Analyse spectrale de mesures de champ magnétique turbulent dans l'espace interplanétaire

Responsables : Olga Alexandrova (olga.alexandrova@obspm.fr)
Baptiste Cecconi (baptiste.cecconi@obspm.fr)

1 Introduction

Lorsque l'on analyse un signal expérimental $u(t)$ mesuré en fonction du temps t par une sonde spatiale, il est important de connaître la représentation de ce signal dans l'espace des fréquences. La transformée de Fourier nous donne cette information. D'habitude on utilise une puissance spectrale (ou *power spectral density, PSD*, i.e. les carrés des amplitudes des coefficients de Fourier) pour caractériser la distribution de l'énergie du signal en fréquence. Le reste de l'information est contenu dans les phases de Fourier (comme le décalage temporel des différents modes). Les phases ne sont pas faciles à manipuler. Pour récupérer cette information sous la forme la plus simple, on a besoin d'une méthode qui fait l'analyse spectrale dans des fenêtres localisées dans le temps.

En plus, dans la réalité, un signal de longueur T peut être très inhomogène, correspondant à la traversée de régions très différentes (par exemple, lorsque le satellite traverse des frontières de la magnétosphère, des événements énergétiques dans le vent solaire comme une éjection de masse coronale, ou dans le cas de la turbulence intermittente). Donc il est important de savoir comment le spectre en fréquence varie dans le temps. Dans ce but, deux classes de transformations sont habituellement utilisées : (i) la transformée de Fourier avec fenêtre glissante et (ii) les transformées d'ondelettes.

Il est clair que la résolution en fréquence Δf (ou en échelle temporelle $\Delta\tau = 1/\Delta f$) et la résolution temporelle ΔT ne peuvent être petites simultanément mais sont reliées par le principe d'incertitude

$$\Delta f \Delta T \sim const. \tag{1}$$

Dans les transformées d'ondelettes les fenêtres d'analyse sont spécialement adaptées aux échelles. Dans la transformée de Fourier avec fenêtre glissante la résolution temporelle diminue quand la fréquence augmente.

1.1 Transformée de Fourier discrète

Les données réelles sont des séries temporelles discrètes $u[j] = u(t_j) = u(t_0 + j\Delta t)$ de N mesures pendant le temps d'enregistrement T avec une résolution $\Delta t = T/N$, et $t_j = j\Delta t$, $j = 0, 1, 2, ..., N-1$.

On définit la transformée de Fourier (TF) discrète comme la convolution du signal avec une base de fonctions orthonormées $\sim e^{i\omega t}$ (avec $\omega = 2\pi f$). Il existe différentes notations pour la TF, nous allons utiliser ici les notations d'*IDL* :

$$\hat{u}[n] = \frac{1}{N}\sum_{j=0}^{N-1} u[j] e^{-2\pi i \frac{nj}{N}} \tag{2}$$

où n est un indice des fréquences et j est un indice des points de mesure temporelle.

— Ecrire l'expression de TF inverse.

Pour un signal réel, l'équation (2) nous dit que les fréquences négatives ne donnent pas une information supplémentaire.

— Montrer que si $u(t)$ est réel,
$$\hat{u}[N-n] = (\hat{u}[n])^* \tag{3}$$

— Cette transformation conserve de l'énergie. Démontrer le Théorème de Parseval :
$$\frac{1}{N}\sum_{j=0}^{N-1} |u[j]|^2 = \sum_{n=0}^{N-1} |\hat{u}[n]|^2 \tag{4}$$

La distribution de l'énergie dans le signal en fréquence est donnée par la densité d'énergie spectrale (ou *power spectral density, PSD*), définie comme :

$$S[n] = 2T|\hat{u}[n]|^2, \;\; n = 1, 2, ..., N/2 \tag{5}$$
$$f_n = n/T, \;\; n = 1, 2, ..., N/2 \tag{6}$$

pour N égale à un nombre pair.

— Donner les unités du $S[n]$.

— Ecrire le Théorème de Parseval en utilisant $S[n]$.

1.2 Transformée d'ondelette de Morlet

L'ondelette de Morlet [Farge, 1992 ; Torrence & Compo 1998] est une onde de fréquence ω_0 modulée par une gaussienne :

$$\psi_0(t) = \pi^{-1/4} e^{-i\omega_0 t} e^{-t^2/2} \tag{7}$$

où t est le temps. La transformée en ondelettes du signal $u(t_j)$ est la convolution de ce signal avec la fonction $\psi_0(t_j)$ dilatée et translatée

$$\mathcal{W}(\tau, t) = \sum_{j=0}^{N-1} u(t_j)\psi^*[(t_j - t)/\tau] \tag{8}$$

où l'astérisque * indique le complexe conjugué, t_j est la position (un paramètre de translation) autour de laquelle on effectue la convolution dans une fenêtre Δt ; et τ est un paramètre de dilatation ou échelle temporelle. Le choix des échelles pour l'analyse en ondelettes non-orthogonales, comme celles de Morlet, est arbitraire. Il est pratique de définir les échelles comme une puissance fractionnaire de deux :

$$\tau_m = \tau_0 2^{m \cdot \delta m}, \;\; m = 0, 1, ..., M \tag{9}$$
$$M = \frac{1}{\delta m}\log_2(N\delta t/\tau_0) \tag{10}$$

où τ_0 est l'échelle la plus petite et M détermine l'échelle la plus grande, τ_M. Au total, il y a $M{+}1$ échelles temporelles. Il est pratique de choisir $\tau_0 = 2\delta t$ pour que l'échelle temporelle τ soit facilement transformable en fréquence physique (Fourier). Ici δt est la résolution temporelle des mesures de $u(t)$. La résolution en échelles de la transformation est déterminée par δm : plus δm est petit, mieux les échelles sont résolues. Ici on utilise $\delta m = 1/8$. Pour $\omega_0 = 6$ dans (7), il y a une relation simple entre les échelles τ et les fréquences de Fourier, voir [Torrence & Compo 1998] : $\frac{1}{f} \simeq \tau$.

La convolution (8) transforme une fonction réelle d'une variable en une fonction complexe de deux variables, temps et échelle temporelle τ (fréquence f). En fait, la résolution temporelle de la transformation en ondelettes ΔT et la résolution en fréquence Δf sont liées par la relation de Heisenberg, voir l'équation (1). L'ondelette de Morlet avec $\omega_0 = 6$ est un bon compromis pour la résolution à la fois en temps et en fréquence.

Le carré du module du coefficient d'ondelette, $|\mathcal{W}(\tau, t)|^2$ (ou $|\mathcal{W}(f, t)|^2$) représente le "quantum" d'énergie des fluctuations de $u(t)$ sur la surface $\Delta T \times \Delta\tau$ centrée autour du moment t_j et de l'échelle τ_m (ou de la fréquence f_m). Pour obtenir le spectre des fluctuations il suffit d'intégrer sur la variable temporelle :

$$\mathcal{W}^2(f) = \frac{1}{N} \sum_{i=0}^{N-1} |\mathcal{W}(f, t_j)|^2. \tag{11}$$

La puissance spectrale équivalente au PSD de TF est, voir [Torrence & Compo 1998] :

$$S_w = 2\delta t \mathcal{W}^2(f). \tag{12}$$

2 Application des transformations de Fourier et en ondelettes

2.1 Analyse des signaux synthétiques

1. Construire un signal synthétique

$$y_0(t) = A_0 \sin(2\pi f_0 t),$$

   avec $t_j = j\Delta t$ et $j = 0, 1, 2, ..., N-1$, pour simplifier poser $\Delta t = 1$, $N = 10^3$. (Dans IDL, la fonction *findgen* permet de construire un tableau. Syntaxe : result = findgen(N).) La sauvegarder dans un sous-programme *signal_0.pro*

2. Ecrire une sous-programme (ou une fonction) qui calcule la TF discrète et le PSD d'un signal réel en utilisant les équations (2) et (5). Sauvegarder sous un nom *trans_fourier.pro*
3. Appliquer la TF discrète sur le signal y_0 dans le programme général *ana_signal.pro*. Quelle est la gamme des fréquences Δf couvertes par la transformation ? Pourquoi ?
4. Construire un signal synthétique

$$y_1(t) = A_0 \sin(2\pi f_0 t) + A_1 \sin(2\pi f_1 t),$$

   La sauvegarder dans un sous-programme *signal_1.pro*

5. Construire un signal synthétique

$$y_2(t) = \begin{cases} A_0 \sin(2\pi f_0 t), & \text{pour } t < T/2 \\ A_1 \sin(2\pi f_1 t), & \text{pour } t \geq T/2 \end{cases}$$

   La sauvegarder dans un sous-programme *signal_2.pro*

6. Appliquer la TF discrète sur $y_1(t)$ et $y_2(t)$ dans le programme général *ana_signal.pro*. Comparer les résultats pour les deux signaux.
7. Dans le même programme *ana_signal.pro* faire un diagramme résumant l'analyse : Les 3 signaux $y_0(t)$, $y_1(t)$, $y_2(t)$ et leurs spectres Fourier $S(y_0)$, $S(y_1)$, $S(y_2)$. Consulter l'aide (*help*) d'IDL pour se renseigner sur la fonction *plot* (syntaxe : ? plot) et la création des images en format *.ps*. (sauvegarder la figure sous fig_ana_signal_1.ps). Exemple d'un diagramme en format *.ps* :

```
set_plot, ps
device, filename='plot.ps', /landscape (,/portrait)
plot, findgen(10)
device, /close
set_plot, 'x'
```

8. Dans le même programme *ana_signal.pro* faire la transformation en ondelettes de Morlet pour $y_1(t)$ et $y_2(t)$ en utilisant la fonction *wavelet.*
(Syntaxe : result = wavelet($y_1, \Delta t$,period=τ,coi=coi).) Quelle est la gamme d'échelles temporelles résolues par la transformation ? Comparer avec Δf.
9. Dans le même programme *ana_signal.pro* faire un diagramme résumant l'analyse : (a) scalogramme $W^2(\tau, t)$ pour $y_1(t)$; (b) scalogramme $W^2(\tau, t)$ pour $y_2(t)$.
10. Comparer le spectre Fourier $S(y_1)$ avec le spectre des wavelets $S_w(y1)$. Sauvegarder comme *.ps*
11. Conclure sur l'efficacité des deux transformations.

2.2 Analyse des données spatiales

1. En fonction du temps disponible télécharger les données sur le site http ://amda.cesr.fr/ ou bien lire les données déjà téléchargées. Dans ce dernier cas, la syntaxe est la suivante :
restore,'data_choc_cluster.sav'.
Taper *help* pour avoir le nom et les dimensions des variables. Faire les tracés des composantes du champ magnetique en fonction du temps, en utilisant *!p.multi* d'IDL (dans un programme ana_data.pro).
2. Choisir un intervalle de temps homogène pour l'analyse spectrale, en utilisant la fonction *where* d'IDL, exemple : z=where(time/3600. ge 17.9 and time/3600. le 18.0,nz).
3. Appliquer la transformation de Fourier sur $B_x(t)$. Déterminer le PSD de B_x. Quelle est la gamme des fréquences Δf couvertes par la transformation ? Pourquoi ?
4. Appliquer la transformation en ondelettes sur $B_x(t)$. Quelle est la gamme d'échelles temporelles résolue par la transformation ? Comparer avec Δf.
5. Dans le même programme *ana_data.pro* faire un diagramme résumant l'analyse de la composante B_x : (a) Signal $B_x(t)$; (b) spectre Fourier $S_x(f)$ avec le spectre des wavelets $S_{w,x}(f)$ (en utilisant *oplot*) ; (c) scalogram $W_x^2(\tau, t)$. (sauvegarder la figure sous fig_ana_data.ps)
6. Conclure sur les limites d'application des deux méthodes pour les données spatiales, efficacité, simplicité d'interprétation des résultats, quantité d'information.
7. Dans la limite du temps disponible, déterminer le PSD total, i.e. la somme des PSD des composantes. Si le PSD suit une loi de puissance, faire un *fitting* linèaire entre les $\log_{10}(PSD)$ et $\log_{10}(f)$, en utilisant la fonction *linfit* d'IDL. Rapporter la valeur de la pente trouvée sur le graphe, en utilisant la procédure *xyouts.*

"Quicklook" des mesures Cluster (mesures à 1 UA) peut être trouvé sur :
http ://www.cluster.rl.ac.uk/csdsweb-cgi/csdsweb_pick

3 Annexe IDL/GDL

Cette annexe liste la plupart des fonctions utiles pour le TP, merci de consulter l'aide en ligne

IDL > ?

si vous avez besoin de plus d'information.

Fonctions mathématiques :

1. abs(x) ; valeur absolue de x
2. exp(x) ; exponentielle de x
3. x^n ; puissance n de x
4. sin(x) ; sin de x
5. mean(x) ; moyenne de x
6. !pi ; π
7. i=complex(0,1) ; nombre complexe i

Autres fonctions utilies :

1. tab = findgen(N) ; tableau avec N valeurs de 0 a N-1
2. tab(i) ; retourne l'element i du tableau tab
3. tab(i :j) ; retourne les elements de i à j du tableau tab

Type de variable :

1. y = intarr(N) ; definition de tableau d'entiers de taille N
2. y = fltarr(N) ; definition de tableau de flottants de taille N
3. y = complexarr(N) ; definition de tableau de complexes de taille N

Fonction where

```
ind = where (condition logique sur un tableau, count)
```

Renvoie dans le vecteur ind les positions (indices) ou le test est concluant et dans count (option facultative) le nombre d'occurrences concluantes

Exemple :

```
ind = where (tab LT 10., n_ind)
```

Renvoie dans le vecteur ind les positions ou les elements de tab sont inferieurs a 10, tab est ici un tableau de dimensions quelconques.

Définir une sous-programme :

```
pro write,str
        print,str
end
```

Utiliser cette sous-programme :

IDL > .r write
IDL > write,'bonjour'
bonjour

Définir une fonction :

```
function nb_char, str
        nstr=strlen(str)
        return, nstr
end
```

Utilisation :

IDL > nb_char('bonjour')
7

Exemple d'une boucle :

```
for k=0,N-1 do begin    ;k=index de la frequence
        FT(k)= ...
endfor
```

Example of a plot

```
window,0

!p.multi=[0,1,2]        ; multi-plot 1 column, 2 raws
!p.charsize=1

plot,x,y,xrange=[x_min, x_max],yrange=[y_min,y_max],$
title='satellite Cluster-1, 2001-03-31' ,xtit='time [sec]' ,ytit='Bx [nT]'

plot_oo,f,PSD,xr=[f_min,f_max],yr=[psd_min,psd_max],$
title='Spectrum',xtit='freq [Hz]' ,ytit='PSD [nT2/Hz]'
```

Example of a plot.ps

```
 set_plot, 'ps'
 device, filename='fig1.ps'

!p.multi=[0,1,2]        ; multi-plot 1 column, 2 raws
!p.charsize=1

plot,x,y,xrange=[x_min, x_max],yrange=[y_min,y_max],$
title='satellite Cluster-1, 2001-03-31' ,xtit='time [sec]' ,ytit='Bx [nT]'

plot_oo,f,PSD,xr=[f_min,f_max],yr=[psd_min,psd_max],$
title='Spectrum',xtit='freq [Hz]' ,ytit='PSD [nT2/Hz]'

 device,/close
 set_plot, 'x'
```