

CHAPITRE 4

TRANSFORMEE DE FOURIER DISCRETE

John Tukey, (1915-2000), statisticien américain. Après une licence en chimie et deux en mathématiques à Brown et Princeton University, il devint professeur de statistiques à Princeton et continua en parallèle une carrière de chercheur aux Bell Labs. Il travailla notamment dans le domaine de l'analyse statistique robuste. Avec l'aide de James Cooley, il mit au point l'algorithme de la Transformée de Fourier Rapide (FFT). John Tukey est également l'inventeur du mot « software ».



James Cooley (1926-), mathématicien américain, pionnier dans le développement du traitement numérique du signal. Alors qu'il travaillait dans les laboratoires de recherche de IBM à New York, il co-inventa la FFT avec J. Tukey. Cet algorithme est aujourd'hui l'un des plus utilisés en sciences et en ingénierie, de la géophysique à l'astronomie en passant la biologie. Il a volontairement été rendu public par IBM dès sa découverte en 1965, sans aucune protection intellectuelle (sans brevet), afin de favoriser le développement de logiciels.



Ce troisième chapitre est consacré à l'étude du seul outil réellement utilisable en pratique pour calculer la transformée de Fourier d'un signal déterministe : la transformée de Fourier Discrète (4.1). Nous verrons que la transformée de Fourier discrète correspond à la décomposition d'un vecteur sur une base orthogonale, et qu'elle n'est donc qu'une simplification (en termes de calculs) de la transformée de Fourier à temps discret introduite au chapitre 2. Cette simplification introduit cependant quelques artefacts dont il faut être conscient, ne fût-ce que pour pouvoir les minimiser lorsque c'est possible.

Nous montrerons qu'il est possible de calculer la transformée de Fourier discrète de façon très efficace, à l'aide d'algorithmes spécialisés, dits de transformée de Fourier rapide (4.2).

La transformée de Fourier discrète doit être considérée comme un outil, d'autant plus efficace qu'on l'utilise sous sa forme rapide. En pratique, elle est essentiellement utilisée comme :

- Outil d'observation spectrale (4.3), dans le but de calculer la TTFD de signaux numériques, et même le plus souvent la transformée de Fourier de signaux analogiques sous-jacents.

- Outil de convolution linéaire rapide (4.4).

La plupart des conclusions qui seront tirées dans ce chapitre seront également applicables à l'analyse spectrale des signaux aléatoires, qui sera l'objet du chapitre suivant.

4.1 Transformée de Fourier Discrète

Si la transformée de Fourier à temps discret introduite au chapitre 2 est conceptuellement plus simple que la transformée de Fourier d'un signal analogique, il n'en reste pas moins que le calcul de la TTFD nécessite en principe une charge de calcul infinie, puisque la série qui la définit comporte un nombre infini de termes, et que l'estimation doit en être faite pour toutes les valeurs de la fréquence normalisée F entre 0 et 1.¹

C'est la raison pour laquelle la *transformée de Fourier Discrète* ou *TFD* (en anglais *Discrete Fourier Transform*, ou *DFT*) a été introduite. Son calcul est en effet limité à un nombre fini de valeurs de n et pour un nombre fini de valeurs de F .

4.1.1 Définition

Considérons une suite finie de N échantillons $\{x(n)\} = \{x(0), x(1), \dots, x(N-1)\}$. On définit sa *transformée de Fourier Discrète* comme la suite $\{X(k)\}$:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{-nk} \quad (k = 0 \dots N-1) \quad (4.1)$$

avec $W_N = e^{j\frac{2\pi}{N}}$

Il ne s'agit donc ni plus ni moins de la TTFD d'un signal numérique dont on ne considère que les N premiers échantillons, calculée pour les N fréquences normalisées $F = \{0, 1/N, 2/N, \dots, (N-1)/N\}$. A ce titre, la TFD jouit des mêmes interprétations que la TTFD : interprétation géométrique (cf. 4.1.2) et décomposition sur une base (cf. 4.1.3). Nous verrons également qu'on peut l'interpréter comme le TTFD d'un signal périodique (cf. 4.1.4).

La *transformée de Fourier Discrète inverse* est donnée par :

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{nk} \quad (n = 0 \dots N-1) \quad (4.2)$$

4.1.2 Interprétation géométrique

Comme pour la TTFD, l'évaluation de l'expression (4.1) pour k fixé correspond à une somme de vecteurs complexes W_N^{-nk} pondérés par les échantillons $x(n)$.

Exemple 4.1

La TFD de la suite de 4 échantillons $\{x(n)\} = \{x(0), x(1), x(2), x(3)\}$ correspond aux quatre valeurs complexes $\{X(k)\} = \{X(0), X(1), X(2), X(3)\}$ obtenues comme à la Fig. 4.1, où l'on

¹ Dans les exemples donnés au chapitre précédent, le premier problème n'apparaissait pas, puisque les séquences numériques étaient toujours limitées en nombre d'échantillons. Le second était « résolu » par Matlab en calculant les TTFD sur une grille de valeurs entre $F=0$ et $\frac{1}{2}$. La fonction `freqz` permet d'ailleurs de fixer le nombre de points de cette grille.

a symboliquement placé les échantillons près des vecteurs complexes auxquels ils se rapportent dans la somme (4.1).

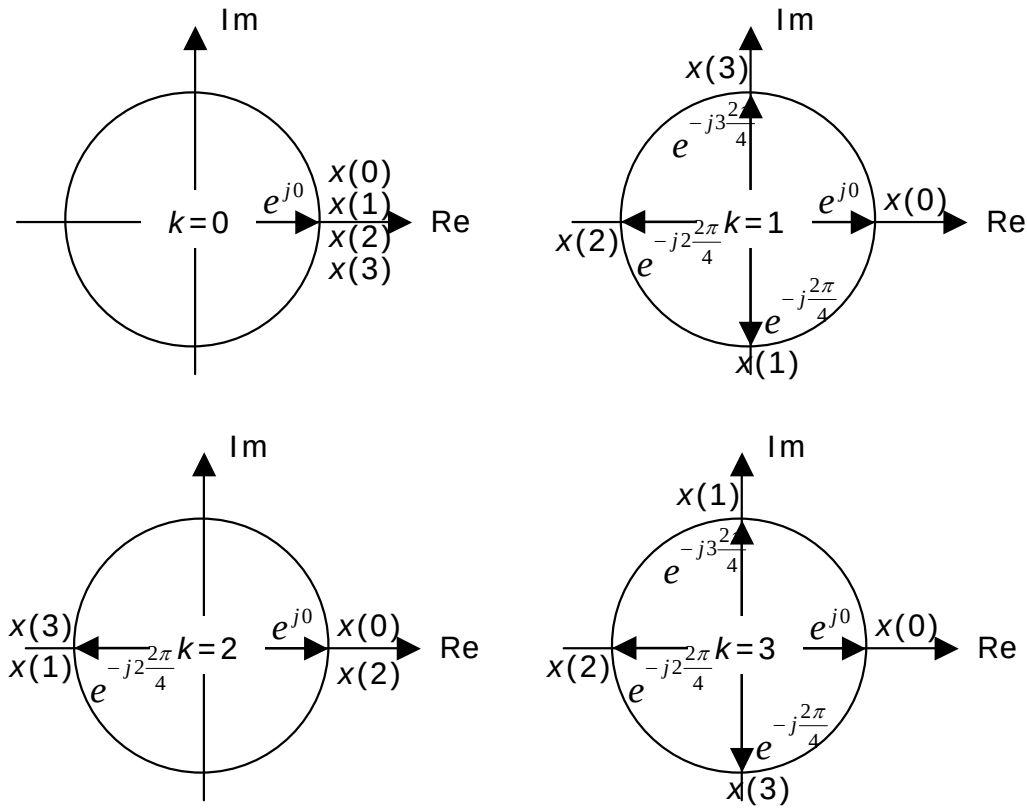


Fig. 4.1 Calcul géométrique de la TFD de $\{x(0), x(1), x(2), x(3)\}$

4.1.3 Interprétation vectorielle

Les éléments de la suite $\{x(0), x(1), \dots, x(N-1)\}$ peut être vue comme les composantes d'un vecteur $\mathbf{x}$ dans un espace à N dimensions.

L'équation (4.2) exprime alors que $\mathbf{x}$ comme une combinaison linéaire de N vecteurs de base $\mathbf{w}_k$ ($k=0, \dots, N-1$), où les composantes de chaque vecteur $\mathbf{w}_k$ sont données par la suite $\{W^0, W^k, W^{2k}, \dots, W^{(N-1)k}\}$ (Fig. 4.2).

Il est facile de montrer que ces vecteurs de base sont orthogonaux² entre eux (leurs composantes décrivent en effet des exponentielles imaginaires dont les périodes sont entières et sont entre elles dans des rapports entiers).

Exemple 4.2

Considérons une TFD sur 16 valeurs d'entrée ($N=16$). Il est facile d'afficher sous Matlab les parties réelles des composantes des cinq premiers vecteurs de base $\mathbf{w}_k$ ($k=0, \dots, 4$).

```
for k=0:4
    w=exp(j*2*pi/16*k*(0:15));
    subplot(5,1,k+1);
```

² Le produit scalaire est ici celui défini classiquement dans les espaces vectoriels.

```

    stem(real(w));
end

```

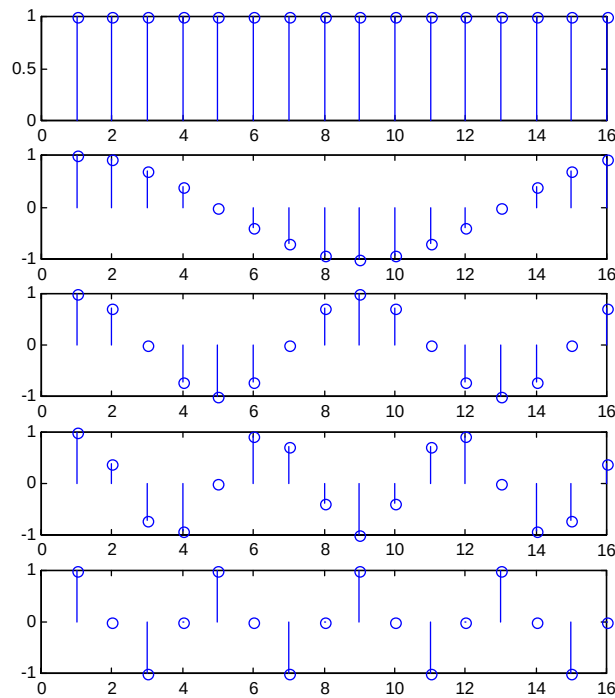


Fig. 4.2 Partie réelle des cinq premières fonctions de base pour une TFD sur 16 valeurs (N=16)

On constate que les vecteurs de base complexes sous-jacents sont bien les exponentielles imaginaires de périodes entières sur 16 échantillons (la première correspondant au cas particulier d'une période de 1 échantillon).

La décomposition peut être exprimée sous forme matricielle :

$$\mathbf{W}\mathbf{X}^T = \mathbf{x}$$

$$\text{avec } \mathbf{W} = [\mathbf{w}_0^T, \mathbf{w}_1^T, \dots, \mathbf{w}_{N-1}^T] \quad (4.3)$$

$$\mathbf{X} = [X_0, X_1, \dots, X_{N-1}]$$

$$\mathbf{w}_k = [W^0, W^k, \dots, W^{k(N-1)}]$$

La matrice $\mathbf{W}$ (dont les vecteurs-colonnes sont les vecteurs de base de la décomposition) étant orthogonale, on obtient donc facilement le vecteur $\mathbf{X}$ (vecteurs des coefficients de la décomposition) par multiplication pour la matrice adjointe :

$$(\mathbf{W}^H \mathbf{W}) \mathbf{X}^T = \mathbf{W}^H \mathbf{x}$$

avec $\mathbf{W}^H \mathbf{W} = \begin{bmatrix} \langle \mathbf{w}_0, \mathbf{w}_0 \rangle & 0 & \dots & 0 \\ 0 & \langle \mathbf{w}_1, \mathbf{w}_1 \rangle & & \\ \dots & \dots & \dots & \\ 0 & 0 & \dots & \langle \mathbf{w}_{N-1}, \mathbf{w}_{N-1} \rangle \end{bmatrix}$ (4.4)

Les éléments de la diagonale principale de cette dernière matrice sont les carrés des normes des vecteurs de base :

$$\langle \mathbf{w}_k, \mathbf{w}_k \rangle = \sum_{n=0}^{N-1} W_N^{nk} W_N^{-nk} = N \quad \text{pour tout } k$$
 (4.5)

Par conséquent :

$$\mathbf{X}^T = \frac{1}{N} \mathbf{W}^H \mathbf{x}$$
 (4.6)

Cette dernière expression n'est rien d'autre que (4.2) : les coefficients X_k de la décomposition sont donnés par les valeurs $X(k)$ de la TFD (produits scalaires de chaque $\mathbf{w}_k$ avec $\mathbf{x}$) sur N (carré de la norme des vecteurs de base) :

$$X_k = \frac{X(k)}{N} \quad (\text{pour } k = 0, 1, \dots, N-1)$$
 (4.7)

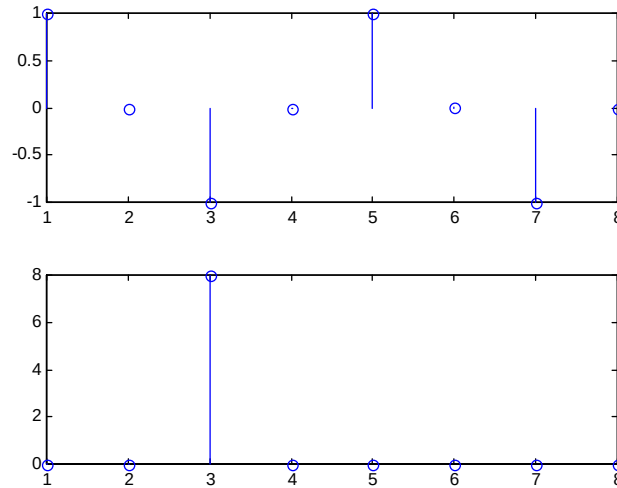
Autrement dit, les valeurs de la TFD d'une suite d'échantillons sont égales à N fois les coefficients de sa décomposition sur les vecteurs de base.

Exemple 4.3

Considérons la TFD d'un des vecteurs de base, par exemple $\mathbf{w}_2$ pour $N=8$. Il est clair en vertu du raisonnement précédent que toutes les valeurs $X(k)$ de la TFD seront nulles, sauf $X(2)$, qui doit valoir 8. On obtient effectivement ce résultat sous Matlab :

```
x=exp(j*2*pi/8*2*(0:7));
X=fft(x,8)3
subplot(2,1,1); stem(real(x)); %on n'affiche que la partie réelle de x
subplot(2,1,2); stem(abs(X)); %on n'affiche que le module de X
```

³ On notera que la fonction Matlab utilisée est `fft` (Fast Fourier Transform, cf. 4.2), et non plus `freqz` comme dans le chapitre précédent. En réalité, `freqz` utilise `fft`.

Fig. 4.3 Partie réelle de w_2 pour $N=8^4$ et module de sa TFD

4.1.4 Interprétation spectrale

Il a été vu au chapitre 2 que la TFD d'un signal périodique de période $T_0 = n_0 T_e$ est égale à $1/T_0$ fois la TFD de ses n_0 premiers échantillons (c'est-à-dire à la TFD d'une seule période du signal) :

$$F_{T_0}(F) = \sum_{k=-\infty}^{\infty} F_k \delta(\varphi - k\varphi_0) \quad \text{avec} \quad F_k = \frac{1}{T_0} \sum_{n=0}^{n_0-1} f(n) e^{-jn k \varphi_0} \quad \text{et} \quad \varphi_0 = \frac{2\pi}{n_0} \quad (4.8)$$

Par comparaison avec l'équation (4.1), on constate que, si l'on considère que $n_0 = N$:

$$X(k) = T_0 F_k \quad (4.9)$$

Autrement dit : les valeurs des coefficients $\{X_k\} = \{X(0), X(1), \dots, X(N-1)\}$ de la TFD d'une suite $\{x(n)\} = \{x(0), x(1), \dots, x(N-1)\}$ ne sont rien d'autre que les raies F_k de la TFD du signal périodique de période N ayant $\{x(n)\}$ comme période, corrigées par un facteur multiplicatif $T_0 = NT_e$ (Fig. 4.4).

⁴ Notons que la raie en 3 correspond bien à $X(2)$: Matlab commence toujours ses indices à 1.

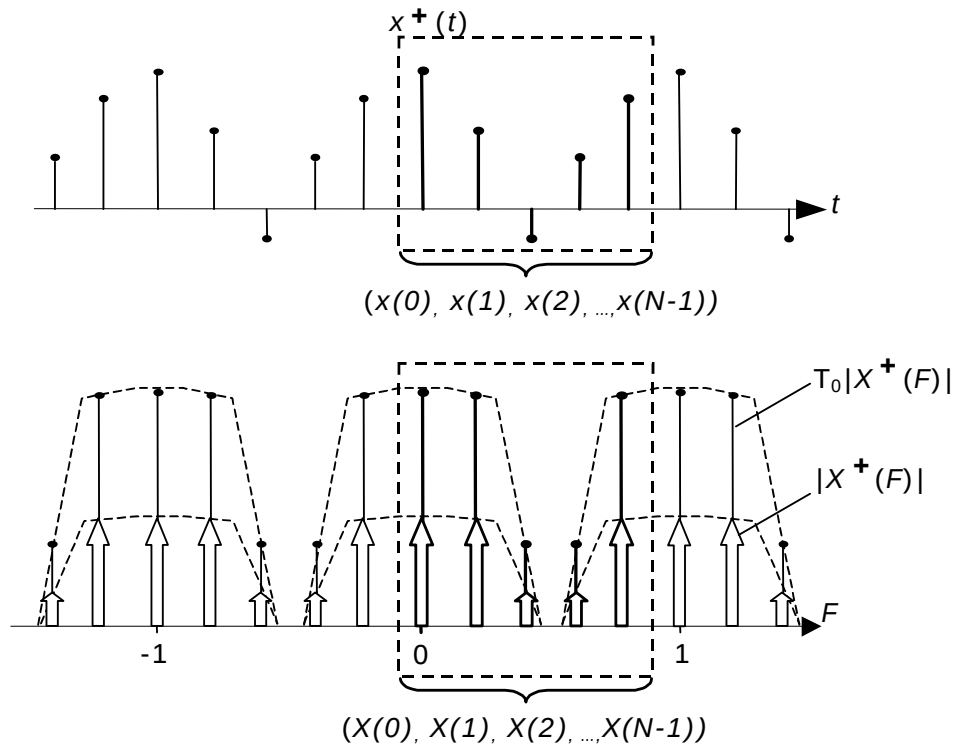


Fig. 4.4 Interprétation de la TFD d'un signal numérique comme la TFD d'un signal périodique sous-jacent

4.1.5 Propriétés

Vu l'interprétation spectrale qui vient d'être donnée, les propriétés de la TFD sont les mêmes que celles de la TFTD, à condition de supposer que le signal d'entrée est périodique de période égale à N échantillons. Comme on sait que la TFTD d'un tel signal périodique est elle-même périodique de période égale à N raies, cela revient finalement à calculer tous les indices *modulo* N . Les propriétés de la TFD sont présentées au Tableau 3.1.

(linéarité)	$\sum_i a_i f_i(n)$	$\Leftrightarrow$	$\sum_i a_i F_i(F)$
(retard)	$f((n - n_0) \bmod N)$	$\Leftrightarrow$	$F(k) e^{-j \frac{2\pi}{N} k n_0}$
(convolution circulaire)	$f(n) \otimes g(n)$	$\Leftrightarrow$	$F(k) G(k)$
(produit)	$f(n) g(n)$	$\Leftrightarrow$	$\frac{1}{N} F(k) \otimes G(k)$
(Parseval)	$\sum_{n=0}^{N-1} f(n) ^2$	$\Leftrightarrow$	$\frac{1}{N} \sum_{n=0}^{N-1} F(k) ^2$

Tableau 4.1 Propriétés principales de la transformée de Fourier à temps discret

La *convolution circulaire*, qui apparaît dans le tableau précédent, est définie par :

$$f(n) \otimes g(n) = \sum_{i=0}^{N-1} f(i)g((n-i) \bmod N) \quad (4.10)$$

Enfin, si l'on exprime $F(k)$ sous la forme :

$$F(k) = A(k)e^{j\varphi(k)} \quad (4.11)$$

on montre facilement (par exemple à partir de l'interprétation géométrique de la TFD ou tout naturellement à partir de son interprétation spectrale) que, dans le cas d'un signal réel, le module $A(k)$ est pair modulo N et la phase $\varphi(k)$ est impaire modulo N . On dit que $\{F(k)\}$ est une suite *conjuguée modulo N* :

$$F(N-k) = F(k)^* \quad (4.12)$$

La parité de $A(k)$ modulo N apparaît très bien à la Fig. 4.4.

Exemple 4.4

Considérons la TFD de la partie imaginaire (le sinus) d'un des vecteurs de base, par exemple $\mathbf{w}_2$ pour $N=8$. Puisque $\text{Im}(\mathbf{w}_2) = -j/2 (\mathbf{w}_2 - \mathbf{w}_{-2}) = -j/2 (\mathbf{w}_2 - \mathbf{w}_{N-2})$, toutes les valeurs $X(k)$ de la TFD seront nulles, sauf $X(2)$ et $X(6)$ qui vaudront $-4j$ et $+4j$, ce qui correspond bien à une suite conjuguée modulo N . On obtient effectivement ce résultat sous Matlab :

```
x=sin(2*pi/8*2*(0:7));
X=fft(x,8)
subplot(3,1,1); stem(x);
subplot(3,1,2); stem(abs(X));
subplot(3,1,3); stem(angle(X));
```

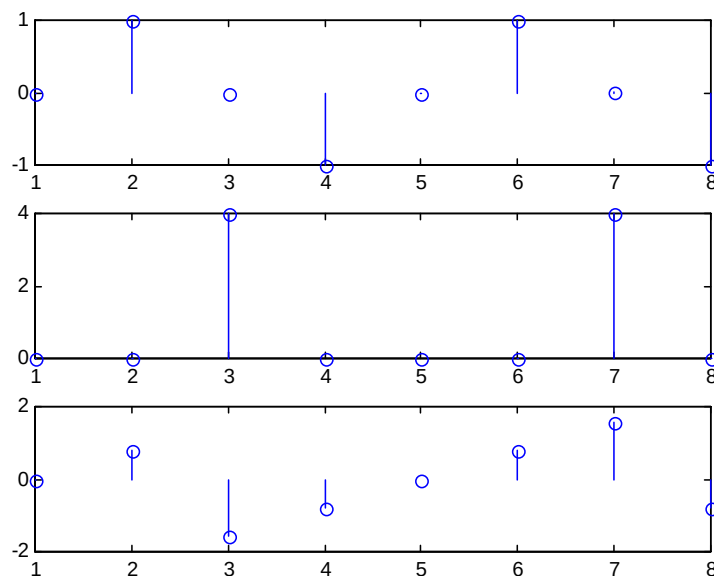


Fig. 4.5 Module et argument de $\text{Im}(\mathbf{w}_2)$ pour $N=8$: suite conjuguée modulo 8

4.2 Transformée de Fourier Rapide

En 1965, Cooley et Tukey proposèrent une méthode qui permet de réduire considérablement le temps de calcul de la TFD d'une suite dont le nombre d'échantillons N est décomposable en facteurs (typiquement, une puissance de 2). Par la suite, de nombreux algorithmes ont été publiés; ils sont connus sous le terme général de *transformation de Fourier rapide* (TFR ; ou *FFT : Fast Fourier Transform*). Tous ces algorithmes sont basés sur un même principe qui consiste à décomposer le calcul de la TFD en plusieurs TFD de longueur plus petite. La mise en œuvre de ce principe conduit à différentes méthodes dont les performances sont comparables. Nous décrirons ici plus particulièrement l'algorithme dit *radix 2 avec entrelacement dans le temps*.

4.2.1 FFT radix 2 avec entrelacement dans le temps

Cette méthode, qui exige une séquence dont la longueur est une puissance de 2 ($N = 2^M$), a rendu envisageable le calcul de TFD de plusieurs milliers de points. Le nom de *radix 2* provient du fait que l'on ramène le calcul d'une TFD de N points à un certain nombre de calculs de TFD de 2 points. L'appellation *entrelacement dans le temps* est lié à la décomposition de la suite $\{x(n)\}$ en suites plus courtes.

Soit $\{X(k)\}$ la TFD d'une suite $\{x(n)\}$ de longueur $N = 2^M$:

$$X(k) = \sum_{n=0}^{N-1} x(n).W^{-nk} \quad k = 0, 1, \dots, N-1 \quad (4.13)$$

Soient les deux suites $a(n)$ et $b(n)$ de longueur $N/2$ et leurs TFD $A(k)$ et $B(k)$:

$$\begin{aligned} a(n) &= x(2n) & a(n) &\Leftrightarrow A(k) \\ b(n) &= x(2n+1) & b(n) &\Leftrightarrow B(k) \end{aligned} \quad n, k = 0, 1, \dots, N/2-1 \quad (4.14)$$

On montre facilement que les $X(k)$ peuvent être calculés partir des $A(k)$ et $B(k)$:

$$\begin{aligned} X(k) &= \sum_{n=0}^{N/2-1} x(2n).W^{-2nk} + \sum_{n=0}^{N/2-1} x(2n+1).W^{-(2n+1)k} \\ &= A(k) + W^{-k}.B(k) \end{aligned} \quad (4.15)$$

$$\begin{aligned} X\left(\frac{N}{2} + k\right) &= \sum_{n=0}^{N/2-1} x(2n).W^{-2n(k+N/2)} + \sum_{n=0}^{N/2-1} x(2n+1).W^{-(2n+1)(k+N/2)} \\ &= A(k) - W^{-k}.B(k) \end{aligned}$$

au moyen de $N/2$ modules élémentaires appelés, connus dans la littérature anglo-saxonne sous le nom de *butterfly* et définis par les relations (4.15) dans lesquelles W^{-k} est appelé *facteur de rotation*. Ces relations sont illustrées la Fig. 4.6 dans laquelle le cercle symbolise la TFD d'une suite de longueur 2 (addition et soustraction des deux éléments).

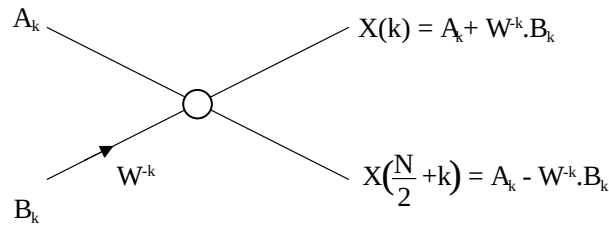
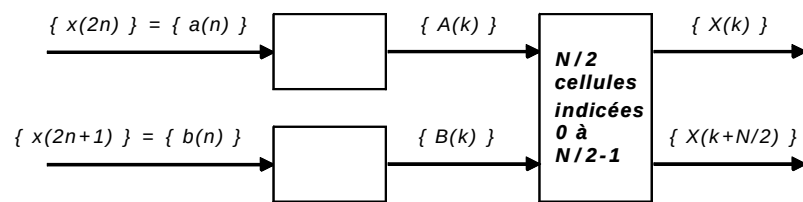
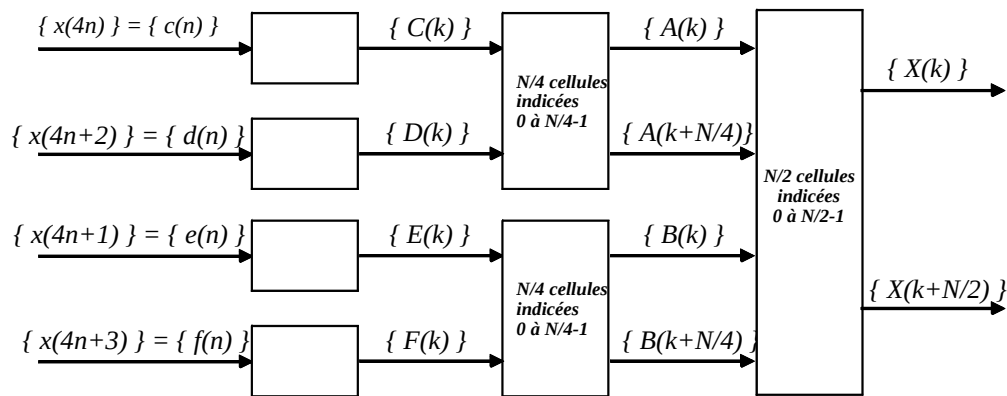


Fig. 4.6 Élément de base pour le calcul de la FFT (croisillon)

Le problème initial est donc ramené à deux sous problèmes (Fig. 4.7) : le calcul de la TFD des deux suites de $N/2$ points $\{a(n)\}$ et $\{b(n)\}$. Ces deux problèmes sont eux-mêmes divisés en problèmes élémentaires de $N/4$ points (Fig. 4.8), et ainsi de suite.

Fig. 4.7 Décomposition en deux sous problèmes de $N/2$ pointsFig. 4.8 Décomposition en quatre sous problèmes de $N/4$ points

Comme N est une puissance de 2, on peut itérer le procédé; ainsi pour $N=8$, on obtient le graphe de la Fig. 4.9. D'une manière générale, pour $N=2^M$, le calcul s'effectue en M étages et chaque étage comprend $N/2$ modules élémentaires; il faut donc au total:

$N \log_2(N)$ additions(ou soustractions) complexes

$\frac{N}{2} \log_2(N)$ multiplications complexes⁵

⁵ Certaines de ces multiplications sont mêmes triviales (celles par 1 et par j).

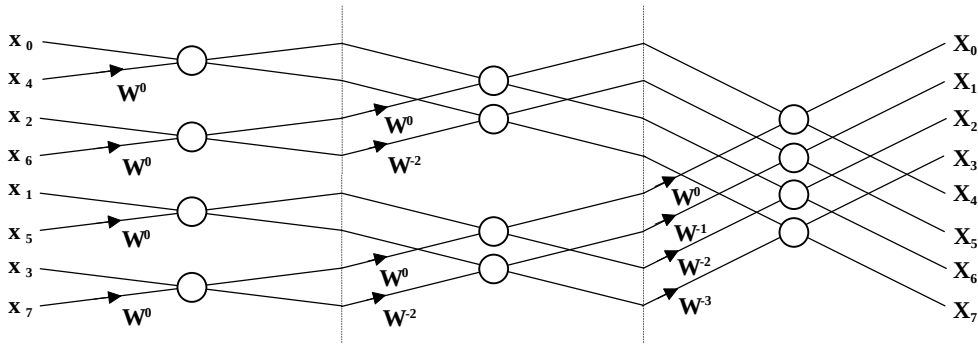


Fig. 4.9 FFT d'une suite de 8 points

On retiendra que la complexité algorithmique de la FFT est de l'ordre de $O(N \log_2(N))$, à comparer au $O(N^2)$ de la TFD. Le gain, de l'ordre de $O(N/\log_2(N))$, est énorme si N est suffisamment grand. Ainsi, pour $N=1024$, le gain en charge (et donc en temps) de calcul vaut 100.

L'algorithme est également efficient en termes de mémoire nécessaire: un seul vecteur complexe de dimension N suffit pour effectuer cette transformation. A chaque étage en effet, les résultats de chaque module élémentaire remplacent les opérandes. Cette réduction du nombre de mémoires implique, pour le graphe de la Fig. 4.9, que la suite d'entrée $x(n)$ ne soit pas classée selon l'ordre croissant des indices mais suivant un classement spécial dit d'*inversion binaire*; dans ce classement la position n_* occupée par l'échantillon d'entrée d'indice n a pour représentation binaire celle de n dans laquelle l'ordre des bits a été inversé :

$$n = (a_{m-1}, \dots, a_1, a_0)_2 \Rightarrow n_* = (a_0, a_1, \dots, a_{m-1})_2$$

Les coefficients W^{-k} indispensables à chaque étage de l'algorithme sont le plus souvent calculés préalablement et gardés en mémoire, ce qui exige une mémoire supplémentaire de $N/2$ valeurs complexes. Lorsque N est très grand, on préfère parfois calculer ces coefficients à chaque étage par récurrence (les puissances de W sont équidistantes d'étage en étage).

4.2.2 Cas particulier des suites réelles

Lorsque la suite d'entrée $\{x(n)\}$ est réelle, on sait que sa TFD est une suite conjuguée paire modulo N ($X(N-k) = X(k)^*$). Il est donc inutile de calculer les valeurs de $X(k)$ pour $k=N/2+1, \dots, N-1$: la TFD est entièrement définie par deux nombres réels ($X(0)$ et $X(N/2)$) et $(N/2)-1$ nombres complexes ($X(k)$ $k=1, \dots, N/2-1$). Nous allons voir qu'on peut effectivement éviter ce calcul inutile en préparant correctement les données en entrée.

Il est clair que si on disposait des TFD $A(k)$ et $B(k)$ des deux suites $\{a(n)\}$ et $\{b(n)\}$ de longueur $N/2$ définies à la section précédente, on pourrait calculer directement les seules valeurs de $X(k)$ qui nous intéressent, puisque :

$$\begin{aligned} X(0) &= A(0) + B(0) \\ X(N/2) &= A(0) - B(0) \\ X(k) &= A(k) + W^{-k} \cdot B(k) \quad k=1, \dots, N/2-1 \end{aligned} \tag{4.16}$$

Or on peut obtenir $A(k)$ et $B(k)$ ($k=0,1,\dots,N/2-1$) en calculant la TFD d'une suite de $N/2$ valeurs complexes $\{y(n)\}$ définie par :

$$y(n) = a(n) + jb(n) \quad k = 0,1,\dots,N/2-1 \quad (4.17)$$

En effet, comme les suites $\{a(n)\}$ et $\{b(n)\}$ sont réelles, leurs TFD sont des suite conjuguées paires modulo $N/2$:

$$\begin{aligned} A(k) &= A^*(N/2-k) \\ B(k) &= B^*(N/2-k) \end{aligned} \quad (k = 0,1,\dots,N/2-1) \quad (4.18)$$

Il vient donc :

$$\begin{aligned} Y(k) &= A(k) + jB(k) \\ Y(N/2-k) &= A^*(k) + jB^*(k) \end{aligned} \quad (k = 0,1,\dots,N/2-1) \quad (4.19)$$

D'où l'on extrait facilement :

$$\begin{aligned} A(k) &= \frac{1}{2} (Y(k) + Y^*(N/2-k)) \\ B(k) &= -\frac{j}{2} (Y(k) - Y^*(N/2-k)) \end{aligned} \quad (k = 0,1,\dots,N/2-1) \quad (4.20)$$

La passage par la TFD des $N/2$ valeurs complexes de $\{y(n)\}$ permet donc de calculer directement les $X(k)$. De cette façon, on réduit le volume des mémoires d'un facteur 2 et on diminue le temps de calcul (d'environ 40 % pour $N=1024$).

4.3 Observation spectrale

Le principal usage de la TFD est lié à la mise en évidence des caractéristiques spectrales des signaux, le plus souvent analogiques. Nous allons voir qu'il est en effet possible d'estimer la transformée de Fourier d'un signal analogique quelconque (périodique ou pas) à l'aide de la TFD.

Rappelons qu'au chapitre 3 nous avons défini la TTFD d'une suite d'échantillons résultant de l'échantillonnage d'une fonction $x(t)$ comme la transformée de Fourier de la fonction impulsionnelle $x^+(t)$ obtenue par multiplication de $x(t)$ avec un train d'impulsions de Dirac illimité dans le temps.

Nous pouvons à présent considérer la TFD d'une séquence $(x(0), x(1), \dots, x(N-1))$ comme la transformée de Fourier de la fonction impulsionnelle $x_N^+(t)$ obtenue par multiplication de $x(t)$ avec un train de N impulsions de Dirac, transformée dont on n'a gardé que les N valeurs correspondant aux fréquences $f=0, F_e/N, 2F_e/N, \dots, (N-1)F_e/N$.

Il nous reste à examiner l'effet de cette troncation dans le temps et de cet échantillonnage en fréquence.

Considérons tout d'abord le train de N impulsions de Dirac (signal fenêtre rectangulaire numérique) $w^+(t)$ valant 1 pour $t=0, T_e, \dots, (N-1)T_e$ et 0 partout ailleurs (Fig. 4.11.c). La TF $W^+(f)$ de ce signal est donnée par la TTFD de la suite d'échantillons $\{w(n)\}$ correspondants :

$$\begin{aligned}
W^+(F) &= \sum_{n=-\infty}^{\infty} w(n)e^{-jn\varphi} \\
&= 1 + e^{-j\varphi} + e^{-j2\varphi} + \dots + e^{-j(N-1)\varphi} \\
&= \frac{1 - e^{-jN\varphi}}{1 - e^{-j\varphi}} \\
&= \left(\frac{e^{-jN/2\varphi}}{e^{-j1/2\varphi}} \right) \frac{\sin(N/2\varphi)}{\sin(1/2\varphi)}
\end{aligned} \tag{4.21}$$

Cette fonction a l'allure d'une fonction **sinc périodique** ; elle vaut N en $F=0$ et s'annule en $F=1/N, 2/N, \dots, (N-1)/N$ (Fig. 4.10).

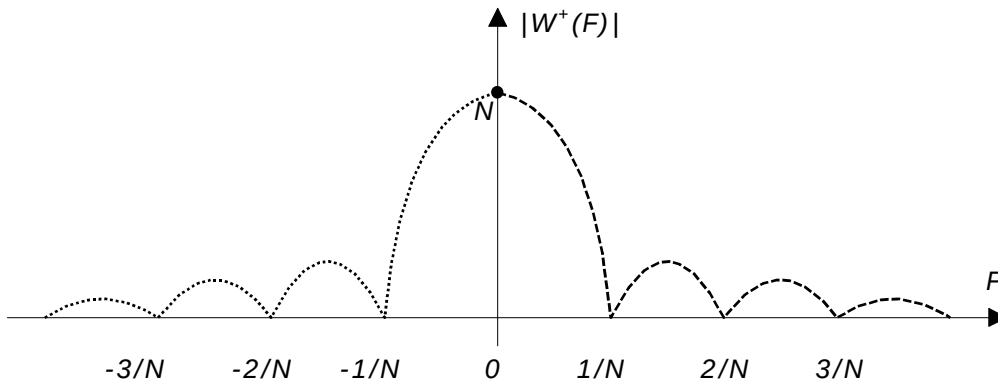


Fig. 4.10 Zoom sur la TFTD de la fenêtre rectangulaire numérique de largeur N

La TF $X_N^+(f)$ de $x_N^+(t)$ correspond donc à la convolution des TF respectives de $x(t)$ et de $w^+(t)$ (Fig. 4.11.d)

La TFD de $[x(0), x(1), \dots, x(N-1)]$ n'est alors rien d'autre que la séquence des valeurs de $X_N^+(f)$ pour $f = \{0, F_e/N, 2F_e/N, \dots, (N-1)F_e/N\}$.

4.3.1 Signal périodique

On sait que la TFTD du signal $x_{T0}(n)$ résultant de l'échantillonnage d'un signal analogique périodique $x_{T0}(t)$ avec un pas d'échantillonnage T_e est un spectre de raies périodique (Fig. 4.11.a et b).

Nous pouvons maintenant utiliser (4.21) pour établir le lien qui unit la transformée de Fourier d'un signal périodique analogique $x_{T0}(t)$ à la TFD de la suite d'échantillons correspondant $x_{T0}(n)$ ($n=0, \dots, N-1$).

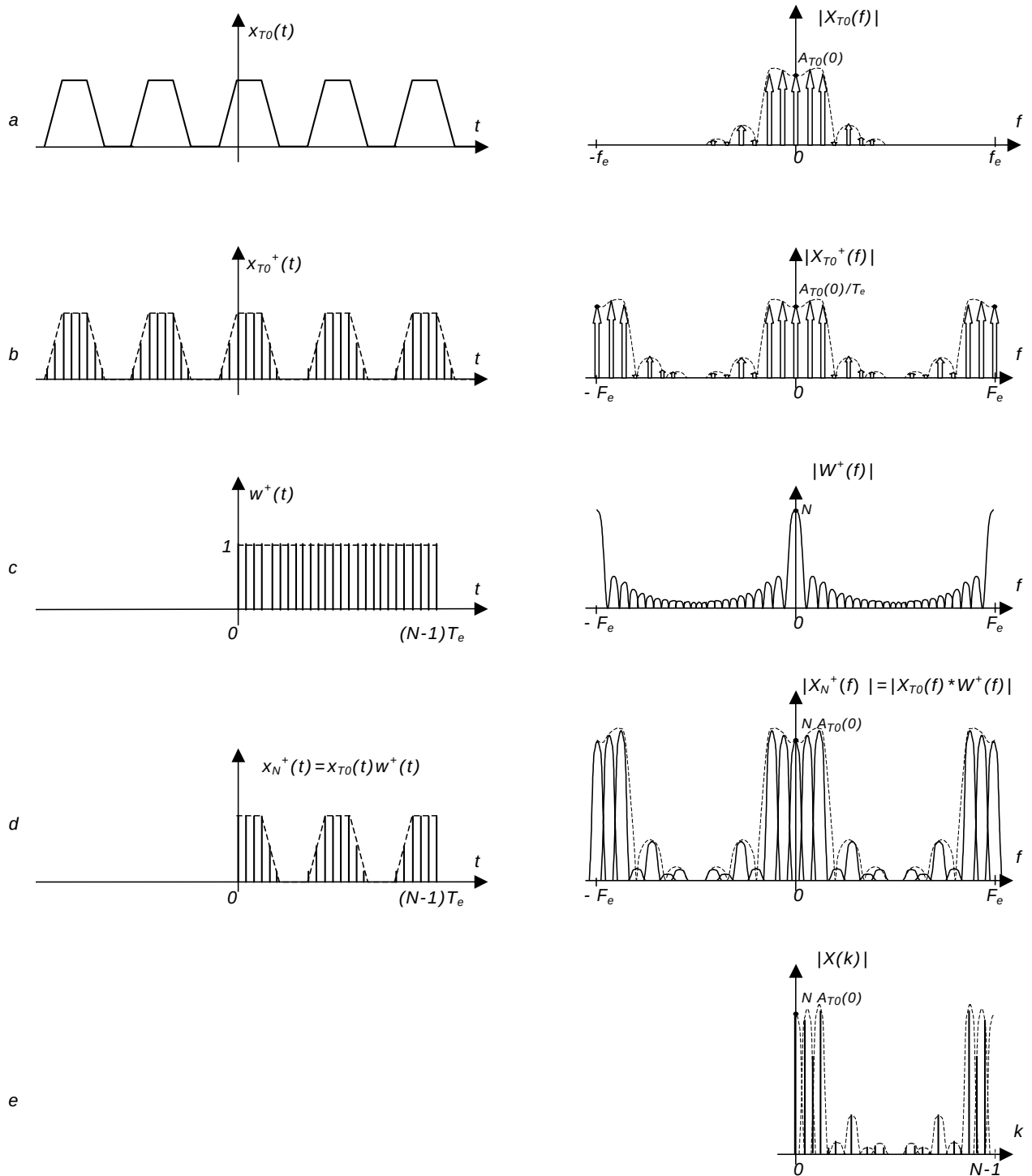


Fig. 4.11 Estimation de la transformée de Fourier d'un signal périodique par TFD : TF (a) ; TFTD (b) ; TFD (e)

En effet si, plutôt que de multiplier $x_{T0}(t)$ par un train infini d'impulsions de Dirac, on multiplie $x_{T0}(t)$ par $w^+(t)$, on constate que le lobe principal de la TF de la fenêtre $w^+(t)$, d'amplitude égale à N , vient se positionner au droit de chacune des raies du spectre de départ.

Dans le cas où $F_e / N < f_0$ (c'est-à-dire dans le cas où $N > T_0 / T_e$, ce qui revient à dire que les N échantillons retenus couvrent plusieurs périodes du signal de départ; cas de la Fig. 4.11.c), les copies du lobe principal ne se recouvrent pas trop, et leur somme laisse encore deviner les raies de départ, multipliées par N (Fig. 4.11.d). Dans le cas contraire, la superposition des lobes ne permet plus de distinguer les raies.

La TFD donne alors les valeurs de cette TFTD pour $F = \{0, 1/N, 2/N, \dots, (N-1)/N\}$ (Fig. 4.11.e).

Exemple 4.5

Affichons la TFD sur $N=16$ points d'une exponentielle imaginaire de fréquence normalisée $F_0=4/16$ et répétons le calcul pour $F_0=4.1/16$, $4.5/16$, et $4.9/16$ (Fig. 4.12). Profitons-en pour afficher la fréquence normalisée en abscisse.

```
x=exp(j*2*pi*4/16*(0:15));
X=fft(x);
subplot(4,1,1); stem((0:15)/16,abs(X)); %on n'affiche que le module de X
x=exp(j*2*pi*4.1/16*(0:15));
X=fft(x);
subplot(4,1,2); stem((0:15)/16,abs(X)); %on n'affiche que le module de X
x=exp(j*2*pi*4.5/16*(0:15));
X=fft(x);
subplot(4,1,3); stem((0:15)/16,abs(X)); %on n'affiche que le module de X
x=exp(j*2*pi*4.9/16*(0:15));
X=fft(x);
subplot(4,1,4); stem((0:15)/16,abs(X)); %on n'affiche que le module de X
xlabel('fréquence normalisée');
```

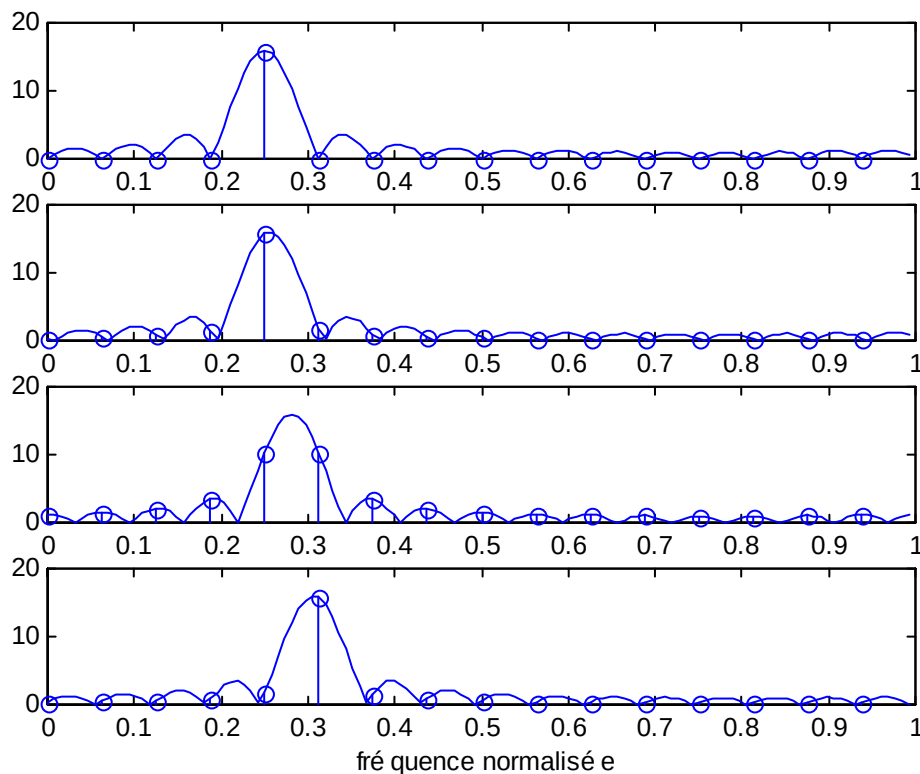


Fig. 4.12 Module de la TFD ($N=16$) d'exponentielles imaginaires de fréquences normalisées $F_0=4/16$, $4.1/16$,

4.5/16, et 4.9/16 (la fonction *sinc* périodique a été ajoutée en superposition à titre purement didactique)

On a superposé aux graphiques la TTFD de la fenêtre rectangulaire à 16 échantillons, pour mieux comprendre ce qui se passe. On constate que la raie de valeur unitaire attendue se présente sous la forme d'une raie unique de valeur 16 pour $F_0=4/16$ (multiple de $1/N$). Lorsque la fréquence normalisée du signal n'est pas un multiple de $1/N$, la TTFD de la fenêtre rectangulaire n'est plus centrée sur une des fréquences normalisées correspondant aux valeurs de la TFD, et plusieurs raies apparaissent. Le cas le pire est celui de 4.5/16, où les deux raies ont la même valeur et où les lobes secondaires de $W(F)$ sont échantillonnés par la TFD.

4.3.2 Signal non périodique

L'interprétation qui vient d'être donnée est également applicable à un signal non périodique $x(t)$, la seule différence étant que la transformée de Fourier de départ, $X(f)$, est continue (Fig. 4.13).

L'effet de la convolution de $X(f)$ avec $W^+(f)$ correspond à une périodification accompagnée d'un filtrage passe-bas de $X(f)$ (Fig. 4.13.d) : les variations brusques de $X(f)$ sont adoucies, et sa dynamique est limitée par celle de $W^+(f)$: on voit parfois apparaître sur $X_N^+(f)$ les lobes secondaires de $W^+(f)$ dans les plages de fréquences où $X(f)$ est très faible.

Le niveau d'amplitude des raies données par la TFD est lié à celui de la transformée de Fourier du signal par la relation :

$$X^+(F=0) = \sum_{n=-\infty}^{\infty} x(n) \quad X_N^+(F=0) = \sum_{n=0}^{N-1} x(n) \quad (4.22)$$

D'où l'on déduit que :

$$X_N^+(0) = kX^+(0) \quad \text{avec } k = \frac{\sum_{n=0}^{N-1} x(n)}{\sum_{n=-\infty}^{\infty} x(n)} \quad (4.23)$$

Il est clair que si la fenêtre est suffisamment longue (c'est-à-dire si N est suffisamment grand), l'effet de filtrage passe-bas sur $X(f)$ deviendra négligeable ($W^+(f)$ tendra vers une impulsion) et l'amplitude de $X_N^+(f)$ tendra vers celle de $X^+(f)$ (c'est-à-dire vers celle de la transformée de Fourier recherchée $X(f)/T_e$, si l'on suppose que le théorème de Shannon a été respecté).

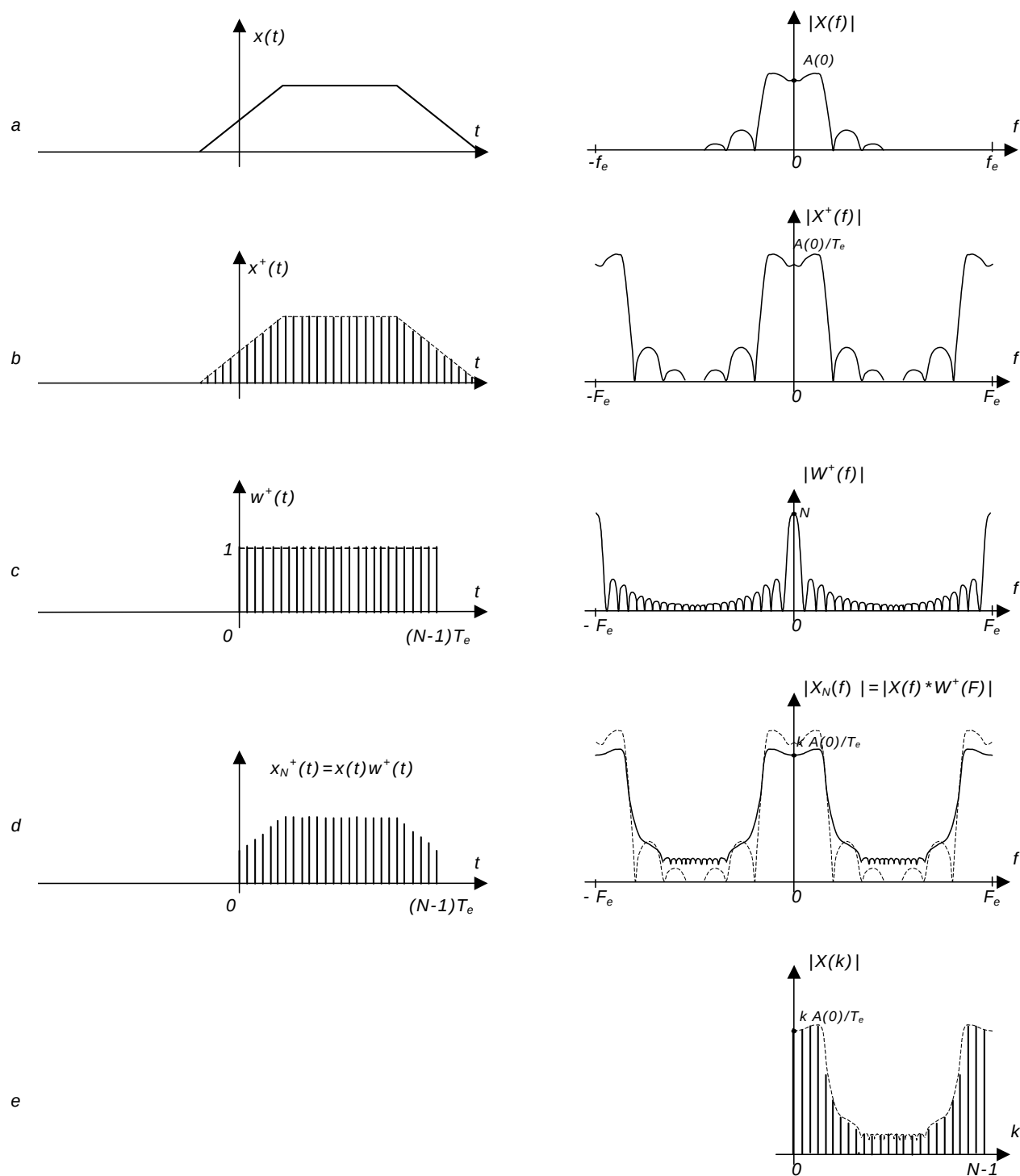


Fig. 4.13 Estimation de la transformée de Fourier d'un signal non périodique par TFD : TF (a) ; TFTD (b) ; TFD (e)

Exemple 4.6

Soit un SLI analogique du second ordre à pôles complexes conjugués, de fonction de transfert :

$$H(p) = \frac{\rho^2}{p^2 + 2\sigma p + \rho^2} \quad \text{avec } \rho = 2\pi 1000 \quad \sigma = 2\pi 5 \quad (4.24)$$

Il y correspond une réponse en amplitude :

$$\begin{aligned} A(\omega) &= 20 \log |H(j\omega)| \\ &= 20 \log \rho^2 - 20 \log \sqrt{(\rho^2 - \omega^2)^2 + 4\sigma^2 \omega^2} \end{aligned} \quad (4.25)$$

Le diagramme de Bode correspondant est donné à la Fig. 4.14, où l'on a représenté également la réponse affichée par Matlab avec :

```
rho=2*pi*1000;sigma=2*pi*5;
freqs([rho*rho],[1,2*sigma,rho*rho],logspace(3,5,200)) ;
```

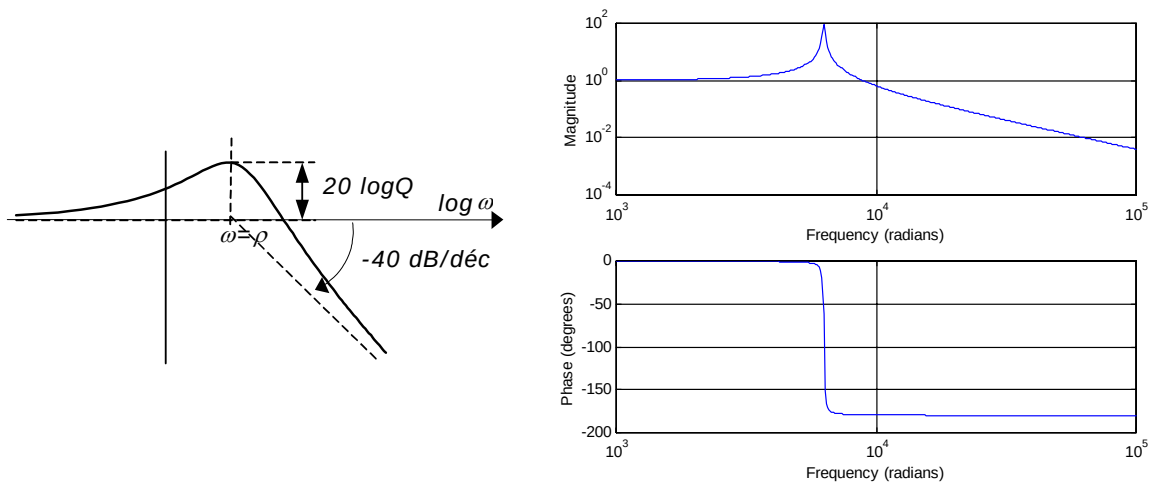


Fig. 4.14 Diagramme de Bode de $H(p) = \frac{\rho^2}{p^2 + 2\sigma p + \rho^2}$

Sa réponse impulsionnelle est donnée par :

$$h(t) = \frac{\rho^2}{\omega} \exp(-\sigma t) \sin(\omega t) \varepsilon(t) \quad \text{avec } \omega = \sqrt{\rho^2 - \sigma^2} \quad (4.26)$$

Supposons que l'on mesure cette réponse impulsionnelle sans connaître la fonction de transfert de système, et que l'on veuille afficher sa réponse en amplitude $A(f)$.

Nous devons pour ce faire fixer deux paramètres : la période d'échantillonnage T_e et le nombre N de points du signal à utiliser pour le calcul de la FFT. La Fig. 4.15 montre les résultats obtenus avec $F_e = 10000$ et différentes valeurs de N . La Fig. 4.15 montre les résultats obtenus avec différentes valeurs de F_e et des valeurs de N correspondant à l'échantillonnage d'une même portion du signal. Les commandes Matlab sont du type :

```
omega=sqrt(rho*rho-sigma*sigma)
N=1024; Te=1/10000;
n=(0:N-1);
h=rho*rho/omega*exp(-sigma*n*Te).*sin(omega*n*Te);
H=fft(h)*Te; % multiplier par Te pour tenir compte de l'échantillonnage
w=n(1:N/2+1)/N*(2*pi/Te) ; H=H(1:N/2+1); % On ne prend que H(F) pour F=0...1/2
subplot(2,1,1); plot(n*Te,h); title('h(t) pour Fe=10000 et N=256');
subplot(2,1,2); semilogx(w,20*log10(abs(H))); title('A(w) (dB)');
```

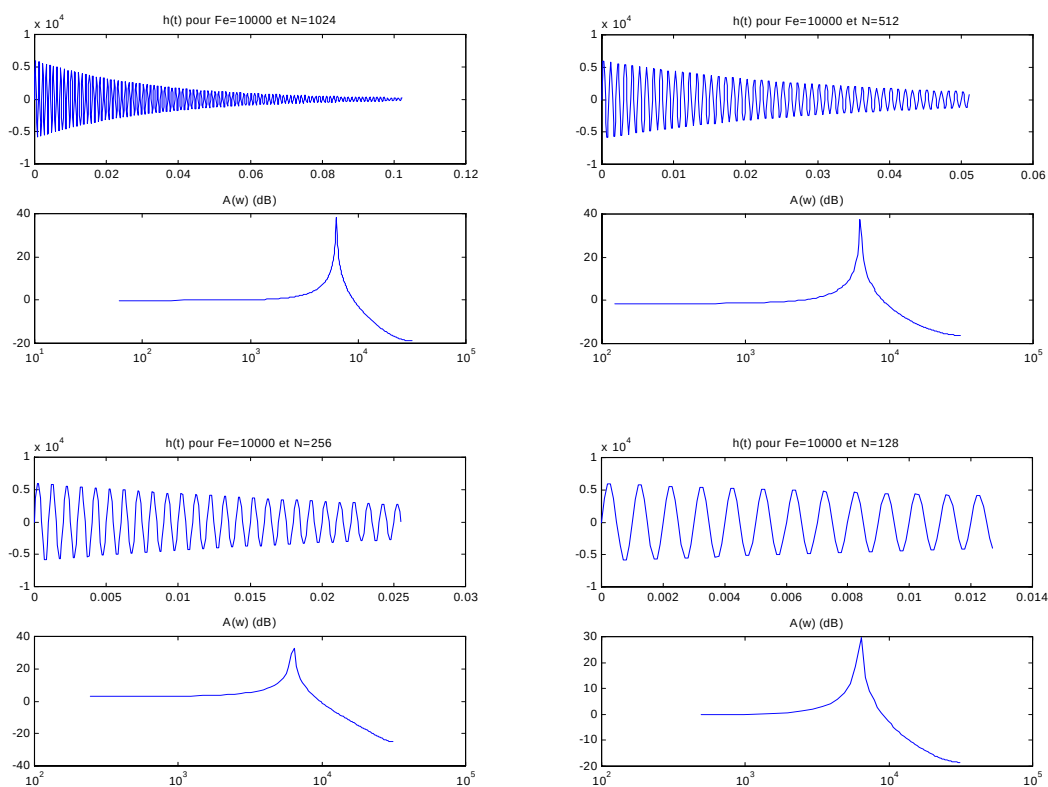


Fig. 4.15 Estimation de la réponse en fréquence en fréquence d'un SLI analogique d'ordre 2 à partir du module de la TFD de sa réponse impulsionnelle échantillonnée ($F_e=10000$; N variable)

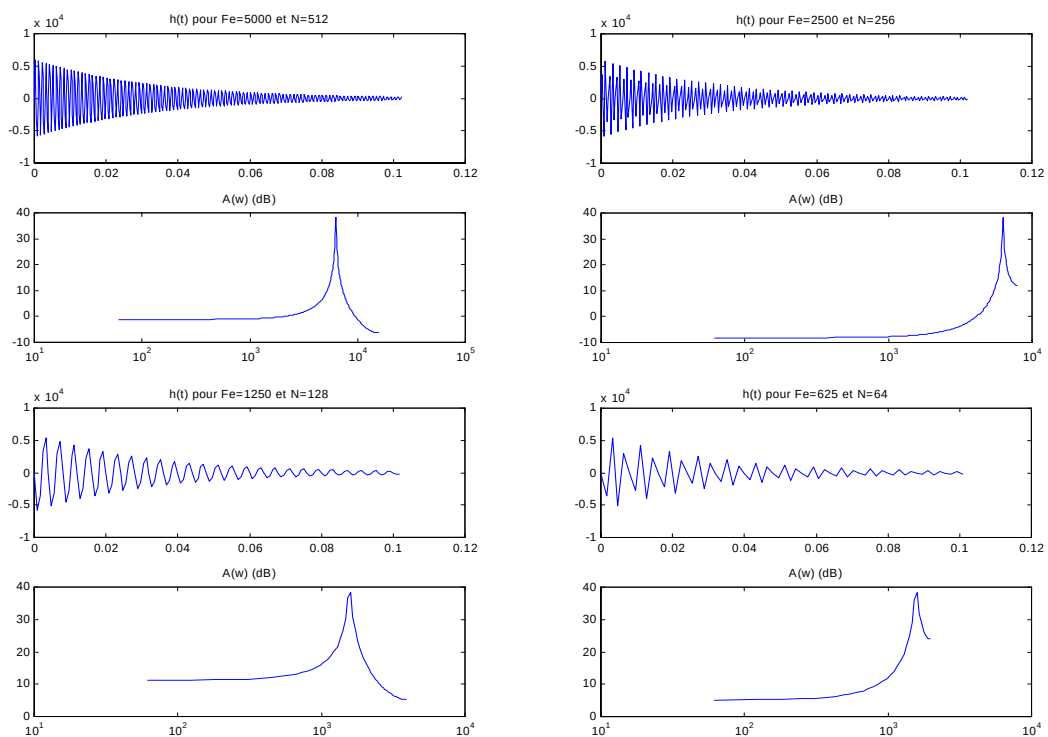


Fig. 4.16 Estimation de la réponse en fréquence en fréquence d'un SLI analogique d'ordre 2 à partir du module de la TFD de sa réponse impulsionnelle échantillonnée ($F_e=10000$; N variable)

de la TFD de sa réponse impulsionnelle échantillonnée (F_e variable; NT_e constant)

On constate à la Fig. 4.15 que, pour une fréquence d'échantillonnage suffisamment élevée, la qualité du résultat est fonction du choix de N : lorsque N diminue, l'effet de la convolution avec $W(F)$ se manifeste par un élargissement de la résonance, et une modifications des amplitudes.

Lorsqu'on diminue F_e à NT_e constant (Fig. 4.16), la plage de fréquences utile diminue, et les spectres images (qui n'apparaissent pas sur les figures) se superposent, conduisant à de fortes variations d'amplitude. Lorsque F_e diminue trop, la résonance apparaît à une de ses fréquences images, et plus à sa fréquence réelle.

Dans l'un et l'autre cas, une diminution de N implique aussi une diminution de la résolution fréquentielle : seules $N/2+1$ valeurs de la TTFD sont affichées par la TFD, entre $F=0$ et $1/2$.

On comprend, au vu de ces figures, qu'il faut être prudent sur le choix de F_e et de N lorsqu'on évalue une transformée de Fourier à l'aide de la TFD.

4.3.3 Calcul d'une TTFD sur N_{TFD} points en fréquence

On a constaté jusqu'ici que la valeur de N détermine simultanément sur la largeur du lobe principal de $W^+(f)$ (et par là sur l'importance de l'effet convolutif associé) et sur la discrétisation fréquentielle réalisée par la TFD. Ces deux effets peuvent en réalité être maîtrisés séparément, en réalisant le calcul de la TTFD du signal fenêtré $\{x_N(n)\}$ sur un nombre de points en fréquence N_{TFD} différent de N .

Complétion par des zéros

Si l'on ajoute des échantillons nuls à la suite des N valeurs ($x_N(0), x_N(1), \dots, x_N(N-1)$), de façon à obtenir une séquence de N_{TFD} valeurs, on obtient une représentation plus réaliste de $\{x_N(n)\}$. Le calcul de la TFD de cette séquence fournit donc un échantillonnage de la TTFD de $\{x_N(n)\}$, cette fois sur N_{TFD} valeurs en fréquence. La complétion d'une séquence par des zéros fournit donc une meilleure représentation de $X_N^+(f)$, sans pour autant augmenter la précision du résultat.

Exemple 4.7

Calculons la TTFD d'une fenêtre rectangulaire de 16 points, par FFT sur 16, 32, et 128 points :

```
x=ones(1,16);
subplot(6,1,1); stem(x);
subplot(6,1,2); plot((0:15)/16, abs(fft(x,16)));
x=[ones(1,16) zeros(1,16)];
subplot(6,1,3); stem(x);
subplot(6,1,4); plot((0:31)/32, abs(fft(x,32)));
x=[ones(1,16) zeros(1,112)];
subplot(6,1,5); stem(x);
subplot(6,1,6); plot((0:127)/128, abs(fft(x,128)));
```

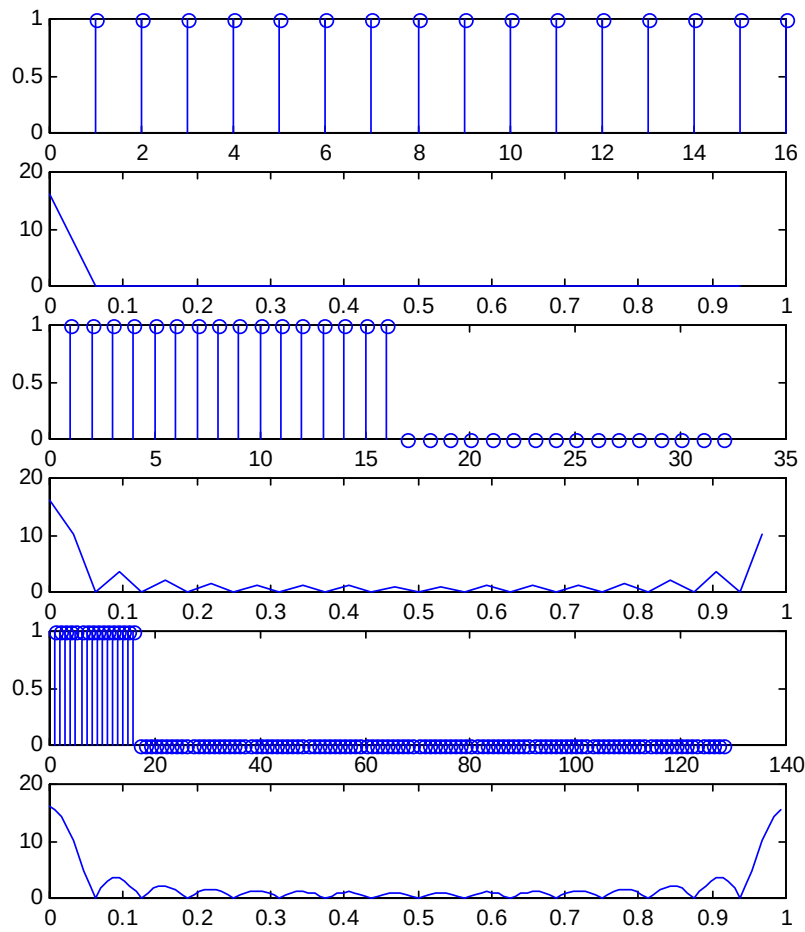


Fig. 4.17 FFT sur 16, 32, et 128 points de 16 échantillons d'une fenêtre rectangulaire.

Périodification de la séquence d'échantillons

De façon similaire, il est possible de calculer la TTFD de $\{x_N(n)\}$ sur un nombre de points N_{TFD} en fréquence *plus petit que* N . Il suffit en effet de découper la séquence $(x_N(0), x_N(1), \dots, x_N(N-1))$ en blocs de N_{TFD} valeurs, et de sommer les échantillons occupant les mêmes positions dans leurs blocs respectifs. Ceci revient à périodifier la séquence d'entrée avec une période de N_{TFD} échantillons :

$$x_{N_{TFD}}(n) = \sum_{i=-\infty}^{\infty} x_N(n - iN_{TFD}) \quad (4.27)$$

Il est facile de montrer que la TFD de la séquence $\{x_{N_{TFD}}(n)\}$ ainsi constituée fournit bien les valeurs de la TTFD de $\{x_N(n)\}$ sur N_{TFD} valeurs entre $F=0$ et 1.

Exemple 4.8

Soit une séquence de $N=8$ échantillons $(x(0), x(1), \dots, x(7))$. Constituons une séquence de longueur $N_{TFD}=4$ par (4.27) : $(x(0)+x(5), x(1)+x(6), x(2)+x(6), x(3)+x(7))$. La TFD de cette séquence est donnée géométriquement à la Fig. 4.1. Il est clair que les 4 valeurs obtenues par cette TFD sont aussi 4 des 8 valeurs que l'on aurait obtenues par TFD de $(x(0), x(1), \dots, x(7))$.

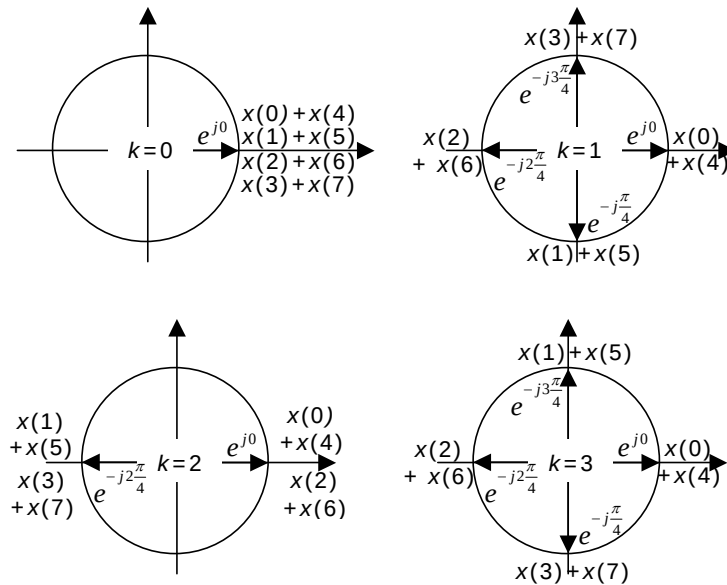


Fig. 4.18 Calcul géométrique de la TFD de $(x(0)+x(5), x(1)+x(6), x(2)+x(6), x(3)+x(7))$

L'intérêt de cette technique se fait surtout sentir lorsque le signal dont on veut calculer la TFD possède un grand nombre d'échantillons non nuls et que l'on veut éviter un effet convolutif trop important par utilisation d'une fenêtre trop courte. Le calcul sur tous les échantillons impliquerait alors une charge de calcul importante, même pour la FFT.

On retiendra que, de la même façon que la complétion par des zéros n'augmente la résolution fréquentielle sans augmenter la précision de la TFD sous-jacente, la périodification diminue la résolution fréquentielle sans affecter la précision de la TFD sous-jacente.

Exemple 4.9

Calculons la TFD d'une fenêtre rectangulaire de 16 points, par FFT sur 16, 8, et 4 points :

```
x=ones(1,16);
subplot(6,1,1); stem(x);
subplot(6,1,2); stem((0:15)/16, abs(fft(x,16)));
x=[ones(1,8)+ones(1,8)];
subplot(6,1,3); stem(x);
subplot(6,1,4); stem((0:7)/8, abs(fft(x,8)));
x=[ones(1,4)+ ones(1,4)+ ones(1,4)+ ones(1,4)];
subplot(6,1,5); stem(x);
subplot(6,1,6); stem((0:3)/4, abs(fft(x,4)));
```

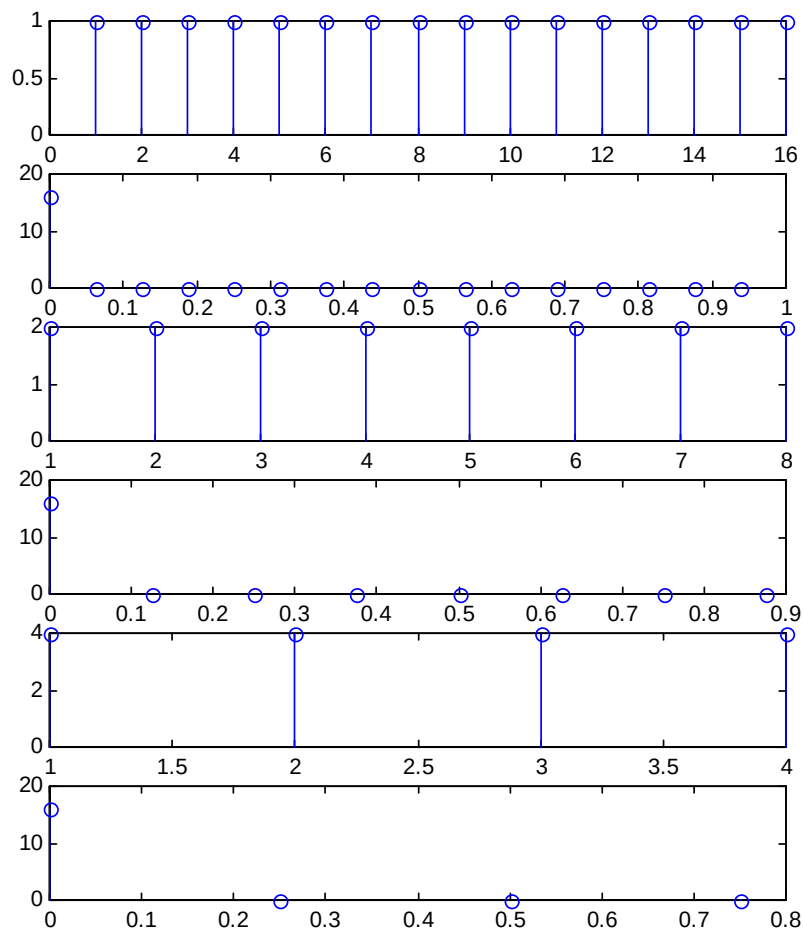


Fig. 4.19 FFT sur 16, 8, et 4 points de 16 échantillons d'une fenêtre rectangulaire.

4.3.4 Pondération par une fenêtre

Nous avons vu que le fait de ne conserver que N échantillons d'un signal $\{x(n)\}$ peut être interprété comme la multiplication de ce signal par une fenêtre rectangulaire. La TFTD de cette fenêtre possède un lobe principal de largeur $1/N$, et l'amplitude de ses lobes secondaires est à peu près 13 en dessous de celle de son lobe principal. Ceci peut être gênant lorsqu'il s'agit de mesurer l'amplitude des raies d'un signal périodique, si certaines raies sont largement à plus de 13 dB sous la raie d'amplitude maximale.

Exemple 4.10

Soit un signal constitué de deux sinusoïdes à 130 et 300 Hz, d'amplitudes 1 et 1/100, échantillonnées à 1000 Hz. On utilise une fenêtre rectangulaire de 64 points et le calcul de la FFT se fait sur 512 points en fréquence :

```
x=sin(2*pi*130*(0 :63)/1000)+ 0.01*sin(2*pi*300*(0 :63)/1000);
subplot(2,1,1); % affichage du signal
plot((0:63)/1000,x);
subplot(2,1,2); % affichage de la TFTD du signal fenêtré
plot((0:511)/512*1000,20*log10(abs(fft(x,512))));
```

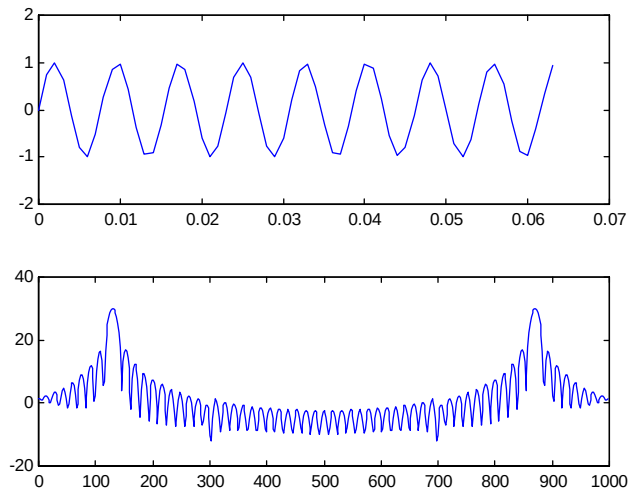


Fig. 4.20 FFT sur 512 points de 64 échantillons d'une somme de deux sinus (à 130 et 300 Hz, échantillonnées à 1000 Hz) d'amplitudes dans un rapport 100

On constate que la raie de la sinusoïde à 300 Hz est couverte par les lobes secondaires de la TFTD de la fenêtre, placée sur la raie à 130 Hz.

On peut alors utiliser sciemment une fenêtre de pondération dont les caractéristiques fréquentielles (largeur du lobe principal, position des lobes secondaires) sont différentes de celles de la fenêtre rectangulaire. Les fenêtres utilisées le plus couramment sont :

$$\text{Hanning : } w(n) = \begin{cases} 0.5 - 0.5 \cos(2\pi n / N) & \text{si } n = 0 \dots N-1 \\ 0 & \text{sinon} \end{cases} \quad (4.28)$$

$$\text{Hamming : } w(n) = \begin{cases} 0.54 - 0.46 \cos(2\pi n / N) & \text{si } n = 0 \dots N-1 \\ 0 & \text{sinon} \end{cases} \quad (4.29)$$

Blackman :

$$w(n) = \begin{cases} 0.42 - 0.5 \cos(2\pi n / N) + 0.08 \cos(4\pi n / N) & \text{si } n = 0 \dots N-1 \\ 0 & \text{sinon} \end{cases} \quad (4.30)$$

La largeur du lobe principal et le niveau des lobes secondaires pour ces différentes fenêtres est donné au Tableau 4.2. On constate que la diminution de l'amplitude des lobes secondaires se fait au prix d'une augmentation de la largeur du lobe principal.

Type	Largeur du lobe principal (en F)	Niveau des lobes secondaires
Rectangulaire	$1/N$	-13 dB
Hanning	$1.5/N$	-30dB
Hamming	$2/N$	-40 dB
Blackman	$2.75/N$	-60 dB

Tableau 4.2 Propriétés des fenêtres de pondération usuelles

Exemple 4.11

Reprenons le signal de l'exemple précédent, et pondérons le avec la fenêtre de Hamming.

```
x=sin(2*pi*130*(0 :63)/1000)+ 0.01*sin(2*pi*300*(0 :63)/1000);
x=x'.*hamming(64);
subplot(2,1,1); % affichage du signal
plot((0:63)/1000,x);
subplot(2,1,2); % affichage de la TFTD du signal fenêtré
plot((0:511)/512*1000,20*log10(abs(fft(x,512))));
```

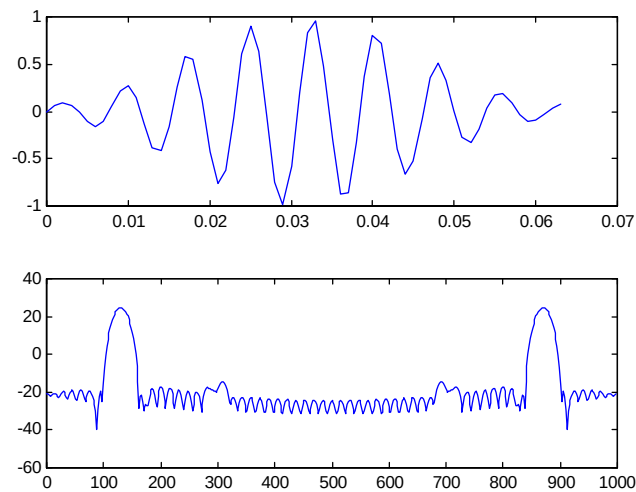


Fig. 4.21 FFT sur 512 points de 64 échantillons d'une somme de deux sinus (à 130 et 300 Hz, échantillonnées à 1000 Hz) d'amplitudes dans un rapport 100, pondérés par une fenêtre de Hamming

Même chose avec une fenêtre de Blackman :

```
x=sin(2*pi*130*(0 :63)/1000)+ 0.01*sin(2*pi*300*(0 :63)/1000);
x=x'.*blackman(64);
subplot(2,1,1); % affichage du signal
plot((0:63)/1000,x);
subplot(2,1,2); % affichage de la TFTD du signal fenêtré
plot((0:511)/512*1000,20*log10(abs(fft(x,512))));
```

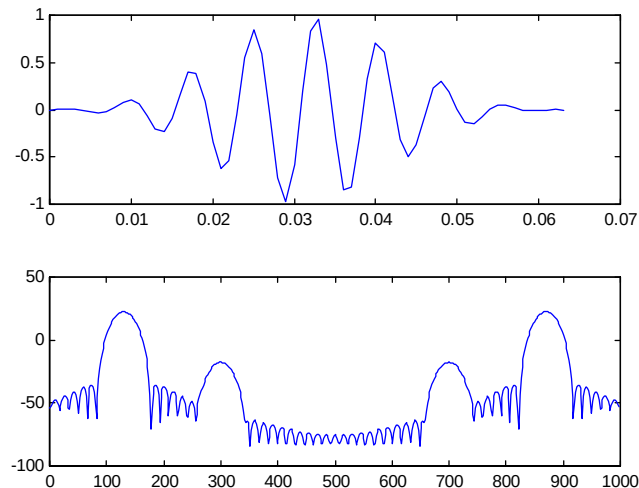


Fig. 4.22 FFT sur 512 points de 64 échantillons d'une somme de deux sinus (à 130 et 300 Hz, échantillonnées à 1000 Hz) d'amplitudes dans un rapport 100, pondérés par une fenêtre de Blackman

L'augmentation de la largeur du lobe principal peut évidemment poser problème, par exemple dans le cas de sinusoides de fréquences fort rapprochées.

Exemple 4.12

Soit un signal constitué de deux sinusoides à 130 et 150 Hz, d'amplitudes identiques, échantillonnées à 1000 Hz. On utilise une fenêtre rectangulaire de 64 points et le calcul de la FFT se fait sur 512 points en fréquence. Montrons l'influence de la fenêtre de pondération :

```
x=sin(2*pi*130*(0 :63)/1000)+ sin(2*pi*150*(0 :63)/1000);
subplot(3,1,1); plot((0:511)/512*1000,20*log10(abs(fft(x,512))));
subplot(3,1,2); plot((0:511)/512*1000,20*log10(abs(fft(x'.*hamming(64),512))));
subplot(3,1,3); plot((0:511)/512*1000,20*log10(abs(fft(x'.*blackman(64),512))));
```

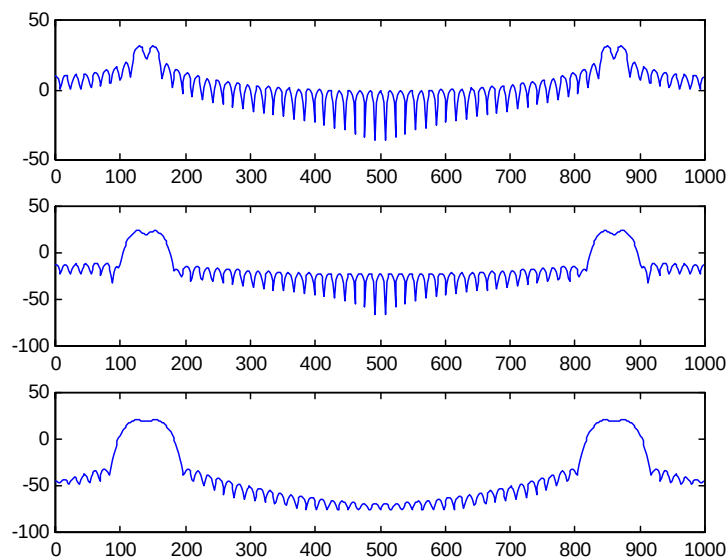


Fig. 4.23 FFT sur 512 points de 64 échantillons d'une somme de deux sinus (à 130 et 300 Hz, échantillonnées à 1000 Hz) d'amplitudes dans un rapport 100

On constate que seule la fenêtre rectangulaire permet dans ce cas de distinguer les deux raies.

4.3.5 Analyse à court-terme

Comme l'analyse en fréquence est une opération qui effectue une moyenne sur tout l'axe du temps, certains aspects temporels peuvent être complètement cachés. Il peut alors être intéressant de découper le signal en tranches, pour faire apparaître l'ordonnancement de portions du signal. On parle alors de *transformée de Fourier à court-terme* (STFT : short-term Fourier Transform).

On constate cependant toujours que le découpage en tranches plus petites augmente la résolution temporelle, mais diminue la résolution fréquentielle.

Exemple 4.13

Soit un signal constitué de deux sinusoïdes successives de fréquences normalisées égales à 0.1 et 0.2 et calculons-en une TFD sur 256 points :

```
T1=128; T2=64;
x1=sin(2*pi*0.1*(0 :T1-1));
x2=sin(2*pi*0.2*(0 :T2-1));
x=[x1 x2];
subplot(2,1,1) ; plot(x);
subplot(2,1,2) ; plot((0:255)/256,abs(fft(x,256)));6
```

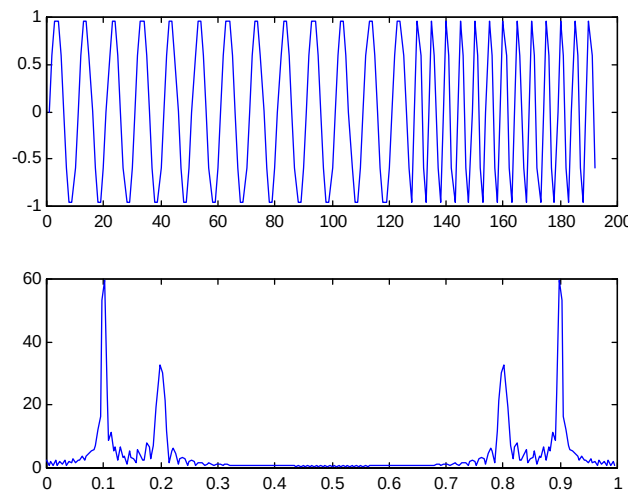


Fig. 4.24 FFT sur 256 points d'une suite de deux sinusoïdes

On constate que la FFT fait bien apparaître les raies des deux sinusoïdes, mais ne renseigne pas sur pas l'ordre dans lequel elles apparaissent. On peut découper le signal en 3 tranches

⁶ On notera au passage que la fonction `fft` effectue d'elle-même la complétion par des 0 si nécessaire.

de 64 échantillons et en afficher les TFD successives, et répéter l'opération avec 6 tranches de 32 échantillons:

```
xt=[x(1:64);x(65:128);x(129:192)];
ft=abs(fft(xt',256));
mesh(1:3,(0:255)/256,ft);
xt=[x(1:32); x(33:64);x(65:96);x(97:128);x(129:160);x(161:192)];
ft=abs(fft(xt',256));
mesh(1:6,(0:255)/256,ft);
```

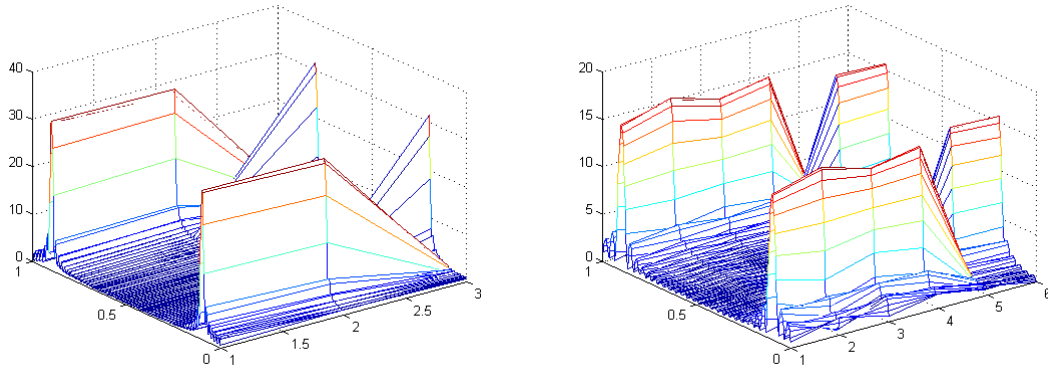


Fig. 4.25 FFT successives de tranches d'une suite de deux sinusoides (gauche : 3 tranches de 64 échantillons ; droite : 6 tranches de 32 échantillons).

4.4 Convolution linéaire

Le calcul numérique de la convolution (linéaire) est un des grands classiques du traitement du signal. Lorsqu'on veut calculer la réponse, à une entrée $x(t)$, d'un SLI analogique caractérisé par une réponse impulsionnelle $h(t)$, il est souvent plus facile d'échantillonner l'entrée et de la convoluer numériquement avec une version échantillonnée de la réponse impulsionnelle, que de calculer l'intégrale de convolution analogique. Nous verrons au chapitre 4 que la convolution numérique est également utilisée pour calculer la fonction d'autocorrélation d'un signal aléatoire.

Rappelons que le produit de convolution numérique (linéaire) est donné par :

$$y(n) = x(n) * h(n) = \sum_{i=-\infty}^{\infty} x(i)h(n-i) \quad (4.31)$$

Le nombre de valeurs non nulles retournées par la convolution de deux séquences finies de N et M valeurs est de $N+M-1$. Il est facile de montrer que le nombre d'opérations nécessaire à la convolution de ces séquences est de l'ordre de NM .

Il est possible de réduire considérablement cette charge de calcul en calculant la TFD (par FFT) de chacune des séquences à convoluer, en multipliant ces TFDs, et en prenant la TFD inverse du résultat. Cette opération nécessite cependant quelques précautions.

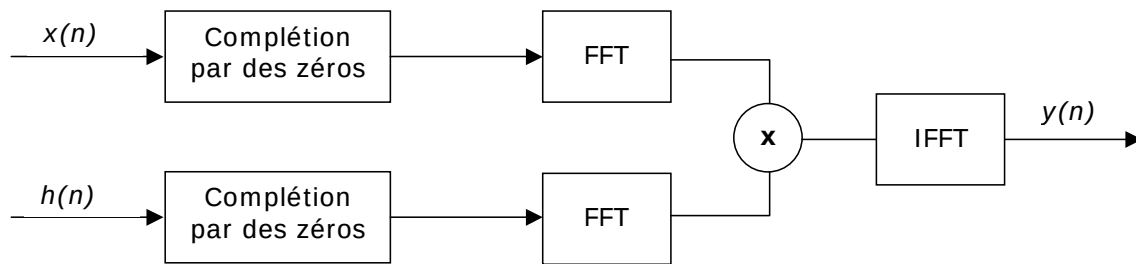


Fig. 4.26 Calcul d'un produit de convolution par FFT/IFFT

Nous avons vu en effet à la section 4.1.5 que le produit de deux TFD est équivalent à la convolution *circulaire* des séquences de départ :

$$y_c(n) = x(n) \otimes h(n) = \sum_{i=0}^{N-1} x(i)h((n-i) \bmod N) \quad (4.32)$$

La convolution circulaire implique que les deux séquences soient de même longueur (ou qu'on les complète par des zéros pour qu'elles le soient), et la séquence résultat est de même longueur que les séquences d'entrée.

On comprend aisément que le résultat du produit de convolution circulaire $y_c(n)$ est identique à celui du produit de convolution linéaire si les deux séquences sont complétées par des zéros de façon à former des séquences de $N+M-1$ valeurs.

En pratique, on ajoutera souvent plus de zéros que nécessaire (jusqu'à atteindre une puissance de 2), afin de profiter de la rapidité de la FFT.

Le gain en charge de calcul par rapport au calcul du produit de convolution par (4.31) est évidemment plus faible que celui de la FFT sur la TFD, puisqu'on utilise ici 3 FFT (2 directes et une inverse). Il reste que, pour des valeurs élevées de N ou M , le calcul de convolution est incomparablement plus efficace par FFT.

Exemple 4.14

Soit $a(n)$ et $b(n)$ constitués respectivement de 200 et 180 échantillons unitaires. On peut en calculer la convolution linéaire à partir de (4.31), puis comparer au calcul par FFT sur 256 et 512 points :

```

a=ones(1,200) ; b=ones(1,180) ;
y1= conv(a,b);
y1(600)=0; % forcer le vecteur à avoir 600 valeurs
subplot(3,1,1); plot(y1);
a(256)=0;b(256)=0; %completer par des zeros
y2=real(ifft(fft(a).*fft(b)));
y2(600)=0 ; % forcer le vecteur à avoir 600 valeurs
subplot(3,1,2); plot(y2);
a(512)=0;b(512)=0; %completer par des zeros
y3=real(ifft(fft(a).*fft(b))); 7
y3(600)=0 ; % forcer le vecteur à avoir 600 valeurs
subplot(3,1,3); plot(y3);
  
```

⁷ L'appel à la fonction `real` est indispensable : vu les erreurs de calcul, l'IFFT renvoie un résultat très légèrement complexe.

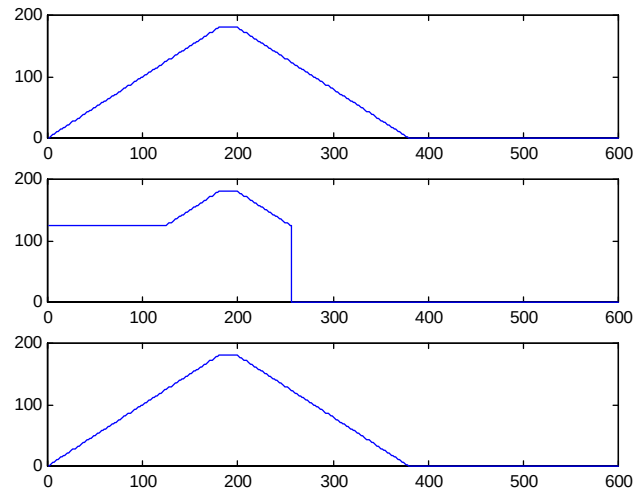


Fig. 4.27 Calcul d'une convolution linéaire (haut : force brute ; centre : par FFT sur 256 points; bas : par FFT sur 512 points)

On constate que le calcul par FFT sur 256 points ne conduit pas au bon résultat : $256 < 200 + 180 - 1$.