TP Analyse spectrale et parcimonie

Les listing de tous les programmes doivent être joints au rapport.

Tous les résultats et courbes doivent être commentés.

I Présentation du TP

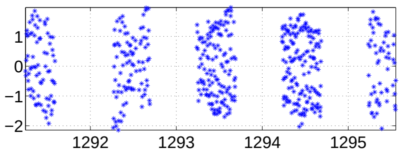
Ce TP vise à illustrer simultanément des aspects liés à l'analyse spectrale de signaux irrégulièrement échantillonnés, en particulier pour les spectres de raies, et les approches de type représentations parcimonieuses des signaux.

Dans un premier temps on étudiera les problèmes liés à l'analyse spectrale par Transformée de Fourier, dans le cas de l'échantillonnage régulier ainsi que dans le cas, fréquent en observation astronomique depuis le sol, d'un échantillonnage irrégulier des données.

Ensuite, on illustrera le comportement de méthodes classiques pour la représentation parcimonieuse des signaux dans ce cas particulièrement difficile de l'analyse de spectre de raies à partir de signaux irrégulièrement échantillonnés. On étudiera à la fois des approches de type gloutonnes (ou « *greedy* ») et des approches de types relaxation convexe, avec pénalisation d'un critère des moindres carrés par la norme ℓ^1.

Nous allons illustrer l'utilisation de ces différents outils sur un exemple simulé correspondant à une situation réaliste de données astronomique.

Les vecteurs `t` et `y` contenus dans le fichier `data.mat` (tracés sur la figure ci-contre) correspondent aux instants d'échantillonnage et à des vitesses radiales observées sur 5 nuits de l'étoile de Herbig HD 104237 auxquelles l'effet des orbites du système multiple ont été soustraits.



Comme il est délicat de bien comprendre les méthodes en travaillant directement sur les données réelles, on va plutôt simuler des données réalistes correspondant à une sommes de sinusoïdes bruitées :

$$x(t_n) = \sum_{k=1}^{K} A_k \cos(2\pi\nu_k t_n + \phi_k) + \epsilon_n.$$

On considèrera en pratique des perturbations ϵ_n indépendants, identiquement distribuées gaussiennes pour un rapport signal sur bruit en puissance moyenne de 10 (20 dB) et $K = 5$ sinusoïdes de paramètres contenus dans les vecteurs de fréquence `f_th` (en jours^{-1}) d'amplitude `A_th` et de phase `phi_th`, avec :

- `f_th = [31.012 32.675 33.283 33.521 35.609]`,
- `A_th = [.25 .75 1 .75 1]`,
- `phi_th = [0.393 0.996 0.492 0.281 0.596]`.

II Analyse spectrale par Transformée de Fourier

II.1 Cas de l'échantillonnage régulier (optionnel)

On cherche dans un premier temps à mesurer les paramètres d'une simple sinusoïde par FFT. Ce problème, bien que très simple, peut provoquer bien des surprises si l'on ne maîtrise pas les outils en question...

On considère un échantillonnage régulier sur un même intervalle de temps et pour un même nombre d'échantillons que les données du fichier `data.mat`.

1. Quelle est la plus haute fréquence que l'on peut mesurer sans erreur pour un tel signal ?
2. Avec ces instants d'échantillonnage réguliers, tracer les représentations temporelle et fréquentielle (par FFT) d'une simple sinusoïde de fréquence `f_0=k_0*F_e/N;` avec `k_0` de votre choix. Peut-on retrouver sans erreur les paramètres (fréquence, amplitude et phase à l'origine) de la sinusoïde ?
3. Faire de même pour une sinusoïde de fréquence `f_0=(k_0+1/2)*F_e/N;` Quelles erreurs obtient-on sur les paramètres (erreur en fréquence, erreur relative en amplitude) de la sinusoïde ? Comment peut-on expliquer de telles erreurs ? Peut-on retrouver la phase à l'origine ?
4. Que proposeriez-vous de faire pour améliorer cette analyse ?
5. Ajouter une deuxième sinusoïde (de même amplitude) à la précédente. A quelle distance minimale faut-il placer la fréquence de cette sinusoïde pour pouvoir la distinguer de la première ? Comment cela s'explique-t-il ? Faire de même pour une sinusoïde d'amplitude 10 fois plus faible. Commenter les résultats obtenus...
6. Pour ces instants d'échantillonnage, construire le signal correspondant aux 5 sinusoïdes du signal à étudier et tracer sa Transformée de Fourier. Peut-on clairement distinguer les 5 sinusoïdes ? Que faudrait-il faire pour améliorer cette analyse ?

II.2 Cas de l'échantillonnage irrégulier

Dans le cas d'un échantillonnage irrégulier, on ne pourra plus calculer la Transformée de Fourier par FFT. Un moyen simple de calculer la Transformée de Fourier consiste à introduire la matrice $\mathbf{W}$ telle que $\boldsymbol{W}(\ell, c) = \exp(2j\pi t_\ell f_c)$. Le vecteur $\widehat{\boldsymbol{x}}$ correspondant à la transformée de Fourier du vecteur $\boldsymbol{x}$ se calcule alors simplement par $\widehat{\boldsymbol{x}} = \mathbf{W}^\dagger \boldsymbol{x}$. Notons que dans le cas de la FFT, cette matrice $\mathbf{W}$ est orthogonale ($\mathbf{W}^\dagger\mathbf{W} = \mathbf{W}\mathbf{W}^\dagger = N\boldsymbol{I}$) ce qui n'est plus vrai dans le cas de l'échantillonnage irrégulier. En Matlab, pour calculer cette Transformée de Fourier sur un vecteur de fréquence `freq =(-M:M)/M*fmax` (vecteur ligne contenant les fréquences de 0 à `fmax` avec un pas de `fmax/M`) pour des échantillons correspondant à un vecteur de temps `t` (vecteur colonne), on construit aisément la matrice `W=exp(2*j*pi*t*freq)`, la représentation fréquentielle (périodogramme) d'un signal se calcule alors aisément par `abs(W'*x)/N;`

1. Quelle est la plus haute fréquence que l'on peut mesurer sans erreur pour un tel signal ? Avec Matlab, construire un vecteur de fréquences `freq` raisonnable pour analyser la transformée de Fourier du signal, en n'hésitant pas à prendre un pas d'échantillonnage suffisamment fin.
2. Avec ces instants d'échantillonnage, tracer les représentations temporelle et fréquentielle d'une simple sinusoïde de fréquence `f_0` de votre choix. Peut-on retrouver sans erreur les paramètres (fréquence, amplitude et phase à l'origine) de la sinusoïde ?

3. Pour ces instants d'échantillonnage, construire le signal correspondant aux 5 sinusoïdes du signal à étudier et tracer sa représentation fréquentielle. Peut-on clairement distinguer les 5 sinusoïdes ?
4. Tracer la fenêtre spectrale correspondant à ces instants d'échantillonnage (elle se calcule simplement en Matlab par `Win=W'*ones(N,1)/N;` ou `N` est le nombre d'échantillons du signal). Commenter cette fenêtre spectrale... Comment peut-on interpréter la représentation fréquentielle du signal à partir de cette fenêtre ?

Lorsqu'ils travaillent avec des signaux à valeurs réelles irrégulièrement échantillonnés, les astronomes préfèrent utiliser le périodogramme de Lomb-Scargle qui correspond à la recherche de signaux sinusoïdaux (avec des fréquences positives) plutôt que la représentation fréquentielle classique qui correspond à la recherche de signaux exponentiels complexes (avec des fréquences négatives et positives). Cependant, pour des raisons de simplicité de mise en œuvre, nous poursuivrons notre analyse à partir de la représentation fréquentielle classique.
Le signal simulé à la question 3 constituera votre jeu de données sur lesquelles vous allez tester les différentes méthodes d'approximation parcimonieuse.

III Parcimonie par approches gloutonnes

III.1 Méthodes de *pre-whitenning* ou *Matching Pursuit* (MP)

L'exemple précédent illustre bien les limites de l'analyse de signaux irrégulièrement échantillonnés par transformée de Fourier. Désirant tout de même retrouver les fréquences de sinusoïdes contenues dans des données de ce type, des astronomes ont proposé, dès les année 1970, d'utiliser des techniques itératives, parfois appelées méthodes de *pre-whitenning*. Le principe de ces méthodes est simple : il consiste à calculer la fréquence maximisant la représentation fréquentielle du signal, de soustraire la composante correspondante des données, et de poursuivre la recherche de sinusoïdes dans ce résidu. De telles méthodes itératives ont été formalisées dans les années 1990 dans le cadre des représentations parcimonieuses des signaux, plus particulièrement en tant qu'algorithme glouton (*greedy*) de type *Matching Pursuit*. Un tel algorithme est schématisé Tab. 1 (remarquons que dans notre cas, on a : $\forall k, \boldsymbol{w}_k^\dagger \boldsymbol{w}_k = N$).

Initialisation $k = 0$		$\boldsymbol{r}_0 = \boldsymbol{x}$, $\Gamma_0 = \emptyset$, $\boldsymbol{a} = \boldsymbol{0}$.
Itérations $n = 1 \ldots$	*a)*	Sélection de l'atome d'indice $k = \arg\max_k \lvert \boldsymbol{w}_k^\dagger \boldsymbol{r}_n \rvert$. Mise à jour de la liste d'indices $\Gamma_n = \Gamma_{n-1} \cup \{k\}$
	b)	Mise à jour de l'amplitude $a_k = a_k + \frac{1}{\boldsymbol{w}_k^\dagger \boldsymbol{w}_k} \boldsymbol{w}_k^\dagger \boldsymbol{r}_n$ et des résidus $\boldsymbol{r}_n = \boldsymbol{r}_{n-1} - \frac{1}{\boldsymbol{w}_k^\dagger \boldsymbol{w}_k} \boldsymbol{w}_k^\dagger \boldsymbol{r}_n \cdot \boldsymbol{w}_k$

Tab. 1 – Schéma de principe d'un algorithme de *Matching Pursuit* pour l'approximation parcimonieuse d'un signal $\boldsymbol{x}$ à l'aide d'un dictionnaire (matrice) $\boldsymbol{W}$ ($\boldsymbol{x} \approx \mathbf{W}\boldsymbol{a}$ avec $\boldsymbol{a}$ parcimonieux)

Divers tests peuvent être utilisés pour arrêter les itérations. On se contentera ici de faire un simple test sur le résidu. Plus précisément, si le signal est parfaitement approximé, le résidu $\boldsymbol{r}$ ne contient que le bruit gaussien de variance σ_b^2 que l'on suppose connu. Dans ce cas le terme

$T = ||\boldsymbol{r}||^2/\sigma_b^2$ est une variable du χ^2 à N degrés de liberté. En pratique, on utilisera donc comme test d'arrêt $T < \tau$ avec τ tel que Proba$(T < \tau) = 95\%$ (τ pourra être calculé grâce à la fonction Matlab `tau = chisqq(0.95,N);`), ce qui signifie que l'on ne peut pas distinguer le résidu du bruit avec une probabilité de 95%.

1. Mettre en œuvre en Matlab un algorithme de *Matching Pursuit* pour estimer les fréquences contenues dans notre signal.
2. Combien de fréquences détecte-t-on avec cet algorithme ? Ces fréquences sont-elles bien localisées ? Les amplitudes correspondantes sont-elles correctement estimées ?
3. Analyser les résultats obtenus par cet algorithme au cours des itérations et essayer de caractériser ses principaux défauts (ordre de détection des fréquences, fausses détections. . .).

III.2 *Orthogonal Matching Pursuit* (OMP) (optionnel)

Un des défaut de l'algorithme du *Matching Pursuit* vient du fait que seule l'amplitude de l'indice sélectionné est mise à jour à chaque itération. Ainsi, il est possible que l'algorithme sélectionne plusieurs fois un même indice afin d'obtenir une amplitude correcte. Ce défaut est simple à corriger. Il suffit de remplacer l'étape de mise à jour *b)* : pour l'amplitude par une projection orthogonale du signal sur les éléments sélectionnés :

$$\boldsymbol{a}(\Gamma_n) = \arg\min_{\boldsymbol{a}(\Gamma_n)} ||\boldsymbol{x} - \mathbf{W}_{\Gamma_n}\boldsymbol{a}(\Gamma_n)||^2 = (\mathbf{W}_{\Gamma_n}^\dagger \mathbf{W}_{\Gamma_n})^{-1}\mathbf{W}_{\Gamma_n}^\dagger \boldsymbol{x}$$

où $\boldsymbol{a}(\Gamma_n)$ correspond aux éléments de $\boldsymbol{a}$ d'indices Γ_n et $\mathbf{W}_{\Gamma_n}$ correspond à la sous matrice de $\mathbf{W}$ contenant uniquement les colonnes d'indices Γ_n ; et pour le résidu : $\boldsymbol{r}_n = \boldsymbol{x} - \mathbf{W}_{\Gamma_n}\boldsymbol{a}(\Gamma_n)$.

1. Mettre en œuvre en Matlab un algorithme d'*Orthogonal Matching Pursuit* pour estimer les fréquences contenues dans notre signal.
2. Combien de fréquences détecte-t-on avec cet algorithme ? Ces fréquences sont-elles bien localisées ? Les amplitudes correspondantes sont-elles correctement estimées ?
3. Analyser les résultats obtenus par cet algorithme au cours des itérations et essayer de caractériser ses principaux défauts et de les comparer à ceux du *Matching Pursuit*.

III.3 *Orthogonal Least Square* (OLS)

Le principal défaut des algorithmes précédents vient de la phase de sélection. Cette phase serait tout à fait cohérente si les colonnes de la matrice $\mathbf{W}$ étaient orthogonales, mais ce n'est pas le cas ici. Comme l'objectif est d'avoir une erreur d'approximation la plus faible possible, il parait intéressant d'exploiter cette propriété dès la phase de sélection. L'algorithme *Orthogonal Least Square* (souvent confondu à tort avec l'algorithme OMP) consiste ainsi à remplacer l'étape de sélection *a)* de l'algorithme OMP par la sélection de l'indice k permettant de réduire le plus l'erreur d'approximation lorsqu'on ajoute la colonne $\boldsymbol{w}_k$ à la matrice $\mathbf{W}_{\Gamma_n}$:

$$k = \arg\min_k \arg\min_{\boldsymbol{a}(\Gamma_n \cup \{k\})} ||\boldsymbol{x} - \mathbf{W}_{\Gamma_n \cup \{k\}}\boldsymbol{a}(\Gamma_n \cup \{k\})||^2.$$

Un tel algorithme est évidement plus coûteux en temps de calculs que les algorithmes MP et OMP, surtout si la mise en œuvre n'est pas optimisée. Vous utiliserez dans cette partie la fonction Matlab `ols` : `[a, ind] = ols(W,x,Inf,test);` avec `test=`$\sigma_b^2\tau$.

1. Combien de fréquences détecte-t-on avec cet algorithme ? Ces fréquences sont-elles bien localisées ? Les amplitudes correspondantes sont-elles correctement estimées ?
2. Analyser les résultats obtenus par cet algorithme et essayer de caractériser ses principaux défauts et de les comparer à ceux du *Matching Pursuit* et de l'*Orthogonal Matching Pursuit*.

IV Parcimonie par relaxation convexe

Un autre moyen d'obtenir une approximation parcimonieuse d'un signal consiste à minimiser un critère quadratique pénalisé par une norme ℓ^1 :

$$\boldsymbol{a} = \arg\min_{\boldsymbol{a}} ||\boldsymbol{x} - \mathbf{W}\boldsymbol{a}||^2 + \lambda ||\boldsymbol{a}||_1 \text{ avec } ||\boldsymbol{a}||_1 = \sum_k |\boldsymbol{a}_k|$$

Ce critère est convexe et ne possède donc pas de minima locaux et, malgré le fait qu'il ne soit pas dérivable, de nombreux algorithmes peuvent être utilisés pour le minimiser. C'est en particulier le cas de la fonction `min_l2_l1_0` qui peut être appelée depuis Matlab après compilation (commande Matlab `mex min_L2_L1_0`) : `a1 = min_L2_L1_0(x,W,lambda,n_it_max);` avec `n_it_max` le nombre maximum d'itérations de l'algorithme. Avec une telle méthode, il n'y a pas de test d'arrêt en fonction du nombre de fréquences trouvées mais un paramètre de pénalisation λ à régler. On peut montrer que la solution $\boldsymbol{a}_1$ ainsi obtenue vérifie : $|\mathbf{W}^\dagger(\boldsymbol{x} - \mathbf{W}\boldsymbol{a}_1)| \leq \lambda$ ce qui signifie que la représentation fréquentielle du résidu est inférieur ou égal à λ/N. On peut donc régler la valeur de λ en fonction de la valeur maximale souhaitée pour la représentation fréquentielle du résidu.

1. Tester cet algorithme pour différentes valeurs de λ jusqu'à obtenir un résultat satisfaisant (pour vérifier si le résultat est satisfaisant, on peut afficher les représentations temporelle et fréquentielle du résidu ainsi que la valeur de test τ utilisée pour contrôler les algorithmes précédents).
2. Combien de fréquences détecte-t-on avec cet algorithme ? Ces fréquences sont-elles bien localisées ? Les amplitudes correspondantes sont-elles correctement estimées ?
3. On remarquera que les fréquences détectées par cette méthode sont souvent doublées sur la grille, autour des vrais valeurs des fréquences. En ne conservant que l'une de ces fréquences doubles, on peut envisager de ré-estimer par moindres carrés les amplitudes correspondant à ces fréquences détectées. En procédant ainsi, combien de fréquences détecte-t-on avec cet algorithme ? Ces fréquences sont-elles bien localisées ? Les amplitudes correspondantes sont-elles correctement estimées ?

V Conclusion

Conclure sur le problèmes posé et les différentes approches utilisées pour tenter de le résoudre... En pratique que recommandez vous de faire ?