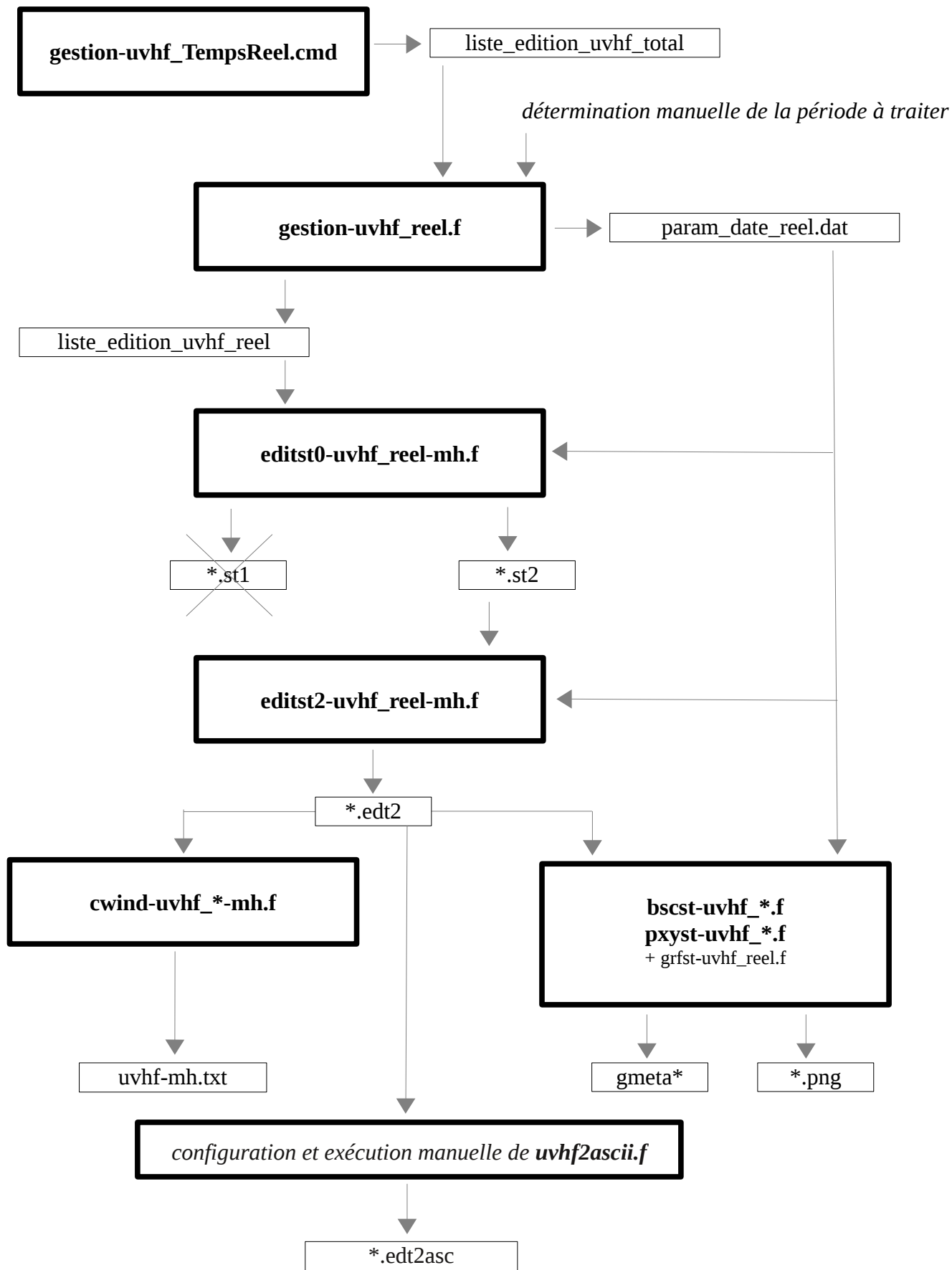


– *Déroulement du traitement* –

Le fichier de commandes gestion-uvhf_TempsReel.cmd lancera de façon autonome le traitement des données après qu'on ait correctement paramétré le fichier gestion-uvhf_reel.f

Il faudra également que les données à traiter aient été placées dans le répertoire DATA.



Description des ENTREES et SORTIES pour chaque programmes

entrées	programme	sorties
		poubelle_reel (fichier de travail ascii) liste_edition_uvvhf_total liste de tout DATA edt2ok (fichier de travail ascii) log_reel consigne les info d'exécution param_date_reel.dat (fichier de travail ascii) pid_radar (fichier de travail ascii) COMPILATION (fichier de travail ascii) imag3d inutilisé gestion-uvvhf_reel.out exécutable grfst-uvvhf_reel.out exécutable editst0-uvvhf_reel-mb.out exécutable editst2-uvvhf_reel-mb.out exécutable editst0-uvvhf_reel-mh.out exécutable editst2-uvvhf_reel-mh.out exécutable cwind-uvvhf_reel-mb.out exécutable cwind-uvvhf_reel-mh.out exécutable bscst-uvvhf_png_reel-mb.out exécutable bscst-uvvhf_png_reel-mh.out exécutable bscst-uvvhf_z_reel-mb.out exécutable bscst-uvvhf_z_reel-mh.out exécutable bscst-uvvhf_long_reel-mb.out exécutable bscst-uvvhf_long_reel-mh.out exécutable pxyst-uvvhf_long_reel-mb.out exécutable pxyst-uvvhf_long_reel-mh.out exécutable pxyst-uvvhf_zi_reel-mb.out exécutable pxyst-uvvhf_zi_reel-mh.out exécutable pxyst-uvvhf_prf-courant_reel-mb.out exécutable pxyst-uvvhf_prf-courant_reel-mh.out exécutable pxyst-uvvhf_tmp-rain_reel-mb.out exécutable pxyst-uvvhf_tmp-rain_reel-mh.out exécutable
*.DAT fichiers bruts à traiter ONESHOOT exécution une seule fois gestion-uvvhf_reel.f/out grfst-uvvhf_reel.f/out editst0-uvvhf_reel-mb.f/out editst2-uvvhf_reel-mb.f/out editst0-uvvhf_reel-mh.f/out editst2-uvvhf_reel-mh.f/out cwind-uvvhf_reel-mb.f/out cwind-uvvhf_reel-mh.f/out bscst-uvvhf_png_reel-mb.f/out bscst-uvvhf_png_reel-mh.f/out bscst-uvvhf_z_reel-mb.f/out bscst-uvvhf_z_reel-mh.f/out bscst-uvvhf_long_reel-mb.f/out bscst-uvvhf_long_reel-mh.f/out pxyst-uvvhf_long_reel-mb.f/out pxyst-uvvhf_long_reel-mh.f/out pxyst-uvvhf_zi_reel-mb.f/out pxyst-uvvhf_zi_reel-mh.f/out pxyst-uvvhf_prf-courant_reel-mb.f/out pxyst-uvvhf_prf-courant_reel-mh.f/out pxyst-uvvhf_tmp-rain_reel-mb.f/out pxyst-uvvhf_tmp-rain_reel-mh.f/out ftp_cmd envoi à E-Profile gmeta2png_reel.cmd gmeta-split.cmd gwenview ou gqview	gestion-uvvhf_TempsReel.cmd	uvvhf_temps_reel-mb.st2 1ère étape de traitement uvvhf_temps_reel-mb.edt2 fichier édité uvvhf_temps_reel-mh.st2 1ère étape de traitement uvvhf_temps_reel-mh.edt2 fichier édité uvvhf-mb.txt fichier E-Profile temps réel uvvhf-mh.txt fichier E-Profile temps réel gmeta gmeta-total-png gmetax gmeta.ps pour_grfst_reel_asc bscst-z_asc pxy-zi_asc param_grfst_reel.dat param_grfst_reel-zzi.dat gmeta-total-prf gmeta-rain-tmp uvvhf_z-bsc-mb.png uvvhf_z-bsc-mh.png uvvhf_a-bsc-mb.png uvvhf_a-bsc-mh.png uvvhf_v-bsc-mb.png uvvhf_v-bsc-mh.png uvvhf_w-bsc-mb.png uvvhf_w-bsc-mh.png uvvhf_s-bsc-mb.png uvvhf_s-bsc-mh.png uvvhf_b-bsc-mb.png uvvhf_b-bsc-mh.png uvvhf_k-bsc-mb.png uvvhf_k-bsc-mh.png uvvhf_d-bsc-mh.png uvvhf_w-prf-mb.png uvvhf_w-prf-mh.png uvvhf_z-prf-mb.png uvvhf_z-prf-mh.png uvvhf_v-prf-mb.png uvvhf_v-prf-mh.png uvvhf_d-prf-mb.png uvvhf_d-prf-mh.png uvvhf_e-prf-mb.png uvvhf_s-prf-mh.png uvvhf_b-prf-mb.png uvvhf_b-prf-mh.png uvvhf_k-prf-mb.png uvvhf_k-prf-mh.png uvvhf_r-tmp-mb.png uvvhf_r-tmp-mh.png gmeta_bsc_long_reel-mb gmeta_bsc_long_reel-mh gmeta_pxy_long_reel-mb gmeta_pxy_long_reel-mh

gestion-uvhf_TempsReel.cmd

pour le VHF de Lannemezan

– **initialisation générale (les principales sont présentées ci-après)** –

ligne 74: création d'un fichier nommé **COMPILATION**

lignes 87 à 90: initialisation des minutes de déclenchement de la filière (utile pour le temps réel, inutile pour rosea)

start_minute1 = 00 start_minute2 = 30 start_minute#

ligne 101: détermination du mode à traiter fmode

mode bas: fmode = 1 mode haut: fmode = 2 mb + mh: fmode = 3 (pour VHF: fmode = 2)

ligne 110: détermination de la nature du traitement imag

imag = 0 : édition seule (pas d'images) imag = 1 : édition + création d'images (sans affichage)

imag = 2 : édition + création et affichage d'images imag = 3 : réseau 3D de profileurs (pour rosea: fmode = 0 ou 1, à définir)

ligne 124: vérifie si compilation nécessaire ou non fichier **COMPILATION**

il faut compiler les sources, seulement si le fichier COMPILATION est présent (pour ne pas compiler à chaque traitement)

– **compilation des programmes fortran** –

si fichier **COMPILATION** présent (ligne 124):

ligne 130 à 131: affichage écran date-heure actuelle (début de compilation)

– **compilation des programmes communs aux modes bas et haut** –

ligne 135: compilation avec ifort de Sources/gestion-uvhf_reel.f

ligne 139: compilation avec ncargf77 de Sources/grfst-uvhf_reel.f

ligne 142: test mode à traiter fmode

si mb seul ou si mb+mh (VHF mh seul, donc inutilisé par rosea pour le moment)

– **compilation des programmes mode bas** –

lignes 144 à 146: compilation avec ifort de Sources/editst0-uvhf_reel-mb.f

 Sources/editst2-uvhf_reel-mb.f

 Sources/cwind-uvhf_reel-mb.f

ligne 147: test nature du traitement image

si le traitement choisi réclame les images imag <> 0

– **compilation des programmes pour les images mode bas** –

ligne 148 à 154: compilation avec ifort de Sources/bscst-uvhf_png_reel-mb.f

 Sources/bscst-uvhf_z_reel-mb.f

 Sources/bscst-uvhf_long_reel-mb.f

 Sources/pxyst-uvhf_long_reel-mb.f

 Sources/pxyst-uvhf_zi_reel-mb.f

 Sources/pxyst-uvhf_prf-courant_reel-mb.f

 Sources/pxyst-uvhf_tmp-rain_reel-mb.f

fin de si (images)

fin de si (mb ou mb+mh)

ligne 158: test mode à traiter fmode

si mh seul ou si mb+mh (VHF mh seul, donc rosea ok actuellement ici)

– **compilation des programmes mode haut** –

lignes 160 à 162: compilation avec ifort de Sources/editst0-uvhf_reel-mh.f editst0 et editst2 sont

 Sources/editst2-uvhf_reel-mh.f nécessaires à rosea)

 Sources/cwind-uvhf_reel-mh.f (réseau temps réel pas rosea)

ligne 163: test nature du traitement image

si le traitement choisi réclame les images (dépendra de ce qu'on décide pour les images rosea) imag <> 0

– **compilation des programmes pour les images mode haut** –

lignes 164 à 170: compilation avec ifort de Sources/bscst-uvhf_png_reel-mh.f

 Sources/bscst-uvhf_z_reel-mh.f

 Sources/bscst-uvhf_long_reel-mh.f

 Sources/pxyst-uvhf_long_reel-mh.f

 Sources/pxyst-uvhf_zi_reel-mh.f

 Sources/pxyst-uvhf_prf-courant_reel-mh.f

 Sources/pxyst-uvhf_tmp-rain_reel-mh.f

fin de si (images)

fin de si (mh ou mb+mh)

ligne 174: suppression du fichier **COMPILATION**

fin de si (compilation)

pas de compilation

lignes 177 à 179: affichage écran date-heure actuelle (fin de compilation)

lignes 183 à 191: ménage dans les 3 répertoires Uvhf_TempsReel, Images et Partage-uvhf

lignes 196 et 197: **DEBUT DE LA BOUCLE INFINIE**

lignes 199 à 201: test si déclenchement de la filière (comparaison entre heure actuelle et `start_minute`) si correspondance entre heure actuelle et `start_minute#` (le traitement se fait dès le 1er passage de la boucle – grâce à l'initialisation de minute par un `start_minute#` – puis attendra les déclenchements définis par `start_minute#`)

– création des fichiers édités `st2` et `edt2` –

lignes 214 à 226: ménage du répertoire de travail Uvhf_TempsReel des fichiers suivants

<code>log_reel</code>	<code>liste_edition_uvhf_reel</code>	<code>uvhf_temps_reel-mb.edt2</code>
<code>poubelle_reel</code>	<code>liste_edition_uvhf_total</code>	<code>uvhf_temps_reel-mh.edt2</code>
<code>pour_grfst_reel_asc</code>	<code>param_grfst_reel.dat</code>	<code>uvhf_temps_reel-mb.st2</code>
<code>uvhf-mh.txt</code>	<code>param_date_reel.dat</code>	<code>uvhf_temps_reel-mh.st2</code>
<code>uvhf-mb.txt</code>		

ligne 229: mise en fichier de la date actuelle dans `poubelle_reel` (pour temps réel, pas pour rosea)

lignes 231 à 240: création de la liste des fichiers bruts présents: `liste_edition_uvhf_total`

lignes 242 à 246: exécution de `gestion-uvhf_reel.out` → détermination des fichiers bruts à traiter (`liste_edition_uvhf_reel`) et des bornes temporelles des fichiers édités et des images à générés (`param_date_reel.dat`)

lignes 249 et 250: test mode à traiter

fmode

si mb seul ou si mb+mh (VHF mh seul, donc inutilisé par rosea pour le moment)

ligne 252: affichage écran heure actuelle (début d'édition mode bas)

ligne 253: exécution de `editst0-uvhf_reel-mh.out`

ligne 254: exécution de `editst2-uvhf_reel-mh.out`

ligne 255: copie de `uvhf_temps_reel-mb.edt2` dans le répertoire Partage-uvhf

fin de si (mb ou mb+mh)

lignes 257 et 258: test mode à traiter

fmode

si mh seul ou si mb+mh (VHF mh seul, donc rosea ok actuellement ici)

ligne 260: affichage écran heure actuelle (début d'édition mode haut)

ligne 261: exécution de `editst0-uvhf_reel-mh.out`

ligne 262: exécution de `editst2-uvhf_reel-mh.out`

ligne 263: copie de `uvhf_temps_reel-mh.edt2` dans le répertoire Partage-uvhf

fin de si (mh ou mb+mh)

– création des txt pour E-Profile – (inutile pour rosea)

lignes 268 à 283: exécution de `cwind-uvhf_reel-mb.out` et/ou `cwind-uvhf_reel-mh.out`

création des fichiers pour E-Profile mb et/ou mh (`uvhf-mb.txt` et `uvhf-mh.txt`)

copie avec renommage pour horodatage dans Partage-uvhf et copie simple dans Images

lignes 287 et 288: envoi des fichiers E-Profile par ftp

– création des images encapsulées `gmeta` et des images `png` –

ligne 292: test nature du traitement

image

si le traitement choisi réclame les images (dépendra de ce qu'on décide pour les images rosea) `imag <> 0`

– création des `png` –

lignes 295 et 296: test mode à traiter

fmode

si mb seul ou si mb+mh (VHF mh seul, donc inutilisé par rosea pour le moment)

lignes 300 à 315: exécution de `pxyst-uvhf_prf-courant_reel-mb.out`

`pxyst-uvhf_tmp-rain_reel-mb.out`, `pxyst-uvhf_zi_reel-mb.out`

`bscst-uvhf_png_reel-mb.out`, `bscst-uvhf_z_reel-mb.out`, `grfst-uvhf_reel.out`

pour créer les images encapsulées `gmeta-total-png`, `gmeta-total-prf` et `gmeta-rain-tmp`

ligne 318: affichage écran heure actuelle (début création `png` mode bas)

lignes 320 à 410: exécution de `grfst-uvhf_reel.out`, `gmeta2png_reel.cmd`, `gmeta-split.cmd`

pour créer les images `png` `uvhf_w-prf-mb.png`

`uvhf_z-bsc-mb.png` `uvhf_z-prf-mb.png`

`uvhf_a-bsc-mb.png` `uvhf_v-prf-mb.png`

`uvhf_v-bsc-mb.png` `uvhf_d-prf-mb.png`

`uvhf_w-bsc-mb.png` `uvhf_e-prf-mb.png`

`uvhf_s-bsc-mb.png` `uvhf_b-prf-mb.png`

`uvhf_b-bsc-mb.png` `uvhf_k-prf-mb.png`

`uvhf_k-bsc-mb.png` `uvhf_r-tmp-mb.png`

Boucle infinie

pas de traitement (star_minute)

Boucle infinie

pas de traitement (star_minute)

édition seule / pas d'images

ligne 411: affichage écran heure actuelle (fin création png mode bas)

lignes 416 à 431: copie des png mode bas dans Partage-uvhf

lignes 435 à 452: copie des png mode bas dans Images avec ajout d'un préfixe pour tri (permet une hiérarchisation des images dans gwenview, sans intérêt pour rosea)

lignes 454 et 455: affichage écran date-heure actuelle (fin copie png mode bas)

fin de si (mb ou mb+mh)

lignes 462 et 463: test mode à traiter

fmode

si mh seul ou si mb+mh (VHF mh seul, donc rosea ok actuellement ici)

lignes 467 à 483: exécution de pxyst-uvhf_prf-courant_reel-mh.out

pxyst-uvhf_tmp-rain_reel-mh.out, pxyst-uvhf_zi_reel-mh.out

bscst-uvhf_png_reel-mh.out, bscst-uvhf_z_reel-mh.out, grfst-uvhf_reel.out

pour créer les images encapsulées gmeta-total-png, gmeta-total-prf et gmeta-rain-tmp

ligne 486: affichage écran heure actuelle (début création png mode haut)

lignes 488 à 583: exécution de grfst-uvhf_reel.out, gmeta2png_reel.cmd, gmeta-split.cmd

pour créer les images png uvhf_w-prf-mh.png

uvhf_z-bsc-mh.png uvhf_z-prf-mh.png

uvhf_a-bsc-mh.png uvhf_v-prf-mh.png

uvhf_v-bsc-mh.png uvhf_d-prf-mh.png

uvhf_w-bsc-mh.png uvhf_s-prf-mh.png

uvhf_s-bsc-mh.png uvhf_b-prf-mh.png

uvhf_b-bsc-mh.png uvhf_k-prf-mh.png

uvhf_k-bsc-mh.png uvhf_r-tmp-mh.png

uvhf_d-bsc-mh.png

ligne 585: affichage écran heure actuelle (fin création png mode haut)

lignes 587 à 605: copie des png mode haut dans Partage-uvhf

lignes 607 à 627: copie des png mode bas dans Images avec ajout d'un préfixe pour tri (permet une hiérarchisation des images dans gwenview, sans intérêt pour rosea)

lignes 629 et 630: affichage écran date-heure actuelle (fin copie png mode haut)

fin de si (mh ou mb+mh)

– **affichage écran des png par gwenview** –

ligne 644: test affichage écran des images png

imag >= 2

si affichage écran des images png a été choisi

lignes 645 à 653: affichage à l'écran avec gwenview du contenu de Images

gestion des anciens affichages éventuels (kill)

fin de si (affichage écran des png)

– **création des gmeta_long** –

lignes 656 et 657: test mode à traiter

fmode

si mb seul ou si mb+mh (VHF mh seul, donc inutilisé par rosea pour le moment)

ligne 659: affichage écran heure actuelle (début création gmeta_long mode bas)

lignes 660 à 667: exécution de bscst-uvhf_long_reel-mb.out, grfst-uvhf_reel.out,

pxyst-uvhf_long_reel-mb.out

pour créer les images encapsulées gmeta_bsc_long_reel-mb, gmeta_pxy_long_reel-mb

lignes 668 à 671: copie des gmeta_long mode bas dans Partage-uvhf

fin de si (mb ou mb+mh)

lignes 673 et 674: test mode à traiter

fmode

si mh seul ou si mb+mh (VHF mh seul, donc rosea ok actuellement ici)

ligne 676: affichage écran heure actuelle (début création gmeta_long mode haut)

lignes 677 à 684: exécution de bscst-uvhf_long_reel-mh.out, grfst-uvhf_reel.out,

pxyst-uvhf_long_reel-mh.out

pour créer les images encapsulées gmeta_bsc_long_reel-mh, gmeta_pxy_long_reel-mh

lignes 685 à 688: copie des gmeta_long mode haut dans Partage-uvhf

fin de si (mh ou mb+mh)

lignes 691 à 693: ménage dans le répertoire de travail Uvhf_TempsReel

→ suppression des images *.png, des fichiers temporaires gmeta et gmeta.ps

fin de si (traitement choisi réclame les images)

ligne 697: affichage écran heure actuelle (fin des images éventuelles)

ligne 699: affichage écran heure actuelle (fin du cycle de traitement)

↓
Boucle infinie

– *attente du prochain cycle* –
↓
ligne 700: affichage écran "prochain traitement dans 30 minutes"
fin de si (correspondance minute actuelle et `start_minute#`)
récupération des minutes de l'heure actuelle
si fichier `ONESHOOT` présent (ligne 715):
↓
sortie de la boucle infinie
fin de si (`ONESHOOT`)
Reboucler en début de boucle infinie
affichage écran "Fin des Traitements et des Affichages"

gestion-uvhf_reel.f

pour le VHF de Lannemezan

Ce programme détermine la période à traiter

- à partir de la date d'exécution en fonctionnement temps réel
- à partir des informations manuellement configurées dans le menu, en fonctionnement temps différé

– **initialisation générale (lignes 70 à 93 + ligne 110)** –

Menu de configuration pour l'utilisation en temps réel et en temps différé.

En temps réel, seul le menu doit être configuré (lignes 70 à 93).

En temps différé, un paramètre supplémentaire doit être configuré (ligne 110).

ligne 76: Nombre de jours à traiter.

njouraffichage = 3 en temps réel

njouraffichage = 31 (pour un mois de 31 jours) → pour rosea

njouraffichage

ligne 79: Heure de départ du traitement. Le début du traitement est configuré pour débiter à 12h afin d'assurer la continuité temporelle en début de la période demandée. La filière ira chercher d'elle-même les données précédant immédiatement la période demandée. Il est donc primordial que ces données soient présentes aux côtés de celles de la période à traiter. Dans l'exemple choisi de traiter tout le mois d'avril 2013, les données du 31 mars au 30 avril ont été placées dans le répertoire DATA

iheuredebtrt = 120000 → pour rosea et le temps réel

iheuredebtrt

ligne 83: chemin et nom de la liste (en lecture) contenant l'ensemble des fichiers du répertoire DATA

listebrute='liste_edition_uvhf_total' → pour rosea et le temps réel

listebrute

ligne 87: chemin et nom de la liste de sortie contenant l'ensemble des fichiers à traiter

listereelle='liste_edition_uvhf_reel' → pour rosea et le temps réel

listereelle

ligne 91: chemin et nom du fichier de sortie contenant les dates des périodes à traiter et à afficher

fichdate='param_date_reel.dat' → pour rosea et le temps réel

nfichdate

ligne 110: En temps réel, cette ligne doit être mise en commentaire. Le programme utilisera alors la date actuelle. En temps différé, cette ligne doit être décommenté et il faut y indiquer la date de fin de la période à traiter (le 30 avril, pour notre exemple), au format aammjj.

idatecourante=130430 → pour rosea et le temps réel

datecourante

editst0-uvhf_reel-mh.f
pour le VHF de Lannemezan

Ce programme est une première étape dans l'édition des données brutes issues de radar profileurs de vent. Il effectue une moyenne et déduit les pics potentiellement météorologiques qui seront exploités pour les tests suivants. Il calcule les moments d'ordre 0 à 3 pour chaque pic, à savoir la réflectivité, la vitesse doppler, la largeur du pic et son asymétrie.

Remarque:

Il faut utiliser des options particulières (assume byterecl) pour la compilation de ce programme (cf entête du programme, lignes 13 à 16). Ceci est inclus dans gestion-uvhf_TempsReel.cmd

editst2-uvhf_reel-mh.f
pour le VHF de Lannemezan

Ce programme

bscst-uvhf_long_reel-mh.f
pour le VHF de Lannemezan

Ce programme

bscst-uvhf_png_reel-mh.f
pour le VHF de Lannemezan

Ce programme

bscst-uvhf_z_reel-mh.f
pour le VHF de Lannemezan

Ce programme

pxyst-uvhf_zi_reel-mh.f
pour le VHF de Lannemezan

Ce programme

grfst-uvhf_reel.f
pour le VHF de Lannemezan

Ce programme

uvhf2ascii.f

pour le VHF de Lannemezan

Ce programme (2724 lignes) lit le fichier édité *.edt2 et écrit un fichier ascii de la période désirée, sur la plage d'altitude demandée, contenant plus ou moins de paramètres. Il ne se contente pas de changer de format (edt2 binaire vers edt2asc ascii) mais il fait également des calculs (par exemple les paramètres turbulents, la hauteur de la tropopause...) et applique ou non certaines corrections (par exemple en altitude, ou de sensibilité d'aspect du signal...). Un menu permet la paramétrisation des calculs, de corrections et de format du fichier de sortie.

ligne 75: commentaire icom = 'radar VHF Lannemezan, 01/04/2013 -> 30/04/2013' icom

lignes 80 et 81: chemin et nom du fichier edt2 à lire
filu = '/Vhf-lzan_TempsReel/Uvhf_TempsReel/uvhf_temps_reel-mh.edt2' filu

lignes 87 et 88: chemin et nom du fichier edt2asc à écrire
fiec = '/Vhf-lzan_TempsReel/Uvhf_TempsReel/uvhf_temps_reel-mh.edt2asc' fiec

lignes 91/92 et 96/97: date et heure de début et de fin de la période à traiter
idatedebut = 130401 iheuredebut = 000000 idatedebut / iheuredebut
idatefin = 130430 iheurefin = 235959 idatefin / iheurefin

ligne 103: type de radar profileur de vent → UHF (=0) ou VHF (=1)
VHF = 1 / UHF = 2 itypeuvhf = 1 pour le VHF de Lannemezan itypeuvhf

ligne 108: correction éventuelle de l'altitude des portes (si mal définies à l'acquisition)
0 = aucune correction 1 = correction coralt = 0 pour le VHF de Lannemezan coralt

ligne 115: correction éventuelle de la sensibilité d'aspect du signal en VHF
0 = aucune correction 1 = correction icorasvvhf = 1 pour le VHF de Lannemezan icorasvvhf

ligne 127: correction éventuelle des portes tronquées (ssi impulsion codée)
0 = aucune correction 1 = correction icorcodimp = 1 pour le VHF de Lannemezan icorcodimp

ligne 141: format d'écriture du fichier ascii de sortie formatecrt = 5 pour rosea formatecrt

lignes 147 et 148: restriction éventuelle de la plage d'altitude
iprtdeb=9999 iprtfin=9999 → aucune restriction pour rosea iprtdeb / iprtfin

ligne 154: paramètres turbulents inclus ou non dans le fichier ascii de sortie
0 = paramètres turbulents limités 1 = tous les paramètres turbulents iecrturb = 0 → pour rosea iecrturb

ligne 159: fichier de sortie en ascii ou en binaire iascbin = 1 → ascii pour rosea iascbin

ligne 164: valeur de la constante radar (différente pour chaque radar)
ctrad = 94 pour le VHF de Lannemezan ctrad

ligne 168: mode de calcul de la vitesse verticale
5 faisceaux = 1 / faisceau vertical = 2 ichoixw = 1 pour le VHF de Lannemezan ichoixw

ligne 175: mode de calcul de la réflectivité ichoixz = 3 pour rosea ichoixz

ligne 179: mode de calcul de epsilon ichoixe = 1 pour rosea ichoixe

lignes 187 à 190: altitudes limites pour la recherche de la hauteur de couche limite (désactivée pour le VHF de Lannemezan dans rosea) et de la hauteur de la tropopause.
zimin=10000 zimax=10000 ztmin=5000 ztmax=16000 → pour rosea zimin / zimax / ztmin / ztmax

lignes 198 et 199: altitudes limites pour le calcul du bruit moyen (numéro des portes)
iprtbrub = 15 iprtbruh = 39 pour le VHF de Lannemezan iprtbrub / iprtbruh