Analyse des Signaux et des Images

TP : Transformée de Fourier Discrète

2

1 Avant-Propos

L'objectif de ce TP est de mettre en pratique à l'aide du logiciel Matlab/Octave et d'un oscilloscope, les notions de signaux numériques, d'échantillonnage, de Transformée de Fourier Discrète, de fenêtrage et de résolution en fréquence. Comme l'enseignement des notions de transformée de Fourier discrète et de fenêtrage se déroule vers la fin du premier semestre, il est procédé à une introduction rapide des concepts principaux.

2 Préparation

La préparation nécessaire à ce TP est la suivante :

- Relire le cours en particulier les notions de signaux numériques et d'échantillonnage.
- Lire les notions sur la Transformée de Fourier Discrète dans le cas où le cours n'a pas encore eu lieu (Partie 3)
- Avoir Matlab/Octave sur son portable et être au point sur la manipulation des opérations de base (Partie 4).

3 Notions de Transformée Discrète

3.1 Un peu de théorie...

La transformée de Fourier continue enseignée dans le cours de mathématiques n'est pas directement implémentable dans un calculateur. En effet, si l'on considère le signal temporel $x(t)$, sa transformée de Fourier continue $X(\nu)$ est :

$$X(\nu) = \int_{-\infty}^{+\infty} \exp\{-2i\pi\nu t\} x(t) dt \tag{1}$$

Elle demande de connaître $X(t)$ sur une infinité de points tandis que l'intégration temporelle demande de sommer une infinité de termes. De manière à pouvoir réaliser une transformée de Fourier sur un calculateur, différentes simplifications sont introduites. Généralement, on travaille avec des signaux numériques (échantillonnés au niveau temporel), ce qui implique que l'on doit transformer l'intégrale en une somme infinie de termes:

$$X_1(\nu) = \sum_{n=-\infty}^{+\infty} x(nTe) \exp\{-2i\pi\nu nTe\} \tag{2}$$

Le second problème est que l'on ne sait pas faire une somme de $-\infty$ à $+\infty$ de termes avec un calculateur. On est donc obligé de prendre une somme de N termes allant de 0 à N-1.

$$X_2(\nu) = \sum_{n=0}^{N-1} x(nTe) \exp\{-2i\pi\nu nTe\} \tag{3}$$

Cette simplification est équivalente en pratique au fait d'observer le signal sur une durée finie ou de réaliser un fenêtrage rectangulaire du signal. Enfin, la transformée précédente $X_2(\nu)$ n'est toujours pas implémentable en l'état puisque ν la fréquence est une variable continue. Or il est impossible de stocker une infinité de valeurs. On est donc obligé de prendre des valeurs discrètes au niveau fréquentiel c'est à dire échantillonné ν. En posant le fait que $\nu = k\nu_s$, on a:

$$X_3(\nu = k\nu_s) = \sum_{n=0}^{N-1} x(nTe) \exp\{-2i\pi nk\frac{\nu_s}{\nu_e}\} \tag{4}$$

Il reste à déterminer en quelles fréquences ν_s, la TFD sera évaluée. Généralement, on pose que ν_s est l'inverse de la durée d'observation du signal, soit:

$$\nu_s = \frac{1}{NT_e} \tag{5}$$

Ce qui donne par simplification, l'expression finale de la transformée discrète :

$$X_{TFD}(\nu = \frac{k}{NT_e}) = \sum_{n=0}^{N-1} x(nTe) \exp\{\frac{-2i\pi nk}{N}\} \tag{6}$$

avec k $\in \{0, 1, 2, ...N-1\}$.

La transformée de Fourier discrète se caractérise par une somme d'échantillons temporels qui sont chacun d'entre eux multipliés par une exponentielle complexe. La phase de cette exponentielle dépendant de différents paramètres comme l'indice de l'échantillon temporel (n) ou fréquentiel (k). Elle est parfaitement adaptée à un calculateur car en pratique on ne manipule qu'une suite de nombres. L'image de la transformée de Fourier discrète est en général un nombre complexe caractérisé par un module et une phase.

On peut aussi l'exprimer la somme sous la forme d'un produit matriciel:

$$\begin{pmatrix} X(k=0) \\ X(k=1) \\ \vdots \\ X(k=N-2) \\ X(k=N-1) \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ 1 & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \exp\{\frac{-2ink}{N}\} & \vdots & \vdots \\ 1 & 1 & \dots & \dots & \dots \\ 1 & 1 & \dots & \dots & \dots \end{pmatrix} \times \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-2) \\ x(N-1) \end{pmatrix} \quad (7)$$

Les différents échantillons de TFD $X(k)$ sont obtenus par une simple multiplication matricielle entre les échantillons temporels $x(n)$ et la matrice formée par les termes $\exp\{\frac{-2i\pi nk}{N}\}$.

3.2 Exemple: TFD d'un signal sinusoidal

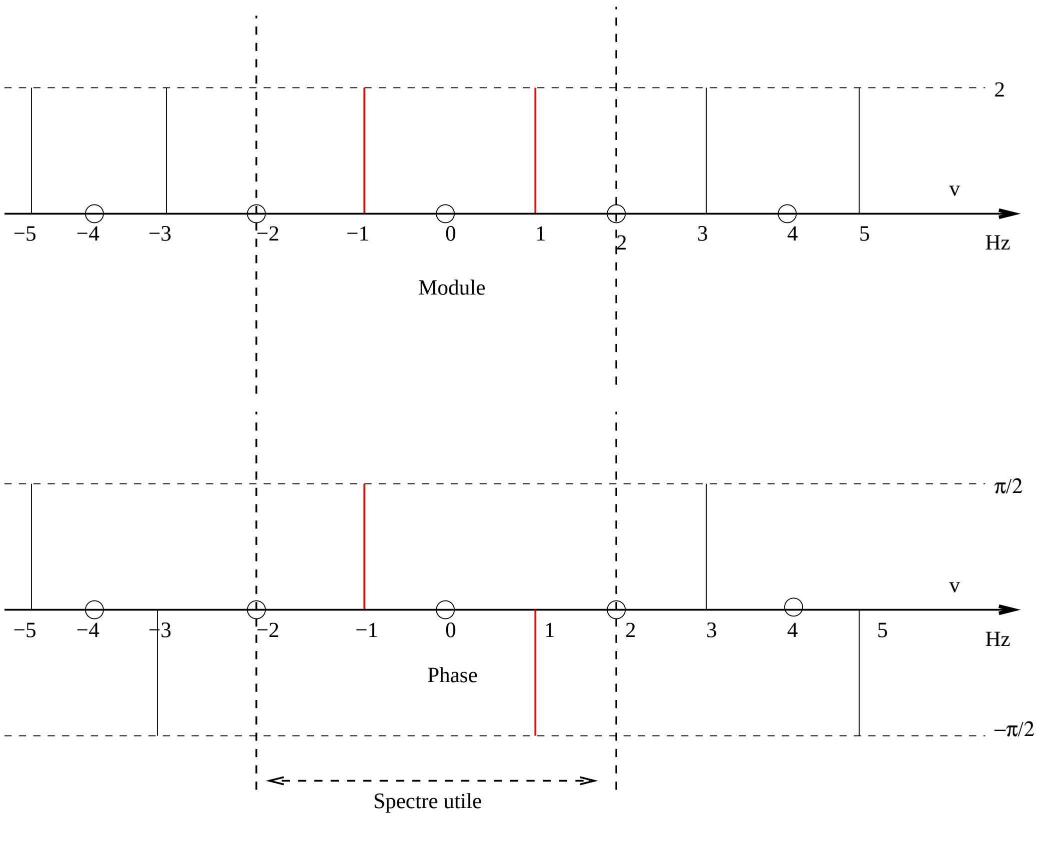
Soit le signal sinusoidal de fréquence unitaire échantillonné à une fréquence de 4 Hz. On cherche à calculer la TFD sur 4 points. Les échantillons temporels sont alors : $x(0) = 0$; $x(1) = 1$; $x(2) = 0$; $x(3) = -1$. Si l'on applique la formule précédente (8), on obtient :

$$\begin{pmatrix} X(k=0) \\ X(k=1) \\ X(k=2) \\ X(k=3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ -2i \\ 0 \\ 2i \end{pmatrix} \quad (8)$$

Les échantillons de TFD temporels sont respectivement $X(0) = 0; X(1) = -2i; X(2) = 0; X(3) = 2i$. Il reste maintenant à tracer le spectre en module et en phase des échantillons fréquentiels. La résolution fréquentielle est égale à

$$\Delta\nu = \frac{1}{NT_e} = 4 * \frac{1}{4} = 1Hz$$

, un échantillon sera tracé tous les Hertz. Notons que le signal de TFD est périodique par translation de période ν_e.



3.3 Fenêtrage et TFD d'un signal périodique

Deux aspects importants vont être étudiés dans le cadre du TP à savoir l'influence du fenêtrage et la condition à respecter afin de calculer la TFD d'un signal périodique. Ces deux aspects sont largement détaillés dans le cours de signaux numériques et dans les travaux dirigés. Il n'est pas question de refaire tout ce développement ici. Seuls les résultats principaux nécessaire au TP sont exposés. Ce dernier sera le cadre idéal afin de découvrir ces deux notions. La figure suivante présente les principales fenêtres disponibles ainsi que leurs principales propriétés.

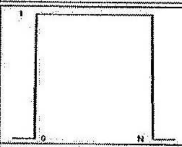
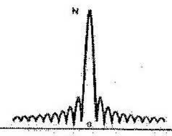
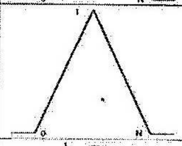
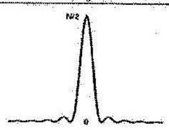
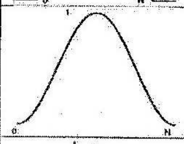
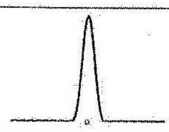
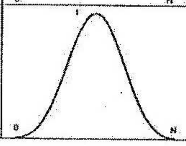
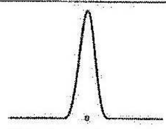
Nom	Equation $\forall n \in \{0 \ldots N-1\}$	Représentation temporelle	Représentation fréquentielle	Largeur lob.princ.	Amp. relative $\frac{lob.princ}{lob.sec.}$
Rectangulaire	$w[n] = 1$			$\frac{2}{N}$	-13 dB
Triangulaire	$w[n] = 1 - \frac{\left\|n-\left(\frac{N-1}{2}\right)\right\|}{\left(\frac{N-1}{2}\right)}$			$\frac{4}{N}$	-25 dB
Hamming	$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$			$\frac{4}{N}$	-41 dB
Blackman	$w[n] = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right)$			$\frac{6}{N}$	-57 dB

Table 3 *Différents types de fenêtres et leurs caractéristiques*

3.3.1 Fenêtrage

De manière à diminuer le nombre de raies parasites introduites par la TFD, on utilise un fenêtrage. En pratique, les échantillons temporels $x(nTe)$ sont multipliés par les échantillons de la fenêtre $h(nTe)$. La TFD est ensuite calculée:

$$X_{TFD}(\nu = \frac{k}{NT_e}) = \sum_{n=0}^{N-1} [x(nTe)h(nTe)] \exp -\{\frac{2ink}{N}\} \tag{9}$$

3.3.2 TFD d'un signal périodique

Soit T_0 la période du signal considérée, T_e la période d'échantillonnage et N le nombre de points de la TFD. La condition nécessaire et suffisante pour calculer la TFD d'un signal périodique afin de ne pas faire apparaître de raies parasites est qu'il faut échantillonner un nombre entier de période L le signal. En pratique, N, T_0, L, T_e vérifient donc la relation suivante:

$$NT_e = LT_0 \tag{10}$$

4 Bref rappel des principales fonctions Matlab

Matlab est un langage de programmation de haut niveau destiné au calcul scientifique. On le trouve dans les applications de : calcul, développement d'algorithmes, modélisation et simulation, analyse et visualisation de données, création de graphiques scientifiques, création d'application avec interfaces utilisateurs. Il existe un grand nombre de toolboxes, familles de fonctions étendant les fonctions de base de Matlab/Octave à un certain type de problème. On trouve ainsi des toolboxes dans les domaines du traitement du signal, de la commande des systèmes, des réseaux de neurones, de la logique floue, des ondelettes, de la simulation, etc..

Dans le cadre du TP, on utilisera principalement l'interpréteur en ligne disponible au lancement de Matlab/Octave lorsqu'une fenêtre s'ouvre et propose le prompt >>.

4.1 Création de variables et de matrices

La programmation en Matlab/Octave consiste en l'enchaînement d'expressions composées de variables, de nombres, de opérateurs et de fonctions. Il n'est pas nécessaire de déclarer le type ou la dimension des variables : Matlab choisit selon le contexte. Par exemple :

$>> a = 2.37$
Crée la variable a de dimension 1 et lui attribue la valeur 2.37.
Autre exemple : $>> a(2) = 1.5$

$a = 2.37 \quad 1.5$

Matlab/Octave permet de déclarer facilement des variables matricielles comme par exemple : $>> mamatrice = [1\ 2\ 3; 4\ 5\ 6\ ; 7\ 8\ 9\ ]$

$mamatrice =$

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

4.2 Opérateurs

Voici la liste des principaux opérateurs classiques définis dans Matlab/Octave applicables sur des nombres ou des matrices :

Addition	+
Soustraction	-
Multiplication	*
Division	/
Puissance	^
Division élément par élément (matrices)	./
Conjugué et transposé (matrice)	'
Multiplication élément par élément (matrices)	.*

4.3 Opérations sur les matrices

Les additions, soustractions transposées et autres inversions de matrices (taper A^{-1}) sont relativement faciles. Un élément de la matrice s'appelle avec les parenthèses. Par exemple $A(1,3)$ renvoie l'élément de la première ligne, troisième colonne. Attention : la colonne 0 n'existe pas. La concaténation de matrice est triviale, attention cependant à respecter les dimensions. (Exemple : $Z = [A, B; C, D]$) L'opérateur : s'utilise de différentes manières : pour créer des listes. Par exemple 1 : 4 renvoie [1234] et 1 : 2 : 5 renvoie [135].

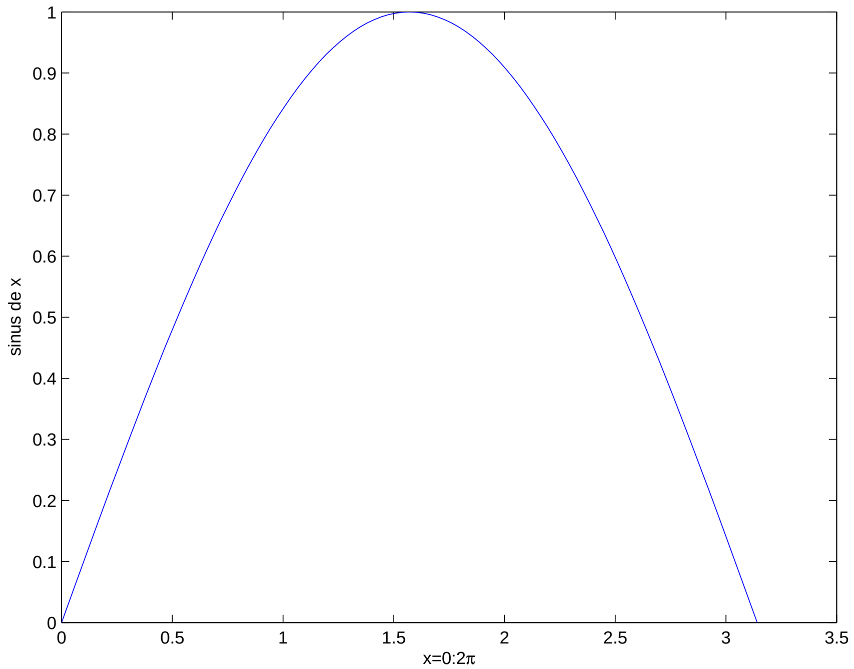
Pour faire appel à des sous-parties de matrices. Par exemple $A[:, 3]$ renvoie la troisième colonne de A et $A[1 : 2 : k, 3]$ renvoie les k premiers éléments impairs de la troisième colonne de A. Pour supprimer une ligne ou une colonne. Par exemple $X(2, :) = []$ supprime la deuxième ligne (et redimensionne donc la matrice).

4.4 Tracer des courbes

La séquence d'instruction suivante crée d'abord une figure puis place les commentaires sur les deux axes et enfin une ligne de titre :

$>> x = [0 : \pi/100 : \pi];$
$>> y = sin(x);$
$>> plot(x, y);$
$>> xlabel('x = 0 : 2\pi');$
$>> ylabel('sinus\ de\ x');$
$>> title('sinusoide', 'Fontsize', 12);$

La figure suivante apparaît dans une nouvelle fenêtre :



4.5 Aide

Matlab/Octave possède une bibliothèque importante de fonctions prédéfinies (ce qui fait d'ailleurs sa richesse). IL est recommandé d'utiliser la commande help suivi du nom de la fonction pour afficher l'aide spécifique à cette fonction.

5 Partie Matlab : TFD et résolution en fréquence d'un signal sinusoidal

La transformée de Fourier discrète correspond à un échantillonnage dans le domaine fréquentiel de la transformée d'un signal discret (TFD). Un algorithme de calcul de TFD est implémenté dans Matlab/Octave, il s'agit de la fonction *fft*. Dans tout l'exercice, on pose que la fréquence d'échantillonnage vaut 40960 Hz.

La résolution en fréquence correspond à l'aptitude à distinguer deux fréquences dans un signal. La résolution peut être quantifiée par l'écart maximal entre les fréquences de 2 sinusoides pour observer le signal résultant avec un écart de plus de 3 dB entre les deux maxima. La résolution est de l'ordre de F_e/N, valeur inverse de la durée d'observation NT_e. Du fait des lobes secondaires, la posibilité de séparer 2 sinusoides de fréquences proches va dépendre des amplitudes respectives des deux sinusoides.

- Quelle est la résolution fréquentielle sachant que les signaux comportent 2048 échantillons ?

- Générer effectivement un signal de 2048 points comportant deux sinusoides de fréquences f_0 et f_1, A_1 et A_0.

  1. F_0= 1000 Hz et F_1=15000 Hz, A_0=1 et A_1=1.

  2. F_0= 2000 Hz et F_1=2040 Hz, A_0=1 et A_1=1.

  3. F_0= 2412 Hz et F_1=2600 Hz, A_0=1 et A_1=0.025

  4. F_0= 2412 Hz et F_1=2568 Hz, A_0=1 et A_1=1

  Expliquer et commenter tous les résultats obtenus

- Recommencer les cas 3 et 4 en utilisant une fenêtre de Hamming.

  Expliquer et commenter tous les résultats obtenus