Université Paris-Sud
Faculté des Sciences d'Orsay



**Problèmes inverses**

# Sujet 2 – Transformées temps fréquence

F. Orieux

orieux@l2s.centralesupelec.fr

2017 — 2018

Dans ce sujet on aborde les notions d'analyse temps fréquences. Elles sont indispensables pour l'analyse de toutes données temporelles et utilisées dans l'analyse d'onde gravitationnelle par exemple. Ici nous nous exerçons sur de la musique se prêtant bien à ce type d'analyse.

**Remarque 1** *Quelques fonctions utilitaires comme* `slice` *sont présentes dans le module* `pbinv.py`*. Ce module est à charger dorénavant pour la suite des travaux.*

## 1 Analyse Harmonique

### 1.1 Périodogramme

Le périodogramme pour un signal de longueur $K$ s'écrit comme le module au carré de la transformée de Fourier à temps discret, normalisé par $K$,

$$P(\nu) = \frac{1}{K}\left|\sum_{k=0}^{K-1} x[k]e^{-2\imath\pi k\nu}\right|^2. \tag{1}$$

C'est une fonction réelle de la variable réelle $\nu$. On le calcul sur machine pour un jeu de $N$ fréquences discrètes $\nu = n/N$ pour $n \in \{0, \dots, N-1\}$ par TFD. On peut aussi le représenter pour $\nu$ allant de $-0{,}5$ à $0{,}5$, en fréquences réduites toujours.

### 1.2 Analyse stationnaire

1. Tracez les périodogrammes des signaux `mallat.sav`, `doppler.sav`, `piano.sav` et `test.sav`. N'omettez pas de graduer correctement les axes en valeur et unité. Pour relier la fréquence réduite à la fréquence réelle, la fréquence d'échantillonage est disponible dans la variable `Fe` conjointement aux signaux.
   **Remarque 2** *Après avoir chargé les données avec la fonction* `scipy.io.readsav`*, regarder le contenu avec les clés, par exemple*

```python
import scipy.io
data = scipy.io.readsav('doppler.sav')
data.keys()
# renvoie 'dict_keys(['dop', 'fe'])'
```

2. Prenez le temps de réfléchir et de répondre à chacune des questions suivantes. Que contiennent les signaux ? Que se passe-t-il si on ne traite qu'un point sur deux ?
3. Calculez le périodogramme sur la première moitié des échantillons puis sur la seconde moitié. Comparez les résultats et que peut-on en conclure sur la stationarité des signaux ?

## 2 La transformée de Fourier à fenêtre glissante

### 2.1 Définition

La transformée de Fourier à fenêtre glissante s'écrit comme

$$\begin{aligned}\bar{X}(t,\nu) &= \int x(\tau)\Pi_{T_o}(\tau - t)e^{-2\imath\pi\nu\tau}\,\mathrm{d}\tau \\ &= \int_{t-T_o/2}^{t+T_o/2} x(\tau)e^{-2\imath\pi\nu\tau}\,\mathrm{d}\tau\end{aligned}$$

c'est à dire la transformé de Fourier du signal multiplié par une fonction porte décalée.

La version discréte s'écrit simplement comme la TFD sur une sous partie des échantillons du signal numérique. On peut alors définir le « spectrogramme » comme le module au carré de la transformée de Fourier discrète à fenêtre glissante sur $N$ points

$$\bar{X}_d(n, \nu_n) = \frac{1}{K}\left|\sum_{k=1}^{K} x[n - K/2 + k]e^{-2\imath\pi k\nu_n}\right|^2 \tag{2}$$

avec les fréquences réduites $\nu_n = n/N$ et $n \in \{0, \dots, N-1\}$.

### 2.2 Algorithme

1. Découper le signal composé de $N$ points en $P$ signaux de $K$ points à l'aide de la fonction `slice`. La fonction retourne le résultat sous forme d'une matrice de taille $K \times P$.
2. Application de la TFD 1D sur chacun des $P$ signaux. La fonction `np.fft.fft` accepte un tableau 2D et effectue la TFD sur l'axe indiqué par le paramètre `axis`.
   **Remarque 3** *Regarder l'aide de la fonction* `np.fft.fft`*. Ici on souhaite transformer les K points temporels en fréquence et avoir ainsi P spectres sur des portions décalés du signal d'origine.*

Pour valider votre algorithme, testez le sur un signal contenant deux sinusoïdes successives et à des fréquences respectives différentes.

**Remarque 4** *Pour obtenir une meilleure visualisation du résultat, on peut utiliser la fonction fournie* `normspec` *sur le spectrogramme.*

*La fonction* `plt.imshow` *possède un paramètre* `aspect` *qui peut permettre un meilleur affichage également.*

## 2.3 Mise en œuvre sur le signal Doppler

1. Appliquez la transformée de Fourier à fenêtre glissante au signal Doppler. Reconnaissez l'axe des temps et l'axe des fréquences.
2. Dans le cycle cardiaque, identifiez la période de calme et la période d'expulsion du sang à plus grande vitesse.

## 2.4 Mise en œuvre sur le signal musical

1. Appliquez la transformée de Fourier à fenêtre glissante au signal musical.
2. Identifiez les notes de plus hautes énergies à l'aide du tableau ci dessous.

**Remarque 5** *Il ne faut pas tenir compte des raies au dessus de* 1100 *Hz.*

**Remarque 6** *Le piano a été accordé avec le* `la3` *à* 444 Hz*, il faut donc pour les notes comprises entre* 780 *et* 1000 Hz *soustraire environ* 8 Hz *pour utiliser le tableau.*

| Note | Hz | Note | Hz |
|---|---|---|---|
| do2 | 130,81 | do4 | 523,25 |
| do2# | 138,59 | do4# | 554,36 |
| ré2 | 146,83 | ré4 | 587,33 |
| mi2b | 155,56 | mi4b | 622,26 |
| mi2 | 164,81 | mi4 | 659,26 |
| fa2 | 174,61 | fa4 | 698,46 |
| fa2# | 185 | fa4# | 740 |
| sol2 | 196 | sol4 | 784 |
| la2b | 207,66 | la4b | 830,62 |
| la2 | 220 | la4 | 880 |
| si2b | 233,08 | si4b | 932,32 |
| si2 | 246,94 | si4 | 987,76 |
| do3 | 261,63 | do5 | 1046,5 |
| do3# | 277,18 | do5# | 1088,72 |
| ré3 | 293,66 | ré5 | 1174,66 |
| mi3b | 311,13 | mi5b | 1244,52 |
| mi3 | 329,63 | mi5 | 1318,52 |
| fa3 | 349,23 | fa5 | 1396,92 |
| fa3# | 369,99 | fa5# | 1480 |
| sol3 | 392 | sol5 | 1568 |
| la3b | 415,31 | la5b | 1661,24 |
| la3 | 440 | la5 | 1760 |
| si3b | 466,16 | si5b | 1864,64 |
| si3 | 493,88 | si5 | 1975,52 |

**Table 1 –** Fréquences des notes de musiques.

# 3 La transformée de Gabor

La transformée de Gabor (TG) peut être vue comme un rafinnement de la transformée de Fourier à court terme. Sa particularité est de multiplier (ou pondérer) par une gaussienne décalée plutôt que par une fonction porte. Sa précision d'analyse dans le plan temps fréquence est optimale.

## 3.1 Transformée direct

### 3.1.1 Définition

La transformé de Gabor s'écrit

$$\mathcal{G}f(t,\nu) = (2\pi\sigma_t^2)^{1/4} \int_{\mathbb{R}} f(\tau) e^{-\frac{(\tau-t)^2}{4\sigma_t^2}} e^{-2\imath\pi\nu\tau} \mathrm{d}\tau.$$

Cette transformée peut également être vue comme une projection sur les atomes de Gabor, dilater par $\nu$ et décalé en $t$

$$g_{t,\nu}(\tau) = (2\pi\sigma^2)^{-1/4} e^{-\frac{(\tau-t)^2}{4\sigma_t^2}} e^{2\imath\pi\nu\tau}.$$

C'est une sinusoïde modulée en amplitude par une gaussienne. C'est également une des premières ondelettes crée. On peut montrer également que $\sigma_t\sigma_\nu = 1/4\pi$.

En pratique sur machine de calcul il faut utiliser la TFD ainsi qu'une fenêtre de pondération de taille finie où l'on conserve au moins 99 % de l'énergie. La TG discrète s'écrit alors

$$G(k,\nu_n) = \sum_{n=1}^{N} x[n] e^{-\frac{(n-k)^2}{4\sigma_t^2}} e^{-2\imath\pi n\nu_n} \tag{3}$$

avec $\nu_n = n/N$ et $n \in \{0, \dots, N-1\}$ et $0 < k \leq N$.

### 3.1.2 Algorithme

1. Calcul de la fenêtre gaussienne tronquée à l'aide de la fonction fournie `gaussian_window`.
2. Découpage du signal composée de $N$ points en $P$ signaux de $K$ points à l'aide de la fonction `slice`.
3. Multiplication des $P$ signaux par la fenêtre gaussienne tronquée.
4. Application de la TFD 1D sur chacun des $P$ signaux.

Testez votre algorithme sur le signal synthétique pour le valider.

### 3.1.3 Mise en œuvre

Appliquez la transformée de Gabor sur les deux signaux précédents et comparer avec la transformée de fourier à fenêtre glissante.

## 3.2 Transformée de Gabor inverse

La transformée de Gabor n'est pas une transformée orthogonale : si les fenêtre d'analyse se recouvre l'information présente dans le spectrogramme est redondante. Il faut tenir compte de cette redondance pour implémenter l'inversion. Si on applique la TFD inverse sur un indice temporel $n$ de la transformée alors nous avons

$$g(n,k) = \sum_{n=0}^{N-1} G(k,\nu_n) e^{2j\pi\nu_n k} = x_n e^{-\frac{(n-k)^2}{4\sigma_t^2}}. \tag{4}$$

Si l'échantillon temporel $x_n$ à été analysé plusieurs fois pour $P$ différentes fenêtres, alors il faut utiliser toutes ces transformées inverses et faire une moyenne pondérées en fonction de sa position dans les différentes fenêtres. On en déduit

$$x_n = \frac{\sum_{p=0}^{P-1} g(n_p,k)}{\sum_{p=0}^{P-1} e^{-\frac{(n_p-k)^2}{4\sigma_t^2}}} \tag{5}$$

où $n_p$ est l'instant correspondant à la fenêtre $p$.

Si les fenêtres sont disjointes alors il suffit simplement de démoduler (diviser) les TFD inverses.

#### 3.2.1 Algorithme

1. Calcul de la fenêtre gaussienne tronquée.
2. Calcul de la TFD inverse sur chacune des $P$ colonnes du spectrogramme.
3. Calcul des $x_n$ à l'aide de l'équation (5).

Testez votre algorithme sur le signal synthétique pour le valider.

#### 3.2.2 Mise en œuvre

Calculez le spectrogramme du signal musical et eliminez une note en appliquant un masque sur le spectrogramme. Enfin utilisez la transformée de Gabor inverse pour synthétiser le signal. Visualiser le signal ainsi obtenu. Si possible écoutez le après l'avoir exporter à l'aide de la fonction `scipy.io.wavfile.write`

```
import scipy.io.wavfile
scipy.io.wavfile.write(...)
```

## 4 Transformée en ondelettes continues

La transformée de Gabor est optimal en localisation temps fréquence. Cependant elle analyse tout le plan avec des boîtes d'Heisenberg de taille fixe. La transformée en ondelette notamment été introduite pour avoir des boîtes de taille variable.

On peut en effet considérer que les composantes basses fréquences sont peu localisées dans le temps. Une précision faible en temps sans conséquence permet d'avoir une bonne précision fréquentielle. Inversement, les évenements rares comme des transitions franches nécessitent une bonne localisation temporelle quite à sacrifier la précision fréquentielle. On dit alors que la tranformée en ondelette est adaptée aux signaux réguliers par morceaux.

### 4.1 Définition

La transformée en ondelette continue est la projection par produit scalaire $W_\psi f(a, b) = \langle f, \psi_{a,b} \rangle$ de notre signal $f$ sur une famille de fonctions $\psi_{a,b}$ où les fonctions de la famille sont isssues de la translation et dilatation du ondelette appelée « mère »

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}} \psi \left( \frac{t-b}{a} \right) \quad \text{où} \quad (a, b) \in \mathbb{R}^* \times \mathbb{R}.$$

On peut donc assimiler la variable $a$ à une échelle ou une fréquence et la variable $b$ au temps.

La transformée en ondelettes continue est redondante. La famille des ondelettes $\psi_{a,b}$ restent une base de $L^2(\mathbb{R})$ avec une discrétisation dyadique des échelles $a = 2^j$. En pratique, le signal que l'on mesure est discret. On procède au calcul avec des pas discrets de $b$. Les coefficients $W_\psi f$ sont alors calculés par convolution discrète.

### 4.2 Applications

Testez la fonction `scipy.signal.cwt` sur les signaux étudiés précédemment ainsi que sur le signal contenu dans le fichier `Mallat.sav`

**Remarque 7** *L'aide de la fonction fournie un example ainsi que l'usage du chapeau mexicain comme ondelette (la dérivé seconde de la gaussienne).*

Comparez la régularité du signal `mallat` et sa transformée en ondelettes continue. Comparer les resultats par rapport aux transformées précédentes. Qu'en concluez vous ?