

Rapport de MT12 : Techniques mathématiques de l'ingénieur
UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

Printemps 2016

Alexandre BALLET et Simon LAURENT

Sujet du rapport :

Transformée de Fourier discrète

Département des étudiants :

Génie Informatique

Professeur :

M. Djalil KATEB

Table des matières

1	Biographies	4
1.1	Claude Shannon	4
1.2	Siméon Denis Poisson	5
1.3	Harry Nyquist	5
2	Présentation de la transformée de Fourier et ses applications	7
2.1	Définition	7
2.2	Applications	7
3	Définition de la DFT	8
3.1	Définition	8
3.2	DFT d'une fonction	11
4	Une famille de fonctions tests	13
4.1	Procédure Scilab	13
4.2	Signal d'origine	15
5	Applications de la DFT	18
5.1	Efficacité de la FFT et Approximation des coefficients de Fourier	18
5.2	Résolution fréquentielle de signaux	22
6	DFT et convolution	28

Chapitre 1

Biographies

1.1 Claude Shannon

Claude Elwood Shannon (30 avril 1916 à Petoskey, Michigan - 24 février 2001 à Medford, Massachusetts) était un ingénieur et chercheur en génie électrique, mathématicien et cryptographe américain. Il est considéré comme le "père de la théorie de l'information", pour avoir publié en 1948 un article phare : "A Mathematical Theory of Communication".

Shannon a passé ses 16 premières années à Gaylord, où il montrait déjà son attirance pour la mécanique et l'électricité. Il construisit des maquettes d'avions, des bateaux téléguidés et même un système de télégraphe sans-fils pour relier sa maison à celle d'un amis située à environ 1km. En 1932, Shannon entra à l'Université du Michigan, où il fut introduit aux travaux de Georges Boole.

En 1936, il intégra le MIT où il travailla sur l'analyseur différentiel de l'inventeur Vannevar Bush, un ancêtre de l'ordinateur. Il appliqua l'algèbre de Boole aux circuits électriques et publia un article prouvant que les circuits de commutation qu'il avait créé pouvaient résoudre tous les problèmes que l'algèbre de Boole pouvait résoudre. L'une de ses applications fut l'additionneur 4 bits.



1.2 Siméon Denis Poisson

Siméon Denis Poisson (21 juin 1781 - 25 avril 1840) était un mathématicien, géomètre et physicien français. Né à Pithiviers, il est le fils du soldat Siméon Poisson.

En 1798, il entra à l'École Polytechnique de Paris et ses professeurs lui laissèrent vite le choix de ses études.

Joseph Louis Lagrange, auxquels cours il assistait sur la théorie des fonctions, reconnu son jeune talent et devenu son ami. Quant à Pierre-Simon Laplace, il considérait Poisson comme son fils.

En 1802, il fut nommé professeur suppléant, puis professeur en 1806, succédant à Jean Baptiste Joseph Fourier, lequel avait été envoyé à Grenoble par Napoléon.



Siméon Denis Poisson

1.3 Harry Nyquist

Harry Theodor Nyquist (7 février 1889 - 4 avril 1976) était un ingénieur en électronique américain. Né en Suède, il émigra aux États-Unis en 1907. Il entra à l'université de Dakota du Nord en 1912, et fut diplômé en 1915 en ingénierie électrique. Il reçut un Ph.D. en physique à Yale en 1917.

Il travailla chez AT&T dans le Département de Développement et de Recherche entre 1917 et 1934, puis continua quand cela devint le Laboratoire Téléphonique de Bell, jusqu'à son départ à la retraite en 1954. Ses premiers travaux théoriques sur la détermination des conditions de la bande passante pour transmettre l'information ont conduit aux avancées de Claude Shannon, ce qui a mené au développement de la théorie de l'information. En particulier, Nyquist détermina que le nombre de pulsations indépendantes par unités de temps qui pouvaient être envoyées

par le télégraphe était limité à deux fois la bande passante du canal. Cette règle est le principe du théorème d'échantillonnage de Nyquist-Shannon.



Harry Nyquist

Chapitre 2

Présentation de la transformée de Fourier et ses applications

2.1 Définition

La Transformée de Fourier décompose une fonction du temps (un signal) en fréquences qui le composent, d'une manière similaire à l'expression d'un accord musical comme l'amplitude de ses notes. Le terme *Transformée de Fourier* fait référence à la fois à la représentation du domaine de fréquences et à l'opérateur mathématique qui associe la représentation du domaine de fréquences à une fonction du temps.

"La Transformée de Fourier est une extension des Séries de Fourier lorsque la période de la fonction représentée est allongée et autorisée à tendre vers l'infini " (Taneja, 2008).

2.2 Applications

La Transformée de Fourier peut être utilisée pour résoudre des équations différentielles, car il est souvent plus facile de trouver la transformée de Fourier de la solution que la solution directe. En effet, la différenciation d'une transformée de Fourier correspond à une multiplication par un polynôme. Il suffit ensuite d'appliquer la transformée de Fourier inverse pour trouver la solution.

Elle est également utilisée en résonance magnétique nucléaire et dans d'autres types de spectroscopies comme le traitement d'IRM ou dans la spectrométrie de masse.

Chapitre 3

Définition de la DFT

3.1 Définition

Definition 1. On appelle transformée de Fourier discrète de la suite (y_k) , $k = 0, \text{ldots}, N - 1$ la suite (z_k) définie par

$$z_k = \frac{1}{N} \sum_{p=0}^{N-1} y_p \omega^{pk}$$

où $\omega = e^{-2i\frac{\pi}{N}}$. On notera $z[k] = DFT[f]k$, $k = 0, \dots, N - 1$.

Nous cherchons à montrer que

$$\sum_{k=0}^{N-1} \omega^{k(p-p')} = \begin{cases} 0 & \text{si } p \neq p' \\ N & \text{si } p = p' \end{cases} \quad (3.1)$$

D'une part, posons $p = p'$:

$$\begin{aligned} \sum_{k=0}^{N-1} \omega^{k(p-p')} &= \sum_{k=0}^{N-1} 1 \\ &= N \end{aligned}$$

D'autre part, posons $p \neq p'$:

$$\begin{aligned}
\sum_{k=0}^{N-1} \omega^{k(p-p')} &= \sum_{k=0}^{N-1} (\omega^{p-p'})^k \\
&= \frac{1 - \omega^{(p-p')^k}}{1 - \omega^{p-p'}} \\
&= \frac{1 - e^{-2i\pi(p-p')}}{1 - e^{-2i\frac{\pi}{N}(p-p')}} \\
&= \frac{1 - e^{-2i\pi k}}{1 - e^{-2i\frac{\pi}{N}(p-p')}} \quad , \text{ où } k \in \mathbb{Z} \\
&= \frac{1 - 1}{1 - e^{-2i\frac{\pi}{N}(p-p')}} \\
&= 0
\end{aligned}$$

Nous cherchons ensuite à montrer que $B = \sqrt{N}A$ est une matrice unitaire, c'est-à-dire :

$$\overline{B^T} B = I \quad , \text{ où } I \text{ est la matrice identité} \quad (3.2)$$

$$\begin{aligned}
B &= \sqrt{N} \frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \dots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)^2} \end{pmatrix} \\
\overline{B^T} B &= \frac{\sqrt{N}}{N} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & (\overline{\omega}) & (\overline{\omega})^2 & \dots & (\overline{\omega})^{N-1} \\ 1 & (\overline{\omega})^2 & (\overline{\omega})^4 & \dots & (\overline{\omega})^{2(N-1)} \\ 1 & (\overline{\omega})^3 & (\overline{\omega})^6 & \dots & (\overline{\omega})^{3(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & (\overline{\omega})^{N-1} & (\overline{\omega})^{2(N-1)} & \dots & (\overline{\omega})^{(N-1)^2} \end{pmatrix} \frac{\sqrt{N}}{N} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \dots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)^2} \end{pmatrix} \\
&= \frac{1}{N} (1 + (\overline{\omega})^{p-1} (\omega)^{p'-1} + \dots + (\overline{\omega})^{(N-1)(p-1)} (\omega)^{(N-1)(p'-1)}) \\
&= \frac{1}{N} (1 + (\omega)^{p'-p} + \dots + (\omega)^{(N-1)(p'-p)}) \\
&= \frac{1}{N} \sum_{k=0}^{N-1} \omega^{k(p'-p)} \\
&= \frac{1}{N} N, \text{ d'après (3.1)} \\
&= 1
\end{aligned}$$

D'où $B = \sqrt{N}A$ est une matrice unitaire.

Nous cherchons à présent A^{-1} :

$$\begin{aligned}
\overline{B^T} B &= I \\
N \overline{A^T} A &= I \\
A^{-1} &= \overline{N A^T} \\
A^{-1} &= \left(\begin{array}{ccccc} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \dots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)^2} \end{array} \right)^T
\end{aligned}$$

A^{-1} est la matrice associée à la transformation inverse de la DFT, que l'on peut retrouver :

$$z'_k = \sum_{p=0}^{N-1} y_p \omega^{-pk} \quad , \quad \text{où } \omega = e^{-2i\frac{\pi}{N}} \quad (3.3)$$

3.2 DFT d'une fonction

Soit f définie sur une période T (périodique de période T) et soit (y_k) une suite d'échantillons de f en N points uniformément répartis sur une période. La DFT de f d'ordre N est l'application qui associe à la suite $(y_k)_{k=0,\dots,N-1}$ la suite constituée de la DFT appliquée au vecteur $y = (y_k)_{k=0,\dots,N-1}$. On note $(DFT[f](k))_{0,\dots,N-1}$ la suite obtenue et $iDFT$ sa réciproque.

Nous cherchons à montrer que l'on peut approcher le coefficient de Fourier c_n à partir de l'approximation de l'intégrale la définissant par une somme de Riemann.

D'après la définition de c_n on a :

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-2i\pi \frac{n}{T} t} dt$$

Si nous discrétisons $f(t)$ en divisant une période T en N segments, on obtient :

$$\begin{aligned}
c_n &= \frac{1}{T} \int_0^T f(t) e^{-2i\pi \frac{n}{T} t} dt \\
&= \frac{1}{T} \sum_{k=0}^{N-1} f(t_k) e^{-2i\pi \frac{n}{T} k \frac{T}{N}} \frac{T}{N} \\
&= \frac{1}{N} \sum_{k=0}^{N-1} y_k e^{-2i\pi n \frac{k}{N}} \\
&= \frac{1}{N} \sum_{p=0}^{N-1} y_p e^{-2i\pi n \frac{p}{N}} \\
&= \frac{1}{N} \sum_{p=0}^{N-1} y_p \omega^{np}
\end{aligned}$$

D'où

$$c_k = \frac{1}{N} \sum_{p=0}^{N-1} y_p \omega^{pk}$$

Nous pouvons en déduire que la DFT d'une suite (y_k) est équivalent au coefficient de Fourier c_k de cette série.

Chapitre 4

Une famille de fonctions tests

4.1 Procédure Scilab

```
1  _T = 1
   _alpha=1/3

function [y] = f(x)
6    x=[pmodulo(x + _T/2,_T) - _T/2]
    y = zeros(x)
    for i=1:length(x)
        if (x(i) < _T*_alpha/2 & x(i)>-_T*_alpha/2) then
11            y(i)=1
        else
            y(i) = 0
        end
    end
16 endfunction

N = 32

21 x=[-3*_T/2:_T/N:3*_T/2]
   scf(0)
   clf
   plot(x,f(x),'xr')
```

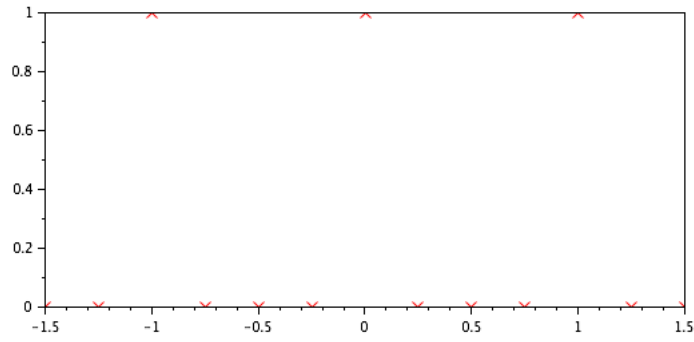


FIGURE 4.1 – Représentation pour $N=4$

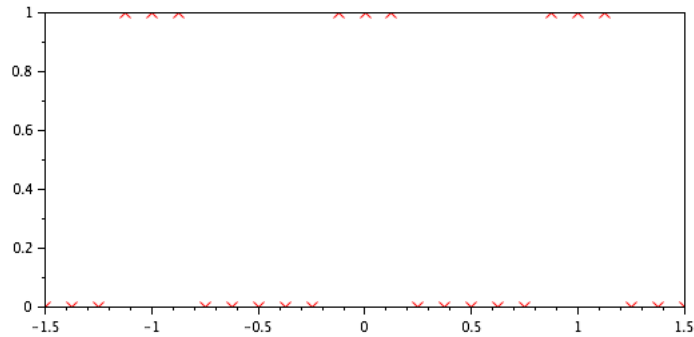


FIGURE 4.2 – Représentation pour $N=8$

L'échantillonnage consiste à transmettre un signal en capturant des valeurs à périodes régulières. Ce dernier produit une suite de valeurs discrètes.

La première étape de ce processus est l'échantillonnage du signal :

L'échantillonnage d'un signal $f(t)$ analogique représenté par une fonction $f(t)$ consiste à construire, à partir de $f(t)$, un signal à temps discret obtenu en mesurant la valeur de $f(t)$ toutes les t_k secondes tel que : $y_k = f(t_k)$, avec $k = 0 \dots, N - 1$ et N le nombre d'échantillons de $f(t)$. On pose $t_k = \frac{ka}{N}$

La seconde étape est la détermination de la fréquence :

La cadence à laquelle les valeurs sont capturées est la fréquence d'échantillonnage (taux d'échantillonnage). L'objectif de l'échantillonnage est la transmission de l'information codée dans un signal. La question du choix de la fréquence d'échantillonnage se pose immédiatement : Si la fréquence d'échantillonnage est trop faible, les acquisitions seront trop espacées et si le signal comporte des détails pertinents entre deux positions de capture, ceux-ci seront perdus ; Plus la fréquence d'échantillonnage est élevée, et plus la transmission coûte en puissance de traitement, en capacités de transmission et en espace de stockage. Il faut donc bien choisir sa fréquence F_e d'échantillonnage. Soit F_e la fréquence d'échantillonnage il faut donc que le signal

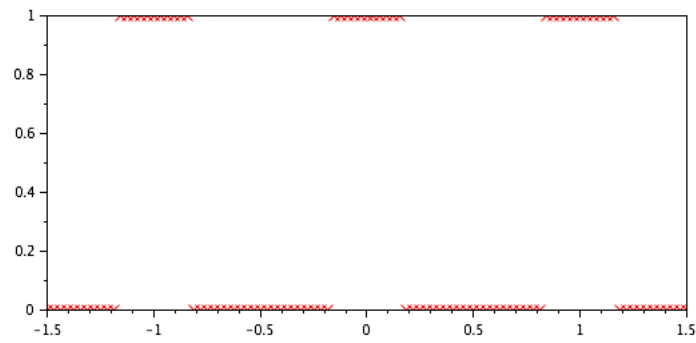


FIGURE 4.3 – Représentation pour $N=32$

soit échantillonné uniformément avec un intervalle de temps égal à $\frac{a}{N}$.

4.2 Signal d'origine

Suite à la modélisation de la fonction ci dessus. on peut utiliser la FFT et son inverse IFFT afin de trouver le signal d'origine. On reprend donc la fonction initiale :

```

1 // Paramètres
  _T = 1
  _alpha=1/3
  N = 256
  x=[-3*_T/2:_T/N:3*_T/2]
6
  // Fonction initiale
  function [y] = f(x)
    x=[pmodulo(x + _T/2,_T) - _T/2]
    y = zeros(x)
11    for i=1:length(x)
        if (x(i) < _T*_alpha/2 & x(i)>_T*_alpha/2) then
            y(i)=1

            else
16                y(i) = 0
            end
        end
    endfunction
21 // Représentation graphique
  scf(0)

```

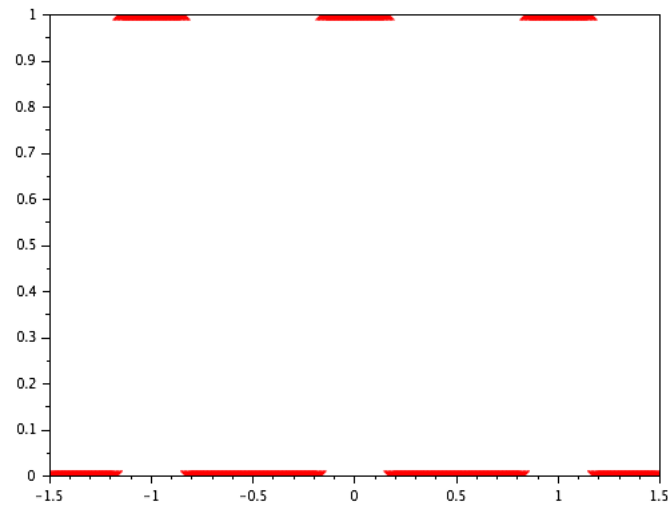


FIGURE 4.4 – *Représentation pour $N=256$*

```
clf
plot(x,f(x),'xr')
```

On applique ensuite la transformée de Fourier sur la fonction

```
1 //FFT et représentation
F=fft(f(x))
scf(0)
clf
plot(x,F)
```

On obtient alors le spectre du signal :

Puis la transformée de la fonction inverse

```
//Signal d'origine
F2=ifft(F)
scf(0)
clf
5 plot(x,F2)
```

On a le résultat suivant :

On remarque que l'on obtient à nouveau la fonction créneau de départ.

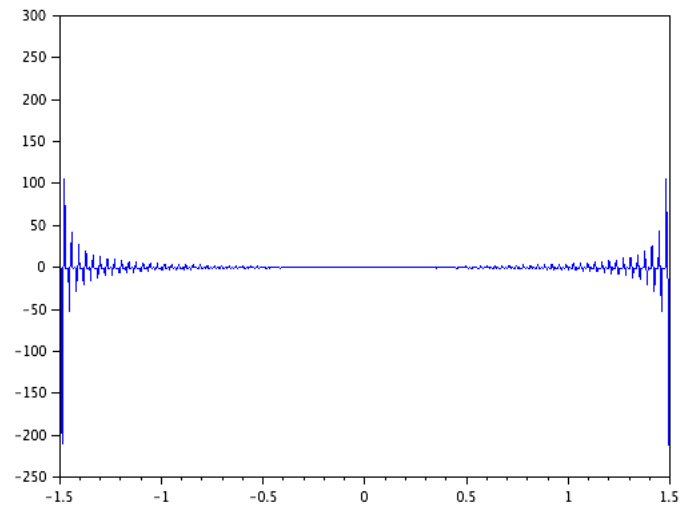


FIGURE 4.5 – FFT pour $N=256$

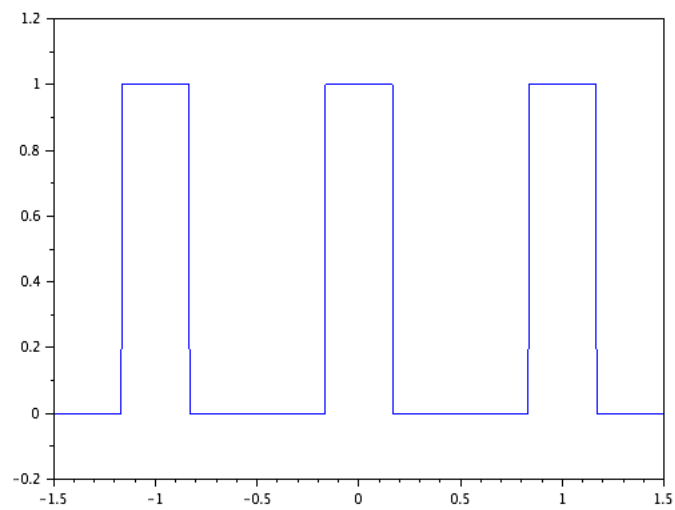


FIGURE 4.6 – $iFFT$ pour $N=256$

Chapitre 5

Applications de la DFT

5.1 Efficacité de la FFT et Approximation des coefficients de Fourier

Dans cette partie nous allons calculer la transformé de Fourier discrète à partir de la fonction discretisée :

```
//Définition de la DFT
a=-1/2
b=1/2
function S=F(f,p,N)
5    // Initialisation à 0
    S=0
    //Pour N échantillons on calcul les valeurs de la dft
    for k=1:N
        S=S+f(a+(k-1)*(b-a)/N)*exp(-2*i*pi*p*(k-1)*(b-a)/N)
10 end
    //On divise la DFT par N
    S=(1/N)*S
endfunction
```

On compare ensuite le résultat avec la FFT

```
//Division par N afin de comparer les résultats
2 fastFourier = fft(x)/N
```

Enfin on analyse le temps d'exécution des deux fonctions grâce au tic() toc() (timer) de Scilab. On obtient :

```
//Calcul de la DFT
```

```

tic()
3 for p=1:N
    y(p)=F(f,p,N)
end
temps=toc()

8 //Comparer fft et DFT
tic()
fastFourier=fft(x)/N;
temps2=toc()

```

```

temps =
    23.531
temps2 =
    0.001

```

Ainsi pour $N=256$, on montre que la FFT est 10^4 fois plus rapide que la DFT. Ceci est du à la complexité des algorithmes. Pour la DFT on est en $O(n^2)$ alors que pour la FFT on est en $O(n * \log(n))$. Il nous reste maintenant à comparer les résultats de coefficients.

```

//Calcul des coefficients Cp
sol1(1)=1/3
sol2(N)=sol1(1)
4 for p=2:N
    sol1(p)=sin(p*1/3*pi)/(p*pi)
    sol2((N+1)-p)=sol1(p)
end
Res=[sol2' sol1']

```

Cependant les coefficients réels de la série de Fourier ont des valeurs sur \mathbb{Z} . Et dans notre cas, nous n'avons que des valeurs pour $n \in \mathbb{N}$. Ainsi on corrige en utilisant la propriété que :

$$\begin{cases} Y_n, 0 \leq n \leq \frac{N}{2} - 1 \\ Y_{n+N}, \frac{-N}{2} \leq n < 0 \end{cases}$$

On calcule donc C_n sur \mathbb{Z} en séparant les négatifs dans un vecteur et les positifs dans un autre. On obtient alors le résultat suivant :

Les figures ci dessus montrent beaucoup de similarité. En effet pour N grand, on a des résultats correctes. Cependant lorsque N est inférieur à 128, les résultats sont médiocres. Comme le montre les graphiques ci dessous :

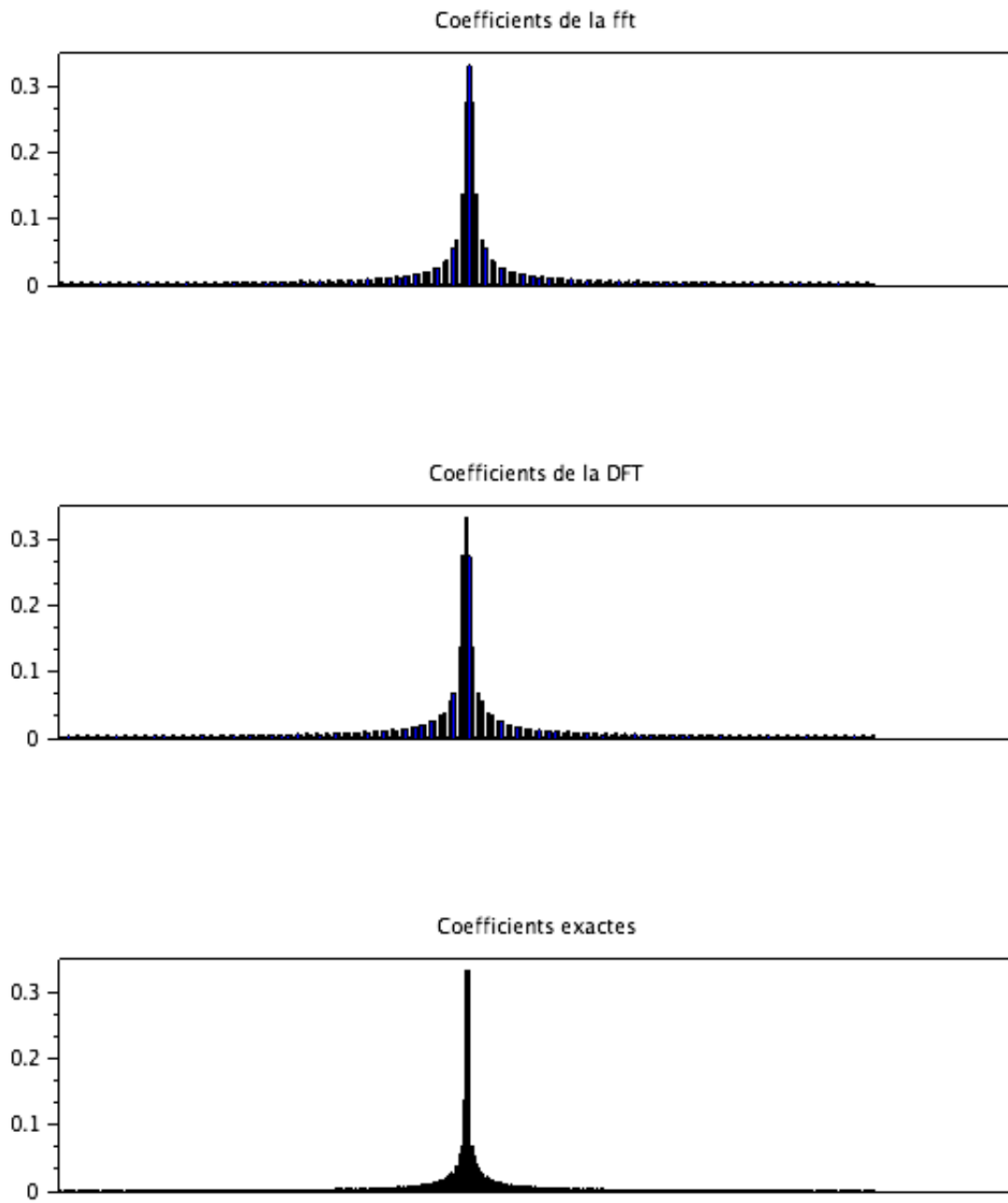


FIGURE 5.1 – *Coefficients via DFT, FFT et Exactes pour $N = 256$*

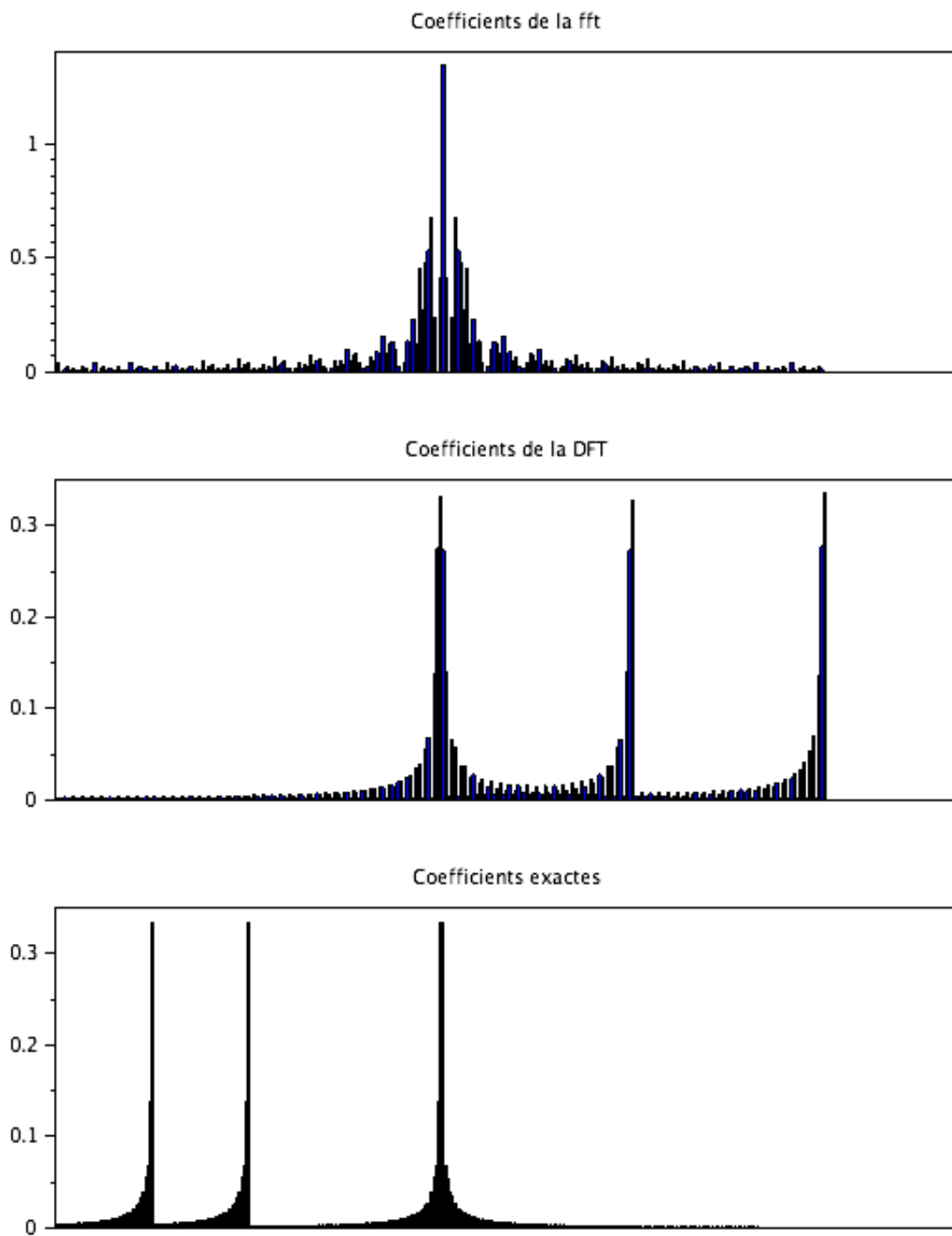


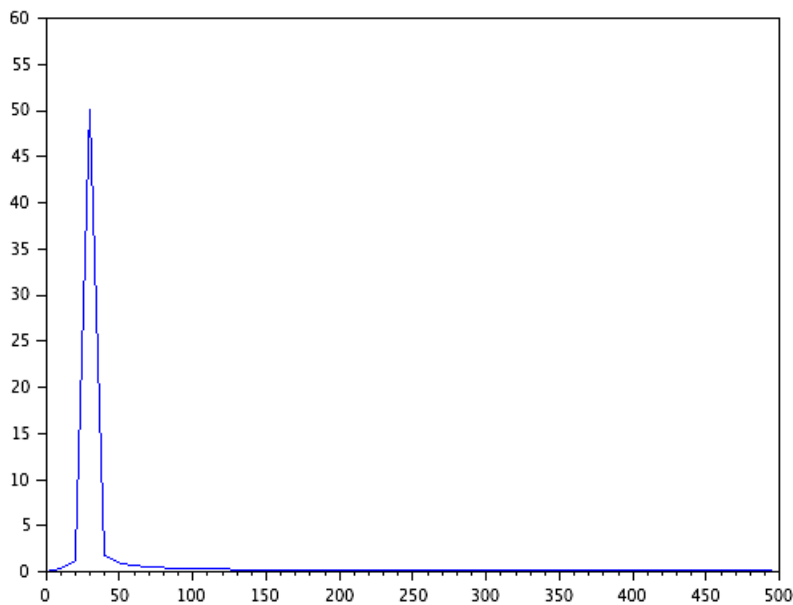
FIGURE 5.2 – *Coefficients via DFT, FFT et Exactes pour $N = 64$*

5.2 Résolution fréquentielle de signaux

Nous étudions un signal s reçu sur un intervalle de temps $[0; 0, 1]$ et échantillonné de manière discrète avec une fréquence $F_e = 1000Hz$. Nous avons alors un pas d'échantillonnage $T_e = \frac{1}{F_e}$ et un nombre d'échantillons $N = F_e T_e$. On commence avec un signal $s(t) = \sin(2\pi 30t)$.

```
clf
2 sample_rate=1000;
t = 0:1/sample_rate:0.1;
//Nombre d'échantillons
N=size(t, '*');
s=sin(2*pi*30*t);
7 y=fft(s);
//y est symétrique, on prend fnc N/2 points
//Fréquences associé
f=sample_rate*(0:(N/2))/N;
n=size(f, '*')
12 clf()
plot(f, abs(y(1:n)))
```

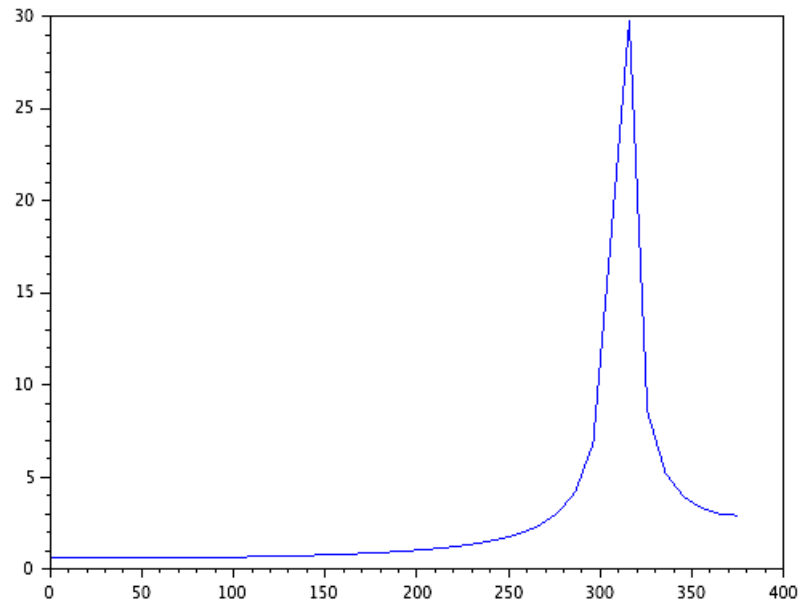
On commence par l'échantillonner en temps. Puis on trace sa FFT. Ce qui donne :



Signal échantillonné à 750Hz

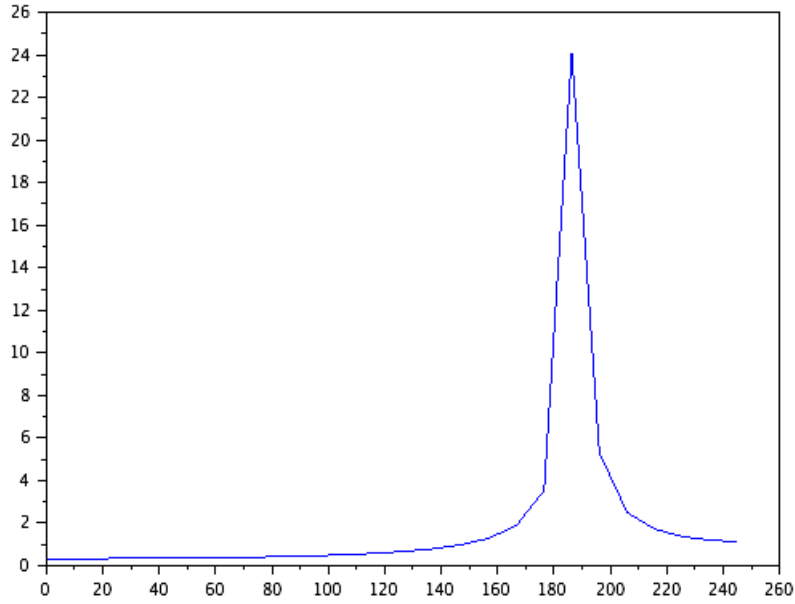
On obtient le pic attendu de fréquence $30Hz$. Si l'on modifie 30 par $k, k \in \mathbb{N}$ on retrouvera un pic de fréquence k . Maintenant, nous allons faire varier la fréquence d'échantillonnage. Pour

cela, on s'intéresse à une fonction $f(t) = \sin(2\pi 312t)$. On démarre l'échantillonnage avec une fréquence $F_e = 750Hz$. On obtient la figure suivante :



Signal échantillonné à 750Hz

Comme prévu. On obtient un pic à 312Hz. De plus, pour des fréquences de 650Hz et plus, on obtient les mêmes figures. Cependant, à partir de 600Hz et moins, le pic de fréquence de fréquence est décalé :



Signal échantillonné à 500Hz

Ici, le pic de fréquence est décalé à 190Hz alors que le signal initial possède un pic à 312Hz. Ainsi tant que la fréquence est deux fois supérieure à 312Hz (624Hz), le signal est correctement reconstitué. En revanche, dès que la fréquence est inférieure, le signal est décalé. Cela illustre le théorème de Shannon.

Théorème d'échantillonnage de Shannon : Soit f à spectre limité : $\hat{f}(\xi) = 0$ dès que $|\xi| > v_c$
Alors $\forall T_e \leq \frac{1}{2v_c}$

$$f(x) = \sum f(nT_e) \frac{\sin\left(\frac{\pi(x-nT_e)}{T_e}\right)}{\frac{\pi(x-nT_e)}{T_e}}$$

Or

$$F_e = \frac{1}{T_e} \text{ donc } \frac{1}{F_e} \leq \frac{1}{2v_c}$$

Ainsi $F_e \geq 2v_c$

La fréquence d'échantillonnage doit être supérieure à deux fois la fréquence la plus élevée d'un signal à spectre limité.

$$F_e > 2f_{max}$$

Dans le cas contraire, il y a perte d'informations et déformation du signal reconstitué.

C'est pourquoi, tant que la Fréquence d'échantillonnage F_e est supérieure à $2 * 312Hz$ (soit deux fois la fréquence), la fréquence est bien construite.

Théorème de Poisson : Soit f une fonction régulière à décroissance rapide vers l'infini, $\forall n$:

$$\|f(x)\| \leq \frac{c}{(a+x^2)^n}, \|x\| \rightarrow \infty$$

On a :

$$\sum_{n \in \mathbb{Z}} f(x+na) = \frac{1}{a} \sum_{n \in \mathbb{Z}} \widehat{f}\left(\frac{x}{a}\right) \exp(2i\pi n \frac{x}{a})$$

Démonstration du théorème de Shannon :

Pour montrer le théorème, il suffit de comparer \widehat{f} et $\widehat{\Phi}$ où

$$\Phi(x) = \sum f(nT_e) \frac{\sin\left(\frac{\pi(x-nT_e)}{T_e}\right)}{\frac{\pi(x-nT_e)}{T_e}}$$

On a

$$\begin{aligned} \widehat{\Phi}(\xi) &= \sum f(nT_e) \widehat{\frac{\sin\left(\frac{\pi(x-nT_e)}{T_e}\right)}{\frac{\pi(x-nT_e)}{T_e}}}(\xi) \\ \widehat{\Phi}(\xi) &= \sum f(nT_e) \int_{\mathbb{R}} \frac{\sin\left(\frac{\pi(x-nT_e)}{T_e}\right)}{\frac{\pi(x-nT_e)}{T_e}} e^{-2i\pi\xi n} \end{aligned}$$

On pose :

$$X = \frac{\pi}{T_e}(x - nT_e)$$

donc $x = \frac{XT_e}{\pi} + nT_e$

Donc

$$\begin{aligned} \widehat{\Phi}(X) &= \sum f(nT_e) \int_{\mathbb{R}} e^{-2i\pi\xi(nT_e + \frac{XT_e}{\pi})} \frac{\sin X}{X} dX \frac{T_e}{\pi} \\ \widehat{\Phi}(X) &= \frac{T_e}{\pi} \sum f(nT_e) e^{-2i\xi nT_e} \int_{\mathbb{R}} e^{\frac{-2i\pi XT_e}{\pi}} \frac{\sin X}{X} dX \end{aligned}$$

On pose

$$s = \frac{X}{\pi} \text{ donc } ds = \frac{dX}{\pi}$$

Ainsi

$$\widehat{\Phi}(X) = T_e \sum f(nT_e) e^{-2i\xi nT_e} \int_{\mathbb{R}} e^{-2i\pi s T_e} \frac{\sin \pi s}{\pi s} ds$$

On sait que

$$\widehat{\chi_{[-\alpha, \alpha]}}(s) = \frac{\sin \pi \alpha s}{\pi s}$$

Or

$$\widehat{\widehat{\chi_{[-\alpha, \alpha]}}}(s) = \frac{\sin(2\pi \alpha s)}{\pi s}$$

Si $\alpha = \frac{1}{2}$:

$$\widehat{\chi_{[-\frac{1}{2}, \frac{1}{2}]}}(s) = \frac{\widehat{\sin \pi \xi}}{\pi \xi}$$

Donc

$$\chi_{[-\frac{1}{2}, \frac{1}{2}]} = \frac{\widehat{\sin \pi \xi}}{\pi \xi}$$

Et

$$\begin{aligned}\widehat{\Phi}(\xi) &= T_e [\sum f(xT_e) e^{-2i\pi \xi n T_e}] \chi_{[-\frac{1}{2}, \frac{1}{2}]}(\xi T_e) \\ \widehat{\Phi}(\xi) &= T_e [\sum f(xT_e) e^{-2i\pi \xi n T_e}] \chi_{[-\frac{1}{2T_e}, \frac{1}{2T_e}]}(\xi T_e)\end{aligned}\tag{5.1}$$

En utilisant la formule de Poisson et en remplaçant f par \widehat{f} on a :

$$\begin{aligned}\sum \widehat{f}(x + na) &= \frac{1}{a} \sum \widehat{f}\left(\frac{n}{a}\right) e^{\frac{2i\pi n x}{a}} \\ \sum \widehat{f}(x + na) &= \frac{1}{a} \sum f\left(\frac{-n}{a}\right) e^{\frac{2i\pi n x}{a}} \\ \sum \widehat{f}(x + na) &= \frac{1}{a} \sum f\left(\frac{n}{a}\right) e^{\frac{-2i\pi n x}{a}}\end{aligned}$$

Si $a = \frac{1}{T_e}$ alors on a :

$$T_e \sum f(nT_e) e^{-2i\pi n x T_e} = \sum \widehat{f}\left(x + \frac{n}{T_e}\right)$$

Donc d'après (5.1) :

$$\widehat{\Phi}(\xi) = \left(\sum \widehat{f}\left(\xi + \frac{n}{T_e}\right)\right) \chi_{[-\frac{1}{2T_e}, \frac{1}{2T_e}]}(\xi T_e)$$

On a :

$$\frac{-1}{2T_e} \leq \xi \leq \frac{1}{2T_e}$$

Donc

$$\widehat{\Phi}(\xi) = \widehat{f}(\xi), \quad \forall \xi \in \mathbb{R}$$

La condition $0 \leq T_e \leq \frac{1}{2v_c}$ s'appelle condition de Shannon-Nyquist.

Démonstration du théorème de Poisson :

Soit $g(x) = \sum_{n \in \mathbb{Z}} f(x + na)$

La fonction est a -périodique, en effet

$$g(x + a) = \sum_{n \in \mathbb{Z}} f(x + a + na) = \sum_{n \in \mathbb{Z}} f(x + (1 + n)a) = \sum f(x + na) = g(x)$$

De plus, g satisfait les conditions de Dirichlet donc :

$$g(x) = S_g(x) = \sum c_k e^{\frac{2i\pi k x}{a}}$$

Où

$$c_k = \frac{1}{a} \int_0^a g(x) e^{\frac{-2i\pi kx}{a}} dx$$

On injecte l'expression de $g(x) = \sum_{n \in \mathbb{Z}} f(x + na)$ dans l'expression de c_k pour avoir :

$$\begin{aligned} c_k &= \frac{1}{a} \int_0^a \left(\sum_{n \in \mathbb{Z}} f(x + na) \right) e^{\frac{-2i\pi kx}{a}} dx \\ &= \frac{1}{a} \sum_{n \in \mathbb{Z}} \int_0^a f(x + na) e^{\frac{-2i\pi kx}{a}} dx \end{aligned}$$

On pose $X = x + na$, $dX = dx$ et on a les bornes $\begin{cases} x = a & X = (1+n)a \\ x = 0 & X = na \end{cases}$

$$\begin{aligned} c_k &= \frac{1}{a} \sum_{n \in \mathbb{Z}} \int_{na}^{(n+1)a} f(X) e^{\frac{-2i\pi k(X-na)}{a}} dX \\ c_k &= \frac{1}{a} \sum_{n \in \mathbb{Z}} \int_{na}^{(n+1)a} f(X) e^{\frac{-2i\pi kX}{a}} dX \end{aligned}$$

On sait que f est intégrable

$$\sum \int_{na}^{(n+1)a} f(x) = \int_{\mathbb{R}} h(x) \text{ d'après Chasles car } \mathbb{R} = \bigcup_{n \in \mathbb{Z}} (na, (n+1)a)$$

Donc

$$c_k = \frac{1}{a} \int_{\mathbb{R}} f(x) e^{\frac{-2i\pi kx}{a}} dx = \frac{1}{a} \hat{f}\left(\frac{k}{a}\right)$$

Ainsi

$$g(x) = S_g(x) = \sum c_k e^{\frac{2i\pi kx}{a}} = \sum_k \frac{1}{a} \hat{f}\left(\frac{k}{a}\right) e^{\frac{2i\pi kx}{a}}$$

On a bien

$$\sum_{n \in \mathbb{Z}} f(x + na) = \sum_k \frac{1}{a} \hat{f}\left(\frac{k}{a}\right) e^{\frac{2i\pi kx}{a}}$$

Chapitre 6

DFT et convolution

La convolution est une méthode pour combiner deux signaux afin d'en produire un troisième. La convolution permet de calculer la sortie d'un système étant donnés l'entrée et la réponse impulsionnelle. Un signal discret peut être représenté comme une somme d'impulsions. On peut donc ainsi donner la définition de la convolution discrètes : Soit l'opération de convolution discrète de $f(t)$ par $h(t)$ par le filtre à réponse impulsionnelle $h(t)$ on a :

$$y(t) = \sum_{k=-\infty}^{\infty} f(x)h(t-x)$$

qu'on peut écrire sous la forme suivante grâce à la commutativité :

$$y(t) = \sum_{k=-\infty}^{\infty} f(t-x)h(x)$$

Definition 2. Le produit de convolution $x * y$ de deux suites $x = (x_n)$ et $y = (y_n)$ périodiques de période N est la suite N -périodique $x * y$ définie par

$$(x * y)(n) = \sum_{k=0}^{N-1} x(n-k)y(k) \quad n = 0, 1, \dots, N-1$$

Elle est appelée convolution circulaire.

Nous nous intéressons à présent à la relation entre DFT et convolution. Nous pouvons montrer que :

$$DFT(x * y) = N.DFT(x).DFT(y)$$

Démonstration de la relation propre à la convolution discrète $DFT(fg) = N.DFT(f).DFT(g)$
En effet :

Soit h la convolution circulaire de x et y :

$$h(n) = \sum_{k=0}^{N-1} x(n-k)y(k) \quad n = 0, 1, \dots, N-1$$

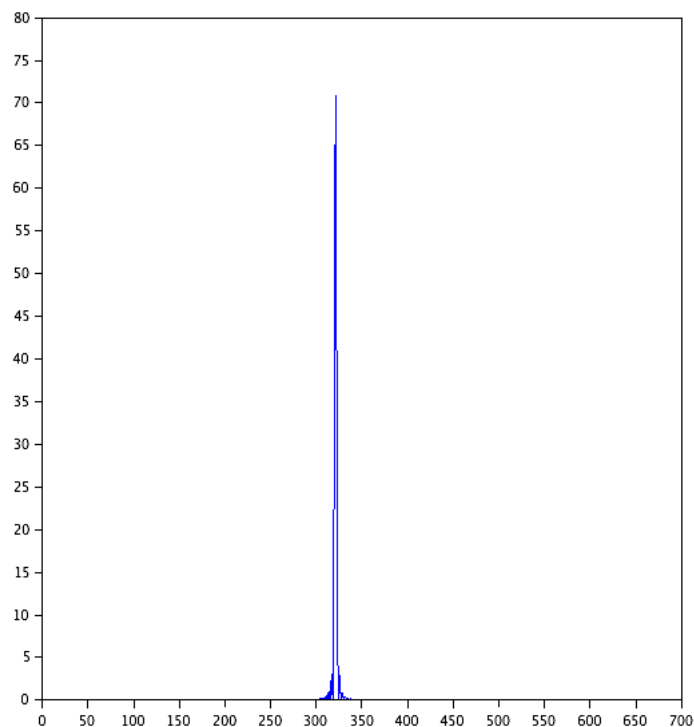
D'où,

$$\begin{aligned} DFT(h)(t) &= \frac{1}{N} \sum_{n=0}^{N-1} \omega^{-tn} h(n) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \omega^{-tn} \left(\sum_{k=0}^{N-1} x(n-k)y(k) \right) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} \omega^{-t(n-k)} x(n-k) \omega^{-tk} y(k) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \omega^{-tk} y(k) \left(\sum_{k=0}^{N-1} \omega^{-t(n-k)} x(n-k) \right) \end{aligned}$$

On pose $m = n - k$:

$$\begin{aligned} DFT(h)(t) &= \frac{1}{N} \sum_{n=0}^{N-1} \omega^{-tk} y(k) \left(\sum_{m=-k}^{N-1-k} \omega^{-tm} x(m) \right) \\ &= N \left(\frac{1}{N} \sum_{n=0}^{N-1} \omega^{-tk} y(k) \right) \left(\frac{1}{N} \sum_{m=0}^{N-1} \omega^{-tm} x(m) \right) \\ &= N.DFT(x).DFT(y) \end{aligned}$$

Le résultat de la convolution de $f * f$ est le suivant :



*fft de $f * f$*

On va illustrer une propriété fondamentale de la convolution :

$$\widehat{f * h} = \widehat{f} \cdot \widehat{h}$$

Et d'après la propriété de la transformée de fourier on peut obtenir :

$$\widehat{\widehat{f * g}} = (f * g)(x) = \widehat{\widehat{f} \cdot \widehat{h}}$$

Si $f * g$ est paire alors $(f * g)(x) = (f * g)(-x)$ donc :

$$(f * g)(x) = \widehat{\widehat{f} \cdot \widehat{h}}$$

Pour illustrer ce phénomène on commence par modéliser la convolution d'une fonction test. Puis on modélise la transformée de fourier des transformée de fourier grâce au programme suivant :

```

clc
2 clf

//définition de la fonction test
function y=f(x)
    if abs(x)<=1/6 then
7         y=1;
        elseif abs(x)>1/6 & abs(x)<=1/2 then
            y=0;
        end
    endfunction
12

//Paramètres
N=640; //Pas de l'échantillonnage
a=-1/2 //Intervalle d'étude
b=1/2
17 //Discrétisation de l'intervalle
for p=1:N
    x(p)=a+(p-1)*((b-a)/N)
    y(p)=f(x(p))
end
22

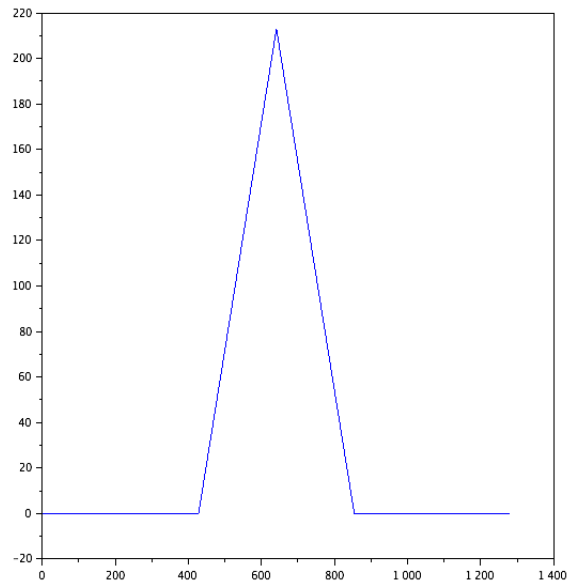
//calcul de la convolution par
//méthode convol
convolution = convol(y,y)
plot(convolution)
27

//Calcul de la convolution par la méthode
//des transformées
F=fft(y)
E=fft(y)
32

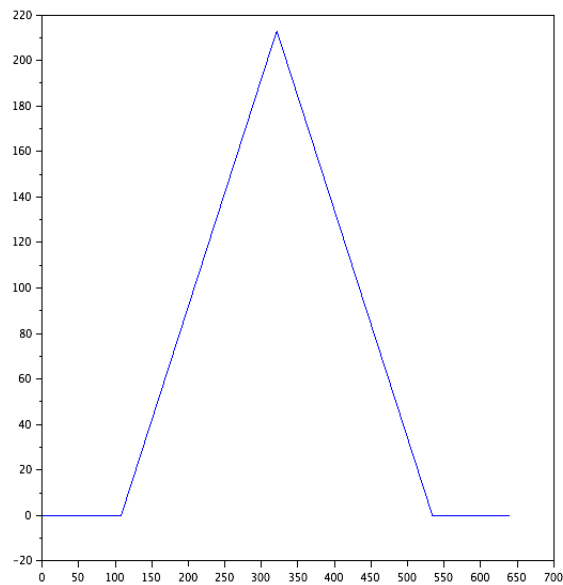
G=E.*F
//Facteur 1/N permet de normer
//les valeurs de la fft
37 K=(G)/N
K=fft(K)

plot(fftshift(K))

```



Méthode de la convolution



Méthode des transformées de fourier

Les résultats obtenus sont identiques, ce qui justifie bien la propriété énoncée. Cependant ce résultat n'est valable que pour un N grand, soit ici $N=640$. En effet lorsque N est trop petit, le pas d'échantillonnage est trop faible pour avoir une bonne approximation de la fft et donc de la convolution.

Table des figures

4.1	<i>Représentation pour $N=4$</i>	14
4.2	<i>Représentation pour $N=8$</i>	14
4.3	<i>Représentation pour $N=32$</i>	15
4.4	<i>Représentation pour $N=256$</i>	16
4.5	<i>FFT pour $N=256$</i>	17
4.6	<i>iFFT pour $N=256$</i>	17
5.1	<i>Coefficients via DFT, FFT et Exactes pour $N = 256$</i>	20
5.2	<i>Coefficients via DFT, FFT et Exactes pour $N = 64$</i>	21