

# Evaluation du PREP

Mohamed JIDANE : DSI/SCCO, DMN, Maroc  
Encadrement : Françoise TAILLEFER, CNRM/GMAP  
Novembre 2013

## 1/ INTRODUCTION

L'objectif de ce travail consiste à évaluer et tester PREP pour différentes configurations.

PREP est le programme qui permet de passer d'une grille SURFEX vers une autre grille SURFEX, ou de rester sur la même grille tout en modifiant les choix pour les schémas de surface.

Il effectue l'initialisation des variables pronostiques du schéma de surface, comme les profils de températures, l'eau et les contenus du sol en glace, les réservoirs d'interception et les réservoirs de neige.

Des développements récents ont été faits par Daan et Tayfun pour optimiser ce code PREP.

Ces optimisations concernent l'utilisation de OpenMP dans trois routines, coûteuses dans leur cas d'étude, (AV\_PATCH\_PGD\_1D, BILIN, AV\_PGD\_1D).

Ils ont aussi fait du développement de code pour réduire le nombre d'appel à la routine BILIN (qui fait l'interpolation bilinéaire) en bouclant sur les patches au lieu de boucler sur les types de végétation ainsi que de boucler seulement sur les types de couverts existants dans le fichier pour calculer les moyennes.

PREP est appelé à être utilisé en opérationnel à METEO-FRANCE pour bientôt pour les besoins de la prévision d'ensemble Arome.

Le travail consiste donc à tester PREP, voir si ça marche et surtout voir combien ça coûte en terme de temps de calcul et de mémoire utilisée pour des configurations de type Arome --> Arome, Arpège --> Arome ou Aladin --> Arome.

Toutes les expériences ont été effectuées avec le cycle 40 et la version 7.3 de SURFEX.

## 2/ EXPERIENCES

Mon pack est basé sur la version de code faite par Yann Seity et qui regroupe le cycle 40, la version 7.3 de SURFEX et la branche NEW\_PREP de Stéphanie Faroux (Surfex code management) qui inclut les modifications de Daan et Tayfun.

Le PREP ne prenant pas encore en charge le format FA, on a du travailler soit sur des dates anciennes où on avait des fichiers SURFEX en LFI, soit faire appel au convertisseur FA --> LFI développé par Philippe Marguinaud.

### A/ Cas Arome --> Arome :

Notre premier test fut de reproduire une expérience faite par Yann (65LK) avec le cycle 36+1 (aller de la grille Arome 2.5 vers un petit domaine midpyr).

Le PREP plantait pour un problème concernant les fichiers des covers :

**ecoclimapl\_covers\_param.bin**  
**ecoclimapll\_eu\_covers\_param.bin**

prep.exe	00000000006AA1E9	read_covers_param	107	read_covers_param.F90
prep.exe	0000000000549DCF	ini_data_cover_	889	ini_data_cover.F90
prep.exe	00000000004DAD0E	init_surf_atm_n_	353	init_surf_atmn.F90
prep.exe	000000000043E916	init_pgd_surf_atm	67	init_pgd_surf_atm.F90
prep.exe	0000000000434131	MAIN__	178	prep.F90

Le même problème est apparu en voulant faire du PREP de Arome 2.5 vers la grille 1.3 (plantage au même endroit dans read\_covers\_param.F90).

Des affichages dans cette routine montrent bien que le problème vient des fichiers covers :

```
>>>process_options(): DR_HOOK_OPT="WALLPROF"
  Reading ecoclimapl_covers_param.bin
    1      1
    2      2
    3      3
    4      4
    5      0
    6 1072693248
forrtl: severe (174): SIGSEGV, segmentation fault occurred
```

En effet, les fichiers \*covers\_param.bin sont faits avec une version de SURFEX antérieure à la version 7.3 et donc il fallait prendre les nouveaux qui sont faits avec la dernière version.

On peut toujours se passer de ces fichiers et laisser SURFEX les recalculer en mettant dans la nameliste :

```
&NAM_READ_DATA_COVER
  LREAD_DATA_COVER=.FALSE.,
/
```

Mais dès lors le temps d'exécution se retrouve très allongé.

En utilisant le profiling de DR\_HOOK, on se rend compte qu'une seule routine (HOR\_EXTRAPOL\_SURF) requiert la majorité du temps de calcul.

PREP Arome grille 2.5 vers grille 1.3 :

	1 thread	24 threads	24 threads + My OMP modifs
Elapsed	00:35:50	00:33:42	00:03:23
MaxRSS		7512328K	7351200K
Top routines	88.01% HOR_EXTRAPOL_SURF 7.35% BILIN_2 0.59% LFILDO_MT 0.51% MODI_AV_PGD:AV_PGD_1D	94.62% HOR_EXTRAPOL_SURF 0.66% BILIN_2 0.60% LFILDO_MT 0.50% MODE_GRIDTYPE_CONF_PROJ: XY_CONF_PROJ	41.81% HOR_EXTRAPOL_SURF 6.75% BILIN_2 6.40% LFILDO_MT 5.20% MODE_GRIDTYPE_CONF_PROJ:X Y_CONF_PROJ

Ce qu'il faut remarquer aussi c'est que bien que la grille Arome 2.5 ne couvre pas la totalité de la grille 1.3, le PREP ne mentionne aucun problème à ce niveau et arrive à calculer des valeurs pour ces points de la grille d'arrivée qui ne sont pas inclus dans la grille de départ.

Pour aller de la grille 1.3 vers la grille 2.5 le PREP plante pour la lecture d'un champ non existant (il essaie de lire L\_GROUND\_DEPTH au lieu de lire L\_GROUND\_DPT) :

```
exit from FMREAD with IRESP:      -47
  | HRECFM = L_GROUND_DEPTH
  ABOR1 CALLED
```

Une correction a été portée aux routines read\_pgd\_isba\_parn.F90 et mode\_read\_extern.F90 pour remédier au problème :

```
YRECFM2='L_GROUND_DPT'
!! YRECFM2='L_GROUND_DEPTH'
!! IF (IBUGFIX>=3) YRECFM2='L_GROUND_DPT'
```

PREP Arome grille 1.3 vers grille 2.5 :

	1 thread	24 threads	24 threads + My OMP modifs
Elapsed	01:57:29	01:52:24	00:07:09
MaxRSS		6478180K	6421064K
Top routines	97.32% HOR_EXTRAPOL_SURF 1.27% BILIN_2 0.22% BILIN_1 0.20% LFILDO_MT	98.40% HOR_EXTRAPOL_SURF 0.28% LFILDO_MT 0.22% BILIN_1 0.16% MODI_AV_PGD:AV_PGD_1D	74.90% HOR_EXTRAPOL_SURF 3.56% LFILDO_MT 3.49% BILIN_1 2.69% MODI_AV_PGD:AV_PGD_1D

La parallélisation OpenMP de la routine HOR\_EXTRAPOL\_SURF nous a permis de rendre le PREP 10 à 15 fois plus rapide.

**B/ Cas Aladin --> Arome :**

On a aussi essayé de faire du PREP pour aller sur la grille Arome 2.5 en partant d'un fichier Aladin de l'époque.

Ce PREP là ne marche pas pour des problèmes concernant la neige : (fichier aladin\_20120115r0\_INIT\_SURF.lfi )

```
>>>process_options(): DR_HOOK_OPT="WALLPROF"
```

```
ALLOCATE TPSNOW%WSNOW 315307 1 1
ALLOCATE TPSNOW%WSNOW 151321 1 1
ALLOCATE TPSNOW%WSNOW 151321 1 1
ALLOCATE TPSNOW%WSNOW 151321 1 1
ALLOCATE TPSNOW%WSNOW 151321 1 1
ALLOCATE TPSNOW%WSNOW 47362 1 1
ALLOCATE TPSNOW%WSNOW 151321 1092544040 1
```

```
fortrtl: severe (41): insufficient virtual memory
```

Image	PC	Routine	Line	Source
libintlc.so.5	00002B5EAB703A1E	Unknown	Unknown	Unknown
libintlc.so.5	00002B5EAB7024B6	Unknown	Unknown	Unknown
libifcoremt.so.5	00002B5EAA0429EE	Unknown	Unknown	Unknown
libifcoremt.so.5	00002B5EA9FAF14E	Unknown	Unknown	Unknown
libifcoremt.so.5	00002B5EAA000DC9	Unknown	Unknown	Unknown
prep.exe	000000000090B8AA	allocate_gr_snow_	73	allocate_gr_snow.F90
prep.exe	00000000008FBEBE	prep_snow_extern_	147	prep_snow_extern.F90
prep.exe	000000000077DD19	prep_hor_snow_fie	132	prep_hor_snow_field.F90
prep.exe	000000000073D17C	prep_hor_snow_fie	119	prep_hor_snow_fields.F90
prep.exe	000000000068F3A5	prep_hor_teb_fiel	115	prep_hor_teb_field.F90
prep.exe	00000000005CF47D	prep_teb_	127	prep_teb.F90
prep.exe	000000000050142A	prep_town_	55	prep_town.F90
prep.exe	000000000045DB23	prep_surf_atm_	112	prep_surf_atm.F90
prep.exe	000000000043425F	MAIN_	183	prep.F90

Pour faire les essais avec des fichiers Aladin récents on a travaillé sur le cas Aladin-Réunion vers la grille Pégase.

Les fichiers de surface sont au format FA, donc on a utilisé le convertisseur développé par Philippe Marguinaud (SFXTOOLS).

La conversion (FA --> LFI) d'un fichier Aladin-Réunion de surface pose quelques problèmes : (field description)

```
SFXTOOLS  sfxfa2lfi  --sfx-fa--file  Reunion_20131115r0_analyse.sfx  --sfx-lfi-file  
pegase_20131115r0_ICMSHSURF+0000.lfi
```

ABOR1 CALLED

```
SFXFLDDDESC_LOOKUP: RI_SEA          WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: RI_SEA          WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: RN_SEA          WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: H_SEA           WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: LE_SEA          WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: LEI_SEA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: GFLUX_SEA       WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: FMU_SEA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: FMV_SEA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: CD_SEA          WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: CH_SEA          WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: CE_SEA          WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: Z0_SEA          WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: Z0H_SEA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: QS_SEA          WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: T2M_SEA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: T2MMIN_SEA      WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: T2MMAX_SEA      WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: Q2M_SEA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: HU2M_SEA        WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: HU2MMIN_SEA     WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: HU2MMAX_SEA     WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: ZON10M_SEA      WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: MER10M_SEA      WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: W10M_SEA        WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: W10MMAX_SEA     WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: LEG_ISBA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: LEGI_ISBA        WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: LEV_ISBA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: LES_ISBA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: LER_ISBA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: LETR_ISBA        WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: EVAP_ISBA        WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: DRAIN_ISBA       WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: RUNOFF_ISBA      WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: DRIVEG_ISBA     WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: RRVEG_ISBA      WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: SNOMLT_ISBA     WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: CD_ISBA         WAS NOT FOUND  
ABORT!  1 SFXFLDDDESC_LOOKUP: CH_ISBA         WAS NOT FOUND
```

ABORT!	1 SFXFLDDDESC_LOOKUP: CE_ISBA	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: Z0_ISBA	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: Z0H_ISBA	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: QS_ISBA	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: SWI3_ISBA	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: TSWI3_ISBA	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: DG3	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: WSAT3	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: WFC3	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: WWILT3	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: CD	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: CH	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: CE	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: Z0	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: Z0H	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: UREF	WAS NOT FOUND
ABORT!	1 SFXFLDDDESC_LOOKUP: ZREF	WAS NOT FOUND

On a ajouté la description de ces champs dans sfxflddesc\_mod.F90 et la conversion fonctionne très bien. Cependant le PREP (pour aller vers la grille Pégase 'Arome-Réunion') plante pour un champ manquant :

```
exit from FMREAD with IRESP:      -47
| HRECFM = DIM_WATER
ABOR1 CALLED
```

En effet, dans le fichier LFI d'entrée on ne dispose que du champ DIM\_FULL. Il fallait mentionner dans la nameliste le PGD de la grille de départ qui contenait tout les champs constants qui ne sont plus écrits dans les fichiers de surface.

```
&NAM_PREP_SURF_ATM
CFILETYPE      = 'LFI ',
CFILE          = 'LFI_SURF',
CFILEPGDTYPE  = 'LFI ',
CFILEPGD       = 'PGD_in',
/
```

Le PREP marche en une dizaine de secondes sur 24 threads (le domaine Pégase ne dispose pas de beaucoup de points terre).

Le fichier en sortie est au format LFI. Pour le convertir au format FA, il faut passer par deux étapes :

- créer un fichier vide au format FA qui contient le cadre de la grille d'arrivée à partir d'un fichier qui a le même cadre (fichier clim PGD par exemple) :

```
lftools faempty PGD.fa PREP.fa
```

- remplir ce fichier par les champs convertis du fichier LFI

```
SFXTOOLS sfxlfi2fa --sfx-fa--file PREP.fa --sfx-lfi-file PREP.lfi
```

Pour voir l'effet du PREP sur le fichier d'entrée et le comparer par rapport à un COUPLINGSURF normal on a tracé quelques champs dont la SST par exemple. Il semble que le PREP est raisonnable dans ses interpolations.

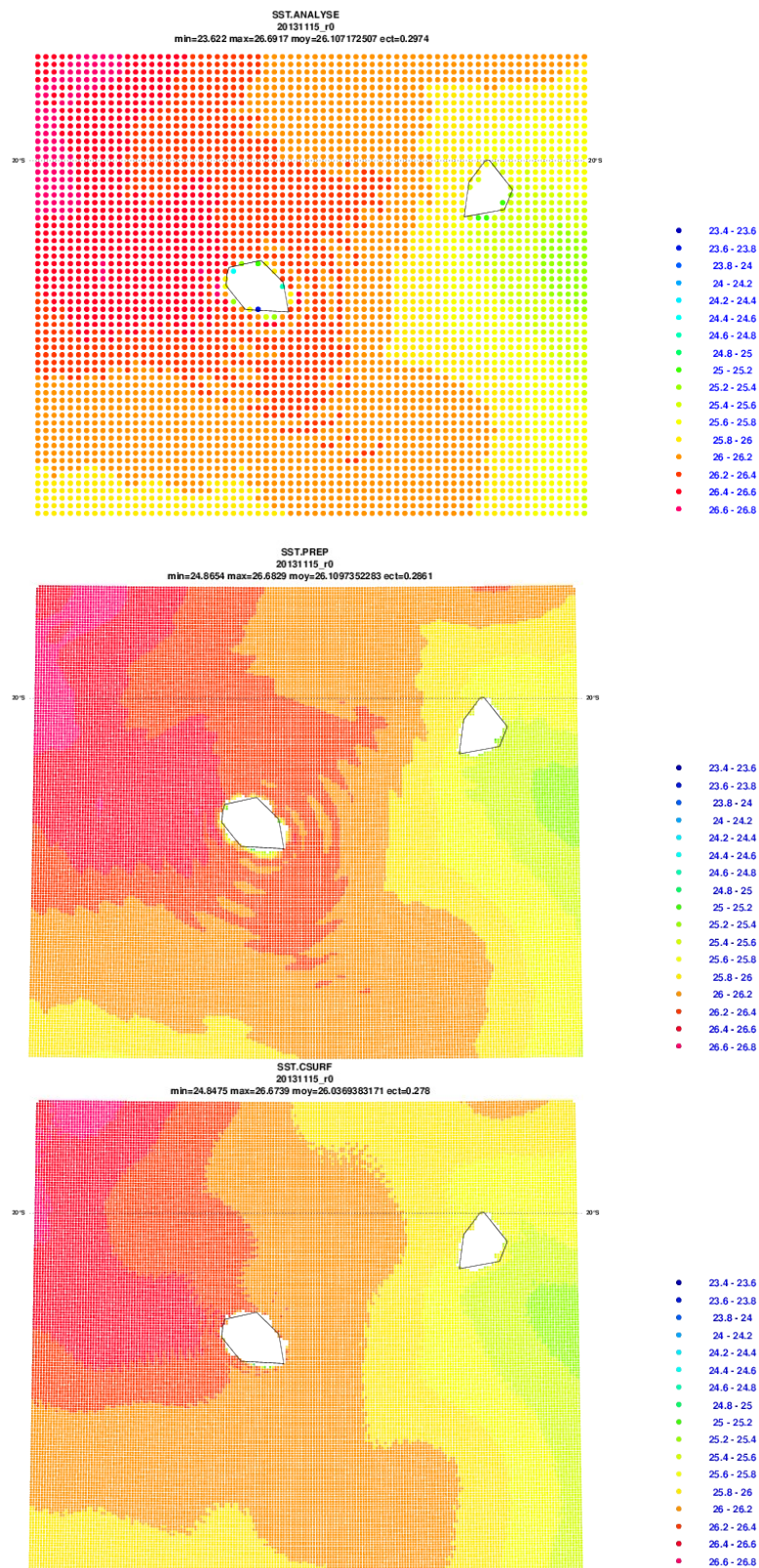


Figure 1 : Comparaison du champ SST entre l'analyse Aladin-Réunion, le fichier issu du PREP et le couplingsurf normal (à partir d'Arpège)

Deux expériences Olive ont été créées pour évaluer l'impact du PREP sur la prévision Pégase :

7BVE : modifiée (mon binaire et PREP à partir de l'analyse Aladin-Réunion)

7BVR : la référence (mon binaire et couplingsurf à partir d'Arpège)

Malheureusement les deux prévisions plantent pour le même problème (RM\_PATCH manquant) :

```
[STDOUT] : ] --- END DUMP OF SURFEX CACHE ---
[STDOUT] : SURFEX FIELD CACHE:RM_PATCH WAS NOT FOUND
[STDOUT] : ABORT! 48 SURFEX FIELD CACHE:RM_PATCH WAS NOT FOUND
[STDOUT] : MPL_ABORT: CALLED FROM PROCESSOR 48 THRD 1
[STDOUT] : MPL_ABORT: THRD 1 SURFEX FIELD CACHE:RM_PATCH WAS
NOT FOUND
[STDOUT] : [myproc#48,tid#1,pid#51865]: MASTER
[STDOUT] : [myproc#48,tid#1,pid#51865]: CNT0<1>
[STDOUT] : [myproc#48,tid#1,pid#51865]: SU0YOMB
[STDOUT] : [myproc#48,tid#1,pid#51865]: SUPHY
[STDOUT] : [myproc#48,tid#1,pid#51865]: SUPHMF
[STDOUT] : [myproc#48,tid#1,pid#51865]: SUPHMSE
[STDOUT] : [myproc#48,tid#1,pid#51865]: SUPHMSE_SURFACE
[STDOUT] : [myproc#48,tid#1,pid#51865]: SUPHMSE_SURFACE:AROINI_SURF
[STDOUT] : [myproc#48,tid#1,pid#51865]: AROINI_SURFC
[STDOUT] : [myproc#48,tid#1,pid#51865]: INIT_SURF_ATM_N
[STDOUT] : [myproc#48,tid#1,pid#51865]: INIT_NATURE_N
[STDOUT] : [myproc#48,tid#1,pid#51865]: INIT_ISBA_N
[STDOUT] : [myproc#48,tid#1,pid#51865]: READ_PGD_ISBA_N
[STDOUT] : [myproc#48,tid#1,pid#51865]: MODI_READ_SURF:READ_SURFX0
[STDOUT] : [myproc#48,tid#1,pid#51865]: READ_SURFX0_ARO
```

Pour régler ce problème il fallait régénérer avec mon binaire (SURFEX v7.3) le PGD en FA et utiliser celui là dans la prévision. Car sinon, on n'a pas la même version dans PGD et dans PREP et du coup il manque des champs dans le PGD.

Les tracés des différences des champs, à échéance 12 et 24 pour l'altitude et à échéance 03 pour le fichier de surface, entre les deux expériences sont sous le webdav : (<http://webdav/public/proc/jidane/>).

### **C/ Cas Arpège --> Arome :**

On a voulu tester aussi du Arpège vers Arome en partant d'un fichier fait par François Bouyssel (SURFEX n'est pas encore activé dans le modèle global).

Le fichier étant au format FA, il fallait le convertir en LFI :

```
SFXTOOLS sfxfa2lfi --sfx-fa--file arp_20121201_ICMSHSURF+0000 --sfx-lfi-file arp.lfi
```

```
ABOR1 CALLED
```

```
SFXFLDDDESC_LOOKUP: NGPGLO
```

```
WAS NOT FOUND
```



ABORT! 1 SFXFLDDDESC\_LOOKUP: NGPGLO WAS NOT FOUND

....

sfxtools 00000000005DA290 sfxflddesc\_mod\_mp 916 sfxflddesc\_mod.F90  
sfxtools 0000000000433CD3 sfxfa2lfi\_ 151 sfxfa2lfi.F90

En ajoutant la description de NGPGLO dans sfxflddesc\_mod.F90 la conversion FA → LFI fonctionne.

Le PREP lui il plante pour un problème de lecture du champ NLOPA :

```
exit from FMREAD with IRESP: -63
| HRECFM = NLOPA
ABOR1 CALLED
abort by abor1_sfx
ABORT! 1 abort by abor1_sfx
[myproc#1,tid#1,pid#70353]: PREP ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#70353]: PREP_SURF_ATM ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#70353]: PREP_SEA ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#70353]: PREP_SEAFLUX ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#70353]: PREP_HOR_SEAFLUX_FIELD
,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#70353]: PREP_SEAFLUX_EXTERN ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#70353]: PREP_GRID_EXTERN ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#70353]: PREP_GRID_GAUSS ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#70353]: MODI_READ_SURF:READ_SURFN1
,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#70353]: MODE_READ_SURF_LFI:READ_SURFN1_LFI
,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#70353]: ERROR_READ_SURF_LFI
,#1263,st=1,wall=0.001s/0.000s
[myproc#1,tid#1,pid#70353]: ABOR1_SFX ,#1,st=1,wall=0.000s/0.000s
```

error when reading articleNLOPA  
default value may be used; who knows?

```
-----
-----
----- FATAL ERROR in SURFEX -----
-----
-----
-
-
ERROR_READ_SURF_LFI: ERROR WHEN READING ARTICLE NLOPA
-
-
-----
-----
```

Pourtant l'article NLOPA est bien présent dans le fichier d'entrée arp.lfi :

```
/home/gmap/mrpe/jidanem/surfex/prep/ > lfitools lfilist arp.lfi | grep NLOPA
[ 'NLOPA', 1302, 10570 ],
```

Après investigation une correction a été apportée à prep\_grid\_gauss.F90 pour pouvoir lire le champ NLOPA :

```
YRECFM = 'NLOPA'  
CALL READ_SURF(HFILETYPE,YRECFM,INLOPA,IRESP,HDIR='A')
```

au lieu de :

```
YRECFM = 'NLOPA'  
CALL READ_SURF(HFILETYPE,YRECFM,INLOPA,IRESP)
```

En effet, la valeur par défaut de HDIR (type of field : 'H' : field with horizontal spatial dim ; '-' : no horizontal dim. ) est 'H'.

Et donc dans 'MODE\_READ\_SURF\_LFI:READ\_SURFN1\_LFI', le buffer NWORKD n'est pas alloué à la bonne valeur :

```
IF (HDIR=='A') THEN  
  ALLOCATE(NWORKD(IL1))  
ELSEIF (HDIR='-') THEN  
  ALLOCATE(NWORKD(NFULL))  
ENDIF  
!  
IF (HDIR=='H') THEN  
  CALL FMREADN1(CFILE_LFI, HREC, CLUOUT_LFI, NFULL, NWORKD, IGRID, &  
                ILENCH, HCOMMENT, NWORKB)  
ELSE  
  CALL FMREADN1(CFILE_LFI, HREC, CLUOUT_LFI, IL1, NWORKD(:), IGRID, &  
                ILENCH, HCOMMENT, NWORKB)  
END IF
```

Le PREP, après la correction, plantait cette fois-ci un peu plus loin :

forrtl: severe (174): SIGSEGV, segmentation fault occurred

Image	PC	Routine	Line	Source
prep.exe	0000000000768EAF	horibl_surf_	244	horibl_surf.F90
prep.exe	0000000000708A46	hor_interpol_gaus	92	hor_interpol_gauss.F90
prep.exe	0000000000628F68	hor_interpol_	71	hor_interpol.F90
prep.exe	0000000000632D39	prep_hor_seaflux_	112	prep_hor_seaflux_field.F90
prep.exe	00000000005B4B9D	prep_seaflux_	106	prep_seaflux.F90
prep.exe	00000000005401DA	prep_sea_	55	prep_sea.F90
prep.exe	000000000045DAFB	prep_surf_atm_	94	prep_surf_atm.F90
prep.exe	000000000043425F	MAIN__	186	prep.F90

L'investigation a montré que le tableau NINLO n'était pas alloué :

```
MJ: PREP_GRID_GAUSS  
MJ: size(INLOPA)= 1200  
MJ: INLOPA(1)= 18  
MJ: size(NINLO)= 0
```

D'où l'ajout de l'allocation de NINLO dans prep\_grid\_gauss.F90 :

```
IF (ALLOCATED(INLOPA)) DEALLOCATE(INLOPA)
ALLOCATE(INLOPA(INLATI))
IF (ALLOCATED(NINLO)) DEALLOCATE(NINLO)
ALLOCATE(NINLO(INLATI))
```

Après la correction, le PREP plante, cette fois-ci, un peu plus loin :

fortrl: severe (174): SIGSEGV, segmentation fault occurred

Image	PC	Routine	Line	Source
prep.exe	00000000009078CF	horibl_surf_	611	horibl_surf.F90
prep.exe	00000000008EBCC1	hor_interpol_gaus	94	hor_interpol_gauss.F90
prep.exe	0000000000628F68	hor_interpol_	71	hor_interpol.F90
prep.exe	0000000000632D39	prep_hor_seaflux_	112	prep_hor_seaflux_field.F90
prep.exe	00000000005B4B9D	prep_seaflux_	106	prep_seaflux.F90
prep.exe	00000000005401DA	prep_sea_	55	prep_sea.F90
prep.exe	000000000045DAFB	prep_surf_atm_	94	prep_surf_atm.F90
prep.exe	000000000043425F	MAIN_	186	prep.F90

Le problème s'est avéré être dû à l'utilisation d'une variable (NNI) qui n'est initialisée que si on travaille avec du GRIB en entrée (dans la routine prep\_grid\_gauss.F90, cas climat).

```
CALL HORIBL_SURF(XILA1,XILO1,XILA2,XILO2,NINLA,NINLO, NNI, PFIELDIN(:,JL), &
INO, ZLON, ZLAT, PFIELDOUT(:,JL), .FALSE., KLUOUT, LINTERP, IMASKIN, &
IMASKOUT)
```

La correction apportée dans hor\_interpol\_gauss.F90 est la suivante :

```
IF (NNI==0) NNI=NILEN
!
ALLOCATE(IMASKIN (NNI))
```

Le PREP après cette correction va, bien sûr, planter mais cette fois-ci pour un problème de géométrie :

prep.exe	000000000049B5F3	abor1_sfx_	96	abor1_sfx.F90
prep.exe	0000000000907BEB	horibl_surf_	567	horibl_surf.F90
prep.exe	00000000008EC039	hor_interpol_gaus	98	hor_interpol_gauss.F90
prep.exe	0000000000628F68	hor_interpol_	71	hor_interpol.F90
prep.exe	0000000000632D39	prep_hor_seaflux_	112	prep_hor_seaflux_field.F90
prep.exe	00000000005B4B9D	prep_seaflux_	106	prep_seaflux.F90
prep.exe	00000000005401DA	prep_sea_	55	prep_sea.F90
prep.exe	000000000045DAFB	prep_surf_atm_	94	prep_surf_atm.F90
prep.exe	000000000043425F	MAIN_	186	prep.F90

```
-----
----- FATAL ERROR in SURFEX -----
-----
```

```

-----
-
HORIBLE_SURF: INPUT DOMAIN SMALLER THAN OUTPUT ONE - LONGITUDE
LOCAL
-
-----

```

Des affichages semblent montrer que c'est la variable XILO2 qui est mal initialisée :

```

XILA1= 89.96250000000000 XILO1= 0.0000000000000000E+000 XILA2=
-89.96250000000000 XILO2= 340.00000000000000 NINLA= 1200
LROTPOLE= T
MJ XLAP= 46.4688478326275 XLOP= 2.57831007808870 XCOEF=
2.2000000000000000

```

En forçant XILO2 à 360. dans prep\_grid\_gauss.F90 :

```

XILA1=90.*(1.-0.5/INLATI)
XILO1=0.
XILA2=-90.*(1.-0.5/INLATI)
XILO2=360.*(INLOPA(1)-1.)/INLOPA(1)
IF (LROTPOLE) THEN
  XILO2=360.
ENDIF

```

Le PREP (Arpège --> Arôme 2.5) semble aller encore plus loin mais il plante, cette fois-ci, pour un problème de nombre de couches de neige :

```

MJ: PREP_SNOW_EXTERN: TZSNOW%NLAYER=      1 K LAYER=      1
..
MJ: PREP_SNOW_EXTERN: TZSNOW%NLAYER=      1 K LAYER=      1
....
MJ: PREP_SNOW_EXTERN: TZSNOW%NLAYER=      1 K LAYER=      1
....
MJ: PREP_SNOW_EXTERN: TZSNOW%NLAYER=      1 K LAYER=      1
.....
MJ: PREP_SNOW_EXTERN: TZSNOW%NLAYER=      0 K LAYER=      1
ABOR1 CALLED
abort by abor1_sfx
ABORT!  1 abort by abor1_sfx
[myproc#1,tid#1,pid#45155]: PREP ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#45155]: PREP_SURF_ATM ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#45155]: PREP_TOWN ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#45155]: PREP_TEB ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#45155]: PREP_HOR_TEB_FIELD ,#11,st=1,wall=1.097s/0.011s
[myproc#1,tid#1,pid#45155]: PREP_HOR_SNOW_FIELDS,#2,st=1,wall=11.590s/0.014s
[myproc#1,tid#1,pid#45155]: PREP_HOR_SNOW_FIELD ,#6,st=1,wall=11.573s/0.814s
[myproc#1,tid#1,pid#45155]: PREP_SNOW_EXTERN ,#5,st=1,wall=1.778s/0.056s
[myproc#1,tid#1,pid#45155]: ABOR1_SFX ,#1,st=1,wall=0.000s/0.000s
SDL_TRACEBACK: Calling INTEL_TRBK, THRD =      1

prep.exe  00000000007C6560 abor1_      36 abor1.F90

```

```

prep.exe 000000000049B5F3 abor1_sfx_ 96 abor1_sfx.F90
prep.exe 00000000008D9826 prep_snow_extern_ 164 prep_snow_extern.F90
prep.exe 00000000007584D9 prep_hor_snow_fie 132 prep_hor_snow_field.F90
prep.exe 000000000071DA5C prep_hor_snow_fie 119 prep_hor_snow_fields.F90
prep.exe 0000000000686875 prep_hor_teb_fiel 115 prep_hor_teb_field.F90
prep.exe 00000000005CD3DD prep_teb_ 127 prep_teb.F90
prep.exe 00000000004FFA9A prep_town_ 55 prep_town.F90
prep.exe 000000000045DBD3 prep_surf_atm_ 112 prep_surf_atm.F90
prep.exe 000000000043425F MAIN__ 186 prep.F90

```

-----  
 ----- FATAL ERROR in SURFEX -----  
 -----

-  
 -  
 PREP\_SNOW\_EXTERN: SNOW NLayer IN EXTERN FILE MUST BE GROWER THAN  
 CURRENT NLayer  
 -  
 -

### D/ Cas Arpège --> Aladin :

Le test du PREP en allant du même fichier global arp.lfi vers le domaine Aladin-Réunion, quant à lui, ne pose aucun problème.

Ci-dessous le profiling de DR\_HOOK :

```

Name of the executable : ./prep.exe
Number of MPI-tasks : 1
Number of OpenMP-threads : 24
Wall-times over all MPI-tasks (secs) : Min=49.070, Max=49.070, Avg=49.070, StDev=0.000
Routines whose total time (i.e. sum) > 1.000 secs will be included in the listing
  Avg-% Avg.time Min.time Max.time St.dev Imbal-% # of calls : Name of the routine
35.07% 17.207 17.207 17.207 0.000 0.00% 312616 : LFILDO_MT
 6.96%  3.413  3.413  3.413 0.000 0.00% 175 : HORIBL_SURF
 6.94%  3.404  3.404  3.404 0.000 0.00% 486 : FMREADX1
 6.52%  3.197  3.197  3.197 0.000 0.00% 6 : MODI_READ_SURF:READ_SURFX2COV
 6.38%  3.131  3.131  3.131 0.000 0.00% 22 : MODI_AV_PGD:AV_PATCH_PGD_1D_3
 4.35%  2.133  2.133  2.133 0.000 0.00% 7 : COEF_VER_INTERP_LIN_SURF
 3.42%  1.679  1.679  1.679 0.000 0.00% 636 : MODE_READ_SURF_LFI:READ_SURFX1_LFI
 3.03%  1.486  1.486  1.486 0.000 0.00% 3 : MODI_INTERP_GRID:INTERP_GRID_1D
 2.90%  1.421  1.421  1.421 0.000 0.00% 1102 : FM_READ
 2.41%  1.182  1.182  1.182 0.000 0.00% 10 : PUT_ON_ALL_VEGTYPES
 2.12%  1.038  1.038  1.038 0.000 0.00% 7 : MODI_VER_INTERP_LIN_SURF:VER_INTERP_LIN2D_SURF
80.09% 39.291  1.038 17.207
+ sacct --format=elapsed,maxrss,averss,maxrssnode,maxrssstask -j 5417823
  Elapsed  MaxRSS  AveRSS MaxRSSNode MaxRSSTask
-----
00:00:54
00:00:50 4486420K 4486420K beaufix287 0

```

### E/ PREP en FA :

Faire du PREP en FA semble indéniable. Partir d'un fichier de surface en FA, le convertir en LFI, faire du PREP derrière pour avoir du LFI, puis le convertir en FA pour pouvoir faire de la prévision ensuite, ne semble pas approprié dans un contexte opérationnel.

Pour tester le format FA en entrée du PREP, le plus évident était juste de spécifier le format FA dans la nameliste et voir ce qui se passe :

FATAL ERROR:

-----

Value "FA " is not allowed for variable CFILETYPE

Possible values are:

" "

"GRIB "

"MESONH"

"ASCII "

"LFI "

-----

----- FATAL ERROR in SURFEX -----

-----

-  
TEST\_NAM\_VAR\_SURF: (1) CHARACTER VALUE NOT ALLOWED

La routine read\_nam\_prep\_surfn.F90 a été modifiée pour accepter le format FA :

```
!CALL TEST_NAM_VAR_SURF(ILUOUT,'CFILETYPE', CFILETYPE, ' ', 'GRIB ',MESONH',ASCII ',LFI ')  
CALL TEST_NAM_VAR_SURF(ILUOUT,'CFILETYPE', CFILETYPE, ' ', 'GRIB ',MESONH',ASCII ',LFI ',FA ')  
!CALL TEST_NAM_VAR_SURF(ILUOUT,'CFILEPGDTYPE', CFILEPGDTYPE, ' ', 'GRIB ',MESONH',ASCII ',LFI ')  
CALL TEST_NAM_VAR_SURF(ILUOUT,'CFILEPGDTYPE', CFILEPGDTYPE, ' ', 'GRIB ',MESONH',ASCII ',LFI ',FA ')
```

Deuxième essai, deuxième plantage :

```
STOP IN READ_PREP_FILE_DATE  
FILETYPE =FA NOT SUPPORTED  
HPROGRAM FA IO_INIT HACTION==READ 19 PGD.fa
```

La routine read\_prep\_file\_date.F90 a aussi été modifiée pour accepter le format FA :

```
81 ELSE IF(HFILETYPE=='FA ') THEN  
82 !  
83 HREC = CREC  
84 IREC = NREC  
85 CALL OPEN_AUX_IO_SURF(HFILE,HFILETYPE,'FULL ')  
86 CALL READ_SURF(HFILETYPE,'DTCUR ',TPTIME,IRESP)  
87 CALL CLOSE_AUX_IO_SURF(HFILE,HFILETYPE)  
88 CREC = HREC  
89 NREC = IREC
```

Après modification, encore un plantage dans le PREP :

```
ABORT! 1 abort by abor1_sfx  
[myproc#1,tid#1,pid#55875]: PREP ,  
[myproc#1,tid#1,pid#55875]: INIT_PGD_SURF_ATM ,  
[myproc#1,tid#1,pid#55875]: INIT_SURF_ATM_N ,  
[myproc#1,tid#1,pid#55875]: READ_SURF_ATM_DATE ,  
[myproc#1,tid#1,pid#55875]: READ_PREP_FILE_DATE ,  
[myproc#1,tid#1,pid#55875]: OPEN_AUX_IO_SURF ,  
[myproc#1,tid#1,pid#55875]: OPEN_AUX_IO_SURF_FA ,  
[myproc#1,tid#1,pid#55875]: MODI_READ_SURF:READ_SURFN0 ,
```

```
[myproc#1,tid#1,pid#55875]: MODE_READ_SURF_ASC:READ_SURFN0_ASC,  
[myproc#1,tid#1,pid#55875]: ERROR_READ_SURF_ASC ,
```

```
----- FATAL ERROR in SURFEX -----  
-----  
-----  
- - - - -
```

```
READ_SURF_ASC: ERROR WHEN READING ARTICLE DIM_FULL
```

Cela soulève une question : pourquoi on passe dans de la lecture du ASCII (READ\_SURFN0\_ASC) ?

Le problème vient de la routine open\_aux\_io\_surf\_fa.F90 qui fait de la lecture d'ASCII et non du FA comme le laisserait supposer son nom !!

```
USE MODD_IO_SURF_ASC,ONLY:NUNIT,CFILEIN,CFILEOUT,NMASK,NLUOUT,NFULL,CMASK  
CALL GET_LUOUT('ASCII ',NLUOUT)  
!  
NUNIT=9  
!  
OPEN(UNIT=NUNIT,FILE=HFILE,FORM='FORMATTED')  
!  
CMASK = HMASK  
CALL READ_SURF('ASCII ','DIM_FULL',ILU,IRET)  
NFULL = ILU  
!  
IL = NFULL  
CALL GET_TYPE_DIM_n(HMASK,IL)  
ALLOCATE(IMASK(IL))  
CALL GET_SURF_MASK_n(HMASK,IL,IMASK,NFULL,NLUOUT)  
!
```

Avec des modifications de type :

```
USE MODD_IO_SURF_FA, ONLY : NUNIT_FA, CFILEIN_FA, NMASK, NLUOUT, NFULL, CMASK  
CALL GET_LUOUT('FA ',NLUOUT)  
CALL FAITOU(IREP,NUNIT_FA,.TRUE.,CFILEIN_FA,'OLD',.TRUE.,.FALSE.,2,0,INBARI, 'CADRE')  
CALL READ_SURF('FA ', 'DIM_FULL',ILU,IRET)
```

On fait un peu mieux mais on plante pour un champ manquant (en fait il le cherche avec le mauvais préfixe) :

```
>>>process_options(): DR_HOOK_OPT="WALLPROF"  
///// LFIUV - Nom='PGD.fa'  
///// LFIUV - Nom SYSTEME='/scratch/utmp/slurm/mrpe731.5340625/PGD.fa'  
///// LFIUV - KREP= 0, KNUMER= 19, LDNOMM= T, CDSTTO='OLD', LDERFA= T, LDIMST= F, KNIMES= 2,  
KNBARP= 7 KNBARI= 114  
///// LFIUV - Unite 19 OUVERTE, derniere Modification OK a 20130702_120350, 114 Articles de  
donnees,233189376 mots en tout  
///// LFINFO - KREP= 0, KNUMER= 19, CDNOMA='CADRE-DIMENSIONS', KLONG= 5,KPOSEX= 9217  
///// LFILEC - KREP= 0, KNUMER= 19, CDNOMA='CADRE-DIMENSIONS', KLONG= 5  
///// LFINFO - KREP= 0, KNUMER= 19, CDNOMA='CADRE-FRANKSCHMI', KLONG= 4,KPOSEX= 9222  
///// LFILEC - KREP= 0, KNUMER= 19, CDNOMA='CADRE-FRANKSCHMI', KLONG= 4
```

```

///// LFINFO - KREP= 0, KNUMER= 19, CDNOMA='CADRE-REDPOINPOL', KLONG= 1536,KPOSEX= 9226
///// LFILEC - KREP= 0, KNUMER= 19, CDNOMA='CADRE-REDPOINPOL', KLONG= 1536
///// LFINFO - KREP= 0, KNUMER= 19, CDNOMA='CADRE-SINLATITUD', KLONG= 18,KPOSEX= 10762
///// LFILEC - KREP= 0, KNUMER= 19, CDNOMA='CADRE-SINLATITUD', KLONG= 18
///// LFINFO - KREP= 0, KNUMER= 19, CDNOMA='CADRE-FOCOHYBRID', KLONG= 5,KPOSEX= 10780
///// LFILEC - KREP= 0, KNUMER= 19, CDNOMA='CADRE-FOCOHYBRID', KLONG= 5
///// LFICAS - KREP= 0, KNUMER= 19, CDNOMA='PGD.SURFEX.FA', KLONG= 1, KPOSEX= 10785, LDAVAN= F
///// LFINFO - KREP= 0, KNUMER= 19, CDNOMA='DATE-DES-DONNEES', KLONG= 11,KPOSEX= 10786
///// LFILEC - KREP= 0, KNUMER= 19, CDNOMA='DATE-DES-DONNEES', KLONG= 11
///// FAINOC - KRANG= 1
///// FAITOU - KREP= 0, KNUMER= 19, LDNOMM= T, CDSTTU=' OLD', LDERFA= T, LDIMST= F, KNIMES= 2,
KNBARP= 0 KNBARI= 114
///// FAITOU - CDNOMC='CADRE'
**** LFILEC - KREP= -20, KNUMER= 19, CDNOMA='FULLDIM_FULL', KLONG= 1
**** LFILEC - ARTICLE "FULLDIM_FULL" NON TROUVE, UNITE 19 ****

```

Il se trouve que dans MODE\_READ\_SURF\_FA:READ\_SURFN0\_FA on a des choses de type :

```

YNAME=TRIM(YMASK)//TRIM(HREC)
CALL FALIT_I(KRESP,NUNIT_FA,YNAME,KFIELD)

```

Donc on recorrige read\_prep\_file\_date.F90 avec :

```

85 CALL OPEN_AUX_IO_SURF(HFILE,HFILETYPE,'SFX. ')

```

Après correction on tombe sur un autre problème :

```

///// LFILEC - KREP= 0, KNUMER= 19, CDNOMA='SFX.DIM_FULL', KLONG= 1
///// FALAIS - KREP= 0, KNUMER= 19, CDNOMA='SFX.DIM_FULL ', KLONGD= 1
**** LFILEC - KREP= -20, KNUMER= 19, CDNOMA='SFX.DTCUR%TDATE', KLONG= 3
**** LFILEC - ARTICLE "SFX.DTCUR%TDATE" NON TROUVE, UNITE 19 ****

```

Pour dépasser ce problème, on fait une modification dans mode\_read\_surf\_fa.F90 :

```

!!YNAME=TRIM(YMASK)//TRIM(HREC)//'%TDATE'
YNAME=TRIM(YMASK)//TRIM(HREC)

```

Et le PREP plante pour un problème de longueur de l'article en question :

```

**** LFILEC - KREP= -21, KNUMER= 19, CDNOMA='SFX.DTCUR', KLONG= 3
**** LFILEC - ARTICLE "SFX.DTCUR" + *LONG* QUE DEMANDE, UNITE 19 ****

```

## F/ Compactage :

Une des premières chose à faire était de dresser la liste des champs SURFEX qui se prêtent mieux à la compression, les caractériser pour voir ceux qui sont 'continus' sans discontinuité, sans grand pic par endroits, sans grande variabilité localement. Certains champs ne doivent pas être compactés (index, mask, ...). Il s'agit d'un premier travail de sélection, tracer les champs est une bonne méthode pour voir leur tête.

En partant d'un fichier de surface Arome récent (2013/11/08), on s'est rendu compte que le calcul de certains champs était certainement erroné.



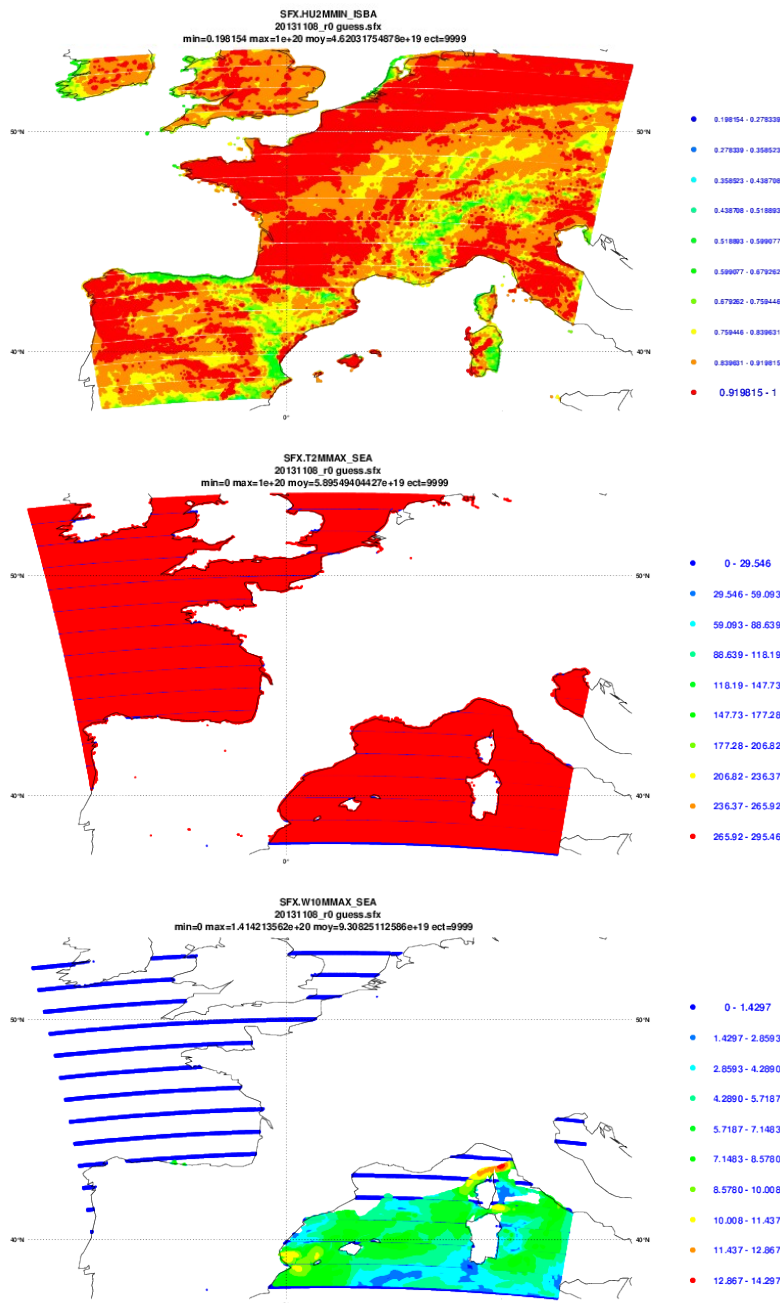


Figure 2 : Tracés de quelques champs bugués

Tous les tracés sont sur le webdav : <http://webdav/public/proc/jidane/>

Les champs bugués sont tous ceux qui sont un min ou un max de paramètre.

Après vérification, ils ne l'étaient pas dans la version opérationnelle précédente, donc soit un problème de code, soit de format avec le passage au format FA.

De toute évidence ces champs-là n'ont aucun intérêt ni pour le PREP ni pour la prévision.

On peut demander la non écriture de ces champs dans le fichier de sortie SURFEX en faisant intervenir la nameliste NAM\_WRITE\_DIAG\_SURF<sub>n</sub>. Cela réduira en plus la taille du fichier en sortie.

```

&NAM_WRITE_DIAG_SURFn
  LPROVAR_TO_DIAG =.FALSE.,
  LSELECT          =.TRUE.,
  CSELECT(1) ='XX', 'YY', 'DX', 'DY', 'SST', 'Z0SEA', 'TS_WATER', 'Z0WATER', 'TG1',
'TG2', 'TG3', 'WG1', 'WG2', ...
/

```

On doit garder LPROVAR\_TO\_DIAG à .FALSE. pour avoir les champs par tile, en mettant LSELECT à .TRUE. on dit qu'on veut sélectionner les champs en sortie et CSELECT contient la liste des champs qu'on veut garder.

Une prévision Arome (expérience 68BJ) a pu être faite en faisant une sélection sur les champs à sortir dans le fichier de surface en activant LSELECT dans NAM\_WRITE\_DIAG\_SURF*n*, mais pour que ça marche il fallait modifier aro\_surf\_diag.F90 qui mettait en dur la sélection à FALSE.

```

!CALL FLAG_UPDATE(.FALSE.,.TRUE.,.FALSE.,.FALSE.)
CALL FLAG_UPDATE(.FALSE.,.TRUE.,LPROVAR_TO_DIAG,LSELECT)

```

Ci-dessous la liste des champs non sélectionnés dans les sorties de surface :

```

SFX.T2MMIN_SEA
SFX.T2MMAX_SEA
SFX.HU2MMIN_SEA
SFX.HU2MMAX_SEA
SFX.W10MMAX_SEA
SFX.W10MMAX_WAT
SFX.T2MMIN_ISBA
SFX.T2MMAX_ISBA
SFX.HU2MMIN_ISBA
SFX.HU2MMAX_ISBA
SFX.T2M
SFX.T2MMIN
SFX.T2MMAX
SFX.Q2M
SFX.HU2M
SFX.HU2MMIN
SFX.HU2MMAX
SFX.ZON10M
SFX.MER10M
SFX.W10M
SFX.W10MMAX

```

Le gain dans la taille des fichiers de sortie de SURFEX est de 15.96% (la taille passe de 1028333568 à 864165888)

Il faudra réfléchir à ce qu'il faudra mettre comme champs au minimum dans ces fichiers pour que le PREP et la prévision fonctionnent sans problème.

Rien de plus n'a été fait au sujet du compactage.

### 3/ Conclusion :

Le PREP a été testé durant ce stage pour différentes configurations.

La parallélisation OpenMP de la routine HOR\_EXTRAPOL\_SURF a permis une exécution 10 à 15 fois plus rapide du PREP.

Le PREP de la grille Arpège à la grille Arome ne marche pas pour le moment suite à des problèmes de différences de schéma de neige, probablement dû à des différences de version de code (prévi Arpège / PREP).

Le PREP ne gère que les fichiers au format LFI ce qui impose de passer par des convertisseurs en entrée et en sortie. Cela semble être une entrave à une utilisation opérationnelle, d'où la nécessité de coder dans le PREP l'utilisation du format FA en version simplifiée.

Une tentative pour lire le format FA en entrée du PREP a été initiée durant ce stage mais sans grand succès, il reste un certain nombre de modifications à faire.

Le compactage des champs de SURFEX dans les fichiers FA n'a pas pu être abordé concrètement dans ce stage faute de temps. Mais l'idée de sélectionner les champs à écrire dans les fichiers de surface, pour réduire leur taille, s'est concrétisée.

Il reste à préciser la liste des champs nécessaires pour le bon fonctionnement du PREP et de la prévision.

## **ANNEXE :**

Chemin de mon pack sur beaufix : /home/gmap/mrpe/jidanem/pack/40\_surfex73\_dev/

### Liste des routines modifiées :

mse/externals/aro\_surf\_diag.F90  
mse/module/sfxflddesc\_mod.F90  
mse/programs/sfxlfi2fa.F90  
mse/programs/prep.F90  
surfex/SURFEX/read\_pgd\_isba\_parn.F90  
surfex/SURFEX/mode\_read\_surf\_fa.F90  
surfex/SURFEX/hor\_extrapol\_surf.F90  
surfex/SURFEX/test\_record\_len.F90  
surfex/SURFEX/mode\_read\_extern.F90  
surfex/SURFEX/prep\_grid\_gauss.F90  
surfex/SURFEX/hor\_interpol\_gauss.F90  
surfex/SURFEX/read\_nam\_prep\_surfn.F90  
surfex/SURFEX/read\_prep\_file\_date.F90  
surfex/SURFEX/open\_aux\_io\_surf\_fa.F90

### Prototype d'un job pour faire du PREP de la grille 1.3 vers la grille 2.5 :

/home/gmap/mrpe/jidanem/surfex/prep/job\_prep7.sh

### Expériences OLIVE :

68BJ : Préviation Arome en activant la sélection sur les champs à écrire dans le fichier de surface.  
7BVR : Préviation Pégase (ref) (mon binaire et CouplingSurf depuis fichier Arpège).  
7BVE : Préviation Pégase (mod) (mon binaire et PREP depuis analyse Aladin Réunion).