Tutoriel propriétés élémentaires de la transformée de Fourier discrète avec Scilab

1. Introduction

Ce document complète le tutoriel **Transformée de Fourier discrète.** Il explique comment implanter soi-même puis avec la fonction ad-hoc de Scilab, une transformée de Fourier discrète. Il en explore ensuite les principales propriétés mathématiques en les illustrant d'exemples.

On notera indifféremment ω ou w la racine *n*ième de l'unité définie par $\omega = e^{i.\frac{2\pi}{n}}$, $n \geq 2$ entier naturel. On rappelle que la transformée de Fourier discrète (TFD) d'une séquence $(x_0, x_1, \ldots, x_{n-1})$ de nombres complexes est la séquence $(X_0, X_1, \ldots, X_{n-1})$ définie par : $X_l = \sum_{k=0}^{n-1} x_k.\omega^{-k.l}$ pour l entre 0 et $n-1$.

2. Mise en œuvre d'une transformée discrète (TFD)

Une TFD transforme linéairement une séquence de n nombres en une nouvelle séquence de même taille. L'opération peut donc se coder à l'aide d'une matrice - dite de TFD - agissant sur le vecteur $(x_0, x_1, \ldots, x_{n-1})$[1] et renvoyant le vecteur $(X_0, X_1, \ldots, X_{n-1})$[2]. La matrice de TFD a pour coefficient (k, l) le nombre $\omega^{-(k-1)(l-1)}$[3].

Quelques commandes Scilab utiles pour créer et manipuler les matrices

On peut créer une matrice sous Scilab en explicitant ses coefficients, le point virgule faisant office de séparateur entre lignes, la virgule entre colonnes ; ainsi :

```
M = [1,2,3 ;4,5,6]
```

crée une matrice à 2 lignes et 3 colonnes. `M(2,1)` retourne le coefficient de la 1ere ligne et 2ème colonne : 4.

Appliquer la matrice M au vecteur colonne `[1 ;1 ;1]` a un sens, au vecteur ligne `[1,1]` n'en a pas. La syntaxe attendue est donc :

```
M * [1 ;1 ;1]
```

On peut définir une matrice de taille donnée :

```
M = zeros(2,3)
```

Cette instruction produit une matrice à 2 lignes et 3 colonnes, dont les coefficients sont initialisés à 0.

On peut concaténer deux vecteurs lignes ou, ici colonnes :

```
v1 = [1 ;2 ;3]
v2 = [4 ;5]
v=[v1 ;v2]
```

[1] À disposer en colonne le temps du calcul.
[2] Idem
[3] Lignes et colonnes sont indexées à partir de 1 sous Scilab.

Activité :

1. **Construire la matrice de TFD quand $n = 5$.**
2. **L'appliquer à $x = [1\,;2\,;3\,;4\,;5]$. Comparer au résultat obtenu à l'aide de la fonction `fft`.**
3. **Border x de 5 zéros et former $= [1\,;2\,;3\,;4\,;5\,;0\,;0\,;0\,;0\,;0]$. Lui appliquer la TFD avec $n = 10$. C'est un procédé connu sous le nom de *Zero padding.* Que constate-t-on ?**

4. Dupliquer x, former $x2 = [1\,;2\,;3\,;4\,;5\,;1\,;2\,;3\,;4\,;5]$, lui appliquer la TFD. Que constate-t-on ?

Exercices 1 et 2 :

On construit la matrice M de TFD en imbriquant deux boucles inconditionnelles :

```
M = zeros(5,5);
w = exp(%i*2*%pi/5);
for i =1:5
    for j=1:5
        M(i,j) = w^(-(i-1)*(j-1))
    end
end
```

On applique la matrice M au vecteur colonne x :

```
M*x
```

L'algorithme de FFT implémente un calcul optimisé de la TFD. La fonction clé-en-main `fft` renvoie effectivement le résultat attendu (reformaté en ligne dans un souci de place, le suffixe .' ayant pour effet de transposer), à savoir :

```
fft(x.')
ans=15.  - 2.5 + 3.4409548i  - 2.5 + 0.8122992i  - 2.5 - 0.8122992i  - 2.5 - 3.4409548i
```

Exercices 2 et 3 :

On définit un vecteur nul, on le juxtapose au vecteur x, on en forme la TFD (de taille 10) :

```
z = zeros(1,5) ;  x1 = [x;z] ; fft(x1) ;
ans  =
    15.
  - 1.736068 - 10.771892i
  - 2.5 + 3.4409548i
    2.736068 - 2.5428988i
  - 2.5 + 0.8122992i
    3.
  - 2.5 - 0.8122992i
    2.736068 + 2.5428988i
  - 2.5 - 3.4409548i
  - 1.736068 + 10.771892i
```

Une valeur sur deux vient de la TFD du vecteur tronqué à 5 coordonnées. Le *Zero Padding* temporel interpole la TFD en fréquence

On réplique le vecteur x, on forme la TFD (de taille 10)

```
x2 = [x;x] ; fft(x2) ;
ans  =
         30.
         0
       - 5. + 6.8819096i
         0
       - 5. + 1.6245985i
         0
       - 5. - 1.6245985i
         0
       - 5. - 6.8819096i
         0
```

Une valeur sur deux vient de la TFD du vecteur tronqué à 5 coordonnées (au coefficient multiplicatif 2 près). Les autres sont nulles.

3. Propriétés mathématiques élémentaires de la transformée de Fourier discrète (TFD)

Propriété : la TFD est un opérateur linéaire. Autrement dit, pour $(x_0, x_1, \dots, x_{n-1})$ et $(x'_0, x'_1, \dots, x'_{n-1})$ deux vecteurs complexes et a et b deux complexes, on a : TFD $((ax_k + bx'_k))$ = aTFD $((x_k))$ + bTFD$((x'_k))$

La transformée de Fourier discrète inverse (TFD^{-1}) d'une séquence $(X_0, X_1, \dots, X_{n-1})$ de nombres complexes est la séquence $(\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{n-1})$ définie par : $\tilde{x}_k = \frac{1}{n}\sum_{l=0}^{n-1} X_l \,.\, \omega^{+k.l}$ pour k entre 0 et $n-1$.

Propriété : la transformée de Fourier est réversible. Précisément, on a : $(x_k) = (\widetilde{(\tilde{x}_k)}) = \mathrm{TFD}^{-1}(\mathrm{TFD}\,((x_k)))$

Activité :

1. Démontrer cette dernière propriété pour la séquence [1 ;0 ;0 ;… ;0]. Comment s'en déduirait le résultat général ?

2. Vérifier la propriété sur la séquence [1 ;2 ;3 ;4 ;5] avec le logiciel.

$x_k = 1$ si $k = 0, 0$ sinon

Exercice 1. On calcule $X_l = \sum_{k=0}^{n-1} x_k.\,\omega^{-k.l}$ pour l entre 0 et $n-1$. Par suite $\tilde{x}_k = \frac{1}{n}\sum_{l=0}^{n-1} \omega^{k.l}$. La somme est en progression géométrique de raison ω^k. Elle est nulle quand $k \neq 0$ puisque $\left(\omega^k\right)^n = 1$, égale à 1 quand $k = 0$. La formule de réciprocité est ici établie. On procède de même avec les séquences [0 ;1 ;0 ;…0] ; [0 ;0 ;1 ;0 ;… ;0] ; … ; [0 ;0 ;… ;0 ;1]. On conclut par linéarité d'une somme.

Exercice 2. La vérification est aisée :

```
ifft(fft([1,2,3,4,5]))
 on obtient bien
    1.    2.    3.    4.    5.
```

Propriété : $\overline{\mathrm{TFD}}\big((\bar{x}_k)\big) = n.\,\mathrm{TFD}^{-1}((x_k))$.

Propriété (formule de Bessel) : $\sum_{k=0}^{n-1}|x_k|^2 = \frac{1}{n}\sum_{l=0}^{n-1}|X_l|^2$

Activité : vérifier la formule de Bessel sur la séquence [1 ;2 ;3 ;4 ;5]. En rappeler l'interprétation temps-fréquence

Ici encore, la vérification est aisée. La somme de gauche provient du produit de la matrice ligne transconjuguée de x et de la matrice colonne x. Celle de droite se traite à l'avenant. D'où les instructions :

```
x'*x
 ans  =
    55.
X = fft(x) ; (1/5) *X'*X
ans  =
    55. + 2.220D-16i
```

x ' code la transconjuguée et x . ' la transposée

Très légère approximation de calcul sur la partie imaginaire

La formule de Bessel donne deux moyens d'accéder à l'énergie d'une séquence, l'un en restant dans le domaine des temps, l'autre en utilisant celui des fréquences.

La formule qui suit met en évidence l'effet d'un décalage fréquentiel (circulaire) : pour obtenir la séquence $(X_{n-1}, X_0, \dots, X_{n-2})$ au lieu de la séquence $(X_0, X_1, \dots, X_{n-1})$ il faut calculer la transformée Fourier discrète de $(x_0, x_1.\,\omega, x_2.\,\omega^2, \dots, x_{n-1}.\,\omega^{n-1})$ au lieu de celle de $(x_0, x_1, \dots, x_{n-1})$. On constate un phénomène voisin lors d'une transposition analogique, à ceci près que le décalage n'est pas circulaire. Énonçons :

Propriété : $\mathrm{TFD}(x_0, x_1.\omega, \dots, x_{n-1}.\omega^{n-1}) = (X_{n-1}, X_0, \dots, X_{n-2})$ où $\omega = e^{i.\frac{2\pi}{n}}$.

Activité :

1. **Vérifier cette formule sur la séquence [1 ;2 ;3 ;4 ;5]**
2. **Comment faire tourner (sur elle-même) la séquence des X_k dans l'autre sens ?**
3. **Déterminer la TFD de la séquence [5 ;1 ;2 ;3 ;4]. Le résultat obtenu était-il prévisible ?**

Exercice 1. Observons la rotation de la transformée de Fourier :

```
u=[1 ;w ;w^2 ;w^3 ;w^4] ; fft(x .* u)
ans  =
 - 2.5 - 3.4409548i
    15. - 3.164D-15i
  - 2.5 + 3.4409548i
  - 2.5 + 0.8122992i
  - 2.5 - 0.8122992i
```

L'opérateur .* multiplie coordonnées à coordonnées les vecteurs x et u

Très légère approximation de calcul

Exercice 2. En répétant l'opération une nouvelle fois, c'est-à-dire en multipliant point à point [1 ;2 ;3 ;4 ;5] par [1 ; ω ; ω^2 ; ω^3 ; ω^4] et encore par ce même [1 ; ω ; ω^2 ; ω^3 ; ω^4], on fera tourner deux fois la transformée. En faisant cela 4 fois, on décale la transformée 4 fois d'un rang vers la droite, donc d'un rang vers la gauche. Bref, on aura ainsi multiplié [1 ;2 ;3 ;4 ;5] par [1 ; ω^4; $(\omega^2)^4$; $(\omega^3)^4$; $(\omega^4)^4$]. Or $\left(\omega^k\right)^4 = (\omega^4)^k = (\omega^{-1})^k = \omega^{-k}$. Conclusion générale : pour faire tourner la transformée d'un cran vers la gauche, on multiplie point à point la séquence de départ par [1 ; ω^{-1} ; ω^{-2} ; ω^{-3} ;…; $\omega^{-(n-1)}$].

Exercice 3. Interrogeons Scilab :

```
fft([5 ;1 ;2 ;3 ;4])
ans  =
    15.
    2.5 + 3.4409548i
    2.5 + 0.8122992i
    2.5 - 0.8122992i
    2.5 - 3.4409548i
```

La transformée a été déphasée linéairement, ici elle a été multipliée point à point par (1 ; ω^{-1} ; ω^{-2} ; ω^{-3} ;ω^{-4}),

Le résultat observé était en partie prévisible, en commençant par interpréter la séquence [5 ;1 ;2 ;3 ;4] comme la décalée de la transformée inverse de la transformée [1 ;2 ;3 ;4 ;5].

4. Convolution périodique (ou circulaire) et transformée de Fourier discrète

Soit la séquence $x = (x_0, x_1, \dots, x_{n-1})$ de taille n. On souhaite la filtrer linéairement, c'est-à-dire la convoluer[4], par la séquence de même taille $h = (h_0, h_1, \dots, h_{n-1})$[5]. Pour simplifier[6], on considérera que cette seconde séquence ne comporte que trois coefficients significatifs : h_0, h_1, h_2, les suivants étant nuls.

La convolution *apériodique* de x par h génère la séquence $y = x * h$ de taille $n + 2$ définie par :

$y_0 = x_0.h_0$ — Phase de mise en route du processus

$y_1 = x_1.h_0 + x_0.h_1$

$y_2 = x_2.h_0 + x_1.h_1 + x_0.h_2$

$y_3 = x_3.h_0 + x_2.h_1 + x_1.h_2$ — Plein régime

[4] Il s'agit ici d'une convolution *apériodique.*

[5] La séquence $(h_0, h_1, \dots, h_{n-1})$ s'appelle la réponse impulsionnelle du filtre.

[6] Tout ceci se prête à généralisation.

....

$y_{n-1} = x_{n-1}.h_0 + x_{n-2}.h_1 + x_{n-3}.h_2$

$y_n = x_{n-1}.h_1 + x_{n-2}.h_2$

$y_{n+1} = x_{n-1}.h_2$ Extinction du processus

Notons qu'on réalise les mêmes calculs quand on multiplie les polynômes $x_0 + x_1X + x_2X^2 + \cdots + x_{n-1}X^{n-1}$ et $h_2X^2 + h_1X + h_0$.

La convolution *périodique* de x par h produit, elle, la séquence $\tilde{y} = x \otimes h$ de taille n définie par :

$\tilde{y}_0 = x_0.h_0 + x_{n-1}.h_1 + x_{n-2}.h_2$

$\tilde{y}_1 = x_1.h_0 + x_0.h_1 + x_{n-1}.h_2$

$\tilde{y}_2 = x_2.h_0 + x_1.h_1 + x_0.h_2$

$\tilde{y}_3 = x_3.h_0 + x_2.h_1 + x_1.h_2$

....

$\tilde{y}_{n-1} = x_{n-1}.h_0 + x_{n-2}.h_1 + x_{n-3}.h_2$

Notons qu'on réalise les mêmes calculs en écrivant les nombres $x_0, x_1, \ldots, x_{n-1}$ sur un anneau ; les nombres h_2, h_1, h_0 sur un autre ; en « enfilant » les deux anneaux, en faisant tourner l'un par rapport à l'autre et en multipliant les termes en phase.

Les termes d'indices 2 à $n-1$ des suites y et $\tilde{y}$ sont identiques. Cette heureuse coïncidence est à la base de la méthode de filtrage dite d'*Overlap and Save.*

Propriété : $x \otimes h = TFD^{-1}\big(TFD(x).*TFD(h)\big)$

Comprendre un produit coordonnée à coordonnée

Activité :

1. Valider cette formule avec les séquences [1 ;2 ;3 ;4 ;5 ;6 ;7 ;8 ;9 ;10] et [1 ;0,5 ;0,1 ;0 ;0 ;0 ;0 ;0 ;0 ;0]

2. Notez-vous une différence entre $\boldsymbol{x \otimes h}$ et $\boldsymbol{h \otimes x}$? Était-ce prévisible ? On parle de commutativité.

3. Notez-vous une différence entre $\boldsymbol{(x \otimes h) \otimes h}$ et $\boldsymbol{x \otimes (h \otimes h)}$? Était-ce prévisible ? On parle d'associativité.

Exercice 1. Les commandes Scilab sont très simples :

```
x=[1;2;3;4;5;6;7;8;9;10]; h =[1;0.5;0.1;0;0;0;0;0;0;0];
ifft(fft(x).*fft(h))    .'
 ans  =
    6.9    3.5    4.1    5.7    7.3    8.9    10.5    12.1    13.7    15.3
```

La phase de mise en route une fois passée, on obtient les résultats attendus. Par exemple :

$y_2 = 3 \times 1 + 2 \times 0{,}5 + 1 \times 0{,}1 = 4.1 = \tilde{y}_2$

Exercices 2 et 3. On ne note aucune différence. Cela tient, entre autre, à ce que ifft($X.*H$)=ifft($H.*X$) où X =fft(x) et H =fft(h), puisque $X.*H = H.*X$.

L'équipe qui a rédigé ce document ressource était composée de :

Cabane Robert, IGEN, groupe des mathématiques

Dubouloz Georges, IA-IPR de mathématiques, académie de Grenoble

Fleurant Sandrine, IA-IPR de mathématiques, académie de Nantes
Zayana Karim, IGEN, groupe des mathématiques