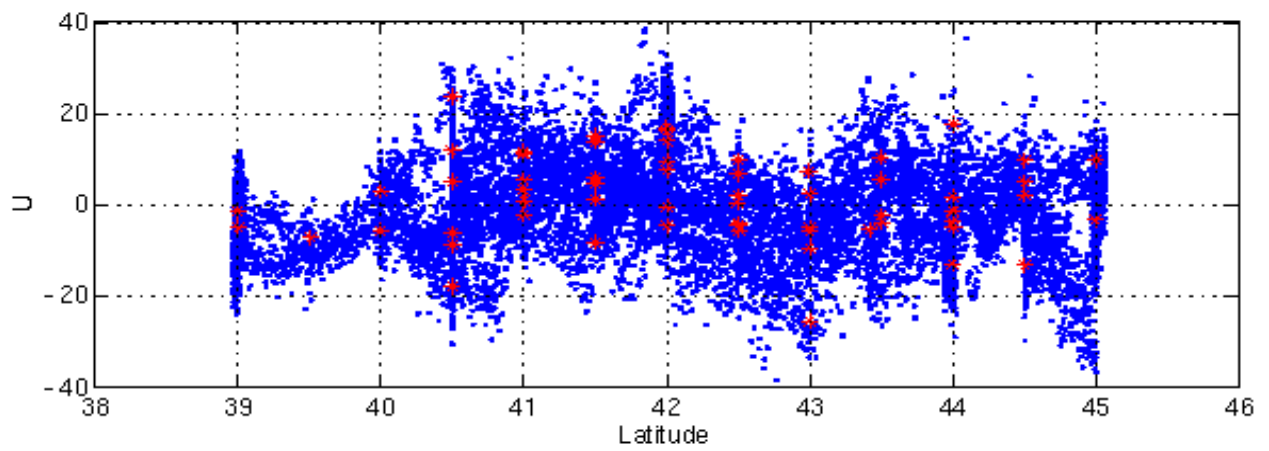
- 2 -



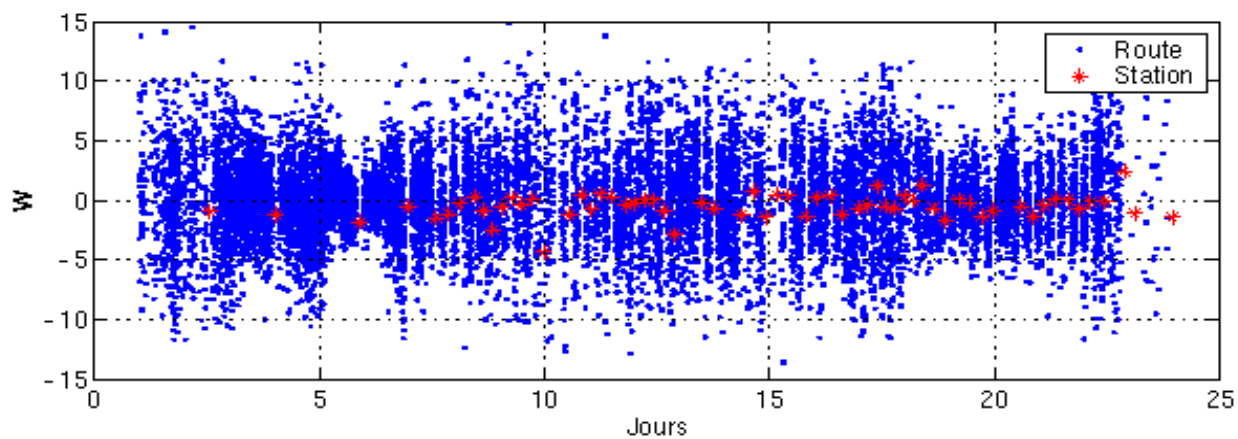
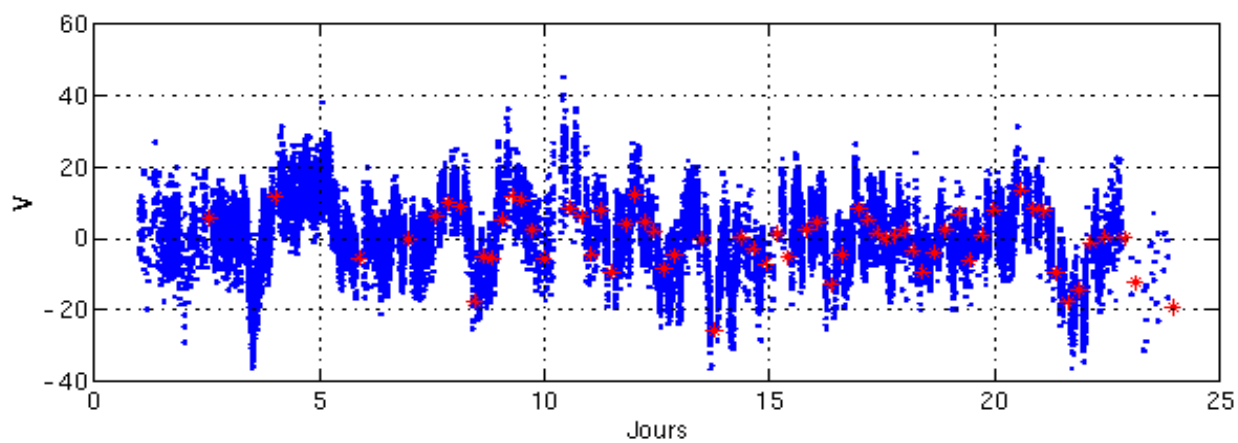
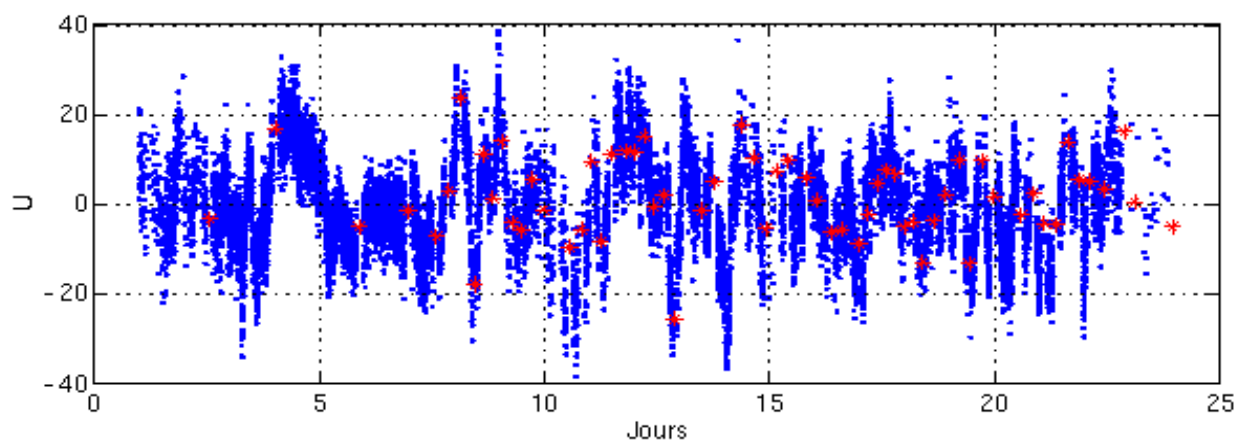


2. Route/Station

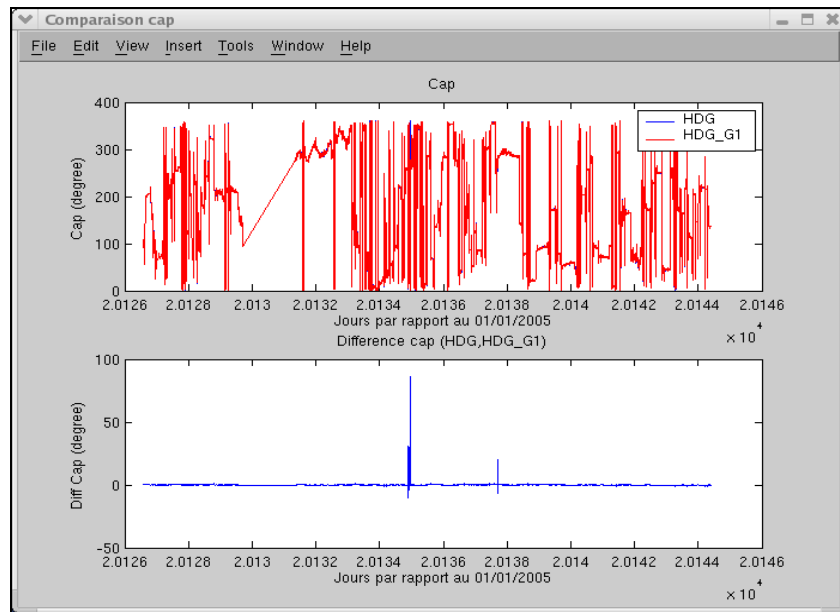
- 1 -



- 2 -

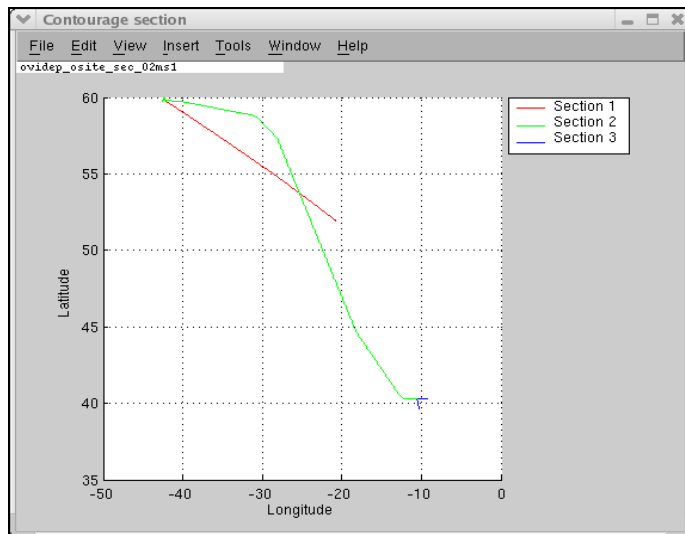


3. Cap

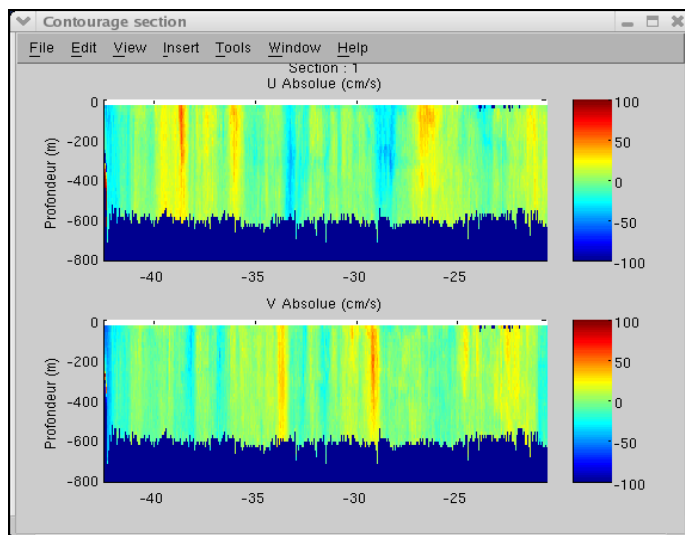


Annexe XX. Exploitation Section – Contourage

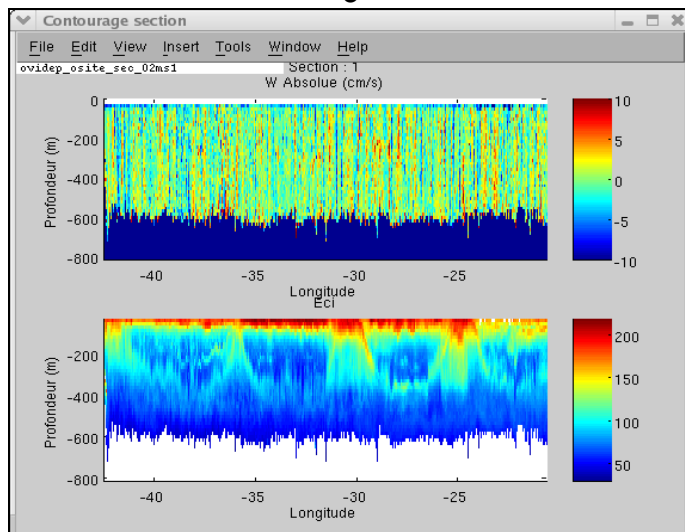
– 1 –



– 2 –

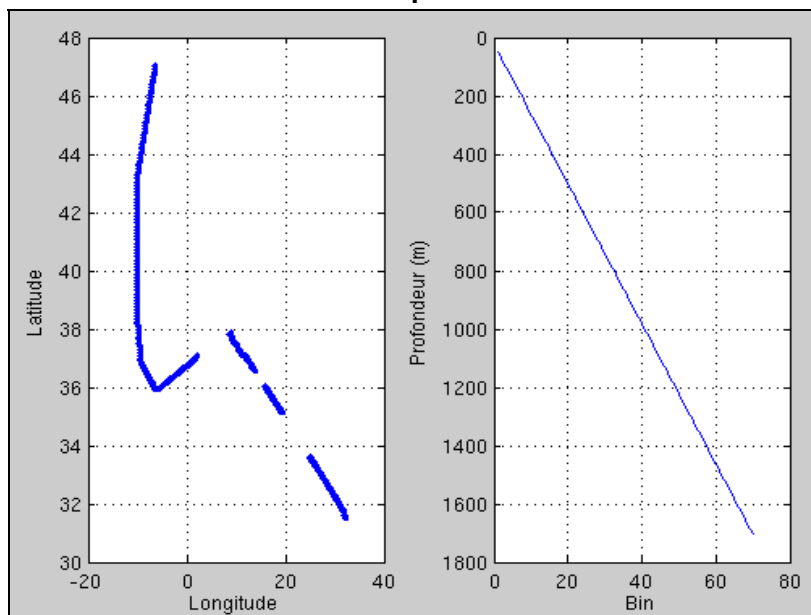


– 3 –

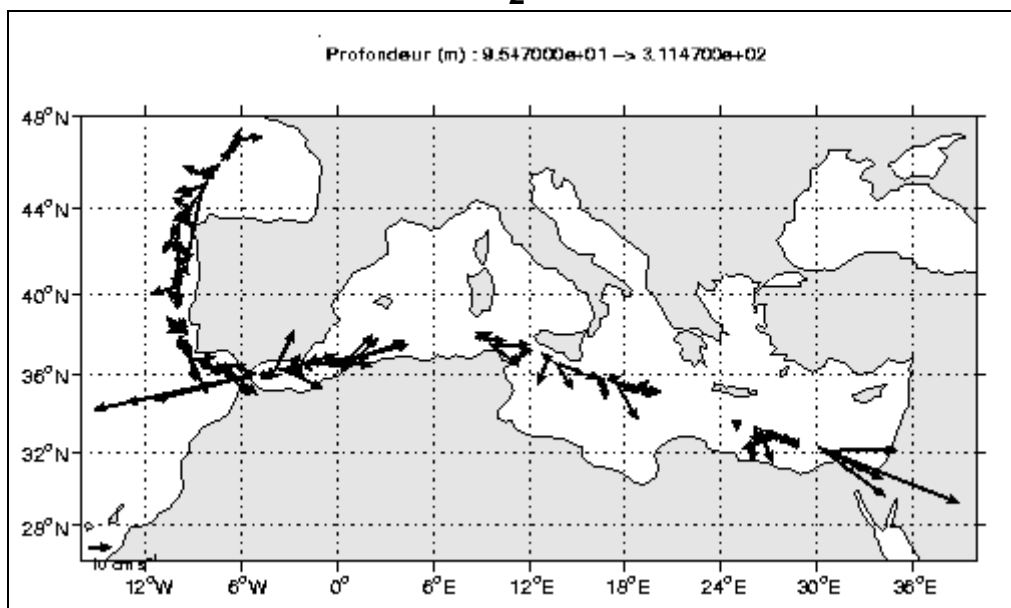


Annexe XXI. Exploitation Section – Vecteur

- 1 -

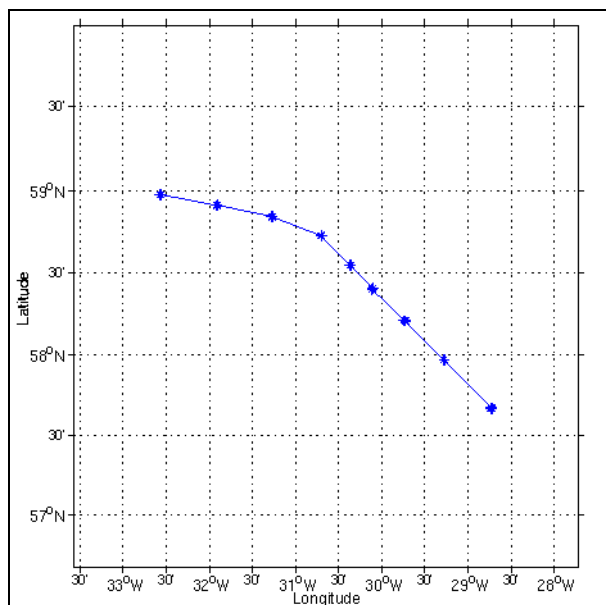


- 2 -

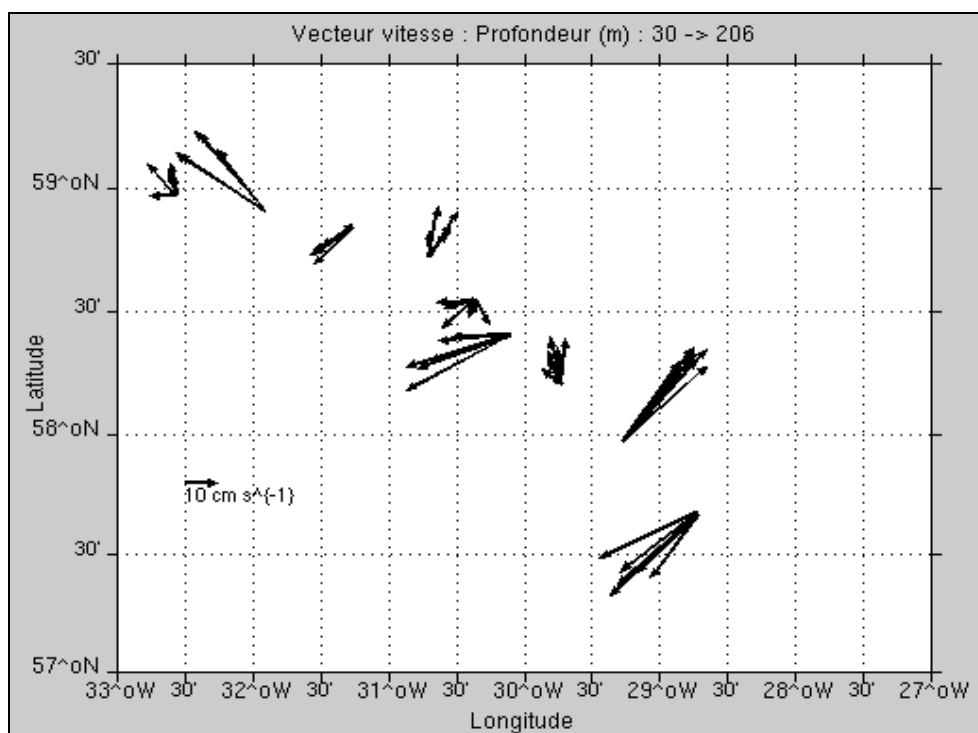


Annexe XXII. Exploitation Station – Vecteur

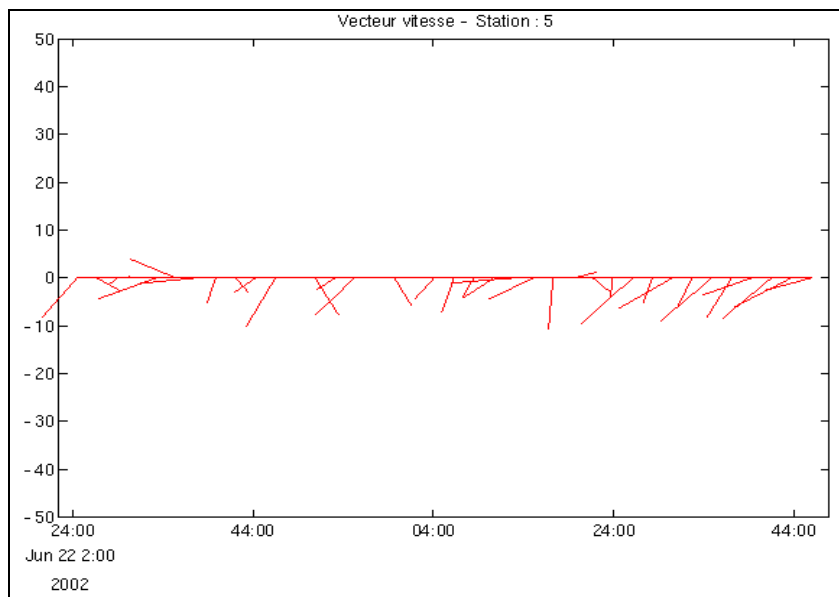
- 1 -



- 2 -



- 3 -



Annexe XXIII. Rapport des dépendances entre fonctions.

Les fonctions surlignées en jaune ont été développées pour le logiciel CASCADE.

En bleu, figurent les fichiers de constantes ou d'appel de variable globale ou fonction externe à CASCADE.

MATLAB (liste des fonctions)	Enfants (fonctions appelées)	Parents (fonctions appelantes)
<u>ce</u>	const CE f_cree_conf f_cree_conf_from_v5 global_CE tabdlg f_annul f_valid f_color_button f_tag_on	
<u>f_Cal_vit_ref</u>		f_VGcor f_lance_cmp_sta_route f_lance_vitadcp_vitna v
<u>f_VGcor</u>	const CE f_Cal_vit_ref f_autonan f_corr_vit_vm f_cree_newvar2 f_test_vars meanoutnan	
<u>f_add_bathy</u>	const CE f_autonan f_cree_newvar f_efface_trace f_test_vars	
<u>f_add_empty_data</u>	const CE f_autonan f_get_fillvalue g_txt_lang g_txt_rep_work global_CE meanoutnan	
<u>f_add_info_nc</u>	const CE f_save_conf	
<u>f_add_tide</u>	f_autonan f_cree_newvar2 f_get_fillvalue f_test_vars	

	global CE greg_0h jul_0h tide_pred	
f_aide_date	ajout_title const CE f_autonan g_txt lang global CE indique_date jul_0h	
f_aide_date>indique_date	greg_0h	
f_appel_filtre	ajout_title ajout version exploit ce_txt U ce_txt V ce_txt W ce_txt flag const CE f_autonan f_cree_newvar f_filtre f_test_vars g_int type filtrage g_txt filename nc g_txt lang g_txt output type g_txt rep work global CE h_trait	
f_autonan		f_VGcor f_add_bathy f_add_empty_data f_add_tide f_aide_date f_appel_filtre f_bad_flag f_change_navig f_comp_bt_ship f_contour_sec f_cree_sec f_lance_cmp_cap f_lance_cmp_sta_route f_lance_vitadcp_vitna v f_net_vit_flag4a8 f_netvit_cour f_recal_cap f_recal_nav f_sta_part_tot f_traceld2d f_trace_derive f_vec_sec f_vect_sta
f_autonan2		f_infos_camp

f_bad_flag	const CE f_autonan f_creer_newvar f_test_vars g_date_last g_date_start g_int_ens1 g_int_ens2 g_int_type_inval g_txt_lang global CE jul_0h	
f_change_navig	f_autonan	f_recal_nav
f_comp_bt_ship	const CE f_autonan	
f_contour_sec	ajout_title ajout_version_exploit ce_txt_lat ce_txt_lon const CE f_autonan g_txt_lang g_txt_output_type g_txt_rep_work global CE	
f_conv_OS	buf_long const CE decode_fix decode_sysconfig dist_km f_creer_newvar2 g_txt_lang g_txt_rep_work global CE greg_0h jul_0h meanoutnan to_day to_signed	
f_corr_heading	const CE degrad	f_recal_cap
f_corr_vit_vm	const CE degrad f_mat_navgeo	f_VGcor
f_creat_cdf_sec_sta	f_creer_newvar2	f_cree_sec f_sta_part_tot
f_cree_conf	global CE	
f_cree_conf_from_v5	V max	

	amplitude angle_desalignement cis_max date1 date2 detect_fond fact_cis filename global_CE iref_exploit nb_a_moyenner nb_ecart pitch_biais seuil_interf type_filtirage vdifflim w_max	
f_cree_rep		
f_cree_sec	ce_msg_titre const_CE degrad f_autonan f_creat_cdf_sec_sta f_get_fillvalue f_test_vars greg_0h jul_0h meanoutnan	
f_creer_newvar		f_add_bathy f_appel_filtre f_bad_flag f_recal_cap f_recal_nav
f_creer_newvar2		f_VGcor f_add_tide f_conv_OS f_creat_cdf_sec_sta f_netvit_cour
f_efface_trace	const_CE	f_add_bathy f_netvit_cour f_recal_cap f_recal_nav
f_filtre		f_appel_filtre
f_get_cref	f_test_vars global_CE	
f_get_fillvalue		f_add_empty_data f_add_tide f_cree_sec f_recal_cap f_sta_part_tot
f_info_trait	const_CE f_test_vars	
f_infos_camp	const_CE	

	f_autonan2 f_test_vars f_verif_correlation_nav_current g_int_cref_max g_int_cref_min g_txt_filename_nc g_txt_lang global_CE greg_0h jul_0h meanoutnan	
f_infos_flags	const_CE	
f_invalid_dates	global_CE greg_0h jul_0h	
f_invalid_ens	const_CE g_txt_lang global_CE	
f_lance_cmp_cap	ajout_version_exploit cap1 cap2 const_CE degrad f_autonan	
f_lance_cmp_sta_route	ajout_title ajout_version_exploit ce_txt_U ce_txt_V ce_txt_W ce_txt_lat ce_txt_lon ce_txt_station const_CE f_Cal_vit_ref f_autonan h_trait jul_0h	
f_lance_vitadcp_vitnav	ajout_title ajout_version_exploit ce_txt_U ce_txt_V const_CE f_Cal_vit_ref f_autonan greg_0h h_trait jul_0h radeg	
f_mat_navgeo	const_CE degrad	f_corr_vit_vm

f_net_vit_flag2	meanoutnan	f_netvit_cour
f_net_vit_flag4a8	ajout titre ajout version exploit ce msg titre const CE f autonan f test vars g txt lang g txt output type g txt rep work global CE	f_netvit_cour
f_net_vitesse	ajout titre const CE g txt lang global CE medianoutnan	f_netvit_cour
f_net_vitesse_test	ajout titre const CE g txt lang global CE medianoutnan	f_netvit_cour
f_netvit_cour	ajout titre ajout version exploit const CE f autonan f creer newvar2 f efface trace f net vit flag2 f_net_vit_flag4a8 f net vitesse f net vitesse test f test vars g int cvmax g int detect fond g int nb ecart g int nb prof g int scisail g int sinterf g int vdiflim g int vmax g int vvmx g txt lang g txt output type g txt rep work global CE h trait	
f_osite_type	const CE	
f_read_HDG		
f_read_plot1d2d	const CE	

	f_test_vars	
f_recal_cap	ajout version exploit ce txt cap const CE f_autonan f_corr_heading f_crear_newvar f_efface_trace f_get_fillvalue f_test_vars greg_0h h_trait jul_0h	
f_recal_nav	ajout version exploit const CE f_autonan f_change_navig f_crear_newvar f_efface_trace f_get_fillvalue f_test_vars g_txt_filename_trinav g_txt_lang g_txt_output_type g_txt_rep_work global CE h_trait jul_0h	
f_save_conf	global CE	f_add_info_nc
f_sta_part_tot	ajout version exploit ce msg titre ce txt U ce txt V const CE f_autonan f_creat_cdf_sec_sta f_get_fillvalue f_test_vars greg_0h jul_0h meanoutnan	
f_test_vars		f_VGcor f_add_bathy f_add_tide f_appel_filtre f_bad_flag f_cree_sec f_get_cref f_info_trait f_infos_camp f_net_vit_flag4a8 f_netvit_cour f_read_plot1d2d

		f_recal_cap f_recal_nav f_sta_part_tot f_trace1d2d f_trace_derive f_write_cref
f_trace1d2d	ajout version exploit ce txt flag ce txt lat ce txt lon const CE f_autonan f_test_vars greg_0h jul_0h m_coast_new m_grid m_ll2xy m_proj update_date	
f_trace1d2d>update_date	m_xy2ll	
f_trace_derive	ajout version exploit const CE f_autonan f_test_vars greg_0h jul_0h	
f_val_chkb		
f_vec_sec	ce txt lat ce txt lon const CE f_autonan g txt output type g txt rep_work global_CE m_coast_new m_contourf m_grid m_gshhs_i m_proj m_text m_vec	
f_vect_sta	ajout title ajout version exploit ce txt lat ce txt lon const CE f_autonan fic trace finite jul_0h m_coast_new m_contourf m_arid	

	m_gshhs_i m_ll2xy m_proj m_vec texte_station timeplt	
f_verif_correlation_nav_current	subfunction fillval2nan meanoutnan	f_infos_camp
f_verif_correlation_nav_current>fillval2nan		
f_write_cref	f_test_vars g_int_cref_max g_int_cref_min global_CE	

