Initiation au traitement du signal et applications
Séance 1

École des Mines de Nancy

Frédéric Sur
http://www.loria.fr/~sur/enseignement/signal/
sur@loria.fr

Consigne : Il vous est demandé de **rédiger les travaux pratiques**. Ouvrez en parallèle un document Word dans lequel vous allez coller les différents résultats obtenus, les commandes qui vous ont permis de les obtenir, et les réflexions que cela vous inspire. L'évaluation de ce cours sera basée sur ces comptes rendus.

Il s'agit d'un premier TP de prise en main. Référez vous à l'aide Matlab pour chercher la syntaxe des fonctions utilisées. Vous pouvez aussi regarder la section *Getting started* de cette aide.
Pour obtenir l'aide sur une fonction, tapez `help nom_fonction`.

Rappels : opérations élémentaires sur les matrices. (`[Entrée]` après chaque commande)

`>> A=zeros(10,20)` crée et affiche une matrice nulle A de taille 10x20.

`>> B=eye(10);` crée (sans l'afficher à cause du `;`) une matrice identité de taille 10.

`>> B` provoque l'affichage de la matrice B.

`>> C=D+E;` stocke le résultat de la somme matricielle $D + E$ dans C.

Les opérations élémentaires sur les matrices se notent `+ - * / ^`
Lorsque l'on veut appliquer l'opérateur sur les termes individuels de la matrice, on le fait précéder d'un point. Par exemple :
`>> C=D.*E;` stocke dans $c_{i,j}$ le produit $d_{i,j}e_{i,j}$ (il faut bien sûr que D et E soient de même taille).
`>> C=D.^2;` stocke dans $c_{i,j}$ le carré $d_{i,j}^2$.

Conseil : copiez les commandes dans un fichier texte afin de pouvoir faire simplement des copier-coller dans Matlab, ou alors utilisez l'historique (flèche vers le haut). Vous pouvez aussi directement copier-coller les instructions depuis ce fichier pdf.

1 Un peu de musique

Sur la page web du TP figurent des fichiers sons[1].
Chargez-les dans des matrices, et notez leur fréquence d'échantillonnage. Par exemple :

```
>> [W1, Fs, nbits] = wavread('diapason.wav');
>> Fs  % Fs est le nombre d'échantillons par seconde dans le signal
>> length(W1)  % donne le nombre d'échantillons formant le signal
>> wavplay(W1,Fs);  % permet de jouer un signal sur la carte son
```

Observez leurs amplitudes, ainsi que leurs spectres.

[1] viennent de : `http ://www.100-pour-100-musique.com/samples-guitare.html` et : `http ://www.ceptive.it/suoni/index.php` (*Strumenti musicali*)

```
>> plot(W1);  % zoomez et remarquez l'aspect ''sinusoidal''
>> fW1=fft(W1);
>> plot(abs(fW1));
>> plot(abs(fftshift(fW1)));  % zoomez sur la partie centrale
```

À quoi sert `fftshift` ? (*indication* : où est le coefficient de fréquence nulle correspondant à la moyenne ?)
Constatez la décroissance des coefficients de Fourier.
Pour faciliter la visualisation, on peut aussi visualiser le logarithme du module des coefficients :

```
>> plot(log(abs(fftshift(fW1))));
```

ou bien :

```
>> semilogy(abs(fftshift(fW1)));
>> grid on
```

Regardez comment sont calculés les coefficients de Fourier par Matlab (`help fft`). Quelles différences pouvez-vous constater avec les formules du cours ?

Que constatez-vous en terme de symétrie sur le graphe des spectres ? Vérifiez la répartition particulière des coefficients de Fourier dans le cas des signaux réels.

Vérifiez la validité de la formule de Parseval.
Indication : `sum(A)` donne la somme des éléments de la matrice ligne A.

À quelle fréquence correspond le plus grand pic dans le spectre ? Pouvez-vous retrouver quelle est la note jouée ?
Indication :

```
>> [X,indice]=max(abs(fW1));
>> l=length(W1);
```

`indice` contient alors l'indice du maximum, et `l` est le nombre d'échantillons dans `W1`. Déduisez la fréquence correspondant à `indice`. (rappel : `Fs` est le nombre d'échantillons par seconde et `fW1(indice)` est le coefficient de fréquence `(indice-1)/a` où `a` est la longueur du morceau en secondes.)

2 Quelques images

Sur la page web du TP vous pouvez charger différentes images : des lignes, des images « génériques », et des images de textures[2].

2.1 Manipulations élémentaires

Chargez les images `lena.tif` ou `emn.tif` dans des variables et visualisez-les. Par exemple :

```
>> Ima=double(imread('lena.tif'));
>> colormap(gray);
>> imagesc(Ima);
```

Affichez le module (ou son logarithme) de la transformée de Fourier des images

```
>> fIma=fft2(Ima);
>> figure
>> colormap(gray)
>> imagesc(log(abs(fftshift(fIma))));
```

Constatez la symétrie des modules, la décroissance des coefficients de Fourier, et vérifiez la formule de Parseval.

[2] viennent de : `http ://www.ux.uis.no/~tranden/brodatz.html`

2.2 Fréquences et images

On va mettre en évidence le rôle des hautes fréquences et des basses fréquences.

```
>> U=zeros(256,256);
>> U(101:157,101:157)=1;
>> fImab=fftshift(fIma).*U;
>> figure
>> colormap(gray)
>> imagesc(real(ifft2(ifftshift(fImab))));
>> V=ones(256,256)-U;
>> fImac=fftshift(fIma).*V;
>> figure
>> colormap(gray)
>> imagesc(real(ifft2(ifftshift(fImac))));
```

Faites l'expérience sur `lena.tif` et `emn.tif`. Que constatez-vous ? À quoi correspondent les hautes fréquences et les basses fréquences dans les images ?

2.3 Premières propriétés de la transformée de Fourier des images

Affichez les modules des transformées de Fourier des images `ligne1.tif` et `ligne2.tif`

Exemple :

```
>> Ima=double(imread('ligne1.tif'));
>> FIma=fft2(Ima);
>> colormap(gray)  % permet de mieux voir
>> imagesc(abs(fftshift(FIma)));
>> imagesc(log(abs(fftshift(FIma))));  % essayez aussi en echelle log
```

Attention à affichez les images en assez grand pour qu'un pixel à l'écran corresponde à un seul coefficient de Fourier.

Que constatez-vous ? Pourquoi ? Rappelez-vous que la TFD-2D est obtenue par une TFD-1D selon les lignes puis une TFD-1D selon les colonnes (ou l'inverse, les deux opérations commutent).

De même, examinez les transformées des images `image1.tif`, et `image2.tif`. Regardez soigneusement autour du point de fréquence nulle, de coordonnées $(257, 257)$ ici. La visualisation est plus simple avec `colormap(colorcube)` et en échelle « classique » (non logarithmique). Vous pouvez aussi « binariser » le spectre en le « seuillant » à $5/1000$ de sa plus grande valeur (par exemple).

Expliquez les points alignés observés .

Interprétez le module de la transformée de Fourier de l'image `emn.tif` à l'aide de ces résultats : à quoi correspondent les lignes verticale et horizontale, la ligne « verticale » un peu penchée, et les deux lignes verticales intersectant la ligne horizontale à peu près à mi-distance du bord ?

2.4 Études d'images de textures

Certaines textures sont caractérisées par le *module* de leur transformée de Fourier. On va donc vérifier expérimentalement que la phase n'apporte pas une information cruciale.

Que fait la suite d'instructions suivantes ?

```
>> T=double(imread('D3.tif'));
>> fT=fft2(T);
>> A=rand(512,512);
```

```
>> P=angle(fft2(A));
>> fTalea=abs(fT).*exp(i*P);
>> Talea=ifft2(fTalea);
>> colormap(gray)
>> imagesc(log(abs(fftshift(fT))))
>> figure, colormap(gray)
>> imagesc(T)
>> figure, colormap(gray)
>> imagesc(real(Talea))  % matrice a priori reelle...
```

Interprétez les modules des transformées de Fourier observés sur les différentes textures (`D3.tif`, `D21.tif`, `D82.tif`, `D95.tif`, `D104.tif`) à l'aide des résultats de la section 2.3. En particulier, lorsque vous observez des spots, mettez en relation la position des spots avec le nombre de degrés de symétries de la cellule unitaire de la texture.

Que pensez-vous des textures synthétisées ? À votre avis, quelles sont les textures qui seront correctement caractérisées par le module de leur transformée de Fourier ?

2.5 Expérience d'Openheim et Lim

Attention, en général, il y a tout de même de l'information dans la phase de la transformée de Fourier. La célèbre expérience suivante[3] montre même qu'une part essentielle de l'information s'y trouve.

Échangez les phases des images `lena.tif` et `emn.tif`.

Que constatez-vous ? Dans la section 2.2 on a mis en évidence l'information contenue dans le module. Quelle information contient la phase ?

3 Bonus, pour ceux qui vont vite : retour au son

Chargez dans une variable le contenu du fichier `foryoublue.wav`.

Faites une convolution du signal avec la matrice `N=ones(1,T)/T` pour différentes valeurs du paramètres `T` (fonction `conv`, voir l'aide). Écoutez le résultat et commentez.

Complétez la matrice `N` avec des zéros pour obtenir une matrice de même longueur que le fichier son original, et examinez le spectre. Montrez qu'il s'agit d'un filtre passe-bas.

Ajoutez un bruit blanc sur le signal initial (fonction `randn`, jouez avec la variance). Appliquez le filtre précédent sur le signal « bruité ». Que constatez-vous ?

[3] cf A. V. Openheim and J. S. Lim, *The importance of phase in signals*. Proceedings of the IEEE Vol. 69 Nr 5, 529-541, 1981.