

Rémy CHUCHLA

**Constitution d'une base de données
de mesures hydrographiques (CTDO₂), de
traceurs géochimiques et de courant (ADCP).
Exemple de campagnes WOCE réalisées
dans l'Atlantique tropical (1993-1996).**

**Document scientifique et technique
du centre ORSTOM de Brest - N° 83**



**L'Institut français de recherche scientifique
pour le développement en coopération**



**L'INSTITUT FRANÇAIS DE RECHERCHE SCIENTIFIQUE
POUR LE DÉVELOPPEMENT EN COOPÉRATION**

**CENTRE DE BREST
BP 70
29280 Plouzané
France**

**Constitution d'une base de données de mesures hydrographiques (CTDO₂),
de traceurs géochimiques et de courant (ADCP)
Exemple de campagnes WOCE réalisées dans l'Atlantique tropical (1993-1996)**

RÉMY CHUCHLA

**Document Scientifique et Technique du Centre ORSTOM de Brest
Doc. Sci. Tech. Cent. Brest ORSTOM
N° 83, Octobre 1998**

Remerciements

Je tiens à remercier Mr Yves Gouriou pour l'aide précieuse qu'il m'a apporté pour la réalisation de cette base de données.

Merci également à Mr Bernard Bourlès pour ses commentaires éclairés et pertinents pour la rédaction de ce document.

PLAN

I Objectif.....	page 1
II Création de la base.....	page 1
II-1 Philosophie générale.....	page 1
II-2 Organisation de la base.....	page 1
II-3 Exemple d'organisation de la base (Campagne Etambot1).....	page 6
II-4 Description des différents programmes de création de la base.....	page 7
III Description des opérations effectuées par campagne.....	page 9
III-1 Campagne Cither1.....	page 9
III-2 Campagne Cither2.....	page 9
III-3 Campagne Cither3.....	page 10
III-4 Campagne Etambot1.....	page 10
III-5 Campagne Etambot2.....	page 10
IV Accès aux données.....	page 11
IV-1 Description des routines d'accès aux données.....	page 11
IV-1-1 Fonction « sub_base.c ».....	page 11
IV-1-2 Fonction « sub_surface.c ».....	page 12
IV-1-3 Sous-programme « lec_base.f ».....	page 12
IV-1-4 Sous-programme « lec_surface.f ».....	page 13
IV-2 Compilation et édition de liens.....	page 14
IV-2-1 Programme C.....	page 14
IV-2-2 Programme Fortran.....	page 14
IV-3 Exemples d'extraction de données.....	page 15
IV-3-1 Programme écrit en LangageC.....	page 15
IV-3-2 Programme écrit en Langage Fortran 77.....	page 16

V Interface graphique.....	page 19
V-1 Description des routines d'accès à la base.....	page 19
V-1-1 Fonction lec_base.m.....	page 19
V-1-2 Fonction sub_surface.m.....	page 20
V-1-3 Fonction sub_info.....	page 20
V-1-4 Exemple d'un programme de tracés.....	page 21
V-1-5 Exemple d'utilisation de la fonction sub_base.....	page 22
VI Conclusion.....	page 23
VII Annexe.....	page 24
VII-1 Exemple de compilation et d'édition de liens	page 24
VII-2 Liste des numéros des paramètres	page 25
VII-3 Codes de qualité utilisés.....	page 28
VII-3 Liste des fichiers de la base (Juillet 1998).....	page 28

I Objectif :

L'usage a montré que les données océanographiques étaient enregistrées sous des formats forts différents. Aussi pour harmoniser cet état il a été décidé de construire une base de données regroupant tous les types de mesure dans une structure unique facilitant ainsi l'accès à ces données.

Les données de cette nouvelle base, organisée par campagne océanographique, seront accessibles par une routine globale. L'utilisateur pourra faire des interrogations sur les données et les extraire pour ses applications propres.

Dans un premier temps cette base ne concerne que les campagnes « WOCE-France » effectuées dans l'Atlantique tropical.

II Création de la base

II-1 Philosophie générale

L'idée est d'avoir une structure identique pour chaque campagne, accessible par une seule routine permettant d'obtenir les données d'une station (profil) pour une campagne donnée.

Cette base est articulée sur plusieurs modules permettant de se déplacer dans les différents fichiers de la base afin d'obtenir les renseignements désirés (Résumé des informations disponibles et extraction des données).

Il y a donc un fichier binaire par type de mesure et par campagne (ctd, chimie, sadcp, l-adcp), accessible directement en puisant les informations nécessaires dans un fichier « **campagne_entête** ». L'aiguillage suivant la campagne et le type de mesure se faisant grâce au fichier « **campagne_info** ».

II-2 Organisation de la base

Pour une campagne il y a deux fichiers d'informations:

Un fichier « *surface* »

Un fichier « *info* »

Structure d'un fichier « surface » :

Le fichier binaire appelé « **campagne surface** » contient les informations générales (date, position et fond) des stations effectuées. (Il est prévu de rajouter les données météo relatives à ces stations dans ce fichier) .

Paramètres	unités	Type de données
Numéro de station :		int*4
Mois :		int*4
Jour :		int*4
An :		int*4
Heure :		int*4
Minute :		int*4
Seconde :		int*4
Latitude :	Degré-centième	float*4
Longitude :	Degré-centième	float*4
Fond :	m	int*4

Ce fichier contient autant d'enregistrements que de stations. Si une station est manquante le champ « Numéro de station » est affecté de la valeur -1 et les autres champs de la valeur -99 .

Structure d'un fichier « info » :

Le fichier ascii appelé « **campagne info** » contient des renseignements sur le type de mesures effectuées (ctd, chimie, l-adcp ..) ainsi que le nombre de paramètres mesurés et leurs codes **WHP*** et un paramètre indiquant la taille réelle d'un enregistrement en octets dans le fichier binaire.

Les codes utilisés sont présentés en annexe.

Exemple d'un fichier info :

```
etambot2 95 1 15 3 4 5 6 103 104 105 106 80 81 82 83 84 85 86 60
etambot2 95 2 18 1 2 3 4 5 6 7 80 81 82 83 101 102 103 104 105 106 107 144
```

Nom de la campagne. (etambot2)

Numéro de la dernière station effectuée. (95)

Type de la mesure : (1)

1	pour les mesures ctd
2	pour les mesures chimie
3	pour les mesures ladcp
4	pour les mesures sadcp-station
5	pour les mesures sadcp-interstation

Nombre de paramètres mesurés . (15)

Code des paramètres mesurés. (3 4 5 6 103 104 105 106 80 81 82 83 84 85 86)

De 1 à 43 : code WHP .

De 45 à 50 : code utilisé pour le ladcp et sadcp.

De 80 à 92 : code utilisé pour les paramètres physiques calculés.

De 101 à 143 : code qualité correspondant au code WHP.

Taille en octets d'un enregistrement dans le fichier binaire correspondant. (60)

Pour chaque type de mesure (ctd , chimie , l-adcp , sadcp ...) il y a deux fichiers :

Un fichier « *entête* »

Un fichier binaire des mesures « *bin* »

Structure d'un fichier « entête » :

Il existe un fichier binaire « **campagne entete-type de mesure** » par type de mesure. Ce fichier constitue l'interface directe avec le fichier binaire correspondant .

Paramètres	type
Numéro de station	long int*4
Type de mesure	long int*4
Nombre de niveaux de la station (nbniv)	long int*4
Position de la station dans le fichier binaire (pos)	long int*4

Quand une station n'existe pas on affecte aux variables nbniv et pos la valeur -1.

* **WHP** : World Hydrographic Program

Structure des fichiers « bin » :

Il y a un fichier binaire « **campagne_bin-type_de_mesure** »_par type de données (ctd , chimie , adcp , sadcp ...) qui contient les données mesurées et calculées.

L'enregistrement des données est fait selon l'ordre défini dans le fichier **info**.

Structure d'un fichier « bin-ctd » :

Nom du para	unités	Code des paramètres	Type de données	Type de paramètre
Pression	Dbar	3	Float*4	Paramètre mesuré
Température	Degré-C	4	Float*4	Paramètre mesuré
Salinité	Pss-78	5	Float*4	Paramètre mesuré
Oxygène	µmol/kg	6	Float*4	Paramètre mesuré
Code_qualité du paramètre 3		103	Float*4	Code qualité
Code_qualité du paramètre 4		104	Float*4	Code qualité
Code_qualité du paramètre 5		105	Float*4	Code qualité
Code_qualité du paramètre 6		106	Float*4	Code qualité
Profondeur	m	80	Float*4	Paramètre calculé
Température-potentielle	Degré-C	81	Float*4	Paramètre calculé
Sigma-t	Kg/m**3	82	Float*4	Paramètre calculé
Sigma-théta	Kg/m**3	83	Float*4	Paramètre calculé
Sigma-1000	Kg/m**3	84	Float*4	Paramètre calculé
Sigma-1500	Kg/m**3	85	Float*4	Paramètre calculé
Sigma-2000	Kg/m**3	86	Float*4	Paramètre calculé
Sigma-3000	Kg/m**3	87	Float*4	Paramètre calculé
Sigma-4000	Kg/m**3	88	Float*4	Paramètre calculé
Fréquence de Brunt vaissala	cph	89	Float*4	Paramètre calculé
Dsdp : gradient de salinité	Pss-78/Dbar	90	Float*4	Paramètre calculé
Dtdp : gradient de temp-t	Degré-C/Dbar	91	Float*4	Paramètre calculé
Hdyn	m.dyn	92	Float*4	Paramètre calculé

Structure d'un fichier « bin-chimie » :

Les paramètres présents dans ces fichiers dépendent des mesures chimiques effectuées pendant la campagne.

Nom du para	unités	Code des paramètres	Type de données	Type de paramètre
Numéro de bouteille		1	Int*4	
Pression brute	Dbar	2	Float*4	Paramètre mesuré
Pression recalculée	Dbar	3	Float*4	Paramètre mesuré
Température sonde	Degré-C	4	Float*4	Paramètre mesuré
Salinité de la sonde	Pss-78	5	Float*4	Paramètre mesuré
Oxygène de la sonde	µmol/kg	6	Float*4	Paramètre mesuré
Salinité mesurée	Pss-78	7	Float*4	Paramètre mesuré
Oxygène mesuré	µmol/kg	8	Float*4	Paramètre mesuré
Silicate	µmol/kg	9	Float*4	Paramètre mesuré
Nitrate	µmol/kg	10	Float*4	Paramètre mesuré
Nitrite	µmol/kg	11	Float*4	Paramètre mesuré
Phosphate	µmol/kg	12	Float*4	Paramètre mesuré
Fréon-11	Pmol/kg	13	Float*4	Paramètre mesuré
Fréon-12	Pmol/kg	14	Float*4	Paramètre mesuré
Argon	µmol/kg	23	Float*4	Paramètre mesuré

Carbone total	$\mu\text{mol/kg}$	31	Float*4	Paramètre mesuré
Alcalinité totale	$\mu\text{mol/kg}$	32	Float*4	Paramètre mesuré
Fugacité de CO2	$\mu\text{atm de co2}$	33	Float*4	Paramètre mesuré
Ph	Ph	34	Float*4	Paramètre mesuré
Methane	Nmol/kg	39	Float*4	Paramètre mesuré
Azote	$\mu\text{mol/kg}$	40	Float*4	Paramètre mesuré
Oxyde nitreux	Nmol/kg	41	Float*4	Paramètre mesuré
Chlorophylle A	$\mu\text{g/l}$	42	Float*4	Paramètre mesuré
Phaeophytine	$\mu\text{g/l}$	43	Float*4	Paramètre mesuré
Profondeur	m	80	Float*4	Paramètre calculé
Température potentielle	Degre_C	81	Float*4	Paramètre calculé
Sigmat-t	Kg/m^{**3}	82	Float*4	Paramètre calculé
Sigma-théta	Kg/m^{**3}	83	Float*4	Paramètre calculé
Sigma-1000	Kg/m^{**3}	84	Float*4	Paramètre calculé
Sigma-1500	Kg/m^{**3}	85	Float*4	Paramètre calculé
Sigma-2000	Kg/m^{**3}	86	Float*4	Paramètre calculé
Sigma-3000	Kg/m^{**3}	87	Float*4	Paramètre calculé
Sigma-4000	Kg/m^{**3}	88	Float*4	Paramètre calculé
Code_qualité du paramètre 1		101	Short int*2	Code qualité
Code_qualité du paramètre 2		102	Short int*2	Code qualité
Code_qualité du paramètre 3		103	Short int*2	Code qualité
Code_qualité du paramètre 4		104	Short int*2	Code qualité
Code_qualité du paramètre 5		105	Short int*2	Code qualité
Code_qualité du paramètre 6		106	Short int*2	Code qualité
Code_qualité du paramètre 7		107	Short int*2	Code qualité
Code_qualité du paramètre 8		108	Short int*2	Code qualité
Code_qualité du paramètre 9		109	Short int*2	Code qualité
Code_qualité du paramètre 10		110	Short int*2	Code qualité
Code_qualité du paramètre 11		111	Short int*2	Code qualité
Code_qualité du paramètre 12		112	Short int*2	Code qualité
Code_qualité du paramètre 13		113	Short int*2	Code qualité
Code_qualité du paramètre 14		114	Short int*2	Code qualité
Code_qualité du paramètre 23		123	Short int*2	Code qualité
Code_qualité du paramètre 31		131	Short int*2	Code qualité
Code_qualité du paramètre 32		132	Short int*2	Code qualité
Code_qualité du paramètre 33		133	Short int*2	Code qualité
Code_qualité du paramètre 34		134	Short int*2	Code qualité
Code_qualité du paramètre 39		139	Short int*2	Code qualité
Code_qualité du paramètre 40		140	Short int*2	Code qualité
Code_qualité du paramètre 41		141	Short int*2	Code qualité
Code_qualité du paramètre 42		142	Short int*2	Code qualité
Code_qualité du paramètre 43		143	Short int*2	Code qualité

Les paramètres toujours présents dans les fichiers « chimie » sont les suivants : numéro de bouteille, données mesurées, données physiques calculées (z , température potentielle , sigma-t , sigma-théta , sigma 1000 , sigma 1500 , sigma 2000 , sigma 3000 , sigma 4000) et code qualité des paramètres mesurés.

Structure d'un fichier « bin ladcp » :

L-ADCP (Lowered ADCP) : mesures de courant faites en fixant un ADCP sur une bathysonde.

Nom du para	unités	Code des paramètres	Type de données	Type de paramètre
Profondeur	m	45	Float*4	Paramètre calculé
Comp. zonale	cm/s	46	Float*4	Paramètre mesuré
Comp. méridienne	cm/s	47	Float*4	Paramètre mesuré
Nbre de ping		48	Float*4	Paramètre mesuré

Le nombre de 'Ping' est le nombre d'impulsions acoustiques émises par le L-ADCP présentes dans chaque tranche d'eau.

Structure d'un fichier « bin sadcp » :

S-ADCP (Shipboard ADCP) : mesures de courant faites avec un ADCP fixé à la coque d'un navire.

Nom du para	unités	Code des paramètres	Type de données	Type de paramètre
Profondeur	m	45	Float*4	Paramètre calculé
Nbre de profils		48	Float*4	Paramètre mesuré
Comp zonale	cm/s	46	Float*4	Paramètre mesuré
Ecart-type U	cm/s	49	Float*4	Paramètre calculé
Comp méridienne	cm/s	47	Float*4	Paramètre mesuré
Ecart-type V	cm/s	50	Float*4	Paramètre calculé

Le paramètre « *Nbre de profils* » représente le nombre de profils ayant été utilisés pour calculer le profil moyen présent dans la base. Les paramètres « *Ecart-type U* » et « *Ecart-type V* » sont les écarts type autour de la moyenne des composantes zonales et méridiennes.

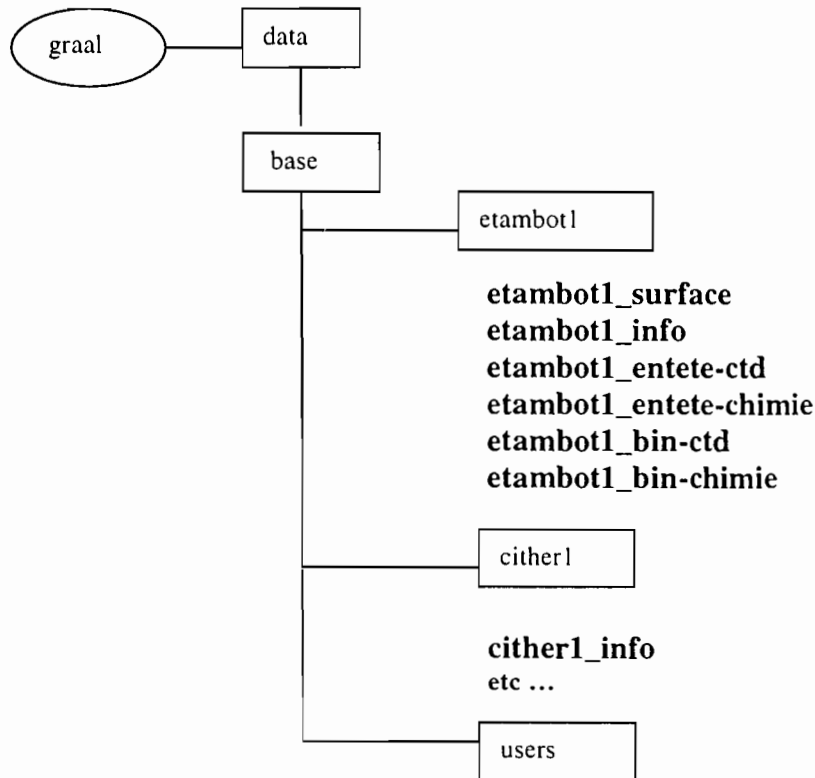
La base comprend deux types de données adcp :

Le type **SADCP** qui est le résultat de la moyenne de tous les profils obtenus pendant une station.

Le type **SINTERADCP** représente la moyenne des profils obtenus entre deux stations.

II-3 Organisation de la base

Le schéma d'organisation des différents fichiers de la base est le suivant.



La **base de données** est placée dans le répertoire : **/home/graal/data/base**

Dans le répertoire base sont inclus tous les répertoires correspondant aux différentes campagnes. A l'intérieur de chaque campagne on trouvera les fichiers utiles pour la consultation des données.

Bien entendu pour des raisons de sécurité, les fichiers contenus dans ces répertoires ne sont accessibles qu'en lecture seule.

```
drwxr-x--x  3  rchuchla  orstenv    1024 Apr  8 12:26 cither1
drwxr-x--x  3  rchuchla  orstenv     512 Mar 25 15:54 cither2
drwxr-x--x  3  rchuchla  orstenv     512 Mar 26 15:51 cither3
drwxr-x--x  3  rchuchla  orstenv    1024 Apr  8 12:28 etambot1
drwxr-x--x  3  rchuchla  orstenv    1024 Apr  8 16:36 etambot2
drwxr-x--x  3  rchuchla  orstenv    1024 Apr  8 16:36 users
```

Tous les modules nécessaires à la visualisation des données et à leur extraction se trouvent dans le répertoire **users**. (routines, bibliothèque et sous programmes).

Dans chaque répertoire relatif à une campagne se trouvent les fichiers suivants :

« Nom_de_la_Campagne »_surface	Fichiers relatifs à une campagne donnée
« Nom_de_la_Campagne »_info	
« Nom_de_la_Campagne »_entete-ctd	Fichiers concernant le type de données CTD
« Nom_de_la_Campagne »_bin-ctd	
« Nom_de_la_Campagne »_entete-chimie	Fichiers concernant le type de données CHIMIE
« Nom_de_la_Campagne »_bin-chimie	
« Nom_de_la_Campagne »_entete-ladcp	Fichiers concernant le type de données LADCP
« Nom_de_la_Campagne »_bin-ladcp	
« Nom_de_la_Campagne »_entete-sadcp	Fichiers concernant le type de données SADCP
« Nom_de_la_Campagne »_bin-sadcp	
« Nom_de_la_Campagne »_entete-sinteradcp	Fichiers concernant le type de données SINTERADCP
« Nom_de_la_Campagne »_bin-sinteradcp	

II-4 Description des différents programmes de création de la base

Après avoir décrit les différents fichiers de la base nous allons maintenant décrire les programmes qui ont été utilisés pour la création de ces fichiers.

La structure du fichier « *surface* » est identique pour toutes les campagnes. Le fichier « *info* » est créé en fonction des paramètres mesurés et calculés lors de la campagne.

Les fichiers binaires sont créés en fonction des paramètres contenus dans le fichier « *info* ».

De même les fichiers « *entête* » sont fonctions du type de mesure et permettent l'accès aux fichiers binaires correspondants.

Dans ce paragraphe nous prendrons comme exemple la campagne etambot1.

Dans le répertoire /home/graal/data/base/etambot1 se trouvent les fichiers issus des traitements constitutifs de la base. Les programmes présents dans ce répertoire ont servi à vérifier le bon déroulement de la création de la base.

/home/graal/data/base/etambot1/source

Dans ce répertoire nous avons trois sous-répertoires :

```
drwxr-x--x  2 rchuchla orstenv    512 Apr  7 16:14 include
drwxr-x--x  3 rchuchla orstenv    512 Mar 27 13:45 lib
drwxr-x--x  2 rchuchla orstenv    512 Apr  8 15:21 main_prog
```

/home/graal/data/base/etambot1/source/main_prog

Dans ce répertoire se trouvent les programmes de création de la base de données et le programme de création du fichier « *campagne_surface* » regroupant les informations de position et date.

```
-rwxr-x--x  1 rchuchla orstenv    43836 Apr  8 15:21 fic_binent-ctd
-rw-r----- 1 rchuchla orstenv     4909 Apr  8 15:43 fic_binent-ctd.c
-rwxr-x--x  1 rchuchla orstenv    40616 Apr  8 10:36 fic_chimie
-rw-r----- 1 rchuchla orstenv     5032 Apr  8 10:36 fic_chimie.c
-rwxr-x--x  1 rchuchla orstenv    19252 Mar 27 12:33 lec_base
-rwxr-x--x  1 rchuchla orstenv     2278 Mar 27 12:33 lec_base.c
-rwxr-x--x  1 rchuchla orstenv     2278 Mar 27 12:33 surface
-rwxr-x--x  1 rchuchla orstenv     2278 Mar 27 12:33 surface.c
```

Les programmes « *fic_binent-ctd* » et « *fic_chimie* » ont permis de générer les fichiers binaires de la base, le premier pour les données ctd et le second pour les données chimie.

Le programme « *surface* » est utilisé pour la création du fichier « *campagne_surface* » .

Voici présenté l'organigramme du programme fic_chimie.c

Ouverture des fichiers « *bin-chimie* » et « *entête-chimie* »

Appel de « *lec_info* » pour avoir le nombre de stations

Boucle sur le nombre de stations

Appel de la routine de lecture du fichier *.sea

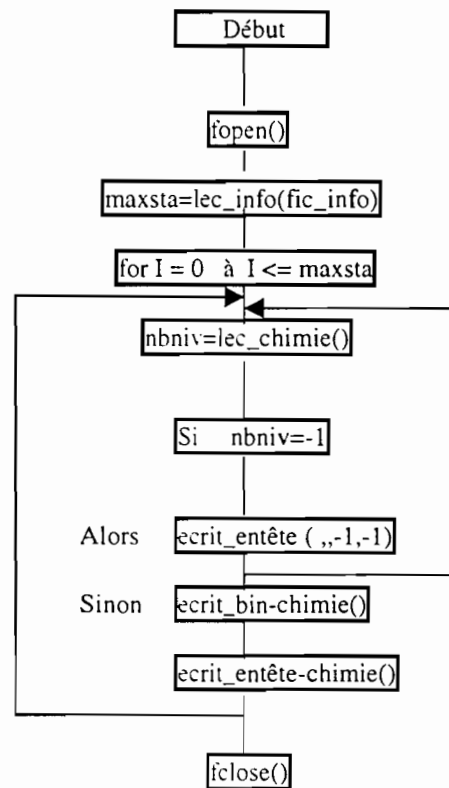
Si la station n'existe pas on incrémente le fichier « *entête* » en positionnant les variables pos et nbniv à -1

Appel de la fonction d'écriture du fichier entête

Appel de la fonction d'écriture du fichier binaire

Appel de la fonction d'écriture du fichier entête

Fin du programme.



III Description des opérations effectuées par campagne

Pour chaque campagne des programmes spécifiques ont été développés. Ces programmes se trouvent dans le répertoire de la campagne.

III-1 CITHER1

FICHER CREE	Fichier en lecture	Programmes utilisés pour la création des fichiers de la base	Commentaires
cither1_surface	cither1b.sum	Surface.c S/p : l_entsum.c ecrit_surface.c	Fichier binaire
cither1_info		Editeur de texte préféré	Fichier ascii
cither1_entete-ctd		fic_binent-ctd.c S/p : lec_info.c	Fichier binaire
cither1_bin-ctd	cith10xx.ecp cither1_surface cither1_info (xx = numéro de station)	Fic_binent-ctd.c S/p : ecrit_bin-ctd.c ecrit_entete-ctd.c lec_info.c	Fichier binaire
cither1_entete-chimie	cither1_info	fic_chimie.c S/p : Lec_info.c	Fichier binaire
cither1_bin-chimie	cither1_info cither1_surface cither1.sea	fic_chimie.c S/p : Lec_info.c Ecrit_entete-chimie.c ecrit_bin-chimie.c	Fichier binaire

III-2 CITHER2

FICHER CREE	Fichier en lecture	Programme utilisé	Commentaires
cither2_surface	*.ecp	Surface.c	Fichier binaire
cither2_info		Editeur de texte	Fichier ascii
cither2_entete-ctd	*.ecp	Fic_binent-ctd.c	Fichier binaire
cither2_entete-chimie	*.ecc	Fic_chimie.c	Fichier binaire
cither2_bin-ctd	*.ecp	Fic_binent-ctd	Fichier binaire
cither2_bin-chimie	*.ecc	Fic_chimie.c	Fichier binaire

III-3 CITHER3

FICHER CREE	Fichier en lecture	Programme utilisé	Commentaires
cither3_surface	*.ecp	Surface.c	Fichier binaire
cither3_info		Editeur de texte	Fichier ascii
cither3_entete-ctd		Fic_binent-ctd.c	Fichier binaire
cither3_bin-ctd	*.ecp	Fic_binent-ctd	Fichier binaire

III-4 ETAMBOT1

FICHER CREE	Fichier en lecture	Programme utilisé	Commentaires
etambot1_surface	Etambot1.sum	surface	Fichier binaire
etambot1_info		Editeur de texte	
etambot1_entete-ctd		Fic_binent-ctd	Fichier binaire
etambot1_entete-chimie		Fic_chimie	Fichier binaire
etambot1_bin-ctd	*.ecp	Fic_binent-ctd	Fichier binaire
etambot1_bin-chimie	*.sea	Fic_chimie	Fichier binaire
etambot1_bin-ladcp	Version-last/eta1xxx.prf	Fic_binent-ladcp	Fichier binaire
etambot1_entete-ladcp		Fic_binent-ladcp	Fichier binaire
etambot1_bin-sadcp	Station.eta1	Fic_binent-sadcp	Fichier binaire
etambot1-entete-sadcp		Fic_binent-sadcp	Fichier binaire
etambot1_bin-sinteradcp	Interstation.eta1	Fic_binent-sadcp	Fichier binaire
etambot1_entete-sinteradcp		Fic_binent-sadcp	Fichier binaire

III-5 ETAMBOT2

FICHER CREE	Fichier en lecture	Programme utilisé	Commentaires
etambot2_surface	Etambot2.sum	surface	Fichier binaire
etambot2_info		Editeur de texte	Fichier ascii
etambot2_entete-ctd		Fic_binent-ctd	Fichier binaire
etambot2_entete-chimie		Fic_chimie	Fichier binaire
etambot2_bin-ctd	*.ecp	Fic_binent-ctd	Fichier binaire
etambot2_bin-chimie	*.sea	Fic_chimie	Fichier binaire
Etambot2_bin-ladcp	Version-rap/eta2xxx.prf	Fic_binent-ladcp	Fichier binaire
Etambot2_entete-ladcp		Fic_binent-ladcp	Fichier binaire
Etambot2_bin-sadcp	Station.eta2	Fic_binent-sadcp	Fichier binaire
Etambot2-entete-sadcp		Fic_binent-sadcp	Fichier binaire

IV Accès aux données

Tous les exemples de programmes d'accès aux données ainsi que les programmes de visualisation se trouvent dans le répertoire `/home/graal/data/base/users/source/main_prog` pour les programmes écrits en langage C et dans le répertoire `/home/graal/data/base/users/fortran` pour ceux écrits en langage Fortran.

IV-1 Description des routines d'accès à la base

L'accès aux données se fait via deux routines `sub_surface()` et `sub_base()` écrites en langage C, et par deux sous-programmes `lec_base` et `lec_surface` écrits en langage Fortran77.

IV-1-1 Fonction sub_base.c

La routine `sub_base` permet d'obtenir pour une campagne donnée *le profil d'un paramètre*.

```
* Fonction :      sub_base
*
* Description:   Accès à la base de données et retour d'un
*                  paramètre d'une station
*
* Bibliothèque utilisée : /home/graal/data/base/users/source/lib/calcul.a
*
* Appel de la fonction:
*
* int sub_base ( char *campagne , int type , int para , int sta , **selection ) ;
*
* Paramètres en entrée:
*
*          campagne      Nom de la campagne
*          type          Type de mesure
*          para          Code WHP du paramètre à extraire
*          sta           Numéro de station
*
* Paramètre en sortie:
*
*          selection     tableau de données ( retourne un réel pour les paramètres mesurés et
*                  calculés et un entier pour les codes de qualité)
*
* Retour:
*
*          nbniv         Nombre de niveaux de la station
*
* Code erreurs :
*
*          -1           Station absente
*          -2           Paramètre absent
*          -3           Campagne absente
*          -4           Type erroné
*
```


IV-1-2 fonction sub_surface.c

La routine sub_surface permet d'obtenir pour une campagne donnée *la date, la position et le fond d'une station*.

```
* Deuxième Fonction appelée :    sub_surface
*
* Description:    Accès au fichier campagne_surface
*
* Bibliothèque utilisée : /home/graal/data/base/users/source/lib/calcul.a
*
* Appel de la fonction:
*
* sub_surface ( campagne , sta , tab ) ;
*
* paramètres en entrée:
*
*      campagne    Nom de la campagne
*      sta         Numéro de station
*
* paramètre en sortie:  tab          date, position et fond de la station
*
*                          mois jour an heure minute seconde latitude longitude fond (m)
*                          12 29 1993 21 50    -99    -4.500 10.350 2503
*
* retour:            -1              Si la station n'existe pas
*
```

IV-1-3 Sous-Programme lec_base.f

Le sous-programme lec_base permet d'obtenir pour une campagne donnée *le profil d'un paramètre*.

```
* Sous-programme appelé :    lec_base
*
* Fonction :    lec_base
*
* Description:    Accès à la base de données et retour d'un
*                  paramètre d'une station
*
* Bibliothèque utilisée : /home/graal/data/base/users/fortran/baselib.a
*
* Appel de la fonction:
*
* lec_base ( campagne , itype , ista , ipara , xtab , nbniv , ierr ) ;
*
* Paramètres en entrée:
*
*      campagne    Nom de la campagne
*      itype       Type de mesure
*      ista        Numéro de station
*      ipara       Code WHP du paramètre à extraire
*
```

```

* Sous-programme lec_base.f (suite)
*
* Paramètre en sortie:
*
*          xtab          tableau de données ( retourne un réel pour les paramètres mesurés et
*                       calculés et un entier pour les codes de qualité)
*          nbniv         nombre de niveaux de la station
*          ierr          code d 'erreur
*
*
* Code erreurs :
*          -1           Station absente
*          -2           Paramètre absent
*          -3           Campagne absente
*          -4           Type erroné
*
*
*

```

IV-1-4 Sous-Programme lec_surface.f

Le sous-programme **lec_surface** permet d'obtenir pour une campagne donnée *la date, la position et le fond d'une station.*

```

* Sous-programme appelé :      lec_surface
*
* Description:      Accès au fichier campagne_surface
*
* Bibliothèque utilisée : /home/graal/data/base/users/fortran/baselib.a
*
* Appel de la fonction:
*
* lec_surface ( campagne , ista , itype , ipara , tab , ierr) ;
*
*
* paramètres en entrée:
*
*          campagne     Nom de la campagne
*          ista         Numéro de station
*          itype        Type de mesure
*          ipara        Code WHP du paramètre à extraire
*
*
* paramètre en sortie:
*
*          tab          date, position et fond de la station
*
*                       mois jour an heure minute seconde latitude longitude fond (m)
*                       12 29 1993 21 50 -99 -4.500 10.350 2503
*
*          ierr         Code erreur
*
*                       -1           Si la station n'existe pas
*
*

```

IV-2 Compilation et édition de liens

IV-2-1 Programme C

Pour la compilation d'un programme il est nécessaire de faire appel à la bibliothèque « **calcul.a** » qui se trouve dans le répertoire `/home/graal/data/base/users/source/lib` .

Les fichiers « **include** » se trouvent dans le répertoire `/home/graal/data/base/users/source/include` . On doit impérativement faire les déclarations suivantes en début de programme :

<code>#include <stdio.h></code>	fichier propre au langage C
<code>#include <string.h></code>	fichier propre au langage C
<code>#include "structure.h"</code>	fichier contenant la définition des structures de données utilisées
<code>#include "inout.h"</code>	fichier contenant toutes les routines utilisées pour la création de la base
<code>#include "chemin.h"</code>	fichier contenant la liste des différents chemins d'accès aux fichiers
<code>#include "constante.h"</code>	fichier contenant toutes les constantes utilisées dans les différents programmes

Pour la compilation et l'édition de liens un fichier « **makefile** » est situé dans le répertoire `/home/graal/data/base/users/source/main_prog` .

On tape alors la commande suivante : « **make -f makefile nom_programme** ». Si la compilation est réussie on peut exécuter le programme en tapant « **nom_programme** ».

IV-2-2 Programme Fortran 77

Pour la compilation d'un programme fortran il est nécessaire de faire appel à la bibliothèque « **baselib.a** » qui se trouve dans le répertoire `/home/graal/data/base/users/fortran` .

Pour la compilation et l'édition de liens taper la commande suivante :

« **f77 Mon_prog.f baselib.a -o Mon_prog** » .

Si la compilation est réussie on peut exécuter le programme en tapant « **Mon_prog** ».

IV-3 Exemple d'un programme d'extraction de données.

IV-3-1 Programme écrit en Langage C

```
* Programme:   lec_base.c      (Brest janvier 1998)
*
*
* Bibliothèque utilisée:      /home/graal/data/base/users/source/lib/calcul.a
*
* Fonctions appelées :        sub_base, sub_surface
*
*
* → !!! Attention :           libérer l'espace mémoire alloué à la fin du programme . Utiliser free ( tab ).
*-----
/*
*
* Programme: lec_base.c      (Brest janvier 1998)
#include <stdio.h>
#include <string.h>
#include "structure.h"
#include "inout.h"
#include "chemin.h"
#include "constante.h"

void main ()

{
char   campagne[255];
int    sta, i, f, para, type , surf;
float *sel , tab[10];

/*-----
INITIALISATION DES VARIABLES D'ENTREE
-----
*/

printf(" entrer le nom de la campagne ?? \n"); /* ( par exemple citherl ) */
scanf("%s",campagne);

printf(" entrer le type de mesure 1=ctd 2= .. ?? \n");
scanf("%d",&type);

printf(" entrer le numéro de la station ?? \n");
scanf("%d",&sta);

printf(" entrer le ncode du paramètre à extraire ( whp) ?? \n");
scanf("%d",&para);

/*-----
Appel de la fonction sub_base
-----
*/

f = sub_base ( campagne , type , sta , para, &sel );
```

```

/*-----
                        Appel de la fonction sub_surface
-----*/

surf= sub_surface ( campagne , sta , tab );

/*-----
                        Impression du paramètre sélectionné
-----*/

for ( i=0 ; i<f ; i++ )

    printf("s=%3f\n",sel[i]);

/*-----
                        Impression de la date, position et fond de la station
-----*/

printf(" station= %f mois= %f jour= %f an= %f heure= %f min= %f sec= %f lat= %f long= %f fond= %f\n",
tab[0], tab[1],tab[2],tab[3],tab[4],tab[5],tab[6],tab[7],tab[8],tab[9] );

/*-----
                        désallocation mémoire
-----*/

free(sel);

}

```

IV-3-2 Programme écrit en Fortran 77

```

c                               Brest Septembre 1998
c
c
c   Routine de lecture des fichiers binaires
c   Base de données réalisée à partir des campagnes
c   etambot ( 1 et 2 ) cither ( 1, 2 et 3 )
c
c   routine de lecture des fichiers "surface" lec_surface()
c
c
c   routine de lecture des données "binaires" lec_base()
c
c
c
c
c   real*4 xtab(4000) , stab(10)
c   integer sta , para , type , ipos ,ierr
c   character*80 name_camp

```

```

c-----
c----- Initialisations -----
c-----

      print*,' nom de la campagne ?? ex: cither1 '
      read* ,name_camp

      print*,' numero de station ..'
      read* ,num_sta

1      print*,' type de mesure ?? (ctd= 1, chimie= 2, ladc= 3,sadc=4,
          sinteradc= 5 )'
      read* ,type

      print*,' numero du para WHP .. ( voir tableau des parametres WHP )'
      read* ,para

c-----
c----- Lecture du fichier surface -----
c-----

      call lec_surface ( name_camp , num_sta , type , para , stab , ipos )

      if (ipos.eq.-3) then
          print* , '****La campagne n existe pas !!'
          goto 9998
      endif

      if(ipos.eq.-1) then
          print* , '**** la station n existe pas !!'
          goto 9998
      endif
      if( ipos.eq.-2 ) then
          print* , '**** le parametre n existe pas !!'
          goto 9998
      endif
      if( ipos.eq.-4 ) then
          print* , '**** le type est errone !!'
          goto 9998
      endif

c-----
c----- Appel de la routine lec_base -----
c-----

      call lec_base( name_camp , type , num_sta , para , xtab , nbniv , ierr )

      if ( ierr.eq.-3)      then
          print*,' Fichier absent !!'
          goto 9998
      endif

```

```

if( ierr.eq.-1.or.nbniv.eq.-1) then
    print* , ' La station',sta,' n existe pas !!'
    goto 9998
endif

if( ierr.eq.-2) then
    print* , ' Le parametre WHP ',numpara,' n existe pas !!'
    goto 9998
endif

if( ierr.eq.-5) then
    print* , ' fin de fichier depassee !!!'
    goto 9998
endif

if ( ierr.eq.-4 ) then
    print* , 'type erronee '
    goto 9998
endif

```

```

c-----
c-----IMPRESSION DES RESULTATS-----
c-----

```

```

    print* , ' station =',int(stab(1)) , ' jour =',int(stab(3)),
1   mois =',int(stab(2))
1   , ' annee = ' , int(stab(4))
    print* , ' heure=', int(stab(5)), ' minute =',int(stab(6))
    print* , ' latitude= ' , stab(8) , ' longitude = ' , stab(9)
    print* , ' fond = ' , int(stab(10))
    print*
    print* , 'Le Nombre de niveaux est: ',nbniv
    print*
    print* , 'xtab ..',(xtab(ik),ik=1,nbniv)

```

```

9998 stop
end

```

```

c-----
c----- Fin du programme Principal -----
c-----

```

V interface graphique (Matlab)

Les programmes de visualisation des données ont été réalisés à partir du logiciel MATLAB. Tous les programmes décrits sont situés dans le répertoire : `/home/graal/data/base/users/source/main_prog`.

L'accès aux données se fait grâce aux routines suivantes: « `lec_base.m` », « `sub_surface.m` » et « `sub_info.m` ».

V-1 Description des routines d'accès à la base de données

V-1-1 Fonction `lec_base.m`

```
% Sous-programme : lec_base
%
% en entrée : nom de la campagne, type de mesure, et numéro de station
%
% en sortie la fonction renvoie :
%
% un tableau b contenant tous les paramètres de la station définis dans le fichier « campagne_info »
% le nombre de niveaux ( nbniv ) de celle-ci.
% si nbniv = -1 station inexistante.
%
%
% Appel de la fonction :
%
% function[ b , nbniv ] = lec_base ( name, type, num_sta )
%
% retourne tous les paramètres d'une station
% lecture et traces fichiers binaires (cither1 etc ..)
% lecture campagne_entete-ctd et campagne_bin-ctd
%
%
% codes erreurs:
% -----
%
% La station n existe pas            nbniv=-1
% nom de fichier erroné            nbniv=-3
% type erroné                        nbniv=-4
%
% -----
%
```


V-1-2 Fonction sub_surface.m

```
% Sous-programme : sub_surface
%
% Description : lecture des fichiers « campagne_surface »
%
% en entrée : nom de la campagne et numéro de station
%
% en sortie la fonction renvoie :
%
% un tableau tab contenant les paramètres suivants :
%
% mois jour année heure minute seconde latitude longitude fond
%
%
% Appel de la fonction :
%
% fonction[ tab ] = sub_surface ( name, num_sta )
%
%
%
% codes erreurs:
% -----
%
% La station n existe pas tab(1) = -1
%
% -----
%
```

V-1-3 Fonction sub_info.m

```
% Sous-programme : sub_info
%
% Description : lecture des fichiers « campagne_info »
%
% en entrée : nom de la campagne et type de mesure
%
% en sortie la fonction renvoie :
%
% - Le nombre de station de la campagne
% - Le nombre de paramètre
% - Un tableau contenant la liste des paramètres
% - la taille d'un enregistrement dans le fichier binaire
%
%
% Appel de la fonction :
%
% fonction[ maxsta , nbpar , par , taille ] = sub_info ( name, type )
%
%
%
```

V-1-4 Exemple d'un programme de tracés

```
% Exemple d'un programme de tracés « visu_base.m » ( température en fonction de la pression )  
% utilisant la routine % « lec_base.m » .
```

```
%  
% Langage utilisé : matlab
```

```
%  
% Sous-programmes :
```

```
%          lec_base  
%          sub_surface  
%          sub_info  
%
```

```
%-----  
%                               initialisations  
%-----
```

```
name= input('entrer le nom de la campagne cither1 ne pas oublier les cotes !!');
```

```
type= input('type de mesure 1=ctd 2=chimie 3=ladcp 4=sadcpstation 5= sadcpinter ');
```

```
num_sta =input( ' entrer le numero de la station a visualiser :');
```

```
%-----  
%                               Appel de la routine de lecture des données  
%-----
```

```
[ b , nbniv ] = lec_base ( name , type , num_sta ) ;
```

```
%-----  
%                               Tracé pression, température ( code 3 et 4 )  
%-----
```

```
if ((nbniv) > 0 )
```

```
    plot(b(:,4).b(:,3));
```

```
    set(gca,'ydir','reverse');
```

```
end
```

V-1-5 Exemple d'utilisation de la fonction sub_surface

```
%-----  
%           Exemple d'un programme de tracés de position de station  
%-----  
  
name=input('entrer le nom de la campagne ne pas oublier les cotes !!');  
  
limite_sud = input('entrer la limite sud ');  
  
limite_nord=input('entrer la limite nord ');  
  
%-----  
%           Appel de la routine sub_info  
%           Récupération du nombre de stations  
%-----  
  
    [maxsta , var2 , var3 , var4 ] =sub_info ( name , 1 )  
  
inter=NaN*ones(maxsta,2);  
  
j=1;  
  
for i=0 : maxsta  
  
%-----  
%           Lecture du fichier surface  
%-----  
  
    [tab]=sub_surface(name,num_sta) ;  
  
    if ( ( tab(8) <= limite_nord ) | ( tab(8) <= limite_sud ) )  
        inter(j,1)=tab(8);  
        inter(j,2)=tab(9);  
        j=j+1;  
    end  
  
end  
  
%-----  
%           Tracé des positions sélectionnées  
%-----  
  
plot(inter(:,2),inter(:,1), 'r*');  
  
set(gca,'xlim',[-60 15]);  
set(gca,'ylim',[-10 10]);  
  
end    %Fin du programme
```

VI Conclusion

A ce jour les données de cinq campagnes forment cette base :

- La campagne CITHER1
-
- La campagne CITHER2
-
- La campagne CITHER3
-
- La campagne ETAMBOT1
-
- La campagne ETAMBOT2

Les données météorologiques seront prochainement incluses ainsi que d'autres campagnes océanographiques.

Ce document sera réactualisé au fur et à mesure de l'évolution de la base et les utilisateurs en seront informés.

VII Annexe

VII-1 Exemple de compilation et d'édition de liens

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

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

# Directory des fichiers include
INCLUDE = /home/graal/data/base/users/source/include

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

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

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

# bibliotheque
LDFLAG2 = /home/graal/data/base/users/source/lib/calcul.a

# bibliotheque
# LDFLAG3 = /home/graal/data/base/users/source/lib

# bibliotheque
#LDFLAG4 = /home/graal/data/base/users/source/lib

# LDFLAGS = $(LDFLAG1) $(LDFLAG2) $(LDFLAG3) $(LDFLAG4)
LDFLAGS = $(LDFLAG1) $(LDFLAG2)

# COMPILATION ET EDITION
#
.c:
    $(CC) $(CFLAGS) -I$(INCLUDE) $< $(LDFLAGS) -o $@
#
```

VII-2 Liste des numéros des paramètres

```
/*-----  
Nom: PARAWHP.H           Brest décembre 1997  
Type: Include  
Description: de 1 à 43    paramètres mesurés code WHP  
                  de 45 à 50 paramètres ladcp et sadcp  
                  de 80 à 92 paramètres calculés  
                  de 101 à 143 code erreurs des paramètres mesurés  
-----*/
```

```
*/
```

```
#if !defined PARAWHP_DEFINED  
#define PARAWHP_DEFINED
```

```
#define BTLNBR      1    /* No. Echantillon */  
#define CTDRAW     2    /* dbar pression brute */  
#define CTDPRS     3    /* dbar pression recalculée */  
#define CTDTMP     4    /* degre-C temperature */  
#define CTDSAL     5    /* pss-78 salinite */  
#define CTDOXY     6    /* Umol/kg oxygene */  
#define SALNTY     7    /* pss-78 salinite */  
#define OXYGEN     8    /* Umol/kg oxyene */  
#define SILCAT     9    /* Umol/kg silicate */  
#define NITRAT    10    /* Umol/kg nitrate */  
#define NITRIT    11    /* Umol/kg nitrite */  
#define PHSPHT    12    /* Umol/kg phosphate */  
#define CFC_11    13    /* Pmol/kg freon11 */  
#define CFC_12    14    /* Pmol/kg freon12 */  
#define REVPRS    15    /* dbar */  
#define REVTMP    16    /* degre-C */  
#define TRITUM    17    /* TU tritium */  
#define HELIUM    18    /* Nmol/kg helium */  
#define DELHE3    19    /* percent delta helium */  
#define DELC14    20    /* /MILLE carbone 14 */  
#define DELC13    21    /* /MILLE carbone 13 */  
#define KR_85     22    /* dpm/mg krypton 85 */  
#define ARGON     23    /* Umol/kg argon */  
#define AR_39     24    /* pctmod argon 39 */  
#define NEON      25    /* Nmol/kg neon */  
#define RA_228    26    /* dm/.1mg radium */  
#define RA_226    27    /* dm/.1mg radium */  
#define O18O16    28    /* /mille 018/016 ratio */
```

```

#define SR_90      29  /* dm/.1mg strontium 90 */
#define CS_137    30  /* dm/.1mg cesium 137 */
#define TCARBON  31  /* Umol/kg carbone total */
#define ALKALI    32  /* Umol/kg alcalinite totale */
#define FCO2      33  /* uatm fugacite de co2 */
#define PH        34  /* ph */
#define CFC_113   35  /* cfc113 */
#define CCL4      36  /* ccl4 36 */
#define IODINE    37  /* iodine */
#define NH4       38  /* nh4 */
#define METHAN    39  /* Nmol/kg methane */
#define AZOTE     40  /* Umol/kg azote */
#define OXYNIT   41  /* Nmol/kg oxyde nitreux */
#define CHLA     42  /* Ug/l chlorophylle A */
#define PHAEO    43  /* Ug/l phaeophytine */
#define PROF     45  /* profondeur en metres */
#define COMPU    46  /* composante zonale de la vitesse ladcp */
#define COMPV    47  /* composante meridienne de la vitesse ladcp */
#define NBOBS    48  /* nbre de ping ladcp et nbre de profil sadcp */
#define SIGU     49  /* sigma U */
#define SIGV     50  /* sigma V */
#define Z        80  /* m profondeur en mètres*/
#define TEMPOT   81  /* degré-C température potentielle*/
#define SIGT     82  /* Kg/M**3 sigma-t*/
#define SIGTH    83  /* Kg/M**3 sigma-theta*/
#define SIG1     84  /* Kg/M**3 sigma-1*/
#define SIG15    85  /* Kg/M**3 sigma-1.5*/
#define SIG2     86  /* Kg/M**3 sigma-2*/
#define SIG3     87  /* Kg/M**3 sigma-3*/
#define SIG4     88  /* Kg/M**3 sigma-4*/
#define BVFR20   89  /* cph fréquence de Brunt Vaissala */
#define DSDP     90  /* gradient vertical de salinité*/
#define DTDP     91  /* gradient vertical de température in situ*/
#define HDYN     92  /* m.dyn Hauteur Dynamique */
#define C_BTLNBR 101  /* Code_qualité para 1 */
#define C_CTDRAW 102  /* Code_qualité para 2 */
#define C_CTDPRES 103  /* Code_qualité para 3 */
#define C_CTDTMP 104  /* Code_qualité para 4 */
#define C_CTDSAL 105  /* Code_qualité para 5 */
#define C_CTDOXY 106  /* Code_qualité para 6 */
#define C_SALNTY 107  /* Code_qualité para 7 */
#define C_OXYGEN 108  /* Code_qualité para 8 */
#define C_SILCAT 109  /* Code_qualité para 9 */

```

```

#define C_NITRAT      110 /* Code_qualité para 10 */
#define C_NITRIT      111 /* Code_qualité para 11*/
#define C_PHSPHT      112 /* Code_qualité para 12 */
#define C_CFC_11      113 /* Code_qualité para 13 */
#define C_CFC_12      114 /* Code_qualité para 14 */
#define C_REVPRS      115 /* Code_qualité para 15 */
#define C_REVTMP      116 /* Code_qualité para 16 */
#define C_TRITUM      117 /* Code_qualité para 17 */
#define C_HELIUM      118 /* Code_qualité para 18 */
#define C_DELHE3      119 /* Code_qualité para 19 */
#define C_DELC14      120 /* Code_qualité para 20 */
#define C_DELC13      121 /* Code_qualité para 21*/
#define C_KR_85       122 /* Code_qualité para 22 */
#define C_ARGON       123 /* Code_qualité para 23 */
#define C_AR_39       124 /* Code_qualité para 24 */
#define C_NEON        125 /* Code_qualité para 25 */
#define C_RA_228      126 /* Code_qualité para 26 */
#define C_RA_226      127 /* Code_qualité para 27 */
#define C_O18016      128 /* Code_qualité para 28 */
#define C_SR_90       129 /* Code_qualité para 29 */
#define C_CS_137      130 /* Code_qualité para 30 */
#define C_TCARBON     131 /* Code_qualité para 31 */
#define C_ALKALI      132 /* Code_qualité para 32 */
#define C_FCO2        133 /* Code_qualité para 33 */
#define C_PH          134 /* Code_qualité para 34 */
#define C_CFC_113     135 /* Code_qualité para 35 */
#define C_CCL4        136 /* Code_qualité para 36 */
#define C_IODINE      137 /* Code_qualité para 37 */
#define C_NH4         138 /* Code_qualité para 38 */
#define C_METHAN      139 /* Code_qualité para 39 */
#define C_AZOTE       140 /* Code_qualité para 40 */
#define C_OXYNIT      141 /* Code_qualité para 41 */
#define C_CHLA        142 /* Code_qualité para 42 */
#define C_PHAEO       143 /* Code_qualité para 43 */

```

```

#endif

```


VII-3 Codes de qualité utilisés

A chaque donnée est associé un code de qualité. Ce code suit les recommandations du Manuel d'Opérations WOCE (WHP Office Report WHPO 90-1, July 1991).

Code=2 : RAS

Code=3 : Résultat contestable, incertain

Code=4 : mauvais résultat de mesure

Code=5 : absence de résultats (problème d'échantillonnage : perte, contamination)

Code=9 : échantillon non soutiré pour la mesure de ce paramètre

VII-3 Liste des fichiers de la base (juillet 1998)

CITHER1	CITHER2	CITHER3	ETAMBOT1	ETAMBOT2
*_info	*_info	*_info	*_info	*_info
*_surface	*_surface	*_surface	*_surface	*_surface
*_bin-ctd	*_bin-ctd	*_bin-ctd	*_bin-ctd	*_bin-ctd
*_entete-ctd	*_entete-ctd	*_entete-ctd	*_entete-ctd	*_entete-ctd
*_bin-chimie	*_bin-chimie		*_bin-chimie	*_bin-chimie
*_entete-chimie	*_entete-chimie		*_entete-chimie	*_entete-chimie
			*_bin-ladcp	*_bin-ladcp
			*_entete-ladcp	*_entete-ladcp
*_bin-sadcp			*_bin-sadcp	*_bin-sadcp
*_entete-sadcp			*_entete-sadcp	*_entete-sadcp
*_bin-sinteradcp			*_bin-sinteradcp	
*_entete-sinteradcp			*_entete-sinteradcp	
*='cither1'	*='cither2'	*='cither3'	*='etambot1'	*='etambot2'