Compression des champs et optimisation de la taille des fichiers coupleurs

Par : Rachida El Ouaraini

Sous la supervision de : Jean-Marc Audoin & Ryad El Khattib

Merci à mes encadrants Jean-Marc et Ryad, à Karim Yessad, Gabor Radnoti, Jean-Daniel Gril et Claude Fischer pour leur aide précieuse qui m'a permis d'avancer dans mon travail.

19/03/2006- 30/04/2006

GMAP/CNRM - Météo-France

Sommaire

Résumé

Ce travail a consisté à résoudre la bug qui existait dans le code Aladin/Arpège quand on faisait appel à la compression par Gribex version 1 pour les champs spectraux.

La bug a été résolue pour NVGRIB=3, la compression des fichiers historiques est dorénavant possible.
Un examen de l'effet de compression sur les champs en points de grille (PDG) et les champs spectraux montre que les champs PDG sont mieux compressés que les champs spectraux : plus on demande des champs PDG en sortie, moins la taille du fichier est grande.

Réduction de la taille des fichiers historiques

Les tests effectués sur la compression des fichiers par gribex version 1 (NVGRIB=3) ont montré que la taille des fichiers historiques a diminué d'environ **20%**. Ce pourcentage est susceptible de baisser ou croître en fonction du nombre de champs demandés en sortie, et plus particulièrement les champs en points de grille puisqu'ils sont mieux compressés.
Les fichiers historiques issus de la chaîne double à Météo-France ont une taille d'environ 90 Mo et contiennent 591 champs de données, dont 404 sont en points de grille. Ce grand nombre de champs PDG est dû à la production des nouveaux champs Lopez.
La compression de ces fichiers permettra très probablement d'avoir un pourcentage de réduction **supérieur** à 20% !

Réduction de la taille des fichiers Fullpos

En utilisant NFPGRIB=3, on a pu réduire la taille des fichiers en sortie du Fullpos **Lalon** d'environ **25%** ! Ce pourcentage est encore une fois susceptible de baisser ou d'augmenter en fonction du nombre de champs demandé en sortie.

L'étape « finale » de mon travail a consisté à tourner la configuration E927 et EE927 avec NFPGRIB=3 *afin de r**éduire la taille des fichiers coupleurs**.*
Après modification du code, on a obtenu des champs spectraux à la sortie de la configuration 927 compressés mais qui sont erronés, ce qui montre qu'il y a toujours une bug à résoudre pour la configuration 927.

I. Contexte

La quantité de données à stocker dans un centre météorologique ne cesse d'augmenter, ce qui pose un problème quand au stockage des données ou quand à leur transfert entre centres météorologiques.
En effet, les moyens de stockage ne sont pas toujours suffisants pour abriter toutes les données produites et les moyens de transmission de données ne sont pas toujours aussi rapides pour transmettre et recevoir les données au temps opportun !
Un exemple sur la contrainte que pose la taille des fichiers dans la communauté Aladin est le couplage au sein du Service Météorologique Marocain. En effet, celui-ci est contraint à utiliser un couplage asynchrone du simple fait que la ligne spécialisée de 128 Kb entre Casablanca et Toulouse ne permet pas aux fichiers coupleurs du modèle Aladin NORAF (domaine assez grand) d'arriver à temps, ce qui contraint la météo marocaine à utiliser des coupleurs produits le réseau précédent.

Le défi s'impose, il s'agit d'optimiser la taille des fichiers météorologiques tout en gardant une bonne qualité d'information. Ceci est possible en utilisant la compression ou le compactage de données.
Jusqu'ici, on n'a utilisé que du compactage avec GRIBEX Version 0. La compression étant possible en faisant appel à GRIBEX version 1.

C'est dans ce cadre que Jadwiga Woyciechowska a commencé pendant l'été de 2005 des tests sur la compression des fichiers Aladin sous la supervision de Ryad EL Khattib. Ces tests qui n'ont porté que sur le modèle Aladin ont montré qu'il y a une bug dans le code.
L'objectif principal de mon stage est de résoudre cette bug dans le code Aladin et Arpège afin de pouvoir compresser les fichiers!

II. Le compactage et la compression des fichiers

Une nuance est à faire entre le compactage et la compression des fichiers. Le compactage consiste à coder les champs sur un nombre de bits inférieur tout en perdant un peu sur la précision des données. La compression permet de coder les champs sur un nombre inférieur de bits sans perdre de l'information !

Le niveau de compactage ou de compression peut être précisé par la clé NVGRIB dans la nameliste NAMFA et la clé NFPGRIB dans la nameliste NAMFPC pour le fullpos et la configuration 927.
En effet, ces clés peuvent assumer 3 valeurs : 0, 1 ou 2.
Avec GRIBEX Version 1, deux autres valeurs sont possibles : -1 et 3.
Le tableau ci-dessous explique la signification de chaque valeur.

Valeur de NVGRIB	***Définition***
0	Pas de compactage. On garde toute la précision de la donnée. Les tableaux spectraux sont rangés différemment dans le modèle et dans le fichier.
1	GRIBEX version 0 est utilisé pour encoder les données dans un format compacté.
2	Il s'agit d'une version améliorée de compactage par Gribex Version 0.
-1	Pas de compactage. La seule différence avec la valeur « 0 » est le type d'arrangement des tableaux spectraux dans les fichiers qui est dans ce cas le même que dans le modèle.
3	GRIBEX version 1 est utilisé. Il effectue la compression de données (Second Order Packing).

Les deux tableaux suivants illustrent l'efficacité de la compression des champs spectraux et points de grille :

Champs spectraux	*Taille avant compression*	*Taille après compression*
SURFPRESSION	21198	17718
CUF1PRESSURE	21198	19395
S001TEMPERATURE	21198	19122
S006HUMI.SPECIFI	21198	16915
S002WIND.U.PHYS	21198	18828

Champs point de grille	*Taille avant compression*	*Taille après compression*
SURFALBEDO.VEG	22511	12657
SURFB.OF.OZONE	22511	17
SURFA.OF.OZONE	22511	8512
SURFRESERV.GLACE	22511	681
SURFDENSIT.NEIGE	22511	596

A Météo-France, seul le compactage des données est utilisé en opérationnel. Les tests effectués pour activer la compression ont révélé qu'il y a des bugs à résoudre dans le code Aladin et Arpège!

III. Problèmes de la compression dans le code Aladin

1. Description de la bug dans le code Aladin

La bug a été découverte par Jadwiga Woychiechowska et Ryad El Khattib lors des tests effectués sur la compression des fichiers pendant l'été 2005. Cette bug apparaissait quant ils activaient la compression (NVGRIB=3) en utilisant un nombre de processeurs différent pour écrire et lire des tableaux spectraux dans deux étapes successives: par exemple un fullpos lancé sur deux processeurs derrière une prévision lancée sur 3 processeurs. Le rapport rédigé par Jadwiga [1] présente quelques exemples de champs « bugués » et qui sont totalement faux. Pour expliquer l'origine de ce problème, il est très important de rappeler que le rangement des coefficients spectraux diffère du modèle au fichier FA pour les valeurs de compactage 0, 1 et 2 et que ce rangement est le même dans le modèle et dans le fichier pour les valeurs -1 et 3.
La lecture et l'écriture des tableaux spectraux ainsi que leur distribution sur les noeuds se fait comme suit:

Pour NVGRIB=0, 1 ou 2, les tableaux spectraux sont lus, réordonnés de la structure fichier à la structure modèle, puis distribués sur les noeuds :

Lecture	1. **Lecture** des tableaux spectraux à partir du fichier (**suspeca.F90**) 2. **Réordonnancement** : structure fichier --> structure modèle (**spreord.F90**) 3. **Distribution spectrale :** tableau global --> tableaux locaux **(disspec0.F90**)
Ecriture	1. **Distribution spectrale** : tableaux locaux --> tableau global semi-distribué (**diwrspe0.F90**) 2. **Réordonnancement :** structure modèle --> structure fichier (**spreord.F90**) 3. **Ecriture** des tableaux dans le fichier (**wrspeca.F90**)

Pour NVGRIB=-1 ou 3, les tableaux spectraux sont lus puis distribués sur les noeuds. Il n'y a pas de réordonnancement, la structure des tableaux est la même dans le fichier et dans le modèle.

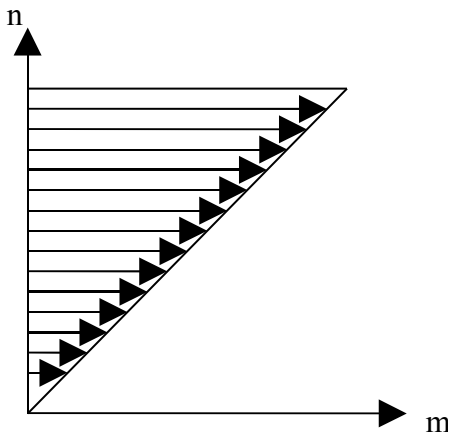
Lecture	1. **Lecture** (**suspeca.F90***)* 2. **Distribution spectrale** : tableau global --> tableaux locaux(**disspec0.F90**)
Ecriture	1. **Distribution spectrale** : tableaux locaux --> tableau global semi-distribué (**diwrspe0.F90**) 2. **Ecriture** du tableau semi-distribué *(***wrspeca.F90***)*

La routine spreord.F90 telle qu'elle est codée actuellement réordonne un tableau spectral de la structure fichier (« horizontale ») à la structure modèle (« verticale ») **semi-distribuée** : on classe tout d'abord les nombres d'onde zonaux (les JM) traités par le premier processeur puis ceux traités par le deuxième processeur et ainsi de suite, et non pas selon JM=0,1,2,3, ...

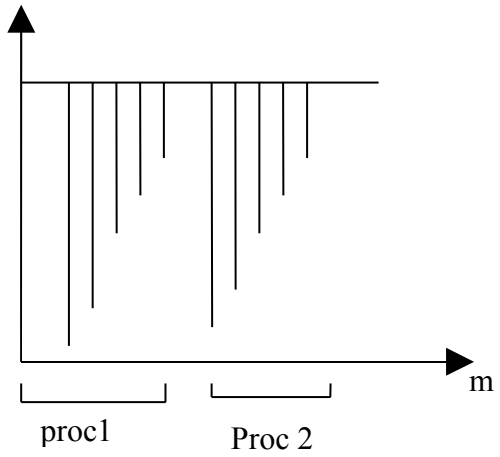
Pour les cas où NVGRIB=3 ou -1, on ne fait pas appel à spreord.F90, et donc les coefficients sont écrits dans le fichier d'une façon semi-distribuée.
A l'étape suivante (par exemple un FullPos derrière une prévision), quand disspec0.F90 veut envoyer ce tableau semi-distribué sur un nombre de processeurs différent à celui utilisé dans l'étape précédente, le programme plante!

Le schéma suivant illustre la bug :

Etape 1: (prévision)
NPROC=2



Structure fichier (NVGRIB=0, 1 ou 2)



Structure modèle après spreord.F90

Etape 2 : (par ex. Fullpos ou une autre prévision)
NPROC=4
NVGRIB=3

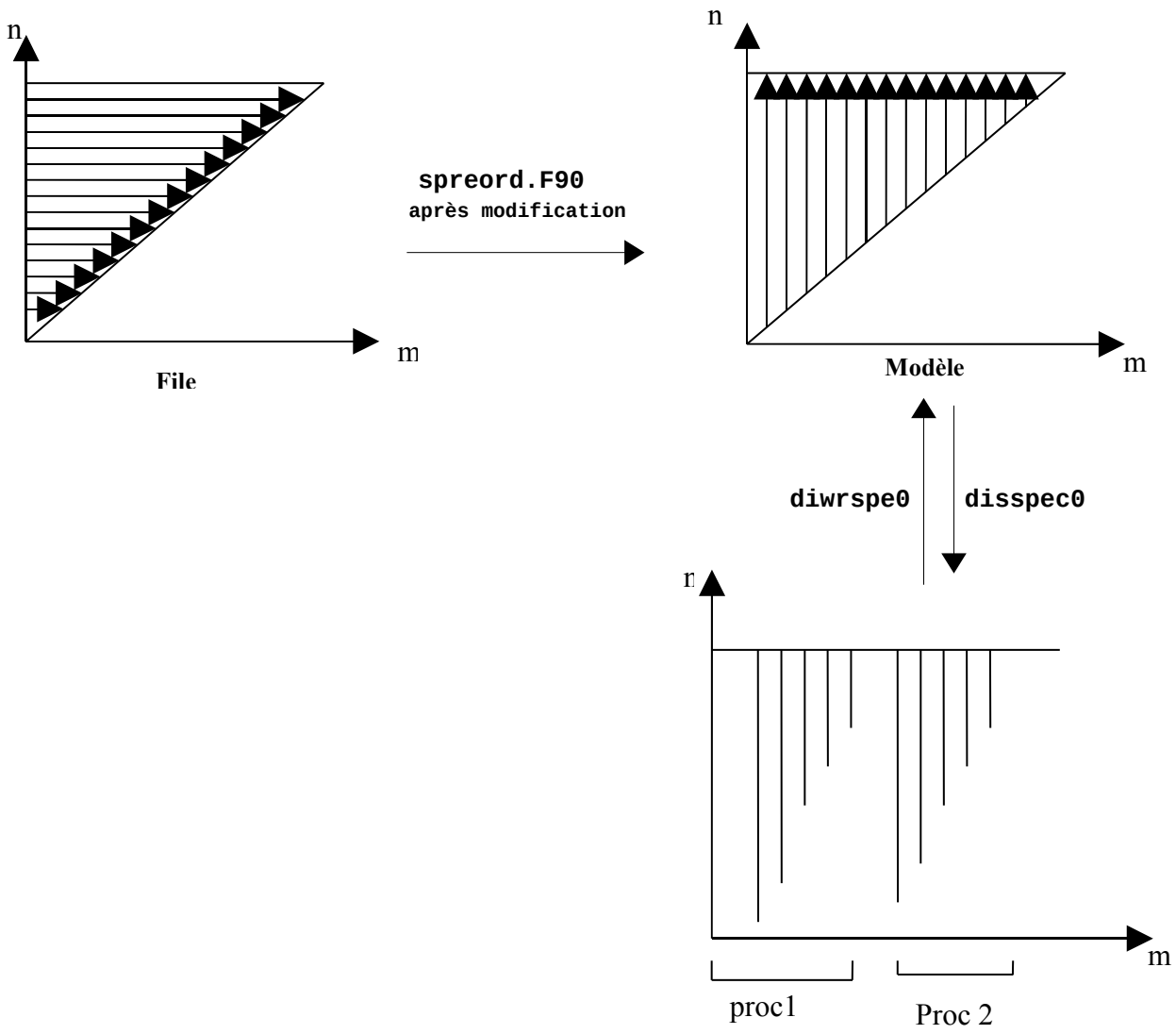
Le tableau global précédent semi-distribué sur 2 procs est lu. disspec0.F90 essaie d'envoyer ce tableau (déjà demi-distribué sur 2 procs) sur 4 procs, le programme plante !

2. Résolution de la bug

La solution de cette bug consiste à modifier :

- spreord.F90 de façon que cette routine change la structure des tableaux spectraux de la structure modèle à la structure fichier et vice versa de telle façon que les JM (nombres d'onde zonaux) soient rangés JM=0, 1, 2, 3, ... et non les JM traités par le premier processeur, ensuite les JM du deuxième et ainsi de suite.
- disspec0.F90 pour que la distribution spectrale des tableaux soit faite au niveau de cette routine et non par spreord.F90 (l'envoi du tableau global).
- diwrspe0.F90 pour la distribution spectrale (réception des tableaux locaux pour constituer un tableau global).

Le schéma suivant illustre la solution proposée :



Les figures 1, 2 et 3 montrent la pression de surface, la pression réduite au niveau de la mer et la température de surface, tracées avec l'option de compactage utilisée en opérationnel (NVGRIB=2) et avec l'option de compression (NVGRIB=3), après avoir résolu la bug dans le code Aladin.
On remarque qu'on trouve de bons résultats et que la compression des champs spectraux est devenue possible.

3. Modification du code ALADIN

Dans la routine **espareord.F90** (appelée par spreord.F90), on a remplacé le tableau NDIM0G par NDIM0GG.
NDIM0GG est un tableau nouvellement calculé. En effet, contrairement à ce qu'il laisse penser, le tableau NDIM0G n'est pas global, son calcul dépend du nombre de processeurs (voir la boucle JMLOC dans le code ci-dessous). On a dû alors créer un nouveau tableau NDIM0GG, qui quant à lui est un tableau global.

Calcul de NDIM0G

```
If (LELAM)  THEN
  IPOS = 1
  DO JA=1,NPRTRW
    DO JMLOC=1,NUMPP(JA)
      IM = NALLMS(NPTRMS(JA)+JMLOC-1)
      NDIM0G(IM) = IPOS
      IPOS = IPOS+2*NCPL2M(IM)
    ENDDO
  ENDDO
ELSE
  IPOS = 1
  DO JA=1,NPRTRW
    DO JMLOC=1,NUMPP(JA)
      IM = NALLMS(NPTRMS(JA)+JMLOC-1)
      NDIM0G(IM) = IPOS
      IPOS = IPOS+(NSMAX+1-IM)*2
    ENDDO
  ENDDO
ENDIF
```

Calcul NDIM0GG

```
IF (LELAM)  THEN
     IPOS=1
     DO JM=0,NMSMAX
       NDIM0GG(JM)=IPOS
       IPOS=IPOS+NCPL4M(JM)
     ENDDO
ELSE
      IPOS=1
      DO JM=0,NSMAX
        NDIM0GG(JM)=IPOS
        IPOS=IPOS+(NSMAX+1-JM)*2
      ENDDO
ENDIF
```

Dans la routine **disspec0.F90**, et avant la phase de distribution des tableaux spectraux sur les noeuds (la phase « SEND »), on transforme le tableau spectral global à un tableau spectral « semi-distribué », i.e. on range d'abord tous les nombres d'ondes zonaux (les « JM ») qui vont être traités par le premier processeur, ensuite tous ceux qui vont être traités par le deuxième processeur, et ainsi de suite.

```
IF (LELAM)  THEN
  IPOS=1
  DO JM=0,NMSMAX
    NDIM0GG1(JM)=IPOS
    IPOS=IPOS+NCPL4M(JM)
  ENDDO
  DO JFLD=1,KFIELDS
    DO JM=0,NMSMAX
      DO JN=0,NISNAX(JM)
        ISP=NDIM0G(JM)+4*JN
        ISPG=NDIM0GG1(JM)+4*JN
        PSPEC2(ISP:ISP+3,JFLD)=PSPEC(ISPG:ISPG+3,JFLD)
      ENDDO
    ENDDO
  ENDDO
  DO JFLD=1,KFIELDS
   DO JM=0,NMSMAX
     DO JN=0,NISNAX(JM)
       ISP=NDIM0G(JM)+4*JN
       PSPEC(ISP:ISP+3,JFLD)=PSPEC2(ISP:ISP+3,JFLD)
     ENDDO
    ENDDO
  ENDDO
ELSE
```

```
      IPOS=1
      DO JM=0,NSMAX
        NDIM0GG2(JM)=IPOS
        IPOS=IPOS+(NSMAX+1-JM)*2
      ENDDO
      DO JFLD=1,KFIELDS
        DO JM=0,NSMAX
          DO JN=JM,NSMAX
            ISP=NDIM0G(JM)+2*(JN-JM)
            ISPG=NDIM0GG2(JM)+2*(JN-JM)
            PSPEC2(ISP,JFLD)=PSPEC(ISPG,JFLD)
            PSPEC2(ISP+1,JFLD)=PSPEC(ISPG+1,JFLD)
          ENDDO
       ENDDO
    ENDDO

    DO JFLD=1,KFIELDS
      DO JM=0,NSMAX
        DO JN=JM,NSMAX
          ISP=NDIM0G(JM)+2*(JN-JM)
          PSPEC(ISP,JFLD)=PSPEC2(ISP,JFLD)
          PSPEC(ISP+1,JFLD)=PSPEC2(ISP+1,JFLD)
        ENDDO
      ENDDO
    ENDDO
  ENDIF
```

Dans la routine **diwrspe0.F90**, et après la phase de réception des tableaux spectraux locaux (la phase « RECEIVE ») à partir des différents noeuds, on réordonne les coefficients spectraux en classant les nombres d'onde zonaux selon : JM=0, 1, 2, 3, ...

```
   IF (LELAM)  THEN
    IPOS=1
    DO JM=0,NMSMAX
      NDIM0GG1(JM)=IPOS
      IPOS=IPOS+NCPL4M(JM)
    ENDDO
    DO JFLD=1,KFIELDS
      DO JM=0,NMSMAX
          DO JN=0,NISNAX(JM)
            ISP=NDIM0G(JM)+4*JN
            ISPG=NDIM0GG1(JM)+4*JN
            PSPEC2(ISPG:ISPG+3,JFLD)=PSPEC(ISP:ISP+3,JFLD)
          ENDDO
      ENDDO
    ENDDO
    DO JFLD=1,KFIELDS
      DO JM=0,NMSMAX
        DO JN=0,NISNAX(JM)
          ISPG=NDIM0GG1(JM)+4*JN
          PSPEC(ISPG:ISPG+3,JFLD)=PSPEC2(ISPG:ISPG+3,JFLD)
        ENDDO
      ENDDO
    ENDDO
  ELSE
    IPOS=1
    DO JM=0,NSMAX
      NDIM0GG2(JM)=IPOS
      IPOS=IPOS+(NSMAX+1-JM)*2
    ENDDO
    DO JFLD=1,KFIELDS
      DO JM=0,NSMAX
        DO JN=JM,NSMAX
          ISP=NDIM0G(JM)+2*(JN-JM)
          ISPG=NDIM0GG2(JM)+2*(JN-JM)
          PSPEC2(ISPG,JFLD)=PSPEC(ISP,JFLD)
          PSPEC2(ISPG+1,JFLD)=PSPEC(ISP+1,JFLD)
        ENDDO
      ENDDO
    ENDDO
    DO JFLD=1,KFIELDS
      DO JM=0,NSMAX
        DO JN=JM,NSMAX
          ISPG=NDIM0GG2(JM)+2*(JN-JM)
          PSPEC(ISPG,JFLD)=PSPEC2(ISPG,JFLD)
          PSPEC(ISPG+1,JFLD)=PSPEC2(ISPG+1,JFLD)
        ENDDO
      ENDDO
    ENDDO
  ENDIF
```

IV. Problèmes de la compression avec la nouvelle géométrie Aladin

La géométrie du modèle Aladin définit le domaine et la projection. Les anciennes routines de géométrie ont été remplacées par de nouvelles routines : EGGX.F90 a été remplacée par EGGPACK.F90.
A noter que l'on utilise la nouvelle géométrie (EGGPACK.F90) dans le modèle. L'ancienne géométrie n'est utilisée que par ECHIEN lors de la lecture d'un fichier possédant le cadre ancien.
Même si le modèle utilise les nouvelles routines de géométrie (EGGPACK), il est possible de spécifier le type du cadre du fichier FA par la clé NCADFORM dans la nameliste NAMOPH :
NCADFORM=0 : L'ancien format du cadre
NCADFORM=1 : Le nouveau format de cadre.

Ancien cadre →	**Nouveau cadre**
NROTEQ	-1
ELONR	ERPK
ELATR	ELON0
ELON1	ELAT0
ELAT1	ELONC
ELON2	ELATC
ELAT2	EDELX
ELON0	EDELY
ELAT0	ELX
ERPK	ELY
NSOTRP	EXWN
NGIV0	EYWN
ELX	ELON1
ELY	ELAT1
EDELX	ELON2
EDELY	ELAT2
EXWN	libre
EYWN	libre

Pour plus d'informations sur le nouveau format de cadre, lire [3].
Important: Toutes les expériences présentées dans ce travail ont été réalisées avec l'ancien cadre FA (NCADFORM=0).

Pour éviter tout problème de compression avec la nouvelle géométrie, quelques routines du logiciel FA devraient être modifiées pour prendre en compte la nouvelle géométrie en utilisant un codage et un décodage des données par Gribex Version 1.
Nous n'avons pu résoudre ce problème de compression avec la nouvelle géométrie Aladin par manque de temps. La résolution de ce problème sera parmi les objectifs du prochain stage prévu pour continuer le travail sur la compression des coupleurs au mois août.

V. Problèmes de la compression des fichiers coupleurs

Après avoir résolu la bug dans le code Aladin pour NVGRIB=3, nous avons testé la compression des fichiers coupleurs en utilisant NFPGRIB=3.
Pour cela, nous avons dû changer la routine **trwvtof.F90** relative à la distribution spectrale dans la partie post-processing du modèle.
Nous avons réalisé 4 expériences : la première et la deuxième consistent à lancer la configuration E927 et EE927 avec NFPGRIB=2. La troisième et la quatrième consistent à lancer la configuration E927 et EE927 avec NFPGRIB=3.

Expérience 1 :
prévision Arpège : NPROC=2, NVGRIB=3
E927 : NPROC=3, NFPGRIB=2

Expérience 2 :
prévision Aladin : NPROC=2, NVGRIB=3
EE927 : NPROC=3, NFPGRIB=2

Expérience 3 :
prévision Arpège : NPROC=2, NVGRIB=3
E927 : NPROC=3, NFPGRIB=3

Expérience 4 :
prévision Aladin : NPROC=2, NVGRIB=3
EE927 : NPROC=3, NFPGRIB=3

Les figures 4, 5 et 6 montrent les champs : pression de surface, température de surface et pression réduite au niveau de la mer, en sortie de la procédure de couplage, en utilisant l'option de compactage utilisée en opérationnel NFPGRIB=2 et en utilisant l'option de compression NFPGRIB=3.
Les figures 4 et 5 montrent que l'on obtient une pression de surface et une pression réduite au niveau de la mer erronées quand on fait appel à la compression. Ceci indique qu'il existe toujours une bug à résoudre dans la partie post-processing pour réussir la compression des fichiers coupleurs.
En revanche, nous avons obtenu une température de surface correcte en mode compression, voir figure 6. Ceci est tout à fait normal puisque la température de surface est un champ en points de grille et que la bug trouvée ne concerne que les champs spectraux : les champs en points de grille sont compressés sans aucun problème!
Les résultats présentés dans les figures 4, 5 et 6 sont issus de la configuration EE927. Les mêmes résultats ont été obtenus pour la configuration E927.

VI. Perspectives

- Tourner la configuration 927 avec NFPGRIB=3.
- Créer une table dynamique qui fait la correspondance « Nom du champs FA --> Code Grib » (actuellement la table est codée en dur).
- Résoudre le problème de la compression avec le nouveau cadre FA (NCADFORM=1).
- Harmoniser la valeur par défaut de NVGRIB par rapport au fichier initial, et la valeur par défaut de NFPGRIB par rapport à NVGRIB.

VII. Références

[1] Woyciechowska, J., Tests with Second Order Packing. Internal Report, July-August 2005.
[2] Paradis, D., Clochard, J., GRIBEX introduction in Fa files, ALADIN Newsletter 27, July-December 2004.
[3] Gril, J.D., Les nouvelles routines de géométrie dans le modèle Aladin. Internal Report, January 2006.