

# Tests et évaluations du FULLPOS/PREP

Mohamed JIDANE : DSI/SCCO, DMN, MAROC  
Encadrement : Françoise TAILLEFER, CNRM/GMAP  
Novembre 2014

## 1/ INTRODUCTION

Ce travail s'inscrit dans la continuation du test et de validation du PREP.

Son objectif consiste à évaluer et tester les dernières modifications qui ont été apportées au PREP, FULLPOS et MSE afin d'être en mesure de faire les interpolations de PREP en utilisant Fullpos (voir annexe).

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.

## 2/ ENVIRONNEMENT DE TESTS

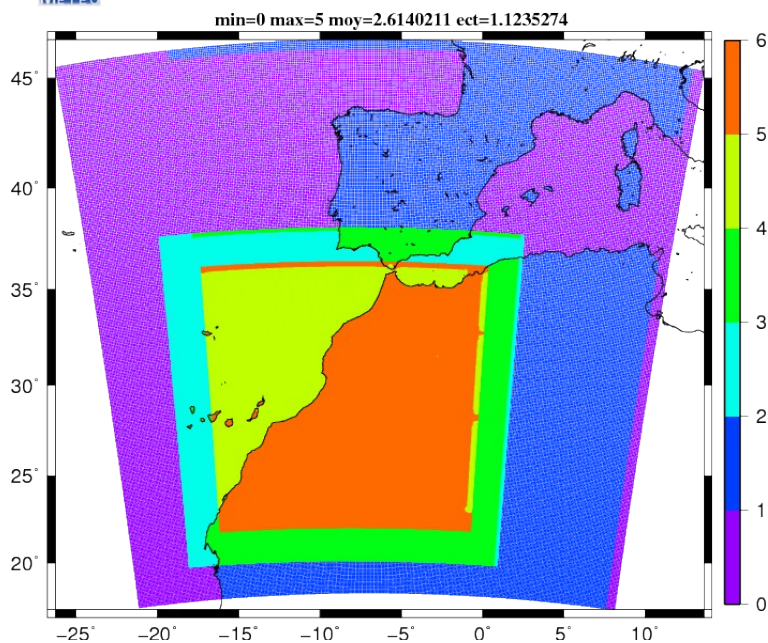
Les tests ont été faits avec le cycle 41\_bf.01 en plus des dernières modifications de Philippe Marguinaud concernant Fullpos/Prep.

Ce cycle contient la dernière version de Surfex 7.3.

On a préféré faire les tests sur les domaines opérationnels du Maroc, donc on a commencé par créer les domaines suivants : ALADMAR(10km, 320x320), AROMAR(2.5km, 800X800) et ALADMAR(5km, 320X320) (domaine fictif pour tester le passage d'une grille fine vers une grille lâche) .



## Les domaines de test



Des tests ont aussi été faits pour passer des grilles de Gauss (T798 et T1198) vers la grille Arome France (1.3km, 1536X1440) et vers les grilles de tests mentionnées auparavant.

### 3/ EXPERIENCES : LAM to LAM

De nouveaux paramètres ont été ajoutés à la nameliste **NAMFPC** pour faire des choix concernant la méthode d'interpolation voulue par Fullpos\_Prep :

<b>NFPSFXINT</b>	<ul style="list-style-type: none"> <li>• 0 = 4-point interpolations (default)</li> <li>• 1 = average over Fullpos halo</li> <li>• 2 = take nearest point</li> </ul>
------------------	---

Pour valider les interpolations faites par Fullpos\_Prep, P. Marguinaud nous a fait un PREP qui lui aussi fait les interpolations sur 4 points.

Pour faire un PREP, on passe par les étapes suivantes :

Transformation des fichiers FA au format LFI :

```
PACK=/home/gmap/mrpm/marguina/pack/41_bf_PREP_dbg.01.I1314183INTELMPI411036MT.x
```

```
ln -sf $WORKDIR/PGD_ALD_MAROC.fa Const.Clim.sfx # PGD grille de départ
ln -sf $WORKDIR/PGD_AROME_MAROC.fa const.clim.sfx.AREA # PGD grille d'arrivée
ln -sf $WORKDIR/p0_2014110400.sfx ICMSHFPOSINIT.sfx # Fichier init de surface

$PACK/bin/SFXTOOLS sfxfa2lfi --sfx-fa--file Const.Clim.sfx --sfx-lfi-file PGD1.lfi
$PACK/bin/SFXTOOLS sfxfa2lfi --sfx-fa--file ICMSHFPOSINIT.sfx --sfx-lfi-file PREP1.lfi
$PACK/bin/SFXTOOLS sfxfa2lfi --sfx-fa--file const.clim.sfx.AREA --sfx-lfi-file PGD2.lfi
```

La nameliste à utiliser pour PREP est de type :

```
&NAM_FILE_NAMES
  HPGDFILE='PGDFILE',
  CINIFILE='INIT_SURF',
/
&NAM_IO_OFFLINE
  LPRINT = T,
  CFORCING_FILETYPE = 'NETCDF',
  CSURF_FILETYPE = 'LFI',
  CTIMESERIES_FILETYPE = 'NETCDF',
  LSET_FORC_ZS = T,
  CPREPPFILE = 'PREP2',
  CPGDFILE = 'PGD2',
/
&NAM_PREP_ISBA
  LISBA_CANOPY=.TRUE.,
/
&NAM_PREP_SEAFLUX
  LSEA_SBL=.FALSE.,
  CFILE_SEAFLX='',
  CTYPE_SEAFLX='',
  CFILEPGD_SEAFLX='',
  CTYPEPGD='',
/
&NAM_PREP_SURF_ATM
  CFILETYPE = 'LFI',
  CFILE = 'PREP1',
  CFILEPGDTYPE = 'LFI',
  CFILEPGD = 'PGD1',
/
&NAM_PREP_WATFLUX
  LWAT_SBL=.FALSE.,
/
&NAM_WRITE_SURF_ATM
  LNOWRITE_TEXFILE=.TRUE.,
/
```

Pour transformer le fichier PREP2.lfi, en sortie du PREP, au format FA, on fait :

```
$PACK/bin/lfitools faempty const.clim.sfx.AREA PREP2.fa
$PACK/bin/SFXTOOLS sfxlfi2fa --sfx-fa--file PREP2.fa --sfx-lfi-file PREP2.lfi
```

A/ Grille lâche vers grille fine :

En essayant de faire un Fullpos\_Prep de la grille Aladin Maroc (10km) vers la grille Arome Maroc (2,5km), le Fullpos\_Prep plante pour missing field dans le fichier INIT :

```
ABORT!      1 RDFA2GP: FIELD IS MISSING :
```

Les champs manquants, et qui sont nécessaires pour le setup de Fullpos et qu'il faut ajouter, sont les suivants :

```
'SURFRESERV.INTER',
'SURFIND.VEG.DOMI',
'SURFRESI.STO.MIN',
'SURFZO.FOIS.G',
'SURFALBEDO',
'SURFIND.FOLIAIRE',
```

```
'SURFEMISSIVITE',  
'SURFRES.EVAPOTRA',  
'SURFGZO.THERM'
```

Pour ajouter les différents champs, un par un, on peut utiliser le script suivant :

```
ulimit -s unlimited  
if [ $# -ne 1 ] ; then  
    echo "Usage: $0 File_name Field_to_add"  
    exit 2  
fi  
file=$1  
suffix=$(echo $2 | cut -c1-4)  
prefix=$(echo $2 | cut -c5-)  
  
cat<<EOF>tempo$$  
OUV  
77 T $file OLD T T 2 0 c  
RAZ  
IENC  
77 $suffix 0 $prefix F  
IRME  
77 T  
EOF  
  
lfitools testfa < tempo$$  
rm -f tempo$$
```

Pour le cas test [Aladin Maroc (10km) vers Arome Maroc (2.5km)], avec un PREP qui fait l'interpolation sur 4 points et un Fullpos\_Prep qui la fait aussi sur 4 points, les différences de champs de surface sont quasi nulles :

Le min et le max des différences, après avoir enlever les valeurs undefined :

SFX.SST.PREP-PFFPOS:	N = 225680	<-19.6007/1.9777>	<19.9781/37.639>	<0/0>
X001TG1.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<0/0>
X001TG2.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-1e-07/0>
X001TG3.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<0/0>
X001WG1.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-1e-10/1e-10>
X001WG2.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-1e-10/1e-10>
X001WG3.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-1e-10/1e-10>
X001WGI1.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-1e-11/1e-11>
X001WGI2.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-1e-11/1.6e-13>
X001WGI3.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<0/0>
X001WR.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-1e-12/1e-13>

Ceci valide, à mon sens, les interpolations 4 points faites par Fullpos\_Prep.

Si on compare l'interpolation par défaut de PREP (9points + extrapolation si nécessaire) avec celle de Fullpos\_Prep (4 points), les différences ne sont pas si fulgurantes (sauf pour RSN\_VEG1 et WSN\_VEG1 sur une petite zone du sud de l'Espagne) et sont concentrées sur les côtes et les reliefs.

SFX.SST.PREP-PFFPOS:	N = 225680	<-19.6007/1.9777>	<19.9781/37.639>	<-0.248958/0.311841>
SFX.TCANYON.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.292517/0.38759>
SFX.TI_ROAD.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.292517/0.38759>
SFX.TROAD1.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191606/0.11764>
SFX.TROAD2.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191606/0.11764>
SFX.TROAD3.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191606/0.11764>
SFX.TROAD4.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191606/0.11764>
SFX.TROAD5.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191606/0.11764>
SFX.TROOF1.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191605/0.117639>
SFX.TROOF2.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191606/0.117639>
SFX.TROOF3.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191605/0.11764>
SFX.TROOF4.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191605/0.11764>
SFX.TROOF5.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191605/0.11764>
SFX.TSRAD_NAT.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-0.473983/9.5108>
SFX.TS_WATER.PREP-PFFPOS:	N = 411	<-16.4295/1.6258>	<20.6591/37.6009>	<-0.904972/0.337679>
SFX.TWALL1.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191606/0.117639>
SFX.TWALL2.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191606/0.117639>
SFX.TWALL3.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191605/0.11764>
SFX.TWALL4.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191605/0.11764>
SFX.TWALL5.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191605/0.11764>
SFX.T_WIN1.PREP-PFFPOS:	N = 829	<-15.4074/1.5545>	<20.4785/37.6407>	<-0.191605/0.11764>
X001ASN_VEG.PREP-PFFPOS:	N = 83	<-3.522/-3.1814>	<36.9827/37.1418>	<-0.674781/0>
X001RSN_VEG1.PREP-PFFPOS:	N = 83	<-3.522/-3.1814>	<36.9827/37.1418>	<-300/0>
X001WSN_VEG1.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-144.656/0>
X001TG1.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-0.473983/9.5108>
X001TG2.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-1.41294/2.72343>
X001TG3.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-1.41929/2.73654>
X001WG1.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-0.187754/0.123808>
X001WG2.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-0.141393/0.12255>
X001WG3.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-0.141597/0.122593>
X001WGI1.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-0.0269629/1e-11>
X001WGI2.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-0.000154802/1e-13>
X001WGI3.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<0/0>
X001WR.PREP-PFFPOS:	N = 373976	<-18.1625/1.9008>	<20.065/37.6407>	<-0.00312713/1.33127e-06>

Les tracés de champs et de leurs différences sont sur le webdav :

<http://webdav/public/proc/jidane/>

(Pour une meilleure lisibilité, les tracés ont été faits en enlevant les valeurs quasi nulles [-0.001, 0.001]).

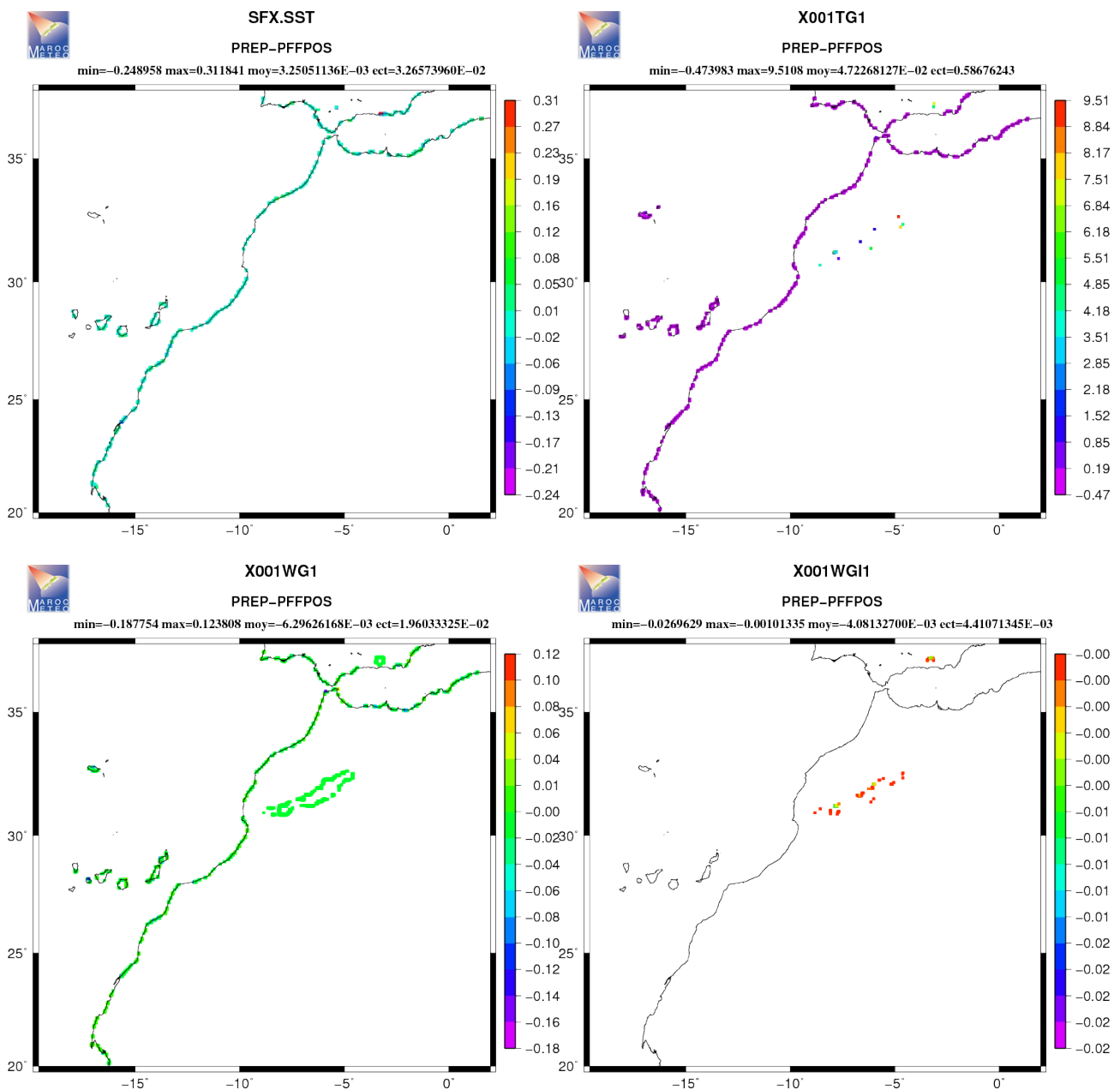


Fig 2 : Différences de quelques champs de surface entre le PREP et le Fullpos\_Prep

Pour apprécier l'impact des écarts de champs sur les fichiers de surface entre le PREP et Fullpos\_Prep, des prévisions jusqu'à 36h d'échéances ont été tournées et des différences de champs ont été tracées (voir sur le webdav).

Les prévisions plantent au départ avec le cycle 41, dans ENDPOS, si on demande dans le Fullpos l'interpolation de la divergence sur les niveaux isothermes :

```

MPL_ABORT: THRD      2  ENDPOS : INTERNAL ERROR : DIV/ISKP
[myproc#23,tid#1,pid#17655]: CNT4 ,#1,st=1,wall=0.000s/0.000s
[myproc#23,tid#1,pid#17655]: DYNFPOS ,#1,st=1,wall=0.000s/0.000s
[myproc#23,tid#1,pid#17655]: STEPO_FPOS ,#4,st=1,wall=26.504s/0.000s
[myproc#23,tid#1,pid#17655]: ENDVPOS ,#2,st=1,wall=7.955s/0.039s
  
```

```
[myproc#23,tid#1,pid#17655]: ENDPOS ,#68,st=1,wall=7.915s/0.039s
[myproc#23,tid#2,pid#17655]: ENDPOS ,#68,st=1,wall=7.833s/0.043s
```

Une correction a été portée à arpifs/setup/suafn2.F90 pour interdire l'interpolation de la divergence sur les niveaux isothermes :

```
diff main/arpifs/setup/suafn2.F90 local/arpifs/setup/suafn2.F90
289c289
< CALL CTRL_TFP(TFP_DIV,I,' ',LLSF,TFP_DYND,IMAX)
---
> CALL CTRL_TFP(TFP_DIV,I,'KD ',LLSF,TFP_DYND,IMAX)
```

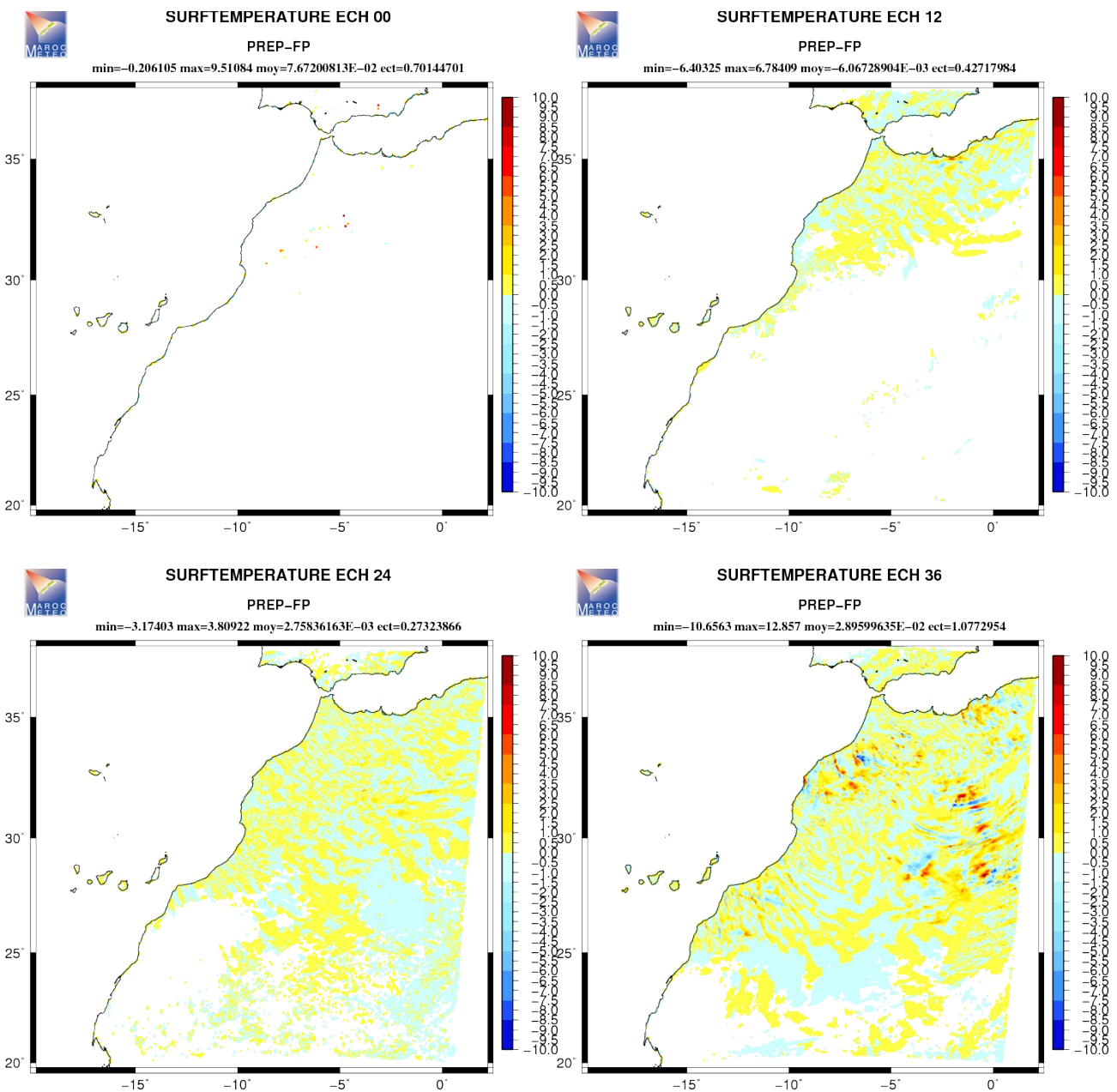


Fig 3 : Evolution des différences de la température de surface entre une prévision qui démarre du fichier de sortie du PREP et celle qui utilise celui qui sort du Fullpos\_Prep

## B/ Grille fine vers grille lâche :

En essayant de faire un Fullpos\_Prep d'une grille fine (Arome 2.5km) vers une grille Aladin 5km, le Fullpos\_Prep plante aussi pour missing field dans le fichier INIT :

```
ABORT!      1 RDFA2GP: FIELD IS MISSING :
```

Les champs manquants, et qu'il faut ajouter, sont les suivants :

```
'PROFTEMPERATURE',  
'PROFRESERV.EAU',  
'PROFRESERV.GLACE',  
'SURFRESERV.EAU',  
'SURFRESERV.INTER',  
'SURFRESERV.GLACE',  
'SURFZO.FOIS.G',  
'SURFALBEDO',  
'SURFEMISSIVITE',  
'SURFET.GEOPOTENT',  
'SURFPROP.VEGEATAT',  
'SURFVAR.GEOP.ANI',  
'SURFVAR.GEOP.DIR',  
'SURFPROP.ARGILE',  
'SURFPROP.SABLE',  
'SURFEPAIS.SOL',  
'SURFIND.VEG.DOMI',  
'SURFRESI.STO.MIN',  
'SURFIND.FOLIAIRE',  
'SURFRES.EVAPOTRA',  
'SURFGZO.THERM'
```

L'interpolation par défaut sur 4 points du Fullpos\_Prep plante après ajout de ces champs :

```
[myproc#1,tid#1,pid#38926]: PREP_HOR_SEAFLUX_FIELD ,#6,st=1,wall=0.013s/0.000s  
[myproc#1,tid#1,pid#38926]:   HOR_INTERPOL ,#2,st=1,wall=0.001s/0.000s  
[myproc#1,tid#1,pid#38926]:     HOR_INTERPOL_BUFFER ,#2,st=1,wall=0.001s/0.000s  
[myproc#1,tid#1,pid#38926]:       ABOR1_SF_X ,#1,st=1,wall=0.000s/0.000s
```

```
----- FATAL ERROR in SURFEX -----  
-----  
-----  
-  
HOR_INTERPOL_BUFFER: UNEXPECTED UNDEFINED VALUES
```

En effet, quand on passe d'une grille fine vers une grille lâche, les quatre points de la grille fine qui entourent un point X de la grille lâche ne contiennent pas forcément une information sur tout les types de nature que représente ce point X.

Pour remédier à ce problème, on choisi de faire plutôt une interpolation sur le halo de Fullpos (d'une taille par défaut de 3). La taille du halo dépendra du rapport des résolutions (grille de départ/grille cible).

Les tracés de champs et de leurs différences sont sur le webdav.



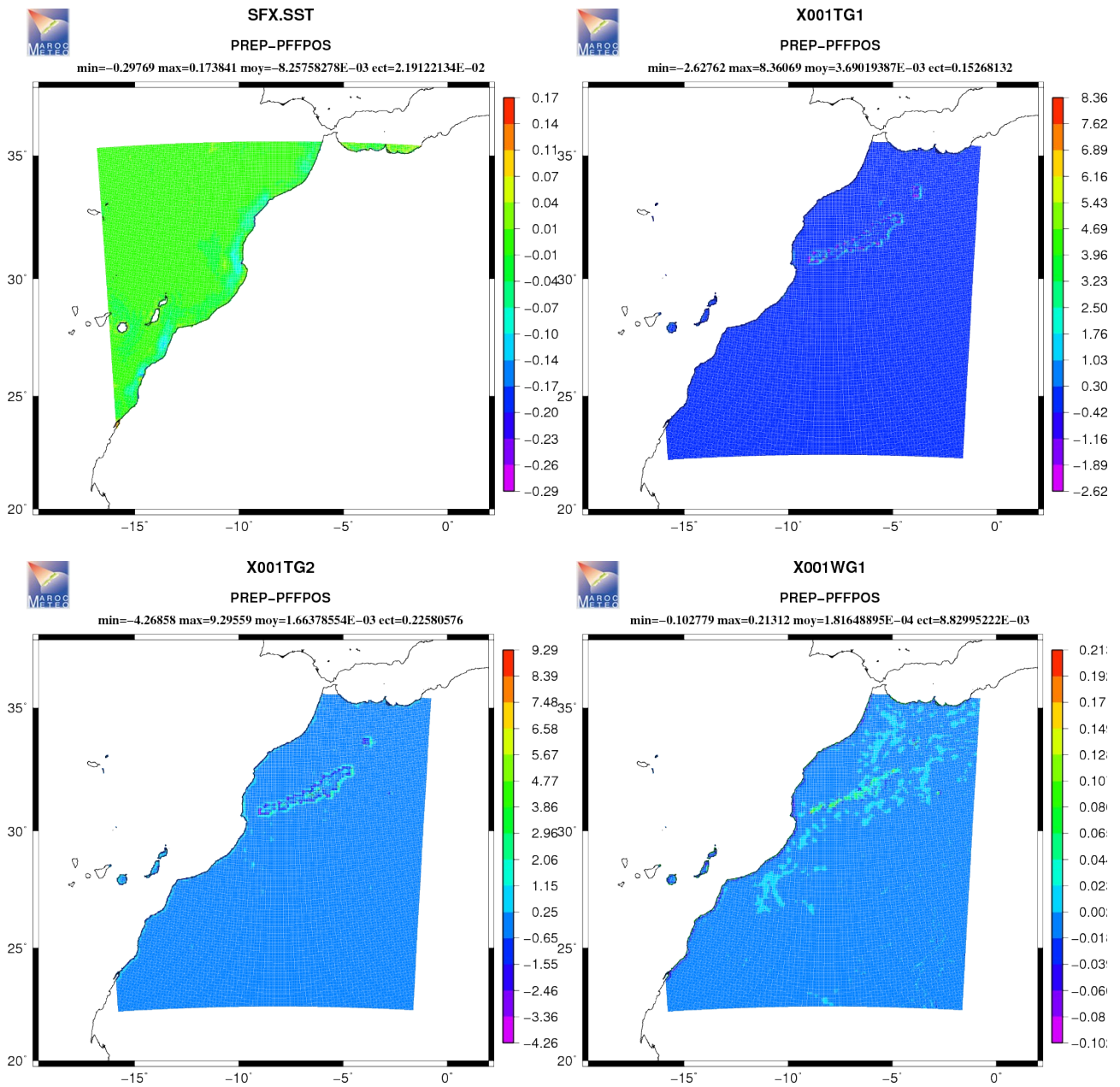


Fig 4 : Différences de quelques champs de surface entre le PREP (9 pts) et le Fullpos\_Prep (halo de taille 3 pts) d'une grille fine (2.5km) vers une grille lâche (5km)

#### 4/ EXPERIENCES : GLOBAL to LAM

Le Fullpo\_Prep qui démarre d'une grille de Gauss vers une grille à aire limitée (Arome Maroc 2.5km) plante :

```
SURFEX FIELD CACHE:IMAX WAS NOT FOUND
ABORT!      1 SURFEX FIELD CACHE:IMAX WAS NOT FOUND
MPL_ABORT:  THRD      1      SURFEX FIELD CACHE:IMAX WAS NOT FOUND
[myproc#1,tid#1,pid#95367]:  MASTER , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#95367]:  CNT0  , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#95367]:  SU0YOMA , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#95367]:  SUFPC  , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#95367]:  SUFPCSF , #1, st=1,
wall=0.000s/0.000s
[myproc#1,tid#1,pid#95367]:  PREP_STEP0 , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#95367]:  PREP_STEPX , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#95367]:  AROINI_SURFB , #1, st=1,
wall=0.000s/0.000s
```

Pour régler ce problème, une correction a été portée à la routine `mse/externals/aroini_surfb.F90` pour ne lire les champs IMAX et JMAX que dans le cas 'CONF PROJ' :

```
IF (CLGRID == 'CONF PROJ') THEN
  YRECFM='IMAX'
  CALL SURFEX_FIELD_BUF_GET (YSURFEX_CACHE_IN, NIMAX_II, YRECFM)
  YRECFM='JMAX'
  CALL SURFEX_FIELD_BUF_GET (YSURFEX_CACHE_IN, NJMAX_II, YRECFM)
ENDIF
```

Une fois ça corrigé on décide de faire un Fullpos\_Prep de la grille de Gauss vers une grille Arome avec une interpolation sur 4 points (NFPSFXINT=0).

Ce Fullpos\_Prep plante comme pour le cas grille fine vers grille lâche :

```
MPL_ABORT:  THRD      1  abort by abor1_sfx
[myproc#1,tid#1,pid#26559]:  MASTER , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:  CNT0  , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:  CNT1  , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:  CNT2  , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:  CNT3  , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:  CNT4  , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:  GRIDFPOS , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:  FP2SX2 , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:  PREP_STEP2 , #1, st=1, wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:  PREP_STEPX , #3, st=1, wall=4.609s/0.000s
[myproc#1,tid#1,pid#26559]:  PREP_SURF_ARO , #3, st=1, wall=1.913s/0.000s
[myproc#1,tid#1,pid#26559]:  PREP_SURF_ATM , #3, st=1, wall=1.913s/0.000s
[myproc#1,tid#1,pid#26559]:  PREP_SEA , #3, st=1, wall=0.011s/0.000s
[myproc#1,tid#1,pid#26559]:  PREP_SEAFLUX , #3, st=1, wall=0.011s/0.000s
```

```
[myproc#1,tid#1,pid#26559]:      PREP_HOR_SEAFLUX_FIELD
,#5,st=1,wall=0.011s/0.000s
[myproc#1,tid#1,pid#26559]:      HOR_INTERPOL ,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:      HOR_INTERPOL_BUFFER
,#1,st=1,wall=0.000s/0.000s
[myproc#1,tid#1,pid#26559]:      ABOR1_SFX ,#1,st=1,wall=0.000s/0.000s
```

Pourtant, on est dans le cas d'une grille lâche (*Gauss*) vers une grille fine (*Arome*) et donc les 4 points de la grille de départ qui entourent un point X de la grille d'arrivée devraient avoir une information sur tout les types de nature que représente ce point X.

Le problème vient du fait que la grille de *Gauss* et la grille d'*Arome* n'ont pas été faites avec les mêmes bases de données (*GTOPO30* pour *Gauss* et *GMTED2010\_30* pour *Arome*).

La seule possibilité qui restait à faire est de faire les interpolations sur un halo de taille à déterminer suivant les expériences (grille de départ, grille d'arrivée et zone d'intérêt) (15 à 16 points dans mes cas de tests).

Le fait de choisir comme méthode d'interpolation la moyenne sur un halo si grand (*NFPSLWIDE=15*, *NFPSFXINT=1*) induit un lissage automatique des champs :

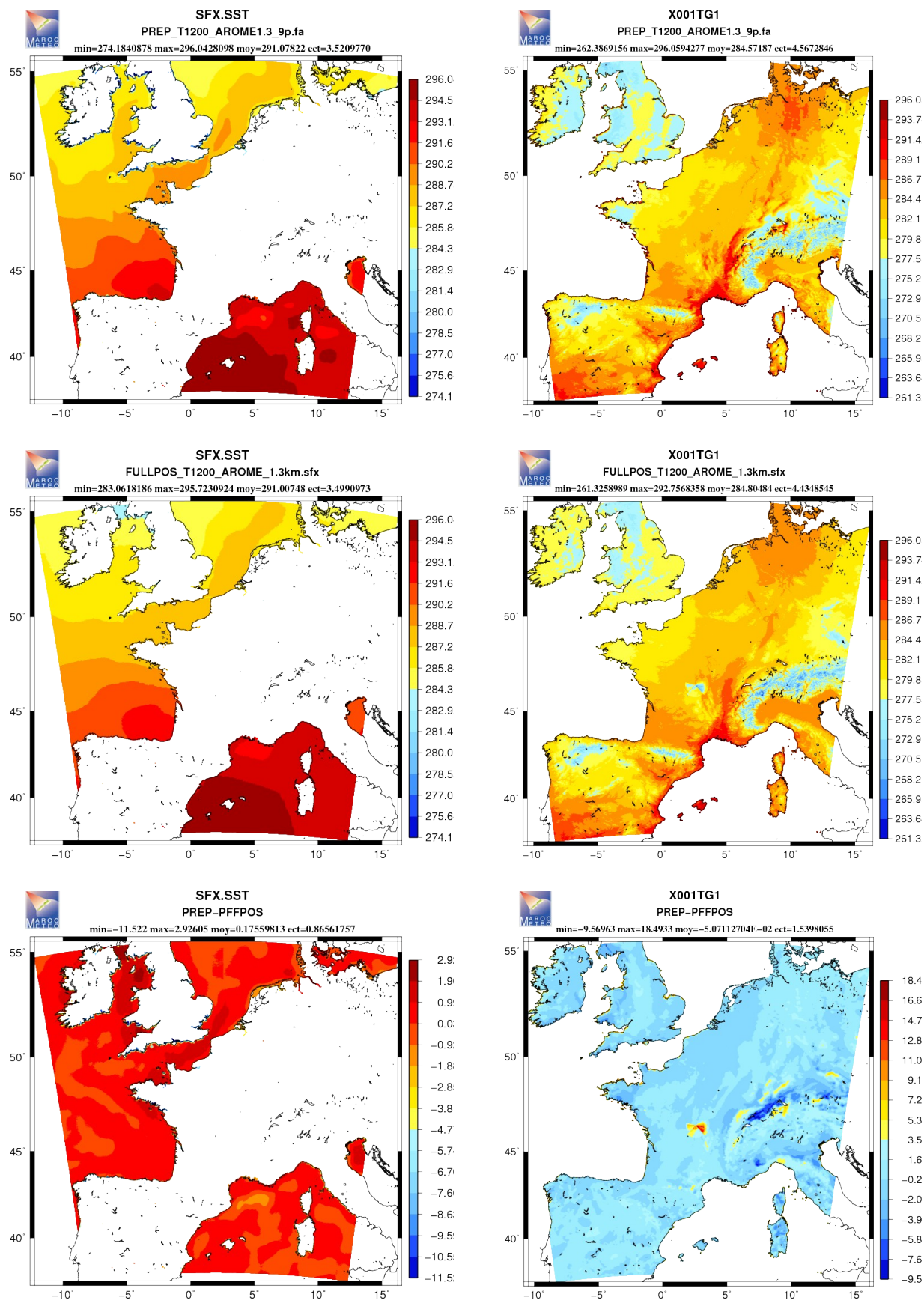


Fig 5 : La SST et la TG1 entre le PREP et le Fullpos\_Prep (moyenne sur un halo de 15 points) de la grille de Gauss T1198 vers Arome 1.3km

Pour éviter ce lissage, on opte plutôt pour avoir le point le plus proche dans le halo au lieu de la moyenne (NFPSFXINT=2).

Le traçage des champs issus du Fullpos\_Prep montre clairement un bug dans cette méthode d'interpolation, à priori en rapport avec un décalage de points par rapport au pôle de la grille de départ :

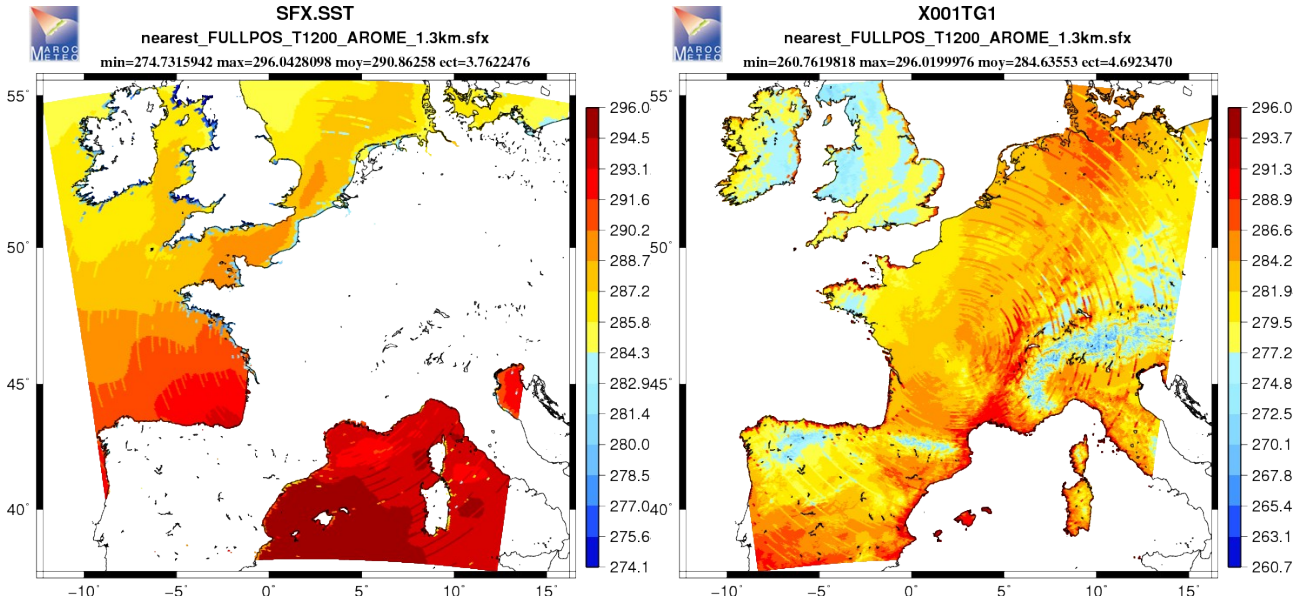


Fig 6 : La SST et la TG1 en sortie du Fullpos\_Prep (NFPSFXINT=2)

On fait le même constat même si on s'éloigne du pôle de la grille de départ :

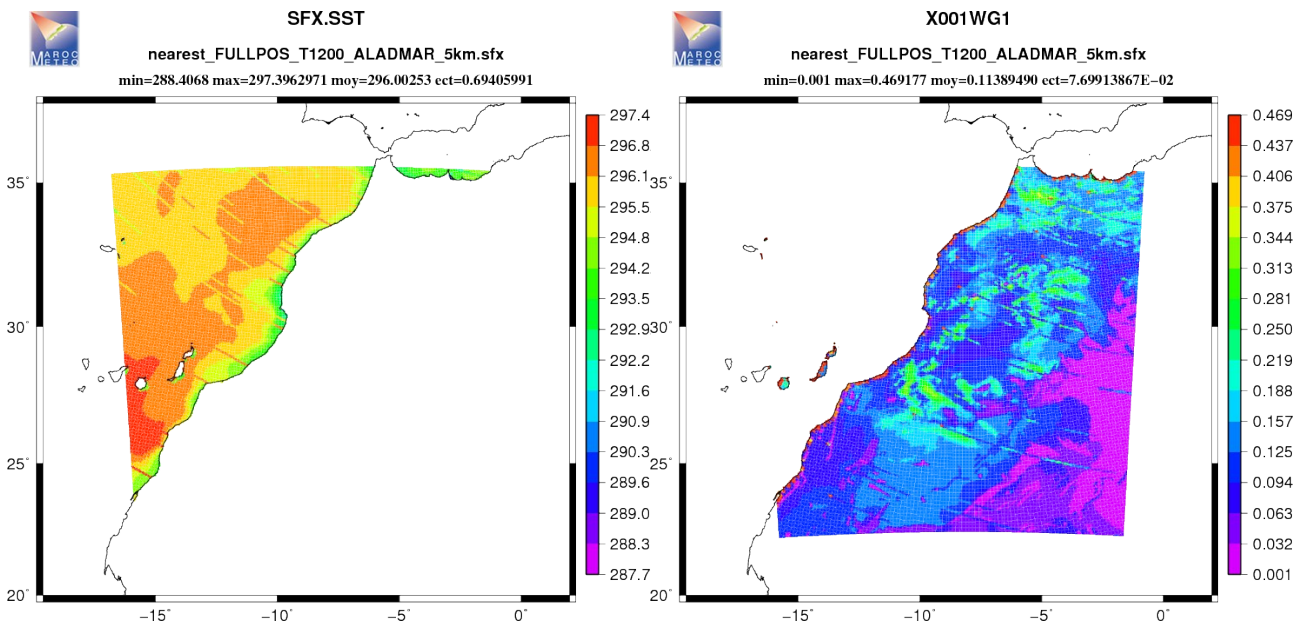


Fig 7 : La SST et la WG1 en sortie du Fullpos\_Prep (NFPSFXINT=2)



## 5/ PERFORMANCE : PREP vs Fullpos\_Prep

Grille de départ	Grille d'arrivée	Méthode d'interpolation	Wall-time
Aladin Maroc 10km (320x320)	Arome Maroc 2.5km (800x800)	PREP (4 points) (24 threads)	21.29 sec
Aladin Maroc 10km (320x320)	Arome Maroc 2.5km (800x800)	PREP (9 points) (24 threads)	24.92 sec
Aladin Maroc 10km (320x320)	Arome Maroc 2.5km (800x800)	Fullpos_Prep (4 points) (1 PROC)	42.79 sec
Aladin Maroc 10km (320x320)	Arome Maroc 2.5km (800x800)	Fullpos_Prep (4 points) (12 PROCs)	19.07 sec
Arome Maroc 2.5km (800x800)	Aladin Maroc 5km (320x320)	PREP (4 points) (24 threads)	19.07 sec
Arome Maroc 2.5km (800x800)	Aladin Maroc 5km (320x320)	PREP (9 points) (24 threads)	14.31 sec
Arome Maroc 2.5km (800x800)	Aladin Maroc 5km (320x320)	Fullpos_Prep (moyenne sur un halo de taille 3 points) (1 PROC)	48.27 sec
Arome Maroc 2.5km (800x800)	Aladin Maroc 5km (320x320)	Fullpos_Prep (moyenne sur un halo de taille 3 points) (12 PROCs)	18.38 sec

Grille de départ	Grille d'arrivée	Méthode d'interpolation	Wall-time
T798	Arome 1.3 (1536X1440)	PREP 4 points (24 threads)	139.91 sec
T798	Arome 1.3 (1536X1440)	PREP 9 points (24 threads)	141.16 sec
T798	Arome 1.3 (1536X1440)	Fullpos_Prep (moyenne sur un halo de taille 15 points) (4 PROCs)	500.09 sec
T1198	Arome 1.3 (1536X1440)	PREP 4 points (24 threads)	224.92 sec
T1198	Arome 1.3 (1536X1440)	PREP 9 points (24 threads)	195.70 sec
T1198	Arome 1.3 (1536X1440)	Fullpos_Prep (moyenne sur un halo de taille 15 points) (6 PROCs)	530.60 sec
T1198	Arome 1.3 (1536X1440)	Fullpos_Prep (le point le plus proche sur un halo de taille 15 points) (4 PROCs)	352.24 sec
T798	Aladin Maroc 10km (320x320)	PREP 4 points (24 threads)	17.93 sec
T798	Aladin Maroc 10km (320x320)	PREP 9 points (24 threads)	16.72 sec

T798	Aladin Maroc 10km (320x320)	Fullpos_Prep (moyenne sur un halo de taille 16 points) (12 PROCS)	27.02 sec
T1198	Aladin Maroc 10km (320x320)	PREP 4 points (24 threads)	39.62 sec
T1198	Aladin Maroc 10km (320x320)	PREP 9 points (24 threads)	40.76 sec
T1198	Aladin Maroc 10km (320x320)	Fullpos_Prep (moyenne sur un halo de taille 15 points) (6 PROCS)	59.70 sec
T798	Arome Maroc 2.5 km (800x800)	PREP 4 points (24 threads)	62.05 sec
T798	Arome Maroc 2.5 km (800x800)	PREP 9 points (24 threads)	60.59 sec
T798	Arome Maroc 2.5 km (800x800)	Fullpos_Prep (moyenne sur un halo de taille 15 points) (12 PROCS)	85.99 sec
T1198	Arome Maroc 2.5 km (800x800)	PREP 4 points (24 threads)	113.51 sec
T1198	Arome Maroc 2.5 km (800x800)	PREP 9 points (24 threads)	113.44 sec
T1198	Arome Maroc 2.5 km (800x800)	Fullpos_Prep (moyenne sur un halo de taille 15 points) (6 PROCS)	156.36 sec
T798	Aladin Maroc 5km (320x320)	PREP 4 points (24 threads)	17.22 sec
T798	Aladin Maroc 5km (320x320)	PREP 9 points (24 threads)	17.25 sec
T798	Aladin Maroc 5km (320x320)	Fullpos_Prep (moyenne sur un halo de taille 15 points) (12 PROCS)	35.65 sec
T1198	Aladin Maroc 5km (320x320)	PREP 4 points (24 threads)	40.31 sec
T1198	Aladin Maroc 5km (320x320)	PREP 9 points (24 threads)	39.67 sec
T1198	Aladin Maroc 5km (320x320)	Fullpos_Prep (moyenne sur un halo de taille 15 points) (6 PROCS)	63.72 sec

## 6/ CONCLUSIONS :

Le Fullpos\_Prep a été testé et évalué durant ce stage pour différentes configurations.

Le Fullpos\_Prep offre la possibilité de faire les interpolations de différentes manières. Le choix par défaut de faire les interpolations sur 4 points a été validé sur le plan numérique.

Ce choix n'est pas valable pour toutes les configurations, notamment lorsque on part d'une grille fine vers une grille lâche, ou bien lorsque les deux grilles de départ et d'arrivée n'ont pas été faites avec les mêmes bases de données.

Dans ce cas-là, il faut travailler avec un halo de points dont la taille dépend, d'une part, du rapport de résolution entre la grille de départ et la grille d'arrivée, et d'autre part, du lieu de la zone d'intérêt.

Le choix de faire la moyenne sur un halo dont la taille est assez grande implique un lissage des champs de surface plus au moins important.

Le choix de prendre le point le plus proche dans le halo, au lieu de la moyenne des points, présente un bug dans le cas où on part d'une grille de Gauss.

En ce qui concerne les performances CPU, le Fullpos\_Prep n'arrive pas vraiment à concurrencer avec le PREP qui semble avoir été très bien optimisé et parallélisé au niveau de l'OpenMP pour ce dernier cycle (cy41).

Pour les choix d'interpolation, il serait peut être judicieux de combiner les trois méthodes existantes et en créer une nouvelle et l'implémenter comme choix par défaut.

Cette nouvelle méthode commencera par essayer de faire les interpolations sur quatre points, et si on ne trouve pas, parmi ces quatre points, un point de même nature que le point cible, basculer vers le point, de même nature, le plus proche dans un halo dont la taille est fonction du rapport de résolution entre la grille de départ et la grille d'arrivée. Si toutefois, on n'arrive toujours pas à trouver un point de même nature dans ce halo que le pont cible, on prend alors une valeur par défaut par type de champs (TG1, TG2, WG1, ...) (définie dans un module quelque part).



# ANNEXE

## Fullpos/PREP

By Philippe Marguinaud

### Introduction

This document describes the changes that were made to PREP, Fullpos and MSE in order to be able to make PREP interpolations using Fullpos.

### PREP data flow

Below is the structure of PREP, starting from PREP\_SURF\_ATM, and limited to what is commonly used for ARPEGE/AROME and interesting for our purpose:

PREP\_SURF\_ATM

- PREP\_SEA
  - PREP\_SEAFLUX
    - PREP\_HOR\_SEAFLUX\_FIELD
      - PREP\_SEAFLUX\_EXTERN
      - HOR\_INTERPOL
      - Extra post-processing
    - Extra post-processing
  - PREP\_INLAND\_WATER
    - PREP\_WATFLUX
      - PREP\_HOR\_WATFLUX\_FIELD
        - PREP\_WATFLUX\_EXTERN
        - HOR\_INTERPOL
        - Extra post-processing
      - Extra post-processing
- PREP\_NATURE
  - PREP\_ISBA
    - PREP\_HOR\_ISBA\_FIELD
      - PREP\_HOR\_SNOW\_FIELDS

- PREP\_HOR\_SNOW\_FIELD
  - PREP\_SNOW\_EXTERN
  - HOR\_INTERPOL
  - Extra post-processing
- PREP\_ISBA\_EXTERN
  - HOR\_INTERPOL
  - Extra post-processing
- Extra post-processing
- PREP\_TOWN
  - PREP\_TEB
    - PREP\_HOR\_TEB\_FIELD
      - PREP\_HOR\_SNOW\_FIELDS (described above)
      - PREP\_TEB\_EXTERN
      - HOR\_INTERPOL
      - Extra post-processing

From the structure of PREP, we see that the same kind of sequence appears multiple times; for instance in PREP\_HOR\_SEAFLUX\_FIELD:

- PREP\_SEAFLUX\_EXTERN
- HOR\_INTERPOL
- Extra post-processing

The PREP\*\_EXTERN routines read fields from input files (or in the case of AROME, a buffer which was allocated and filled with values taken from input files). Then HOR\_INTERPOL interpolates these fields, and eventually, some post-processing occurs on the target grid.

## Getting the list of fields to be interpolated from PREP

A new argument has been added to all PREP routines; it is a scalar of the following derived type:

```

TYPE PREP_CTL
  LOGICAL :: LSTEP0 = .FALSE. ! PREP step 0 (count fields to be interpolated)
  LOGICAL :: LSTEP1 = .FALSE. ! PREP step 1 (retrieve fields to be interpolated)
  LOGICAL :: LSTEP2 = .FALSE. ! PREP step 2 (pass interpolated fields to PREP and finish)
  LOGICAL :: LPART1 = .TRUE.  ! Invoke PREP*_EXTERN routines
  LOGICAL :: LPART2 = .FALSE. ! Retrieve or pass back interpolated fields from/to PREP
  LOGICAL :: LPART3 = .TRUE.  ! Do interpolations
  LOGICAL :: LPART4 = .FALSE. ! Unused for now
  LOGICAL :: LPART5 = .TRUE.  ! Post-processing after interpolations
  LOGICAL :: LPART6 = .TRUE.  ! Post-processing after interpolations (higher level)
  TYPE (PREP_CTL_FLD), POINTER :: HEAD => NULL (), TAIL => NULL ()
END TYPE

```

This argument controls what PREP is doing, for instance in PREP\_HOR\_SEAFLUX\_FIELD :

```

IF (YDCTL%LPART1) THEN
  CALL PREP_SEAFLUX_EXTERN(...)
ENDIF
CALL PREP_CTL_INT_PART2 (YDCTL, HSURF, 'SEA', CMASK, ZFIELDIN)
IF (YDCTL%LPART3) THEN
  ALLOCATE (ZFIELDOUT (SIZE (XLAT), SIZE (ZFIELDIN, 2)))
  CALL HOR_INTERPOL (ILUOUT, ZFIELDIN, ZFIELDOUT)
ENDIF

CALL PREP_CTL_INT_PART4 (YDCTL, HSURF, 'SEA', CMASK, ZFIELDIN, ZFIELDOUT)

IF (YDCTL%LPART5) THEN
  ! Post-processing on output grid
ENDIF

```

Note that the last member of YDCTL is a pointer to a doubly linked list of fields. On the first pass, this list is filled with fields to be interpolated, then on the second pass scanned for interpolated fields, as we will see later.

Invoking PREP in the usual manner involves declaring a variable of type PREP\_CTL, with its defaults and passing it to PREP\_SURF\_ATM.

Some routines have not (yet) been modified to record PREP fields; invoking them using partial PREP settings would yield an error:

- PREP\_FLAKE
- PREP\_HOR\_TEB\_GARDEN\_FIELD
- PREP\_HOR\_TEB\_GREENROOF\_FIELD

These routines do not appear to be used (for now) in AROME.

## Interpolating PREP fields with Fullpos

### *Fullpos setup*

The following routines have been extended:

- SUFPC, by SUFPCSF (reckon the number of PREP fields and append them to the namelist)
- SUAFN1, by SUAFN1SFX
- SUAFN2, by SUAFN2SFX
- SUAFN3, by SUAFN3SFX

It is worth noting that no SURFEX or PREP fields need to be specified in the namelist. PREP fields are anonymous (they are named PREP.NNNNNN), and their list is set up by PREP itself, but it possible to see their meaning inside the listing (NODE.001\_01); for instance:

SUAFN1SFX: PREP.000001	IS ZS	OVER SEA	/SEA	(:, 1)
SUAFN1SFX: PREP.000002	IS SST	OVER SEA	/SEA	(:, 1)
SUAFN1SFX: PREP.000003	IS ZS	OVER WATER	/WATER	(:, 1)
SUAFN1SFX: PREP.000004	IS TSWATER	OVER WATER	/WATER	(:, 1)
SUAFN1SFX: PREP.000005	IS ZS	OVER NATURE	/NATURE	(:, 1, 1)

SUAFN1SFX: PREP.000006	IS WG	OVER NATURE	/NATURE	(:, 1, 1)
SUAFN1SFX: PREP.000007	IS WG	OVER NATURE	/NATURE	(:, 2, 1)
SUAFN1SFX: PREP.000008	IS WG	OVER NATURE	/NATURE	(:, 3, 1)
SUAFN1SFX: PREP.000009	IS WG	OVER NATURE	/NATURE	(:, 4, 1)
SUAFN1SFX: PREP.000010	IS WG	OVER NATURE	/NATURE	(:, 5, 1)
SUAFN1SFX: PREP.000011	IS WG	OVER NATURE	/NATURE	(:, 6, 1)
SUAFN1SFX: PREP.000012	IS WG	OVER NATURE	/NATURE	(:, 7, 1)
SUAFN1SFX: PREP.000013	IS WG	OVER NATURE	/NATURE	(:, 8, 1)
SUAFN1SFX: PREP.000014	IS WG	OVER NATURE	/NATURE	(:, 9, 1)
SUAFN1SFX: PREP.000015	IS WG	OVER NATURE	/NATURE	(:, 10, 1)
SUAFN1SFX: PREP.000016	IS WG	OVER NATURE	/NATURE	(:, 11, 1)
SUAFN1SFX: PREP.000017	IS WG	OVER NATURE	/NATURE	(:, 12, 1)
...				

The interesting routine to look at in order to understand how Fullpos gets the list of fields is PREP\_STEP0, where PREP is run using dummy PGDs, that is, PGDs whose meta-data are the same as real PGDs, but with only one point of SEA, one point of WATER, one point of TOWN. These dummy PGDs are created in routines PREP1\_DUMM and PREP2\_DUMM.

### **PREP part 1**

In GRIDFPOSSFX\_INIT, the first part of PREP is run. For this purpose, we use a control structure with the following settings :

YDCTL%LSTEP0=. FALSE .
YDCTL%LSTEP1=. TRUE .
YDCTL%LSTEP2=. FALSE .
YDCTL%LPART1=. TRUE .
YDCTL%LPART2=. TRUE .
YDCTL%LPART3=. FALSE .
YDCTL%LPART4=. FALSE .
YDCTL%LPART5=. FALSE .
YDCTL%LPART6=. FALSE .

We then retrieve the list of fields to be interpolated in YDCTL; these fields are now ready to be interpolated by Fullpos.

Masks of the target PGD are read, so that interpolation calculations occur only when required.

### **Fullpos interpolations**

Fields to be interpolated are retrieved using HPOS, which has been extended by HPOSSFX in order to access fields created by the first part of PREP.

PREP fields may have undefined values. Therefore, it has been necessary to create special interpolation routines :

- FPINT4X : 4-point interpolation, with possibly missing points among the 4 used points
- FPAVG : average over the Fullpos halo
- FPNEAR : nearest value in the Fullpos halo

FPINT4X should be used when going from a loose mesh to a finer one, whereas FPAVG and FPNEAR should be used when doing the opposite. FPAVG is expensive, because for each target point, the whole halo has to be scanned and values added (this routine does not vectorise). FPNEAR is much cheaper. The Fullpos halo width has to be tuned to take into account the mesh ratio between

the origin and target grids.

Eventually, note that the second part of PREP will only succeed if masks from the origin and target PGDs are consistent. In principle, this should be the case if the two following conditions are fulfilled:

- the same ECOCLIMAP geophysical data have been used to create both origin and target PGDs
- no minimum thresholds have been used on TOWN, NATURE, SEA, WATER fractions when PGDs were created

## **PREP part 2**

Once fields have been interpolated by Fullpos, it is possible to pass them back to PREP. This is done in FP2SX2 and PREP\_STEP2. In PREP\_STEP2, we use the following control settings:

```
YDCTL%LSTEP0=. FALSE .  
YDCTL%LSTEP1=. FALSE .  
YDCTL%LSTEP2=. TRUE .  
YDCTL%LPART1=. FALSE .  
YDCTL%LPART2=. TRUE .  
YDCTL%LPART3=. TRUE .  
YDCTL%LPART4=. FALSE .  
YDCTL%LPART5=. TRUE .  
YDCTL%LPART6=. TRUE .
```

LPART2 is true, which means that the fields we pass back to PREP will go through HOR\_INTERPOL. But as a 1D geometry has been specified (through the use of the GRID\_TYPE parameter, see prep1\_dumm.F90 and prep1\_real.F90), PREP interpolations will use the routine HOR\_INTERPOL\_BUFFER, which is merely a repacking of input fields, taking into account the masks of the target grid.

Note that HOR\_INTERPOL\_BUFFER has been modified, so that fields interpolated by Fullpos be checked for UNDEFINED values.

Then fields created by the second part of PREP are written to an output file by WRGP2FAFP.

## **MSE**

The regular PREP takes its data from three files:

- PREP file on origin geometry
- PGD file on origin geometry
- PGD file on target geometry

In ARPEGE/AROME, the first two are read from a pre-allocated, pre-filled buffer. The last one (PGD file on target geometry) was only actually used in the couplingsurf configuration (ISBA -> SURFEX). Note that in couplingsurf, the PREP and PGD file on origin geometry do not exist.

Hence, until now, a single input cache for SURFEX existed. But for running from within Fullpos, it has been necessary to distinguish between origin and target geometry. For this purpose, two structures holding SURFEX fields have been created in MODD\_IO\_SURF\_ARO :

```
TYPE (SURFEX_FIELD_BUF_CACHE) , POINTER :: YSURFEX_CACHE_IN => NULL ()  
TYPE (SURFEX_FIELD_BUF_CACHE) , TARGET :: YSURFEX_CACHE_IN_IO_SURF
```

```
TYPE(SURFEX_FIELD_BUF_CACHE), TARGET :: YSURFEX_CACHE_IN_AUX_IO_SURF
```

YSURFEX\_CACHE\_IN\_IO\_SURF holds fields on the origin grid, while YSURFEX\_CACHE\_IN\_AUX\_IO\_SURF holds PGD fields on the target grid.

YSURFEX\_CACHE\_IN is a pointer read in READ\_SURF\*\_ARO routines, and may point to either YSURFEX\_CACHE\_IN\_IO\_SURF or YSURFEX\_CACHE\_IN\_AUX\_IO\_SURF.

The association is made in AROINIT\_IO\_SURF\_n or AROOPEN\_AUX\_IO\_SURF.

Note that this is guaranteed to work as long as PREP does not open both files at the same time, which is the case for now.

## Namelist parameters

A few namelist parameters have been appended to the already (long) namelist NAMFPC :

NFPSURFEX	<ul style="list-style-type: none"><li>• 1 = couplingsurf (ISBA -&gt; SURFEX)</li><li>• 2 = Fullpos/PREP (SURFEX -&gt; SURFEX)</li></ul>
NFPSLWIDE	Size of the Fullpos halo (default is 3)
NFPSFXINT	<ul style="list-style-type: none"><li>• 0 = 4-point interpolations (default)</li><li>• 1 = average over Fullpos halo</li><li>• 2 = take nearest point</li></ul>
NFPSFXWRT	<ul style="list-style-type: none"><li>• 0 = write SURFEX file PFFPOSAREA+0000.sfx (default)</li><li>• 1 = write PREP fields in PFFPOSAREA+0000 (useful for debugging)</li></ul>

In order to avoid reading the whole list of upper air fields, it is possible to specify:

```
&NAMDIM
  NPROMA=-50,
  NFLEVG=1,
/
&NAMFPG
  FPVALH(0)=0.,
  FPVALH(1)=0.,
  FPVBH(0)=0.,
  FPVBH(1)=1.,
/
```

Eventually, let us recall that PREP or SURFEX fields to be interpolated should not be specified in the namelist: the list of fields to be interpolated is set up by PREP itself.