
Les procédures Roproc
pour le traitement des données
de CLUSTER / STAFF-SC, FGM et ORBITE

Patrick Robert, CETP

V 2.0, Septembre 2003

Ce document remplace la version initiale V1.0 de mars 2002

TABLE DES MATIERES

AVANT PROPOS	7
I - PHILOSOPHIE DES MODULES ROPROC	8
1) PRINCIPES	8
2) ENCHAINEMENT DES PROCEDURES	8
3) LES DIFFERENTES FAMILLES ROPROC.....	10
a) différence entre les 3 familles	10
b) terminologie des commandes	11
c) liste des commandes disponibles	11
d) préfixes associés aux expériences	11
4) REALISATION DE SCRIPTS POUR LE TRAITEMENT EN SERIE	12
II - LES PROCEDURES WAVE_ROPROC:.....	13
1) GENERALITES.....	13
2) LE PROGRAMME DE CALIBRATION COWAVE (N1 VERS N2)	13
a) but	13
b) fichier de paramètre cowave.in	13
c) lancement du programme.....	16
d) compte rendu d'exécution cowave.out	16
e) description du fichier résultat cowave.resu.....	16
f) vérification de la cohérence du fichier cowave.resu	19
g) visualisation des formes d'ondes du fichier cowave.resu	19
3) LES FILTRES POUR LES CHANGEMENTS DE REPERES.....	19
a) introduction.....	19
b) wave_to_GSE.....	20
c) wave_to_GSM.....	20
d) wave_to_MVA.....	20
e) wave_to_MFA	20
f) wave_filter	20
4) LES TRAITEMENTS DE NIVEAU N3.....	21
a) préambule	21
b) spectrogrammes	21
c) spectres moyens intégrés sur un intervalle de temps.....	22
d) puissance moyenne dans une bande de fréquence au cours du temps	22
e) polarisation des ondes.....	23
5) LES TRAITEMENTS A 4 SATELLITES.....	24
a) production d'un jeu de 4 fichiers cowave.resu.....	24
b) production de 4 spectrogrammes	24
c) visualisation d'un spectrogrammes avec les 4 composantes Bz de chaque satellite	25
d) production de 4 spectres moyens Bx, By, Bz.....	25
e) production de 4 Puissances intégrées Bx, By, Bz.....	26
6) UTILITAIRES DIVERS:	26
a) copie dans le répertoire courant d'un jeu d'exemple de fichier *.in.....	26
b) fichier readme.txt donnant la procédure d'installation de la chaîne Wave_Roproc.....	26
c) liste des fichiers N1p disponibles dans la base	27
d) lecture et décommutation des entête de bloc d'un fichier N1p.....	27
e) liste des entête de fichier d'une série de fichier N1p.....	27
f) test du temps donné dans les entête de bloc des fichiers N1p	28
g) test de la lecture des fichiers des gains d'antenne	30
h) calcul des composantes GSE d'un vecteur donné dans le GEI.....	31

III - LES PROCEDURES VECTOR_ROPROC:	32
1) GENERALITES.....	32
2) LE PROGRAMME DE FORMATAGE COVECTOR.....	33
a) but	33
b) fichier de paramètre covector.in	33
c) lancement du programme.....	33
d) compte rendu d'exécution covector.out	33
e) description du fichier covector.resu	34
f) visualisation du contenu du fichier covector.resu.....	35
3) LES FILTRES POUR LES CHANGEMENTS DE REPERE	35
a) introduction.....	35
b) vector_to_GSM	35
c) vector_to_SR2	35
d) vector_to_MVA	36
4) LES TRAITEMENTS A 4 SATELLITES.....	36
a) production d'un jeu de 4 fichiers covector.resu.....	36
b) visualisation séparée des 4 fichiers.....	37
c) visualisation commune des 4 fichiers.....	37
5) LES PROCEDURES PARTICULIERES A CLUSTER/FGM.....	37
a) introduction.....	37
b) alignement en temps de 4 fichiers covector.resu.....	38
c) ajout des positions des satellites.....	38
d) calcul de la densité de courant par la méthode du rotationnel.....	38
e) visualisation des résultats	39
f) résultats en repère MVA	39
g) macro_commande cocurldiv_total.....	40
h) macro_commande cocurldiv_lprtot	41
6) CONVERSION DE FORMATS	42
a) Vector_to_Wave.....	42
b) Merge_Alivevector.....	43
7) UTILITAIRES DIVERS	43
a) copie dans le répertoire courant d'un jeu d'exemple de fichier *.in.....	43
b) fichier readme.txt donnant la procédure d'installation de la chaîne Vector_Roproc.....	43
IV - LES PROCEDURES ORBIT_ROPROC:	44
1) GENERALITES.....	44
2) CREATION DE FICHIERS ASCII DES POSITIONS	44
a) liste des positions et de la géométrie du tétraèdre pour une journée: listorbit.....	44
b) visualisation 2D des positions et de la géométrie du tétraèdre.....	46
c) positions seules pour une durée quelconque: cresatpos.....	47
d) visualisation 3D des trajectoires.....	48
3) UTILITAIRES	48
a) copie dans le répertoire courant d'un jeu d'exemple de fichier *.in.....	48
b) fichier readme.txt donnant la procédure d'installation de la chaîne Orbit_Roproc.....	48
V - EXEMPLES DE VISUALISATION POUR CLUSTER	49
1) STAFF	49
a) visuwave.ps	49
b) visuspectro.ps.....	50
c) visuspectro.ps ; exemple avec fenêtre plus large	51
d) visuspectro.ps ; exemple avec apodisation.....	52
e) visuspectro_4Bz.ps	53
f) visuspectra.ps.....	54
g) visupower.ps.....	55
h) visupolar.ps.....	56
i) visupolar.ps en repère MFA.....	57
2) EFW	58
a) visuwave.ps	58
b) visuspectro.ps.....	59

3) FGM	60
a) visuvector.ps.....	60
b) visuvector_4sat.ps	61
c) visucurl.ps	62
d) visunorma.ps	63
e) visuspectro.ps	64
f) visuspectra.ps.....	65
g) visupower.ps.....	66
4) ORBIT	67
a) visuorbit_posit.ps	67
b) visuorbit_tetra.ps	68
c) visusatpos.ps sur 2.5 jours	69
d) visusatpos.ps sur 6 minutes	70
e) autre exemple de visusatpos.ps pour un tétraèdre régulier.....	71
ANNEXE 1 : FONCTION DE TRANSFERT DES EXPERIENCES « ONDES »	72
1) CLUSTER/STAFF-SC	73
a) Partie NBR 0-12.5 Hz	73
b) Partie HBR 0-12.5 Hz	75
2) GEOS / UBF-S300	77
3) STATION MOBILE / SEARCH-COILS	78
4) INTERBALL.....	79
5) CUSP / B	80
6) CUSP / J.....	82
ANNEXE 2 : DEFINITION DES DIFFERENTS REPERES UTILISES	84
1) REPERE SR2.....	84
2) REPÈRE GSE	84
3) REPERE GEI.....	84
4) REPERE GSM.....	85
5) REPERE MVA	85
6) REPERE MFA	85
ANNEXE 3 : UTILISATION SUR LA MACHINE SAWARA DU CETP	86
ANNEXE 4 : INSTALLATION SUR UNE MACHINE SUN / SOLARIS.....	87
1) INSTALLATION	87
2) UTILISATION	88
ANNEXE 5 : RESUME DES PROCEDURES DISPONIBLES POUR CLUSTER.....	89
1) WAVE_ROPROC_STAFF	89
2) WAVE_ROPROC_EFW	91
3) VECTOR_ROPROC_FGM.....	92
4) ORBIT_ROPROC_CLUSTER	94
BIBLIOGRAPHIE	95

Les procédures Roproc

pour le traitement des données

de CLUSTER / STAFF, FGM et ORBITE

AVANT PROPOS

Les procédures Roproc (Les «Robert's processing») ont été développées au CETP pour l'exploitation scientifique des données CLUSTER, et sont en constante évolution durant ces premières années d'exploitation. Elles s'appuient pour STAFF sur la base en ascii des données N1 décommutées, pour FGM sur la base des PPDB (fichiers ascii au format standard FGM), et sont aussi valables pour la bases des fichiers à haute résolution en cours de formation. Pour les fichiers d'orbite, les procédures s'appuient sur la base des fichiers d'orbite produits lors du traitement de routine des données STAFF.

Ces procédures produisent des fichiers de résultats en ascii, auto-documentés, et de format aisément compréhensible même en l'absence de programme de lecture ; ils sont repris par des procédures de visualisation créant des fichiers PostScript.

Les procédures STAFF sont largement dérivées des anciens procédures de traitement des données GEOS, récemment remises à niveau pour l'archivage de ces données au CDPP, et étendues à 4 satellites. Les procédures pour FGM et pour les orbites ont été développées dans le même esprit, en utilisant au maximum les bibliothèques existantes, éventuellement étendues. Elles utilisent notamment la bibliothèque de changement de coordonnées Rocotlib.

Tous les codes de traitement sont en FORTRAN 77, toutes les visualisations sont en PV-WAVES ou en IDL (au choix). Les visualisations s'appuient sur un standard développé pour CLUSTER, et sont regroupées en bibliothèques, avec des modules spécialisés, ce qui facilite grandement la création d'éventuelles nouvelles visualisations.

Les procédures Roproc sont livrés sous la forme d'une arborescence de fichiers exécutables, et peuvent être installées sur une machine SUN/Solaris par simple recopie de l'arborescence du logiciel ; PV-WAVES (V7.0) ou IDL (V 5.3) sont nécessaires pour faire les visualisations.

I - PHILOSOPHIE DES MODULES ROPROC

1) Principes

Les modules Roproc sont des procédures écrites en shell UNIX, et fonctionnant actuellement en libre service sur la machine SUN **sawara** du CETP, et on déjà été portées facilement sur d'autres machines SUN/Solaris. Elles lancent des fichiers exécutables de deux type : des programmes de calcul, écrits en FORTRAN 77, et des programmes de visualisations, écrits en PV-WAVES compatible IDL. Les premiers génèrent des fichiers résultats, en ascii, et peuvent servir d'entrée aux second, qui génèrent des fichiers PostScript visualisables par Ghostview ou imprimables directement.

- Une procédure Roproc de calcul, comme par exemple le programme de calibration **cowave.exe**, a généralement besoin d'un fichier de paramètres (le **cowave.in**) contenant la date et l'heure de début, la durée, etc. Elle génère lors du run un compte rendu d'exécution (le **cowave.out**), et génère un fichier résultat en ascii (le **cowave.resu**).

Ce programme exécutable **cowave.exe** est lui-même lancé à partir d'un shell UNIX, le **cowave.bat**, qui effectue au préalable certaines opérations systèmes (nettoyage du directory courant, test sur l'existence de fichiers indispensables, etc.) avant le lancer une instruction du type :

```
cowave.exe <cowave.in >cowave.out
```

- Une procédure Roproc de visualisation, comme par exemple le **visuwave.dat**, lit un fichier résultant d'un calcul (le **cowave.resu** précédent), et crée un fichier PostScript (le **visuwave.ps**).

Comme précédemment, le fichier exécutable **visuwave.dat** est lui-même lancé à partir d'un shell UNIX, le **visuwave.bat**, qui effectue également certaines opérations systèmes avant le lancer la procédure **visuwave.dat** avec les paramètres convenables.

Les procédures *.bat ne requièrent en général aucun argument et n'attendent aucune réponse. Elles peuvent être exécutées en mode « batch », en différé ou en tâche de fond.

Toutes ces procédures sont généralement utilisées à partir d'une commande du genre **rsc_cowave**, dont le préfixe **rsc** signifie en l'occurrence **Roproc_Cluster_Staff** (voir § 3).

2) Enchaînement des procédures

Certaines procédures s'enchaînent les unes les autres : par exemple, pour calculer et visualiser un spectrogramme, l'ensemble des opérations est résumée sur la figure 1 :

- 1) On commence par lancer le programme de calibration **cowave.bat**, qui lit les paramètres d'entrée dans le fichier **cowave.in** (date, heure, durée, etc.), lit les données de télémessure dans la base des fichiers N1, génère le compte rendu d'exécution dans le fichier **cowave.out**, et enfin génère le fichier **cowave.resu** qui contient les formes d'onde calibrées.
- 2) on lance ensuite le programme de calcul du spectrogramme, **cospectro.bat**, qui lit le fichier **cowave.resu**, génère un compte rendu d'exécution **cospectro.out** et génère un fichier ascii **cospectro.resu** contenant les valeurs des spectres consécutifs.

- 3) Enfin, on lance la procédure de visualisation **visuspectro.bat**, qui lit le fichier **cospectro.resu** et génère à son tour le fichier **visuspectro.ps**.

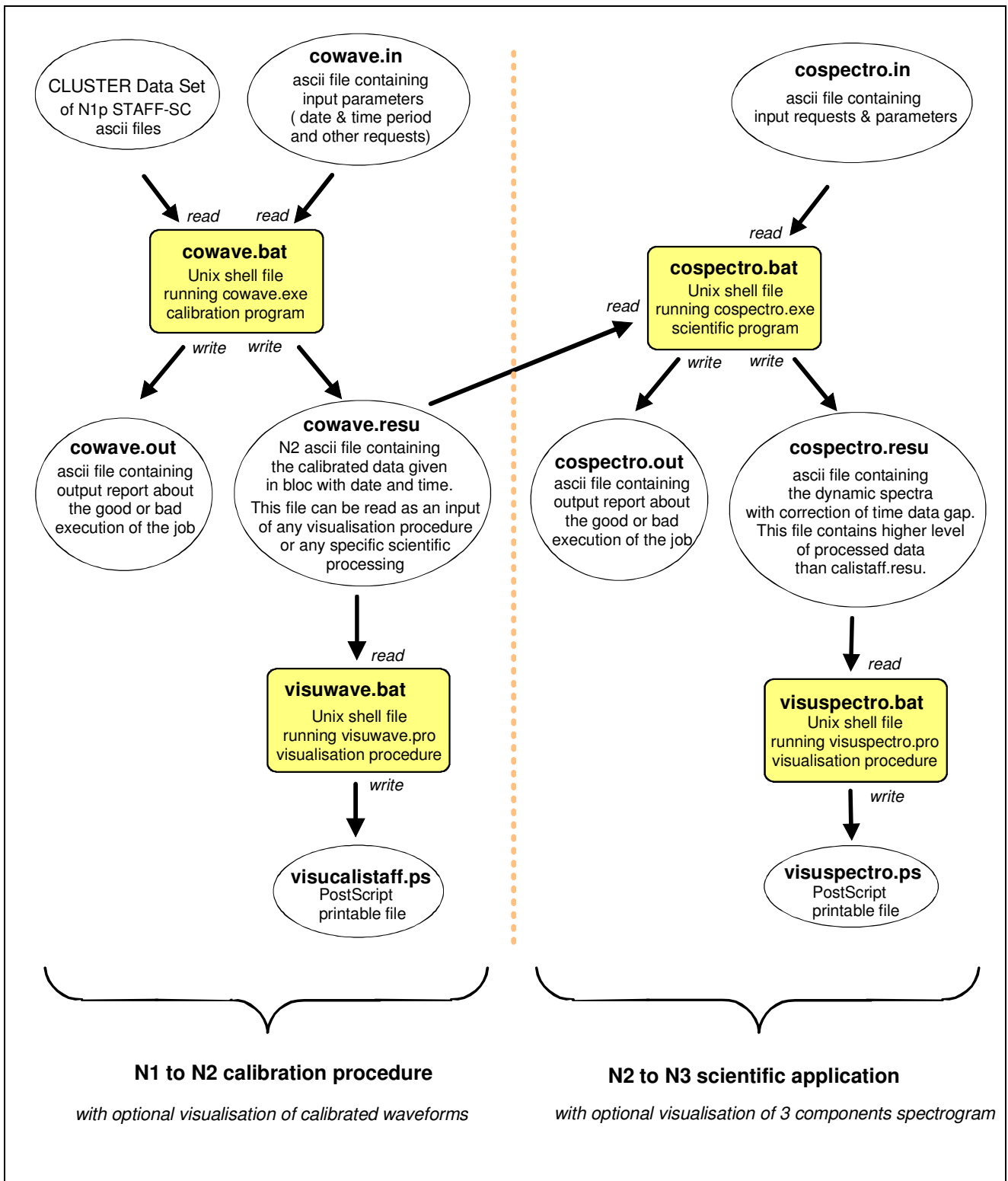


Figure 1 : schéma d'enchaînement des procédure pour calculer et visualiser un spectrogramme de données CLUSTER/ STAF-SCF calibrées.

3) Les différentes familles Roproc

a) différence entre les 3 familles

Les procédures Roproc sont classées en 3 familles, qui peuvent traiter aujourd'hui 3 types de données de plusieurs expériences de satellites ou bases d'observation différents :

- les données du genre « forme d'onde » : les **Wave_Roproc**
- les données du genre magnétomètre, ou suite de vecteurs datés : les **Vector_Roproc**
- les données d'orbite, ou de trajectoire : les **Orbit_Roproc**

Les différences entre les 3 familles sont liées à la nature des données :

- Les **Wave_Roproc** s'appliquent à des données régulièrement échantillonnées, à fréquence constante, d'où leur application pour toutes les expériences de type « forme d'onde » comme CLUSTER/STAFF-SC, ou GEOS/UBF. Les fichiers traités sont généralement volumineux, et le temps n'est repéré que par « bloc » de données, par exemple toutes les secondes pour une forme d'onde échantillonnée à 25 Hz. Dans la version actuelle, ces procédures ne traitent que 3 composantes.
- Les **Vector_Roproc** s'appliquent à des données de type vecteur, dépendant du temps ou d'un autre paramètre. Cette dépendance n'est pas forcément régulière, les vecteurs pouvant ne pas être régulièrement espacés. Ils peuvent être de dimension quelconque. Ce format est utilisé pour traiter les données de FGM de CLUSTER, car la datation utilisée dans les fichiers d'origine (date ISO, c'est à dire de la forme YY-MM-DDThh :mm :ss.mscZ) n'est pas assez précise pour estimer une fréquence d'échantillonnage constante. Les fichiers **Vector_Roproc** sont plus volumineux que les fichiers **Wave_Roproc**, car le temps, ou le paramètre de dépendance, est donné pour chaque vecteur.

Le format **Wave_Roproc** peut dans tous les cas être converti en un format **Vector_Roproc**, mais l'inverse n'est vrai que pour des vecteurs à 3 composantes, dépendant du temps, et régulièrement espacés. Néanmoins, dans le cas de CLUSTER/FGM, on peut estimer raisonnablement une fréquence d'échantillonnage constante et passer ces données au format **Wave_Roproc**, ce qui permet de faire des spectres, spectrogrammes, et calcul de polarisation d'onde sur ces données.

- Les **Orbit_Roproc** sont dédiées à des données de position dépendant du temps ; Le format correspondant est historique, mais il pourrait dans un temps ultérieur disparaître au profit du format plus général **Vector_Roproc**.

Les fichiers correspondant ont généralement pour nom *cowave.resu*, *covector.resu* ou *cosatpos.resu*. Dans les 3 cas, ces formats sont en ascii, et clairement lisibles même sans programme de lecture ; un exemple de chacun d'entre eux est donné dans les chapitres suivants.

b) terminologie des commandes

La terminologie des commandes Roproc est la suivante :

« préfixe »_ « nom de la procédure »

Par exemple **rcs_cowave** lancera une **Roproc_Cluster_Staff** de type cowave, ou calcul de forme d'onde calibrée, sur des données de CLUSTER/STAFF-SC. Elle est du type « Wave_Roproc ».

De même **rcf_visuvector** lancera une **Roproc_Cluster_Fgm** de type visuvector, ou visualisation d'un vecteur « time dependant », sur les données de FGM. Elle est du type « Vector_Roproc ».

c) liste des commandes disponibles

La liste des commandes disponibles pour chaque expérience est obtenue par la commande :

« préfixe »_menu

Par exemple la liste des Roproc disponibles pour CLUSTER/STAFF-SC est donnée par **rsc_menu**, tandis que celles pour CLUSTER/FGM est donnée par **rcf_menu**.

d) préfixes associés aux expériences

Les préfixes associés aux expériences traitées par chaque famille, sont les suivants :

Wave_Roproc		Vector_Roproc		Orbit_Roproc	
4 satellites CLUSTER					
rcc	CLUSTER / STAFF-SC	rcf	CLUSTER / FGM	rco	CLUSTER orbite
rce	CLUSTER / EFW *				
2 satellites GEOS					
rgu	GEOS / ULF S300	rgm	GEOS / S331 – Magnétom.	rgo	GEOS orbite
Station mobile					
rsu	Mobile Station / ULF				
rsp	Mobile Station / Photometre				
Fusée CUSP					
rcb	CUSP / B			rcp	CUSP position
rcj	CUSP / J				
Autre...					
Autre expérience de type forme d'onde		Autre expérience de type suite de vecteurs datés		Autre satellite, fusée ou station	

* pour EFW, il n'y a pas de base de fichier existante. La procédure **rce_cowave** lit un flat file créé par ISDAT, et le transforme au format cowave.resu, ce qui permet d'utiliser les procédures Wave_Roproc sur ces données, et faire par exemple des spectres et spectrogrammes. La création et la gestion des flat files est à la charge de l'utilisateur.

4) Réalisation de scripts pour le traitement en série

Les procédures Roproc sont assimilables à des commandes UNIX, et sont donc utilisables à l'intérieur d'un shell, ou d'un fichier de commande. L'intérêt est de pouvoir enchaîner plusieurs commandes successives, et de créer ainsi des « macro » commandes.

Par exemple, si on veut créer un grand nombre de spectrogrammes avec les données de STAFF-SC dans le repère MFA (Magnetic Field Aligned), il faut enchaîner plusieurs commandes de type *rcs*, et ceci un grand nombre de fois. Pour éviter d'avoir à répéter ces opérations fastidieuses, on peut créer un fichier de commandes, que l'on peut nommer par exemple *M_visuspectro_mfa.bat*, et dont le contenu sera le suivant :

```
rsc_cowave  
rsc_wave_to_mfa  
rsc_cospectro  
rsc_visuspectro
```

Exemple de fichier de macro-commandes

Au lancement de cette commande, on produira par *rsc_cowave* une forme d'onde selon les paramètres du fichier *cowave.in*, puis on transposera par *rsc_wave_to_mfa* le contenu du fichier *cowave.resu* dans le repère MFA, et on produira alors par *rsc_cospectro* le fichier *cospectro.resu* contenant les données du spectrogramme en repère MFA. Enfin on produira par *rsc_visuspectro* le fichier PostScript *visuspectro.ps* prêt à être visualisé.

Ce fichier *M_visuspectro_mfa.bat* doit avoir une permission d'exécution pour pouvoir être lancé comme une commande usuelle (faire *chmod +x g_visuspectro_mfa.bat*).

On peut, au moyen de ces fichiers de commandes, créer de très grosses procédures répétitives, en modifiant au besoin à l'intérieur du shell les paramètres du fichier *cowave.in*, comme la date et l'heure par exemple. C'est par ce type de procédure que les spectrogrammes de routines, disponible sur le serveur web du laboratoire (<http://www.cetp.ipsl.fr>), sont créés pour plusieurs jours consécutifs au moyen d'une seule commande.

II - LES PROCEDURES WAVE_ROPROC:

1) Généralités

Les procédures *Wave_Roproc* sont destinées aux données de type « forme d'onde », calibrées ou non, échantillonnées à fréquence constante (voir § I-3). La procédure de base est *cowave.bat*. Elle lit le fichier de télémessure N1 correspondant à la date indiquée dans le fichier de paramètre *cowave.in*, et calibre la séquence demandée suivant une série de fenêtres dont la taille est fixée par le nombre de points sur lesquels on applique le traitement de calibration à base de FFT. Le fichier de paramètre *cowave.in* est décrit en détail ci-après, et des exemples d'utilisation sont donnés.

Après obtention du fichier de données calibrées *cowave.resu*, on peut appliquer des « filtres » sur ce fichier pour en transformer la nature, mais pas le format. Par exemple, on peut appliquer des changements de repères ; le repère initial étant le repère SR2 (voir annexe 2), on peut transformer les données dans le repère GSE, ou encore le repère de variance minim ou le repère de la ligne de force (voir paragraphe II-3).

On peut visualiser les formes d'ondes du fichier *cowave.resu* par la procédure *visuwave.bat*.

Lorsque on dispose du fichier *cowave.resu*, correctement filtré et dans le repère voulu, on peut alors appliquer des traitements de niveau 3, comme des calculs de spectrogrammes, spectres moyens dans un intervalle de temps, puissance intégrée dans un intervalle de fréquence ou encore calcul de la polarisation des ondes (voir § II-4). Chaque programme de calcul génère un fichier *.*resu*, et dispose de son programme de visualisation qui lit ce fichier et génère un fichier de type *.*ps*, que l'on peut visualiser, imprimer ou inclure dans un document ou publication, après transformation éventuelle, par un logiciel comme LATEX, Adobe Illustrator, Micrograph Designer ou Corel Draw.

2) Le programme de calibration *cowave* (N1 vers N2)

a) but

Comme on vient de le voir, c'est donc le programme de départ, qui lit les fichiers de base (les fichiers N1p du genre 20010126_sat4_sc.n1p par exemple pour Cluster/STAFF-SC), extrait la période de données indiquées dans le *cowave.in*, effectue les traitements de calibration, filtrage, etc. et écrit les résultats dans le fichier *cowave.resu* avec un format général multi-expérience décrit ci-après. Ce programme doit être écrit pour chaque expérience, selon le format des données de base et les traitements de calibration spécifiques à chaque expérience.

b) fichier de paramètre *cowave.in*

Ce fichier est capital car non seulement il indique la période de temps que l'on veut calibrer, mais aussi la manière dont on calibre et le découpage en séquences consécutives. Un exemple est donné ci-dessous, ainsi que l'explication de chacun des paramètres.

Exemple de fichier cowave.in :

```
* 4
/users/ida/robert/cluster/staff/Wave_Roproc_Staff/gainantset/
26 01 2001      # day-month-year of period
11 30 0         # Hour-min-sec   of start period
3.10           # Duration of the time period (mn)
512            # Number of point of the window (256-65536)
0.1            # Frequency cut-off for calibration
0.3 00.00      # Min/Max frequencies for filtering
5              # Processing levels asked (1-5)
NBR           # NBR or HBR
```

Dans cet exemple, chaque ligne correspond a une série de 1 ou plusieurs paramètres dont la signification est détaillée ci-dessous.

Signification et limites de chaque paramètre

• Chemin du fichier N1 :

L'astérisque en premier caractère de la première ligne suivi d'un blanc et du numéro de satellite déclenche la recherche automatique de l'emplacement du fichier N1 dans l'arborescence de la machine **sawara**, à partir de la date donnée ligne 3. Sur une autre machine, ou si on veut traiter un autre fichier N1 que ceux de la base de **sawara**, il faut indiquer le chemin complet du fichier, soit par exemple :

```
/data1/ida.robert/cluster/staff/N1p/20010126_sat4_sc.n1p
```

La nomenclature des noms de fichiers est (année) (mois) (jour)_sat (num. satellite)_sc.n1p

• Chemin des fichiers de calibration :

En principe, ne pas le changer, sauf si on veut utiliser des fichiers de calibration autres que ceux mis en libre service, ou si on est sur une autre machine. Les noms de ces fichiers doivent être C1.cal, C2.cal, C3.cal et C4.cal.

• Date et heure :

Pas de format particulier, mais toujours dans l'ordre jour-mois-an et heure-minute-seconde, soit dans cet exemple le 26 janvier 2001 à 11h 30m 0s.

• Durée : En minutes décimales, pour toute la durée de signal à calibrer.

• Nombre de points de la séquence :

La durée du signal à traiter est ici de 3.10 minutes, et la calibration se fera par séquences ou fenêtres successives de Nbp=512 points, ce qui veut dire que les 3.10 minutes seront découpée en Ns séquence de 512 points. En Normal Bit Rate (que l'on sélectionne a la dernière ligne du fichier), le fréquence d'échantillonnage est de 25 Hz (alors qu'elle est de 450 Hz en High Bit Rate). Chaque séquence aura donc une durée de $512/25=20.48$ s. Comme on a demandé 3.1 minutes de signal, on traitera $Ns=3.10*60/20.48=9$ séquences de 20.48 s.

Le tableau ci-dessous donne les valeurs permises du nombre de points de chaque séquences, et les durées correspondantes en Normal Bit Rate et en High Bit Rate, pour CLUSTER/STAFF-SC.

Nombre de points	Durée en NBR (25 Hz)	Durée en HBR (450 Hz)
128	5.12 s *	0.2844 s
256	10.24 s *	0.5689 s
512	20.48 s *	1.138 s
1024	40.96 s *	2.275 s
2048	81.92 s soit 1 m 22 s	4.551 s
4096	163.8 s soit 2 m 44 s	9.102 s *
8192	327.7 s soit 5 m 28 s	18.20 s *
16384	655.4 s soit 10 m 55 s	36.40 s *
32768	1310.7 s soit 21 m 51 s	72.82 s soit 1 m 13 s
65536	2621.4 s soit 43 m 41 s	145.6 s soit 2 m 26 s

Les astérisques indiquent les valeurs recommandées si on veut ensuite faire des spectrogrammes. En effet, si on veut simplement visualiser une forme d'onde, il est recommandé d'utiliser une séquence longue, pour avoir une seule, ou un très petit nombre, de forme d'onde dans le fichier *cowave.resu*. Par exemple, on choisira Nbp=2048 et durée=1mn, ce qui produira une seule séquence dans le fichier *cowave.resu*, et un seul fichier Postscript quand on lancera la procédure *visuwave*.

Par contre, si on veut faire un spectrogramme, un spectre moyen ou une puissance intégrée (voir § II-4), on choisira une petite valeur de Nbp, par exemple 512, et une grande durée, par exemple ½ heure ou 3 heures comme dans les spectrogrammes de routine de la base Roplobank (accessible par <http://www.cetp.ipsl.fr>, « la mission CLUSTER »). Des valeurs de 512 et ½ heure conduiront à un spectrogramme dont les cellules en temps/fréquence seront respectivement de 20.48 s et 0.0488 Hz et il y aura 88 séquences en temps pour couvrir la ½ heure demandée.

• *Fréquence de coupure basse :*

La fréquence minimum de coupure dans le repère tournant sera de 0.1 Hz, ce qui est une valeur raisonnable dans la mesure où la fonction de transfert est nulle à la fréquence zero, la calibration n'étant possible qu'à partir du moment où la fonction de transfert n'est pas trop petite (voir annexe 1).

• *bornes de filtrage f_1 et f_2 :*

Après passage dans le repère fixe SR2, les fréquences ne sont plus les mêmes que dans le repère tournant initial de mesure. Pour avoir la même gamme de fréquences sur les 3 composantes fixes, on re-filtre ici le signal entre 0.3 Hz et sa fréquence maximum selon le bit rate (12.5 Hz ou 225 Hz). On aurait pu ici indiquer une valeur quelconque pour f_2 , 5.0 par exemple.

A noter : la fréquence de spin est d'environ 0.25 Hz dans le repère tournant, ce qui fait qu'un onde à une fréquence f dans le repère SR2 sera dédoublé en $f \pm f_s$ dans le repère tournant des antennes, et chacune des 2 fréquences sera calibrée avec une valeur de la fonction de transfert différente, et donc une sensibilité différente selon que l'onde sera majoritairement polarisée à gauche ou à droite par rapport à l'axe de spin. Cet effet est bien sur plus important a basse fréquence qu'à haute fréquence, et peu conduire à une mauvaise estimation de l'amplitude des ondes. Par exemple, il faut savoir qu'une onde de fréquence égale à f_s et polarisée à droite dans le plan de spin (et donc tournant dans le même sens que le satellite et à la même vitesse), aura une fréquence nulle dans le repère des antennes, et ne sera donc pas « vue » par les capteurs. Au contraire, une onde gauche à cette même fréquence f_s sera « vue » à $2f_s$, et donc bien mesurée.

- *Niveau du traitement :*

Le niveau de calibration demandé ici est le niveau 5 (maximum), les différents niveaux possibles étant résumés ci dessous :

- 1: Volts, signal brut non calibré, repère tournant, avec le continu sous forme de sinusoïde importante
- 2: Volts, signal non calibré mais où le spin a été retiré : repère tournant, sans le champ continu
- 3: nTesla, signal calibré, spin retiré, repère tournant, sans le champ continu
- 4: nTesla, signal calibré, spin retiré, repère fixe SR2, sans le champ continu
- 5: nTesla, signal calibré, spin retiré, repère fixe SR2, mais avec le champ continu : le signal de spin a été calibré séparément, et les composantes du champ continu calculées en repère fixe ont été rajoutées sur les composantes X et Y. A ce niveau de calibration, les 2 composantes X et Y peuvent être comparées avec les données à haute résolution du magnétomètre de FGM.

- *Bit rate :*

NBR ou HBR. Ici, on ne sélectionne que les données en Normal Bit Rate (fréquence d'échantillonnage en NBR :25 Hz, et 450 Hz en HBR). Attention, majuscules obligatoires.

c) lancement du programme

Une fois que les paramètres du fichier *cowave.in* sont correctement fixés, la procédure ***cowave.bat*** pour les données de CLUSTER/STAFF-SC est lancée par la commande :

rcs_cowave

qui vas générer dans le directory courant le compte rendu d'exécution *cowave.out* et le fichier résultat *cowave.resu* dont le format est décrit ci-après.

d) compte rendu d'exécution cowave.out

Ce fichier *cowave.out*, correspondant aux paramètres du *cowave.in* précédent, est important car il contient tous les diagnostics d'éventuel mauvais déroulement de la procédure, ainsi qu'un certain nombre d'informations utiles. En cas d'anomalies mineures, mais gênantes, comme des discontinuité en temps par exemple, ou des saturations de la forme d'onde brute, celles-ci figurent en clair dans ce compte rendu.

e) description du fichier résultat cowave.resu

Ce fichier a un format général, utilisé pour d'autres expériences spatiales en dehors de CLUSTER (satellites GEOS-1 et 2, données sol correspondantes, fusée CUSP...). Son entête contient des informations globales, et chaque séquence est datée et peut contenir une forme d'onde à différents niveaux de traitement.

```
START OF FILE HEADER
START COMMENTS
File type cowave.resu: Result of cowave.exe
The cowave.resu files are files containing level N1 or N2 waveform data.
The format of this file is common for the processing of ULF Magnetic data
from experiments on spacecraft GEOS-1, GEOS-2, CLUSTER and Moving Ground
Station. It allows to compare N1 data and N2 calibrated data and can be
the N2 input data level to produce calibrated waveform plots, spectra,
spectrograms, integrated power, or any else N3 results.
File is structured as:
. File Header, including comments and fixed data,
. Several windows data, each window corresponds to a waveform,
  and could contains several data panels, corresponding themselves to
  various levels of processing.
Author: Patrick ROBERT, CETP, 1996-2000.          revision October 2001.
          Revision for ground mobile station data, January 2002.
          Revision for any kind of waves, August 2002.
          Revision for general titles management, January 2003.
END COMMENTS
-----
Roproc Data Set number:
4
Full Title:
CLUSTER/STAFF-SC/Tango (#4)
Project:
CLUSTER
Experiment:
STAFF-SC
Spacecraft ID:
Tango
Spacecraft number:
4
Field Nature:
B Magnetic Field
Input data file:
/datal/cluster/STAFF/Nlp/0101/010126_sat4_sc.nlp
Directory of calibration data:
/users/ida/robert/cluster/staff/Wave_Roproc_Staff/gainantset/
Date, time and duration asked:
26 01 2001 11:30:00 duration= 3.10 mn.
Number of points for the windows:
512
Frequency cut-off for the calibration:
0.100000E+00
Frequency bounds for the filtering in fixed system:
0.300000E+00 0.125000E+02
Panels asked:
0 0 0 0 1 0 0
Number of panels per window:
1
Number of windows:
10
Bit Rate:
NBR
Sampled Frequency:
0.250000E+02
Spin Frequency:
0.249530E+00
R. Asc. and Dec. of Spin axis in GEI:
0.102630E+03 -0.638400E+02
END OF FILE HEADER
-----
START OF WINDOW# 1
Starting date (nJul00,iday,imonth,iyear):
391 26 01 2001
Starting time (ih, im, is, ims, imicro):
11 30 00 207 446
Wave form anomaly level:
0
-----
START OF WINDOW# 1/PANNEL# 1 data
Processing step:
5
Title:
step 5: Calibrated data in SR2 system [nT] with DC ( 0.30- 12.50 Hz)
Time resolution (s):
0.400000E-01
Number of curves:
3
Number of points per curves:
512
Curves data:
0.153764E+02 0.184120E+02 0.145867E-02
0.153773E+02 0.184045E+02 0.203346E-03
0.153777E+02 0.183942E+02 -0.291320E-02
0.153808E+02 0.183900E+02 -0.726937E-02
0.153847E+02 0.183856E+02 -0.101087E-01
0.153932E+02 0.183839E+02 -0.133473E-01
0.154079E+02 0.183832E+02 -0.183715E-01
0.154164E+02 0.183844E+02 -0.183692E-01
```

```
...

0.152066E+02 0.185231E+02 -0.792746E-02
0.152000E+02 0.185089E+02 -0.614141E-02
0.151933E+02 0.184921E+02 -0.410617E-02
0.151870E+02 0.184741E+02 -0.203343E-02
0.151771E+02 0.184575E+02 0.355870E-03
END OF WINDOW#      1/PANNEL# 1 data
-----
START OF WINDOW#      2
Starting date (nJul00,iday,imonth,iyear):
 391 26 01 2001
Starting time (ih, im, is, ims, imicro):
 11 30 20 687 301
Wave form anomaly level:
 0
-----
START OF WINDOW#      2/PANNEL# 1 data
Processing step:
 5
Title:
step 5: Calibrated data in SR2 system [nT] with DC ( 0.30- 12.50 Hz)
Time resolution (s):
 0.400000E-01
Number of curves:
 3
Number of points per curves:
 512
Curves data:
 0.183870E+02 0.192051E+02 -0.170291E+00
 0.184137E+02 0.191727E+02 -0.154691E+00
 0.184379E+02 0.191424E+02 -0.134834E+00
 0.184620E+02 0.191105E+02 -0.118801E+00
 0.184826E+02 0.190754E+02 -0.993200E-01

*****

etc...

*****

END OF WINDOW#      9/PANNEL# 1 data
-----
START OF WINDOW#      10
Starting date (nJul00,iday,imonth,iyear):
 391 26 01 2001
Starting time (ih, im, is, ims, imicro):
 11 33 04 526 108
Wave form anomaly level:
 0
-----
START OF WINDOW#      10/PANNEL# 1 data
Processing step:
 5
Title:
step 5: Calibrated data in SR2 system [nT] with DC ( 0.30- 12.50 Hz)
Time resolution (s):
 0.400000E-01
Number of curves:
 3
Number of points per curves:
 512
Curves data:
 0.135957E+02 0.208781E+02 -0.886869E-02
 0.136148E+02 0.209031E+02 -0.646010E-03
 0.136316E+02 0.209248E+02 0.419098E-02
 0.136480E+02 0.209423E+02 0.111612E-01
 0.136619E+02 0.209515E+02 0.137446E-01

...

0.136898E+02 0.213638E+02 -0.426859E-01
0.136915E+02 0.214111E+02 -0.321814E-01
0.136928E+02 0.214599E+02 -0.245787E-01
0.136968E+02 0.215090E+02 -0.150862E-01
END OF WINDOW#      10/PANNEL# 1 data
END OF FILE
```

f) vérification de la cohérence du fichier *cowave.resu*

Le fichier *cowave.resu* peut être simplement relu par la procédure ***readwave.bat***, qui va vérifier que le fichier créé a un format et une terminaison correcte. Cette procédure est lancée par la commande :

r_{cs}_readwave

La source (en PV-Wave compatible IDL) de cette procédure est publique, et peut être utilisée pour construire d'autres traitements éventuels que ceux proposés ici.

g) visualisation des formes d'ondes du fichier *cowave.resu*

Le contenu du fichier *cowave.resu* peut être visualisable par la procédure ***visuwave.bat***. Celle-ci est lancée par la commande :

r_{cs}_visuwave

Cette procédure produit un fichier PostScript par séquence du fichier *cowave.resu* (donc attention à ne pas l'utiliser sur un fichier destiné à faire des spectrogrammes, par exemple). Les fichiers PostScript créés auront la nomenclature suivante :

visuwave_w001_1.ps

Où *w001* indique la séquence (ou fenêtre) 001, et *1* est relatif au satellite 1. Un exemple est donné au § V. On a la possibilité de fixer les bornes des 3 axes *xyz* du graphe au moyen d'un fichier optionnel, le *visuwave.in*. Si ce fichier n'existe pas, il est créé par défaut avec les valeurs ci-dessous. S'il existe avant de lancer la commande ***r_{cs}_visuwave***, ses valeurs sont prises en compte pour la visualisation.

Le contenu par défaut du fichier optionnel *visuwave.in* est le suivant :

```
0  0. 0.  0. 0.  0. 0.  # nbp ymi1 yma1 ymi2 yma2 ymi3 yma3  for figure limits
```

3) les filtres pour les changements de repères

a) introduction

Ces filtres ne modifient pas le format des fichiers *cowave.resu*, mais effectuent simplement des traitements, notamment des changements de repère, sur les données calibrées initialement en repère SR2. Dans la mesure où les changements de repère nécessitent 3 composantes traitées de la même manière, ces filtres ne peuvent pas opérer sur le niveau 5 de calibration, donc seul le niveau 4 est autorisé en entrée. Les niveaux 1 à 3 sont rejetés par la procédure, le niveau 5 est accepté mais le continu est enlevé avant changement de repère.

Ces procédures font d'abord une copie du fichier *cowave.resu* initial dans le fichier *cowave.resu.old*, puis génèrent un nouveau fichier *cowave.resu* qui contient les données transformées.

b) wave_to_GSE

La transformation d'un fichier *cowave.resu* de données calibrées du repère SR2 au repère GSE se fait par la commande :

rcs_wave_to_gse

Avec les restrictions ci dessus.

c) wave_to_GSM

La transformation d'un fichier *cowave.resu* de données calibrées du repère SR2 au repère GSM se fait par la commande :

rcs_wave_to_gsm

Avec les restrictions ci dessus.

d) wave_to_MVA

La transformation d'un fichier *cowave.resu* de données calibrées du repère SR2 au repère de variance minimum (MVA) se fait par la commande :

rcs_wave_to_mva

Toujours avec les restrictions ci dessus.

e) wave_to_MFA

La transformation d'un fichier *cowave.resu* de données calibrées du repère SR2 au repère MFA se fait par la commande :

rcs_wave_to_mfa

Toujours avec les restrictions ci dessus. De plus, cette procédure accède à la base des PPD des données du magnétomètre FGM, et effectue un traitement de lissage sur ces données. Il faut donc que d'une part ces données soient présentes, et d'autre part qu'elles ne présentent pas de discontinuités pendant la période de signal à transformer, sinon la procédure avorte.

f) wave_filter

La transformation d'un fichier *cowave.resu* de données calibrées en un fichier où les formes d'ondes ont été filtrées entre f1 et f2 se fait par la commande :

rcs_wave_filter

Cette commande nécessite un fichier de paramètres *wave_filter.in* de la forme suivante :

<pre>0.2 5. # f1,f2 for filter (0,0 = no filter, 0.1,0 = 0.1,fmax)</pre>

4) les traitements de niveau N3

a) préambule

Tous ces traitements lisent les données calibrées d'un fichier *cowave.resu*. Ces données peuvent avoir au préalable été filtrées ou changées de système de coordonnées par les filtres décrits précédemment au § II-3. Ils produisent d'autres fichiers de type **.resu*, auxquels sont associés des procédures de visualisations générant des fichier PostScripts **.ps*.

b) spectrogrammes

- Le calcul d'un spectrogramme à partir d'une série de formes d'onde contenues dans un fichier *cowave.resu* est fait par la procédure ***cospectro.bat***, qui est lancée par la commande :

rcs_cospectro

Cette procédure génère un rapport d'exécution *cospectro.out*, et un fichier résultat *cospectro.resu* qui contient, outre les spectrogrammes de 3 composantes, la puissance intégrée dans la totalité de la bande de fréquence du fichier *cowave.in*, et la puissance intégrée dans une bande de fréquence paramétrable. Il contient également la valeur moyenne du champ continu perpendiculaire au spin, si les données du fichier *cowave.resu* on été produites avec l'option 5 du traitement. En dehors de cette option, cette valeur n'a pas une grande signification.

Cette procédure nécessite un fichier *cospectro.in*, où on défini la bande de fréquence paramétrable où on calculera la puissance intégrée ($f_2=0$. signifie $f_2=f_{max}$, selon la fréquence d'échantillonnage), ainsi que l'option définissant l'usage des composantes cartésiennes (x,y,z) ou circulaires (mode gauche, droit, z). Cette dernière n'a un sens physique que si le fichier *cowave.resu* a été au préalable transformé pour avoir les données dans le repère magnétique MFA (par le filtre ***wave_to_mfa.bat***). On indique également dans ce fichier si on veut apodiser les formes d'ondes avant le calcul des spectres (en principe, celles-ci l'on déjà été lors de la procédure de calibration, mais on peut par ce paramètre forcer une seconde apodisation).

Si le fichier *cospectro.in* n'existe pas au lancement de la procédure ***rcs_cospectro***, il est créé avec les valeurs par défaut ci-dessous ; ilest de la forme :

1.	0.	# f1,f2 for integrated power; 0 = fmax, following bit rate
0		# 1 for Left/Right/Z, 0 for X/Y/Z
0		# 0 for no apodisation, 1 for trapezoidal apodisation

Le calcul d'un spectrogramme tiens compte des anomalies de données détectées lors de la création du fichier *cowave.resu*. En particulier, il met des valeurs nulles en l'absence de données, et un code particulier (reconnu par le programme de visualisation) quand les données présentent des anomalies rendant impossible le calcul du spectre courant (bloc de données incomplet, saturation des formes d'ondes, etc.).

La résolution des cellules $\Delta t \cdot \Delta f$ du spectrogramme est liée à la fenêtre courante de Nbp points définie dans le fichier *cowave.in*, qui fixe la résolution en temps Δt . La résolution en fréquence est simplement $1/\Delta t$. Le nombre de colonne du spectrogramme est égal au nombre de séquences du fichier *cowave.in*.

- La visualisation d'un spectrogramme à partir des données contenues dans le fichier *cospectro.resu* est fait par la procédure ***visuspectro.bat***, qui est lancée par la commande :

rcs_visuspectro

Cette procédure génère un fichier PostScript *visuspectro.ps*, dont un exemple est donné au § V. Le calcul des bornes en fréquence et celui des puissances minimum et maximum (en log) associée à la mire de couleur est automatique, mais on peut les forcer au moyen du fichier *visuspectro.in*. Si ce fichier est absent lors du lancement de la commande **rcs_visuspectro**, les valeurs par défaut indiquées ci-dessous sont prises en compte ; s'il est présent, ce sont les valeurs indiquées dans ce fichier qui sont prises en compte. Les valeurs (0.,0.) entraînent le calcul automatique des bornes.

Le contenu du fichier optionnel *visuspectro.in* est le suivant :

```
0. 0. 0. 0. # f1, f2, Pmin,Pmax for figure limits
```

c) spectres moyens intégrés sur un intervalle de temps

Pour calculer un spectre moyen sur une période de temps, il faut au préalable avoir créé un fichier *cowave.resu*, puis un fichier *cospectro.resu* sur cette même période. La durée d'un spectre moyen correspond au Δt du spectrogramme, donc à la fenêtre courante de Nbp points définie dans le fichier *cowave.in*, la résolution en fréquence étant simplement $1/\Delta t$.

- Le calcul et la visualisation du spectre moyen à partir des données contenues dans le fichier *cospectro.resu* est fait par la procédure ***visuspectra.bat***, qui intègre en temps tous les spectres successifs du spectrogramme. Cette procédure est lancée par la commande :

rcs_visuspectra

Cette procédure génère un fichier PostScript *visuspectra.ps*, dont un exemple est donné au § V. Le calcul des bornes est automatique, mais on peut les forcer au moyen du fichier *visuspectra.in*. Les valeurs en ordonnée sont indiquées en log. Si ce fichier est absent lors du lancement de la commande **rcs_visuspectra**, les valeurs par défaut indiquées ci-dessous sont prises en compte ; s'il est présent, ce sont les valeurs indiquées dans ce fichier qui sont prises en compte. Les valeurs (0.,0.) entraînent le calcul automatique des bornes.

Le contenu du fichier optionnel *visuspectra.in* est le suivant :

```
0. 0. 0. 0. # f1, f2, Yl1,Yl2 for figure limits
```

d) puissance moyenne dans une bande de fréquence au cours du temps

Pour calculer une puissance intégrée dans une bande de fréquence au cours du temps, il faut au préalable avoir créé un fichier *cowave.resu*, puis un fichier *cospectro.resu* sur la période voulue. La durée d'un spectre moyen correspond au Δt de la puissance au cours du temps, donc à la fenêtre courante de Nbp points définie dans le fichier *cowave.in*.

- Le calcul et la visualisation de cette puissance à partir des données contenues dans le fichier *cospectro.resu* est fait par la procédure **visupower.bat**, qui intègre en fréquence tous les spectres successifs du spectrogramme. Cette procédure est lancée par la commande :

rcs_visupower

Cette procédure génère un fichier PostScript *visupower.ps*, dont un exemple est donné au § V. Elle nécessite un fichier *visupower.in*, où on définit les bornes de la bande de fréquence à intégrer. Ces bornes doivent bien sûr être comprises entre les bornes de filtrage définies dans le fichier *cowave.in*. Les valeurs (0.,0.) entraînent le calcul automatique des bornes.

Ce fichier *visupower.in* est de la forme :

```
0.2 12.5 # F1, f2: frequency range for integrated power
0. 0.    # Y1, Y2 for figure limits
```

e) polarisation des ondes

On peut calculer la polarisation des ondes contenues dans le fichier *cowave.resu* de la même manière que l'on calcule un spectrogramme. Ce calcul n'aura un sens physique que si le fichier *cowave.resu* a été au préalable transformé pour avoir les données dans le repère magnétique MFA (par le filtre *wave_to_mfa.bat*).

Le calcul de la polarisation se fait dans une cellule $\Delta t \cdot \Delta f$. La largeur en temps Δt correspond à la fenêtre courante de Nbp points définie dans le fichier *cowave.in*. La largeur en fréquence Δf est simplement $1/\Delta t$ comme dans le calcul du spectrogramme.

- Le calcul de la polarisation à partir de la série de formes d'onde contenues dans le fichier *cowave.resu* est fait par la procédure **copolar.bat**, qui est lancée par la commande :

rcs_copolar

Cette procédure génère un rapport d'exécution *copolar.out*, et un fichier résultat *copolar.resu* qui contient les résultats du calcul de polarisation.

Le fichier *copolar.resu* contient dans chaque cellule $\Delta t \cdot \Delta f$:

- la puissance totale,
- la direction de la normale d'onde (dans le repère du fichier *cowave.resu*),
- l'excentricité de l'ellipse de polarisation ,
- la direction du grand axe de l'ellipse.

- La visualisation de cette polarisation à partir des données contenues dans le fichier *copolar.resu* est fait par la procédure **visupolar.bat**, visualise les résultats ci-dessus sous forme de diagramme fréquence-temps. Cette procédure est lancée par la commande :

rcs_visupolar

Cette procédure génère un fichier PostScript *visupolar.ps*, dont un exemple est donné au § V. Elle nécessite un fichier *visupolar.in*, où on définit un seuil au delà duquel on visualise les données ; en effet, visualiser la polarisation du bruit de fond n'a pas grand intérêt, et rend confus l'image à interpréter. Pour avoir une idée du seuil, exprimé en $\log(nT^2/Hz)$, il suffit d'examiner la mire du spectrogramme correspondant, ou de faire un premier passage avec une valeur basse.

Ce fichier *visupolar.in* est de la forme :

```
-5.5 # threshold for visualisation (log nT2/Hz)
```

5) Les traitements à 4 satellites

a) production d'un jeu de 4 fichiers *cowave.resu*

Pour calibrer les données des 4 satellites, on pourrait lancer 4 fois successives la procédure *cowave.bat*, en changeant à chaque fois le numéro de satellite dans le fichier *cowave.in*. Mais d'abord cette manière de procéder est fastidieuse, et de plus les fichiers *cowave.resu* seraient écrasés à chaque lancement de la procédure *cowave.bat*. Pour remédier à cela, la procédure *cowave4.bat* permet en une seule opération de générer les 4 fichiers *cowave.resu*, qui sont rangés dans un directory dont le nom est demandé lors du lancement de la procédure. Celle-ci est lancée par la commande :

rcs_cowave4

Si le directory indiqué n'existe pas, il est créé ; si il existe déjà, il est purgé de ses fichiers. Après exécution de la procédure, le directory résultat contient les fichiers suivants :

cowave_1.in	cowave_1.out	cowave_1.resu
cowave_2.in	cowave_2.out	cowave_2.resu
cowave_3.in	cowave_3.out	cowave_3.resu
cowave_4.in	cowave_4.out	cowave_4.resu

On peut alors générer les fichiers PostScript de type *visuwave_w001_1.ps* (où w001 indique la séquence 001, et _1 est relatif au satellite 1) pour l'ensemble des 4 fichiers *cowave_*.resu* par la procédure *visuwave4.bat*, lancé par la commande :

rcs_visuwave4

Si les fichiers *cowave_*.resu* contiennent par exemple 2 séquences, le directory résultat sera complété par les fichiers suivants :

visuwave_w001_1.ps	visuwave_w002_1.ps
visuwave_w001_2.ps	visuwave_w002_2.ps
visuwave_w001_3.ps	visuwave_w002_3.ps
visuwave_w001_4.ps	visuwave_w002_4.ps

b) production de 4 spectrogrammes

De la même manière que pour la procédure *cowave4.bat*, il est possible de générer en une seule commande les 4 fichiers *cospectro_*.resu* au moyen de la procédure *cospectro4.bat*. Il faut bien sûr avoir déjà créé les 4 fichiers *cowave_*.resu* par la procédure *cowave4.bat* ci dessus, et lancer la commande suivante :

rcs_cospectro4

Cette commande demande bien sûr le nom du directory résultat où sont rangés les 4 fichiers *cowave_*.resu*. Si le directory n'existe pas, ou si les fichiers *cowave_*.resu* n'existent pas, la commande avorte. Si d'anciens fichiers *cospectro_*.resu* existent, ils sont détruits et remplacés par les nouveaux. Après exécution de cette commande, le directory résultat est complété par les fichiers suivants :

```
cospectro_1.resu
cospectro_2.resu
cospectro_3.resu
cospectro_4.resu
```

On peut alors visualiser ces 4 fichiers et produire les 4 fichiers *visuspectro_*.ps* par la procédure ***visuspectro4.bat***, qui est lancée par la commande :

rcs_visuspectro4

Cette commande demande bien sûr le nom du directory résultat où sont rangés les 4 fichiers *cospectro_*.resu*. Après exécution de cette commande, le directory résultat est complété par les fichiers suivants :

```
visuspectro_1.ps
visuspectro_2.ps
visuspectro_3.ps
visuspectro_4.ps
```

c) visualisation d'un spectrogramme avec les 4 composantes Bz de chaque satellite

Il est possible, au lieu de générer 4 fichiers *visuspectro_*.ps*, chacun d'entre eux affichant les 3 spectrogrammes Bx, By, Bz, de générer un seul fichier *visuspectro_4Bz.ps*, qui affiche sur la même page les 4 spectrogrammes Bz, ce qui permet plus facilement les comparaisons entre satellite, mais bien sûr sur une seule composante. Ce fichier *visuspectro_4Bz.ps* est généré par la commande :

rcs_visuspectro_4Bz

Cette commande demande bien sûr le nom du directory résultat où sont rangés les 4 fichiers *cospectro_*.resu*. Après exécution de cette commande, le directory résultat est complété par le fichier suivant :

```
visuspectro_4Bz.ps
```

Un exemple de ce fichier est donné au § V.

d) production de 4 spectres moyens Bx, By, Bz

De la même manière que pour la procédure ***visuspectro4.bat***, il est possible de générer en une seule commande les 4 fichiers *visuspectra_*.ps* au moyen de la procédure ***visuspectra4.bat***. Il faut bien sûr avoir déjà créé les 4 fichiers *cospectro_*.resu* par la procédure ***cospectro4.bat*** ci dessus, et lancer la commande suivante :

rcs_visuspectra4

Cette commande demande bien sûr le nom du directory résultat où sont rangés les 4 fichiers *cospectro_*.resu*. Si le directory n'existe pas, ou si les fichiers *cospectro_*.resu* n'existent pas, la commande avorte. Si d'anciens fichiers *visuspectra_*.ps* existent, ils sont détruits et remplacés par les nouveaux.

Après exécution de cette commande, le directory résultat est complété par les fichiers suivants :

```
visuspectra_1.ps  
visuspectra_2.ps  
visuspectra_3.ps  
visuspectra_4.ps
```

e) production de 4 Puissances intégrées Bx, By, Bz

De la même manière que ci dessus, il est possible de générer en une seule commande les 4 fichiers *visupower_*.ps* au moyen de la procédure ***visupower4.bat***. Il faut bien sûr avoir déjà créé les 4 fichiers *cospectro_*.resu* par la procédure ***cospectro4.bat*** ci dessus, et lancer la commande suivante :

rcs_visupower4

Cette commande demande bien sûr le nom du directory résultat où sont rangés les 4 fichiers *cospectro_*.resu*. Si le directory n'existe pas, ou si les fichiers *cospectro_*.resu* n'existent pas, la commande avorte. Si d'anciens fichiers *visupower_*.ps* existent, ils sont détruits et remplacés par les nouveaux. Après exécution de cette commande, le directory résultat est complété par les fichiers suivants :

```
visupower_1.ps  
visupower_2.ps  
visupower_3.ps  
visupower_4.ps
```

6) utilitaires divers:

a) copie dans le répertoire courant d'un jeu d'exemple de fichier *.in

Pour utiliser les procédures ***Wave_Roproc***, il faut toujours indiquer les paramètres d'entrée dans des fichiers **.in*, comme *cowave.in*, *cospectro.in*, *visupolar.in*, etc.

Pour avoir des exemple sous la main, la commande suivante permet de recopier dans le directory courant un exemple de tous les fichier **.in* :

rcs_copy_in

Après exécution, le directory courant contient les fichiers :

```
cowave.in      wave_filter.in  cospectro.in   visupower.in   visupolar.in  
libloc.in      listhead.in    test_time.in   tgeigse.in
```

b) fichier readme.txt donnant la procédure d'installation de la chaîne Wave_Roproc

Si on souhaite installer les procédures ***Wave_Roproc*** sur une autre machine SUN/Solaris, le mode d'emploi est donné par la commande :

rcs_readme

qui liste le fichier README.txt du package (il faut bien sûr l'avoir déjà installé ailleurs ; sinon, on peut aller voir directement ce fichier au début de l'arborescence du package).

c) liste des fichiers N1p disponibles dans la base

La base des fichiers N1p sur la machine **sawara** du CETP est constamment complétée par l'arrivée régulière des CR-ROM et de leur traitement de niveau N0 → N1. Pour obtenir la liste de l'ensemble des fichiers N1p existants, il suffit de taper la commande :

rcs_**l**ist_**n**1p **AA MM**

où AA et MM sont l'année et le mois, comme 01 07 pour juillet 2001. La nomenclature des noms des fichiers WAVE/SC N1p est :

010722_sat3_sc.n1p

pour le satellite 3 le 22 juillet 2001. Il y a un directory par mois. Cette commande n'est valable que sur la machine **sawara**, où les fichiers N1p sont rangés dans le directory /data1/cluster/STAFF/N1p, ce qui n'est bien sûr pas forcément le cas sur une autre machine ou pour une autre expérience.

d) lecture et decommutation des entête de bloc d'un fichier N1p

Il peut être utile, lors de la mise au point de programmes ou de test de fichier, d'avoir la liste des entête de bloc d'un fichier N1p, notamment pour tester la consécuitivité en temps des blocs. La procédure **libloc.bat** lit l'ensemble du fichier N1p, et génère un fichier **libloc.resu** contenant uniquement les entête de bloc. Ce fichier peut être long, et contenir par exemple 162051 lignes correspondant aux 162051 blocs du fichier N1 de l'exemple ci-dessous. Cette procédure génère également un rapport d'exécution **libloc.out**. Elle est lancée par la commande :

rcs_**l**ib**l**oc

Cette procédure nécessite un fichier **libloc.in**, contenant le chemin complet du fichier N1p à analyser, comme par exemple :

/data1/ida/robert/cluster/staff/N1p/010226_sat3_sc.n1p

Un exemple du fichier **libloc.resu** est donné ci dessous :

```
1 SAT: 3 N: 25 DT: 2001/02/26 00:30:04.985.177 ST: 0f00 0002 108.061920
2 SAT: 3 N: 25 DT: 2001/02/26 00:30:05.985.177 ST: 2f00 0002 197.864594
3 SAT: 3 N: 25 DT: 2001/02/26 00:30:06.985.177 ST: 2f00 0002 287.667267
4 SAT: 3 N: 25 DT: 2001/02/26 00:30:07.985.177 ST: 2f00 0002 17.489862

...

162049 SAT: 3 N: 25 DT: 2001/02/26 10:33:41.755.209 ST: 0000 0000 117.740028
162050 SAT: 3 N: 25 DT: 2001/02/26 10:33:42.755.209 ST: 0000 0000 207.528702
162051 SAT: 3 N: 25 DT: 2001/02/26 10:33:42.755.209 ST: 0000 0000 207.528702
```

e) liste des entête de fichier d'une série de fichier N1p

Il peut être utile d'avoir la liste des entêtes des fichiers N1p, pour une série de fichiers N1p, afin d'obtenir par exemple l'évolution de la fréquence de spin au cours du temps, ou l'évolution de la direction de l'axe de spin.

Ceci peu être obtenu en lançant la commande :

rsc_listhead

Cette procédure nécessite un fichier *listhead.in*, contenant les noms des fichiers dont on doit lire l'entête. un exemple est donné ci-dessous :

```
000916_sat2_sc.nlp
000905_sat3_sc.nlp
010101_sat1_sc.nlp
010101_sat2_sc.nlp
010101_sat3_sc.nlp
010101_sat4_sc.nlp
```

Cette procédure génère un fichier *listhead.out*, donnant la liste des entêtes, comme dans l'exemple ci-dessous :

```
-----
header des fichiers n1
-----
File      rasc    dec    Srate    Fs      Ts      Spha    Cmx     Cmy     Cnz     A1      A2
000916_sat2_sc.n1  70.40  -65.50  13.579570  0.226326  4.4184  333.860  769.00   0.00   0.00  -0.0100  0.0100
000905_sat3_sc.nlp  70.60  -66.00  14.965550  0.249426  4.0092  333.940  760.00   0.00   0.00   0.0000  0.0000
010101_sat1_sc.nlp  94.57  -60.88  14.995017  0.249917  4.0013  333.926  761.30  -0.10   0.10   0.0000  0.0300
010101_sat2_sc.nlp  94.00  -61.12  14.943371  0.249056  4.0152  333.859  769.20  -0.20   0.00  -0.0100  0.0200
010101_sat3_sc.nlp  93.95  -61.08  14.982331  0.249706  4.0047  333.942  760.40   0.20  -0.10  -0.0800  -0.0300
010101_sat4_sc.nlp  94.14  -61.05  14.974176  0.249570  4.0069  333.889  769.60  -0.10  -0.10   0.0300  0.0800
-----
Termine
-----
```

On retrouve notamment dans l'ordre : nom du fichier, ascension droite et déclinaison de l'axe de spin dans le repère inertiel GEI (degrés), nombre de tours par secondes, fréquence de rotation (Hz), période de rotation (s).

f) test du temps donné dans les entête de bloc des fichiers N1p

Ce programme a été fait pour tester la consécuitivité du temps dans les blocs de datation. Il permet notamment de voir comment est rafraîchi le temps au niveau des millisecondes et des microsecondes. Il a permis de voir que les microsecondes ne se rafraîchissaient pas correctement dans l'actuelle version des fichiers N1p.

La procédure est exécutée en lançant la commande :

rsc_test_time

Cette procédure nécessite un fichier *test_time.in*, contenant le nom du fichier à lire, et le nombre de blocs à décrire. Un exemple est donné ci-dessous :

```
/data1/ida/robert/cluster/staff/N1p/010101_sat1_sc.nlp
55    # nombre de blocs
```

Cette procédure génère le fichier *test_time.out*, dont un exemple est donné ci-dessous.

Exemple de fichier *test_time.out*

```
-----
verification du temps des blocs
-----
path du fichier nl ?
/data1/ida/robert/cluster/staff/Nlp/010101_sat1_sc.nlp
nombre de bloc a lire.
  55
-----
readind cluster.n1 file

file header:
Right Asc., Declin. =    94.5700   -60.8800
Spin rate, Spin phase=    2.71859    333.926
Centre of masse     =    761.300   -1.00000E-01    1.00000E-01
Euler Angle 1 and 2 =    0.    3.00000E-02

fe,fs,dt,ts=  0.    4.53099E-02  0.    22.0702
duree=  0.
rotdeg0=    333.926
  1 SAT: 1 N:  25 DT: 2001/01/01 00:00:02.571.117 ST: 0000 0000 331.853638
  2 SAT: 1 N:  25 DT: 2001/01/01 00:00:03.571.117 ST: 0000 0000  61.840134
  3 SAT: 1 N:  25 DT: 2001/01/01 00:00:04.571.117 ST: 0000 0000 151.806671
  4 SAT: 1 N:  25 DT: 2001/01/01 00:00:05.571.117 ST: 0000 0000 241.773209
  5 SAT: 1 N:  25 DT: 2001/01/01 00:00:06.571.117 ST: 0000 0000 331.739746
  6 SAT: 1 N:  25 DT: 2001/01/01 00:00:07.570.994 ST: 0000 0000  61.695221
  7 SAT: 1 N:  25 DT: 2001/01/01 00:00:08.570.994 ST: 0000 0000 151.661758
  8 SAT: 1 N:  25 DT: 2001/01/01 00:00:09.570.994 ST: 0000 0000 241.628296
  9 SAT: 1 N:  25 DT: 2001/01/01 00:00:10.570.994 ST: 0000 0000 331.594818
 10 SAT: 1 N:  25 DT: 2001/01/01 00:00:11.570.994 ST: 0000 0000  61.580631
 11 SAT: 1 N:  25 DT: 2001/01/01 00:00:12.570.876 ST: 0000 0000 151.536560
 12 SAT: 1 N:  25 DT: 2001/01/01 00:00:13.570.876 ST: 0000 0000 241.503098
 13 SAT: 1 N:  25 DT: 2001/01/01 00:00:14.570.876 ST: 0000 0000 331.469635
 14 SAT: 1 N:  25 DT: 2001/01/01 00:00:15.570.876 ST: 0000 0000  61.456112
 15 SAT: 1 N:  25 DT: 2001/01/01 00:00:16.570.876 ST: 4000 0000 151.422653
 16 SAT: 1 N:  25 DT: 2001/01/01 00:00:17.570.729 ST: 4700 0000 241.375961
 17 SAT: 1 N:  25 DT: 2001/01/01 00:00:18.570.729 ST: 4700 0000 331.342499
 18 SAT: 1 N:  25 DT: 2001/01/01 00:00:19.570.729 ST: 4700 0000  61.239017
 19 SAT: 1 N:  25 DT: 2001/01/01 00:00:20.570.729 ST: 0700 0000 151.205551
 20 SAT: 1 N:  25 DT: 2001/01/01 00:00:21.570.729 ST: 0000 0000 241.172089
 21 SAT: 1 N:  25 DT: 2001/01/01 00:00:22.570.729 ST: 0000 0000 331.138641
 22 SAT: 1 N:  25 DT: 2001/01/01 00:00:23.570.633 ST: 0000 0000  61.096535
 23 SAT: 1 N:  25 DT: 2001/01/01 00:00:24.570.633 ST: 0000 0000 151.063080

      . . . . .

 45 SAT: 1 N:  25 DT: 2001/01/01 00:00:46.570.153 ST: 4700 0000 330.272858
 46 SAT: 1 N:  25 DT: 2001/01/01 00:00:47.570.153 ST: 4700 0000  60.259342
 47 SAT: 1 N:  25 DT: 2001/01/01 00:00:48.570.036 ST: 0700 0000 150.215347
 48 SAT: 1 N:  25 DT: 2001/01/01 00:00:49.570.036 ST: 0000 0000 240.181885
 49 SAT: 1 N:  25 DT: 2001/01/01 00:00:50.570.036 ST: 0000 0000 330.148438
 50 SAT: 1 N:  25 DT: 2001/01/01 00:00:51.570.036 ST: 0000 0000  60.134914
 51 SAT: 1 N:  25 DT: 2001/01/01 00:00:52.570.036 ST: 0000 0000 150.101456
 52 SAT: 1 N:  25 DT: 2001/01/01 00:00:53.570.036 ST: 0000 0000 240.067993
 53 SAT: 1 N:  25 DT: 2001/01/01 00:00:54.569.916 ST: 0000 0000 330.023743
 54 SAT: 1 N:  25 DT: 2001/01/01 00:00:55.569.916 ST: 0000 0000  60.009529
 55 SAT: 1 N:  25 DT: 2001/01/01 00:00:56.569.916 ST: 0000 0000 149.976059
-----
Termine
-----
```

g) test de la lecture des fichiers des gains d'antenne

Il peut être nécessaire de tester que les gains des antennes soient lus correctement, et d'en examiner les valeurs. On les obtiens en lançant la procédure :

rcs_**t**est**g**ain

Cette procédure génère le fichier *testgain.out*, qui est de la forme suivante :

```
Cluster # (1-4) ?
1
-----
initcal:
/users/ida/robert/cluster/staff/dev/lib/gainantset

satellite: 1
-----
gainant:
Opening /users/ida/robert/cluster/staff/dev/lib/gainantset/C1.cal

BEGIN
# This is the calibration table of the X sensor
# 10 Hz filter
# This is the calibration table of the Y sensor
# 10 Hz filter
# This is the calibration table of the Z sensor
# 10 Hz filter
# This is the calibration table of the X sensor
# 180 Hz filter
# This is the calibration table of the Y sensor
# 180 Hz filter
# This is the calibration table of the Z sensor
# 180 Hz filter
END
-----
      NBR
      f      real(x)      imag(x)      real(y)      imag(y)      real(z)      imag(z)
0.2500 -0.002800    0.004819    -0.002770    0.004811    -0.002755    0.004813
0.5000 -0.001535    0.013034    -0.001495    0.012917    -0.001520    0.012949
0.7500  0.002393    0.020426    0.002297    0.020132    0.002469    0.020073
1.0000  0.008458    0.026144    0.008118    0.026119    0.008248    0.026079
1.2500  0.015663    0.030902    0.015204    0.030823    0.015427    0.030683
1.5000  0.023848    0.033809    0.023874    0.033806    0.023748    0.033661
1.7500  0.033046    0.034762    0.033171    0.034878    0.032732    0.035137
. . . . .
14.2500 0.026281   -0.000622    0.028093   -0.001736    0.028009   -0.001799
14.5000 0.023222   -0.001779    0.024240   -0.002098    0.024866   -0.002334
14.7500 0.021192   -0.003773    0.021515   -0.005489    0.021982   -0.005692
15.0000 0.019192   -0.005164    0.018910   -0.007587    0.019248   -0.007742
-----
      HBR
      f      real(x)      imag(x)      real(y)      imag(y)      real(z)      imag(z)
4.0000 0.022383    0.105005    0.022368    0.104773    0.022262    0.104845
8.0000 0.083301    0.184814    0.084745    0.184216    0.083201    0.182059
12.0000 0.166733    0.236670    0.168747    0.236119    0.166934    0.236828
16.0000 0.265506    0.261003    0.263660    0.260745    0.263291    0.258810
20.0000 0.371866    0.254518    0.369099    0.252966    0.370849    0.250639
. . . . .
224.0000 0.214725   -0.099950    0.207537   -0.103101    0.204829   -0.104571
228.0000 0.181328   -0.108010    0.174766   -0.109927    0.172181   -0.110957
232.0000 0.151558   -0.111356    0.145696   -0.112482    0.143362   -0.113004
236.0000 0.125555   -0.111163    0.120354   -0.111753    0.118319   -0.111851
240.0000 0.103063   -0.108571    0.098455   -0.108761    0.096715   -0.108558
```

h) calcul des composantes GSE d'un vecteur donné dans le GEI

Il peut être intéressant de pouvoir calculer dans le repère GSE un vecteur initialement donné dans le repère inertiel GEI, comme par exemple l'axe de spin qui est indiqué dans les fichiers *cowave.resu*, ou *cospectro.resu*, ou sur les fichiers PostScript *visuwave.ps* ou *visuspectro.ps*. Ceci est fait par la procédure ***tgeigse.bat***, lancée par la commande :

rcs_**t**geigse

Cette procédure nécessite le fichier *tgeigse.in*, qui est de la forme suivante :

```
31 01 2001      # JJ MM AN
05 54 00        # HH MM SS
102.50 -63.90   # Right ascension, declination in degrees
```

Les résultats sont dans le fichier *tgeigse.out*, dont un exemple est donné ci-dessous :

```
iday,imon,iyear ? ex: 14 01 2001
 31  1  2001
ih,im,is ?
 5  54  0
rasn,dec ?
102.5000  -63.9000

xgei,ygei,zgei=   -9.52203E-02   0.429511  -0.898028

ROCOTLIB/CDIPDI: dipole direction in GEO system is computed for year 1990

xgse,ygse,zgse=   -9.06401E-02   -4.70072E-02  -0.994774
```

III - LES PROCEDURES VECTOR_ROPROC:

1) Généralités

Les procédures **Vector_Roproc** sont destinées aux données de type « suite de vecteurs », dépendant du temps ou d'un autre paramètre, et pas nécessairement régulièrement échantillonnées comme c'est le cas des fichiers de type **Wave_Roproc** (voir § I-3). La procédure de base est **covector.bat**. Elle lit le fichier de données initial, extrait la période indiquée dans le fichier de paramètre **covector.in**, et produit un fichier ascii **covector.resu** au format général **Vector_Roproc**. Le fichier de paramètre **covector.in** est décrit en détail ci-après, ainsi qu'un exemple de fichier **covector.resu** montrant la généralité et la simplicité du format.

Ce format est utilisé pour les données de CLUSTER/FGM, d'une part parce qu'il est proche du format des données de base, et ensuite parce que la fréquence d'échantillonnage de ces données n'est pas assez régulière pour que la datation se fasse par bloc comme dans le cas des données de forme d'onde **Wave_Roproc**. Les fichiers de base peuvent être ceux des PPDB (Primary Parameters Data Base) ou les fichiers de données à haute résolution. La terminologie des noms de ces fichiers est la suivante :

bav1_011127.dat
bhr1_011127.dat

où *bav1* signifie B_average Spacecraft # 1, et *bhr1* signifie B_high_Resolution Spacecraft # 1 ; le champ *011127* correspond à la date des données du fichier (2001 Novembre 27). Un fichier ne peut couvrir au maximum qu'un seul jour de données.

Après obtention du fichier **covector.resu** au format **Vector_Roproc**, on peut appliquer des « filtres » sur ce fichier pour en transformer la nature, mais pas le format. Par exemple, on peut appliquer des changements de repères ; le repère initial étant le repère GSE (voir annexe 2), on peut transformer les données dans le repère GSM, ou encore le repère de variance minimum ou le repère SR2 pour des comparaisons avec STAFF-SC (voir paragraphe II-3).

On peut visualiser le contenu du fichier **covector.resu** par la procédure **visuvector.bat**.

A partir du fichier **covector.resu**, d'autres programmes de traitement sont applicable. Comme précédemment, chaque programme de calcul génère un fichier **.resu*, et dispose de son programme de visualisation qui lit ce fichier et génère un fichier de type **.ps*, que l'on peut visualiser, imprimer ou inclure dans un document ou publication, après transformation éventuelle, par un logiciel comme LATEX, Adobe Illustrator, Micrograph Designer ou Corel Draw.

2) Le programme de formatage covector

a) but

Comme on l'a vu, c'est le programme de départ, qui lit les fichiers de base (les *bav1_011127.dat* par exemple pour Cluster/FGM), extrait la période de données indiquées dans le *covector.in*, et les écrit dans le fichier *covector.resu* avec un format général multi-expérience décrit ci-après. Ce programme doit être écrit pour chaque expérience, selon le format des données de base.

b) fichier de paramètre covector.in

Ce fichier indique le chemin du fichier de données initiales, et la période de temps que l'on veut extraire. Un exemple est donné ci-dessous, ainsi que l'explication de chacun des paramètres.

```
/data1/cluster/FGM/PPD/0211/ASC/bav3_20021123.dat
23 11 2002      # iday, imon, iyear ? ex 21 1 2001
2 0 0          # ih1, im1, is1 ?
240.0           # duration in mn.
```

- *Chemin du fichier de données initial :*

Il peut être quelconque, mais pour CLUSTER, le nom du fichier doit respecter la nomenclature indiquée précédemment.

- *Date et heure :*

Pas de format particulier, mais toujours dans l'ordre jour-mois-an et heure-minute-seconde, soit dans cet exemple le 23 novembre 2002 à 2h 0m 0s.

- *Durée :* En minutes décimales.

c) lancement du programme

Une fois que les paramètres du fichier *covector.in* sont correctement fixés, la procédure *covector.bat* pour les données de CLUSTER/FGM (*Roproc_Cluster_Fgm* requête) est lancée par la commande :

rcf_cowave

qui vas générer dans le directory courant le compte rendu d'exécution *covector.out* et le fichier résultat *covector.resu* dont le format est décrit ci-après.

d) compte rendu d'exécution covector.out

Ce fichier *covector.out*, correspondant aux paramètres du *covector.in* précédent, est important car il contient les informations sur le bon ou mauvais déroulement de la procédure. En cas d'anomalies, comme par exemple ici où 2 points du fichier initial sont en overflow (égaux à $-1.e31$), celles-ci figurent en clair dans ce compte rendu.

e) description du fichier covector.resu

Ce fichier a donc un format général, multi-expérience ; son entête contient des informations globales, et dans le cas de données de type « VECTIME » (seul type actuellement reconnu), chaque vecteur est daté, et la date est au format ISO. Un exemple, correspondant au fichier *covector.in* précédent, est donné ci-dessous :

```
# -----
# File type covector.resu, result of covector.exe program
# File type used by Vector_Roproc Library
# Version 1.0, P. Robert, CETP, Septembre 2002
# -----
# BEGIN CONSTANT
# INT : DATA_SET_NUMBER      :      19
# CHA : MISSION_NAME          : CLUSTER
# CHA : INSTRUMENT_NAME       : FGM
# CHA : SPACECRAFT_NAME       : Samba
# INT : SPACECRAFT_NUMBER     :      3
# CHA : MAIN_TITLE            : CLUSTER / FGM / Samba (#3)
# CHA : TARGET_NAME           : Magnetospheric Study
# CHA : DATA_TYPE            : VECTIME
# CHA : NATURE                : B DC magnetic field, PPD
# CHA : UNIT                  : nT
# CHA : COORDINATE_SYSTEM     : GSE
# INT : NB_COMPONENTS         :      3
# INT : NB_VECTORS            :    3580
# CHA : FORMAT                : (a24,3e15.6)
# CHA : CREATION_DATE         : 2003-09-08T16:02:42.000Z
# END CONSTANT
# -----
# BEGIN COMMENTS
# Cluster is a set of 4 Spacecraft, launched in summer 2000.
# The Flux Gate Magnetometer experiment (FGM) measure the 3 components
# of the DC magnetic field.
# The time resolution is varying: the low time resolution, also called
# Prime Parameter Data (PPD) is of the order of the spin period (4 s),
# while the high resolution is around 0.04 s.
# END COMMENTS
# -----
# BEGIN DATA_DESCRIPTION
# This vector data file contains VECTIME data type, and therefore
# is structured as a temporal serie of data bloc.
# Each bloc begin with the date-time epoch in ISO format.
# The rest of the bloc data is a serie of NB_COMPONENTS values,
# corresponding to the number of component of the time data vector.
# END DATA_DESCRIPTION
# -----
# BEGIN DATA
2002-11-23T02:00:03.698Z  -0.307890E+02   0.372240E+02  -0.916300E+02
2002-11-23T02:00:07.719Z  -0.306500E+02   0.371760E+02  -0.916510E+02
2002-11-23T02:00:11.739Z  -0.304530E+02   0.370470E+02  -0.917070E+02
2002-11-23T02:00:15.760Z  -0.305530E+02   0.370410E+02  -0.917620E+02
2002-11-23T02:00:19.781Z  -0.305380E+02   0.370450E+02  -0.918200E+02
2002-11-23T02:00:23.801Z  -0.303880E+02   0.370290E+02  -0.918450E+02
2002-11-23T02:00:27.822Z  -0.302390E+02   0.369340E+02  -0.918790E+02

*****
etc...
*****

2002-11-23T05:59:25.372Z   0.727110E+02  -0.271360E+03  -0.156350E+03
2002-11-23T05:59:29.393Z   0.728870E+02  -0.271440E+03  -0.156250E+03
2002-11-23T05:59:33.414Z   0.728980E+02  -0.271630E+03  -0.156120E+03
2002-11-23T05:59:37.434Z   0.730290E+02  -0.271770E+03  -0.156080E+03
2002-11-23T05:59:41.455Z   0.727940E+02  -0.271890E+03  -0.156080E+03
2002-11-23T05:59:45.476Z   0.726210E+02  -0.271990E+03  -0.156080E+03
2002-11-23T05:59:49.496Z   0.724790E+02  -0.272110E+03  -0.156030E+03
2002-11-23T05:59:53.517Z   0.722060E+02  -0.272400E+03  -0.155750E+03
2002-11-23T05:59:57.538Z   0.720220E+02  -0.272680E+03  -0.155470E+03
2002-11-23T06:00:01.558Z   0.719730E+02  -0.272910E+03  -0.155350E+03
# END DATA
```

f) visualisation du contenu du fichier *covector.resu*

Le contenu du fichier *covector.resu* peut être visualisable par la procédure ***visuvector.bat***. Celle-ci est lancée par la commande :

r_{cs}_visuvector

Cette procédure produit un fichier PostScript de nom *visuvector.ps*. Un exemple est donné au § V-6. On a la possibilité de fixer les bornes des 3 axes *xyz* du graphe au moyen d'un fichier optionnel, le *visuvector.in*. Si ce fichier n'existe pas, il est créé par défaut avec les valeurs ci-dessous. S'il existe avant de lancer la commande **r_{cs}_visuvector**, ses valeurs sont prises en compte pour la visualisation. Les valeurs (0.,0.) entraînent le calcul automatique des bornes.

Le contenu par défaut du fichier optionnel *visuvector.in* est le suivant :

```
0. 0. 0. 0. 0. 0. # x1,x2, y1,y2, z1,z2 for figure limits
```

3) Les filtres pour les changements de repère

a) introduction

Comme pour les procédures ***Wave_Roproc***, ces filtres ne modifient pas le format des fichiers *covector.resu*, mais effectuent simplement des traitements, notamment des changements de repère. Dans la mesure où les changements de repère nécessitent 3 composantes, ces filtres n'opèrent que sur des vecteurs de dimension 3, et donnés initialement dans un repère connu, comme le repère GSE par exemple (cas de CLUSTER/FGM).

Ces procédures font d'abord une copie du fichier *covector.resu* initial dans le fichier *covector.resu.old*, puis génèrent un nouveau fichier *covector.resu* qui contient les données transformées.

b) vector_to_GSM

La transformation d'un fichier *covector.resu* de données CLUSTER/FGM (initialement en GSE) qui les met dans le repère GSM se fait par la commande :

r_{cf}_vector_to_gsm

c) vector_to_SR2

La transformation d'un fichier *covector.resu* de données CLUSTER/FGM (initialement en GSE) qui les met dans le repère SR2 se fait par la commande :

r_{cf}_vector_to_SR2

Cette commande nécessite un fichier de paramètres *vector_to_sr2.in* de la forme suivante :

102.50 -63.90 # Right ascension, declination in degrees
--

où on donne la direction de l'axe de spin du satellite dans le repère inertiel GEI, sous la forme ascension droite et déclinaison, en degrés (voir annexe 2).

d) **vector_to_MVA**

La transformation d'un fichier *covector.resu* de données (initialement dans un repère quelconque) qui les met dans le repère MVA (Minimum Variance Analysis) se fait par la commande :

rcf_vector_to_mva

Les directions des axes du repère MVA, calculées dans le repère de départ, ainsi que les valeurs propres sont données dans le fichier *vector_to_mva.out*.

4) **Les traitements à 4 satellites**

a) **production d'un jeu de 4 fichiers covector.resu**

Pour extraire et formater les données CLUSTER/FGM des 4 satellites, on pourrait lancer 4 fois successives la procédure *covector.bat*, en changeant à chaque fois le numéro de satellite dans le fichier *covector.in*. Mais d'abord cette manière de procéder est fastidieuse, et de plus les fichiers *covector.resu* seraient écrasés à chaque lancement de la procédure *covector.bat*. Pour remédier à cela, la procédure *covector4.bat* permet en une seule opération de générer les 4 fichiers *covector.resu*, qui sont rangés dans un directory dont le nom est demandé lors du lancement de la procédure. Celle-ci est lancée par la commande :

rcf_covector4

Si le directory indiqué n'existe pas, il est créé ; si il existe déjà, il est purgé de ses fichiers. Cette procédure ne fonctionne que si les noms de fichiers du *covector.in* respectent la terminologie indiquée précédemment (tel que *bav1_011127.dat*). Après exécution de la procédure, le directory résultat contient les fichiers suivants :

covector_1.in	covector_1.out	covector_1.resu
covector_2.in	covector_2.out	covector_2.resu
covector_3.in	covector_3.out	covector_3.resu
covector_4.in	covector_4.out	covector_4.resu

b) visualisation séparée des 4 fichiers

Les 4 fichiers *covector_*.resu* étant disponibles, on peut alors générer les 4 fichiers PostScript correspondants de type *visuvector_*.ps* par la procédure ***visuvector4.bat***, lancé par la commande :

rcf_visuvector4

Le directory résultat sera alors complété par les fichiers suivants :

```
visuvector_1.ps  
visuvector_2.ps  
visuvector_3.ps  
visuvector_4.ps
```

c) visualisation commune des 4 fichiers

Les 4 fichiers *covector_*.resu* étant disponibles, on peut également alors générer un seul fichier *visuvector_4sat.ps*, contenant en superposition le tracé des données des 4 satellites, par la procédure ***visuvector_4sat.bat***, lancé par la commande :

rcf_visuvector_4sat

Ce fichier *visuvector_4sat.ps* est généré dans le directory courant, mais une copie est également faite dans le directory résultat.

5) Les procédures particulières à CLUSTER/FGM

a) introduction

Ces procédures ont pour but le calcul de la densité moyenne de courant $\langle J \rangle$ à travers la surface S_i vu par les 4 satellites au moyen du calcul du rotationnel de B , et en utilisant le théorème d'Ampère :

$$\vec{\nabla} \times \vec{B} = \oint \vec{B} \cdot d\vec{l} = \mu_0 \int \vec{J} \cdot d\vec{s} = \mu_0 \vec{I} = \mu_0 \langle J \rangle S_i$$

Pour aboutir à ce calcul, il faut successivement :

- aligner en temps les données du magnétomètre, c'est à dire interpoler 3 des 4 fichiers de données pour avoir une base de temps commune ;
- ajouter à ces fichiers les positions de chaque satellite ;
- calculer ensuite le courant à partir des valeurs des champs et des positions, en faisant l'hypothèse de linéarité du champ magnétique le long de chaque arête du tétraèdre.

Les procédures suivantes effectuent ces opérations.

b) alignement en temps de 4 fichiers covector.resu

Après exécution de la procédure *rcf_covector4*, le directory résultat contient les fichiers suivants :

covector_1.in	covector_1.out	covector_1.resu
covector_2.in	covector_2.out	covector_2.resu
covector_3.in	covector_3.out	covector_3.resu
covector_4.in	covector_4.out	covector_4.resu

L'ensemble des 4 fichiers *covector_*.resu* sera aligné en temps par la procédure *alivectime.bat*, lancé par la commande :

rcf_alivectime

Cette commande demande une valeur maximum pour l'interpolation entre 2 points consécutifs, par exemple 6 s. pour les PPD (échantillonnées à environ 4 s.), et 6 ms pour les hautes résolutions (échantillonnées à environ 23 Hz). Il est conseillé de regarder systématiquement le rapport d'exécution *alivectime.out*, afin de vérifier que le re-échantillonnage des données s'est fait correctement ; en effet, il suffit d'un point manquant ou d'un point faux pour que la procédure ne s'exécute pas, et s'arrête avec un rapport d'erreur.

Si la procédure s'est déroulée correctement, le temps de chaque vecteur des 4 fichiers *covector_*.resu* sera alors identique au temps du satellite 1 pris comme référence de temps.

c) ajout des positions des satellites

L'ensemble des 4 fichiers *covector_*.resu* sera complété par la position de chaque satellite par la procédure *addvecpos4.bat*, lancé par la commande :

rcf_addvecpos4

Chaque vecteur des 4 fichiers *covector_*.resu* aura alors 6 composantes, 3 pour le champ magnétique mesuré et 3 pour la position du satellite. Les positions sont calculées dans le même repère que le champ magnétique du fichier *covector.resu* initial ; pour l'instant, les repères possibles sont limités au GSE et au GSM. Comme toujours, un rapport d'exécution *addvecpos_.out* est généré pour chaque satellite. Cette commande existe aussi en version individuelle qui travaille sur un seul fichier *covector.resu*:

rcf_addvecpos

d) calcul de la densité de courant par la méthode du rotationnel

Une fois que l'on dispose d'un jeu de 4 fichiers *covector_*.resu* à 6 composantes (3 magnétiques et 3 de position), et que ces fichiers sont alignés en temps, on peut calculer le courant moyen $\langle J \rangle$ par la méthode du rotationnel. En pratique, les opérations préliminaires d'alignement en temps et de production d'un jeu de données à 6 composantes est obtenu par l'enchaînement des commandes suivantes :

rcf_covector4	→ <i>covector_*.resu</i> ; données non alignées pour les 4 sat.
rcf_alivectime	→ <i>covector_*.resu</i> ; données alignées en temps
rcf_addvecpos4	→ ajoute les positions aux 4 fichiers alignés <i>covector_*.resu</i>

Le calcul du rotationnel, et donc de la densité moyenne $\langle J \rangle$, de la divergence de $\mathbf{B}$, du champ $\mathbf{B}_{\text{moyen}}$, etc. sont fait par la méthode des coordonnées barycentriques [4], mise en œuvre par la procédure *cocurldiv.bat*, lancé par la commande :

rcf_cocurldiv

Cette commande produit les fichiers suivants :

<i>bmoy.resu</i>	contenant les valeurs du champ magnétique moyen (format <i>Vector_Roproc</i>)
<i>current.resu</i>	contenant les valeurs du $\langle J \rangle$ calculé (format <i>Vector_Roproc</i>)
<i>cocurldiv.resu</i>	contenant les valeurs de $\mathbf{B}$, de $\langle J \rangle$, de $\text{div}\mathbf{B}$, de $\mathbf{J}_{\parallel}$, $\mathbf{J}_{\perp}$, etc. (format peu explicite, utilisé seulement par le <i>visucocurldiv.bat</i>)
<i>cocurldiv.out</i>	contenant le rapport d'exécution de la procédure.

e) visualisation des résultats

Les fichiers *bmoy.resu* et *current.resu* étant au format *Vector_Roproc*, ils peuvent être visualisés séparément par la procédure standard *visuvector.bat*, après copie de ces fichiers dans un fichier de nom générique *covector.resu*. Pour obtenir les visualisations de toutes les informations du fichier *cocurldiv.resu*, on utilise la procédure *visucocurldiv.bat*, lancé par la commande :

rcf_visucocurldiv

qui produit les fichiers suivants :

<i>visumag.ps</i>	visu des 4 champs $\mathbf{B}$
<i>visubmoy.ps</i>	$\mathbf{B}_{\text{moyen}}$, xyz même échelle sur même figure.
<i>visupos.ps</i>	visu des positions des 4 sat., en cartésien, avec E et P
<i>visucurl.ps</i>	résultat principal : visu de $\mathbf{J}$, $\text{div}\mathbf{B}$, $\mathbf{J}_{\parallel}$, $\mathbf{J}_{\perp}$
<i>visunorma.ps</i>	$\mathbf{N} = \mathbf{J} \times \mathbf{B}_{\text{moyen}}$, xyz même échelle sur même fig.

f) résultats en repère MVA

Les résultats du calcul de J et autres paramètres contenus dans le fichier *cocurldiv.resu* peuvent être tous transcrits dans le repère de variance minimum MVA du champ $\mathbf{B}$ moyen par la procédure *covarmin.bat*, lancé par la commande :

rcf_covarmin

qui, à partir du seul fichier *cocurldiv.resu*, effectue tous les calculs nécessaires. Cette procédure produit un fichier *covarmin.resu*, dont le format est semblable à celui du fichier *cocurldiv.resu*, ainsi qu'un rapport d'exécution *covarmin.out*. Cette procédure doit être lancée dans un sous-directory de celui qui contient le fichier *cocurldiv.resu*, car elle va le lire dans le directory père. Comme pour ce dernier, on peut obtenir les visualisations de toutes les informations du fichier *covarmin.resu*, en utilisant la procédure *visuvarmin.bat*, lancé par la commande :

rcf_visuvarmin

qui produit les fichiers suivants, en repère MVA :

<i>visumag.ps</i>	visu des 4 champs B
<i>visubmoy.ps</i>	B _{moyen} , xyz même échelle sur même figure.
<i>visupos.ps</i>	visu des positions des 4 sat., en cartésien, avec E et P
<i>visucurl.ps</i>	résultat principal : visu de J , divB , $\nabla \times \mathbf{J}$, $\mathbf{J} \perp$
<i>visunorma.ps</i>	N = J × B _{moyen} , xyz même échelle sur même fig.

Ces fichiers ont les mêmes noms que ceux créés par *rcf_visucocurldiv*, mais comme *rcf_covarmin* doit être lancé dans un sous-directory, ces fichiers sont créés dans ce même sous-directory.

g) macro_commande cocurldiv_total

L'ensemble des procédures précédentes peuvent être enchaînées automatiquement par la procédure *cocurldiv_total.bat*, lancé par la commande :

rcf_cocurldiv_total

qui, à partir du seul fichier *covector.in*, lance successivement des commandes *rco* et *rcf* appropriées, en recréant au besoin les autres fichiers *.in nécessaires, et produit les visualisations utiles en re-nommant éventuellement les fichiers PostScript créés :

rcf_covector4	
rcf_visuvector_4sat	→ <i>visuvector_ini.ps</i> ; données non alignées
rcf_alivectime	
rcf_visuvector_4sat	→ <i>visuvector.ali.ps</i> ; données alignées en temps
rco_cresatpos	
rco_visusatpos	→ <i>visusatpos_day.ps</i> ; trajectoires en 3D pour la journée
rco_cresatpos	
rco_visusatpos	→ <i>visusatpos.ps</i> ; trajectoires en 3D pour la période étudiée
rcf_addvecpos4	→ ajoute les positions aux 4 fichiers alignés <i>covector_*.resu</i>
rcf_cocurldiv	→ <i>bmoy.resu</i> ; B _{moyen} , format covector.resu → <i>current.resu</i> ; J _{moyen} , format covector.resu → <i>cocurldiv.resu</i> ; valeurs de <J>, de divB , de $\nabla \times \mathbf{J}$, $\mathbf{J} \perp$, etc.
rcf_visucocurldiv	→ <i>visumag.ps</i> ; obsolete, remplacé par le visuvector.ali.ps → <i>visupos.ps</i> ; positions des 4 sat., en cartésien, avec E et P → <i>visucurl.ps</i> ; résultat principal → <i>visubmoy.ps</i> ; B _{moyen} , xyz même échelle sur même fig. → <i>visunorma.ps</i> ; N = J × B _{moyen} , xyz même échelle sur même fig.
rcf_visuvector	→ <i>visuvec_bmoy.ps</i> ; B _{moyen} , plot standard Roproc
rcf_visuvector	→ <i>visuvec_curr.ps</i> ; J _{moyen} , plot standard Roproc
rcf_vector_to_mva	
rcf_visuvector	→ <i>visuvec_bmoy_mva.ps</i> ; B _{moyen} dans son MVA, plot standard

Cette macro-commande effectue aussi le calcul et la visualisation de ces grandeurs dans le repère de variance minimum MVA du champ magnétique moyen, en lançant successivement les commandes suivantes, dans un sous-directory VARMIN créé vide :

rcf_covarmin	→ <i>covarmin.resu</i> ; valeurs de $\langle J \rangle$, de $\text{div}B$, de $I_{\parallel}$, $J_{\perp}$, etc.
rcf_visucovarmin	→ <i>visumag.ps</i> ; visu des 4 champs B
	→ <i>visupos.ps</i> ; positions des 4 sat., en cartésien, avec E et P
	→ <i>visucurl.ps</i> ; résultat principal
	→ <i>visubmoy.ps</i> ; B_{moyen} , xyz même échelle sur même fig.
	→ <i>visunorma.ps</i> ; $N = J \times B_{\text{moyen}}$, xyz même échelle sur même fig.

L'intérêt de cette commande est évident : on obtient en une seule commande toutes les visualisations utiles pour l'étude d'un phénomène. C'est aussi un bon exemple de réalisation de shell-script, ou macro-commande, mentionnée au § I-4. Néanmoins, du fait qu'elle enchaîne un nombre important de procédures, chacune d'entre elle pouvant générer une erreur, le déroulement de chaque étape doit être surveillé, notamment la procédure de re-échantillonnage **rcf_alivectime** sensible au moindre trou de données.

h) macro_commande cocurldiv_lprt

La procédure **cocurldiv_lprt.bat** permet en une seule commande d'imprimer tous les fichiers produits par la macro-commande **rcf_cocurldiv_total**. Elle est lancée par la commande :

rcf_cocurldiv_lprt « *nom_imprimante* » « *nom_directory_resultat* »

où « *nom_imprimante* » est le nom d'une imprimante (couleur) connue de la machine utilisée, et « *nom_directory_resultat* » est le nom du directory où ont été produits les différents PostScripts, par exemple sur **sawara** :

rcf_cocurldiv_lprt **vtk1-b323 exemple**

Cette commande imprime dans l'ordre les fichiers suivants :

```
visuvector_ini.ps
visuvector_ali.ps
visusatpos_day.ps
visusatpos.ps
visupos.ps
visucurl.ps
visubmoy.ps
visunorma.ps

visuvec_bmoy.ps
visuvec_curr.ps
visuvec_bmoy_mva.ps
visuvec_curr_mva.ps

VARMIN/covarmin.out.ps
```

```
VARMIN/visumag.ps  
VARMIN/visupos.ps  
VARMIN/visucurl.ps  
VARMIN/visubmoy.ps  
VARMIN/visunorma.ps
```

L'association des 2 macros commandes ***rcf_cocurldiv_total*** et ***rcf_cocurldiv_lprt*** permet d'obtenir en très peu de manipulation tous les calculs et graphes utiles pour l'étude d'un courant ; Néanmoins, vu la production et l'impression des 18 fichiers PostScripts par *run*, ces macros-commandes doivent être utilisées avec précaution.

6) Conversion de formats

a) Vector_to_Wave

A condition que le fichier *covector.resu* soit du type « VECTIME », et que les données soient équidistantes en temps, on peut alors définir une fréquence d'échantillonnage constante pour ces données, et transformer ce fichier *covector.resu* au format forme d'onde *cowave.resu*. Cette transformation s'obtient par la procédure ***vector_to_wave.bat***, lancé par la commande :

rcf_vector_to_wave

Cette procédure nécessite le fichier *vector_to_wave.in*, de la forme :

512	# nbp of a cowave.resu windows
0.25	# spin frequency
102.50 -63.90	# Right ascension, declination of spin axis in GEI (degres)

où on indique le nombre de points désiré pour le découpage en séquences (voir § II-2-b), la fréquence de spin et la direction de l'axe de spin dans le repère inertiel GEI, afin d'avoir la compatibilité avec les fichiers *cowave.resu* directement issu des procédures ***cowave.bat***.

Il faut noter que la fréquence d'échantillonnage est directement calculé à partir des dates du fichier *covector.resu*, qui doivent donc être régulièrement espacées, sans discontinuités. La valeur trouvée est indiquée dans le rapport d'exécution *vector_to_wave.out*.

L'intérêt pour CLUSTER est immédiat : à partir des données du magnétomètre FGM, extraites, vérifiées et mise au format *covector.resu* par la commande ***rcf_covector***, on peut créer des fichiers FGM au format *cowave.resu*, et donc utiliser les commandes ***Wave_Roproc*** pour calculer et visualiser des spectrogrammes, spectres moyens et puissances intégrées des données FGM, au moyens des commandes propres à CLUSTER/STAFF-SC telles que:

```
rct_spectro  
rct_spectro4  
  
rct_visuspectro  
rct_visuspectro4
```

```
rcs_visuspectra  
rcs_visuspectra4  
  
rcs_visupower  
rcs_visupower4
```

b) Merge_Alivector

La procédure ***merge_alivector.bat*** permet de fusionner 4 fichiers *covector_*.resu* en 1 seul, c'est à dire de prendre 4 fichiers *covector_*.resu* à 3 composantes [x,y,z] chacun, et de créer 1 seul fichier *covector.resu* à 12 composantes (4 x [x,y,z]). Il faut que le temps sur les 4 fichiers *covector_*.resu* soit le même, et donc que ces fichiers soient issus de la procédure ***alivetime.bat***.

Cette procédure est lancée par la commande :

```
rcf_merge_alivector
```

qui demande le nom (ou le chemin) du directory contenant les 4 fichiers *covector_*.resu* alignés en temps.

7) Utilitaires divers

a) copie dans le répertoire courant d'un jeu d'exemple de fichier *.in

Pour utiliser les procédures ***Vector_Roproc***, il faut toujours indiquer les paramètres d'entrée dans des fichiers **.in*, comme *covector.in*, *vector_to_sr2.in*, *visuvector.in*, etc.

Pour avoir des exemple sous la main, la commande suivante permet de recopier dans le directory courant un exemple de tous les fichier **.in* :

```
rcf_copy_in
```

Après exécution, le directory courant contient les fichiers :

```
covector.in      visuvector.in      vector_to_sr2.in      vector_to_wave.in
```

b) fichier readme.txt donnant la procédure d'installation de la chaîne Vector_Roproc

Si on souhaite installer les procédures ***Vector_Roproc*** sur une autre machine SUN/Solaris, le mode d'emploi est donné par la commande :

```
rcf_readme
```

qui liste le fichier README.txt du package (il faut bien sûr l'avoir déjà installé ailleurs ; sinon, on peut aller voir directement ce fichier au début de l'arborescence du package).

IV - LES PROCEDURES ORBIT_ROPROC:

1) Généralités

Les procédures **Orbit_Roproc** sont dédiées aux calcul et visualisations des positions et trajectoires des satellites. Elles ont été initialement développées pour CLUSTER, et sont en cours de normalisation pour leur utilisation à d'autres satellites. Elles sont pour l'instant limitées à CLUSTER, et seulement de 2 type : la procédure **listorbite.bat** et **cresatpos.bat**, décrites ci-dessous, avec leur programme de visualisation respectifs. Le format des fichier *.resu pourrait être modifié prochainement, et passer notamment au format **Vector_Roproc**, plus général. Le préfixe de ces procédures est **rco** (**Roproc_Cluster_Orbite**).

2) Création de fichiers ascii des positions

a) liste des positions et de la géometrie du tétraèdre pour une journée: listorbit

On peut créer un fichier **listorbit.resu** des données des positions des 4 satellites de CLUSTER au cours d'une journée par la procédure **listorbite.bat**, lancée par la commande :

rco_listorbit

Cette commande nécessite un fichier de paramètres **listorbit.in** de la forme suivante :

```
/data1/cluster/ORBIT/DATA
27 08 2002 # day-month-year
15.0      # delta T in mn.
gse       # gei or gse
l         # s or l (short or long)
```

où on donne le chemin du fichier d'orbite, dont le nom est du type **020827_sat1.orb** pour le 27 août 2002, la date du jour à traiter, qui doit être cohérente avec le nom du fichier, et le temps entre chaque point en minutes. Une journée entière sera traitée, le nombre de points étant alors $24 \times 60 / dT$.

On a également le choix du repère (repère inertiel GEI ou géocentrique GSE, voir annexe 2), indiqué en minuscule. On peut obtenir une version « courte » des résultats, où seules sont données les positions des 4 satellites, ou une version « longue », où la géométrie du tétraèdre est également calculée. Des exemples de fichier résultats, **listorbit.resu**, en version courte et longue, sont donnés ci-dessous. Compte tenu de la brièveté du rapport d'exécution, le fichier **listorbite.out** n'est pas créé, mais simplement displayé sur le standart output (l'écran).

Exemple de fichier *listorbit.resu* en version « short » :

#START_HEADER															
File type : listorbit.resu															
Creator : listorbit.exe															
Version : April 2001															
Author : P. Robert, CNRS/CETP															
Coordinates: gse															
Form : short															
Content : see following line															
JD	YY	MM	DD	hh	mm	ss	s/c	X (km)	Y	Z	Vx	Vy	Vz	Rev	E

#END_HEADER															
949	2002	08	07	00	00	00	1	-107116.60	-54096.62	-5832.57	-0.18	0.41	-1.07	325.49	
							2	-109829.84	-53364.94	-4299.41	-0.16	0.33	-1.05	325.49	
							3	-109960.59	-53833.25	-7627.68	-0.20	0.35	-1.02	325.47	
							4	-110583.89	-56541.77	-4589.18	-0.20	0.35	-0.98	325.48	
949	2002	08	07	00	15	00	1	-107278.06	-53703.15	-6792.80	-0.16	0.42	-1.07	325.50	
							2	-109973.49	-53045.30	-5241.43	-0.14	0.34	-1.05	325.49	
							3	-110138.73	-53496.79	-8546.75	-0.18	0.36	-1.02	325.48	
							4	-110764.90	-56200.17	-5468.28	-0.18	0.36	-0.98	325.49	
949	2002	08	07	00	30	00	1	-107419.43	-53299.69	-7751.77	-0.14	0.43	-1.06	325.50	
							2	-110097.58	-52716.30	-6182.53	-0.12	0.35	-1.05	325.50	
							3	-110297.54	-53150.96	-9464.33	-0.16	0.37	-1.02	325.48	
							4	-110927.19	-55849.12	-6346.46	-0.16	0.37	-0.98	325.49	

etc.															

949	2002	08	07	23	30	00	1	-19173.78	14592.10	-51716.00	2.61	0.82	1.09	325.90	
							2	-23372.28	9324.92	-52968.08	2.58	0.88	0.92	325.90	
							3	-30598.19	9081.36	-55626.88	2.41	0.88	0.73	325.89	
							4	-30975.06	8527.28	-52086.88	2.51	0.94	0.74	325.90	
949	2002	08	07	23	45	00	1	-16804.35	15320.13	-50691.12	2.65	0.79	1.19	325.91	
							2	-21028.52	10116.28	-52094.09	2.62	0.87	1.02	325.90	
							3	-28407.55	9876.53	-54932.65	2.45	0.87	0.81	325.89	
							4	-28691.22	9369.87	-51386.17	2.56	0.92	0.82	325.90	

Exemple de fichier *listorbit.resu* en version « long » :

#START_HEADER															
File type : listorbit.resu															
Creator : listorbit.exe															
Version : April 2001															
Author : P. Robert, CNRS/CETP															
Coordinates: gse															
Form : long															
Content : see following line															
JD	YY	MM	DD	hh	mm	ss	s/c	X (km)	Y	Z	Vx	Vy	Vz	Rev	E

#END_HEADER															
949	2002	08	07	00	00	00	1	-107116.60	-54096.62	-5832.57	-0.18	0.41	-1.07	325.49	
							2	-109829.84	-53364.94	-4299.41	-0.16	0.33	-1.05	325.49	
							3	-109960.59	-53833.25	-7627.68	-0.20	0.35	-1.02	325.47	
							4	-110583.89	-56541.77	-4589.18	-0.20	0.35	-0.98	325.48	
							1	2256.13	362.53	-245.36					
							2	-457.10	1094.21	1287.80					
							3	-587.85	625.89	-2040.47					
							4	-1211.16	-2082.62	998.03					
								3201.18	3373.43	4421.19	3363.60	3277.93		4117.90	
								0.658891E+07	0.523417E+07	0.474405E+07		0.536326E+07			
								4576.60	3418.38	2732.13					
								0.25307	121.93	43.76					
								0.20075	110.51	-59.72					

```

949 2002 08 07 00 15 00 1 -107278.06 -53703.15 -6792.80 -0.16 0.42 -1.07 325.50
2 -109973.49 -53045.30 -5241.43 -0.14 0.34 -1.05 325.49
3 -110138.73 -53496.79 -8546.75 -0.18 0.36 -1.02 325.48
4 -110764.90 -56200.17 -5468.28 -0.18 0.36 -0.98 325.49
1 2260.74 408.20 -280.49
2 -434.69 1066.05 1270.88
3 -599.93 614.57 -2034.43
4 -1226.09 -2088.82 1044.04
3178.81 3361.90 4488.60 3340.10 3260.52 4144.56
0.664808E+07 0.518020E+07 0.468877E+07 0.532187E+07
4644.73 3404.95 2663.17
0.26692 122.01 43.81
0.21785 110.52 -59.72
949 2002 08 07 00 30 00 1 -107419.43 -53299.69 -7751.77 -0.14 0.43 -1.06 325.50
2 -110097.58 -52716.30 -6182.53 -0.12 0.35 -1.05 325.50
3 -110297.54 -53150.96 -9464.33 -0.16 0.37 -1.02 325.48
4 -110927.19 -55849.12 -6346.46 -0.16 0.37 -0.98 325.49
1 2266.00 454.33 -315.50
2 -412.15 1037.72 1253.74
3 -612.11 603.05 -2028.06
4 -1241.76 -2095.11 1089.81
3158.38 3352.39 4558.38 3316.49 3244.95 4171.04
0.670986E+07 0.512393E+07 0.463742E+07 0.528088E+07
4714.30 3392.27 2593.73
0.28043 122.09 43.85
0.23540 110.53 -59.73

*****
etc.
*****

949 2002 08 07 23 45 00 1 -16804.35 15320.13 -50691.12 2.65 0.79 1.19 325.91
2 -21028.52 10116.28 -52094.09 2.62 0.87 1.02 325.90
3 -28407.55 9876.53 -54932.65 2.45 0.87 0.81 325.89
4 -28691.22 9369.87 -51386.17 2.56 0.92 0.82 325.90
1 6928.56 4149.42 1584.89
2 2704.39 -1054.42 181.92
3 -4674.64 -1294.17 -2656.64
4 -4958.31 -1800.83 889.84
6847.77 13500.28 13311.14 7909.81 7731.45 3593.71
0.238376E+08 0.197468E+08 0.200550E+08 0.136597E+08
15693.27 3588.40 3411.77
0.77134 79.38 23.07
0.04922 96.70 -65.67

```

b) visualisation 2D des positions et de la géométrie du tétraèdre

Le contenu du fichier *listorbit.resu*, en version « short » ou « long », peu être visualisé par la procédure *visuorbit.bat*, lancé par la commande :

rco_visuorbit

La version courte génère un fichier *visuorbit_020807_posit.ps*, qui visualise les distances et vitesses de chaque satellite au cours du temps, tandis que la version longue génère ce fichier, plus le fichier *visuorbit_020807_tetra.ps* qui visualise la géométrie du tétraèdre (inter-distances, paramètres d'élongation et de planarité, etc.). Des exemples de ces fichiers sont donnés au § V-4.

c) positions seules pour une durée quelconque: *cresatpos*

On peut créer un fichier *cresatpos.resu* des données des positions des 4 satellites de CLUSTER au cours d'une journée par la procédure *cresatpos.bat*, lancée par la commande :

rco_cresatpos

Cette commande nécessite un fichier de paramètres *cresatpos.in* de la forme suivante :

```
/data1/cluster/ORBIT/DATA
26 8 2002      # day-month-year
00 00 00      # hh-mm-ss
3300.0        # time period in mn.
5.0           # time resolution in mn.
gse           # gei or gse
```

où on donne le chemin du fichier d'orbite, dont le nom est du type 020826_sat1.orb pour le 26 août 2002, la date du jour à traiter (jour, mois, an), qui doit être cohérente avec le nom du fichier, le temps de début (en heures, minutes, secondes), la durée totale de la période et le temps entre chaque point (en minutes), et enfin le repère souhaité (GEI ou GSE, en minuscule).

C'est la seule **Cluster_Roproc** qui peut travailler sur des durées supérieures à 24 heures ; on peut en effet calculer les positions sur une orbite complète (2.5 jours environ), voire davantage.

Un exemple du fichier *cresatpos.resu* est donné ci-dessous :

gse														
391	2001	01	26	00	00	00	1	-11392.55	-16723.99	-18661.31	-2.99	-0.13	3.84	90.98
							2	-10739.79	-16545.55	-19565.88	-3.04	-0.18	3.76	90.98
							3	-10729.13	-16627.35	-19192.11	-3.03	-0.18	3.80	90.98
							4	-10569.62	-16795.00	-19752.95	-3.02	-0.21	3.74	90.98
391	2001	01	26	00	05	00	1	-12279.43	-16746.67	-17494.08	-2.92	-0.03	3.94	90.98
							2	-11642.47	-16584.58	-18421.38	-2.97	-0.08	3.87	90.98
							3	-11628.75	-16666.91	-18036.35	-2.96	-0.09	3.91	90.98
							4	-11468.07	-16842.14	-18614.21	-2.96	-0.11	3.85	90.98
391	2001	01	26	00	10	00	1	-13144.28	-16739.26	-16295.51	-2.84	0.07	4.05	90.99
							2	-12524.79	-16594.56	-17244.73	-2.90	0.01	3.97	90.98
							3	-12507.54	-16676.58	-16848.35	-2.89	0.02	4.01	90.98
							4	-12346.92	-16860.46	-17443.71	-2.89	-0.01	3.95	90.98

etc.														

393	2001	01	28	06	50	00	1	14932.47	-9548.32	-39241.41	-3.12	-1.26	1.37	91.94
							2	15653.98	-9304.06	-39605.92	-3.11	-1.25	1.33	91.94
							3	15629.58	-9198.12	-39431.43	-3.11	-1.28	1.34	91.94
							4	15607.27	-9384.26	-39794.37	-3.09	-1.26	1.34	91.94
393	2001	01	28	06	55	00	1	13991.35	-9924.86	-38822.09	-3.15	-1.24	1.43	91.94
							2	14716.57	-9676.52	-39199.44	-3.13	-1.23	1.38	91.94
							3	14693.68	-9579.65	-39020.88	-3.13	-1.26	1.40	91.94
							4	14677.29	-9762.36	-39384.17	-3.11	-1.25	1.40	91.94

d) visualisation 3D des trajectoires

Le contenu du fichier *cresatpos.resu*, peut être visualisés par la procédure ***visusatpos.bat***, lancé par la commande :

rco_visusatpos

Cette commande génère un fichier *visusatpos.ps*, qui visualise en 3D sous la forme d'un cube en perspective la trajectoires des 4 satellites, ainsi que leur projection sur les 3 plans orthogonaux du repère GSE ou GEI choisi. La distance géocentrique est également indiquée. Deux exemples de visualisation, sur une durée longue et une durée courte, sont données au § V-4.

3) utilitaires

a) copie dans le répertoire courant d'un jeu d'exemple de fichier *.in

Pour utiliser les procédures ***Orbit_Roproc***, il faut toujours indiquer les paramètres d'entrée dans des fichiers **.in*, comme *listorbit.in*, ou *cresatpos.in*.

Pour avoir des exemples sous la main, la commande suivante permet de recopier dans le directory courant un exemple de tous les fichier **.in* :

rco_copy_in

Après exécution, le directory courant contient les fichiers :

```
listorbit.in  
cresatpos.in
```

b) fichier readme.txt donnant la procédure d'installation de la chaîne Orbit_Roproc

Si on souhaite installer les procédures ***Orbit_Roproc*** sur une autre machine SUN/Solaris, le mode d'emploi est donné par la commande :

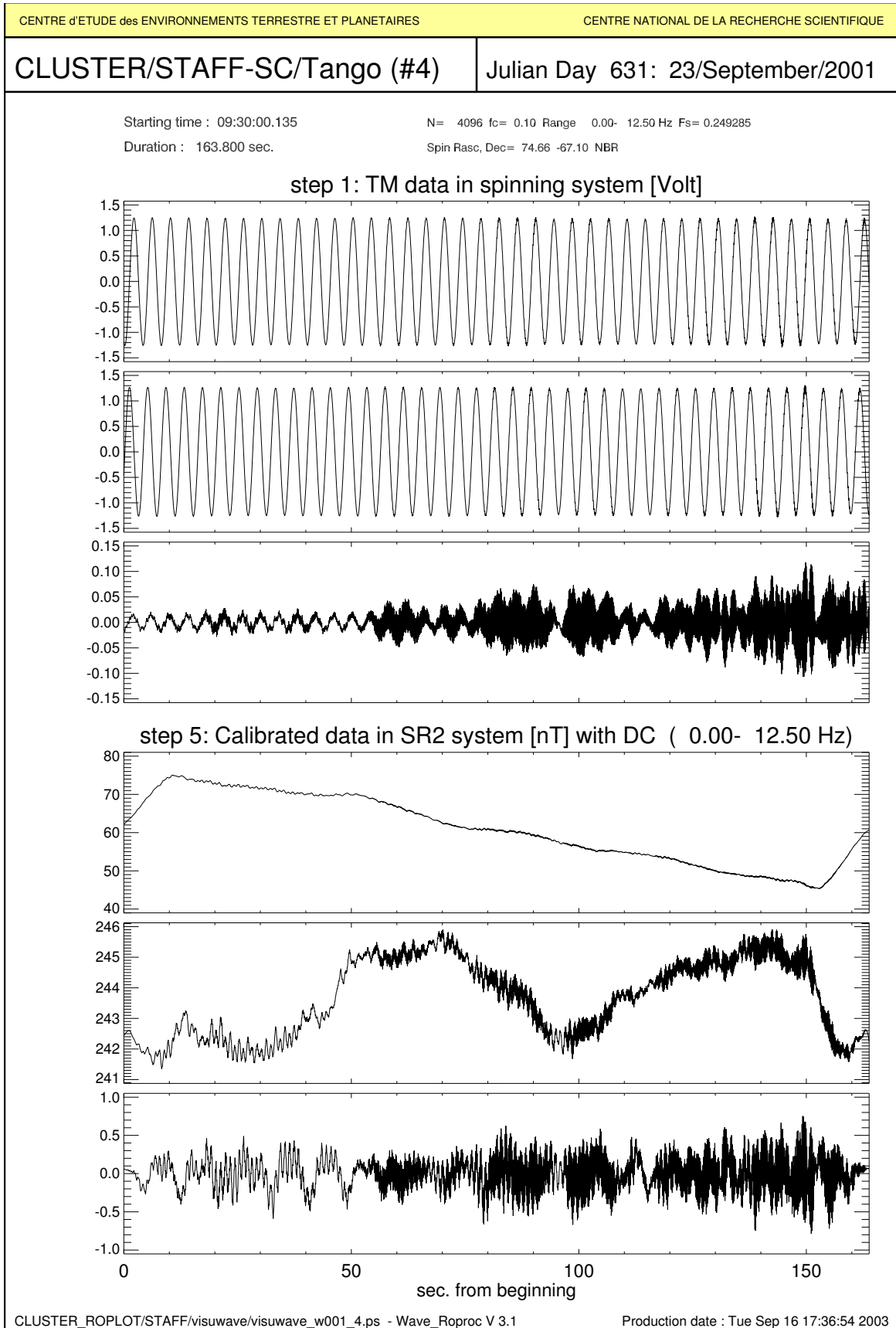
rco_readme

qui liste le fichier README.txt du package (il faut bien sûr l'avoir déjà installé ailleurs ; sinon, on peut aller voir directement ce fichier au début de l'arborescence du package).

V - EXEMPLES DE VISUALISATION POUR CLUSTER

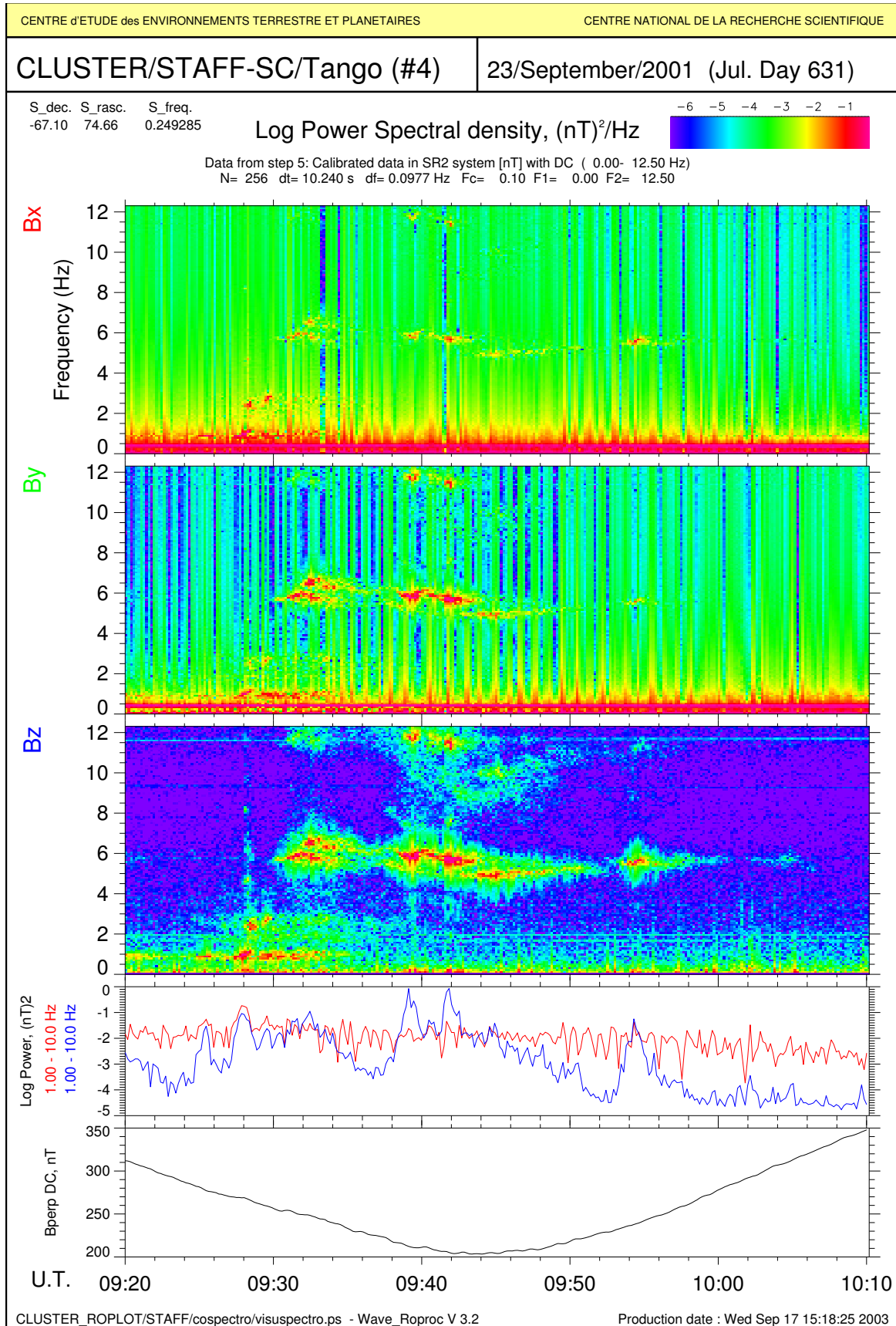
1) STAFF

a) visuwave.ps



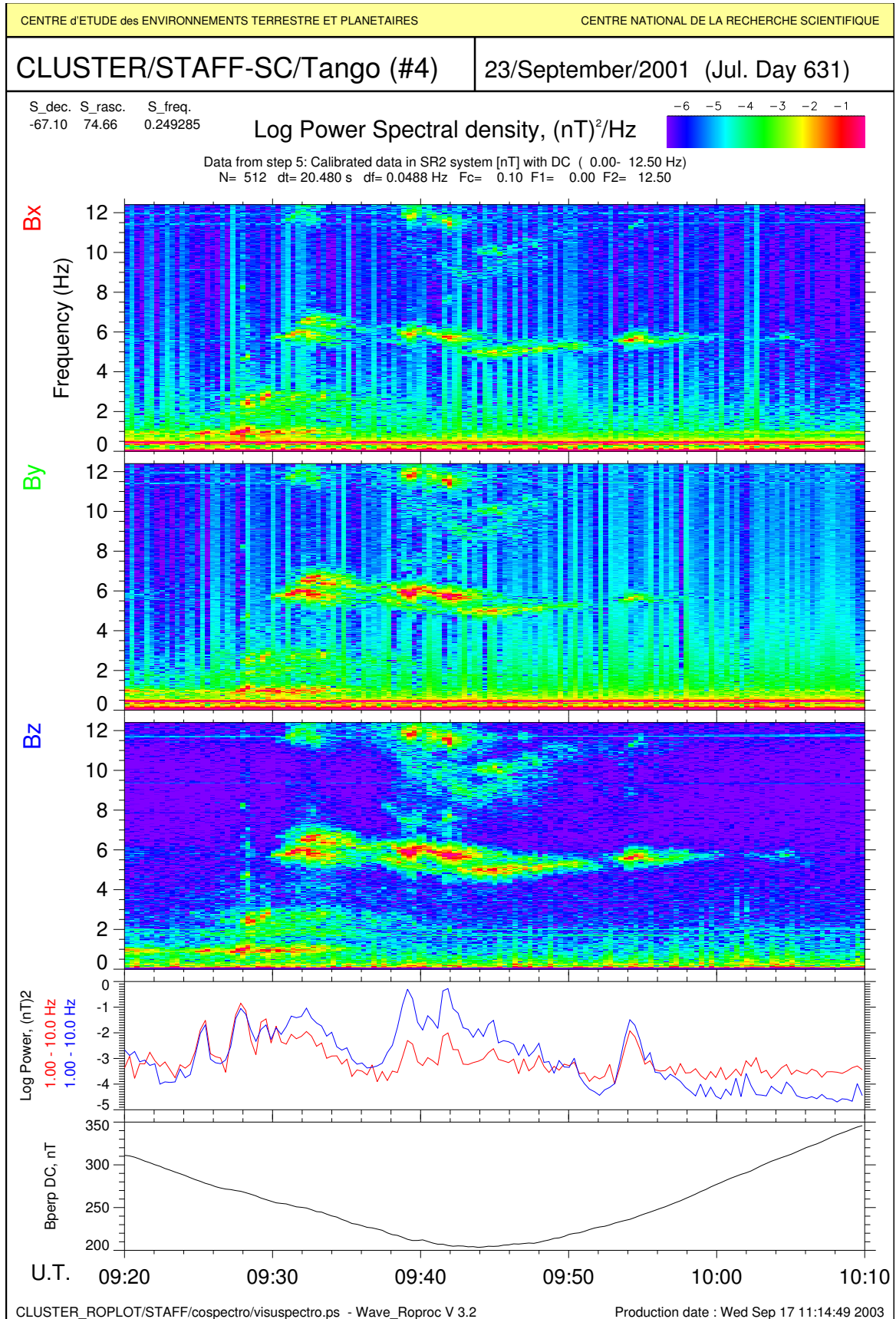
b) visuspectro.ps

Cet exemple montre la limitation du despin classique: en dessous de 4 périodes de spin (ici $N=256$, soit $dt=10.2$ s., soit environ 2.5 périodes de spin), le despin est insuffisant et les composantes Bx et By mal despinées ; sur la figure suivante, $N=512$, soit 20.4 s., ou plus de 4 périodes, et le despin est efficace.



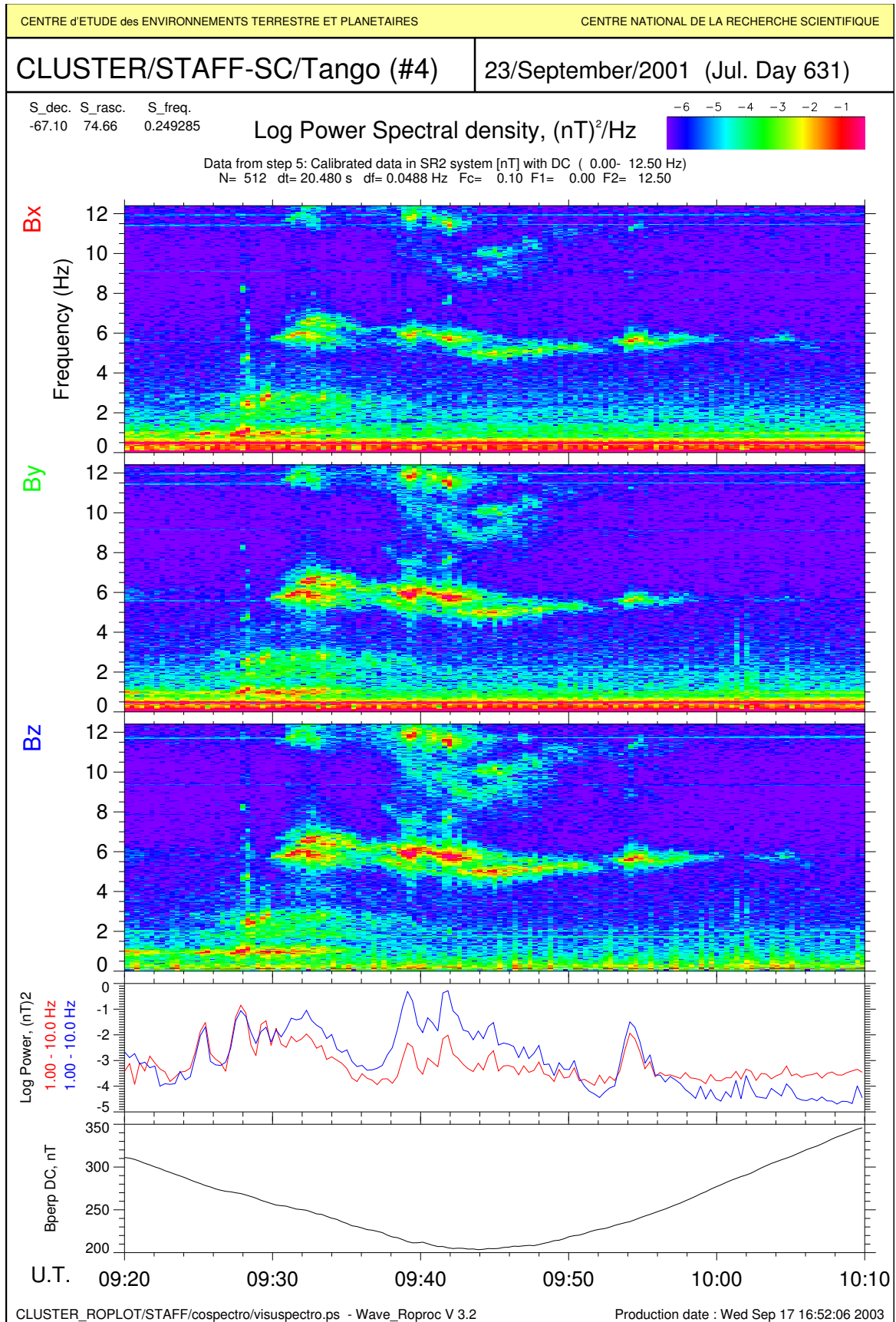
c) visuspectro.ps ; exemple avec fenêtre plus large

Exemple montrant cette fois l'efficacité du despin classique à partir d'échantillon de plus de 4 périodes de spin (N=512, soit 20.4 s), à comparer avec la figure précédente. Dans ce choix a petite échelle (moins d'une heure), le gain sur l'efficacité du despin se fait au détriment de la résolution en temps.



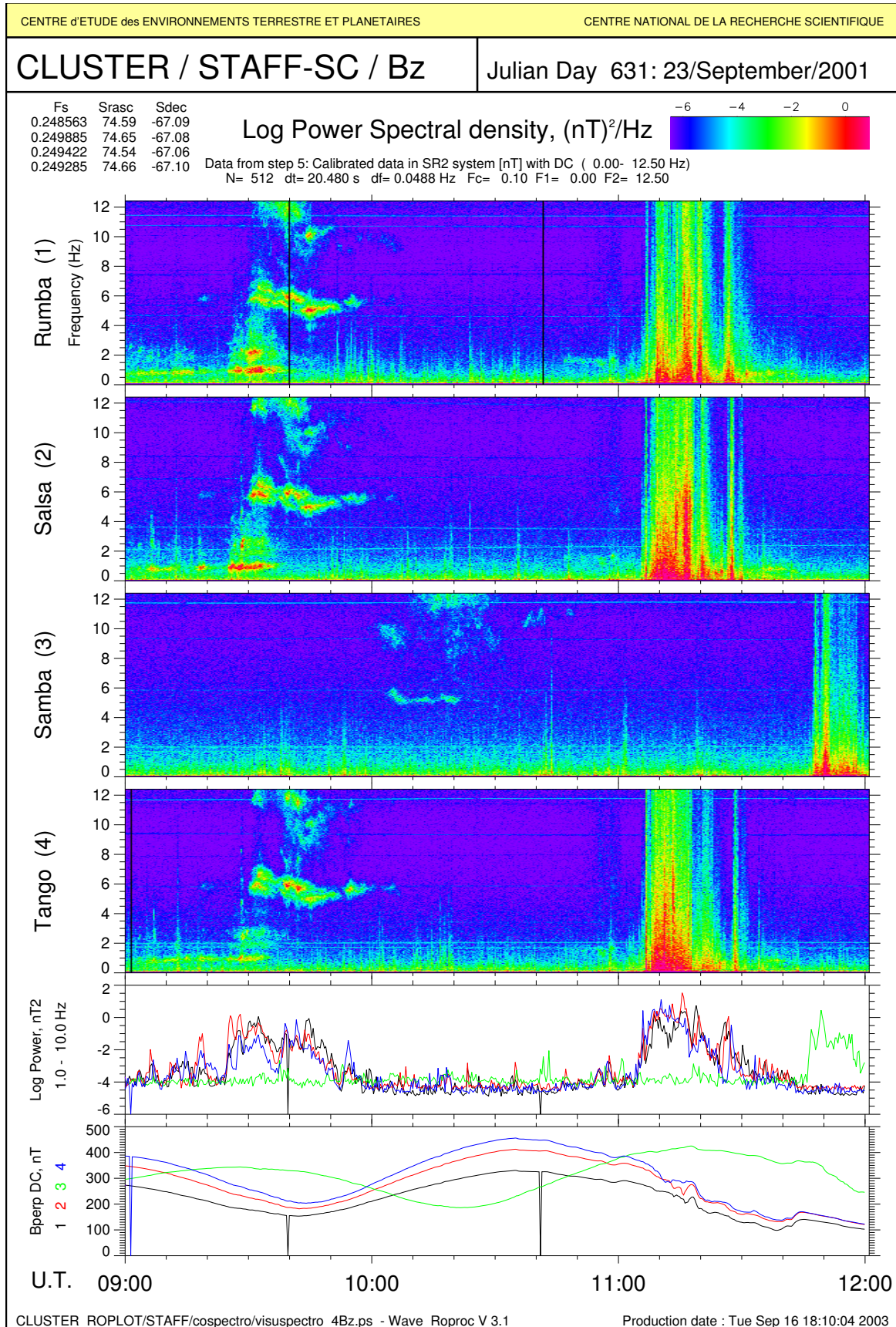
d) visuspectro.ps ; exemple avec apodisation

Augmentation de la qualité du spectrogramme avec l'option apodisation=1



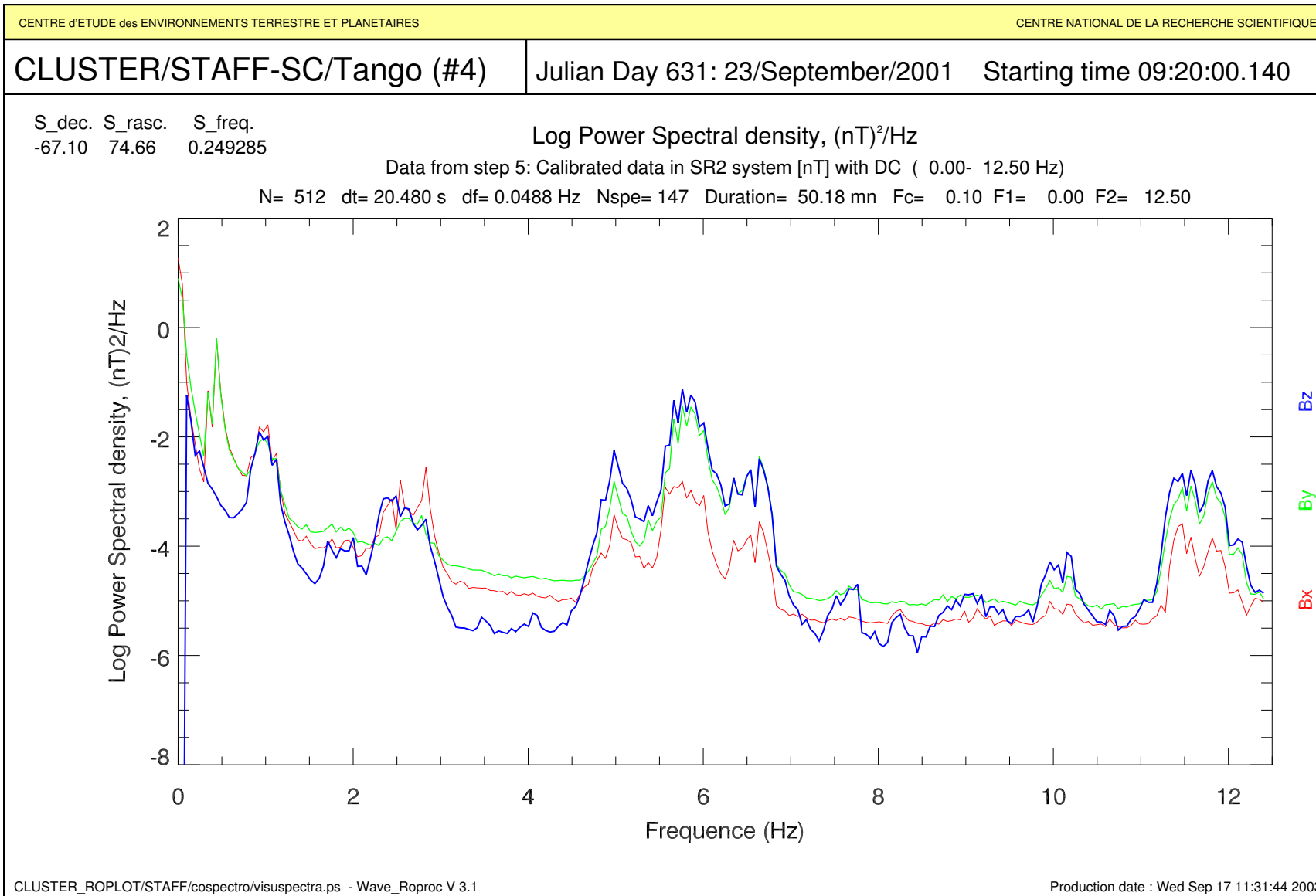
e) visuspectro_4Bz.ps

Cet exemple illustre bien le décalage en temps des événements vus par chaque satellites. Le choix de $N=512$ et de la période totale de 3 heures est un bon compromis pour la résolution $\Delta F \cdot \Delta T$ du spectrogramme.



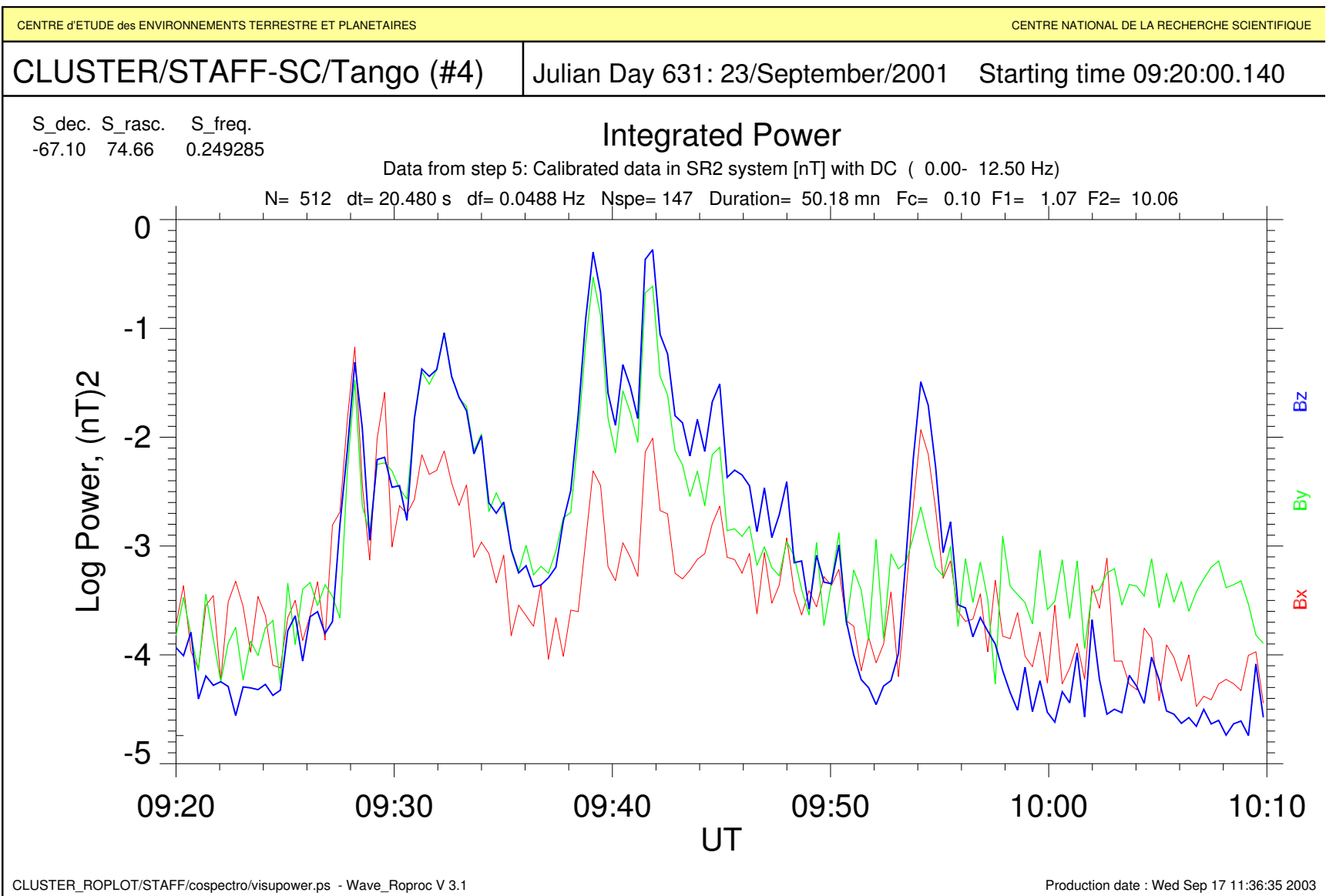
f) visuspectra.ps

Ce spectre correspond à l'intégration en temps du spectrogramme précédent. Les fréquences décelées sur le spectrogramme sont maintenant clairement visibles et leur intensité mesurables.



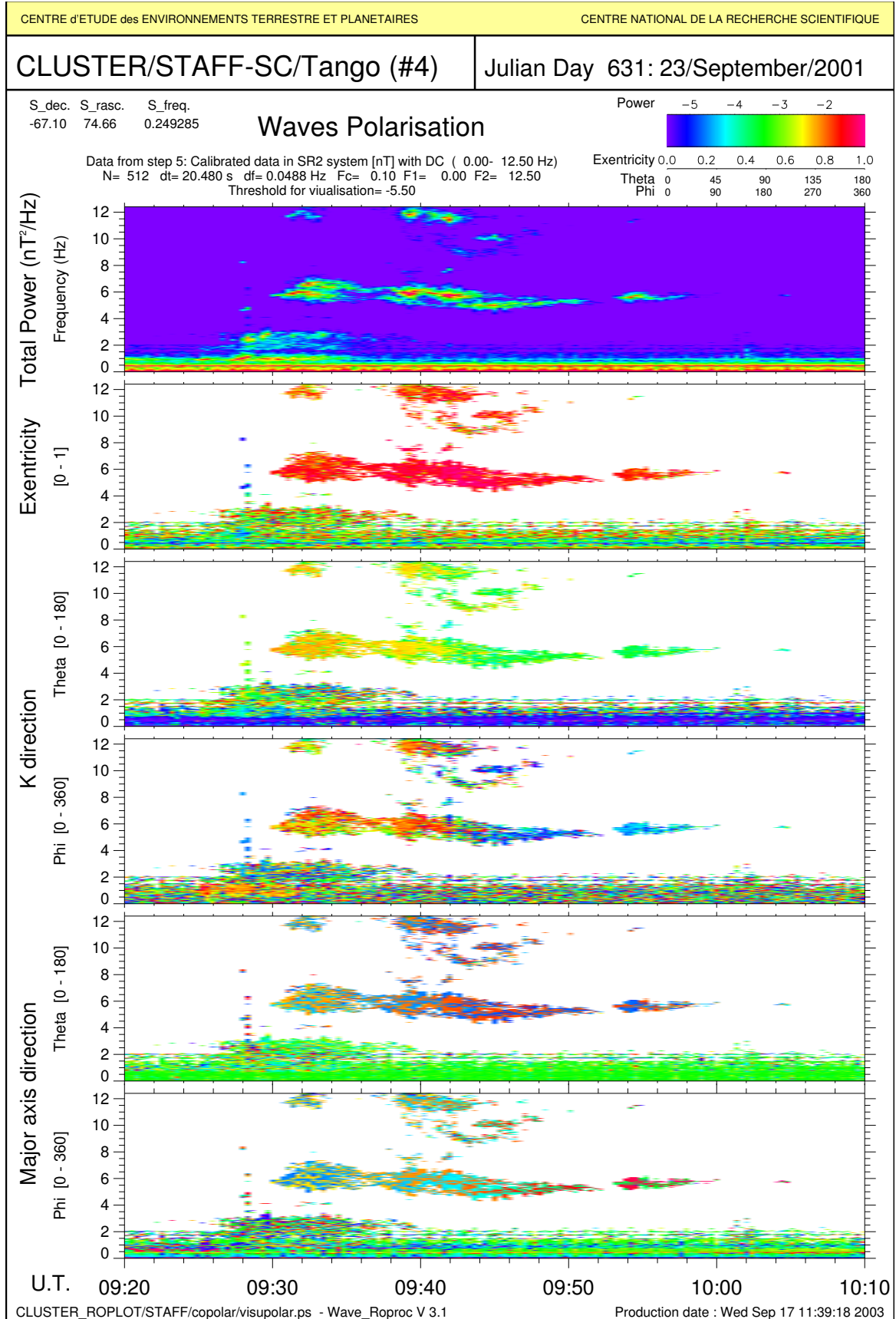
g) visupower.ps

Cette puissance au cours du temps correspond à l'intégration en fréquence du spectrogramme précédent. Les plages de temps décelées sur le spectrogramme sont maintenant clairement visibles et leur intensité mesurables.



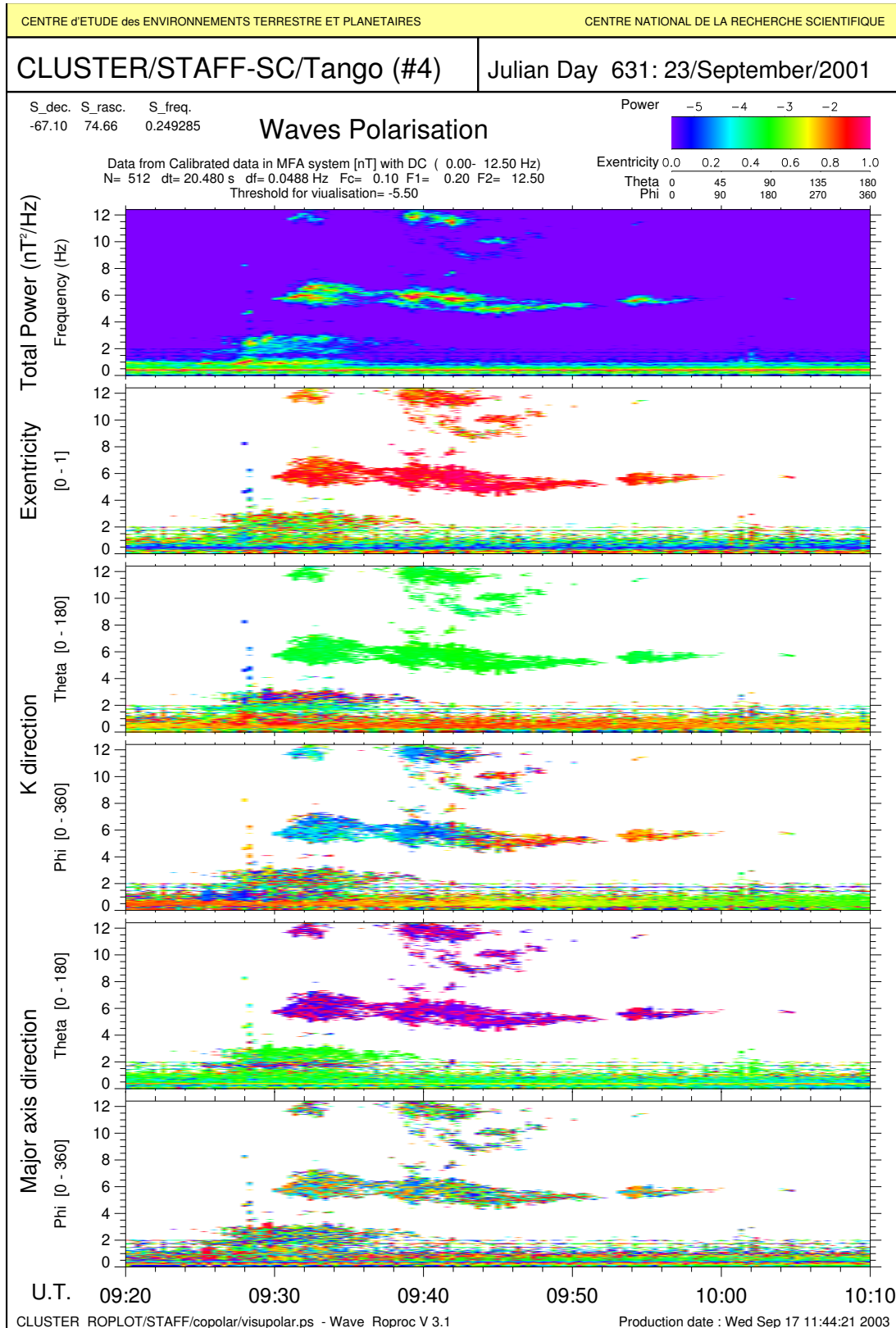
h) visupolar.ps

Exemple montrant une polarisation quasi-linéaire (excentricité proche de 1.), mais le calcul étant effectué dans le repère satellite SR2, la direction de K n'a pas grand sens physique (voir figure suivante).



i) visupolar.ps en repère MFA

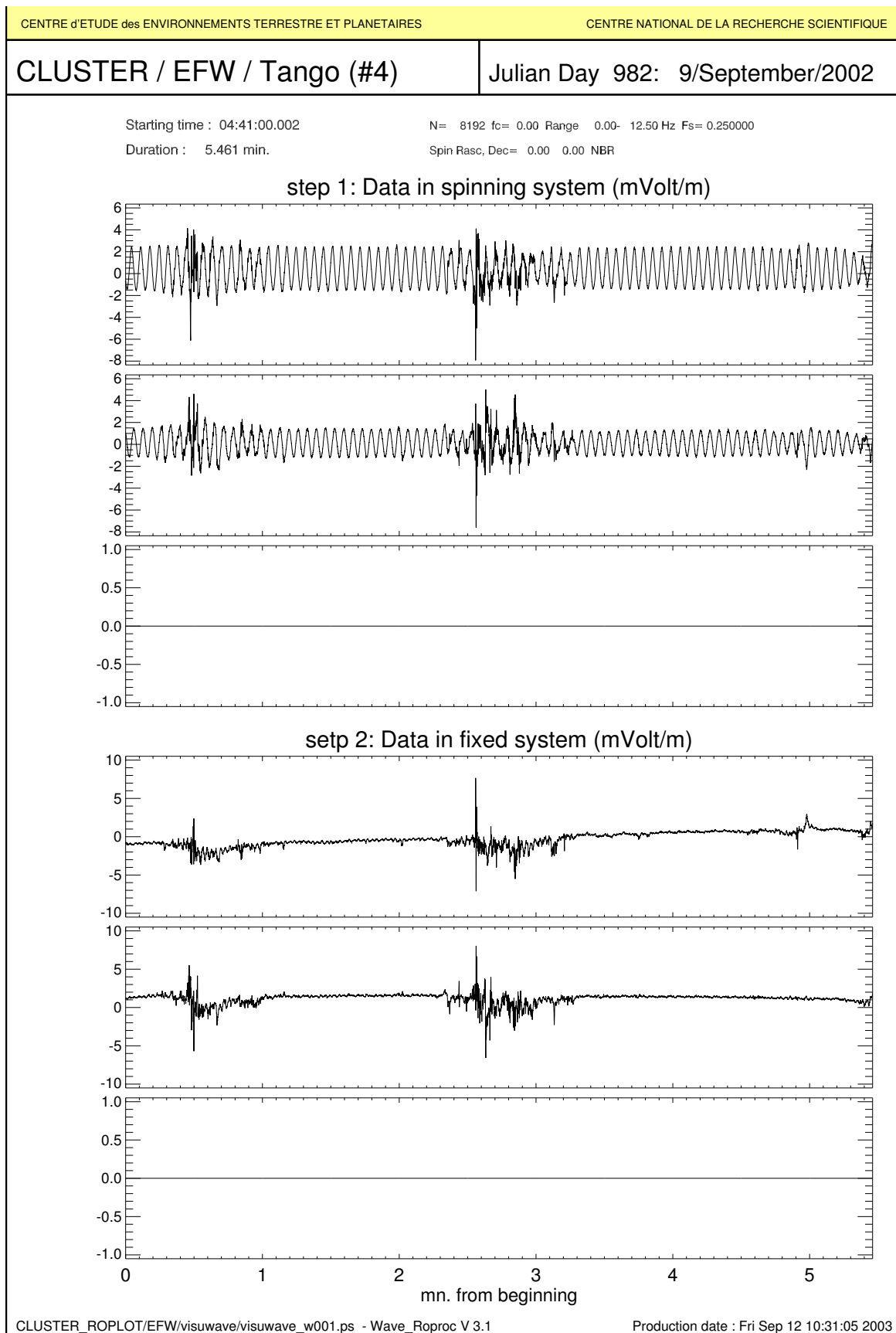
Illustration de l'utilité du filtre **wave_to_mfa** où la polarisation des ondes prends tout son sens physique : on trouve le vecteur K perpendiculaire au champ continu B_0 ($\theta = 90^\circ$, en vert), l'ellipse quasi-dégénérée en une droite (excentricité proche de 1, en rouge), et le grand axe aligné avec B_0 ($\theta = 0$ ou 180 , en rouge ou bleu). Avec des valeurs de θ proches de 0 ou 180° , les angles de phase ϕ sont évidemment mal définis.



2) EFW

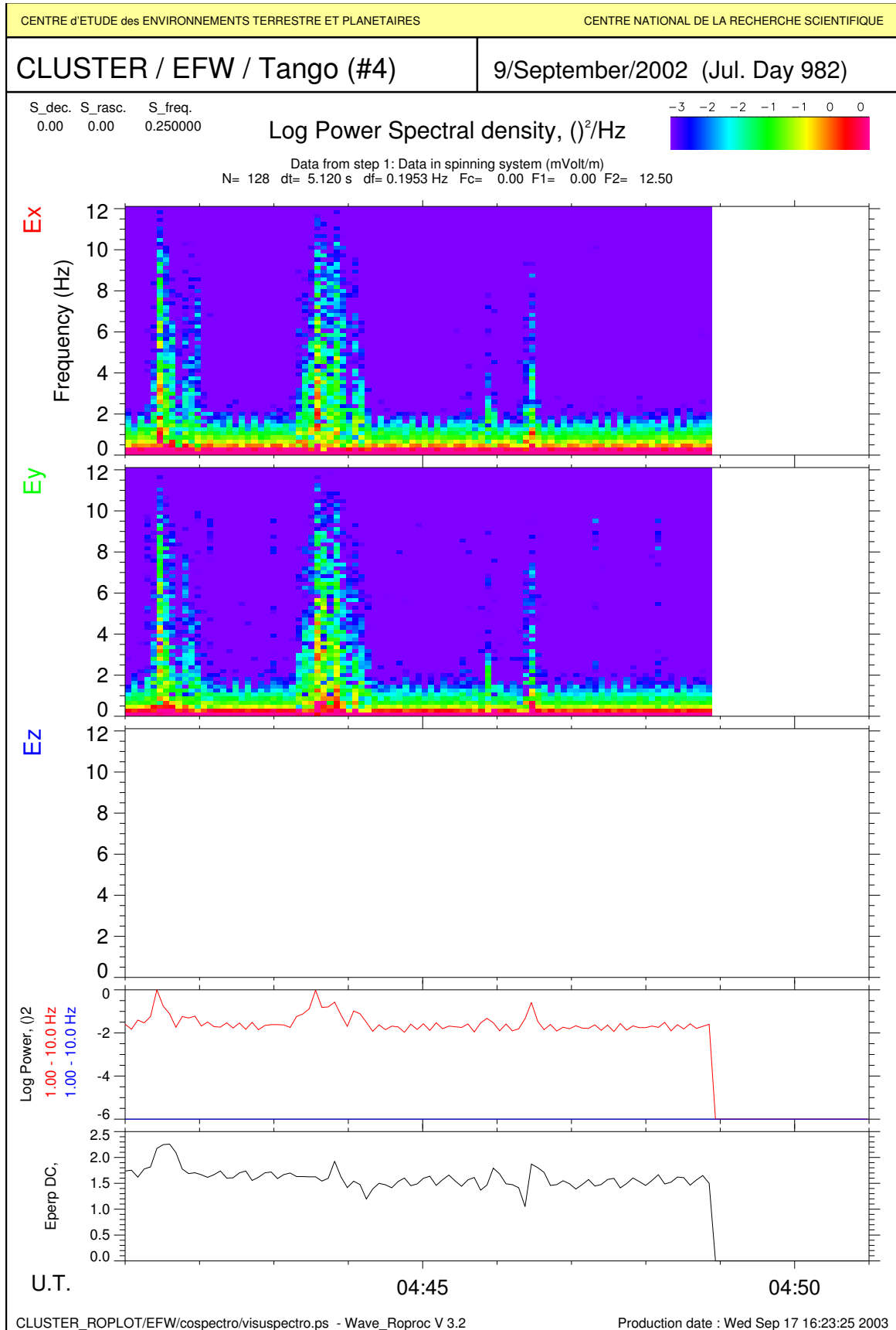
a) visuwave.ps

Il existe seulement 2 composantes x et y pour cette expérience ; z est mis à zero.



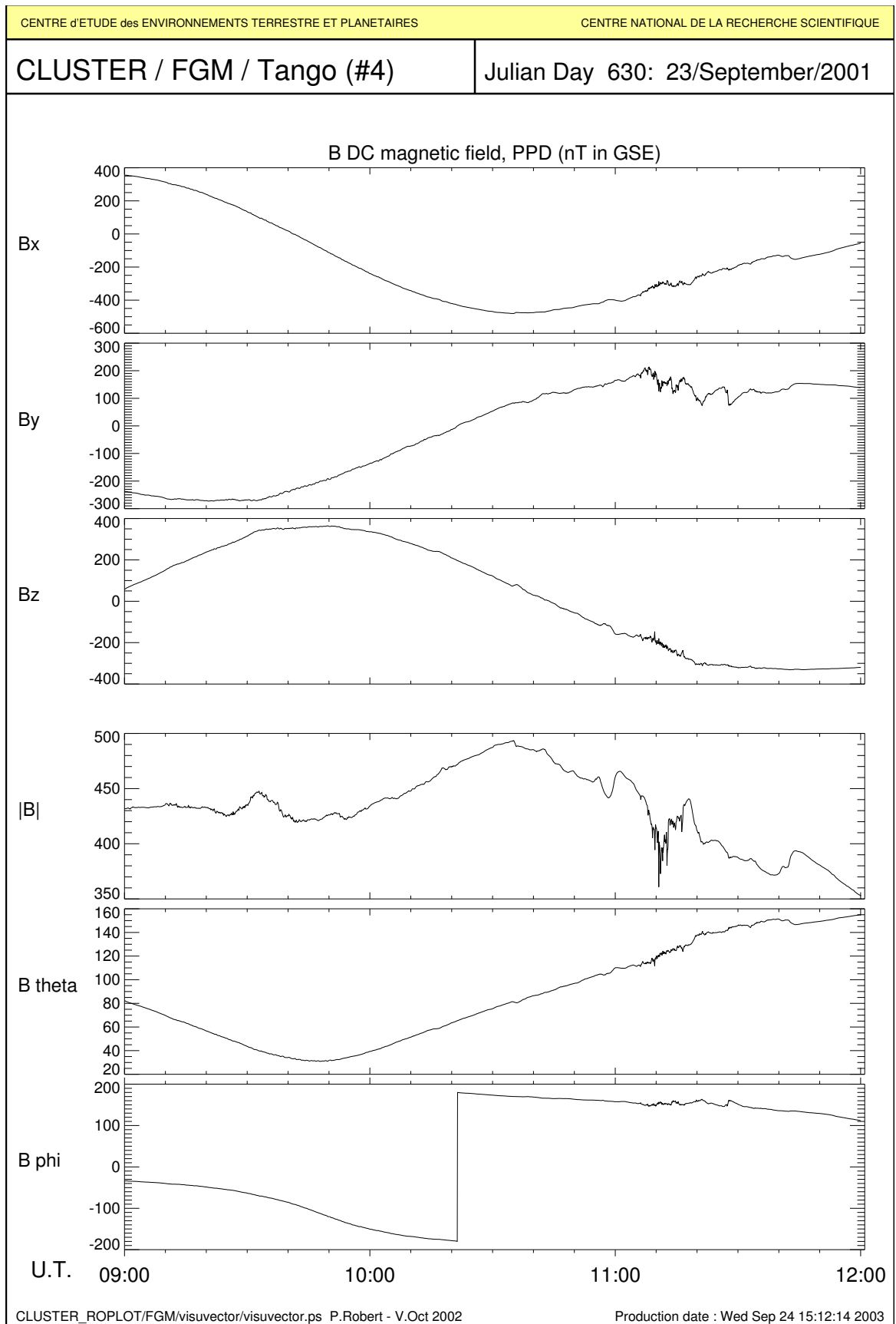
b) visuspectro.ps

Par l'intermédiaire de la procédure *cowave* réalisée pour EFW, un « flat file » issu de ISDAT peut être converti au format *cowave.resu* des procédures *Wave_Roproc*, et on peut donc appliquer sur ces données les mêmes traitements que ceux de STAFF-SC.



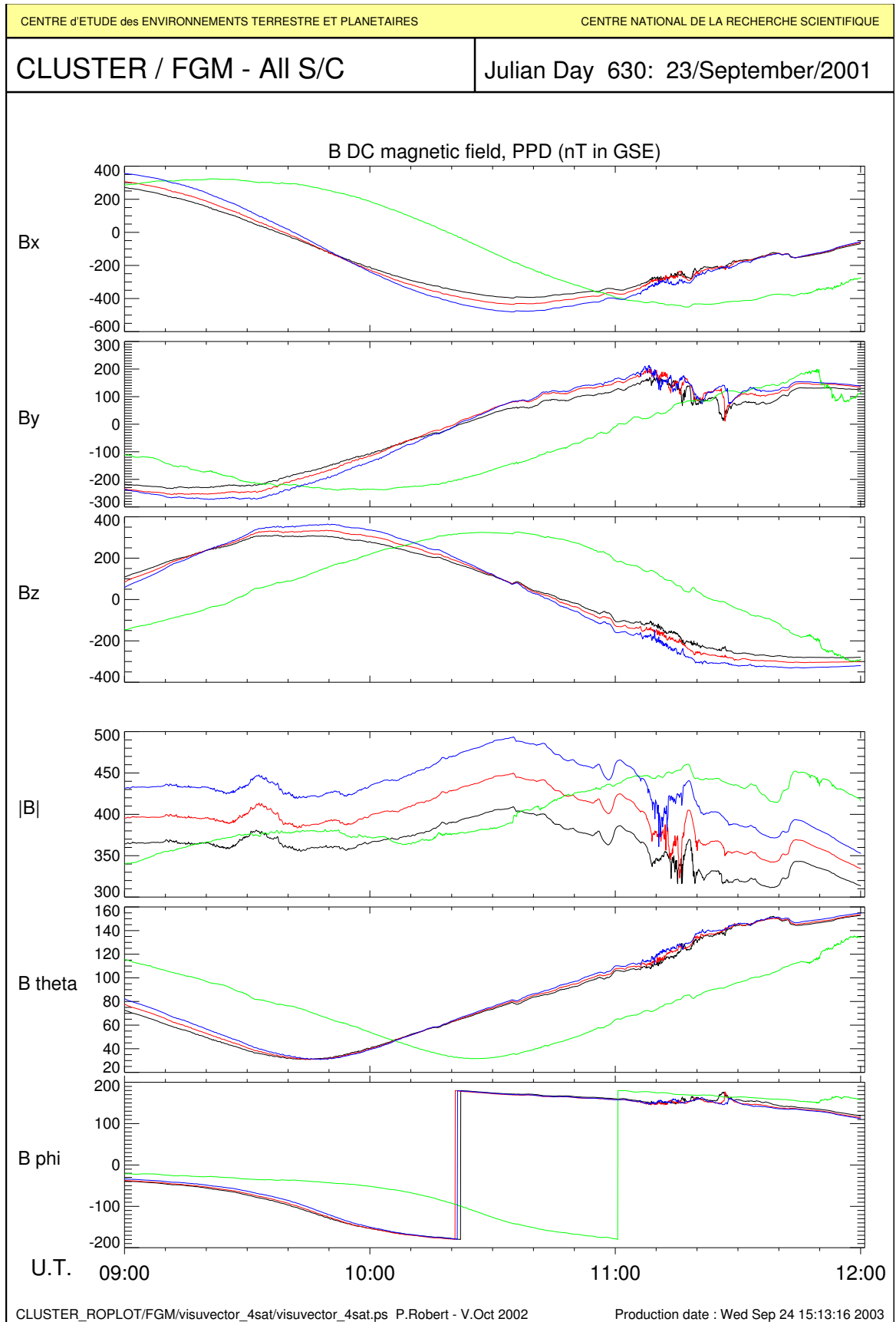
3) FGM

a) visuvector.ps



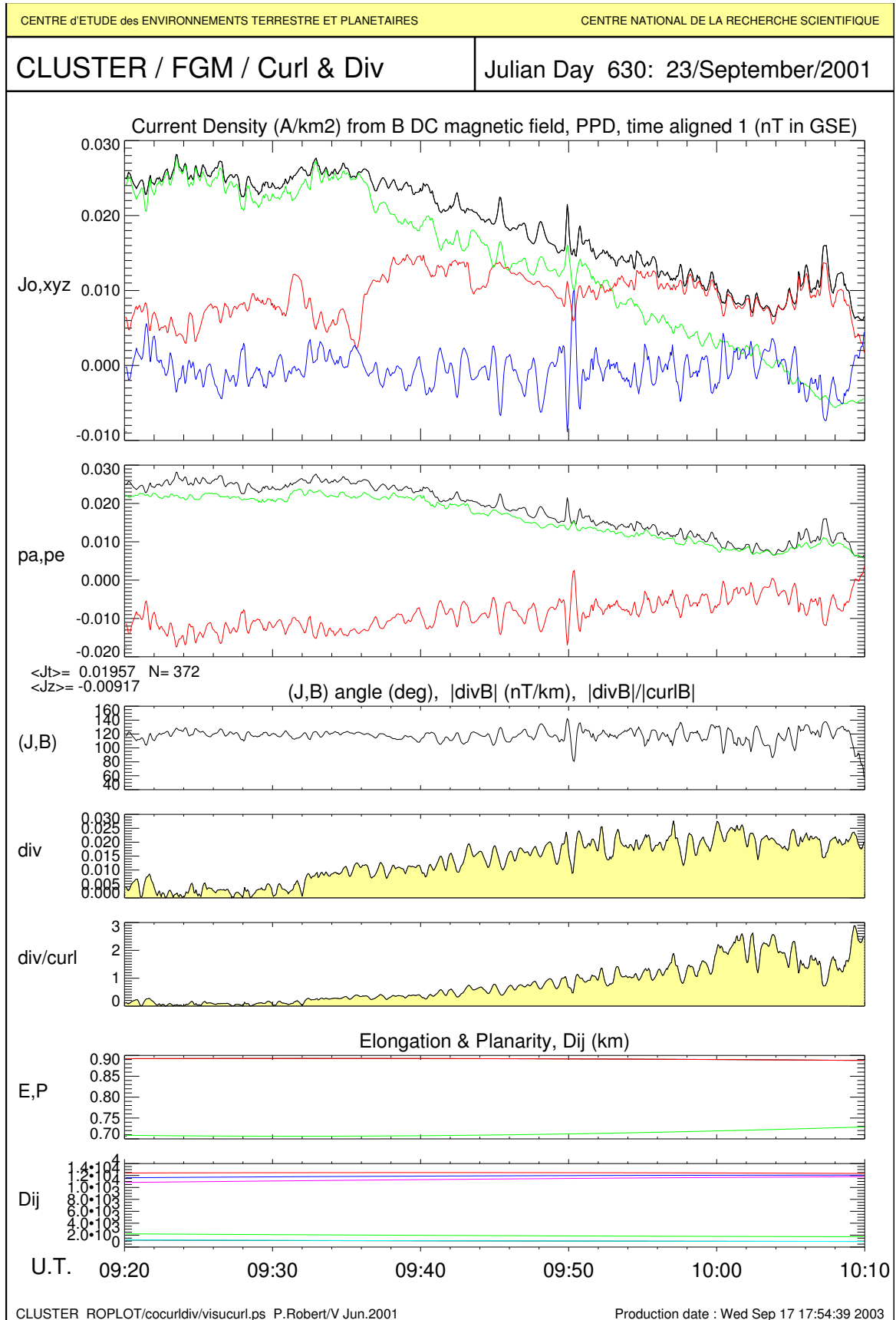
b) visuvector_4sat.ps

Conventions de couleurs habituelles : sat. 1 en noir, sat. 2,3,4 en rouge, vert, bleu.



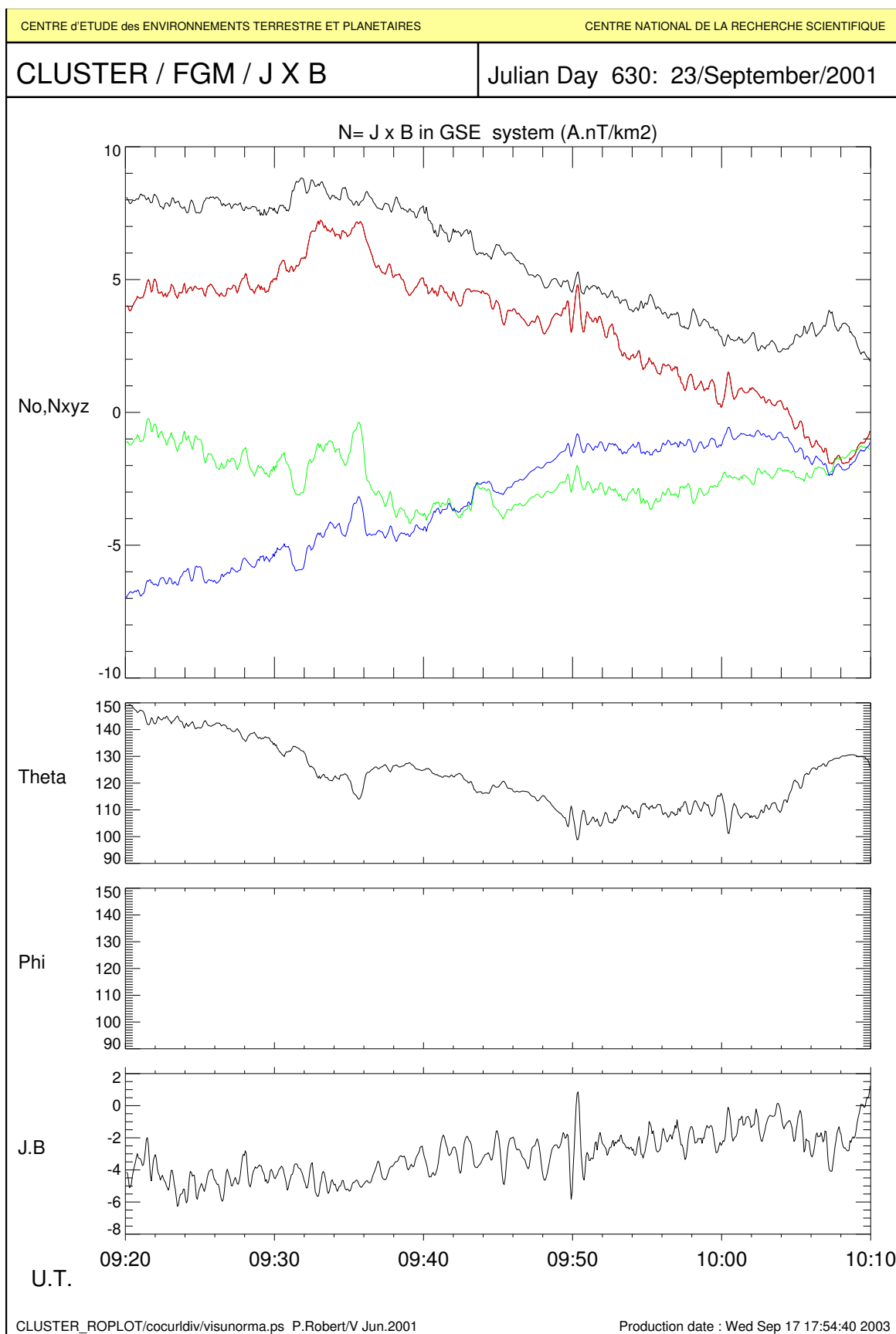
c) visucurl.ps

Les courbes (pa,pe) en rouge et vert se réfèrent aux composantes de J parallèle et perpendiculaires au champ magnétique moyen.



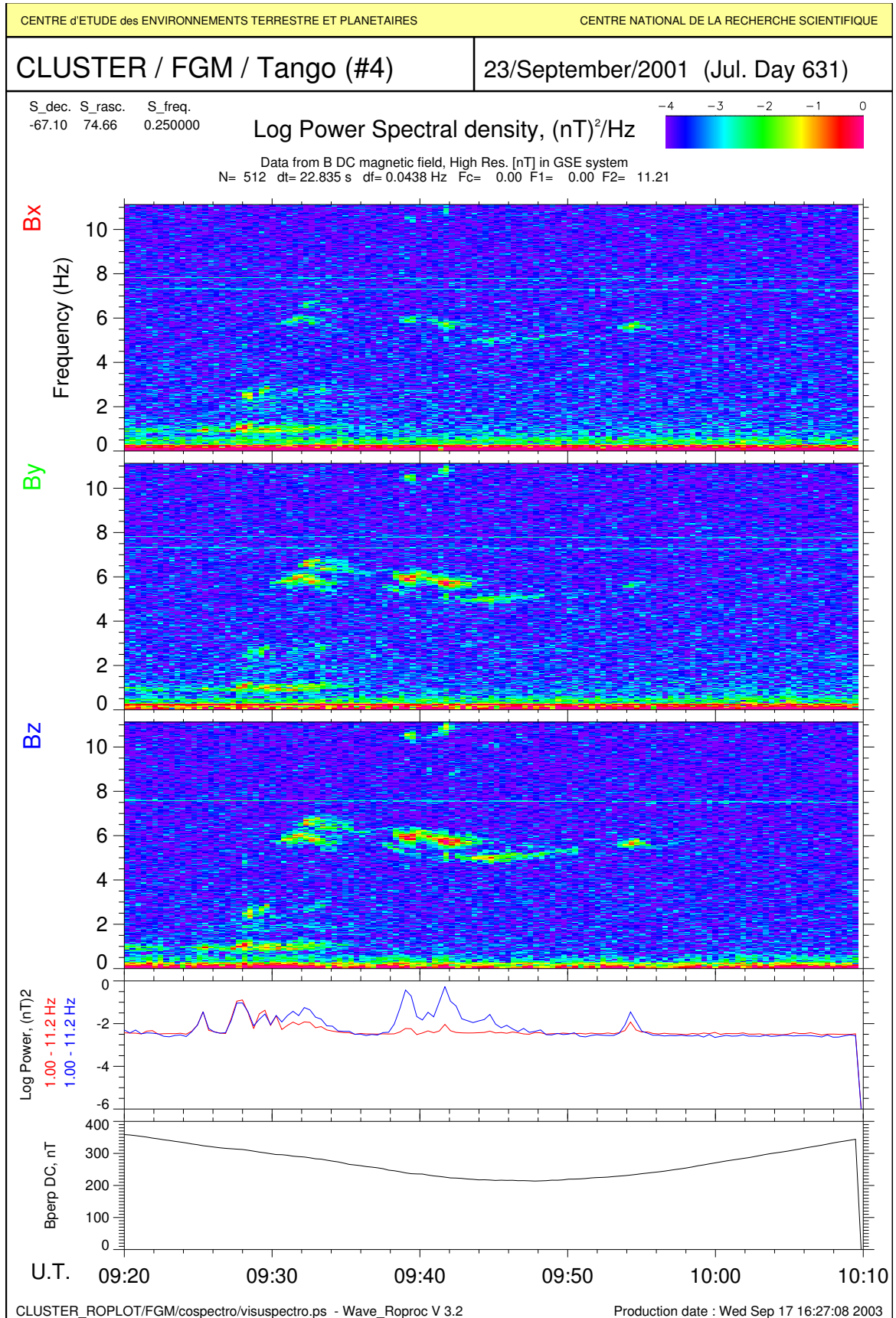
d) visunorma.ps

Conventions de couleurs habituelles : module en noir, x, y, z, en rouge, vert, bleu.



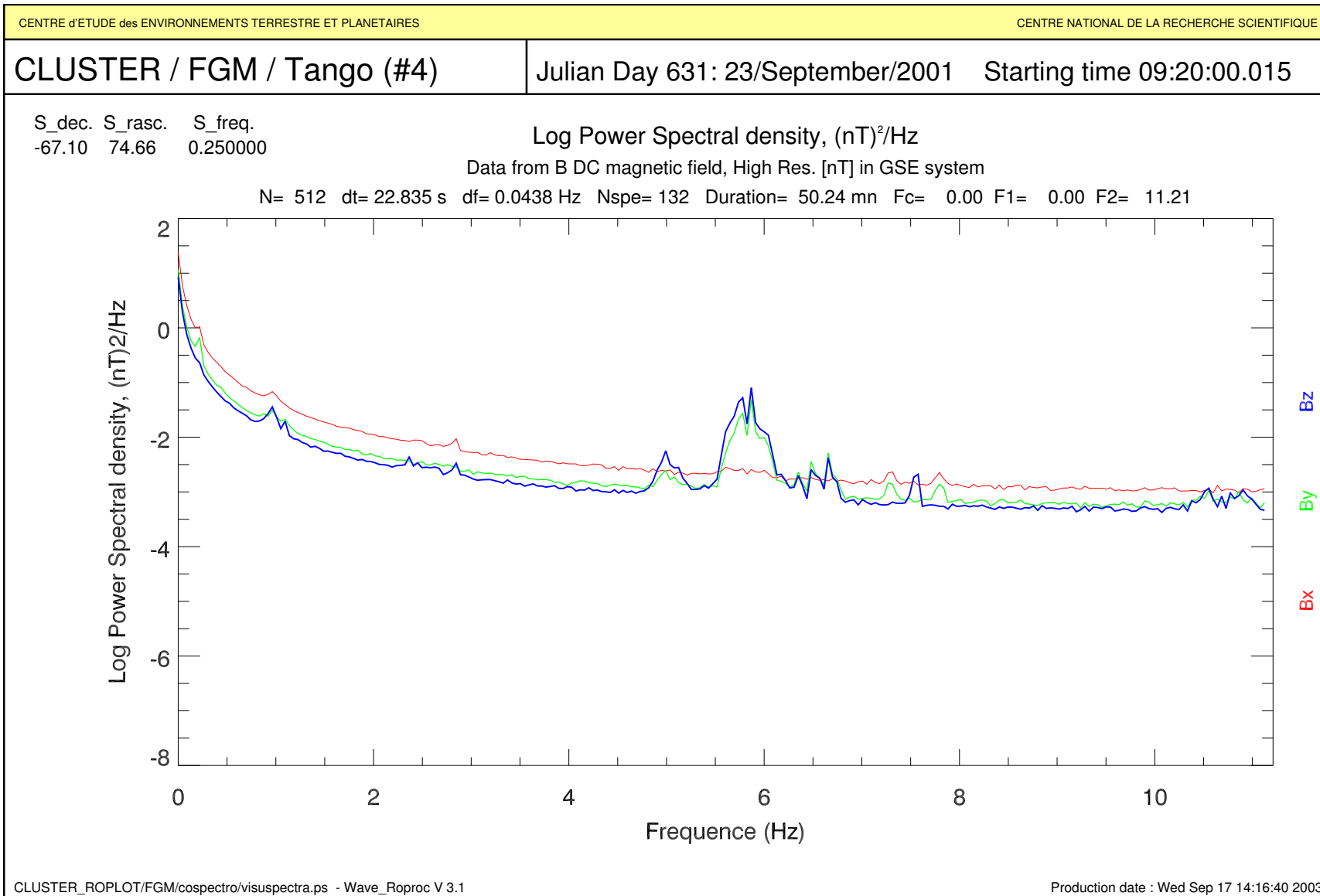
e) visuspectro.ps

Exemple d'utilisation des procédures *rsc_cospectro* et *rsc_visuspectro* de la famille *Wave_Ropro* sur des données de FGM à haute résolution converties à ce format par la procédure *rsc_vector_to_wave*. Ce résultat peut être comparé à celui obtenu avec les données STAFF au § V-1-d.



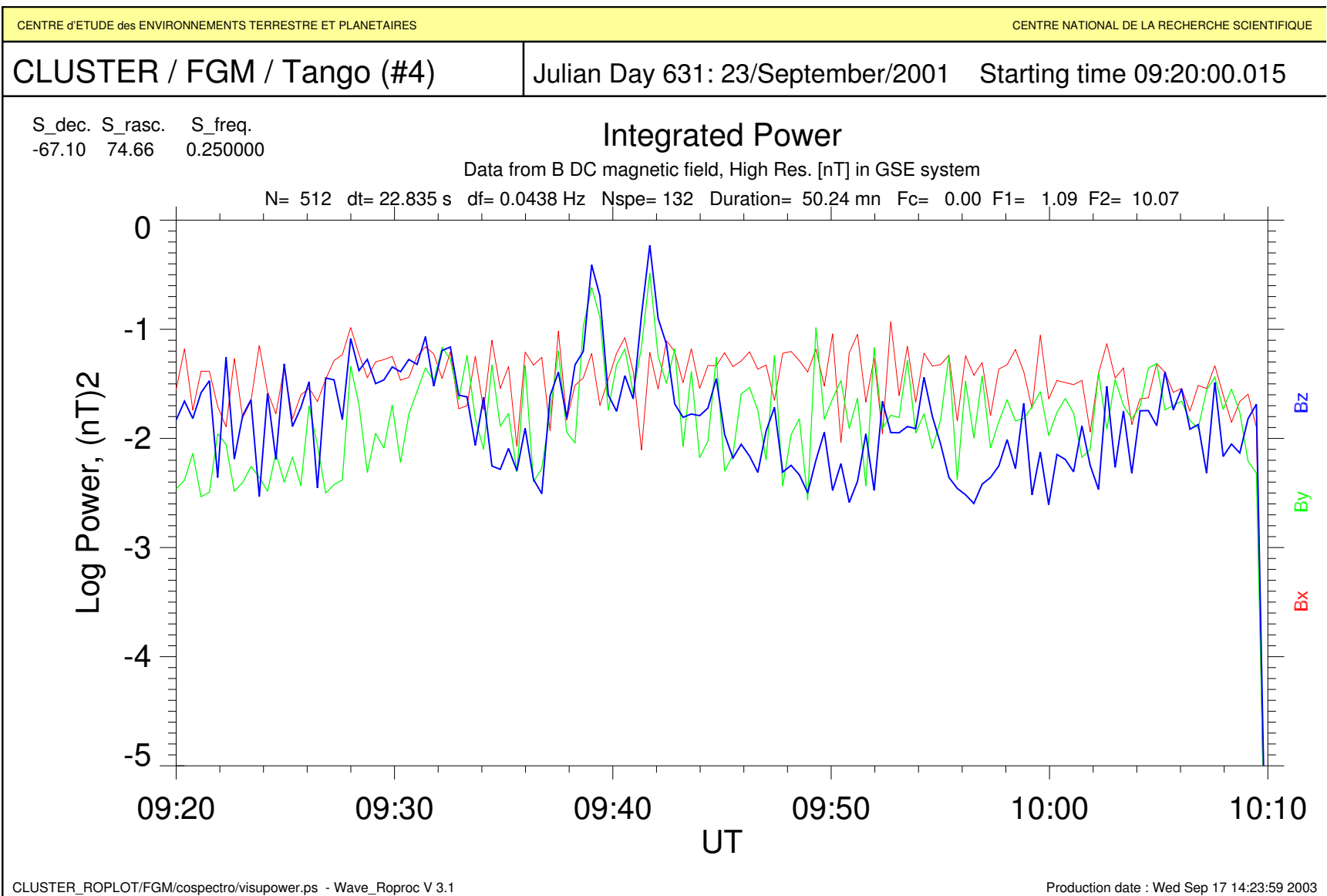
f) visuspectra.ps

Autre exemple d'utilisation des procédures *res_cospectro* et *res_visuspectra* de la famille *Wave_Roproc* sur des données de FGM converties à ce format par la procédure *ref_vector_to_wave*. Ce résultat peut être comparé à celui obtenu avec les données STAFF au § V-1-f.



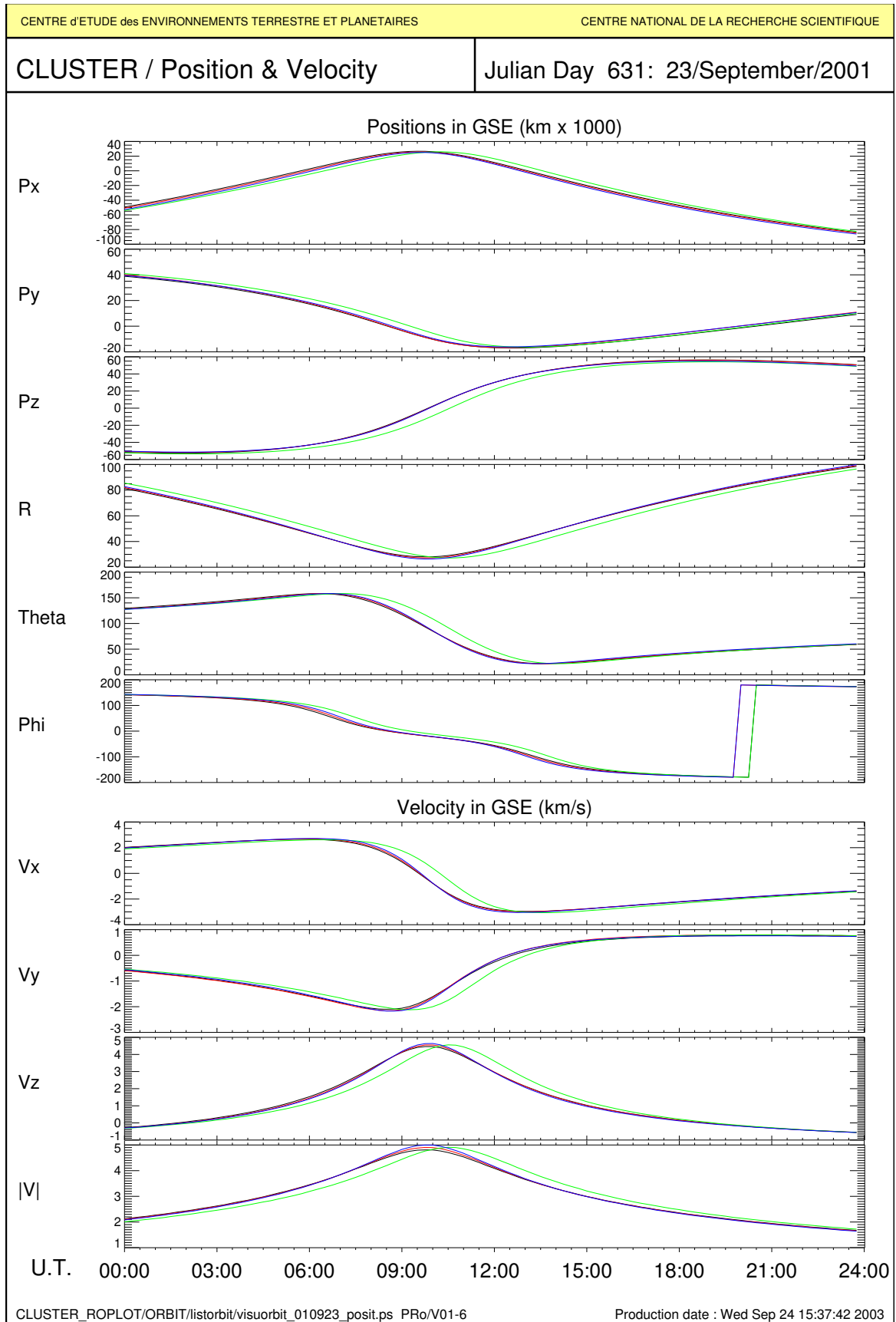
g) visupower.ps

Autre exemple d'utilisation des procédures *res_cospectro* et *res_visupower* de la famille *Wave_Roproc* sur des données de FGM converties à ce format par la procédure *rcf_vector_to_wave*. Ce résultat peut être comparé à celui obtenu avec les données STAFF au § V-1-g.



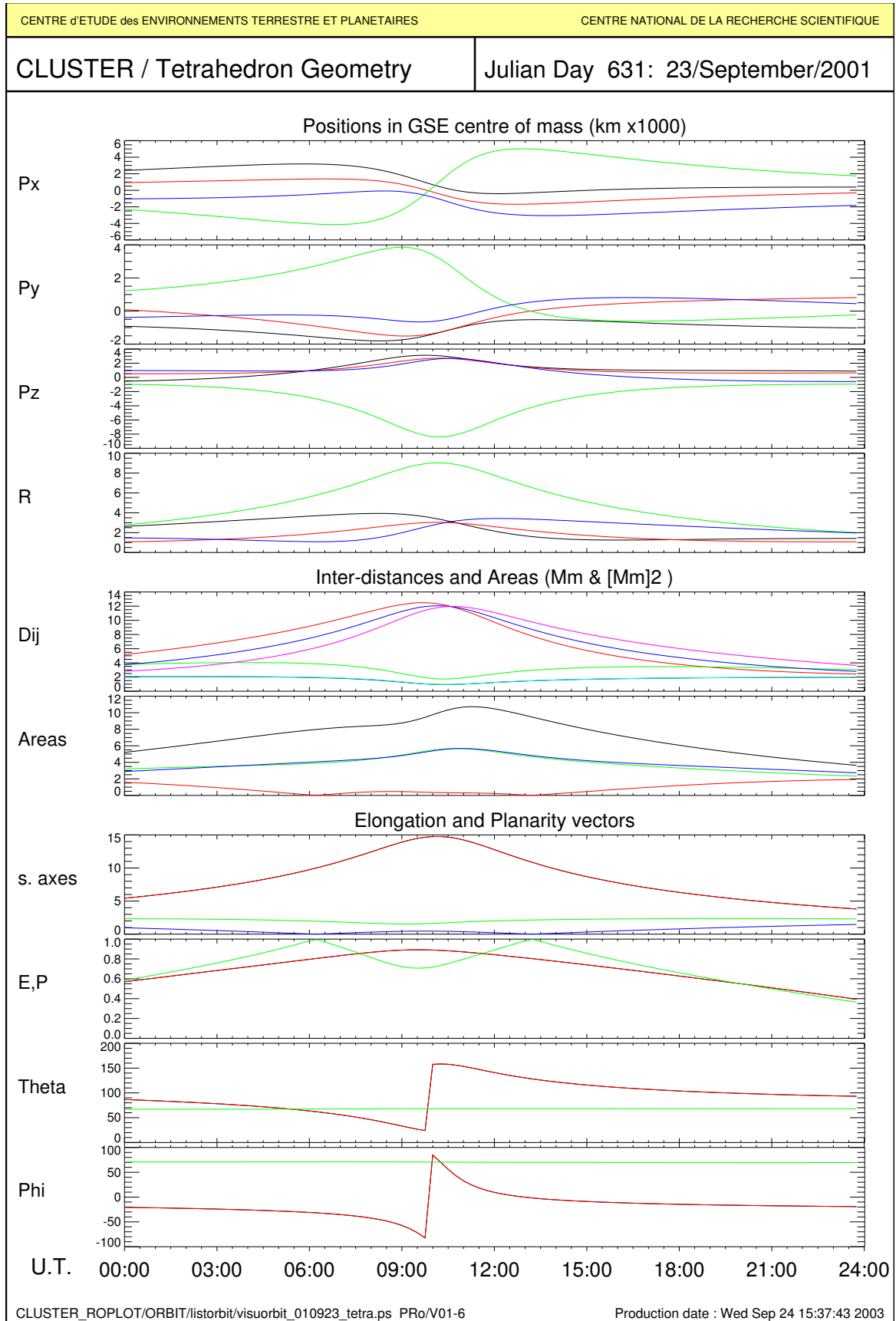
4) ORBIT

a) visuorbit_posit.ps



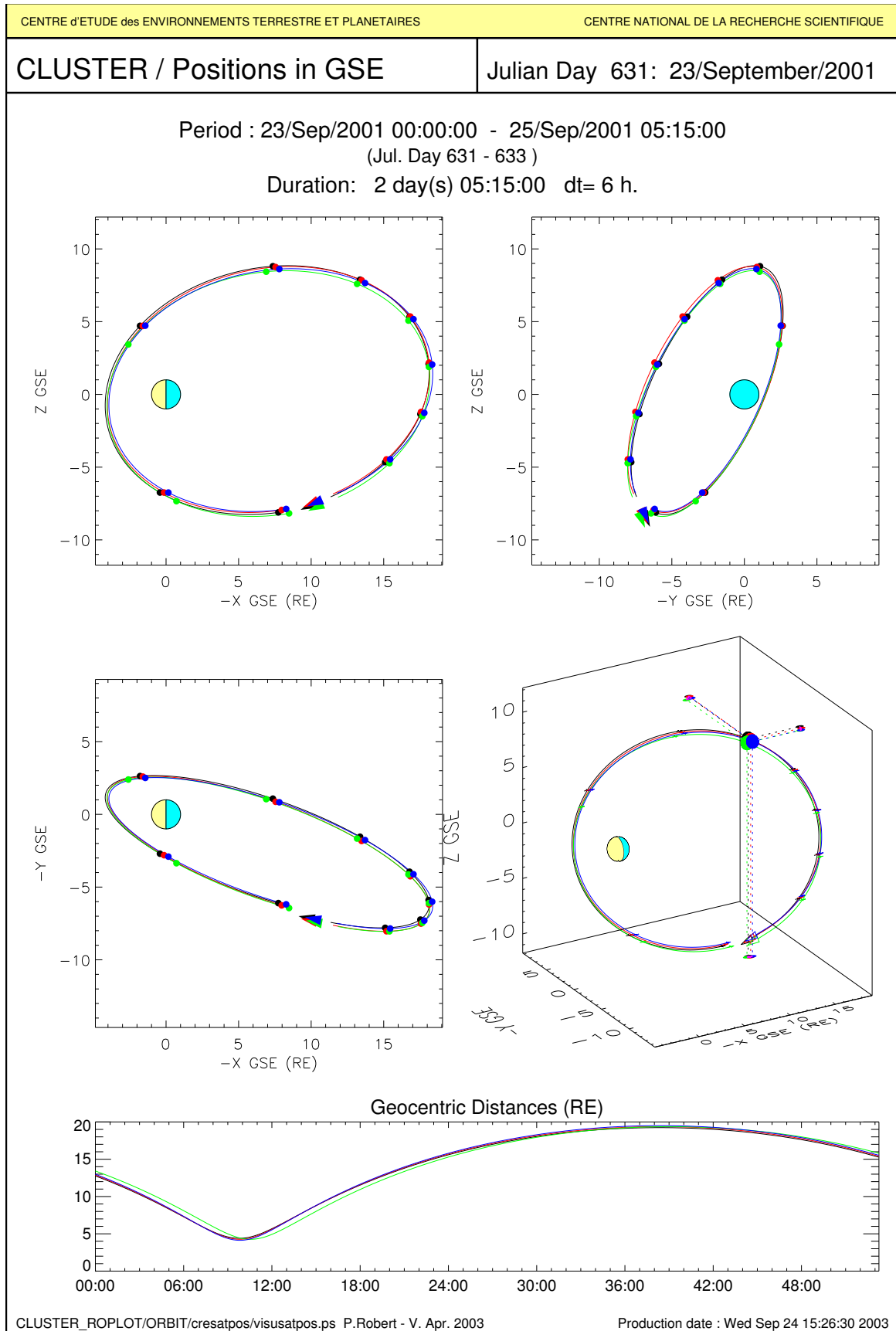
b) visuorbit_tetra.ps

L'unité Mm signifie Mega mètres = km x 1000



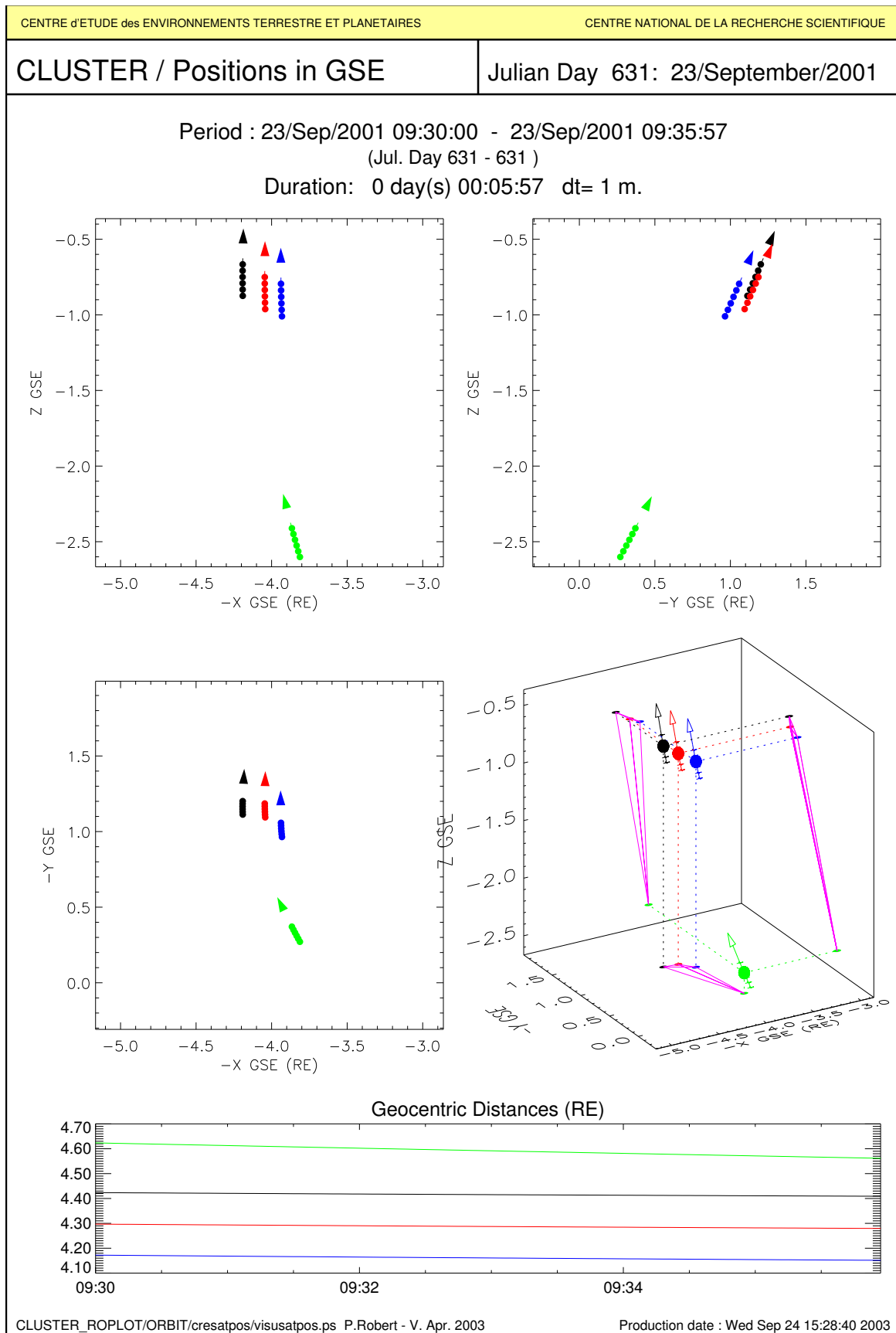
c) visusatpos.ps sur 2.5 jours

Conventions de couleurs habituelles : sat. 1 en noir, sat. 2,3,4 en rouge, vert, bleu. Mais à cette échelle, les trajectoires des 4 satellites sont quasiment confondues.



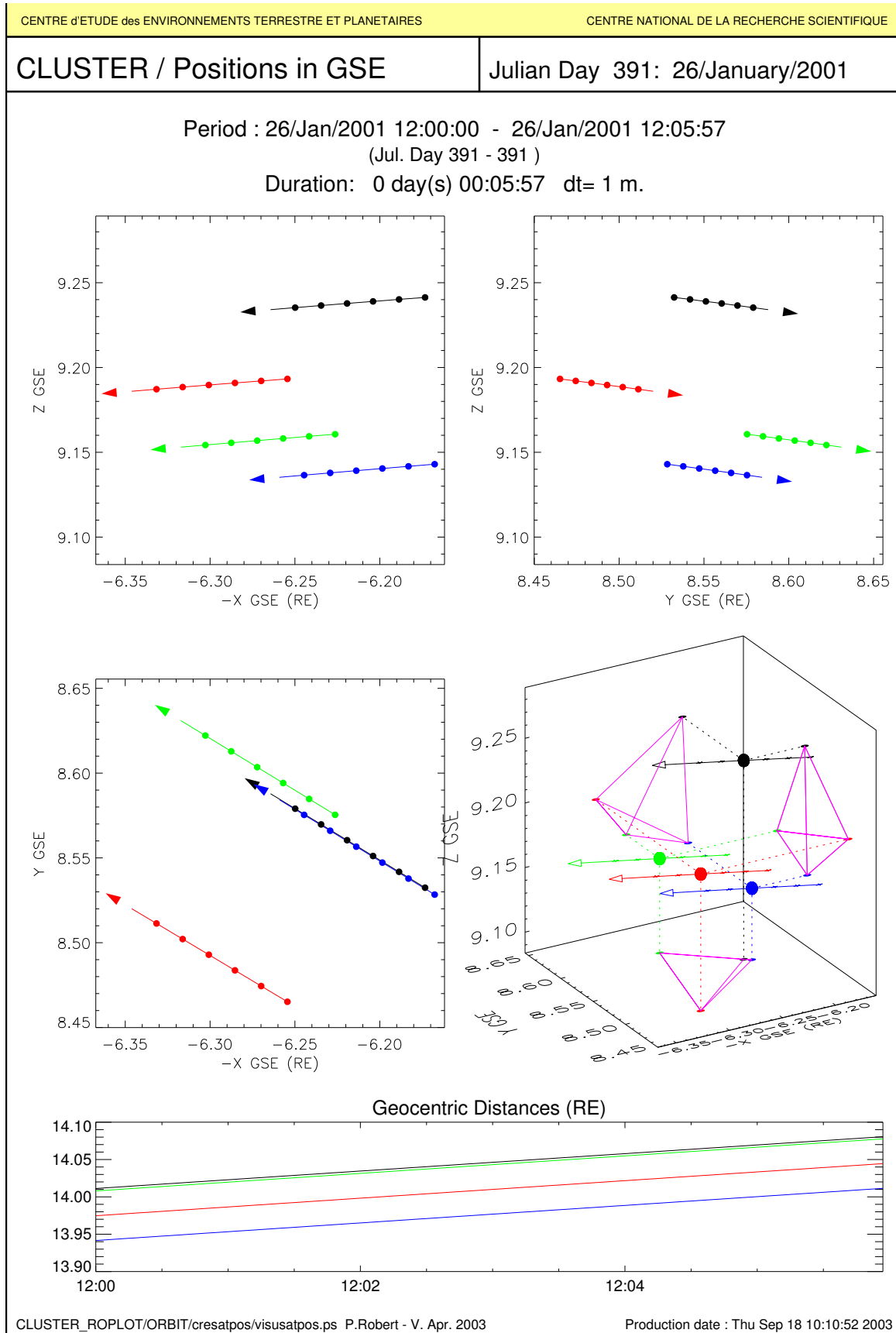
d) visusatpos.ps sur 6 minutes

Ce zoom sur les positions permet de distinguer les positions respectives de chaque satellite, et de visualiser la forme du tétraèdre, ici fortement allongée.



e) autre exemple de visusatpos.ps pour un tétraèdre régulier

Cet exemple permet de vérifier que le tétraèdre est presque régulier ($E = 0.2$, $P = 0.3$).



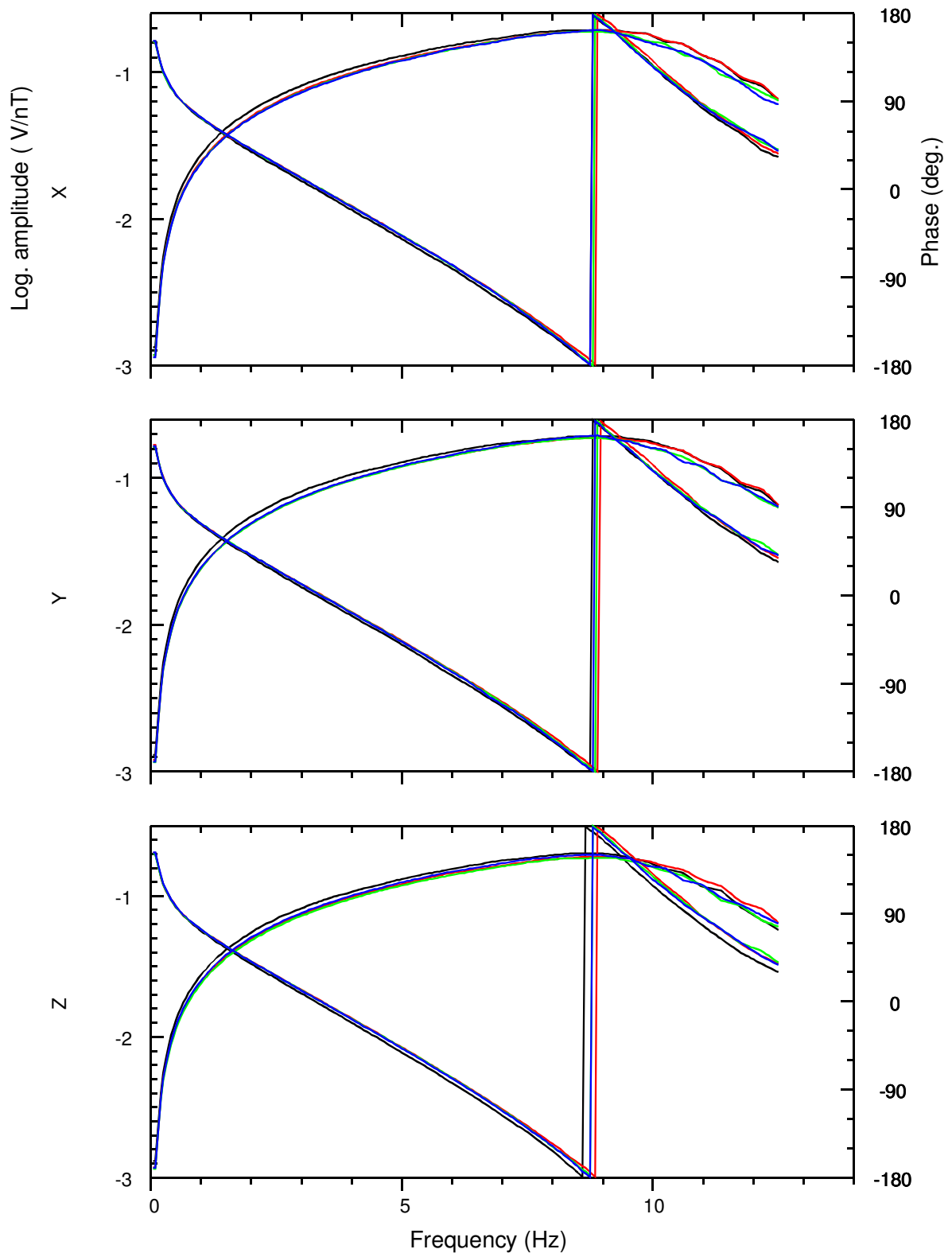
ANNEXE 1 : FONCTION DE TRANSFERT DES EXPERIENCES « ONDES »

Les figurent ci-après donnent un aperçu des fonctions de transfert des expériences « ondes » qui ont déjà pu être traitées par les procédures « *Wave_Roproc* ». Le format des fichiers correspondant ont été unifiés au format initial des tables de calibration de CLUSTER. Il est possible qu'un format plus général soit utilisé ultérieurement.

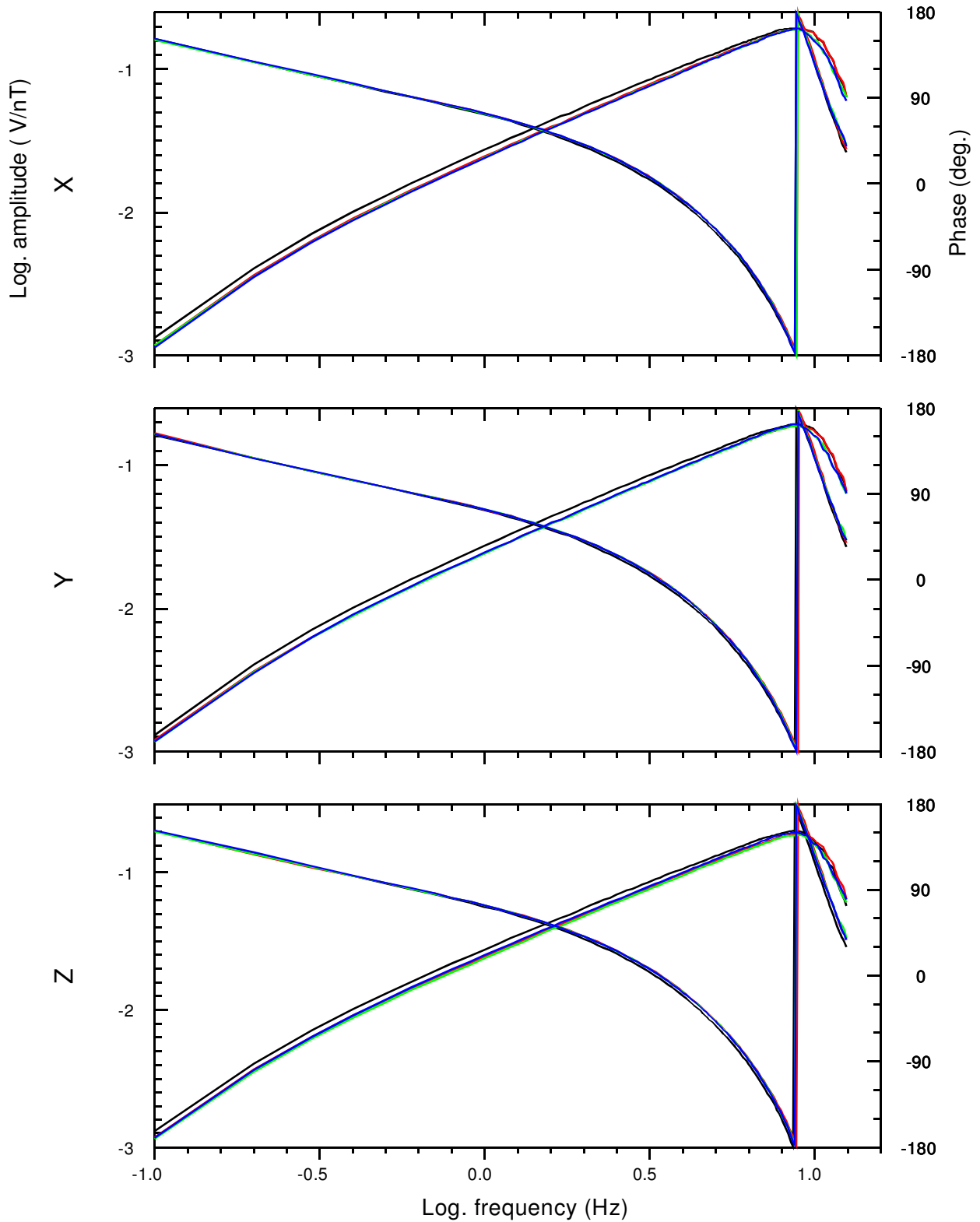
1) CLUSTER/STAFF-SC

Les figures ci-dessous donnent les fonctions de transfert des search_coils de l'expérience STAFF-SC dans les 2 gammes NBR et HBR, en amplitude et en phase, pour les 3 composantes des 4 satellites.

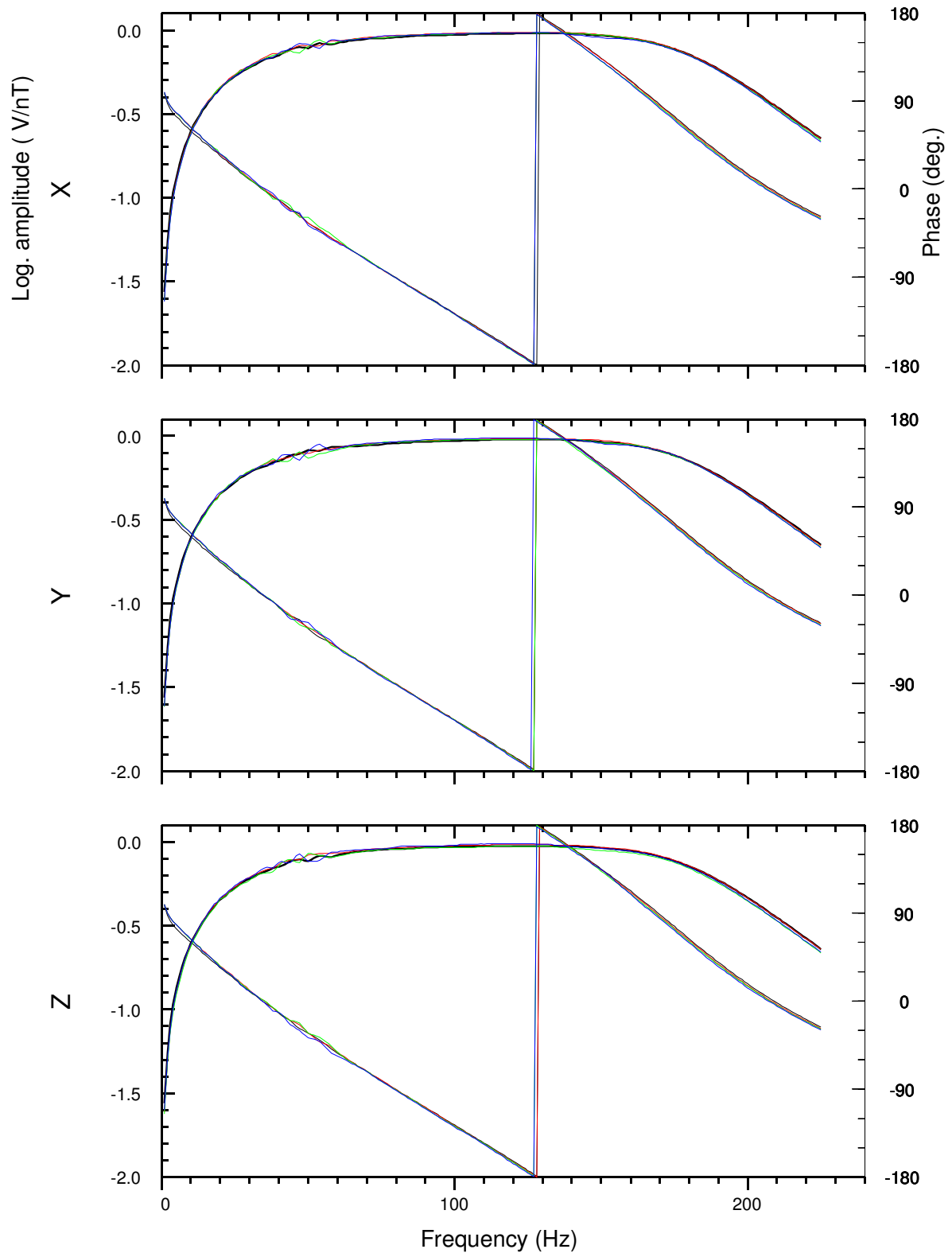
a) Partie NBR 0-12.5 Hz



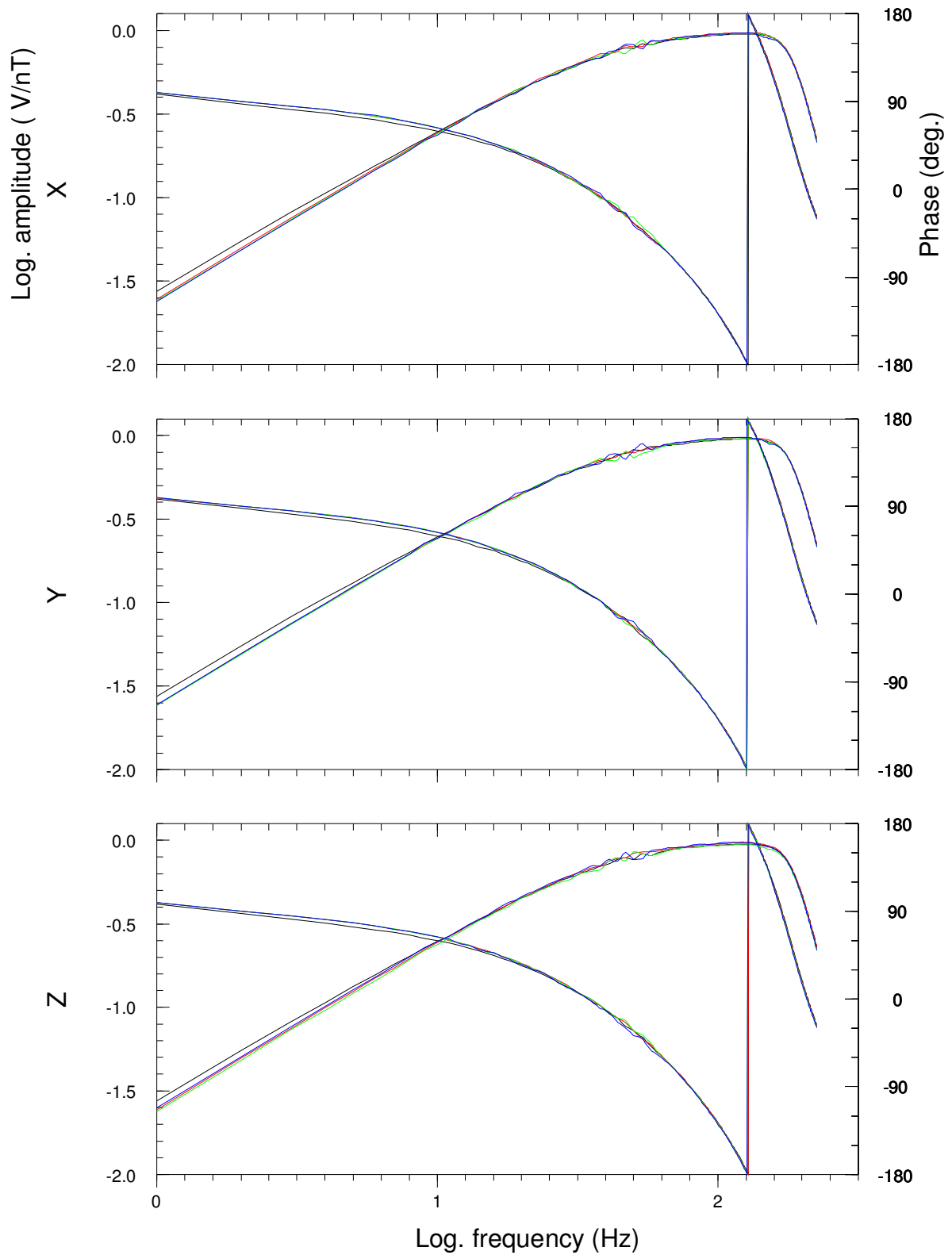
Même chose, mais en représentation log-log :



b) Partie HBR 0-12.5 Hz

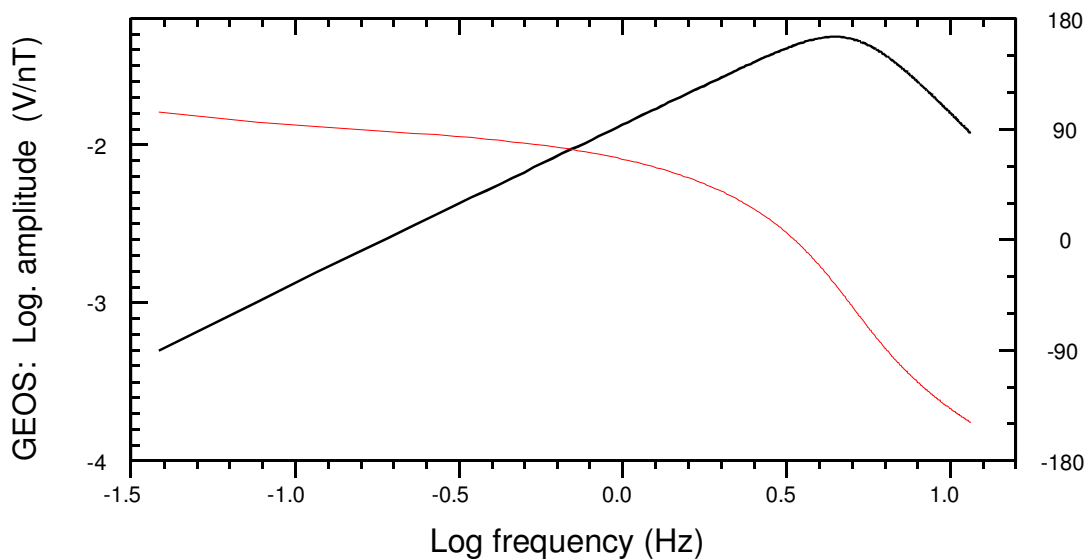
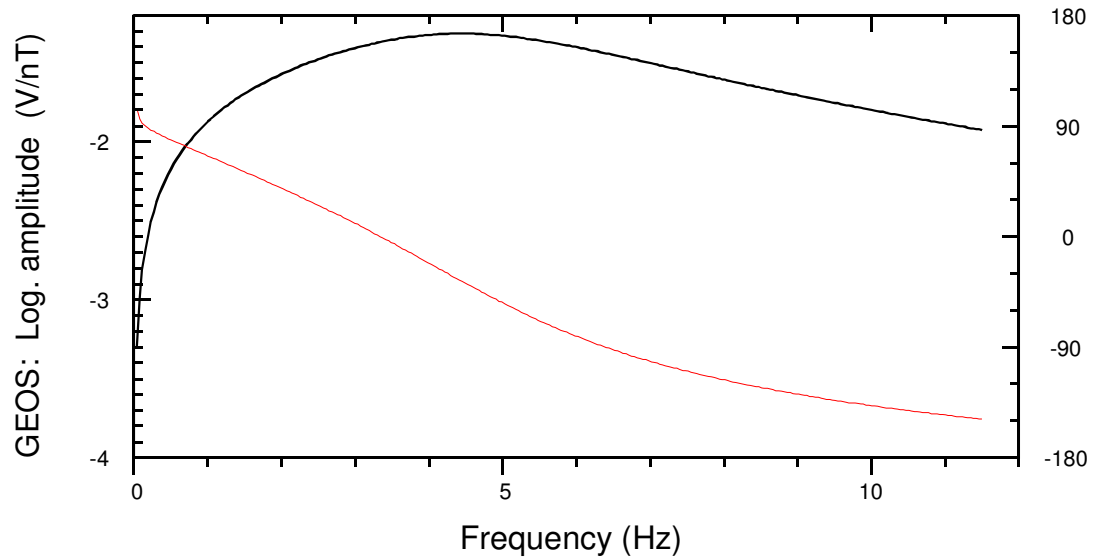


Même chose, mais en représentation log-log :



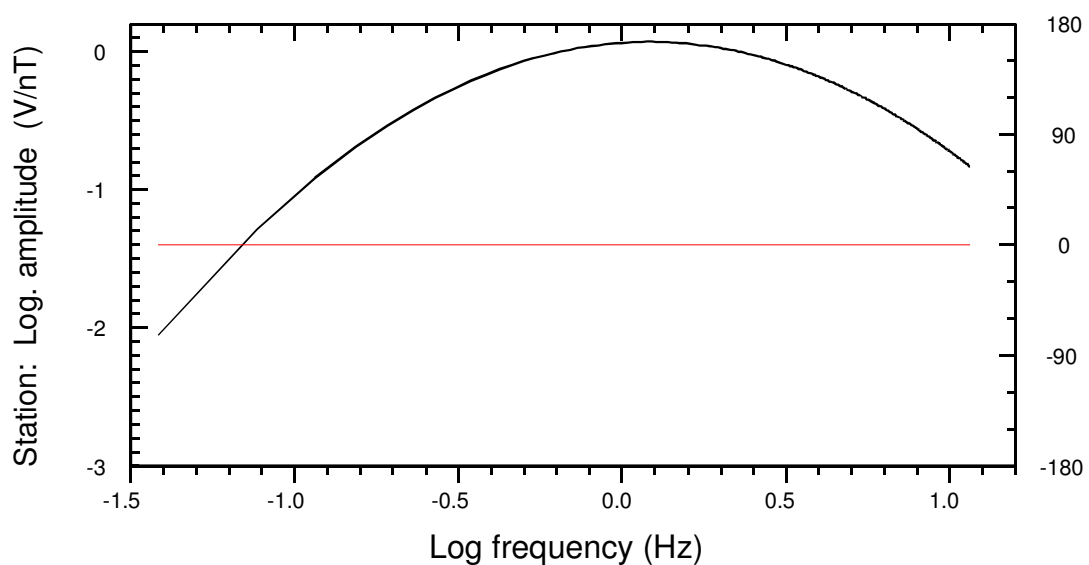
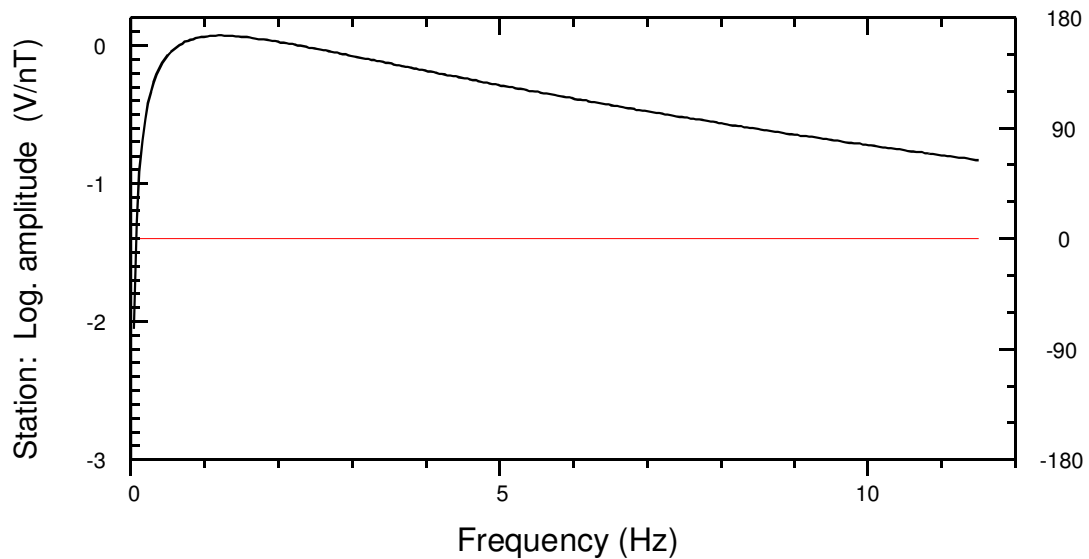
2) GEOS / UBF-S300

Les figures ci-dessous donnent les fonctions de transfert des antennes des search-coils de l'expérience S300 de GEOS-1 et GEOS-2, en amplitude et en phase. Les 3 composantes de chacun des 2 satellites ont la même fonction de transfert.



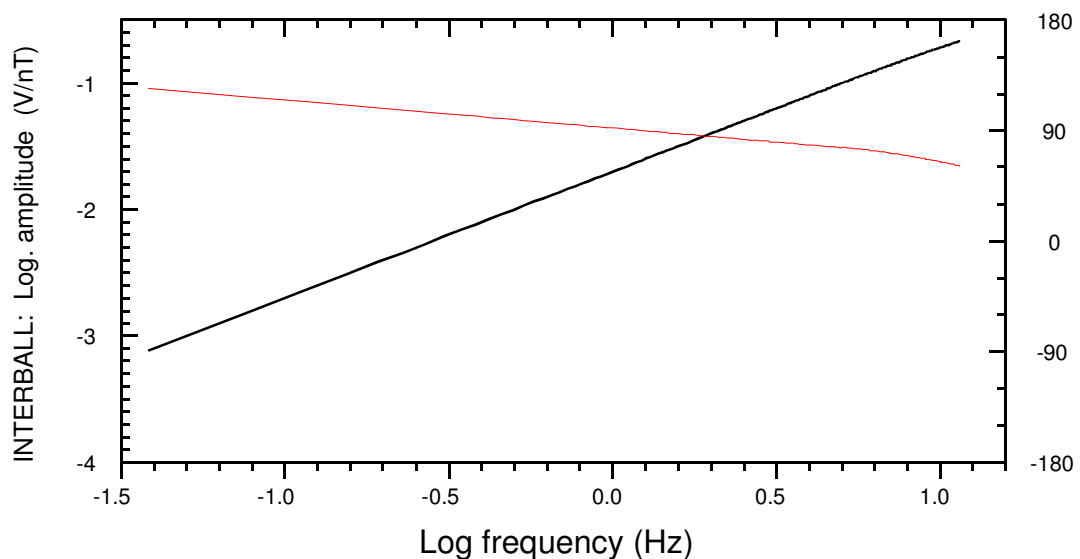
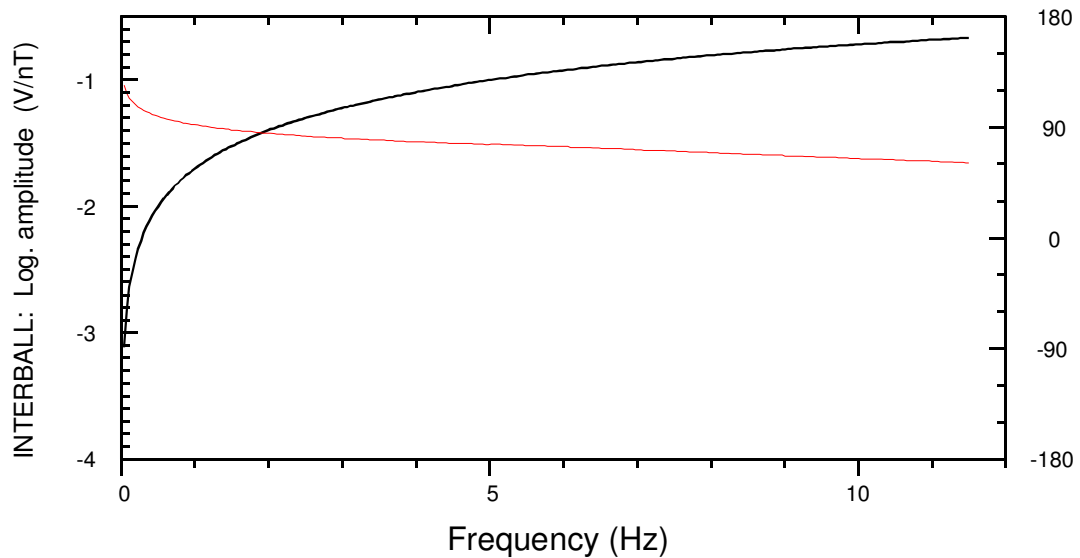
3) Station Mobile / Search-coils

Les figures ci-dessous donnent les fonctions de transfert des antennes des search-coils de la station mobile, positionnée à Husafell, Skiboten et Kitdalen, pendant les campagnes de mesures associées avec les satellites GEOS-1 et GEOS-2. Seule l'amplitude est connue, et la phase est mise arbitrairement à zéro.



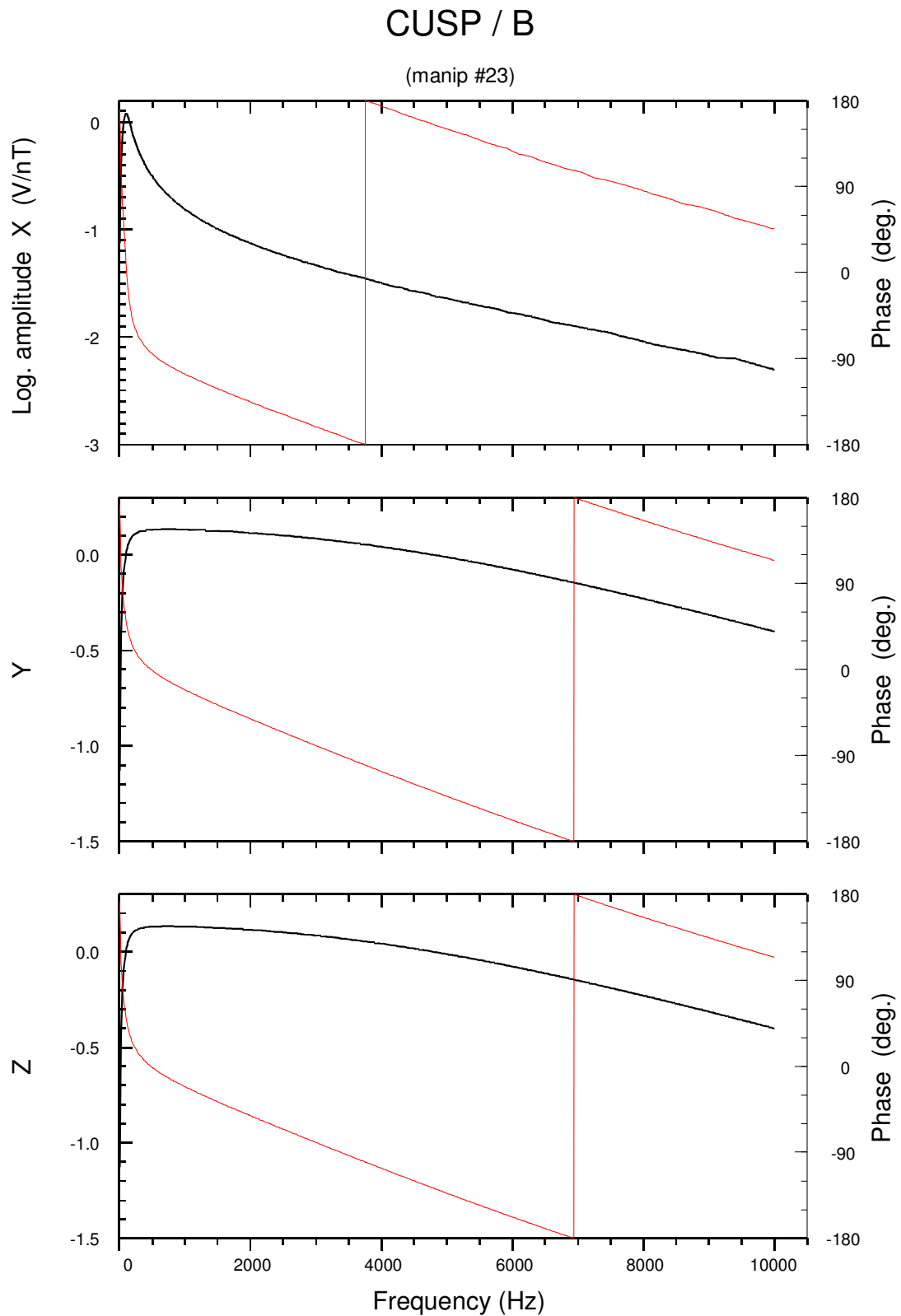
4) INTERBALL

Les figures ci-dessous donnent les fonctions de transfert des antennes des search-coils de l'expérience IESP, en amplitude et en phase. Les 3 composantes de chacun des 2 satellites ont la même fonction de transfert.

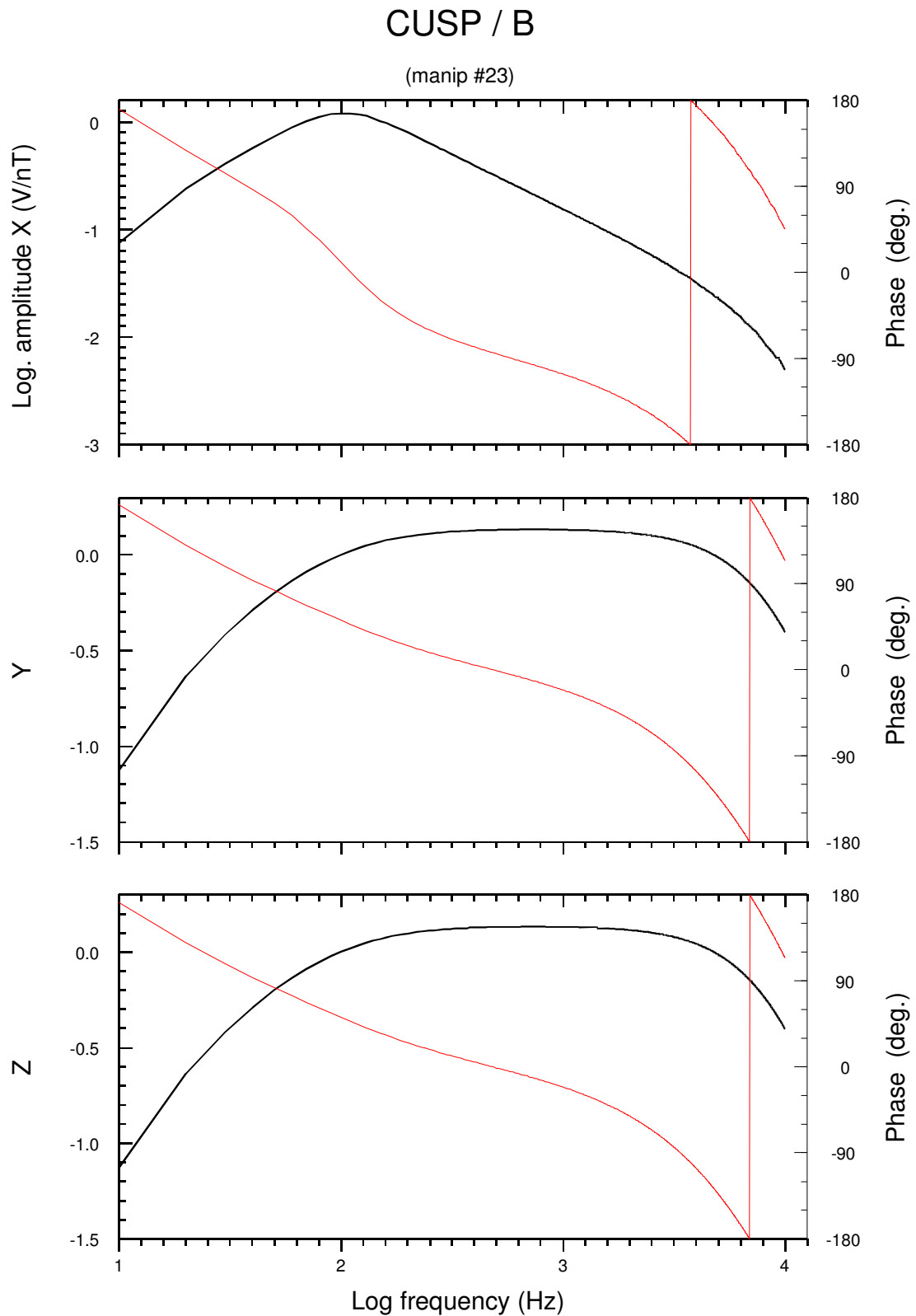


5) CUSP / B

La figure ci-dessous donne les fonctions de transfert, en amplitude et en phase, des 3 search-coils de l'expérience CUSP (tir de la fusée du 14 décembre 2002).

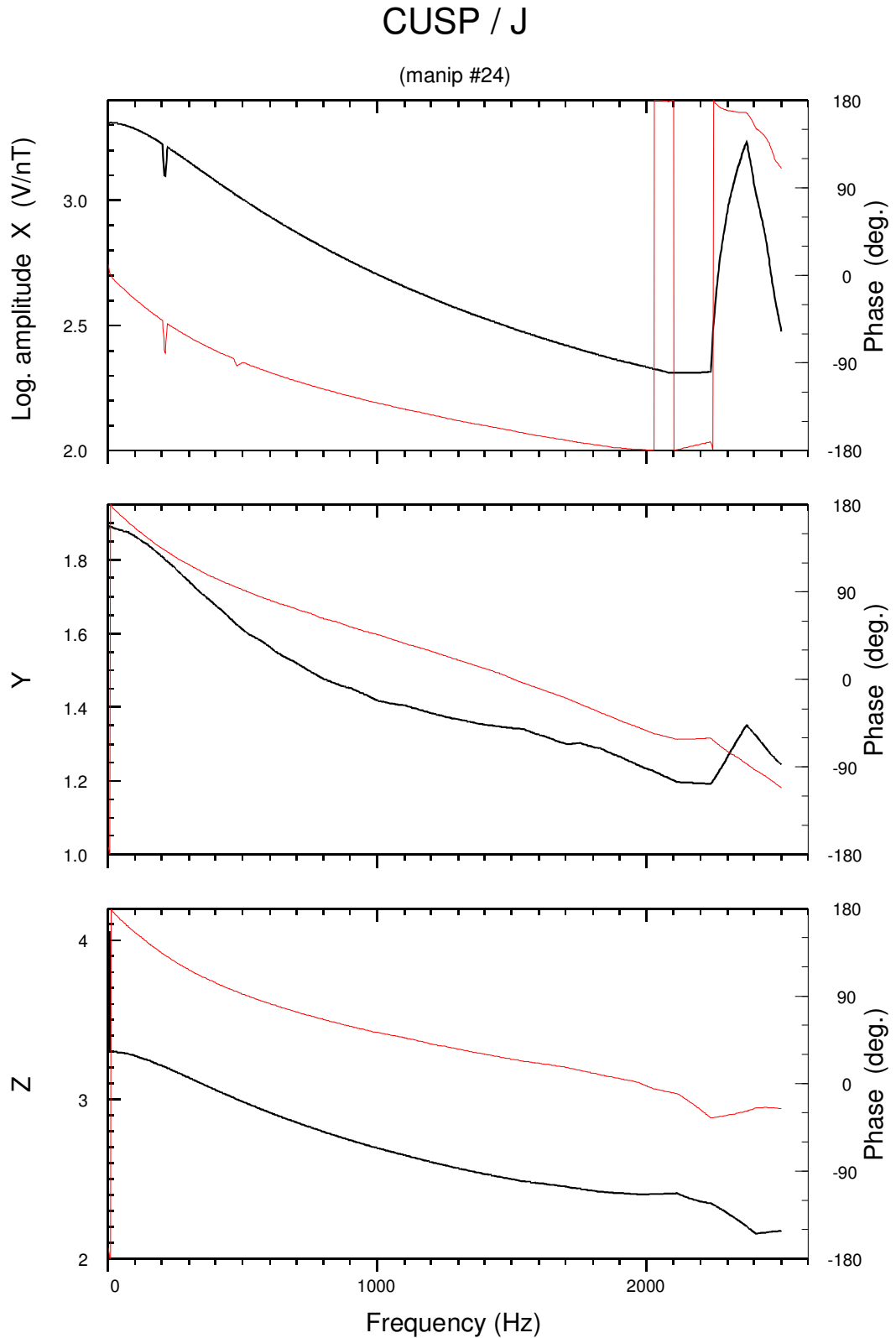


Même chose, mais en représentation log-log :

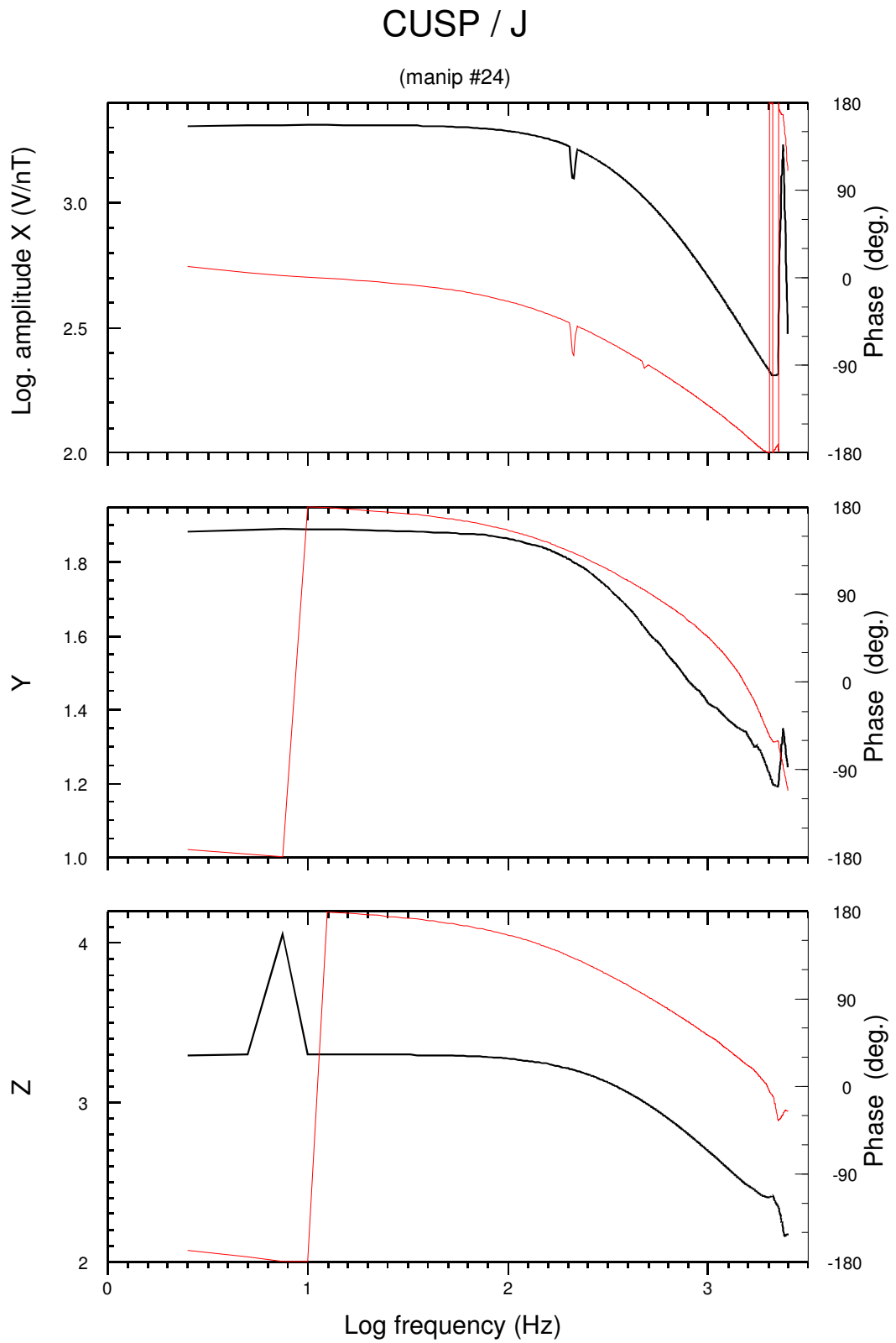


6) CUSP / J

La figure ci-dessous donne les fonctions de transfert, en amplitude et en phase, des 3 boucles de courant de l'expérience CUSP, (tir de la fusée du 14 décembre 2002).



Même chose, mais en représentation log-log :



ANNEXE 2 : DEFINITION DES DIFFERENTS REPERES UTILISES

La définition des différents repères utilisés est rappelée ci-dessous. Ces définitions et figures sont extraites des 2 documents suivants :

CLUSTER SOFTWARE TOOLS, PART I : COORDINATE TRANSFORMATION LIBRARY

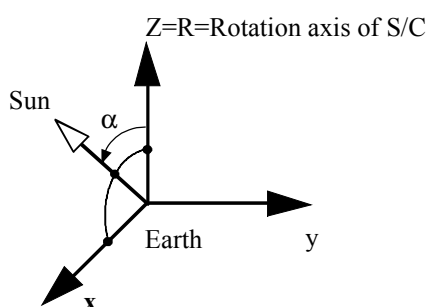
Document de travail DT/CRPE/1231 Par Patrick Robert, CRPE/TID, Juillet 1993.

CLUSTER DATA PROCESSING

Transformation of STAFF waveform into a Magnetic Field Aligned coordinate system

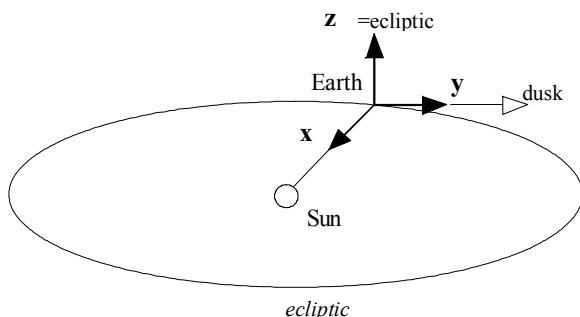
Rapport interne RI-CETP/6/2000 Par Patrick Robert & Claude Devilledary, version 1.6, Octobre 2000

1) Repère SR2



The Spin Reference #2 coordinate system (SR2), also called Despun Satellite system (DS), is a system where Z is along the spin axis, and X in the plane defined by the spin axis and the direction of the Sun.

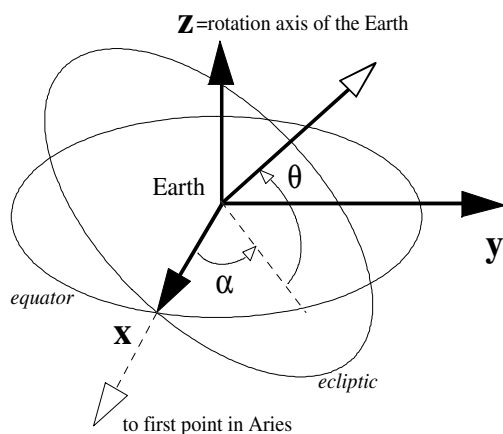
2) Repère GSE



The Geocentric Solar Ecliptic system (GSE) has the X axis toward the Sun, and the Y axis in the ecliptic plane, to the dusk, opposing to the planetary motion. The Z-axis is parallel to the ecliptic pole.

The GSE system has a yearly rotation with respect to the inertial system.

3) Repère GEI

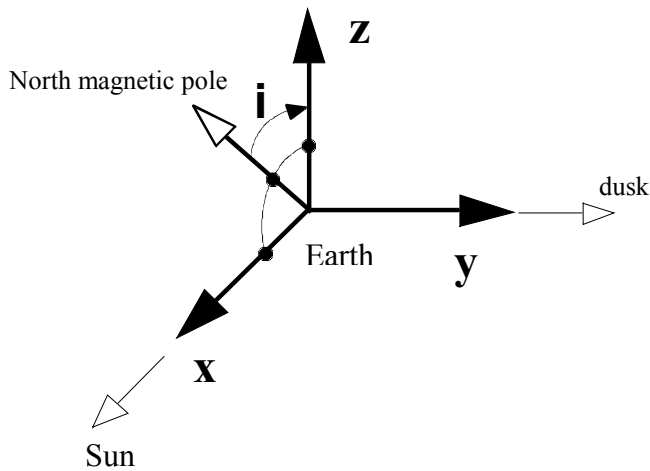


The Geocentric Ecliptic Inertial system (GEI) has his Z-axis parallel to the rotation axis of the Earth, and the X-axis defined by the intersection of the equator plane and the ecliptic plane, and is pointing towards the first point of Aries (Sun position at the vernal equinox).

right ascension $\alpha = \tan^{-1}(V_y/V_x)$
with α in $[0^\circ, 180^\circ]$ for $V_y > 0$
 α in $[180^\circ, 360^\circ]$ for $V_y < 0$

declination $\theta = \sin^{-1}(V_z/V)$
with θ in $[-90^\circ, 90^\circ]$

4) Repère GSM

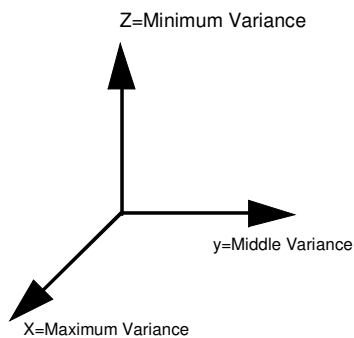


The Geocentric Solar Magnetospheric system (GSM) has the X-axis pointing from the Earth towards the Sun, while the X-Z plane contains the dipole axis.

The Y-axis is perpendicular to the Earth's magnetic dipole, towards the dusk and include in the magnetic equator plane.

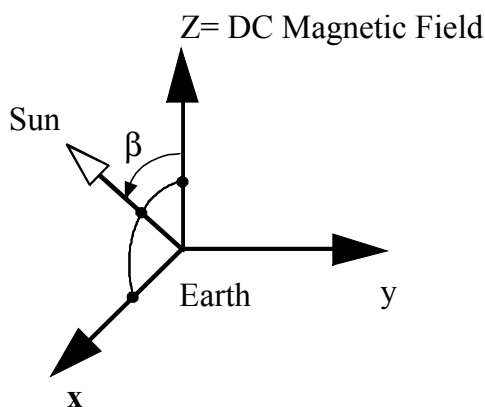
The positive Z-axis is chosen to be in the same sense as the northern magnetic pole; the dipole tilt angle i is positive when the north magnetic pole is tilted towards the Sun. In addition to a yearly period due to the motion of the Earth about the Sun, the GSM system rocks about the Solar direction with a 24 h period.

5) Repère MVA



The Minimum Variance Analysis system (MVA) is a system where the variance of the signal is minimum in the plane perpendicular to the Z axis, and maximum in the plane perpendicular to the X axis.

6) Repère MFA



The Magnetic Field Aligned system (MFA) is defined as a system where Z is along the DC magnetic field, and X in the plane containing the DC magnetic field vector and the direction of the Sun (see reference above).

ANNEXE 3 : UTILISATION SUR LA MACHINE SAWARA DU CETP

Les **Roproc** ont été installées sur la machine **sawara** du CETP spécialement pour le traitement des données CLUSTER (STAFF-SC, FGM, Orbit, et en partie EFW). Toutes les procédures disponibles, résumées dans l'annexe 5, sont accessibles à partir de n'importe quelle arborescence sous cette machine, en particulier dans un directory sous /data2, à condition que votre fichier .cshrc contiennent les lignes suivantes :

```
#-----  
# extensions du path pour utiliser les Cluster's Roproc  
#-----  
  
# extension du path pour utiliser CLUSTER/Wave_Roproc_Staff:  
set path=(/users/ida/robert/cluster/staff/Wave_Roproc_Staff_IDL $path)  
  
# extension du path pour utiliser CLUSTER/Wave_Roproc_Efw:  
set path=(/users/ida/robert/cluster/efw/Wave_Roproc_Efw $path)  
  
# extension du path pour utiliser CLUSTER/Vector_Roproc_Fgm:  
set path=(/users/ida/robert/cluster/fgm/Vector_Roproc_Fgm $path)  
  
# extension du path pour utiliser CLUSTER/Orbit_Roproc_Cluster:  
set path=(/users/ida/robert/cluster/orbit/Orbit_Roproc_Cluster $path)  
  
#-----  
# variables d'environnement  
#-----  
  
# lancement du logiciel graphique des Roproc (IDL ou PV-WAVE)  
  
setenv GRASOFT $IDL_DIR/bin/idl  
  
# path des procedures Roproc  
  
setenv DIR_ROPROC /users/ida/robert/cluster/staff/Wave_Roproc_Staff_IDL  
  
#-----
```

Alors, dans chaque fenêtre que vous ouvrirez, vous aurez accès au menu général de chacun des trois groupes en lançant les commandes :

rsc_menu	(Roproc_Cluster_Staff menu)	pour les procédures	Wave_Roproc_STAFF
rce_menu	(Roproc_Cluster_Efw menu)	pour les procédures	Wave_Roproc_EFW
rcf_menu	(Roproc_Cluster_Fgm menu)	pour les procédures	Vector_Roproc_FGM
rco_menu	(Roproc_Cluster_Orbit menu)	pour les procédures	Orbit_Roproc_CLUSTER

Ces menus sont donnés dans l'annexe 5. Ils résument l'ensemble des procédures disponibles au fur et à mesure qu'elles sont développées.

ANNEXE 4 : INSTALLATION SUR UNE MACHINE SUN / SOLARIS

1) Installation

Pour chaque famille, le logiciel est livré sous la forme d'un fichier archive de la forme :

Wave_Roproc_Staff_IDL_v1p8.tar

L'installation consiste en les étapes suivantes:

- a) Copier le fichier *Wave_Roproc_Staff_IDL_v1p8.tar*, version pour SUN/Solaris, dans le directory où on souhaite installer le logiciel, par exemple *Roproc*, qui peut ne pas être vide.

- b) Créer l'arborescence des répertoires et fichiers par :

tar xvf Wave_Roproc_Staff_IDL_v1p8.tar

on obtient alors, dans le répertoire *Roproc*, le nouveau répertoire *Wave_Roproc_Staff* contenant tous les fichiers exécutable, d'exemple, etc.

- c) Faire les modifications du fichier *.cshrc* situé dans le home-directory, en incluant les lignes mentionnées dans le § précédent.

- d) Ouvrir une nouvelle fenêtre, qui prendra en compte ces modifications, ou exécuter la commande :

source \$HOME/.cshrc

- e) Se placer dans un directory quelconque, de préférence vide.

- f) Exécuter la commande

racs_menu

- g) Si l'installation est correcte et le fichier *.cshrc* mis à jour, le contenu du menu *racs* doit s'afficher.

L'installation est terminée. Toutes les commandes *racs* indiquées dans le menu sont maintenant disponibles, à partir de n'importe quel répertoire. Les premières utilisations sont expliquées au § suivant.

2) Utilisation

Le logiciel étant installé, on montre ci-dessous comment procéder pour calculer et tracer un spectrogramme, par exemple :

a) Se placer de préférence dans un directory vide.

b) Exécuter tout d'abord la commande :

rcs_copy_in

pour créer un exemple des fichiers de paramètres *.in

c) Editer le fichier *cowave.in*, modifier les paramètres à votre convenance ; ne pas oublier de mentionner le bon chemin pour le fichier N1 utilisé, ainsi que le chemin des fichiers de gains d'antenne. Régler les paramètres durée, nombre de points, etc. Se reporter au § II-2-b pour voir le détail de chaque paramètre.

d) Exécuter la commande :

rcs_cowave

les fichiers *cowave.out* et *cowave.resu* sont créés ; éventuellement, examiner le fichier *cowave.out* pour vérifier que la procédure s'est correctement exécutée.

e) Editer et modifier éventuellement le fichier *cospectro.in* (voir § II-4-b).

f) Exécuter la commande :

rcs_cospectro

les fichiers *cospectro.out* et *cospectro.resu* sont créés ; éventuellement, examiner le fichier *cospectro.out* pour vérifier que la procédure s'est correctement exécutée.

g) Exécuter la commande :

rcs_visuspectro

et on obtient le fichier PostScript *visuspectro.ps*

Toutes ces commandes sont bien sur « scriptable », c'est à dire que l'on peut les enchaîner dans un fichier de commandes pour se créer des « macro-commandes » (voir § I-4).

ANNEXE 5 : RESUME DES PROCEDURES DISPONIBLES POUR CLUSTER

On ne mentionne ici que les procédures installées sur la machine **sawara** du CETP, et relatives à CLUSTER.

1) *Wave_Roproc_Staff*

La commande *rsc_menu* (*Roproc_Cluster_Staff Menu*) donne directement le résumé des commandes disponibles, et génère le texte suivant ci-après ; il est susceptible d'évoluer avec le temps, d'autres commandes pouvant être développées.

```
=====
Les procedures Wave_Roproc pour la manip CLUSTER/STAFF-SC:

1) programmes de base:
-----

rsc_cowave      : calibre les donnees , cre cowave.resu (<.in, >.out)
rsc_cowave4     : idem sur les 4 satellites
rsc_readwave    : utilitaire PV-Waves ou IDL pour relire un cowave.resu
rsc_visuwave    : lit cowave.resu et cre un .ps par sequence
rsc_visuwave4   : idem sur les 4 satellites, un .ps par sat.

2) filtres pour cowave.resu:
-----

rsc_wave_filter      : filtre cowave.resu (<.in, >.out)
rsc_wave_filter4     : idem sur 4 sat. (avec cowave4)
rsc_wave_to_gse      : met cowave.resu dans le repere gse (>.out)
rsc_wave_to_gse4     : idem sur 4 sat. (avec cowave4)
rsc_wave_to_gsm      : met cowave.resu dans le repere gsm (>.out)
rsc_wave_to_gsm4     : idem sur 4 sat. (avec cowave4)
rsc_wave_to_mfa      : met cowave.resu dans le repere mfa (>.out)
rsc_wave_to_mva4     : idem sur 4 sat. (avec cowave4)
rsc_wave_to_mva      : met cowave.resu dans le repere mva (>.out)
rsc_wave_to_mva4     : idem sur 4 sat. (avec cowave4)

3) programmes de niveau 2:
-----

rsc_cospectro      : lit cowave.resu et cre cospectro.resu (<.in, >.out)
rsc_cospectro4     : idem 4 satellites (4 cospectro_N.resu) (<.in, >.out)

rsc_visuspectro    : lit cospectro.resu et cre visuspectro.ps
rsc_visuspectro4   : idem 4 satellites, cre 4 visuspectro_N.ps
rsc_visuspectro_4Bz: visualise les 4 cospectro_N.resu; cre visucospectro_4Bz.ps

rsc_visuspectra    : lit cospectro.resu et cre visuspectra.ps
rsc_visuspectra4   : idem 4 satellites, cre 4 visuspectra_N.ps

rsc_visupower      : lit cospectro.resu et cre visupower.ps (<.in)
rsc_visupower4     : idem 4 satellites, cre 4 visupower_N.ps (<.in, >.out)

rsc_copolar        : lit cowave.resu et cre copolar.resu ( >.out)
rsc_copolar4       : idem 4 satellites (4 copolar_N.resu) ( >.out)
```


2) Wave_Roproc_Efw

La commande *rce_menu* (*Roproc_Cluster_Efw Menu*) donne le résumé des commandes disponibles pour l'expérience EFW de CLUSTER, et génère le texte suivant ci-après. Mais pour EFW, il n'y a pas de base de fichier existante. La procédure *rce_cowave* lit un flat file créé par ISDAT, et le transforme au format *cowave.resu*, ce qui permet d'utiliser les procédures *Wave_Roproc* sur ces données, et faire par exemple des spectres et spectrogrammes. La création et la gestion des flat files est à la charge de l'utilisateur.

```

Les procedures Wave_Roproc pour la manip CLUSTER/EFW:

1) programmes de base:
-----
rce_cowave      : calibre les donnees , cre cowave.resu (<.in, >.out)
rce_readwave    : utilitaire PV-Waves ou IDL pour relire un cowave.resu
rce_visuwave     : lit cowave.resu et cre un .ps par sequence

2) filtres pour cowave.resu:
-----
rce_wave_filter : filtre cowave.resu (<.in, >.out)

3) programmes de niveau 2:
-----
rce_cospectro   : lit cowave.resu et cre cospectro.resu (<.in, >.out)
rce_visuspectro : lit cospectro.resu et cre visuspectro.ps
rce_visuspectra : lit cospectro.resu et cre visuspectra.ps
rce_visupower   : lit cospectro.resu et cre visupower.ps (<.in)

4) utilitaires divers:
-----
rce_copy_in     : copy dans le dir. courant un exemple des .in
rce_readme      : liste le fichier README.txt donnant la procedure d'installation

5) pense bete:
-----

Pour cowave, les valeurs possibles de nbp et leur equivalent
en temps sont donnees ci dessous:
( * -> valeur recommandee pour les spectrogrammes)

```

Nb. de pts	Duree en NBR (25 Hz)		
128	5.12	s *	
256	10.24	s *	
512	20.48	s *	
1024	40.96	s *	
2048	81.92	s soit	1 m 22 s
4096	163.8	s soit	2 m 44 s
8192	327.7	s soit	5 m 28 s
16384	655.4	s soit	10 m 55 s
32768	1310.7	s soit	21 m 51 s
65536	2621.4	s soit	43 m 41 s

```
Wave_Roproc pour CLUSTER/EFW:
  version initiale: 1.1 au 21 Aout      2002
  derniere version : 1.2 au 20 Septembre 2002

En cas de problemes, n'hesitez pas a me contacter:
Patrick Robert, CETP, tel: 01 39 25 49 09, Patrick.Robert@cetp.ipsl.fr
```

3) *Vector_Roproc_Fgm*

La commande *rcf_menu* (*Roproc_Cluster_Fgm Menu*) donne le résumé des commandes disponibles, et génère le texte suivant ci-après ; il est susceptible d'évoluer avec le temps, d'autres commandes pouvant être développées.

```
=====
Les procedures Vector_Roproc pour CLUSTER/FGM:
```

```
1) Visualisation des fichiers FGM format PPD (aucun traitement)
-----
```

```
rcf_visuveciso      : lit visuveciso.in et cre un visuveciso.ps (1 sat.)
rcf_visuveciso4     : lit visuveciso.in et cre 4 visuveciso_N.ps
rcf_visuveciso_4sat : lit visuveciso_4sat.in et genere un .ps pour les 4 sat.
```

```
rcf_visuveciso_arg.bat      : les parametres sont en argument, permet d'ecrire
                             des procedures automatique pour la generation
                             en serie de PostScript.
rcf_visuveciso_4sat_arg.bat : idem pour les visus a 4 sat.
```

```
2) Programmes de base:
-----
```

```
rcf_covector:  lit un fichiers veciso.dat et cre covector.resu (<.in, >.out)
               le fichier veciso.dat peut etre un PPD ou un High resolution.
               Le format covector.resu est un ascii fixe standard Vector_Roproc.
```

```
rcf_covector4: idem 4 sat.
```

```
rcf_visuvector      : lit visuvector.resu et cre un visuvector.ps (1 sat.)
rcf_visuvector4     : lit 4 visuvector_N.resu et cre 4 visuvector_N.ps
rcf_visuvector_4sat : lit 4 visuvector_N.resu et genere un .ps pour les 4 sat.
```

```
3) filtres pour covector.resu:
-----
```

```
rcf_vector_gse_to_gsm : met covector.resu dans le repere gsm (>.out)
rcf_vector_gse_to_gsm4 : idem 4 sat.
```

```
rcf_vector_gse_to_sr2 : met covector.resu dans le repere sr2 (<.in,>.out)
rcf_vector_gse_to_sr24 : idem 4sat
```

```
rcf_vector_to_mva      : met covector.resu dans le repere mva (>.out)
rcf_vector_to_mva4     : idem 4 sat.
```

```
TBD* rcf_vector_filter : filtre covector.resu (<.in, >.out)
TBD* rcf_vector_to_mfa : met covector.resu dans le repere mfa (>.out)
```

```
rcf_alivectime      : modifie 4 fichiers covector_N.resu pour aligner le temps
rcf_addvecpos       : ajoute les position du sat. a un fichier covector
rcf_cocurldiv       : calcule curB et divB partir des fichiers alignes
rcf_cocurldiv_total : enchaines toutes les procedures pour cocurldiv
rcf_cocurldiv_lprt0t : imprime tous les fichiers utiles du directory donne
```

```
rcf_vector_to_wave: met covector.resu au format cowave.resu (<.in, >.out)
                    permet d'utiliser toutes les procedures Wave_Roproc
                    de Cluster/Staff, en particulier :
```

```
rscs_cospectro
rscs_cospectro4

rscs_visuspectro
rscs_visuspectro4

rscs_visuspectra
rscs_visuspectra4

rscs_visupower
rscs_visupower4
```

```
rcf_merge_alivector: prends 4 fichiers covector.resu a 3 composantes et cre
                    un seul fichier covector a 12 composantes + le temps
```

5) utilitaires divers:

```
rcf_copy_in  : copy dans le dir. courant un exemple des .in
rcf_readme   : liste le fichier README.txt donnant la procedure d'installation
```

TBD* = To Be Defined: en cours de realisation, pas encore disponible

Vector_Roproc pour CLUSTER/FGM:

version	initiale:	1.1	au	15	Novembre	2001	(pour CLUSTER/FGM))
	version :	1.2	au	28	Mai	2002	
	version :	1.3	au	2	Octobre	2002	(generalisation multi-experiences)
	version :	1.4	au	10	Fevrier	2003	(vector_to_wave conversion)
derniere	version :	1.5	au	20	Mai	2003	(merge_alivector conversion)

En cas de problemes, n'hesitez pas a me contacter:

Patrick Robert, CETP, tel: 01 39 25 49 09, Patrick.Robert@cetp.ipsl.fr

4) *Orbit_Roproc_Cluster*

La commande *rco_menu* (*Roproc_Cluster_Orbit Menu*) donne le résumé des commandes disponibles, et génère le texte suivant ci-après ; il est susceptible d'évoluer avec le temps, d'autres commandes pouvant être développées.

```
=====
Les procedures Orbit_Roproc_Cluster:

1) programmes de base:
-----

rco_listorbit   : cre un fichier d'orbite pour 1 journee, avec la postion
                  et eventuellement la geometrie du tetraedre (<.in, >.out)
rco_visuorbit   : visualise les donnees de listorbit.resu (cre un .ps)


rco_cresatpos   : cre un fichier des positions pour une periode de temps
                  quelconque, inferieure ou superieure a 1 jour (<.in, >.out)
rco_visusatpos  : visualise en projection 3D les donnees de cresatpos.resu
                  (cre un .ps)


crevisusatpos_arg.bat: automatisme pour production en serie, ex:
crevisusatpos_arg.bat 24 09 2001 00 10 00    5.0 0.1 gse


2) utilitaires"
-----

rco_copy_in     : copy dans le dir. courant un exemple des .in
rco_readme      : liste le fichier README.txt donnant la procedure d'installation



^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Orbit_Roproc pour CLUSTER:

version initiale: 1.1 au   4 Decembre 2001
                 version  : 1.2 au 28 Mai       2002
derniere version : 1.3 au 28 Aout        2002


En cas de problemes, n'hésitez pas à me contacter:

Patrick Robert, CETP, tel: 01 39 25 49 09, Patrick.Robert@cetp.ipsl.fr
```

BIBLIOGRAPHIE

- [1] Patrick **Robert**, “CLUSTER SOFTWARE TOOLS, PART I: COORDINATE TRANSFORMATION LIBRARY”, Document de travail DT/CRPE/1231, Juillet 1993.
- [2] Patrick **Robert** & Claude Devilledary, “CLUSTER DATA PROCESSING, Transformation of STAFF waveform into a Magnetic Field Aligned coordinate system, version 1.6”, Rapport interne RI-CETP/6/2000, Octobre 2000
- [3] Patrick **Robert**, “ROCOTLIB: a Coordinate Transformation Library for Solar-Terrestrial studies, version 1.7”, Rapport Interne RI-CETP/02/2003, Janvier 2003.

Code source, exemple d'utilisation et document disponible au CDP, <http://cdpp.cesr.fr>

- [4] Gérard **Chanteur**, Geometrical tools for Cluster data analysis, ESA WPP-047, Proceedings of the International Conference on Spatio-Temporal Analysis for Resolving Plasma Turbulence (START), Aussois, Jan. 31-Febr. 5, 1993, pp. 341-344, 1993.

———— NOTES ————