



INSTITUT FRANCAIS DE RECHERCHE SCIENTIFIQUE POUR LE
DEVELOPPEMENT EN COOPERATION

Traitement des données L-ADCP

Yves Gouriou¹ et Catherine Hémon²

17 février 1997

Centre ORSTOM de Cayenne : Documents Scientifiques N° 0.P. 21

¹ Centre ORSTOM - BP 165 - 97323 Cayenne – ² Ifremer, ZI de la pointe du diable, CS10070, 29280 Plouzané

TABLE DES MATIERES

I - Introduction		p 4
II - Principe du traitement		p 5
II - 1 Rappel du principe de la mesure		p 5
II - 2 Calcul de la profondeur		p 5
II - 3 Réflexions sur le fond		p 6
II - 4 Création d'un profil vertical de vitesses horizontales		p 6
II - 5 Calcul de la vitesse absolue		p 6
III - La chaîne de traitement		p 9
III - 1 Présentation rapide de la chaîne de traitement		p 9
III - 2 Format des données brutes		p 9
III - 3 Décodage des données brutes	LADCPBIN.C	p 10
III - 3 - a Organigramme du programme		p 10
III - 3 - b Lecture du fichier de description de la station	lec_descr()	p 11
III - 3 - c Lecture des enregistrements binaires brutes	lect_bbadcp()	p 13
III - 3 - d Ecriture des cycles de mesures en ASCII et libération de la mémoire	ecr_ens_brut() libere_mem()	p 13
III - 3 - e Ecriture du fichier paramètre	ecr_para()	p 15
III - 4 Calcul du profil de vitesses horizontales	LADCPUV.C	p 16
III - 4 - a Organigramme du programme		p 16
III - 4 - b Généralités		p 16
III - 4 - c Nettoyage des données brutes	nettoyage()	p 19
III - 4 - d Première estimation de la profondeur	prof_max() ind_pmax() w_moyen()	p 19
III - 4 - e Rejet des données perturbées par les réflexions sur le fond de l'océan	rej_fond()	p 20
III - 4 - f Calcul de la profondeur d'un cycle de mesure et correction des vitesses	corr_vit()	p 22
III - 4 - g Résolution verticale du profil de vitesse		p 23
III - 4 - h Test sur les vitesses horizontales	test_dm_vit() test_et_vit()	p 23
III - 4 - i Calcul des gradients verticaux de vitesses horizontales dans chaque cellule	alloc_struct() grad_brut()	p 24
III - 4 - j Test sur les gradients verticaux de vitesses horizontales	test_dm_gra() test_et_grad()	p 25
III - 4 - k Calcul des gradients verticaux de vitesses horizontales dans chaque tranche d'eau	grad_moy()	p 25

III - 4 - 1	Calcul de la vitesse absolue	ec_typ_gra() vit_rel() calc_deriv() ecr_res() vit_moy()	p 26
III - 4 - m	Traitement séparé des profils descente et montée		p 28
III - 5	Lecture de la configuration du L-ADCP	LADCPCNF.C	p 29
IV	Les fichiers utilisés et créés par la chaîne de traitement		p 31
V	Visualisation des différentes étapes		p 43
Annexe A	Organisation et Installation du logiciel		p 48
Annexe B	Liste des programmes et fonctions de la chaîne de traitement		p 51
Annexe C	Structures des données utilisées par le programme		p 53
Annexe D	Exemple de fichier de configuration pour un L-ADCP		p 56

I - Introduction

Ce document présente la chaîne de traitement des données acquises par un courantomètre à effet DOPPLER (ADCP : Acoustic Doppler Current Profiler) BB-150 KHz de marque RDI, fixé à une bathysonde afin d'obtenir des profils de courant absolu de la surface au fond de l'océan. La conception de ce logiciel a été initiée par I.Ogier³, élève à l'ISITV⁴, au cours d'un stage de fin d'année, au printemps 1995, au centre ORSTOM de Cayenne. L'algorithme utilisé est en grande partie inspiré d'un logiciel transmis par J.Fischer de IFM⁵ Kiel et décrit en détail dans la publication suivante :

J.Fischer and M.Visbeck, Deep velocity profiling with self_contained ADCPs, *J. Atmos. Oceanic Technol*, 10(5), 764-773, 1993.

Ce document se contente de donner des indications générales sur le fonctionnement de chacun des programmes et fonctions composant la chaîne de traitement. Il est nécessaire d'avoir assimilé le principe de fonctionnement d'un ADCP pour bien utiliser cette chaîne de traitement. Celui-ci est décrit dans le manuel fournit par la société RDI :

Direct Reading and Self-Contained Broadband Acoustic Doppler Current Profiler, Technical Manual, RD Instrument, 1995.

³I.Ogier, Elaboration de la chaîne d'analyse et de traitement des données d'un courantomètre à effet Doppler, Rapport de Stage ISITV, Centre ORSTOM de Cayenne, 1995.

⁴ Institut des Sciences de l'Ingénieur de Toulon et du Var

⁵Institut Für Meereskunde - Kiel - Germany

II - Principe du traitement

II - 1 Rappel du principe de la mesure⁶

Le courantomètre à effet DOPPLER (ADCP : **A**coustic **D**oppler **C**urrent **P**rofiler) émet une onde sonore ('ping' en anglais) qui se réfléchit sur les particules en suspension dans l'eau de mer. Ces particules sont supposées sans mouvement propre; leur vitesse représente alors la vitesse de la masse d'eau dans laquelle elles se trouvent. La fréquence de cette onde est modifiée par la réflexion sur les particules en mouvement. Le décalage en fréquence (effet Doppler), entre l'onde émise et l'onde réfléchie, est directement proportionnel à la composante de vitesse des particules suivant l'axe de propagation de l'onde. L'utilisation de plusieurs ondes sonores, en général 4 faisceaux ('beam' en anglais) orientés différemment, permet par simple calcul trigonométrique, de calculer les vitesses horizontales et verticale des courants.

L'ADCP a la possibilité de réaliser une moyenne des impulsions ('ping') émises afin de diminuer le bruit. Un cycle de mesures ('ensemble' en anglais) est donc le résultat de la moyenne de plusieurs impulsions ('pings').

A chaque cycle de mesures, l'ADCP évalue les composantes horizontales et verticale de la vitesse du courant sur un profil dont la portée et la résolution verticale dépendent de la configuration de l'appareil. Le profil de vitesse est découpé, par l'ADCP, en segments uniformes appelés 'cellules' ('depth cells' ou 'bins' en anglais). La vitesse associée à chaque cellule est une moyenne des vitesses sur l'épaisseur de la cellule.

Lorsqu'un ADCP est couplé à une bathysonde (L-ADCP : Lowered ADCP), il réalise, lors de la descente et la montée, des profils successifs qui ont des zones de recouvrement, c'est à dire que pour une même profondeur il existe plusieurs mesures de vitesse. La construction d'un profil de vitesses horizontales absolues se fait en associant tous ces profils. Les grandes étapes du traitement sont brièvement décrites jusqu'à la fin de ce chapitre. Le détail du traitement est exposé au chapitre III.

II - 2 Calcul de la profondeur

La combinaison des différents profils de vitesse mesurés par le L-ADCP, pour obtenir un profil unique de la surface au fond, nécessite de connaître la profondeur de chaque cellule; or le L-ADCP dont nous disposons n'est pas équipé de capteur de pression. Il faut donc utiliser une méthode indirecte pour calculer ces profondeurs.

Le L-ADCP enregistre pour chaque cycle de mesure l'heure d'acquisition. Par ailleurs le L-ADCP calcule les composantes horizontales et verticale de la vitesse du courant dans chaque cellule. Dans l'océan la vitesse verticale des courants est inférieure de plusieurs ordres de grandeur à la vitesse verticale de la bathysonde (~1 m/s) et donc du L-ADCP. La vitesse verticale que mesure le L-ADCP peut donc être confondue avec la vitesse verticale de la bathysonde. L'intégration dans le temps de la vitesse verticale permet alors de calculer la profondeur de chaque cycle de mesures. Le traitement des profils de descente et de montée n'étant pas séparé, la profondeur en fin d'intégration doit être nulle (le L-ADCP est en surface en fin de station). La différence entre la

⁶ Principles of operation: A practical primer, RD Instrument, pp 37, 1999.

profondeur calculée et la profondeur théorique est une indication de la qualité de l'intégration. Afin de diminuer l'erreur sur l'estimation des profondeurs, la différence d'immersion observée en fin d'intégration est répartie linéairement sur l'ensemble des profils.

II - 3 Réflexions sur le fond

Lorsque le L-ADCP est proche du fond de l'océan, les ondes sonores sont réfléchies par celui-ci et perturbent le signal réfléchi par les particules en suspension et reçu par le L-ADCP. Ce sont d'abord les cellules les plus proches du fond qui sont perturbées et au fur et à mesure que le L-ADCP s'approche du fond, les cellules les plus proches des bases acoustiques. Le profil d'Amplitude d'Echo Réfléchi présente un maximum d'énergie dans les cellules dont le signal réfléchi est perturbé par la réflexion sur le fond. Le logiciel détecte ce maximum et élimine ces données. Lorsque le L-ADCP est très proche du fond, le profil d'Amplitude d'Echo Réfléchi ne présente plus de maximum, toutes les mesures sont perturbées par les réflexions sur le fond. Ces cycles de mesures sont éliminés et il peut y avoir un trou de plusieurs minutes dans les données.

II - 4 Création d'un profil vertical de vitesses horizontales

Les mesures de vitesses faites par le L-ADCP sont une combinaison des vitesses des courants et des mouvements de l'appareil. Lors d'un cycle de mesure, la vitesse mesurée dans chaque cellule est la somme de la vitesse du courant dans cette cellule et de la vitesse due au mouvement propre du L-ADCP. Cette dernière composante est identique pour chaque cellule d'un même cycle de mesure.

Pour un cycle de mesures donné, la différenciation verticale des vitesses mesurées élimine donc la composante de vitesse due aux mouvements du L-ADCP, mais aussi la composante moyenne de la vitesse.

Afin de créer un profil vertical de vitesses horizontales, il faut combiner les différents cycles de mesure dont on connaît maintenant la profondeur. Pour ce faire, le logiciel 'découpe' l'océan, entre la surface et le fond, en tranches d'eau dont l'épaisseur est choisie égale à la hauteur d'une cellule. Les vitesses et les gradients verticaux de vitesses horizontales calculés pour chaque cellule sont répartis et moyennés dans chaque tranche d'eau. A partir de ces profils de gradients verticaux de vitesses horizontales et de vitesse moyenne brute il est maintenant possible de calculer un profil de vitesse absolue.

II - 5 Calcul de la vitesse absolue

Nous avons repris la méthode décrite par *Fischer et Visbeck*⁷ pour déterminer la vitesse absolue. Cette méthode est indépendante de toute source extérieure de mesure (ADCP de coque par exemple). La détermination de la vitesse absolue nécessite de connaître les mouvements du L-ADCP par rapport au fond, or ceux-ci ne sont pas connus à chaque instant. *Fisher et Visbeck* résolvent ce problème de manière intégrale et il est seulement nécessaire de connaître la position du

⁷J.Fischer and M.Visbeck, Deep velocity profiling with self_contained ADCPs, *J. Atmos. Oceanic Technol*, 10(5), 764-773, 1993.

L-ADCP en début et fin de station.

A une profondeur Z donnée, la vitesse absolue est égale à :

$$U_{\text{absolue}}(z) = U_{\text{réf}} + U_{\text{baro}}(z) \quad (1)$$

$U_{\text{absolue}}(z)$: Vitesse absolue des courants
 $U_{\text{réf}}$: Vitesse de référence
 $U_{\text{baro}}(z)$: Vitesse barocline
 z : Profondeur

La vitesse barocline, fonction de la profondeur, U_{baro} , peut facilement être calculée à partir des gradients verticaux de vitesses horizontales.

Le calcul de la vitesse de référence $U_{\text{réf}}$ est détaillé ci-dessous.

Le L-ADCP mesure les vitesses sur une base temporelle :

$$U_{\text{mesuré}}(t) = U_{\text{absolue}}(t) - U_{\text{L-ADCP}}(t) \quad (2)$$

$U_{\text{mesuré}}(t)$: Vitesse brute mesurée par le L-ADCP
 $U_{\text{absolue}}(t)$: Vitesse absolue des courants
 $U_{\text{L-ADCP}}(t)$: Mouvement propre du L-ADCP
 t : Temps

La vitesse absolue peut se décomposer en :

$$U_{\text{absolue}}(t) = U_{\text{réf}} + U_{\text{baro}}(t) \quad (3)$$

$U_{\text{réf}}$: Vitesse de référence constante
 U_{baro} : Vitesse barocline obtenue par intégration du cisaillement de courant

Les mouvements du L-ADCP se décomposent en :

$$U_{\text{L-ADCP}}(t) = U_{\text{bateau}}(t) + U_{\text{bati}}(t) \quad (4)$$

U_{bateau} : Dérive du bateau pendant la station
 U_{bati} : Mouvement haute fréquence du bâti sur lequel est fixé le L-ADCP

Si l'on remplace les équations (3) et (4) dans (2) on obtient :

$$U_{\text{mesuré}}(t) = U_{\text{réf}} + U_{\text{baro}}(t) - U_{\text{bateau}}(t) - U_{\text{bati}}(t) \quad (5)$$

L'intégration sur la durée de la station (T) permet de calculer la vitesse de référence :

$$U_{\text{réf}} = \frac{1}{T} \left\{ \int U_{\text{mesuré}}(t) dt - \int U_{\text{baro}}(t) dt + \int U_{\text{bateau}}(t) dt \right\} \quad (6)$$

Car l'intégrale des mouvements hautes fréquences du bâti, U_{bati} , sur la durée de la station est

nulle et de plus :

$$\int \mathbf{U}_{\text{L-ADCP}} dt = \int \mathbf{U}_{\text{bateau}} dt = dX \quad (7)$$

où dX est la distance parcourue par le bateau entre le début et la fin de la station.

La vitesse de référence U_{ref} , et par conséquent la vitesse absolue (en utilisant l'équation (1)), peut être déterminée sans ambiguïté .

Le profil de vitesse barocline n'est pas calculé en fonction du temps mais en fonction de la profondeur $U_{\text{baro}}(z(t))$. La série temporelle de vitesse barocline peut être obtenue en utilisant :

$$z(t) = \int W(t)dt$$

où W est la vitesse verticale mesurée.

Cette méthode ne permet pas de distinguer le profil de descente et le profil de montée : le profil final combine les mesures effectuées lors de la descente et lors de la montée de l'appareil.

III - La chaîne de traitement.

III - 1 Présentation rapide de la chaîne de traitement

La chaîne de traitement est écrite en langage C. Les programmes ont été écrits sous environnement UNIX. Ils suivent en général la norme ANSI. Seul le programme LADCPBIN.C utilise des commandes UNIX pour concaténer plusieurs fichiers binaires. La visualisation des différentes étapes du traitement se fait grâce à des scripts Matlab dont la description est donnée au chapitre V.

La chaîne de traitement est composée de trois programmes principaux :

- *LADCPBIN.C* décode les fichiers binaires créés lors de l'acquisition des données. Il conserve les données nécessaires au traitement et convertit les fichiers binaires en un fichier ASCII qui sera utilisé lors du traitement.
- *LADCPUV.C* effectue le traitement proprement dit à partir du fichier ASCII créé par *LADCPBIN.C*.
- *LADCPCNF.C* lit le premier cycle de mesures d'un fichier brut afin de contrôler la configuration utilisée lors de l'acquisition. Ce programme n'intervient pas dans le traitement des données.

III - 2 Format des données brutes

Le L-ADCP enregistre les données dans un fichier binaire au format DOS. En général l'opérateur arrête l'acquisition des données en fin de station, un fichier comprend alors les enregistrements obtenus pendant la descente et la montée de l'appareil. Si l'opérateur a fixé une taille maximum pour les fichiers, un profil peut être découpé en plusieurs fichiers binaires si l'enregistrement dépasse cette limite. La chaîne de traitement a été conçue pour traiter un profil complet (i.e. tous les fichiers binaires L-ADCP associés au profil). Dans un fichier, l'unité d'enregistrement est le cycle de mesures (*'ensemble'*). Ceux-ci sont enregistrés séquentiellement suivant le format fixe suivant :

Trois enregistrements obligatoires

1 - Une entête	(<i>'HEADER DATA'</i>)
2 - Données Fixes	(<i>'FIXED LEADER DATA'</i>)
3 - Données Variables	(<i>'VARIABLE LEADER DATA'</i>)

Huit enregistrements optionnels choisis lors de la configuration de l'appareil

4 - Vitesse	(<i>'VELOCITY DATA'</i>)
5 - Amplitude de Corrélation	(<i>'CORREL MAGNITUDE'</i>)
6 - Intensité d'écho	(<i>'ECHO INTENSITY'</i>)
7 - Pourcentage de valeurs correctes	(<i>'PERCENT-GOOD DATA'</i>)
8 - Statut des rejets	(<i>'STATUS DATA'</i>)
9 - Recherche du fond	(<i>'BOTTOM TRACK DATA'</i>)

La nomenclature utilisée par le constructeur est indiquée en italique (entre parenthèse). Chaque enregistrement est décrit en détail dans la documentation RDI.

La configuration du L-ADCP permet de positionner de nombreux paramètres qui déterminent les conditions de mesures. Néanmoins, pour que la chaîne de traitement puisse fonctionner, il est impératif que les données soient enregistrées en coordonnées terrestres et que les informations de vitesse, d'intensité d'écho, d'amplitude de corrélation et de pourcentage de valeurs correctes soient acquises.

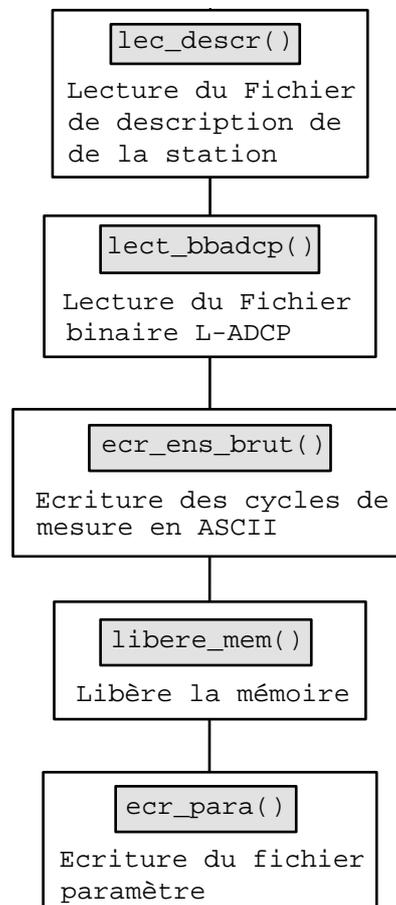
<u>Nomenclature RDI</u>	<u>Signification</u>
EX11xxx	Les mesures de vitesse sont transformées lors de l'acquisition en coordonnées terrestres (U, V, W).
EXxx1xx	Les informations des capteurs de roulis et tangage sont utilisées dans le calcul des vitesses.
WD11110000	Autorise l'acquisition de vitesse, d'intensité d'écho, d'amplitude de corrélation et de pourcentage de valeurs correctes

A titre d'exemple un fichier de configuration complet est donné en annexe D.

III - 3 Décodage des données brutes : **LADCPBIN.C**

III - 3 - a Organigramme du programme

Ce programme décode le fichier binaire d'acquisition et conserve les données utiles au traitement dans un fichier ASCII. L'organigramme simplifié du programme est le suivant :



III - 3 - b Lecture du fichier de description de la station : **lec_descr()**

Préalablement au traitement de chaque station l'opérateur doit créer un fichier de description en ASCII qui contient des informations utiles au bon déroulement du traitement. Ce fichier doit contenir :

- les noms des différents fichiers qui seront utilisés (en lecture ou en écriture).
- les positions GPS de début et fin de station.
- les heures de début et fin de station.
- la profondeur du L-ADCP au moment où les positions GPS ont été relevées.
- la déviation magnétique.
- les valeurs utilisées dans les tests du programme

Un fichier de description d'une station L-ADCP est reproduit ci-après. La fonction **lec_descr()** lit les informations qui suivent les lignes en caractères gras.

Nom du repertoire ou sont stockes les fichiers.

c:/ladcp/data/sta010/

Nom du fichier binaire ADCP (donnees brutes binaire) :

etal_010.000 etal_010.001 etal_010.002

Nom du fichier ascii ADCP (donnees brutes ascii) :

etal_010.asc

Nom du fichier des parametres :

etal_010.par

Nom du fichier ADCP (donnees traitees) :

etal_010.res

Position GPS en debut de station

Lat_deg Lat_min - Lon_deg Lon_min
 6 51.833 -50 -32.505

Date lors du point GPS de debut de station

yyyy mm jj hh mm ss
 1995 09 11 06 36 53

Profondeur en m de l'ADCP lors du point GPS de debut de station (flottant)

0.

Position GPS en fin de station

Lat_deg Lat_min - Lon_deg Lon_min
 6 51.885 -50 -32.405

Date lors du point GPS en fin de station

yyyy mm jj hh mm ss
 1995 09 11 09 47 50

Profondeur en m de l'ADCP lors du point GPS de fin de station (flottant)

0.

Deviation magnetique (format flottant) :

-18.50

TEST : Limite inferieure du PerCent Good (format flottant) :

30.

TEST : Limite superieure de tangage en degre (format flottant) :

18.

TEST : Limite superieure de roulis en degre (format flottant) :

15.

TEST : Limite superieure de l'erreur de vitesse en cm/s (format flottant) :

2.

TEST : Nb d'ecart-type pour le test de rejet vitesse (format flottant) :

2.

TEST : Nb de diff. moyenne pour le test de rejet vitesse (format flottant) :

4.

TEST : Nb d'ecart-type pour le test de rejet gradient (format flottant) :

2.

TEST : Nb de diff. moyenne pour le test de rejet gradient (format flottant) :

4.

TEST : Seuil pour la detection du fond (format flottant) :

15.

Detection des donnees du fond (oui=1 - non=0)

1

Methode de calcul de la vitesse absolue (1 ou 2)

1

Ces informations sont utilisées dans les fonctions suivantes :

Position GPS et dates de début et fin de station :	calc_deriv() - lec_ens_brut()
Limite inférieure du 'PerCent Good' :	nettoyage()
Limite supérieure de roulis et tangage :	nettoyage()
Limite supérieure de l'erreur de vitesse :	nettoyage()
Nombre d'écart-types pour le test de rejet vitesse :	test_et_vit()
Nombre de différences moyennes pour le test de rejet vitesse :	test_et_vit()
Nombre d'écart-types pour le test de rejet gradient :	test_et_grad()
Nombre de différences moyennes pour le test de rejet gradient :	test_et_grad()
Seuil pour la détection du fond :	rej_fond()
Méthode de calcul de la vitesse absolue :	vit_abs()

Remarques :

- Il est préférable de créer un répertoire par station afin de regrouper les différents fichiers créés lors du traitement. Le répertoire indiqué au début du fichier de description indique où se trouve les fichiers binaires L-ADCP et où seront écrits les différents fichiers créés lors du traitement.
- Il est préférable de respecter l'extension des noms de fichier car les programmes de tracé les utilisent pour se repérer.
- Les profondeurs et déviation magnétique doivent être écrites en format 'flottant'. La déviation magnétique doit être exprimée en 'degrés.centièmes'. **ATTENTION** : si les cartes de déclinaison magnétiques indiquent une correction ouest, il faut indiquer une déclinaison négative dans le fichier de description.
- Les latitudes et longitudes doivent être en degrés, minutes et centièmes. Les degrés sont écrits en format 'entier' et les minutes.centièmes en format 'flottant'. Les longitudes ouest et les latitudes sud sont comptées négativement. Si la position est négative, le signe '-' doit être placé devant les degrés et devant les minutes.centièmes.

- L'avant dernière ligne du fichier de description permet d'indiquer si le programme doit utiliser la fonction **rej_fond()** qui permet d'éliminer les données perturbées par les réflexions sur le fond. Si le profil ne s'est pas approché du fond, il est conseillé de ne pas faire cette détection (0).
- 2 méthodes peuvent être utilisées pour calculer la vitesse de référence qui sera ajoutée au profil de vitesse barocline pour obtenir le profil de vitesse absolue. Ces méthodes sont exposées au chapitre II-4-1.
- Il faut respecter les lignes blanches dans le fichier.

III - 3 - c Lecture des enregistrements binaires bruts : **lect_bbadcp()**

La fonction **lect_bbadcp()** permet de décoder un cycle de mesures (*ensemble*). A chaque enregistrement d'un cycle de mesure correspond une fonction de lecture. **lect_bbadcp()** appelle chacune de ces fonctions. Chaque enregistrement est rangé dans une structure de données définie dans le fichier *include lecadcp.h*. Les fonctions appelées par **lect_bbadcp()** sont listées ci-dessous (en caractères gras) :

Enregistrement d'un cycle de mesure

- 1 - Entête ('*HEADER DATA*')
- 2 - Données Fixes ('*FIXED LEADER DATA*')
- 3 - Données Variables ('*VARIABLE LEADER DATA*')
- 4 - Vitesse ('*VELOCITY DATA*')
- 5 - Amplitude de Corrélation ('*CORREL MAGNITUDE*')
- 6 - Intensité d'écho ('*ECHO INTENSITY*')
- 7 - Pourcentage de valeurs correctes ('*PERCENT-GOOD DATA*')
- 8 - Statut des rejets ('*STATUS DATA*')
- 9 - Recherche du fond ('*BOTTOM TRACK DATA*')

Fonctions

- lect_header()**
- lect_leadfix()**
- lect_leadvar()**
- lect_vit()**
- lect_cor()**
- lect_echo()**
- lect_pcg()**
- lect_status()**
- lect_bottom()**

Certaines de ces fonctions appellent les fonctions **dec_uns_short_int()** et **dec_short_int()** qui permettent de décoder du binaire.

Remarques :

- Bien qu'il existe une fonction de lecture de l'enregistrement '*BOTTOM TRACK DATA*' et de l'enregistrement '*STATUS DATA*' ces enregistrements ne sont pas utilisés dans le traitement.
- La fonction **lect_bottom()** n'a pas été testée.

III - 3 - d Ecriture des cycles de mesure en ASCII et libération de la mémoire : **ecr_ens_brut()** - **libere_mem()**

Lorsqu'un cycle de mesures a été décodé, il est immédiatement écrit dans un fichier ASCII par la fonction **ecr_ens_brut()**, et la mémoire allouée aux différentes structures de données est libérée par la fonction **libere_mem()**. La fonction **ecr_ens_brut()** effectue les calculs suivants pour chaque cellule :

- moyenne des intensités d'écho ('*ECHO INTENSITY*') des 4 faisceaux du L-ADCP.
- moyenne des amplitudes de corrélation ('*CORREL MAGNITUDE*') des 4 faisceaux du L-ADCP.
- convertit les vitesses en cm/s.
- remplace les vitesses erronées par la valeur **DUMVEL** (déclarée dans le fichier *include constant.h* - **DUMVEL** = -999.90)

La fonction `ecr_ens_brut()` ne conserve que les informations utiles au traitement. Chaque cycle de mesures comporte une entête, puis les mesures associées à chaque cellule ('*depth cells*' ou '*bins*'). Il y a un enregistrement par cellule. Les cycles de mesures sont écrits séquentiellement. Un cycle de mesures est reproduit ci-après :

```
000371 019 94/11/06 19:26:38 05 3.470 173.81 0.83 4.02 1465.00 0.00
-0.90 4.70 87.90 -0.20 62.250 117.750 100.00
-2.50 4.80 87.70 -0.10 65.000 119.250 100.00
-5.90 2.80 87.20 -0.10 55.250 112.750 100.00
-8.10 2.90 87.30 0.30 50.500 108.500 100.00
-8.70 6.10 87.70 -1.40 43.750 90.500 100.00
-9.00 7.70 87.30 0.20 41.000 76.500 100.00
-7.50 8.50 87.60 0.00 40.250 66.750 100.00
-8.80 7.90 87.70 -0.50 39.750 62.250 100.00
-9.70 8.60 87.50 -0.20 44.500 57.750 83.00
-11.40 7.60 95.60 -999.90 40.500 45.500 0.00
-15.80 8.40 53.80 -999.90 37.000 31.000 0.00
-10.60 19.30 55.40 -0.80 37.500 31.250 16.00
-14.30 20.60 52.80 -999.90 37.500 33.250 0.00
-999.90 -999.90 -999.90 -999.90 41.750 27.750 0.00
-999.90 -999.90 -999.90 -999.90 37.250 20.750 0.00
.....
```

L'entête comprend les informations suivantes (entre parenthèses sont indiqués les noms des variables utilisées dans la fonction et leur valeur dans l'exemple ci-dessus) :

- Le numéro du cycle de mesures (*num_ens* : 000371)
- Le nombre de cellules (*nb_cell* : 019)
- L'année de l'acquisition (*an* : 94)
- Le mois de l'acquisition (*mois* : 11)
- Le jour de l'acquisition (*jour* : 06)
- L'heure de l'acquisition (*heure* : 19)
- Les minutes de l'acquisition (*min* : 26)
- Les secondes de l'acquisition (*sec* : 38)
- Les centièmes de seconde de l'acquisition (*cent* : 05)
- La température du capteur interne en °C (*temp* : 3.470)
- Le cap mesuré par le capteur interne (*heading* : 173.81)
- Le tangage mesuré par le capteur interne (*pitch* : 0.83)
- Le roulis mesuré par le capteur interne (*roll* : 4.02)
- La vitesse du son utilisée (*vit_son* : 1465.00)
- La vitesse verticale du bâti au moment de l'acquisition du cycle de mesures (*w_moy* : 0.00)

Attention

La vitesse verticale (*w_moy*) du bâti supportant le L-ADCP n'est pas un paramètre mesuré. Cette variable est calculée lors du traitement des données.

L'entête est suivie d'enregistrements qui décrivent les variables de chaque cellule. Dans le cas présenté ci-dessus il y a 19 cellules (*nb_cell* = 19) et donc 19 enregistrements suivent l'entête. Ces enregistrements sont composés de (entre parenthèses sont indiquées les valeurs de la 1^{ère} cellule de l'exemple précédent) :

- La composante zonale de la vitesse en cm/s (-0.90)
- La composante méridienne de la vitesse en cm/s (4.70)
- La composante verticale de la vitesse en cm/s (87.90)

- L'erreur de vitesse en cm/s (-0.20)
- La moyenne des intensités d'écho des 4 faisceaux (62.250)
- La moyenne des amplitudes de corrélation des 4 faisceaux (117.750)
- La valeur du pourcentage de données correctes extraites du 4^{ème} faisceau (100.00)

Remarque :

- Le programme ne conserve que les données de '*PERCENT-GOOD*' qui indique le pourcentage de bonnes mesures réalisées avec 4 faisceaux.
- Le programme n'utilise pas les données de l'enregistrement '*STATUS DATA*'.

III - 3 - e Ecriture du fichier paramètre : **ecr_para()**

La fonction **ecr_para()** permet de sauvegarder dans un fichier ASCII des informations sur la configuration de l'acquisition qui seront utilisées par le programme de traitement **ladcpuv.c**. La fonction crée un fichier tel que celui qui est reproduit ci-dessous :

```
Numero du dernier ensemble:
8586
Nombre d'ensembles:
8586
Nombre de cellules par ensemble:
19
Taille d'une cellule:
16.00
Profondeur 1ere cellule:
33.33
Salinité:
35.000
```

Le nombre de cellules par ensemble, la taille d'une cellule (en m), la profondeur de la 1^{ère} cellule (en m) et la salinité sont lus dans le 1^{er} cycle de mesure. La taille de la cellule, fonction de la vitesse du son, est celle calculée par l'appareil en supposant une vitesse du son de 1500 m/s. La vitesse du son variant entre la surface et le fond de l'océan, la taille de la cellule est calculée lors du traitement pour chaque cycle de mesures en fonction de la vitesse du son. Le numéro d'*ensemble* est un numéro attribué par le L-ADCP lors de l'acquisition, le nombre d'*ensembles* est le nombre de cycles de mesures effectivement lu. Le nombre d'*ensembles* (cycles de mesures) et le numéro du dernier *ensemble* sont connus à la fin de la lecture de tous les cycles de mesures. Lorsque ces deux nombres diffèrent c'est qu'il y a un 'trou' dans les signaux récupérés par le L-ADCP. Il convient alors d'être prudent dans le traitement.

Remarques :

- L'information sur le nombre de cellules par cycle de mesures (*'ensemble'*) est aussi conservée dans chaque entête du fichier ASCII (variable *nb_cell*). Cette information qui ne varie pas d'un cycle de mesures à l'autre est redondante et pourrait être éliminée dans les entêtes des cycles de mesures du fichier ASCII.
- La salinité n'est pas utilisée dans la suite du traitement. Si un capteur de salinité était ajouté au L-ADCP, il serait préférable que l'information de salinité soit conservée dans les entêtes des cycles de mesures du fichier ASCII.

III - 4 Calcul du profil de vitesses horizontales : **LADCPUV.C**

III - 4 - a Organigramme du programme

L'organigramme simplifié du programme **ladcpuv.c** est présenté à la page 17.

III - 4 - b Généralités

Pour fonctionner le programme **ladcpuv.c** utilise les fichiers suivants (les fonctions de lecture de ces fichiers sont écrites en caractères gras) :

- le fichier de description de la station (voir chapitre III - 3 - b) **lec_descr()**
- le fichier des paramètres du profil (voir chapitre III - 3 - d) **lec_para()**
- le fichier ASCII des données brutes (voir chapitre III - 3 - e) **lec_ens_brut()**

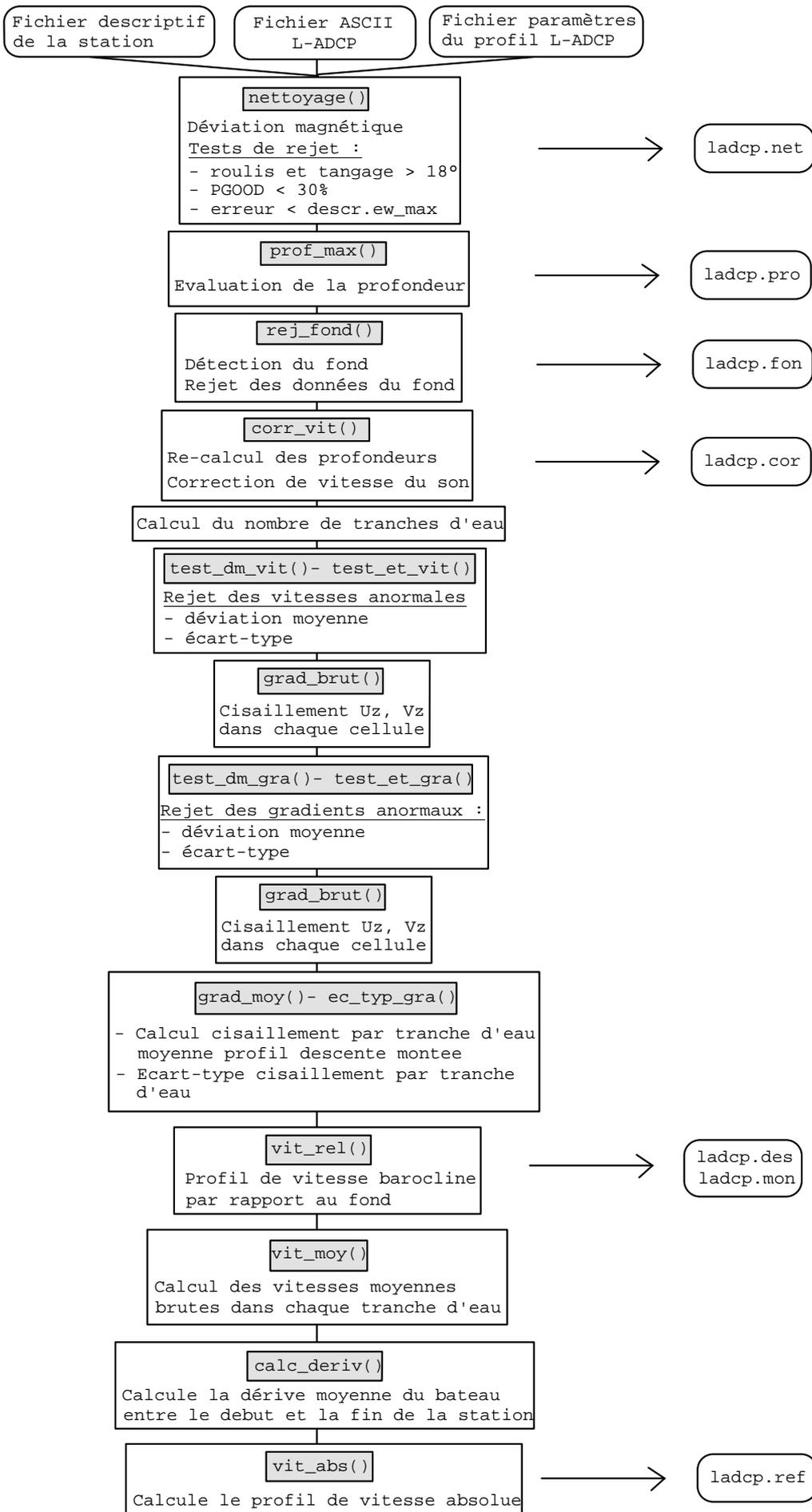
Les données lues par ces fonctions sont rangées dans des structures de données déclarées dans le fichier *'include'* **struct.h**. Ces structures de données sont détaillées ci-dessous.

Fonction **lec_descr()** ---> Structure **DESCRIPTIF**

```
typedef struct descriptif {
    char repertoire[NB_CAR_MAX];      Répertoire de travail
    POS  p_deb;                       Position du L-ADCP en début de station
    POS  p_fin;                       Position du L-ADCP en fin de station
    DATE d_deb;                       Date et heure de début de station
    DATE d_fin;                       Date et heure de fin de station
    float prof_deb;                   Profondeur du L-ADCP en début de station
    float prof_fin;                   Profondeur du L-ADCP en fin de station
    float dev_magn;                   Déviation magnétique
    float pgood_min;                 Valeur minimum du percent Good Data
    float pitch_max;                 Valeur maximum du tangage
    float roll_max;                  Valeur maximum du roulis
    float ew_max;                    Valeur maximum de l'erreur de vitesse
    float seuil_et_vit;               Coefficient utilisé dans le test d'écart-type vitesse
    float seuil_dm_vit;               Coefficient utilisé dans le test d'écart à la moyenne vitesse
    float seuil_et_gra;               Coefficient utilisé dans le test d'écart-type gradient
    float seuil_dm_gra;               Coefficient utilisé dans le test d'écart à la moyenne gradient
    float ddb;                        Seuil utilisé dans la détection du fond
    int  detect_fond;                 Code pour déterminer si le fond doit être détecté
    int  methode;                     Méthode de calcul de la vitesse absolue
} DESCRIPTIF;
```

Commentaires :

- Les dates sont rangées dans une structure de données **DATE** (fichier *'include'* **struct.h**).
- Les positions sont rangées dans une structure de données **POS** (fichier *'include'* **struct.h**).



Fonction `lec_para()` ---> Structure **PARA**

```
typedef struct parametre {
    unsigned int    num_der_ens;    Numéro du dernier cycle de mesures lu
    unsigned int    nb_ens;        Nombre de cycle de mesures lus
    int             nb_cell;       Nombre de cellules par cycle de mesures
    float           taille_cell;   Taille verticale d'une cellule
    float           prof_cell1;    Profondeur de la 1ère cellule par rapport aux bases acoustiques
    float           sal;          Salinité
} PARA;
```

Fonction `lec_ens_brut()` ---> Structures **ENTETE** et **CELL**

```
typedef struct entete {
    int    num_ens;        Numéro du cycle de mesures
    float temp;          Température mesurée par le L-ADCP
    float sec;           Temps en seconde mesuré par le L-ADCP
    float heading;      Cap mesuré par le L-ADCP
    float pitch;        Tangage mesuré par le L-ADCP
    float roll;         Roulis mesuré par le L-ADCP
    float vit_son;      Vitesse du son mesurée par le L-ADCP
    float w_moy;        Vitesse verticale moyenne calculée par le logiciel
    float tail_cel_cor; Taille d'une cellule corrigée après correction de vitesse du son
} ENTETE;

typedef struct cell {
    float prof;          Profondeur de la cellule (en m) calculée par le logiciel
    float ru;           Composante zonale de la vitesse (cm/s) mesurée par le L-ADCP
    float rv;           Composante méridienne de la vitesse (cm/s) mesurée par le L-ADCP
    float rw;           Composante verticale de la vitesse (cm/s) mesurée par le L-ADCP
    float error;        Erreur de vitesse (cm/s) calculée par le L-ADCP
    float amp_echo;     Intensité d'écho mesurée par le L-ADCP
    float correl;       Amplitude de corrélation mesurée par le L-ADCP
    float pcg;          Percent Good mesuré par le L-ADCP
} CELL;
```

Commentaires :

- Le programme de lecture rassemble toutes les entêtes des cycles de mesures dans un tableau de structure de données **ENTETE** et toutes les informations des cellules dans un tableau de structure de données **CELL**.
- La taille du tableau d'entêtes est égale au nombre de cycles de mesures lus, soit **nb_ens** (structure **PARA**).
- La taille du tableau des cellules est égale au nombre de cycles de mesures lus multiplié par le nombre de cellules d'un cycle de mesures, soit **nb_ens*nb_cell** (structure **PARA**). Il faut donc faire un calcul pour repérer la $j^{\text{ème}}$ cellule du $i^{\text{ème}}$ cycle de mesures. L'exemple ci-dessous détaille le calcul de l'indice de la 1^{ère} ($j1$) et dernière ($j2 - 1$) cellule du $i^{\text{ème}}$ cycle de mesures (en langage C le premier indice d'un tableau est 0) :

$$j1 = i * nb_cell \quad \text{indice de la 1ère cellule}$$

$$j2 = j1 + nb_cell \quad \text{indice de la dernière cellule} + 1$$
- La fonction `lec_ens_brut()` teste les heures de début et fin de station qui se trouvent dans le fichier de description de la station et ne conserve que les cycles de mesures qui se trouvent dans cette plage horaire.

Le programme `ladcpuv.c` crée un certain nombre de fichiers qui permettent de suivre les différentes étapes du traitement. Les fichiers, décrits au chapitre IV, sont utilisés par des scripts

MATLAB pour visualiser les résultats intermédiaires (voir chapitre V).

III - 4 - c Nettoyage des données brutes : **nettoyage()**

La fonction **nettoyage()** fait une première validation des données brutes. Elle permet :

- d'éliminer les mesures pour lesquelles le pourcentage de données correctes ('*PERCENT GOOD*', `cell[j].pcg`) est inférieur à **descr.pgood_min** (constante lue dans le fichier de description de la station (valeur utilisée : 30%)).
- d'éliminer les mesures pour lesquelles l'erreur de vitesse (`cell[i].error`) est supérieure à **descr.ew_max** (constante lue dans le fichier de description de la station). **ATTENTION** : la qualité des résultats est sensible à la valeur de ce paramètre. L'erreur de vitesse peut-être visualisée par le programme Matlab **don_brut.m** (cf. chapitre V), l'opérateur peut alors juger de la valeur à attribuer à ce paramètre. La qualité du traitement peut être contrôlée en visualisant les profils de vitesse de montée et de descente par le programme Matlab **descmont.m** (cf. chapitre V)
- d'éliminer les mesures pour lesquelles les angles de tangage et de roulis (`entete[i].pitch` et `entete[i].roll`) sont supérieurs à **descr_pitch_max** et **descr_roll_max** (constantes lues dans le fichier de description de la station).
- de corriger les mesures de vitesses horizontales de la déclinaison magnétique. Une déclinaison ouest est négative.

Lorsque des mesures ne satisfont pas aux tests, les composantes horizontales et verticale de la vitesse (`cell[j].ru`, `cell[j].rv`, `cell[j].rw`) sont remplacées par **DUMVEL** (constante fixée à -999.90 déclarée dans le fichier '*include*' **constant.h**)

III - 4 - d Première estimation de la profondeur : **prof_max() - ind_pmax() - w_moyen()**

Afin d'éliminer les mesures perturbées par la réflexion sur le fond (cf. chapitre III-4-e), il est nécessaire de connaître, même approximativement, la profondeur du fond. La fonction **prof_max()** permet de calculer la profondeur de chaque cellule et la fonction **ind_pmax()** détermine ensuite l'indice de l'ensemble le plus profond.

Dans la fonction **prof_max()** la profondeur de chaque cycle de mesures, au niveau des bases acoustiques du L-ADCP, est estimée en intégrant verticalement (du début du profil à la fin du profil) la vitesse verticale mesurée par le L-ADCP. L'algorithme utilisé est le suivant :

$$P_i = P_{i-1} + W_{i-1} * (T_i - T_{i-1})$$

- Le temps, en seconde, de l'acquisition de chaque cycle de mesures est conservé dans le tableau de structure de données **entete** (`entete[i].sec`).
- La vitesse verticale d'un cycle de mesures est calculée par la fonction **w_moyen()**. Cette fonction fait la moyenne des vitesses verticales mesurées dans les cellules d'un cycle de mesures. Les vitesses verticales des cellules qui s'écartent de plus de 2 fois l'écart-type de la moyenne sont éliminées et la moyenne est calculée à nouveau. Ce processus est répété 2 fois. Si la vitesse verticale d'un cycle de mesure n'a pu être calculée, le logiciel utilise la vitesse verticale

du cycle de mesures précédent pour faire l'intégration.

L'estimation des vitesses à partir de l'effet Doppler suppose que la vitesse du son soit connue. Dans l'eau de mer la vitesse du son dépend de la température, de la salinité et de la pression. Or le L-ADCP, en l'absence de capteur de pression et de salinité, ne peut estimer la vitesse du son et par conséquent calcule des vitesses de courant erronées. Ces vitesses peuvent être corrigées a posteriori lorsque la vitesse du son est connue. La formule de calcul est donnée dans le Manuel Technique RDI :

$$V_{\text{corrigé}} = V_{\text{mesuré}} * C_{(t,s,p)} / C_{\text{ref}}$$

C_{ref} : Vitesse du son utilisée par le L-ADCP pour calculer les vitesses.

$C(t,s,p)$: Vitesse du son réelle, fonction de la température, la salinité, et la pression.

Pour corriger les composantes horizontales de la vitesse (u, v), il suffit de connaître la vitesse du son au niveau des bases acoustiques. Par contre la correction de la composante verticale (w) impose de connaître la vitesse du son au niveau de chaque cellule de mesure. Pour l'instant la vitesse verticale (w) est corrigée en utilisant uniquement la vitesse du son au niveau des bases acoustiques.

La fonction **w_moyen()** nécessite la connaissance de la profondeur du cycle de mesures afin de calculer la vitesse du son et de corriger les mesures de vitesses verticales faites par le L-ADCP. Lorsque la profondeur est connue, celle-ci est convertie en pression par la fonction **press80()**, la vitesse du son est calculée, au niveau des bases acoustiques, par la fonction **svel()** en utilisant la température du capteur du L-ADCP et une salinité constante de 35.

La profondeur de chaque cellule est calculée et rangée dans la structure de données **CELL** (`cell[j].prof`). La vitesse verticale utilisée dans l'intégration est rangée dans la structure **ENTETE** (`entete[i].w_moy = w_moyen(vit_verticale, para.nb_cell)`).

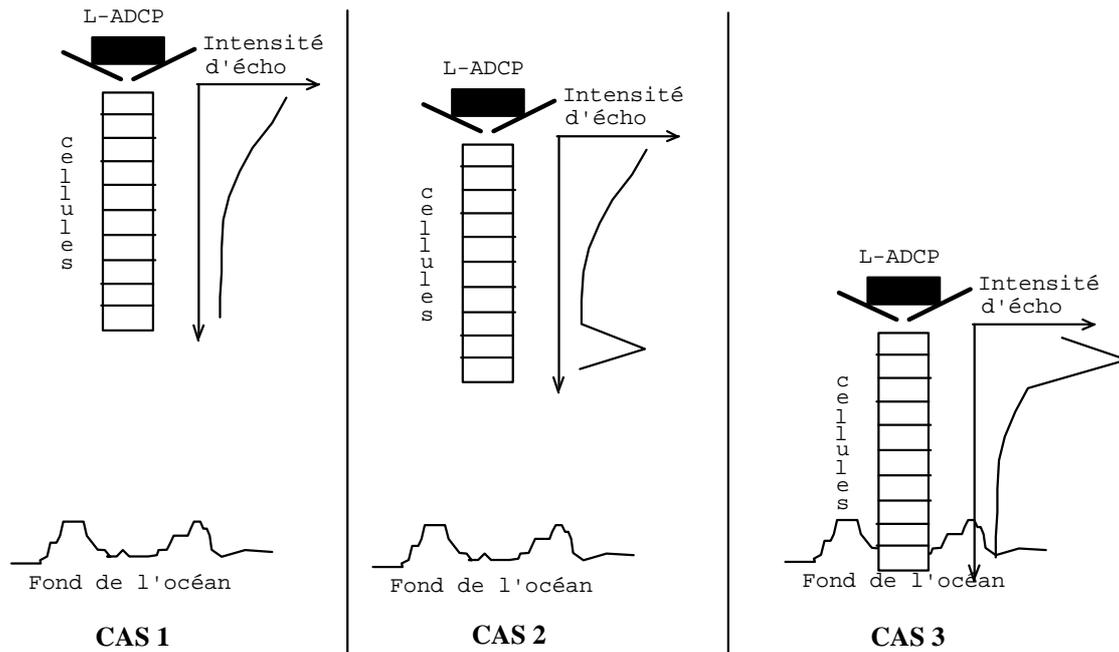
Remarques :

- La procédure d'intégration de la vitesse verticale utilise les vitesses verticales de tous les cycles de mesures, y compris ceux dont la mesure est perturbée par les réflexions sur le fond. L'estimation de la profondeur peut en être affectée.
- La vitesse du son est estimée en utilisant une salinité constante de 35.
- La vitesse verticale est corrigée uniquement en calculant la vitesse du son au niveau des bases acoustiques, or pour corriger la vitesse verticale il faudrait calculer la vitesse du son au niveau de chaque cellule d'un même cycle de mesures. Cette approximation n'a pas été testée.
- Il serait souhaitable d'utiliser les mesures de la sonde pour estimer la profondeur de chaque cycle de mesures et calculer le profil de vitesse du son.

III - 4 - e Rejet des données perturbées par les réflexions sur le fond de l'océan : **rej_fond()**

Lorsque le L-ADCP s'approche du fond de l'océan, les ondes sonores se réfléchissent sur les particules en mouvement et sur le fond de l'océan. Ces réflexions sur le fond perturbent les mesures de vitesse qu'il faut alors éliminer (*cf. Fischer et Visbeck*). Lorsque les réflexions parasites dues au fond se produisent, le profil d'intensité d'écho (*ECHO INTENSITY*) présente un maximum. La fonction **rej_fond()** détecte ce maximum et élimine les cellules perturbées, ainsi que les cellules

situées en-dessous. L'algorithme de détection peut être schématisé ainsi :



1 - Lorsque le L-ADCP est suffisamment éloigné du fond le profil d'intensité d'écho ne présente pas de maximum particulier ---> **CAS 1**

2 - Quand le L-ADCP s'approche du fond, ce sont les cellules les plus proches du fond qui sont perturbées par les réflexions parasites. Un maximum d'intensité d'écho est visible dans les dernières cellules ---> **CAS 2**

3 - Au fur et à mesure de la descente, le maximum d'intensité se déplace vers les cellules proches des bases acoustiques ---> **CAS 3**

4 - Si le L-ADCP continue à descendre, toutes les cellules sont polluées par les réflexions parasites et le profil d'intensité d'écho ne présente plus de maximum.

5 - Quand le L-ADCP commence à remonter ce sont les cellules les plus proches des bases acoustiques qui présentent un maximum dans le profil d'intensité d'écho. On se retrouve dans le **CAS 3**

6 - Lors de la montée du L-ADCP, le maximum d'intensité d'écho se déplace vers les cellules éloignées des bases acoustiques et il y a un moment où le L-ADCP va se retrouver dans le **CAS 2** : ce sont les dernières cellules qui seront perturbées et enfin le profil d'intensité d'écho redeviendra correct.

La fonction utilise la profondeur du fond estimée auparavant dans **prof_max()**. La détection des cycles de mesures perturbées par les réflexions parasites commence à une distance du fond égale à 2 fois la pénétration verticale d'un cycle de mesures (le nombre de cellules d'un cycle multiplié par la taille d'une cellule) (se reporter à *Fischer et Visbeck* pour le calcul de la profondeur où se produit les premières réflexions parasites). Ensuite le maximum d'intensité d'écho est détecté en comparant la différence d'intensité d'écho entre 2 cellules voisines à une valeur seuil **descr.ddb** (constante lue dans le fichier de description de la station (valeur conseillée : 15). Le logiciel fait un histogramme des profondeurs où se produit le maximum d'intensité d'écho. Le maximum de cet histogramme indique où se trouve le fond. Une fois le fond déterminé avec précision, les cellules dont la profondeur estimée est supérieure au fond sont mises à la valeur **DUMVEL** (déclarée dans le fichier *include constant.h*). Si toutes les cellules d'un cycle de mesure sont mises à **DUMVEL**, cela signifie que toutes les mesures ont été faites virtuellement sous le fond et la vitesse verticale de la structure ENTETE (entete[i].w_moyen) est aussi mise à **DUMVEL**. C'est cette variable qui sera testée (entete[i].w_moyen) ultérieurement par **corr_vit()** pour estimer si il n'y aucune mesure valide dans un cycle de mesures à cause des réflexions sur le fond (dans ce cas la vitesse sera considérée

nulle) ou à cause d'autres raisons (dans ce cas **corr_vit()** utilisera la vitesse verticale du cycle de mesures précédent).

Remarque :

Cet algorithme a été conçu uniquement pour détecter la réflexion principale sur le fond. Néanmoins *Fischer et Visbeck* indiquent qu'il peut aussi exister des réflexions secondaires dont l'influence ne paraît pas être importante. Aucun test n'a été prévu pour détecter ces réflexions et pour corriger leur influence éventuelle.

III - 4 - f Calcul de la profondeur d'un cycle de mesures et correction des vitesses : **corr_vit()**

Lorsque les cellules perturbées par le fond ont été éliminées, il est alors possible d'estimer plus précisément la profondeur de chaque cycle de mesures et de corriger les vitesses mesurées par le L-ADCP des effets de la variation de la vitesse du son. Cette fonction est très proche de la fonction **prof_max()** et en diffère simplement par la correction de vitesse du son apportée aux vitesses. La vitesse du son calculée est conservée dans la structure de données **ENTETE**.

Pour un cycle de mesures l'épaisseur des cellules dépend de la vitesse du son au niveau des bases acoustiques. Par défaut l'épaisseur des cellules est calculée en supposant une vitesse du son constante de 1500 m/s. La fonction **corr_vit()** calcule l'épaisseur réelle des cellules en tenant compte de la vitesse du son réelle (formule extraite du manuel RDI - annexe F-5) :

```
entete[i].tail_cel_cor = para.taille_cell * vit_son / 1500
```

La profondeur de chaque cellule est calculée et conservée dans le tableau `cell[j].prof`.

La fonction **corr_vit()** considère que les cycles de mesures entièrement perturbés par le fond ont des vitesses verticales nulles. Il s'agit des cycles de mesures où la variable `entete[i].w_moyen` a été remplacée par **DUMVEL** dans la fonction **rej_fond()**, lorsque toutes les cellules étaient perturbées par les réflexions sur le fond. Dans ce cas le L-ADCP ne faisant plus de mesures correctes cela revient à supposer qu'il est immobile.

Lorsque l'intégration est terminée la profondeur estimée doit être proche de la profondeur du L-ADCP en fin de station. La profondeur en fin de station est fournie dans le fichier de description de la station. La différence entre la profondeur calculée et la profondeur réelle donne une indication de la qualité de l'intégration (cette différence est inscrite à l'écran lors du calcul). A la fin de la fonction **corr_vit()**, cette différence est répartie linéairement sur toutes les estimations de profondeur, excepté pour les cycles de mesures perturbés par le fond puisque ceux-ci n'ont pas été utilisés dans l'intégration.

III - 4 - g Résolution verticale du profil de vitesse

A ce stade du traitement commence l'évaluation du profil de vitesse absolue entre la surface et le fond. Il faut projeter les mesures acquises avec une base temporelle, sur un profil vertical. La résolution verticale de ce profil est déterminée en 'découpant' verticalement l'océan (entre la surface et le fond de l'océan) en tranches d'eau dont la hauteur est identique à la hauteur d'une cellule de mesure (calculée avec une vitesse du son de 1500 m/s). Ce calcul est fait dans le programme principal **ladcpbin.c** :

```
nbre_tranche = (int) (cell[j1].prof / para.taille_cell + NB_TR_MIN
```

nbre_tranche est le nombre de tranches d'eau du profil final.
 cell[j1].prof est la profondeur (en m) de la première cellule du cycle de mesures le plus profond.
 para.taille_cell est la hauteur d'une cellule.
 NB_TR_MIN est le nombre minimum de tranches d'eau (défini dans le fichier *include constant.h*).

Remarque :

Le programme impose la résolution verticale du profil de vitesse final. Elle est égale à la taille d'une cellule ('bin'). Il pourrait être envisagé de laisser libre le choix de la résolution verticale.

III - 4 - h Test sur les vitesses horizontales :

test_dm_vit() - test_et_vit()

Cette étape permet de contrôler la qualité des mesures de vitesses horizontales (u et v) de chaque cellule. Les vitesses horizontales de chaque cellule sont réparties dans les tranches d'eau en fonction de leur profondeur. Ensuite des tests de déviation moyenne et d'écart-type sont appliqués à chaque tranche d'eau afin d'éliminer les mesures de vitesse aberrantes.

Le test de déviation moyenne est :

$$\left| U_j^{\text{tr}} - \frac{1}{n} \sum_j U_j^{\text{tr}} \right| < \text{descr.seuil_dm} * \frac{1}{n} \sum_j \left(U_j^{\text{tr}} - \frac{1}{n} \sum_j U_j^{\text{tr}} \right)$$

tr Tranche d'eau n° 'tr'
 j ^{j^{ème}} cellule
 n Nombre de mesures dans la tranche d'eau 'tr'
 U Vitesse horizontale
 descr.seuil_dm Constante lue dans le fichier de description de la station (valeur conseillée : 4)

Le test d'écart-type est :

$$\left| U_j^{\text{tr}} - \frac{1}{n} \sum_j U_j^{\text{tr}} \right| < \text{descr.seuil_et} * \sigma_{u_j^{\text{tr}}}$$

tr Tranche d'eau n° 'tr'
 j ^{j^{ème}} cellule
 n Nombre de mesures dans la tranche d'eau 'tr'

U	Vitesse horizontale
σ	Ecart-type des vitesses dans la tranche d'eau 'tr'
descr.seuil_et	Constante lue dans le fichier de description de la station (valeur conseillée : 2)

Les mesures de vitesse qui ne satisfont pas les tests ci-dessus sont éliminées (leur valeur est remplacée par **DUMVEL**, définie dans **constant.h**).

Ces tests sont effectués séparément pour les mesures du profil de montée et le profil de descente. Le premier argument de la fonction indique si les tests sont appliqués au profil de descente (-1) ou au profil de montée (1).

Remarque :

- Si une cellule se trouve à cheval sur deux tranches d'eau, les valeurs de vitesse de cette cellule sont placées dans la tranche d'eau dont la profondeur est la plus proche. Lors du calcul final du profil de vitesse, les mesures de ces cellules seront réparties proportionnellement dans les tranches d'eau qu'elles chevauchent.

III - 4 - i Calcul des gradients verticaux de vitesses horizontales pour chaque cellule : **alloc_struct() - grad_brut()**

La fonction **grad_brut()** calcule les gradients verticaux de vitesses horizontales nécessaires pour l'estimation du profil vertical de vitesses absolues (voir chapitre II - 5 et III - 4 - 1). Les gradients verticaux sont calculés par différence finie au premier ordre. Pour estimer le gradient au niveau de la cellule j, on utilise les informations de vitesse des cellules j et l :

$$\text{grad}[j].\text{dru} = (\text{cell}[l].\text{ru} - \text{cell}[j].\text{ru}) / (\text{cell}[l].\text{prof} - \text{cell}[j].\text{prof})$$

l est l'indice de la première cellule, sous la cellule j, dont les valeurs de u et v permettent de calculer le gradient. l peut varier de j+1 au nombre de cellules maximum d'un cycle de mesures.

La profondeur de chaque gradient est conservée dans la structure de données **GRAD** :

$$\text{grad}[j].\text{prof} = (\text{cell}[j].\text{prof} + \text{cell}[l].\text{prof}) * 0.5$$

La fonction **alloc_struct()** permet d'allouer la mémoire pour la structure de données **GRAD** dans laquelle sont rangés les gradients verticaux de vitesse de chaque cellule. Cette structure est déclarée dans le fichier *include struct.h* :

```
typedef struct grad {
    float dru;           Gradient vertical de vitesse zonale au niveau de la cellule j
    float drv;           Gradient vertical de vitesse méridienne au niveau de la cellule j
    float prof;          Profondeur en m du gradient
    long int cell2;      Indice l de la seconde cellule utilisée pour calculer le gradient
} GRAD;
```

Remarque :

- Les gradients verticaux sont calculés par différence finie au premier degré. Aucun autre schéma de dérivation n'a été testé (différence centrée, etc.).
- Le programme **grad_brut()** range les gradients verticaux de vitesses dans un tableau de

structure de données **GRAD**. La taille de ce tableau est égale au nombre de cycles de mesures lus multiplié par le nombre de cellules d'un cycle de mesures, soit **nb_ens** * **nb_cell** (structure **PARA**).

III - 4 - j Test sur les gradients verticaux de vitesses horizontales **test_dm_gra() - test_et_grad()**

Cette étape permet de contrôler la qualité des gradients verticaux de vitesses horizontales calculés par la fonction **grad_brut()**. Les gradients verticaux de chaque cellule sont répartis dans les tranches d'eau en fonction de leur profondeur. Ensuite des tests de déviation moyenne et d'écart-type sont appliqués à chaque tranche d'eau afin d'éliminer les gradients aberrants. Les valeurs des vitesses ayant servi à calculer ces gradients aberrants sont aussi éliminées au niveau des cellules (valeurs mises à **DUMVEL**). Ces vitesses ne seront donc pas utilisées pour le calcul du profil final. L'algorithme et les tests sont en tout point identiques à ceux utilisés dans les fonctions **test_dm_vit()** et **test_et_vit()**, détaillés au chapitre III - 4 - h.

Remarque :

- Si une cellule se trouve à cheval sur deux tranches d'eau, les valeurs de gradient de cette cellule sont placées dans la tranche d'eau dont la profondeur est la plus proche. Lors du calcul final du profil de vitesse, les mesures de ces cellules seront réparties proportionnellement dans les tranches d'eau qu'elles chevauchent.

III - 4 - k Calcul des gradients verticaux de vitesses horizontales dans chaque tranche d'eau : **grad_moy() - ec_typ_gra()**

Une fois que l'ensemble des tests est terminé, la fonction **grad_moy()** calcule la moyenne des gradients verticaux de vitesses horizontales en répartissant les estimations de chaque cellule dans chaque tranche d'eau. Les gradients des cellules qui se trouvent à cheval sur des tranches d'eau sont répartis proportionnellement dans chacune des tranches d'eau qu'elles chevauchent. Les profils de descente et de montée ne sont pas séparés. Les moyennes sont rangées dans une structure de données de type **RESULT** (**result[i].u_grad**, **result[i].v_grad**) qui est déclarée dans le fichier *include struct.h* (la place mémoire pour cette structure est allouée dans la fonction **alloc_struct()**). Cette structure de données permet de conserver les résultats des calculs effectués dans chaque tranche d'eau

```
typedef struct result{
    float u_abs;      Vitesse zonale absolue dans une tranche d'eau
    float v_abs;      Vitesse méridienne absolue dans une tranche d'eau
    float u_baro;     Vitesse zonale par rapport au fond dans une tranche d'eau
    float v_baro;     Vitesse méridienne par rapport au fond dans une tranche d'eau
    float u_mes;      Vitesse zonale mesurée dans une tranche d'eau
    float v_mes;      Vitesse méridienne mesurée dans une tranche d'eau
    float u_grad;     Gradient vertical de vitesse zonale mesurée dans une tranche d'eau
    float v_grad;     Gradient vertical de vitesse méridienne mesurée dans une tranche d'eau
    float n_grad;     Nombre de gradients par tranche d'eau
    float et_u_grad;  Ecart-type du gradient vertical de vitesse zonale dans une tranche d'eau
    float et_v_grad;  Ecart-type du gradient vertical de vitesse méridienne dans une tranche d'eau
} RESULT;
```

La fonction **ec_typ_gra()** calcule l'écart-type des gradients verticaux dans une tranche d'eau. Les résultats sont rangés dans la structure de données **RESULT** (result[i].et_u_grad, result[i].et_v_grad).

Remarque :

- Les gradients verticaux sont calculés par différence finie au premier degré. Aucun autre schéma de dérivation n'a été testé (différence centrée, etc.)
- Le calcul d'écart-type n'est pas tout à fait rigoureux car il utilise la moyenne calculée par **grad_moy()**.

III - 4 - 1 Calcul de la vitesse absolue

vit_rel() - calc_deriv() - vit_abs() - ecr_res() - vit_moy()

Le calcul du profil final de vitesse absolue a été détaillé au chapitre II - 5. Pour une tranche d'eau la vitesse absolue est :

$$U_{\text{absolue}}(z) = U_{\text{réf}} + U_{\text{baro}}(z)$$

$U_{\text{réf}}$ est calculée par l'équation suivante :

$$U_{\text{réf}} = \frac{1}{T} \left\{ \int U_{\text{mesuré}}(t) dt - \int U_{\text{baro}}(t) dt + dX \right\}$$

La vitesse barocline, U_{baro} , est calculée par la fonction **vit_rel()**. Cette fonction intègre verticalement, depuis la tranche d'eau la plus profonde, les gradients verticaux de vitesses zonales calculés par la fonction **grad_moy()**.

La dérive du bateau, dX , entre le début et la fin de la station est calculée par la fonction **calc_deriv()**. Les dérivées calculées sont rangées dans la structure de données **ABS** (abs.u_deriv, abs.v_deriv). Cette fonction utilise les informations de position entrées par l'utilisateur dans le fichier de description de la station (voir chapitre III - 4 - b).

```
typedef struct abs {
    float u_deriv;           Composante zonale de la dérive du bateau
    float v_deriv;           Composante méridienne de la dérive du bateau
    float u_ref;             Composante zonale de la vitesse de référence (méthode 1)
    float v_ref;             Composante méridienne de la vitesse de référence (méthode 1)
    float u_baro;            Intégrale de la composante zonale de la vitesse barocline
    float v_baro;            Intégrale de la composante méridienne de la vitesse barocline
    float u_mes;             Intégrale de la composante zonale de la vitesse brute mesurée
    float v_mes;             Intégrale de la composante méridienne de la vitesse brute mesurée
    float u_ref_cell;        Composante zonale de la vitesse de référence (méthode 2)
    float v_ref_cell;        Composante méridienne de la vitesse de référence (méthode 2)
} ABS;
```

Les derniers termes de l'équation et la vitesse absolue sont calculés par la fonction **vit_abs()** :

$$\frac{1}{T} \left\{ \int U_{\text{mesuré}}(t) dt \right\}$$

est facilement calculable car les mesures sont acquises sur une base temporelle. L'heure d'acquisition, en seconde, de chaque cycle de mesures est enregistrée dans chaque entête.

$$\frac{1}{T} \left\{ \int U_{\text{baro}}(t) dt \right\}$$

le calcul de cette intégrale nécessite de transformer le profil vertical de vitesse barocline calculé par la fonction **vit_rel()** en une série temporelle. Comme la profondeur de chaque cellule est connue, la série temporelle est créée en répartissant proportionnellement les vitesses relatives de chaque tranche d'eau (result[k].u_rel, result[k].v_rel) dans les cellules de mesure. Puis l'intégration est faite de la même manière que pour le terme $U_{\text{mesuré}}$.

La fonction **vit_abs()** permet de faire l'intégration de la vitesse mesurée et de la vitesse barocline suivant 2 méthodes :

Méthode 1 - La moyenne des vitesses $U_{\text{mesurée}}$ et U_{baro} est calculée pour chaque cycle de mesures, puis ces valeurs moyennes sont intégrées sur la durée du profil. A la fin de l'intégration la vitesse de référence, $U_{\text{réf}}$, est calculée. Par exemple, pour $U_{\text{mesuré}}$, on calcule :

$$\frac{1}{T} \left\{ \int \left(\frac{1}{n} \sum_i U_{\text{mesuré}}^i(t) \right) dt \right\}$$

où :

n : est le nombre de cellules d'un cycle de mesures.

i : est une cellule d'un cycle de mesures.

Méthode 2 - L'intégration temporelle de $U_{\text{mesurée}}$ et U_{baro} est réalisée au niveau de chaque cellule. On obtient ainsi en fin d'intégration une valeur intégrale de $U_{\text{mesurée}}$ et U_{baro} pour les cellules n°1, n°2, ..., n°19. A la fin de l'intégration une vitesse de référence est calculée pour chaque cellule. La vitesse de référence finale, $U_{\text{réf}}$, est alors calculée en faisant la moyenne des valeurs de chaque cellule. Ne sont prises en compte que les cellules pour lesquelles l'intégration a été faite avec un nombre 'suffisant' de mesures (i.e. les cellules où la vitesse de référence a été calculée avec au moins 90% du nombre maximal de mesures) :

$$\frac{1}{n} \sum_i \left(\frac{1}{T} \left\{ \int U_{\text{mesuré}}^i(t) dt \right\} \right)$$

La vitesse absolue dans chaque tranche d'eau est alors calculée et les valeurs sont rangées dans la structure de données **RESULT** (result[k].u_abs, result[k].v_abs). Les résultats sont écrits dans un fichier ASCII par la fonction **ecr_res()**. Le nom de ce fichier est donné par l'utilisateur dans le fichier de description de la station. Le début d'un fichier résultat est reporté ci-après :

8.00	-36.47	18.15	-40.29	15.15	-999.90	-999.90	-999.90	-999.90	0.00	-999.90	-999.90
24.00	-4.41	0.08	-8.24	-2.92	32.05	-18.07	2.93	4.38	7.97	-20.21	2.34
40.00	31.85	-15.32	28.03	-18.32	36.27	-15.40	6.14	11.66	27.22	-6.98	-5.43
56.00	57.48	-18.99	53.65	-21.99	25.62	-3.68	26.35	13.47	36.13	31.98	-20.28
72.00	49.09	-11.54	45.27	-14.54	-8.39	7.45	9.78	9.25	44.84	60.58	-23.22
88.00	37.56	-14.17	33.74	-17.17	-11.53	-2.64	5.63	4.22	52.73	52.70	-14.40
104.00	32.03	-15.64	28.21	-18.64	-5.53	-1.47	3.48	5.04	60.57	40.24	-16.53
120.00	29.08	-13.35	25.26	-16.35	-2.95	2.30	2.01	2.97	68.14	34.85	-18.14
136.00	25.82	-11.85	21.99	-14.85	-3.27	1.50	2.59	3.11	72.21	31.80	-15.82
152.00	22.96	-6.61	19.14	-9.60	-2.85	5.24	2.21	3.38	72.60	28.48	-14.19
168.00	19.47	2.12	15.65	-0.88	-3.49	8.72	2.19	2.53	76.64	25.26	-8.75

184.00 13.59 8.50 9.77 5.50 -5.88 6.38 2.85 3.98 78.97 21.21 0.46

Chaque enregistrement du fichier comprend :

- 1 - La profondeur moyenne de la tranche d'eau (m).
- 2 - La composante zonale de la vitesse absolue (cm/s).
- 3 - La composante méridienne de la vitesse absolue (cm/s).
- 4 - La composante zonale de la vitesse barocline (cm/s).
- 5 - La composante méridienne de la vitesse barocline (cm/s).
- 6 - Le cisaillement vertical de la composante zonale de la vitesse.
- 7 - Le cisaillement vertical de la composante méridienne de la vitesse.
- 8 - L'écart-type du cisaillement vertical de la composante zonale de la vitesse.
- 9 - L'écart-type du cisaillement vertical de la composante méridienne de la vitesse.
- 10 - Le nombre d'estimations de valeurs de cisaillement dans chaque tranche d'eau.
- 11 - La composante zonale de la vitesse brute mesurée (cm/s).
- 12 - La composante méridienne de la vitesse brute mesurée (cm/s).

Le nombre d'estimations de cisaillement vertical par tranche d'eau n'est pas un nombre entier car la répartition des cisaillements, estimés pour chaque cellule, dans chaque tranche d'eau se fait de manière pondérée : si une cellule est à cheval sur 2 tranches d'eau le cisaillement dans cette cellule est réparti proportionnellement dans chaque tranche d'eau.

Dans l'exemple ci-dessus, bien qu'il n'y ait aucune mesure dans la première tranche d'eau (1^{ère} ligne, 10^{ème} colonne à 0), la vitesse absolue a été calculée dans cette tranche (colonne 2 et 3, 1^{ère} ligne). Ceci s'explique parce que lors de l'intégration verticale des gradients de vitesse, le calcul dans la tranche i fait intervenir le cisaillement et la vitesse absolue de la tranche $i + 1$.

La fonction **vit_moy()** n'a pas d'utilité propre à la connaissance du profil de vitesse absolue, elle permet simplement de faire la moyenne par tranche d'eau des mesures brutes de vitesse, avant traitement.

III - 4 - m Traitement séparé des profils descente et montée

La vitesse de référence ne peut être obtenue qu'en intégrant les vitesses mesurées sur la durée de la station. Il est intéressant de pouvoir comparer les profils de vitesse barocline de la descente et la montée pour vérifier si :

- les gradients verticaux de vitesses sont les mêmes pour les 2 profils.
- les maxima et minima de vitesse se trouvent à la même profondeur à la descente et la montée. C'est une vérification indirecte de la qualité de l'intégration verticale de la vitesse verticale utilisée pour calculer les profondeurs des cycles de mesure.

La deuxième partie du programme **ladcpuv.c** calcule les profils de vitesse barocline de la 'descente' et la 'montée'.

III - 5 Lecture de la configuration du L-ADCP : LADCPCNF.C

Ce programme lit les 3 premiers enregistrements du premier cycle de mesures du profil binaire L-ADCP et crée un fichier où sont reportées les informations qui décrivent la configuration du L-ADCP. La description détaillée de ces informations est donnée dans le manuel du L-ADCP⁸.

Le programme demande :

- 1 - le nom du fichier binaire généré par le L-ADCP.
- 2 - le nom du fichier où sera inscrite la configuration de l'appareil.

Un fichier résultat est reporté ci-dessous. La nomenclature de chaque information est celle utilisée dans la documentation RDI.

FIXED LEADER DATA

```

CPU F/W Version :          5
CPU F/W Revision :        27

System Configuration LSB BITS : 01001001
System Configuration MSB BITS : 01000010

Number of beams           BM : 4

Number of Cells           WN : 19
Pings Per Ensemble       WP : 6
Depth Cell Length        WS : 1600
Blank After Transmit     WF : 1600
Profiling Mode           WM : 4
Low Correlation Threshold WC : 32
Number of code repetitions cr : 8
PerCentGood Minimum      WG : 25
Error Velocity Threshold WE : 9999

Minutes                   TP : 0
Secondes                  TP : 0
Hundredths                TP : 0

Coordinate Transformation EX : 00011111

Heading Alignment        EA : 0
Heading Bias             EB : 0
Sensor Source            EZ : 01111111

Sensor Available         sa : 00011101
Bin 1 distance           dis1 : 3330

Transmit Pulse Length    WT : 1653
Starting Depth Cell      WL : 0
Ending Depth Cell        WP : 5
False Target Threshold   WA : 255
Transmit Lag Repetition  LagD : 207
    
```

VARIABLE LEADER DATA

```

Ensemble Number : 1

RTC Year          TS : 94
RTC Month         TS : 11
RTC Day          TS : 6
RTC Hour         TS : 18
RTC Minute       TS : 36
RTC Second       TS : 41
RTC Hundredth    TS : 5
Ensemble Number fiel  MSB : 0

BIT result Byte 13      : 00000000

BIT result Byte 14      : 00000000

Speed of Sound        EC : 1531
Depth of Transducer   ED : 0
Heading              EH : 22915
Pitch                EP : -52
Roll                 ER : 110
Salinity             ES : 35
Temperature          ET : 2360

Maximum Ping Time - Minute MPT : 59
Maximum Ping Time - Second MPT : 59
Maximum Ping Time - Hundre. MPT : 34

Heading Standard Deviation : 2
Pitch Standard Deviation   : 25
Roll Standard Deviation    : 5

ADC Channel 0          : 48
ADC Channel 1          : 138
ADC Channel 2          : 2
ADC Channel 3          : 205
ADC Channel 4          : 164
ADC Channel 5          : 152
ADC Channel 6          : 0
ADC Channel 7          : 0
    
```

IV - Les fichiers utilisés et créés par la chaîne de traitement

Ce chapitre recense l'ensemble des fichiers utilisés et créés par la chaîne de traitement.

Fichier Binaire L-ADCP

Il s'agit du fichier généré par le L-ADCP. Il est écrit en binaire DOS et est décodé par le programme **ladcpbin.c**. Un profil complet L-ADCP, profil de 'descente' et profil de 'montée', peut être contenu dans plusieurs fichiers binaires. Le programme LADCPBIN.C se charge de concaténer les différents fichiers afin de reconstituer le profil complet. Les noms de ces fichiers doivent être placés dans le fichier de description de la station.

Fichier ASCII de description de la station

Ce fichier permet de grouper les informations nécessaires au traitement d'une station L-ADCP. Il est généré par l'utilisateur à partir d'un traitement de texte.

```

Nom du repertoire ou sont stockes les fichiers.
/users/dupont/cruise/ladcp/sta010/

Nom du fichier binaire ADCP (donnees brutes binaire) :
etal_010.000 etal_010.001 etal_010.002

Nom du fichier ascii ADCP (donnees brutes ascii) :
etal_010.asc

Nom du fichier des parametres :
etal_010.par

Nom du fichier ADCP (donnees traitees) :
etal_010.res

Position GPS en debut de station
Lat_deg Lat_min - Lon_deg Lon_min
 6 51.833      -50 -32.505

Date lors du point GPS de debut de station
yyyy mm jj hh mm ss
1995 09 11 06 36 53

Profondeur en m de l'ADCP lors du point GPS de debut de station (flottant)
0.

Position GPS en fin de station
Lat_deg Lat_min - Lon_deg Lon_min
 6 51.885      -50 -32.405

Date lors du point GPS en fin de station
yyyy mm jj hh mm ss
1995 09 11 09 47 50

Profondeur en m de l'ADCP lors du point GPS de fin de station (flottant)
0.

Deviation magnetique (format flottant) :
-18.50

TEST : Limite inferieure du PerCent Good (format flottant) :
30.

TEST : Limite superieure de tangage (format flottant) :
18.

TEST : Limite superieure de roulis (format flottant) :
15.

TEST : Limite superieure de l'erreur de vitesse (format flottant) :
2.

TEST : Nb d'ecart-type pour le test de rejet vitesse (format flottant) :
2.

TEST : Nb de diff. moyenne pour le test de rejet vitesse (format flottant) :
4.

TEST : Nb d'ecart-type pour le test de rejet gradient (format flottant) :
2.

TEST : Nb de diff. moyenne pour le test de rejet gradient (format flottant) :
4.

TEST : Seuil pour la detection du fond (format flottant) :
15.

Detection des donnees du fond (oui=1 - non=0)
1

Methode de calcul de la vitesse absolue
1

```

Fichier ASCII L-ADCP (*.asc)

Ce fichier est généré par le programme **ladcpbin.c** après lecture du fichier binaire brut. Il ne contient que les informations nécessaires au traitement. Le nom de ce fichier doit être placé dans le fichier de description de la station. Il est important d'utiliser l'extension **.asc** dans le nom du fichier car les programmes de tracé l'utilisent.

```
000371 019 94/11/06 19:26:38 05 3.470 173.81 0.83 4.02 1465.00 0.00
-0.90 4.70 87.90 -0.20 62.250 117.750 100.00
-2.50 4.80 87.70 -0.10 65.000 119.250 100.00
-5.90 2.80 87.20 -0.10 55.250 112.750 100.00
-8.10 2.90 87.30 0.30 50.500 108.500 100.00
-8.70 6.10 87.70 -1.40 43.750 90.500 100.00
-9.00 7.70 87.30 0.20 41.000 76.500 100.00
-7.50 8.50 87.60 0.00 40.250 66.750 100.00
-8.80 7.90 87.70 -0.50 39.750 62.250 100.00
-9.70 8.60 87.50 -0.20 44.500 57.750 83.00
-11.40 7.60 95.60 -999.90 40.500 45.500 0.00
-15.80 8.40 53.80 -999.90 37.000 31.000 0.00
-10.60 19.30 55.40 -0.80 37.500 31.250 16.00
-14.30 20.60 52.80 -999.90 37.500 33.250 0.00
-999.90 -999.90 -999.90 -999.90 41.750 27.750 0.00
-999.90 -999.90 -999.90 -999.90 37.250 20.750 0.00
-999.90 -999.90 -999.90 -999.90 36.000 12.000 0.00
.....
```

L'entête comprend les informations suivantes :

- Le numéro du cycle de mesures (*num_ens* : 000371)
- Le nombre de cellules (*nb_cell* : 019)
- L'année de l'acquisition (*an* : 94)
- le mois de l'acquisition (*mois* : 11)
- Le jour de l'acquisition (*jour* : 06)
- L'heure de l'acquisition (*heure* : 19)
- Les minutes de l'acquisition (*min* : 26)
- Les secondes de l'acquisition (*sec* : 38)
- Les centièmes de seconde de l'acquisition (*cent* : 05)
- La température du capteur interne en °C (*temp* : 3.470)
- Le cap mesuré par le capteur interne (*heading* : 173.81)
- Le tangage mesuré par le capteur interne (*pitch* : 0.83)
- Le roulis mesuré par le capteur interne (*roll* : 4.02)
- La vitesse du son utilisée (*vit_son* : 1465.00)
- La vitesse verticale du bâti au moment de l'acquisition du cycle de mesures (*w_moy* : 0.00)

L'entête est suivie d'enregistrements qui décrivent les variables de chaque cellule. Dans le cas présenté ci-dessus il y a 19 cellules (*nb_cell* = 19). Ces enregistrements sont composés de :

- La composante zonale de la vitesse en cm/s
- La composante méridienne de la vitesse en cm/s
- La composante verticale de la vitesse en cm/s
- L'erreur de vitesse en cm/s
- La moyenne des intensités d'écho des 4 faisceaux
- La moyenne des amplitudes de corrélation des 4 faisceaux
- La valeur du pourcentage de données correctes extraites du 4^{ème} faisceau

Fichier ASCII des paramètres (*.par)

Ce fichier est généré par le programme **ladcpbin.c** après lecture du fichier binaire brut. Il ne contient plus que les informations sur la configuration de l'acquisition des données. Le format de ce fichier est décrit au chapitre III-3-e. Le nom de ce fichier doit être placé dans le fichier de description de la station. Il est important d'utiliser l'extension **.par** dans le nom du fichier car les programmes de tracé l'utilisent.

```
Numero du dernier ensemble:  
8586  
Nombre d'ensembles:  
8586  
Nombre de cellules par ensemble:  
19  
Taille d'une cellule:  
16.00  
Profondeur 1ere cellule:  
33.33  
Salinité:  
35.000
```

Fichier ASCII résultat (*.res)

Ce fichier est généré par le programme **ladcpuv.c**. Il contient les profils de vitesses absolues. Le format de ce fichier est décrit au chapitre III-4-1. Le nom de ce fichier doit être placé dans le fichier de description de la station. Il est important d'utiliser l'extension **.res** dans le nom du fichier car les programmes de tracé l'utilisent.

8.00	-36.47	18.15	-40.29	15.15	-999.90	-999.90	-999.90	-999.90	0.00	-999.90	-999.90
24.00	-4.41	0.08	-8.24	-2.92	32.05	-18.07	2.93	4.38	7.97	-20.21	2.34
40.00	31.85	-15.32	28.03	-18.32	36.27	-15.40	6.14	11.66	27.22	-6.98	-5.43
56.00	57.48	-18.99	53.65	-21.99	25.62	-3.68	26.35	13.47	36.13	31.98	-20.28
72.00	49.09	-11.54	45.27	-14.54	-8.39	7.45	9.78	9.25	44.84	60.58	-23.22
88.00	37.56	-14.17	33.74	-17.17	-11.53	-2.64	5.63	4.22	52.73	52.70	-14.40
104.00	32.03	-15.64	28.21	-18.64	-5.53	-1.47	3.48	5.04	60.57	40.24	-16.53
120.00	29.08	-13.35	25.26	-16.35	-2.95	2.30	2.01	2.97	68.14	34.85	-18.14
136.00	25.82	-11.85	21.99	-14.85	-3.27	1.50	2.59	3.11	72.21	31.80	-15.82
152.00	22.96	-6.61	19.14	-9.60	-2.85	5.24	2.21	3.38	72.60	28.48	-14.19
168.00	19.47	2.12	15.65	-0.88	-3.49	8.72	2.19	2.53	76.64	25.26	-8.75
184.00	13.59	8.50	9.77	5.50	-5.88	6.38	2.85	3.98	78.97	21.21	0.46
200.00	4.37	9.14	0.55	6.15	-9.22	0.64	3.05	3.90	81.09	14.90	7.03
216.00	-4.48	5.43	-8.31	2.43	-8.85	-3.71	4.60	3.26	85.58	5.47	7.65

Chaque enregistrement du fichier comprends :

- 1 - La profondeur moyenne de la tranche d'eau (m).
- 2 - La composante zonale de la vitesse absolue (cm/s).
- 3 - La composante méridienne de la vitesse absolue (cm/s).
- 4 - La composante zonale de la vitesse barocline (cm/s).
- 5 - La composante méridienne de la vitesse barocline (cm/s).
- 6 - Le cisaillement vertical de la composante zonale de la vitesse (cm/s).
- 7 - Le cisaillement vertical de la composante méridienne de la vitesse (cm/s).
- 8 - L'écart-type du cisaillement vertical de la composante zonale de la vitesse.
- 9 - L'écart-type du cisaillement vertical de la composante méridienne de la vitesse.
- 10 - Le nombre d'estimations de valeurs de cisaillement dans chaque tranche d'eau.
- 11 - La composante zonale de la vitesse brute mesurée (cm/s).
- 12 - La composante méridienne de la vitesse brute mesurée (cm/s).

Fichier ASCII intermédiaire : ladcp.net

Ce fichier est généré par le programme **ladcpuv.c** après le passage par la fonction **nettoyage()**. Ces données peuvent être visualisées par le script MATLAB **nettoyag.m** (cf. chapitre V). Le début d'un tel fichier est reporté ci-dessous :

```
000022 67171.1484 23.760 57.54 2.37 6.36 1531.00 0.00
 6.04 -2.68 3.50 -3.30 117.250 124.250 100.00
38.91 -20.86 4.40 -4.40 123.750 119.750 100.00
80.06 -46.47 -0.20 -5.10 130.000 115.250 100.00
77.19 -34.88 8.00 9.00 111.000 105.250 100.00
60.02 -24.59 6.40 0.00 98.750 107.500 100.00
51.23 -29.47 6.90 0.40 92.250 101.000 100.00
47.56 -28.08 6.70 -0.80 82.750 100.750 100.00
45.33 -23.64 5.50 0.10 79.250 102.250 100.00
43.15 -20.75 0.40 1.60 84.750 87.000 83.00
42.92 -14.01 6.20 3.10 73.500 103.000 100.00
36.67 -10.04 2.60 5.40 73.500 87.250 83.00
37.18 4.96 -1.60 5.10 76.250 70.500 83.00
23.70 1.92 2.70 3.80 64.250 65.750 83.00
16.37 2.65 14.70 -1.80 61.750 49.000 66.00
-0.62 -3.86 8.00 2.30 59.750 47.000 50.00
-999.90 -999.90 -999.90 2.30 59.250 30.250 16.00
-999.90 -999.90 -999.90 -999.90 61.000 14.500 0.00
-999.90 -999.90 -999.90 -999.90 57.000 10.000 0.00
-999.90 -999.90 -999.90 20.10 62.250 18.250 16.00
```

L'entête comprend les informations suivantes :

- Le numéro du cycle de mesures (00022)
- L'heure de l'acquisition en secondes (67171.1484)
- La température du capteur interne en °C (23.760)
- Le cap mesuré par le capteur interne (57.54)
- Le tangage mesuré par le capteur interne (2.37)
- Le roulis mesuré par le capteur interne (6.36)
- La vitesse du son utilisée (1531.00)
- La vitesse verticale du bâti au moment de l'acquisition du cycle de mesures (w_{moy} : 0.00)

L'entête est suivie d'enregistrements qui décrivent les variables de chaque cellule. Dans le cas présenté ci-dessus il y a 19 cellules et donc 19 enregistrements suivent l'entête. Ces enregistrements sont composés de :

- La composante zonale de la vitesse en cm/s
- La composante méridienne de la vitesse en cm/s
- La composante verticale de la vitesse en cm/s
- L'erreur de vitesse en cm/s
- La moyenne des intensités d'écho des 4 faisceaux
- La moyenne des amplitudes de corrélation des 4 faisceaux
- La valeur du pourcentage de données correctes extraites du 4ème faisceau.

Fichier ASCII intermédiaire : **ladcp.pro**

Ce fichier est généré par le programme **ladcpuv.c** après le passage par la fonction **prof_max()**. Ces données peuvent être visualisées par le script MATLAB **profmax.m** (cf. chapitre V).

Le début d'un tel fichier est reporté ci-dessous :

```
22 33.299999 -0.200046
23 33.940022 7.900000
24 32.749092 -14.700000
25 31.906530 -10.400000
26 31.388531 -6.400000
27 31.113077 -3.400000
28 29.930248 -14.600000
```

Chaque enregistrement est composé :

- du numéro du cycle de mesure
- de la profondeur du cycle de mesure en m
- de la vitesse verticale en cm/s

Fichier ASCII intermédiaire : ladcp.fon

Ce fichier est généré par le programme **ladcpuv.c** après le passage par la fonction **rej_fond()**. Ces données peuvent être visualisées par le script MATLAB **rejfond.m** (cf. chapitre V). Son format est identique à celui du fichier **ladcp.net**.

Le début d'un tel fichier est reporté ci-dessous :

```
000022 67171.1484 23.760 57.54 2.37 6.36 1531.00 0.00
 6.04 -2.68 3.50 -3.30 117.250 124.250 100.00
38.91 -20.86 4.40 -4.40 123.750 119.750 100.00
80.06 -46.47 -0.20 -5.10 130.000 115.250 100.00
77.19 -34.88 8.00 9.00 111.000 105.250 100.00
60.02 -24.59 6.40 0.00 98.750 107.500 100.00
51.23 -29.47 6.90 0.40 92.250 101.000 100.00
47.56 -28.08 6.70 -0.80 82.750 100.750 100.00
45.33 -23.64 5.50 0.10 79.250 102.250 100.00
43.15 -20.75 0.40 1.60 84.750 87.000 83.00
42.92 -14.01 6.20 3.10 73.500 103.000 100.00
36.67 -10.04 2.60 5.40 73.500 87.250 83.00
37.18 4.96 -1.60 5.10 76.250 70.500 83.00
23.70 1.92 2.70 3.80 64.250 65.750 83.00
16.37 2.65 14.70 -1.80 61.750 49.000 66.00
-0.62 -3.86 8.00 2.30 59.750 47.000 50.00
-999.90 -999.90 -999.90 2.30 59.250 30.250 16.00
-999.90 -999.90 -999.90 -999.90 61.000 14.500 0.00
-999.90 -999.90 -999.90 -999.90 57.000 10.000 0.00
-999.90 -999.90 -999.90 20.10 62.250 18.250 16.00
```

L'entête comprend les informations suivantes :

- Le numéro du cycle de mesures (00022)
- L'heure de l'acquisition en secondes (67171.1484)
- La température du capteur interne en °C (23.760)
- Le cap mesuré par le capteur interne (57.54)
- Le tangage mesuré par le capteur interne (2.37)
- Le roulis mesuré par le capteur interne (6.36)
- La vitesse du son utilisée (1531.00)
- La vitesse verticale du bâti au moment de l'acquisition du cycle de mesures (w_{moy} : 0.00)

L'entête est suivie d'enregistrements qui décrivent les variables de chaque cellule. Dans le cas présenté ci-dessus il y a 19 cellules et donc 19 enregistrements suivent l'entête. Ces enregistrements sont composés de :

- La composante zonale de la vitesse en cm/s
- La composante méridienne de la vitesse en cm/s
- La composante verticale de la vitesse en cm/s
- L'erreur de vitesse en cm/s
- La moyenne des intensités d'écho des 4 faisceaux
- La moyenne des amplitudes de corrélation des 4 faisceaux
- La valeur du pourcentage de données correctes extraites du 4ème faisceau.

Fichier ASCII intermédiaire : **ladcp.cor**

Ce fichier est généré par le programme **ladcpuv.c** après le passage par la fonction **cor_vit()**. Ces données peuvent être visualisées par le script MATLAB **corrvit.m** (cf. chapitre V). Son format est identique à celui du fichier **ladcp.pro**.

Le début d'un tel fichier est reporté ci-dessous :

```
22 33.299999 -0.200046
23 33.940022 7.900000
24 32.749092 -14.700000
25 31.906530 -10.400000
26 31.388531 -6.400000
27 31.113077 -3.400000
28 29.930248 -14.600000
```

Chaque enregistrement est composé :

- du numéro du cycle de mesure
- de la profondeur du cycle de mesure en m
- de la vitesse verticale en cm/s

Fichier ASCII intermédiaire : ladcp.ent

Ce fichier est généré par le programme **ladcpuv.c** après le passage par la fonction **cor_vit()**. Ces données peuvent être visualisées par le script MATLAB **trac_ent.m** (cf. chapitre V).

Le début d'un tel fichier est reporté ci-dessous :

```
1.900341 35 24.299999 354.920013 5.850000 10.170000 1532.719116 55.592754
6.148071 36 24.459999 347.040009 6.240000 9.440000 1533.182617 48.655792
10.114826 37 24.610001 350.410004 5.250000 9.440000 1533.615356 48.655792
14.783436 38 24.750000 348.179993 5.000000 9.380000 1534.034180 65.984802
20.200146 39 24.860001 347.040009 4.300000 8.740000 1534.391724 67.128250
25.312473 40 24.950001 338.799988 3.990000 7.820000 1534.695068 58.469337
30.980656 41 25.030001 332.809998 3.730000 7.170000 1534.983276 80.851791
37.388226 42 25.080000 322.480011 2.900000 5.860000 1535.211304 76.727203
```

Chaque enregistrement est composé :

- 1 - de la profondeur du cycle de mesure au niveau des bases acoustiques
- 2 - du numéro du cycle de mesure
- 3 - de la température mesurée par le L-ADCP
- 4 - du cap en degré
- 5 - du tangage en degré
- 6 - du roulis en degré
- 7 - de la vitesse du son en m/s
- 8 - de la vitesse verticale du L-ADCP en cm/s

Fichier ASCII intermédiaire : ladcp.ref

Ce fichier est généré par le programme **ladcpuv.c** après le passage par la fonction **vit_abs()**. Il permet de conserver les détails du calcul de la vitesse de référence. Un tel fichier est reporté ci-après :

Composante zonale de la derive du bateau : 1.86 cm/s
 Composante meridienne de la derive du bateau : 0.81 cm/s

METHODE 1

Composante zonale de la vitesse de reference : 1.99 cm/s
 Composante meridienne de la vitesse de reference : -6.81 cm/s
 Composante zonale de la vitesse barocline : 3.42 cm/s
 Composante meridienne de la vitesse baroclinee : -3.41 cm/s
 Composante zonale de la vitesse mesuree : 3.56 cm/s
 Composante meridienne de la vitesse mesuree : -11.02 cm/s

METHODE 2

Composante zonale de la vitesse de reference : 2.10 cm/s
 Composante meridienne de la vitesse de reference : -6.94 cm/s

Information par cellule :

Cellule	U_REF	V_REF	U_BARO	V_BARO	U_MES	V_MES	N_MES
1	2.11	-6.89	3.68	-3.73	3.93	-11.43	4889.0
2	2.17	-6.79	3.62	-3.50	3.94	-11.10	4735.0
3	2.23	-7.06	3.70	-3.32	4.07	-11.19	4680.0
4	2.13	-7.06	3.55	-3.18	3.83	-11.05	4565.0
5	2.00	-7.03	3.35	-3.18	3.49	-11.02	4442.0
6	1.92	-6.80	3.19	-3.10	3.25	-10.72	4167.0
7	1.53	-6.89	3.05	-3.10	2.72	-10.80	3885.0
8	1.02	-6.97	2.94	-3.09	2.10	-10.88	3471.0
9	0.57	-7.22	3.18	-3.10	1.89	-11.13	2908.0
10	-0.04	-7.18	3.29	-3.16	1.39	-11.15	2368.0
11	-0.40	-7.48	3.41	-3.17	1.15	-11.47	1861.0
12	-0.71	-7.17	3.88	-3.22	1.31	-11.20	1392.0
13	-1.07	-7.28	4.43	-3.26	1.50	-11.35	1052.0
14	-1.94	-8.25	5.65	-3.28	1.85	-12.34	733.0
15	-6.17	-7.86	8.09	-3.39	0.07	-12.06	458.0
16	-8.14	-9.25	12.07	-3.58	2.06	-13.64	287.0
17	-9.11	-8.69	12.56	-3.56	1.59	-13.06	202.0
18	-8.86	-6.80	13.56	-3.54	2.84	-11.15	130.0
19	-7.36	-7.87	14.84	-3.47	5.62	-12.16	123.0

Fichier ASCII intermédiaire : ladcp.des - ladpc.mon

Ce fichier est généré par le programme **ladcpuv.c** après le calcul des profils de vitesses baroclines de descente et de montée. Ces données peuvent être visualisées par le script MATLAB **desmont.m** (cf. chapitre V). Ces fichiers ont un format identique au fichier résultat (*.res).

8.00	-36.47	18.15	-40.29	15.15	-999.90	-999.90	-999.90	-999.90	0.00	-999.90	-999.90
24.00	-4.41	0.08	-8.24	-2.92	32.05	-18.07	2.93	4.38	7.97	-20.21	2.34
40.00	31.85	-15.32	28.03	-18.32	36.27	-15.40	6.14	11.66	27.22	-6.98	-5.43
56.00	57.48	-18.99	53.65	-21.99	25.62	-3.68	26.35	13.47	36.13	31.98	-20.28
72.00	49.09	-11.54	45.27	-14.54	-8.39	7.45	9.78	9.25	44.84	60.58	-23.22
88.00	37.56	-14.17	33.74	-17.17	-11.53	-2.64	5.63	4.22	52.73	52.70	-14.40
104.00	32.03	-15.64	28.21	-18.64	-5.53	-1.47	3.48	5.04	60.57	40.24	-16.53
120.00	29.08	-13.35	25.26	-16.35	-2.95	2.30	2.01	2.97	68.14	34.85	-18.14
136.00	25.82	-11.85	21.99	-14.85	-3.27	1.50	2.59	3.11	72.21	31.80	-15.82
152.00	22.96	-6.61	19.14	-9.60	-2.85	5.24	2.21	3.38	72.60	28.48	-14.19
168.00	19.47	2.12	15.65	-0.88	-3.49	8.72	2.19	2.53	76.64	25.26	-8.75
184.00	13.59	8.50	9.77	5.50	-5.88	6.38	2.85	3.98	78.97	21.21	0.46
200.00	4.37	9.14	0.55	6.15	-9.22	0.64	3.05	3.90	81.09	14.90	7.03
216.00	-4.48	5.43	-8.31	2.43	-8.85	-3.71	4.60	3.26	85.58	5.47	7.65

Chaque enregistrement du fichier comprend :

- 1 - La profondeur moyenne de la tranche d'eau (m).
- 2 - La composante zonale de la vitesse absolue (cm/s).
- 3 - La composante méridienne de la vitesse absolue (cm/s).
- 4 - La composante zonale de la vitesse barocline (cm/s).
- 5 - La composante méridienne de la vitesse barocline (cm/s).
- 6 - Le cisaillement vertical de la composante zonale de la vitesse (cm/s).
- 7 - Le cisaillement vertical de la composante méridienne de la vitesse (cm/s).
- 8 - L'écart-type du cisaillement vertical de la composante zonale de la vitesse.
- 9 - L'écart-type du cisaillement vertical de la composante méridienne de la vitesse.
- 10 - Le nombre d'estimations de valeurs de cisaillement dans chaque tranche d'eau.
- 11 - La composante zonale de la vitesse brute mesurée (cm/s).
- 12 - La composante méridienne de la vitesse brute mesurée (cm/s).

V - Visualisation des différentes étapes

Un certain nombre de scripts Matlab ont été créés afin de suivre les différentes étapes du traitement des données du L-ADCP. Ces scripts utilisent des fonctions Matlab fournies avec le module général, exceptée la fonction **SUPTITLE.M**. Cette fonction se trouve dans le répertoire **ladcp/matlab** (voir ANNEXE A). Pour que les différents scripts soient accessibles de n'importe quel répertoire, il suffit de rajouter dans le fichier **.CSHRC** la ligne suivante :

```
setenv MATLABPATH /users/dupont/ladcp/matlab
```

Chacun des scripts utilise les fichiers créés par les logiciels de traitement. L'ouverture de ces fichiers se fait par l'intermédiaire d'une boîte de contrôle. Dans cette boîte, pour naviguer dans les répertoires il faut sélectionner le répertoire puis 'cliquer' sur la touche *'filter'*, pour sélectionner un fichier il faut 'cliquer' sur la touche *'done'*.

Ces scripts sont décrits sommairement ci-dessous.

don_brut.m

Ce script permet de tracer les données brutes générées par le programme **ladcpbin.c** (extension **.asc**).

Ce script demande en entrée :

- 1 - Le numéro de station (utilisé pour le titre du dessin)
- 2 - Le nom du fichier paramètre du profil (extension **.par**)
- 3 - Le nom du fichier des cycles de mesures (extension **.asc**)

Ce programme trace :

- 1 - La vitesse zonale brute
- 2 - La vitesse méridienne brute
- 3 - La vitesse verticale brute
- 4 - L'erreur de vitesse
- 5 - L'amplitude d'écho moyenne des 4 faisceaux
- 6 - La corrélation moyenne des 4 faisceaux
- 7 - Le PerCent Good du 4^{ème} faisceau
- 8 - La température
- 9 - Le cap en degré
- 10 - Le tangage (Pitch) en degré
- 11 - Le roulis (Roll) en degré
- 12 - La vitesse du son

Les 7 premiers paramètres sont tracés cellule par cellule. Les 5 derniers paramètres sont tracés en fonction du numéro du cycle de mesures ce qui permet de se rendre compte si il y a des données manquantes.

nettoyag.m

Ce script doit être utilisé une fois le traitement terminé.

Il permet de tracer les données brutes du L-ADCP après traitement par la fonction **nettoyage()** du programme **ladcpuv.c**. Les graphiques très proches de ceux générés par le script **don_brut.m** permettent de se rendre compte des mesures éliminées après **nettoyag()**.

Ce script demande en entrée :

- 1 - Le numéro de station (utilisé pour le titre du dessin)
- 2 - Le nom du fichier paramètre du profil (extension **.par**)
- 3 - Le nom du fichier des cycles de mesures qui doit être **ladcp.net**

Ce script trace :

- 1 - La vitesse zonale brute
- 2 - La vitesse méridienne brute
- 3 - La vitesse verticale brute
- 4 - L'erreur de vitesse
- 5 - L'amplitude d'écho moyenne des 4 faisceaux
- 6 - La corrélation moyenne des 4 faisceaux
- 7 - Le PerCent Good du 4^{ème} faisceau
- 8 - La température
- 9 - Le Cap en degré
- 10 - 'Pitch' en degré
- 11 - 'Roll' en degré
- 12 - La vitesse du son

Les 7 premiers paramètres sont tracés cellule par cellule. Les 5 derniers paramètres sont tracés en fonction du numéro du cycle de mesures ce qui permet de se rendre compte si il y a des données manquantes.

prof_max.m

Ce script doit être utilisé une fois le traitement terminé.

Il permet de tracer les données du L-ADCP après traitement par la fonction **prof_max()** du programme **ladcpuv.c**. Cette fonction fait un premier calcul de la profondeur de chaque cellule et corrige les vitesses grâce au nouveau calcul de vitesse du fond.

Ce script demande en entrée :

- 1 - Le numéro de station (utilisé pour le titre du dessin)
- 2 - Le nom du fichier paramètre du profil (extension **.par**)
- 3 - Le nom du fichier des cycles de mesures qui doit être **ladcp.pro**

Ce script trace :

- 1 - La profondeur des cycles de mesures
- 2 - La vitesse verticale moyenne corrigée de la vitesse du son

rej_fond.m

Ce script doit être utilisé une fois le traitement terminé.

Il permet de tracer des données du L-ADCP après traitement par la fonction **rej_fond()** du programme **ladcpuv.c**. Cette fonction permet de rejeter les mauvaises mesures dues aux réflexions parasites du signal sonore sur le fond de l'océan. L'algorithme détecte des sauts anormaux d'amplitude d'écho.

Ce script demande en entrée :

- 1 - Le numéro de station (utilisé pour le titre du dessin)
- 2 - Le nom du fichier paramètre du profil (extension **.par**)
- 3 - Le nom du fichier des cycles de mesures qui doit être **ladcp.fon**

Ce script trace :

- 1 - La vitesse zonale brute
- 2 - La vitesse méridienne brute
- 3 - La vitesse verticale brute
- 4 - L'erreur de vitesse
- 5 - L'amplitude d'écho moyenne des 4 faisceaux
- 6 - La corrélation moyenne des 4 faisceaux
- 7 - Le PerCent Good du 4^{ème} faisceau
- 8 - La température
- 9 - Le Cap en degré
- 10 - 'Pitch' en degré
- 11 - 'Roll' en degré
- 12 - La vitesse du son

Les 7 premiers paramètres sont tracés cellule par cellule. Les 5 derniers paramètres sont tracés en fonction du numéro du cycle de mesure ce qui permet de se rendre compte si il y a des données manquantes.

Les 2 graphiques les plus importants sont celui de "l'Amplitude d'écho moyen" et celui de "Corrélation moyenne". Par comparaison avec les graphiques précédents ils permettent de se rendre compte si les réflexions sur le fond de l'océan ont bien été éliminées.

corr_vit.m

Ce script doit être utilisé une fois le traitement terminé.

Il trace des données du L-ADCP après traitement par la fonction **corr_vit()** du programme **ladcpuv.c**. Cette fonction affine le calcul de profondeur des cycles de mesures après élimination des données perturbées par les réflexions sur le fond de l'océan.

Ce script demande en entrée :

- 1 - Le numéro de station (utilisé pour le titre du dessin)
- 2 - Le nom du fichier paramètre du profil (extension **.par**)
- 3 - Le nom du fichier des cycles de mesures qui doit être **ladcp.cor**

Ce script trace :

- 1 - La profondeur des cycles de mesures
- 2 - La vitesse verticale moyenne de chaque cycle de mesures

trac_ent.m

Ce script doit être utilisé une fois le traitement terminé.

Il trace des données du L-ADCP après traitement par la fonction **corr_vit()** du programme **ladcpuv.c**.

Ce script demande en entrée :

- 1 - Le numéro de station (utilisé pour le titre du dessin)
- 3 - Le nom du fichier des entêtes qui doit être **ladcp.ent**

Ce script trace le profil de montée et de descente :

- 1 - de la température mesurée par le L-ADCP
- 2 - du cap
- 3 - du tangage
- 4 - du roulis
- 5 - de la vitesse du son calculée par le logiciel
- 6 - de la vitesse verticale du L-ADCP calculée par le logiciel

trac_res.m

Ce script doit être utilisé une fois le traitement terminé.

Il permet de tracer les données du L-ADCP après traitement complet.

Ce script demande en entrée :

- 1 - Le numéro de station (utilisé pour le titre du dessin)
- 2 - Le nom du fichier des profils (extension **.res**)

Ce script trace :

- 1 - La vitesse moyenne absolue (**résultat final**)
- 2 - La vitesse moyenne par rapport au fond
- 3 - Le cisaillement moyen
- 4 - L'écart-type du cisaillement moyen
- 5 - Le nombre de valeurs par tranche d'eau
- 6 - La vitesse moyenne brute (avant traitement)

descmont.m

Ce script doit être utilisé une fois le traitement terminé.

Il permet de tracer les profils de descente et montée du L-ADCP après traitement complet des données, mais avant le calcul de la vitesse de référence. Les profils sont référencés par rapport au fond.

Ce script demande en entrée :

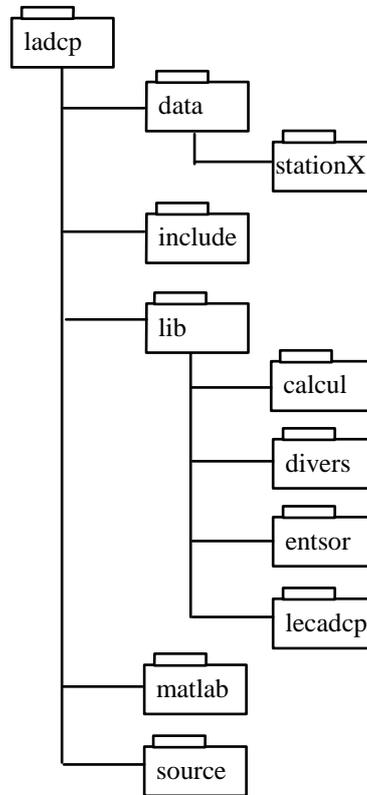
- 1 - Le numéro de station (utilisé pour le titre du dessin)
- 2 - Le nom du fichier du profil descente **ladcp.des**
- 3 - Le nom du fichier du profil montée **ladcp.mon**

Pour chaque composante de vitesse, zonale et méridienne, ce script trace :

- 1 - La vitesse par rapport au fond, calculée par intégration verticale du cisaillement vertical.
- 2 - L'écart-type du cisaillement vertical.
- 3 - Le cisaillement vertical
- 4 - La vitesse moyenne brute telle qu'elle est mesurée par l'ADCP

ANNEXE A : Organisation et Installation du logiciel.

1 - Sous un répertoire utilisateur "/users/dupont/" créer l'arborescence suivante :



2 - Créer les bibliothèques **calcul.a**, **divers.a**, **entsor.a**, **lecadcp.a**.

Les répertoires de chaque bibliothèque contiennent un makefile dont le nom est respectivement : **mcalcul**, **mdivers**, **mentsor**, **mlecadcp**. Avant d'utiliser ces makefiles il faut modifier les chemins qui donnent accès aux fichiers **INCLUDE**, ainsi que l'endroit où seront placées les bibliothèques. Pour exemple le début du fichier **mcalcul** est reporté ci-après. Les chemins qui doivent être modifiés sont mis en gras :

```

#
# @(#)Makefile
#

LIB= /users/dupont/ladcp/lib/calcul.a
FFDIR= /users/dupont/ladcp/include
    
```

Pour créer les bibliothèques se placer successivement dans chacun des 4 répertoires et lancer les commandes :

```

make -f mcalcul
make -f mdivers
make -f mentsor
make -f mlecadcp
    
```

3 - Compiler et faire l'édition de lien des 3 programmes se trouvant dans le répertoire **SOURCE**.

```
ladcpnf.c
ladcpbin.c
ladcpuv.c
```

Pour cela utiliser le fichier **MAKEFILE** qui se trouve dans le répertoire **SOURCE**. Il faut auparavant modifier les chemins d'accès aux bibliothèques. Le **MAKEFILE** est reporté ci-dessous. Les chemins mis en gras doivent être éventuellement modifiés.

```
# MODELE DE MAKEFILE POUR LA COMPILATION ET L'EDITION DE LIENS
# D'UN PROGRAMME QUELCONQUE

# OPTIONS DE COMPILATION:
#   décommenter les lignes suivantes selon votre besoin
# -----
# en mode DEBUG
FFLAGS = -g
CFLAGS = -g

# Directory des fichiers include
INCLUDE = /users/dupont/ladcp/include

# en mode FPA (accélérateur flottant)
#FFLAGS = -fpa
#

# OPTIONS D'EDITIONS DE LIENS
#   décommenter les lignes suivantes selon votre besoin
# -----
LDFLAG1 =
LDFLAG2 =
LDIR1 =

# bibliothèque MATH
LDFLAG1 = -L/usr/lib -lm

# bibliothèque
LDFLAG2 = /users/dupont/ladcp/lib/calcul.a

# bibliothèque
LDFLAG3 = /users/dupont/ladcp/lib/entsor.a

# bibliothèque
LDFLAG4 = /users/dupont/ladcp/lib/divers.a

# bibliothèque
LDFLAG5 = /users/dupont/ladcp/lib/lecadcp.a

LDFLAGS = $(LDFLAG1) $(LDFLAG2) $(LDFLAG3) $(LDFLAG4) $(LDFLAG5)
#
# COMPILATION ET EDITION
#
.c:
    $(CC) $(CFLAGS) -I$(INCLUDE) $< $(LDFLAGS) -o $@
#
```

La commande de compilation et d'édition de lien est la suivante :

```
make -f makefile ladcpnf
make -f makefile ladcpbin
```

```
make -f makefile ladcpuv
```

4 - Placer le chemin du répertoire MATLAB dans le fichier **.CSHRC**.

Ainsi les différents scripts matlab seront accessibles directement :

```
setenv MATLABPATH /users/dupont/ladcp/matlab
```

5 - Créer le répertoire des données.

Dans l'exemple ci-dessus le répertoire des données **DATA** a été placé sous le répertoire **L-ADCP**, puis un répertoire par station est créé.

ANNEXE B : Liste des programmes et fonctions de la chaîne de traitement.

Programmes

ladpcnf.c
 ladcpbin.c
 ladcpuv.c

Fonctions

Bibliothèque

Nom du Fichier

AllocF_1D()	DIVERS	allocfld.c
AllocI_1D()	DIVERS	allocild.c
Alloc_struct()	DIVERS	allocstru.c
calc_deriv()	CALCUL	cal_der.i.c
caldat()	DIVERS	caldat.c
corr_vit()	CALCUL	corr_vit.c
dec_uns_short_int()	LECADCP	unsint.c
dec_short_int()	LECADCP	shortint.c
ec_typ_gra()	CALCUL	ectyp_gr.c
ecr_ens_brut()	ENTSOR	ecr_ens.c
ecr_int1()	ENTSOR	ecr_int1.c
ecr_int2()	ENTSOR	ecr_int2.c
ecr_int3()	ENTSOR	ecr_int3.c
ecr_para()	ENTSOR	ecr_para.c
ecr_res()	ENTSOR	ecr_res.c
grad_brut()	CALCUL	gra_brut.c
grad_moy()	CALCUL	gra_moy.c
ind_pmax()	CALCUL	ind_pmax.c
julday()	DIVERS	julday.c
lec_descr()	ENTSOR	lec_desc.c
lec_ens_brut()	ENTSOR	lec_ens.c
lec_para()	ENTSOR	lec_para.c
lect_bbadcp()	LECADCP	lec_adcp.c
lect_bottom()	LECADCP	lec_bot.c
lect_cor()	LECADCP	lec_cor.c
lect_echo()	LECADCP	lec_echo.c
lect_leadfix()	LECADCP	lec_fix.c
lect_header()	LECADCP	lec_head.c
lect_pcg()	LECADCP	lec_pcg.c
lect_status()	LECADCP	lec_stat.c
lect_leadvar()	LECADCP	lec_var.c
lect_vit()	LECADCP	lec_vit.c
libere_mem()	LECADCP	lib_mem.c
nettoyage()	CALCUL	nettoyag.c
press80()	DIVERS	press80.c
prof_max()	CALCUL	prof_max.c
rej_fond()	CALCUL	rej_fond.c
svel()	DIVERS	svel.c
test_dm_gra()	CALCUL	tstdmgra.c
test_dm_vit()	CALCUL	tstdmvit.c
test_et_gra()	CALCUL	tstetgra.c
test_et_vit()	CALCUL	tstetvit.c
vit_abs()	CALCUL	vit_abs.c
vit_moy()	CALCUL	vit_moy.c
vit_rel()	CALCUL	vit_rel.c
w_moyen()	CALCUL	w_moyen.c

Include

calcul.h
 constant.h

divers.h
entsor.h
lecadp.h
struct.h

Description succincte des fonctions, classées par bibliothèque

CALCUL

calc_deriv()	Calcule la dérive du bateau entre la mise à l'eau et la récupération du L-ADCP.
corr_vit()	Corrige les vitesses à partir d'une estimation de la vitesse du son.
ec_typ_gra()	Calcule pour chaque tranche d'eau l'écart-type du gradient vertical de vitesse horizontale.
grad_brut()	Calcule les gradient verticaux de vitesses horizontales pour chaque cellule.
grad_moy()	Calcule les gradients verticaux moyens de vitesses horizontales par tranche d'eau.
ind_pmax()	Evalue l'indice de l'ensemble atteignant la plus grande profondeur.
nettoyage()	Rejette les mesures brutes erronées (tests d'angle, de PGOOD).
prof_max()	Première estimation de l'immersion des cycles de mesures.
rej_fond()	Rejette les mesures polluées par les réflexions du signal sonore sur le fond de l'océan.
test_dm_gra()	Test de rejet, par écart moyen, des gradients verticaux de vitesses horizontales.
test_dm_vit()	Test de rejet, par écart moyen, des vitesses horizontales.
test_et_gra()	Test de rejet, par écart type, des gradients verticaux de vitesses horizontales.
test_et_vit()	Test de rejet, par écart type, des vitesses horizontales.
vit_abs()	Calcule le profil de vitesse absolue après intégration verticale des gradients de vitesses.
vit_moy()	Calcule la vitesse moyenne par tranche d'eau avant intégration verticale des gradients de vitesses.
vit_rel()	Calcule le profil de vitesse barocline par intégration verticale des gradients de vitesses.
w_moyen()	Calcule la vitesse verticale de descente moyenne du L-ADCP pour un cycle de mesures.

DIVERS

AllocF_1D()	Allocation d'un tableau de flottants à une dimension.
AllocI_1D()	Allocation d'un tableau d'entiers à une dimension.
Alloc_struct()	Allocation de mémoires des structures RESULT et GRAD.
caldat()	Calcule la date à partir du jour Julien.
julday()	Calcule le jour Julien à partir d'une date.
press80()	Conversion de la profondeur (mètres) en pression (decibars).
svel()	Calcule la vitesse du son dans l'eau de mer.

ENTSOR

ecr_ens_brut()	Ecriture d'un cycle de mesures en ASCII.
ecr_int1()	Ecriture de résultats intermédiaires.
ecr_int2()	Ecriture de résultats intermédiaires.
ecr_int3()	Ecriture des résultats intermédiaires
ecr_para()	Ecriture des paramètres d'acquisition.
ecr_res()	Ecriture du profil final.
lec_descr()	Lecture du fichier de description de la station.
lec_ens_brut()	Lecture d'un cycle de mesures en binaire (fichier brut).
lec_para()	Lecture des paramètres d'acquisition.

LECADCP

dec_short_int()	Lecture de binaire (short integer).
dec_uns_short_int()	Lecture de binaire (unsigned short int).
lect_bbadcp()	Lecture d'un cycle de mesures en binaire.
lect_bottom()	Lecture de l'enregistrement BOTTOM-TRACK DATA
lect_cor()	Lecture de l'enregistrement CORREL-MAGNITUDE
lect_echo()	Lecture de l'enregistrement ECHO INTENSITY
lect_leadfix()	Lecture de l'enregistrement LEADER FIXED DATA
lect_header()	Lecture de l'enregistrement HEADER DATA
lect_pcg()	Lecture de l'enregistrement PERCENT GOOD DATA
lect_status()	Lecture de l'enregistrement STATUS DATA

<code>lect_leadvar()</code>	Lecture de l'enregistrement LEADER VARIABLE DATA
<code>lect_vit()</code>	Lecture de l'enregistrement VELOCITY DATA
<code>libere_mem()</code>	Libère la mémoire allouée lors de la lecture d'un cycle de mesures.

ANNEXE C : Structures de données utilisées par le programme (struct.h)

Structures PARA

<pre>typedef struct parametre { unsigned int num_der_ens; unsigned int nb_ens; int nb_cell; float taille_cell; float prof_cell1; float sal; } PARA;</pre>	<p>Numéro du dernier cycle de mesures lu. Nombre de cycles de mesures lus. Nombre de cellules par cycle de mesures. Taille verticale d'une cellule. Profondeur de la 1^{ère} cellule par rapport aux bases acoustiques. Salinité.</p>
---	--

Structure DESCRIPTIF

```
typedef struct date {
    int    an;
    int    mois;
    int    jour;
    int    heure;
    int    min;
    int    sec;
} DATE;
```

```
typedef struct position {
    int    lat_deg;
    float lat_min;
    int    lon_deg;
    float lon_min;
} POS;
```

<pre>typedef struct descriptif { char repertoire[NB_CAR_MAX]; POS p_deb; POS p_fin; DATE d_deb; DATE d_fin; float prof_deb; float prof_fin; float dev_magn; float pgood_min; float pitch_max; float roll_max; float ew_max; float seuil_et_vit; float seuil_dm_vit; float seuil_et_gra; float seuil_dm_gra; float ddb; int detect_fond; int methode; } DESCRIPTIF;</pre>	<p>Répertoire de travail Position du L-ADCP en début de station Position du L-ADCP en fin de station Date et heure de début de station Date et heure de fin de station Profondeur du L-ADCP en début de station Profondeur du L-ADCP en fin de station Déviation magnétique Valeur minimum du percent Good Data Valeur maximum du tangage Valeur maximum du roulis Valeur maximum de l'erreur de vitesse Coefficient utilisé dans le test d'écart-type vitesse Coefficient utilisé dans le test d'écart à la moyenne vitesse Coefficient utilisé dans le test d'écart-type gradient Coefficient utilisé dans le test d'écart à la moyenne gradient Seuil utilisé dans la détection du fond Code pour déterminer si le fond doit être détecté Méthode de calcul de la vitesse absolue</p>
---	--

Structure GRAD

<pre>typedef struct grad { float dru; float drv; float prof; long int cell2; } GRAD;</pre>	<p>Gradient vertical de vitesse zonale entre 2 cellules</p> <p>Gradient vertical de vitesse méridienne entre 2 cellules</p> <p>Profondeur du gradient vertical</p> <p>Indice de la 2^{ème} cellule ayant servi à calculer le gradient vertical</p>
--	--

Structure ENTETE

<pre>typedef struct entete { int num_ens; float temp; float sec; float heading; float pitch; float roll; float vit_son; float w_moy; float tail_cel_cor; } ENTETE;</pre>	<p>Indice du cycle de mesures .</p> <p>Température mesurée par le L-ADCP .</p> <p>Temps en seconde mesuré par le L-ADCP .</p> <p>Cap mesuré par le L-ADCP .</p> <p>Tangage mesuré par le L-ADCP .</p> <p>Roulis mesuré par le L-ADCP .</p> <p>Vitesse du son mesurée par le L-ADCP .</p> <p>Vitesse verticale moyenne calculée par le logiciel .</p> <p>Taille d'une cellule corrigée après correction de vitesse du son.</p>
---	---

Structure CELL

<pre>typedef struct cell { float prof; float ru; float rv; float rw; float error; float amp_echo; float correl; float pcg; } CELL;</pre>	<p>Profondeur de la cellule (en m) calculée par le logiciel .</p> <p>Composante zonale de la vitesse (cm/s) mesurée par le L-ADCP .</p> <p>Composante méridienne de la vitesse (cm/s) mesurée par le L-ADCP .</p> <p>Composante verticale de la vitesse (cm/s) mesurée par le L-ADCP .</p> <p>Erreur de vitesse (cm/s) mesurée par le L-ADCP .</p> <p>Intensité d'écho mesurée par le L-ADCP .</p> <p>Amplitude de corrélation mesurée par le L-ADCP .</p> <p>Percent Good mesuré par le L-ADCP .</p>
--	---

Structure INTCELL

```
typedef struct intcell {
    float    u_baro;
    float    v_baro;
    float    n_baro;
    float    u_mes;
    float    v_mes;
    float    n_mes;
    float    u_ref;
    float    v_ref;
} INTCELL;
```

Structure ABS

```
typedef struct abs {
    float u_deriv;      Composante zonale de la dérive du bateau
    float v_deriv;      Composante méridienne de la dérive du bateau
    float u_ref;        Composante zonale de la vitesse de référence (méthode 1)
    float v_ref;        Composante méridienne de la vitesse de référence (méthode 1)
    float u_baro;       Intégrale de la composante zonale de la vitesse barocline
    float v_baro;       Intégrale de la composante méridienne de la vitesse barocline
    float u_mes;        Intégrale de la composante zonale de la vitesse brute mesurée
    float v_mes;        Intégrale de la composante méridienne de la vitesse brute mesurée
    float u_ref_cell;   Composante zonale de la vitesse de référence (méthode 2)
    float v_ref_cell;   Composante méridienne de la vitesse de référence (méthode 2)
} ABS;
```

Structure RESULT

```
typedef struct result{
    float u_abs;        Vitesse zonale absolue dans une tranche d'eau
    float v_abs;        Vitesse méridienne absolue dans une tranche d'eau
    float u_baro;       Vitesse zonale par rapport au fond dans une tranche d'eau
    float v_baro;       Vitesse méridienne par rapport au fond dans une tranche d'eau
    float u_mes;        Vitesse zonale mesurée dans une tranche d'eau
    float v_mes;        Vitesse méridienne mesurée dans une tranche d'eau
    float u_grad;       Gradient vertical de vitesse zonale dans une tranche d'eau
    float v_grad;       Gradient vertical de vitesse méridienne dans une tranche d'eau
    float n_grad;       Nombre de gradients par tranche d'eau (flottant)
    float et_u_grad;    Ecart-type du gradient vertical de vitesse zonale dans une tranche d'eau
    float et_v_grad;    Ecart-type du gradient vertical de vitesse méridienne dans une tranche d'eau
} RESULT;
```

Annexe D : Exemple de fichier de configuration pour un L-ADCP

Liste des Commandes Utilisées pour le L-ADCP durant La Campagne ETAMBOT 1 (Septembre 1995)

COMMANDES	DESCRIPTIF	PAGE
CR1	Reset the ADCP command set to EEPROM	C-16
BP0	ADCP does not collect Bottom-track data	C-12
CP255	Controls power output of ADCP using a high power module - 255 = max.	C-16
EA+00000	Heading Alignment	C-20
EB+00000	Heading Bias	C-21
EC1500	Speed of sound for ADCP data processing	C-22
ED00000	Depth of transducer	C-22
ES35	Sets the water' salinity value	C-23
EX11111	earth coordinate - use tilts in transformation - allows 3 beams solution - allow bin mapping	C-24
EZ0011101	speed of sound : EC - depth : ED - heading, pitch, roll : internal sensors - salinity : ES - temperature : internal sensor	C-25
TE00:00:00.00	Minimum interval between data collection cycles	C-44
TP00:00.50	Time between pings	C-45
WA255	False target threshold max (255 disable)	C-46
WC025	Low correlation threshold	C-46
WD111100000	Acquire : velocity - correlation - echo intensity - Percent Good	C-47
WE5000	Error velocity threshold	C-47
WF0800	Blank after transmit	C-47
WG025	Percent Good Ping minimum	C-48
WL000,005	Water reference layer Averaging	C-49
WM1	Profiling mode : Dynamic Sea State	C-50
WN020	Number of depth cells	C-53
WP00010	Ping per ensemble	C-53
WS1600	Depth cell size	C-54
WV250	Model Ambiguity velocity	C-54
CF11101	automatic ensemble cycling - automatic ping cycling - binary data output - disable serial output	C-14
CK	Stores present parameters to EEPROM	C-15