RAPPORT DE STAGE EFFECTUE AU CNRM/GMAP TOULOUSE

Du 22 Décembre 2002 au 22 Février 2003

CANARI utilisant le cycle 25 et ODB sur la machine IBM RS/6000

Elaboration de scripts pour l'utilisation des IDFI dans ALADIN

Travaux pour ALBACHIR Nord Afrique, la BDM et le prétraitement

Rashyd ZAABOUL

Service Prévision Numérique

Direction de la Météorologie Nationale (CASABLANCA/MAROC)

Superviseurs :

Claude FISCHER

Elisabeth GERARD

Introduction

Le travail que j'ai effectué le long de mon séjour à Toulouse du 22 Décembre 2002 au 22 Février 2003 a comporté deux grandes parties. La première concernait le portage du code ODB du cycle 25 et la réussite du tournage de l'analyse de surface CANARI sur la machine IBM/RS6000 utilisant ODB comme système de gestion des observations. La seconde étape était consacrée à la participation dans l'élaboration des scripts pour l'utilisation des initialisations par les filtres digitaux DFI. J'ai travaillé en parallèle sur la mise à jour des programmes du prétraitement installés à CASABLANCA et aussi à la création de fichiers climatologiques et aux tests pour la nouvelle configuration du modèle ALBACHIR Nord Afrique.

1. Portage du code ODB du cycle 25

Le travail du portage du code ODB sous le cycle cy25t1 n'a pas nécessité beaucoup de modifications car le code était déjà porté au centre européen sur la machine IBM/RS6000. Sans l'aide de *Radi Ajjaji* ce travail m'aurait pris beaucoup de temps vu la charge de la ligne CASABLANCA-TOULOUSE. Je me suis occupé du debuging et *Radi* m'a aidé surtout pour résoudre le problème des externals. Maintenant toutes les applications à base d'ODB tournent sur la machine IBM à CASABLANCA. L'analyse des champs de surface est élaborée avec CANARI. Les modifications apportées au code ODB du cycle cy25t1 sont expliquées en annexe 1.

Les applications suivantes ont été testées par Radi et sont opérationnelles maintenant à CASABLANCA :

BATODB

MANDAODB

SHUFFLE (pour tout type de transformation)

LAMFLAG

SCREENING

MINIMISATION

CANARI (pour l'analyse de surface et d'altitude)

Je n'ai pas fais de test scientifique pour comparer les résultats de l'analyse CANARI du cycle cy25t1 et ceux de l'opérationnel. L'objectif du stage était juste le portage du code et le temps ne permet pas de faire de telles expériences.

2. Scripts d'assimilation 3DVAR, Blending et BlendVar.

Durant la deuxième partie de mon stage j'ai travaillé en collaboration avec *Pascale Riber* et *Claude Fischer* sur l'outil de génération de scripts pour des expériences d'assimilation 3DVAR, du Blending ou du BlendVar dans ALADIN. Nous avons apporté des modifications aux scripts originaux pour tenir compte principalement du cas d'un blendvar et aussi de l'application des filtres digitaux lors de la production. L'initialisation se faisait par un filtrage digitale simple appliqué à l'analyse. Les expériences ont montrées que ce type d'initialisation est un peu fort et a pour conséquence la destruction des incréments d'analyse. Pour remédier à cet inconvénient on a appliqué un nouveau mode d'initialisation en ajoutant des incréments filtrés à l'analyse. Cette initialisation se fait en deux étapes :

La première étape consiste à calculer un biais en partant d'une prévision 6 heures (un guess) en mettant la clé LBIAS=.TRUE. dans la namelist NAMINI. Ceci permet de récupérer dans le fichier ICMSH${CNMEXP}BIAS le biais

G – F(G). F étant le filtre digitale choisi et G désigne le guess.

La deuxième étape consiste à ajouter ce biais à l'analyse filtrée. Ce qui revient à calculer F(A) + G – F(G) en mettant la clé LINCR=.TRUE. dans la namelist NAMINI. A étant l'analyse.

Les scripts ont été testés en lançant plusieurs expériences. Dans chaque expérience on a choisi de faire soit une analyse 3DVAR, soit un blending ou un blendvar accompagné d'une prévision 24 heures. Pour les expériences classiques permettant de faire un cycle d'assimilation 3DVAR ou un blending, nous n'avons pas eu de surprise. Les cycles tournent normalement. Pour le cycle blendvar on a apporté des modifications aux scripts pour tenir compte de l'alternance entre le blending normale et l'assimilation de données. La version finale de ces scripts se trouve sous kami à l'emplacement suivant : ~mrpe729/cy25/analyse/src

On a ajouté un script qui permet de réaliser une initialisation à base de filtres digitaux comme expliqué ci-dessus. Le script s'appelle *aldincr* et se trouve au même emplacement que les autres scripts. J'ai lancé un cycle d'assimilation de 24 heures en utilisant ce mode d'initialisation pour tester si le script et son insertion dans le package est informatiquement correcte. Pour le réglage du filtre (fenêtre et fréquence de coupure) il faut lancer un cycle d'assimilation de 15 jours avec une prévision de 48 heures. Le temps de mon stage ne m'a pas permis d'aller jusqu'au là. Je me suis contenté de faire les tests informatiques.

Fichiers clim. et coupleurs pour ALBACHIR Nord Afrique

Durant mon séjour à Toulouse j'ai été sollicité par *Radi* pour créer et tester des fichiers climatologiques ainsi que des coupleurs pour le modèle ALBACHIR Nord Afrique et aussi pour le modèle ALBACHIR qui couvre le territoire marocain. Les caractéristiques du domaine ALBACHIR Nord Afrique sur lequel la DMN a commencé dès le 18/02/2003 à tourner une chaîne en double sont les suivantes :

Latitude Nord : 45.192449202570210

Latitude Sud : -3.678988513812508

Longitude Est : 62.641606799769940

Longitude Ouest : 322.828781350508600

La grille correspondante a les caractéristiques suivantes (la zone d'extension comprise) :

Nombre de points sur l'axe zonale : 240

Nombre de points sur l'axe méridional : 160

Nombre de niveaux verticaux : 41

Soit une résolution de 42225.19405092247 m.

J'ai préparé les fichiers climatologiques du domaine ALBACHIR Nord Afrique et les namelists pour produire des fichiers de couplage sous le cycle cy25t1 à partir d'ARPEGE. Ces fichiers se trouvent sous delage à l'emplacement suivant :
/cnrm2_a/mrpe/mrpe729/e923/albach_na_new

Les scripts et les namelists se trouvent sous kami : ~mrpe729/cy25/e923 et ~mrpe729/cy25/e927

La création de ce nouveau domaine va nous permettre de tourner un modèle sur l'ancien domaine d'ALBACHIR mais avec une résolution plus fine. Nous avons commencé par tester ALBACHIR 9 km qui s'est avéré gourmand en temps de calcule (d'après *Radi*). J'ai préparé des fichiers climatologiques sur le même domaine mais à une résolution de 11 km et de 13 km pour les ramener à CASABLANCA afin de faire les tests préliminaires en couplant le modèle ALBACHIR avec ALBACHIR Nord Afrique. Ces tests pourront nous amener à choisir la résolution adéquate du modèle ALBACHIR. Les fichiers préparés se trouvent sous delage aux emplacements suivants :

/cnrm2_a/mrpe/mrpe729/e923/albach_11km_new /cnrm2_a/mrpe/mrpe729/e923/albach_13km_new

Les scripts et les namelists utilisés se trouvent sous kami ~mrpe729/cy25/e923

J'ai fais des tests en créant des fichiers coupleurs pour la résolution 11 km et 13 km en partant des fichiers d'ALBACHIR Nord Afrique (configuration ee927)

BDM et prétraitement

Durant ces deux mois j'ai travaillé avec Gilles Geley, Dominique Paulais et Philippe Caille sur la mise à jour des programmes du prétraitement que nous avons à CASABLANCA. Les sources installés datent, pour la plupart d'entre eux, de 1997 ou 1998 et ne tiennent pas compte des dernières modifications faites à Météo France surtout pour le décodage des SHIP, des SYNOP et des SYNOR. J'ai pris la dernière version de ces codes pour mettre à jour nos programmes tout en gardant les modifications qui ont été faites à CASABLANCA surtout pour l'introduction de certains paramètres dans les tables pour des besoins locaux.

Annexe 1 : Modifications du code odb25t1

Dans les fichiers qui suivent on a ajouté #include "RS6K.h" pour remédier aux problèmes des externals. Le contenu le l'include RS6K.h est donné en annexe 2.

obd/aux/binio.c

obd/aux/kmg.c

obd/aux/pseudo_main.c

obd/include/cmaio.h

obd/include/codb.h

obd/include/hdr.h

obd/include/odb.h

obd/include/pcma_extern.h

obd/include/svipc.h

Le compilateur C de la machine IBM confond la variable n et le retour à la ligne \n et aussi la variable s et le format d'écriture d'une chaîne de caractères %s. Nous avons modifié les programmes suivants pour remédier à ce problème.

obd/compiler/genc.c

```
#ifdef RS6K

#define NL(i)  { int l; for(l=0; l<(i); l++) fputc('\n',cfp); }

#define TAB(i) { int l; for(l=0; l<(i); l++) fputs("  ",cfp); }

#define LB(i)  { TAB(1); fputs("{\"",cfp); TAB(i); }

#define RB(i)  { int l, lmax=(i); for(l=1; l<=lmax; l++) { \

#endif
```

obd/compiler/lex.l

```
#ifdef RS6K

#define SRC(r) { if (fpsrc) { fprintf(fpsrc,"%s%s ",r,yytext); } }

#endif
```

obd/compiler/odb98.c

```
#ifdef RS6K
#define SYSTEM_NAME "(RS6000/SP/IBM)"
```

obd/include/codb.h

```
#ifdef RS6K
#define GEN_GET       codb_dget
#define GEN_PUT       codb_dput
#endif
```

obd/include/timerdefs.h

```
    &(tbytes / (max(1E-20,timediff(t1,t2)) * 1048576D0 ))
    &max(1E-20,timediff(t1,t2)) ) * 100D0
```

Dans errtrap.c on a utilisé le xl__trbk au lieu de errtra pour faire un traceback dans le cas d'une erreur.

obd/lib/errtrap.c

```
#ifdef RS6K
    xl__trbk();

    sleep(5);
#endif
```

Annexe 2 : RS6K.h

```
#ifdef RS6K

/*odb.h*/
#define codb_trace_  codb_trace
#define codb_trace_end_ codb_trace_end
#define codb_abort_func_ codb_abort_func
```

```
#define codb_register_abort_func_ codb_register_abort_func

#define codb_init_ codb_init

#define codb_trace_init_ codb_trace_init

#define codb_set_signals_ codb_set_signals

#define odb_inlist_ odb_inlist

#define odb_notinlist_ odb_notinlist

#define odb_debug_print_ odb_debug_print

#define cma_readb_ cma_readb

#define codb_twindow_ codb_twindow

#define codb_tdiff_ codb_tdiff

#define codb_trace_ codb_trace

#define codb_filesize_ codb_filesize

#define codb_datetime_ codb_datetime

#define codb_abort_func_ codb_abort_func

#define cdummy_load_ cdummy_load

#define codb_write_metadata_ codb_write_metadata

#define codb_twindow_ codb_twindow

#define codb_tdiff_ codb_tdiff

#define codb_getnames_ codb_getnames

#define codb_getprecision_ codb_getprecision

#define codb_setval_ codb_setval

#define codb_getval_ codb_getval

#define codb_hash_init_ codb_hash_init

#define codb_vechash_ codb_vechash

#define codb_ui_unique_ codb_ui_unique

#define codb_d_unique_ codb_d_unique

#define codb_r_unique_ codb_r_unique

#define codb_tablesize_ codb_tablesize

#define codb_write_metadata2_ codb_write_metadata2

#define codb_write_metadata3_ codb_write_metadata3

#define codb_close_ codb_close
```

```
#define codb_store_ codb_store

#define codb_load_ codb_load

#define codb_makeview_ codb_makeview

#define codb_linkview_ codb_linkview

#define codb_save_peinfo_ codb_save_peinfo

#define codb_mp_select_ codb_mp_select

#define codb_select_ codb_select

#define codb_cancel_ codb_cancel

#define codb_swapout_ codb_swapout

#define codb_getsize_ codb_getsize

#define codb_get_view_info_ codb_get_view_info

#define codb_restore_peinfo_ codb_restore_peinfo

#define codb_iget_ codb_iget

#define codb_get_rowvec_ codb_get_rowvec

#define codb_sortkeys_ codb_sortkeys

#define codb_put_control_word_ codb_put_control_word

#define codb_get_npes_ codb_get_npes

#define codb_get_poolnos_ codb_get_poolnos

#define codb_mask_control_word_ codb_mask_control_word

#define codb_get_control_word_ codb_get_control_word

#define codb_iput_ codb_iput

#define codb_dget_ codb_dget

#define codb_rget_ codb_rget

#define codb_rput_ codb_rput

#define codb_static_init_ codb_static_init

#define codb_linkdb_ codb_linkdb

#define codb_read_metadata_ codb_read_metadata

#define codb_datetime_ codb_datetime

#define codb_analysis_datetime_ codb_analysis_datetime

#define codb_create_pool_ codb_create_pool

#define codb_read_metadata2_ codb_read_metadata2
```

```
#define codb_openprt_ codb_openprt

#define codb_d2u_ codb_d2u

#define codb_lineprt_ codb_lineprt

#define codb_sqlprint_ codb_sqlprint

#define codb_closeprt_ codb_closeprt

#define codb_packer_ codb_packer

#define codb_gethandle_ codb_gethandle

#define codb_pc_filter_ codb_pc_filter

#define codb_print_vars_ codb_print_vars

#define codb_getindex_ codb_getindex

#define codb_putindex_ codb_putindex

#define get_thread_id_ get_thread_id

#define get_max_threads_ get_max_threads


#define cmdb_reg_ cmdb_reg

#define cmdb_vecreg_ cmdb_vecreg

#define cmdb_debug_ cmdb_debug

#define cmdb_set_ cmdb_set

#define cmdb_reset_ cmdb_reset

#define cmdb_name_ cmdb_name

#define cmdb_get_ cmdb_get


/*cmaio.h*/

#define cma_open_ cma_open

#define cma_attach_ cma_attach

#define cma_detach_ cma_detach

#define cma_read_ cma_read

#define cma_write_ cma_write

#define cma_info_ cma_info

#define cma_close_ cma_close

#define cma_debug_ cma_debug
```

```
#define cma_stat_ cma_stat

#define cma_get_report_ cma_get_report

#define cma_get_ddrs_ cma_get_ddrs

#define cma_wrapup_ cma_wrapup

#define cma_prt_stat_ cma_prt_stat

#define cma_rewind_ cma_rewind

#define cma_readi_ cma_readi

#define cma_readb_ cma_readb

#define cma_writei_ cma_writei

#define cma_writeb_ cma_writeb

#define cma_bin_info_ cma_bin_info

#define cma_bin_file_ cma_bin_file

#define cma_is_packed_ cma_is_packed

#define cma_on_mrfs_ cma_on_mrfs

#define cma_get_concat_ cma_get_conca

#define cma_get_concat_byname_ cma_get_concat_byname

#define cma_flpcheck_ cma_flpcheck

#define cma_flperr_ cma_flperr



/*pcma_extern.h*/

#define upcma_info_ upcma_info

#define pcma2cma_ pcma2cma



/*svipc.h*/

#define svipc_open_ svipc_open

#define svipc_close_ svipc_close

#define svipc_write_ svipc_write

#define svipc_read_ svipc_read

#define svipc_inquire_ svipc_inquire

#define svipc_setvar_ svipc_setvar

#define svipc_debug_ svipc_debug
```

```
/*odb.h*/

#define ctxprint_ ctxprint

#define ctxreg_ ctxreg

#define ctxdebug_ ctxdebug

#define ctxid_ ctxid


/*odb.h*/

#define write_ddl_ write_ddl


/*odb.h*/

#define iolockdb_ iolockdb


/*odb.h*/

#define cmpl_abort_ cmpl_abort


/*odb.h*/

#define util_walltime_ util_walltime

#define util_cputime_ util_cputime

#define util_ihpstat_ util_ihpstat

#define ctransfer_ ctransfer

#define kmg_ kmg

#define pseudo_main_ pseudo_main

#define binary_open_ binary_open

#define binary_flush_ binary_flush

#define binary_close_ binary_close

#define binary_put_ binary_put

#define binary_get_ binary_get

#endif
```