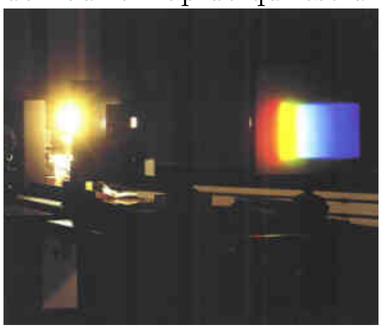
LA TRANSFORMEE DE FOURIER RAPIDE

1 INTRODUCTION

Pour étudier les phénomnes de propagation de la chaleur , Jean Baptiste Fourier invente une théorie dans laquelle une distribution donnée peut se décomposer en une somme d'un fondenmental et de ses harmoniques. Ses travaux sont publiés en 1822. L'académie des sciences lui décerne un prix spécial . La théorie de l'analyse spectrale est née, pour se répandre dans tous les domaines scientifiques : biologie, diagnostique médical, traitement du signal, astronomie,compression des images ...
Pour avoir une idée intuitive de la transformée de Fourier étudions le fonctionnement de notre cerveau et de notre oreille interne lorsque nous écoutons de la musique. Les ondes sonores viennent frapper notre tympan qui se déplace à leur rythme. La cochlée ,ou oreille interne, transforme les variations de pressions captées par le tympan en une information fréquentielle qui est interprétée par le cerveau comme une tonalité et une texture.
La transformée de Fourier est une technique mathématique qui réalise quelque chose d'analogue : décomposer une fonction du temps en un spectre de fréquences, à la manière d'un prisme qui décompose la lumière en un spectre de couleurs. La transformée inverse permer de retrouver la fonction initiale de la meme façon qu'un second prisme placé à la suite d'un premier (et inversé) permet de reformer le rayon décomposé par le prisme précédent.

Nous allons d'abord étudier la transformée de Fourier discrète qui est la transformée de Fourier appliquée à un échantillon fini , puis la transformée de Fourier rapide qui est un algorithme de calcul de cette transformée.



Le calcul de la transformée de Fourier discrète (ffd) appliquée n nombres nécessite plus de n^2 multiplications et donc devient vite impossible à effectuer dés que n est grand. L'algorithme, désormais célèbre, de la transformée de Fourier rapide permet de ramener ce nombre d'opérations à environ $n \times log_2(n)$. Il a été inventé en 1960 par Cooley et Tukey, ingénieurs dans un centre de recherche d'IBM. Ainsi cette réduction de complexité suffit à faire passer d'impossibles à facilement résolubles nombre de problèmes.
Pour illustrer l'intérét de cette transformation nous l'appliquerons à la multiplication des polynômes. Cette technique appliquée à la multiplication des grands nombres a permis d'obtenir des millions de décimales dans les calculs des constantes mathématiques comme "π" ou "e".

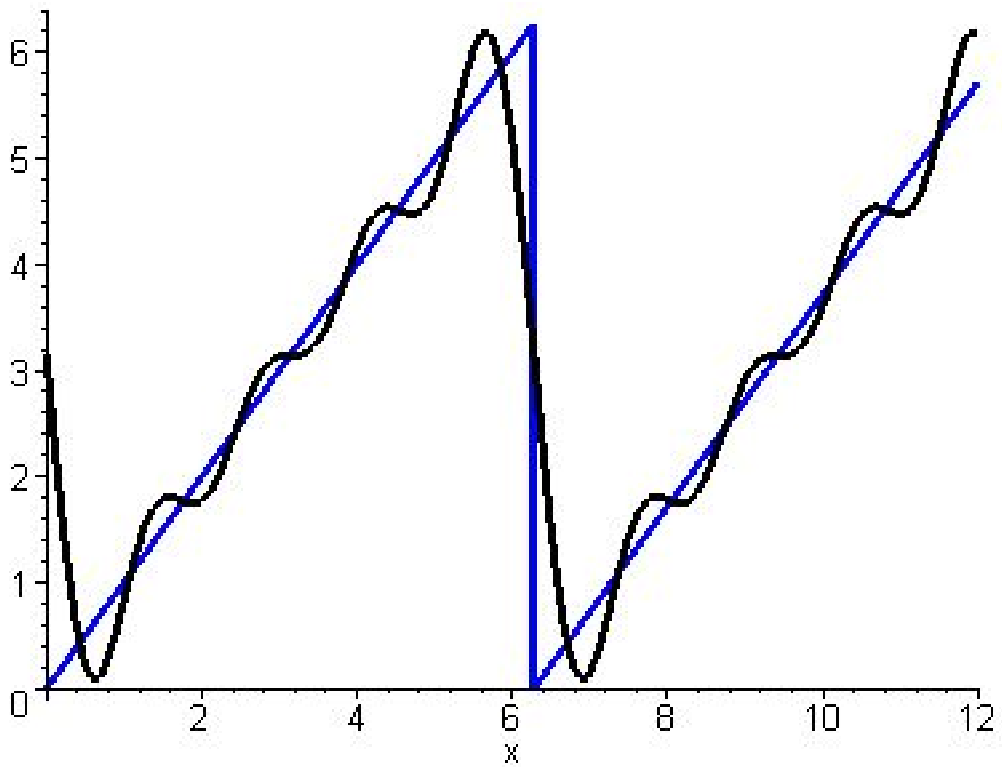
2 SERIE DE FOURIER

Sous les conditions de Dirichlet , un signal périodique de période T $x(t)$peut être décomposé en une somme infinie de sinusoides de pulsation $n\omega = \frac{n2\pi}{T}$ qui sont des multiples de la pulsation fondamentale $\omega = \frac{2\pi}{T}$. $x(t) = \sum_{n=-\infty}^{n=+\infty} C_n e^{I\frac{2\pi nt}{T}}$. Ces sinusoides ont leur amplitude et leur déphasage déterminés par la formule :

$$C_n = \frac{1}{T}\int_0^T x(t)e^{-I\frac{2\pi nt}{T}}dt$$

Ainsi par exemple, un synthétiseur de musique peut rendre le timbre des divers instruments de musique en dosant par différents harmoniques le son fondamental dont la fréquence détermine sa hauteur dans la gamme . Par

exemple le do1 a pour fréquence 65,4 hertz.
Plus physiquement, pour illustrer ce phénomène, présentons le signal périodique suivant : $x(t) = 2\pi \times frac(\frac{t}{2\pi})$ de période 2π représenté en noir sur le graphique suivant.
Alors $C_n = \frac{1}{2\pi} \int_0^{2\pi} te^{-Int}dt = \frac{1}{2\pi} \frac{2\pi}{-In} = \frac{1}{-In}$ si n n'est pas zéro sinon $C_0 = \pi$
Notre signal périodique admet alors l'écriture en série de Fourier suivante : $x(t) = C_0 + C_1 e^{It} + C_{-1} e^{-It} + ... + C_n e^{Int} + C_{-n} e^{-Int} + ... = \pi - 2(sin(t) + sin(2t)/2 + ... + sin(nt)/n + ...$
Nous voyons sur le graphique les 3 premiers termes de cette série qui approchent la fonction $x(t)$.



Pour ceux qui ne sont pas habitués aux écritures avec les nombres complexes ces formules peuvent être présentées de la manière suivante : a_0 = valeur moyenne de x sur la période T $= \frac{1}{T} \int_T f(t)dt$

$$a_n = \frac{2}{T} \int_T x(t)cos(nwt)dt$$

$$b_n = \frac{2}{T}\int_T x(t)sin(nwt)dt$$

Dans ces écritures l'intégrale est faite sur une période, c'est à dire T.* Selon le cas on choisit l'intervalle [0 ;T] ou

$$[-\frac{T}{2};\frac{T}{2}]$$

, les résultats sont identiques.

$$w = pulsation = \frac{2\pi}{T} = 2\pi f$$

Test la période f la fréquence
Sous les conditions courantes (=Dirichlet) le signal périodique x(t) est approché par les premiers termes de sa série de Fourier :

$$x(t) = a_0+a_1cos(wt)+b_1sin(wt)+a_2cos(2wt)+b_2sin(2wt)+a_3cos(3wt)+b_3sin(3wt)$$

On n'a écrit que les trois premiers termes du développement(en plus de la valeur moyenne de x(t)). Le premier terme :

$$a_1cos(wt) + b_1sin(wt)$$

s'appelle le fondamental car pour une note de musique c'est lui qui donne la hauteur de la note dans la gamme do, r ,mi...
Les deux autres termes :

$$a_2cos(2wt) + b_2sin(2wt)$$

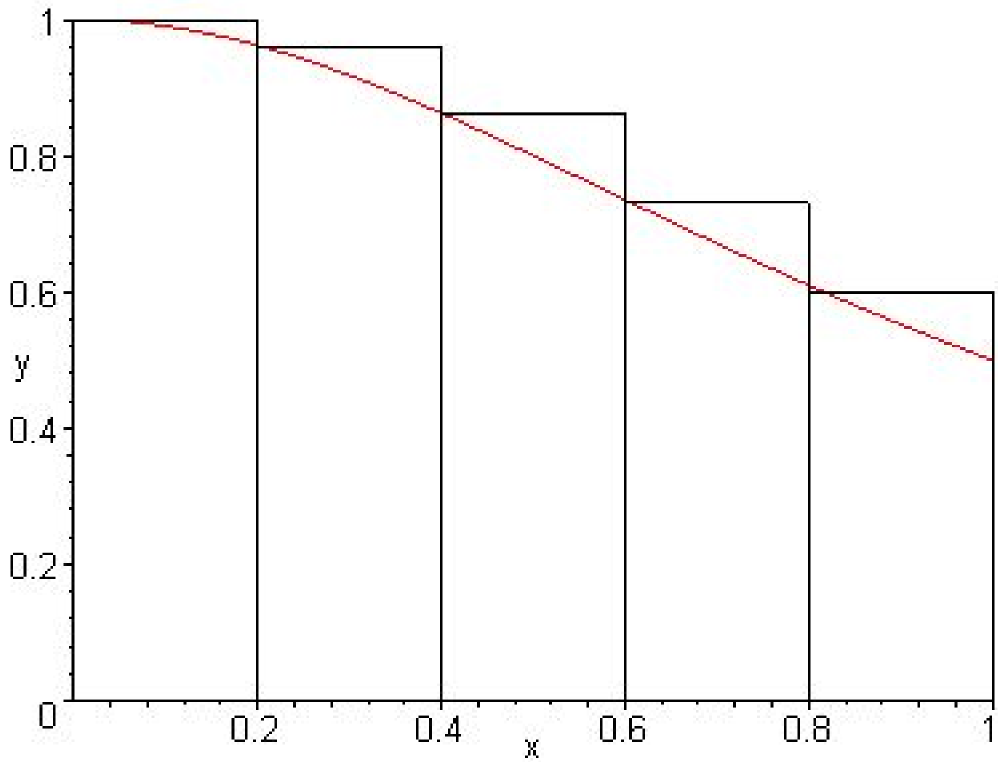
et

$$a_3cos(3wt) + b_3sin(3wt)$$

s'appellent les harmoniques de rang 2 et 3. Ce sont eux qui donnent le timbre de la note. Ils diffèrent si la note été produite par un piano , une tompette , une guitare ...
Cette série converge vers le signal x(t). Cela signifie que plus on prend de termes plus le résultat est proche de x(t).
Aux points de continuité de x on a l'galité entre x(t) et sa série de Fourier :

$$x(t) = a_0 + \sum_{n=1}^{\infty} a_kcos(nwt) + b_nsin(nwt)$$

Pour obtenir les coefficients C_n , pour une fonction x(t) périodique ,nous faisons l'intégrale : $C_n = \frac{1}{T}\int_0^T x(t)e^{-I\frac{2\pi nt}{T}}dt$.
Cette intégrale peut être calculée de manière approchée par la méthode des rectangles en divisant l'intervalle T sur lequel on la calcule en N segments de longueur T/N, leurs entrémités successives sont alors $\frac{kT}{N}$. On approxime l'intégrale en la remplaçant par la somme des rectangles de base T/N et de hauteur $x(\frac{kT}{N})$ pout k de 0 N-1.
Sur le dessin suivant on illustre la méthode des rectangles pour une fonction périodique de période 1.



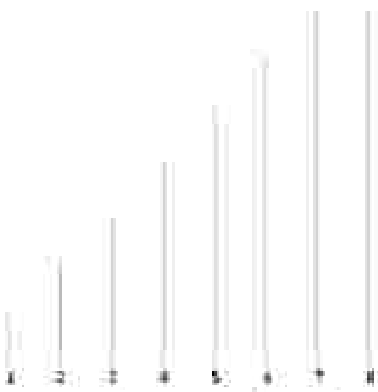
On obtient alors l'appoximation suivante :
$C_n \approx \frac{1}{T}\sum_{k=0}^{k=N-1} x(\frac{kT}{N})e^{-I\frac{2\pi nTk}{TN}} \times \frac{T}{N} \approx \frac{1}{N}\sum_{k=0}^{k=N-1} x(\frac{kT}{N})e^{-I\frac{2\pi nk}{N}}$

C'est cette écriture qui va nous donner la définition de la transformée de Fourier discrète. La pluspart des auteurs ne met pas le facteur $1/N$ situ devant le signe sigma.

3 TRANSFORMEE DE FOURIER DISCRETE : TFD

La transformée de Fourier discrète, notée TFD, est la représentation fréquentielle d'une suite temporelle finie. Cette suite est en général une suite de nombres $x(n)$ rels avec n variant de 0 à N-1, mais elle peut être * complexe si on mesure par exemple une intensité i avec son déphasage ϕ représenté par $z = ie^{j\phi}$. Dans l'étude d'un signal cette suite est obtenue en mesurant N valeurs d'un signal périodique . Autrement dit on échantillonne le signal en l'observant à intervalles de temps Δt égaux.

3.1 dfinition



Soit donc une suite $x(k)$ finie de nombres réels ou complexes pour k variant de 0 N-1. La transformée Fourier discrète (tfd) de $x(n)$ est la suite finie $X(n)$ de nombres complexes définie par : $X(n) = \sum_{k=0}^{k=N-1} x(k)e^{-I\frac{2\pi nk}{N}}$ où I est le nombre complexe de module 1 et d'argument $\frac{\pi}{2}$.
Si on note $\omega = e^{-I\frac{2\pi}{N}}$ la première racine nième de 1 rencontrée sur le cercle trigonométrique en partant de 1 et en allant dans le sens négatif on peut écrire cette relation :

$$X(n) = \sum_{k=0}^{k=N-1} x(k)w^{nk}$$

Ainsi les N valeurs $X(n)$ apparaissent comme les valeurs prises par le polynôme $P(z) = \sum_{k=0}^{k=N-1} x(k)z^n$ sur les N racines Nimes de l'unité : $1, w, w^2, ...w^{N-1}$.

On peut interpréter matriciellement cette définition :
Si on note A la matice , N lignes et N colonnes, de terme $a(i,j) = w^{(i-1)(j-1)}$, $\vec{x}$ le vecteur colonne formé par les n valeurs $x(k)$,

$\vec{X}$ le vecteur colonne formé par les n valeurs $X(n)$ on a : $\vec{X} = A.\vec{x}$,i et j varient de 1 à N.
La tfd est donc une application linéaire de $\mathbb{C}^n$ dans lui même de matrice A qui fait correspondre à un vecteur $\vec{x}$ (dit dans le domaine temporel) un vecteur $\vec{X}$ (dit dans le domaine fréquentiel en théorie du signal).

3.2 exemple

Soit pour fixer les idées N=8 et par exemple la suite $x(k)$ suivante $(1,2,3,4,5,6,7,8)$.
Alors la matrice A (8×8) est :

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \frac{\sqrt{2}}{2} - \frac{\sqrt{-1}\sqrt{2}}{2} & -\sqrt{-1} & -\frac{\sqrt{-1}\sqrt{2}}{2} - \frac{\sqrt{2}}{2} & -1 & -\frac{\sqrt{2}}{2} + \frac{\sqrt{-1}\sqrt{2}}{2} & \sqrt{-1} & \frac{\sqrt{-1}\sqrt{2}}{2} + \frac{\sqrt{2}}{2} \\
1 & -\sqrt{-1} & -1 & \sqrt{-1} & 1 & -\sqrt{-1} & -1 & \sqrt{-1} \\
1 & -\frac{\sqrt{-1}\sqrt{2}}{2} - \frac{\sqrt{2}}{2} & \sqrt{-1} & \frac{\sqrt{2}}{2} - \frac{\sqrt{-1}\sqrt{2}}{2} & -1 & \frac{\sqrt{-1}\sqrt{2}}{2} + \frac{\sqrt{2}}{2} & -\sqrt{-1} & -\frac{\sqrt{2}}{2} + \frac{\sqrt{-1}\sqrt{2}}{2} \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & -\frac{\sqrt{2}}{2} + \frac{\sqrt{-1}\sqrt{2}}{2} & -\sqrt{-1} & \frac{\sqrt{-1}\sqrt{2}}{2} + \frac{\sqrt{2}}{2} & -1 & \frac{\sqrt{2}}{2} - \frac{\sqrt{-1}\sqrt{2}}{2} & \sqrt{-1} & -\frac{\sqrt{-1}\sqrt{2}}{2} - \frac{\sqrt{2}}{2} \\
1 & \sqrt{-1} & -1 & -\sqrt{-1} & 1 & \sqrt{-1} & -1 & -\sqrt{-1} \\
1 & \frac{\sqrt{-1}\sqrt{2}}{2} + \frac{\sqrt{2}}{2} & \sqrt{-1} & -\frac{\sqrt{2}}{2} + \frac{\sqrt{-1}\sqrt{2}}{2} & -1 & -\frac{\sqrt{-1}\sqrt{2}}{2} - \frac{\sqrt{2}}{2} & -\sqrt{-1} & \frac{\sqrt{2}}{2} - \frac{\sqrt{-1}\sqrt{2}}{2}
\end{bmatrix}$$

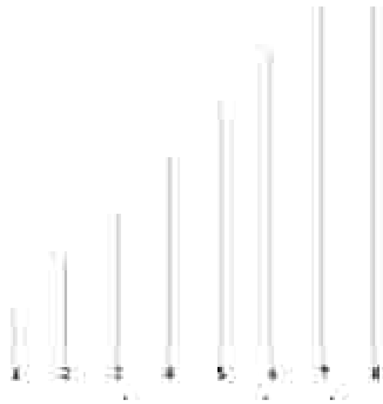
Multiplions la matrice A par le vecteur :

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{bmatrix}$$

pour obtenir son transformé Fourier discret : $\vec{X}$:

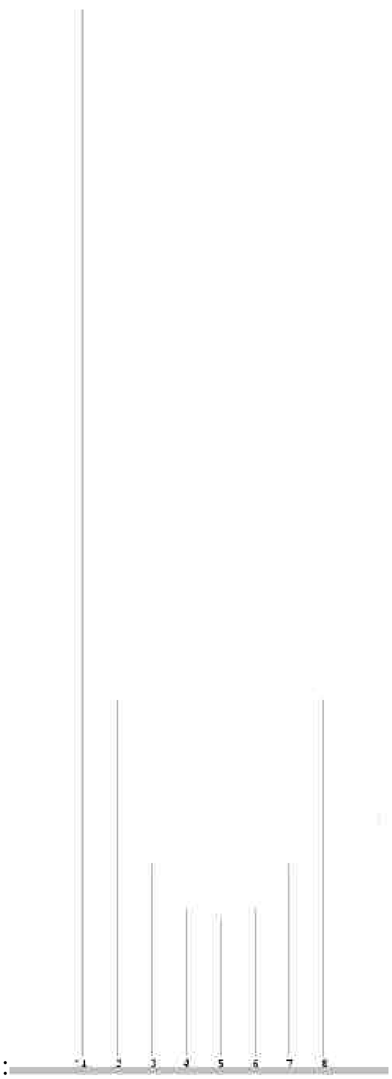
$$\begin{bmatrix} 36 \\ -4+4\sqrt{-1}\sqrt{2}+4\sqrt{-1} \\ -4+4\sqrt{-1} \\ -4+4\sqrt{-1}\sqrt{2}-4\sqrt{-1} \\ -4 \\ -4-4\sqrt{-1}\sqrt{2}+4\sqrt{-1} \\ -4-4\sqrt{-1} \\ -4-4\sqrt{-1}\sqrt{2}-4\sqrt{-1} \end{bmatrix}$$

Nous avions une suite réelle $x(k)$ et sa transformée ffd est $X(n)$ qui est complexe :
On peut représenter $x(k)$ et la suite des modules de $X(n)$ par des diagammes en bâtons. En terme de signal on dit que x(k) est le signal . Son transformé est son spectre.

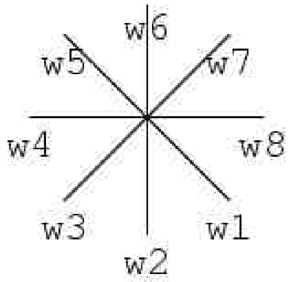
signal x(k) :  module du spectre $|X(n)|$

$$\begin{bmatrix} 36 \\ \sqrt{16+\left(4\sqrt{2}+4\right)^2} \\ 4\sqrt{2} \\ \sqrt{16+\left(4\sqrt{2}-4\right)^2} \\ 4 \\ \sqrt{16+\left(-4\sqrt{2}+4\right)^2} \\ 4\sqrt{2} \\ \sqrt{16+\left(-4\sqrt{2}-4\right)^2} \end{bmatrix}$$

Soit à peu près :36 ; 10.4 ; 5.6 ; 4.3 ; 4 ; 4.3 ;5.6 ; 10.4 ;

:

Soit l'application linéaire f8 de $\mathbb{C}^8$ dans lui même de matrice A . La tfd d'ordre N=8 évalue donc f8 sur le vecteur $\vec{x}$. La matrice A est formée des puissances successives de w avec $w = exp(-\frac{I2\pi}{8})$.



3.3 propriétés

La matrice A est symétrique. La première ligne et la premire colonne sont formées uniquement de 1, autrement dit $X(0) = \sum_{k=0}^{k=N-1} x(k)$.
Comme les racines nièmes de l'unité satisfont la relation de conjugaison $w^{N-i} = \overline{w}^i$, les lignes et les colonnes d'indice i et N-i sont conjuguées.

3.4 inverse de la ffd

Théorème fondamental : La tfd qui transforme la suite x(k) en

$$X(n) = \sum_{k=0}^{k=N-1} x(k)e^{-I\frac{2\pi nk}{N}}$$

admet pour fonction réciproque la transformation notée itfd (inverse transformée Fourier discrète) définie par :
$x(k) = \frac{1}{N}\sum_{n=0}^{n=N-1} X(n)e^{I\frac{2\pi nk}{N}}$ pour k variant de 0 à N-1.

Démonstration :

$$X(n) = \sum_{k=0}^{k=N-1} x(k)e^{-i\frac{2\pi nk}{N}} \text{ pour n=0..N-1}$$

Posons $z(k) = \displaystyle\sum_{n=0}^{n=N-1} x(n)e^{-I\frac{2\pi nk}{N}}$

Alors on a : $z(k) = \displaystyle\sum_{n=0}^{n=N-1} (\sum_{k'=0}^{k'=N-1} x(k')e^{-I\frac{2\pi(nk-nk')}{N}})$

$$z(k) = \sum_{n=0}^{n=N-1} (\sum_{k'=0}^{k'=N-1} x(k') e^{-I\frac{2\pi(n(k-k')}{N}})$$

$$z(k) = \sum_{n=0}^{n=N-1} x(k') (\sum_{k'=0}^{k'=N-1} e^{I\frac{2\pi(n(k'-k)}{N}})$$

On reconnait la somme des termes d'une suite géométrique de 1er terme 1 et de raison $e^{I\frac{2\pi(k'-k)}{N}}$ ayant N termes. Par suite on obtient :

$$z(k) = \sum_{k'=0}^{k'=N-1} x(k') \frac{e^{I2\pi(k'-k)} - 1}{e^{I\frac{2\pi(k'-k)}{N}} - 1}$$

Si $k' \neq k$ alors $e^{I2\pi(k'-k)} - 1 = 0$ donc

$$\sum_{n=0}^{n=N-1} e^{I\frac{2\pi n(k'-k)}{N}} = \frac{e^{I2\pi(k'-k)} - 1}{e^{I\frac{2\pi(k'-k)}{N}} - 1} = 0$$

Si $k' = k$ alors $\sum_{n=0}^{n=N-1} e^{I\frac{2\pi 0}{N}} = \sum_{n=0}^{n=N-1} 1 = N$

Donc : $z(k) = Nx(k)$

Donc la transformation inverse de :

$$x(k) \rightarrow X(n) = \sum_{k=0}^{k=N-1} x(k) e^{-I\frac{2\pi nk}{N}}$$

est :

$$X(n) \rightarrow x(k) = \frac{1}{N} \sum_{n=0}^{n=N-1} X(n) e^{I\frac{2\pi nk}{N}}$$

3.5 matrice de la itfd

La matrice B de l'inverse de la transformée dicrète a donc pour terme bij ,à la ième ligne et jième colonne : $b(i,j) = \frac{e^{I\frac{2\pi(i-1)(j-1)}{N}}}{N}$. Pour fixer les idées regardons ce que vaut B pour n=8

$$\begin{bmatrix} 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 \\ 1/8 & \frac{\sqrt{2}}{16}+\frac{\sqrt{-1}\sqrt{2}}{16} & \frac{\sqrt{-1}}{8} & -\frac{\sqrt{2}}{16}+\frac{\sqrt{-1}\sqrt{2}}{16} & -1/8 & -\frac{\sqrt{2}}{16}-\frac{\sqrt{-1}\sqrt{2}}{16} & -\frac{\sqrt{-1}}{8} & \frac{\sqrt{2}}{16}-\frac{\sqrt{-1}\sqrt{2}}{16} \\ 1/8 & \frac{\sqrt{-1}}{8} & -1/8 & -\frac{\sqrt{-1}}{8} & 1/8 & \frac{\sqrt{-1}}{8} & -1/8 & -\frac{\sqrt{-1}}{8} \\ 1/8 & -\frac{\sqrt{2}}{16}+\frac{\sqrt{-1}\sqrt{2}}{16} & -\frac{\sqrt{-1}}{8} & \frac{\sqrt{2}}{16}+\frac{\sqrt{-1}\sqrt{2}}{16} & -1/8 & \frac{\sqrt{2}}{16}-\frac{\sqrt{-1}\sqrt{2}}{16} & \frac{\sqrt{-1}}{8} & -\frac{\sqrt{2}}{16}-\frac{\sqrt{-1}\sqrt{2}}{16} \\ 1/8 & -1/8 & 1/8 & -1/8 & 1/8 & -1/8 & 1/8 & -1/8 \\ 1/8 & -\frac{\sqrt{2}}{16}-\frac{\sqrt{-1}\sqrt{2}}{16} & \frac{\sqrt{-1}}{8} & \frac{\sqrt{2}}{16}-\frac{\sqrt{-1}\sqrt{2}}{16} & -1/8 & \frac{\sqrt{2}}{16}+\frac{\sqrt{-1}\sqrt{2}}{16} & -\frac{\sqrt{-1}}{8} & -\frac{\sqrt{2}}{16}+\frac{\sqrt{-1}\sqrt{2}}{16} \\ 1/8 & -\frac{\sqrt{-1}}{8} & -1/8 & \frac{\sqrt{-1}}{8} & 1/8 & -\frac{\sqrt{-1}}{8} & -1/8 & \frac{\sqrt{-1}}{8} \\ 1/8 & \frac{\sqrt{2}}{16}-\frac{\sqrt{-1}\sqrt{2}}{16} & -\frac{\sqrt{-1}}{8} & -\frac{\sqrt{2}}{16}-\frac{\sqrt{-1}\sqrt{2}}{16} & -1/8 & -\frac{\sqrt{2}}{16}+\frac{\sqrt{-1}\sqrt{2}}{16} & \frac{\sqrt{-1}}{8} & \frac{\sqrt{2}}{16}+\frac{\sqrt{-1}\sqrt{2}}{16} \end{bmatrix}$$

Vérifions que lorsque l'on multiplie A par B on obtient l'identité.

```
with(linalg):f:=(i,j)->exp(-I*2*Pi*(i-1)*(j-1)/8):A:=matrix(8,8,f);
multiply(A,B));
```

on obtient bien la matrice de l'identité :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Nous obtenons avec ces deux transformations un procédé de composition-déconposition semblable au prisme qui décompose et recompose la lumière, mais que l'on applique à un suite de nombres. Dans la théorie du signal on transforme un signal en son spectre.

3.6 programme maple de tfd

```
> restart:
> dft:=proc(L)
  local k,n,w,N,X;
  N:=nops(L); X:=vector(N,L);
  w:=evalf(exp(-2*I*Pi/N));
 for n from 1 to N do
 X[n]:=sum('L[k]*w^((n-1)*(k-1))','k'=1..N);
X[n]:=expand(X[n]);
od;
RETURN(evalf(evalm(X)))
end:
A:=dft([1,2,3,4,5,6,7,8]);


> idft:=proc(L)
  local k,n,w,N,X;
  N:=nops(L); X:=vector(N,L);
  w:=evalf(exp(2*I*Pi/N));
 for n from 1 to N do
 X[n]:=sum('L[k]*w^((n-1)*(k-1))','k'=1..N);
X[n]:=expand(X[n]/N);
od;
RETURN(evalf(evalm(X))) end:
> B:=idft([36., -4.+9.656854248*I, -4.+4.*I, -4.+1.656854248*I,
-4., -4.-1.656854248*I, -4.-4.*I, -4.-9.656854248*I]);

> latex(A);latex(B);
```

$A = [36.0, -3.999999{+}9.65685\,i, -3.999999{+}3.99999\,, i, -3.99999{+}1.656854\,i, -3.999999, -3.99999$

$B = [1.0{+}0.000005\,i, 2.000, 3.000{-}0.000000\,i, 4.0000{+}0.0000000\,i, 4.999999999{-}0.00000\,i, 5.999999$

avec des arrondis.

Le défaut de ces programmes est de faire n^2 multiplications suivies d'un peu moins d'additions. Dés que n est grand les temps de calculs deviennent dissuasifs.

```
> randlist:=proc(n) local f; f:=rand(-50..50); [seq(f(),k=1..n)] end;

> u:=randlist(128);

> s:=time();a:=dft(u);time()-s;
```

Pour calculer la transformée d'une suite de 128 nombres il faut avec mon ordinateur qui a 4 ans 42 secondes, pour 256 nombres 195 secondes. Avec des suites plus importantes on perd patience. La transformée de fourier rapide tfr (ou fast fourier transform fft en anglais) va ramener un calcul à environ $n * log_2(n)$ opérations.

4 algorithme récursif de la transformée de Fourier rapide tfr

Cet algorithme utilise les propriétés de la tfd et ramène le calcul de la tfd de taille N à deux transformées de taille N/2 , suivi de N/2 multiplications. On applique récursivement cette propriété, c'est pourquoi on part de N qui est une puissance de 2.

On veut donc calculer :

$X(k) = \sum_{t=0}^{t=N-1} x(t)e^{-I\frac{2\pi kt}{N}}$

Si on note $w = e^{-I\frac{2\pi}{N}}$ la première racine nième de 1 rencontrée sur le cercle t rigonométrique en partant de 1 et en allant dans le sens négatif on peut écrire cette relation :

$$X(k) = \sum_{t=0}^{t=N-1} x(t)w^{tk}$$

Posons $N = 2p$ et regroupons les termes d'indices pairs d'une part dans P_k et d'autre part , dans les termes d'indices impairs mettons w^k en facteur $w^k \times I_k$:

$P_k = x(0) + x(2)w^{2k} + ... + x(N-2)w^{(N-2)k}$

$w^k \times I_k = x(1)w^k + x(3)w^{3k} + ... + x(N-1)w^{(N-1)k} = w^k(x(1) + x(3)w^{2k} + ... + x(N-1)w^{(N-2)k})$

avec $I_k = x(1) + x(3)w^{2k} + ... + x(N-1)w^{N-2}$

On a alors : $X(k) = P_k + w^k I_k$ On a aussi $P_{k+p} = x(0) + x(2)w^{2k+2p} + ... +$

$x(2N-2)w^{(N-2)(k+p)}$ or $w^{2p} = w^N = 1$ donc $P_{k+p} = P_k$
On a la même propriété pour les impairs : $I_{k+p} = I_k$

Donc $X_{k+p} = P_{k+p} + w^{k+p} I_{k+p} = P_k - w^k I_k$
Ainsi pour calculer $X(k)$ il suffit de calculer seulement pour k de 0 à $p-1 = \frac{N}{2} - 1$ P_k et $w^k \times I_k$
Ajouter ces deux résultats pour obtenir : $X(k) = P_k + w^k \times I_k$
Les soustraire pour obtenir : $Xk + p = P_k - w^k \times I_k$.
On divise ainsi le nombre de multiplications par 2.
En prenant $N = 2^n$ on peut réitérer cet algorithme .
Voici donc l'algorithme récursif de Cooley-Tuckey pour calculer la transformée Fourier rapide d'une suite x(t) prenant $N = 2^p$ valeurs.
début
1) Calculer $w = e^{-2I\pi/N}$
2)Si n=1 rendre x(t) sinon
21) poser $m = \frac{N}{2}$
22) appeler la transformée de Fourier avec les $m = \frac{N}{2}$ points de coordonnées paires et w^2 , mettre le résultat dans X
23) appeler la transformée de Fourier avec les $m = \frac{N}{2}$ points de coordonnées im paires et w^2 , mettre le résultat dans Y
fin de si
3) pose u=1.
4) pour k de 1 à m-1 faire :
41) stocker $(X(k) + u \times Y(k)$ dans Z(k)
42) stocker $(X(k) - u \times Y(k)$ dans Z(k+m)
43) remplacer u par $u \times w$
fin faire
5) Rendre la suite Z(k)
fin

Voici le programme Maple correspondant :

```
> fftr:=proc(z)
  local k,n,w,x,y,X,Y,Z;
  n:=nops(z); if n=1 then RETURN(evalm(z)) fi;
  x:=[seq(z[2*k-1],k=1..n/2)]; y:=[seq(z[2*k],k=1..n/2)];
  X:=fftr(x); Y:=fftr(y); w:=exp(-2*I*Pi/n);
```

```
  Z:=[seq(expand(X[k]+w^(k-1)*Y[k]),k=1..n/2),seq(expand(X[k]-w^(k-1)*Y[k]),k=1..n
  RETURN(evalm(Z))
end:
> fftr([1,2,3,4,5,6,7,8],1);
```

$[36, -4+4\,i+4\,i\sqrt{2}, -4+4\,i, -4-4\,i+4\,i\sqrt{2}, -4, -4+4\,i+-4\,i\sqrt{2}, -4-4\,i, -4-4\,i+-4\,i\sqrt{2}]$

5 algorithme itératif de tfr pour N=8

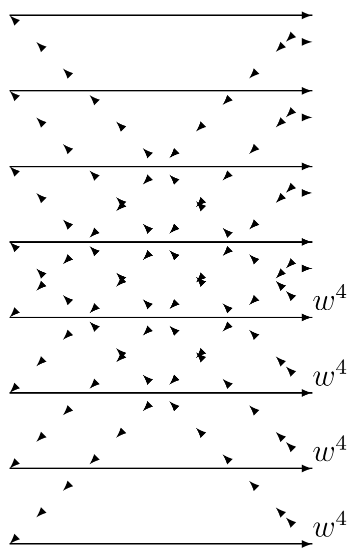
Le programme récursif ci-dessus est gourmand en mémoire. Il part du signal à trouver et le décompose en 2, ces morceaux sont décomposé en deux ...etc. On part de la fin. Nous allons maintenant partir du signal donné x(k) et faire des étapes successives , chaque étape groupant les résultats précédents , pour arriver au signal transformé. La sucession des calculs peut trés facilement se suivre "à la main".
Nous expliquons sur le cas particulier $N = 2^3 = 8$ cet algorithme. Le cas général est alors bien plus compréhensible.
Soit donc une suite x(k) pour k variant de 0 à 7, représentant le signal initial.
Décomposons la variable k en base 2 : $k = k_2 \times 2^2 + k_1 \times 2 + k_0$.
Le transformé est la suite $X(n) = \sum_{k=0}^{k=N-1} x(k)w^{nk}$ avec $w = e^{-2I\pi/8}$
Décomposons aussi la variable n en base 2 : $n = n_2 \times 2^2 + n_1 \times 2 + n_0$.
On a $w^8 = 1$. Alors en développant $(n_2 \times 4 + n_1 \times 2 + n_0)(k_2 \times 4 + k_1 \times 2 + k_0)$ on obtient pour le premier terme $w^{n4k_2} = w^{16n_2k_2+8n_1k_1+n_04k_2} = w^{n_04k_2}$ tout le reste disparait comme puissance de 1. Pour le deuxième terme du développement on a : $w^{n2k_1} = w^{8n_2k_1+(2n_1+n_0)2k_1} = w^{(2n_1+n_0)2k_1}$.
Pour le troisième terme on trouve : $w^{n\times k_0} = w^{(4n_2+2n_1+n_0)k_0}$ qui n'est pas simplifiable.
La tfr pour N=8 points s'écrit :

$$X(4n_2+2n_1+n_0) = \sum_{k_0=0}^{k_0=1}\sum_{k_1=0}^{k_1=1}\sum_{k_2=0}^{k_2=1} x(4k_2+2k_1+k_0)w^{n_04k_2} = w^{(2n_1+n_0)2k_1}w^{(4n_2+2n_1+n_0)k_0}$$

Posons :

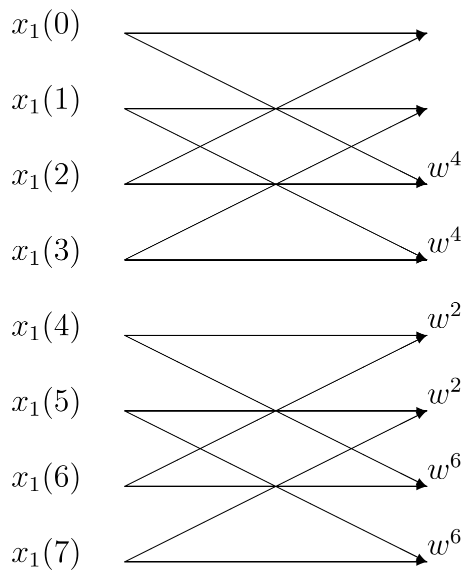
$x_1(4n_0 + 2k_1 + k_0) = \sum_{k_2=0}^{k_2=1} x(4k_2 + 2k_1 + k_0)w^{n_04k_2} = x(2k_1 + k_0) + x(4 + 2k_1 + k_0)w^{4n_0}$ (on somme par rapport à k_2 d'où l'augmentation de 4 de l'in-

dice à chaque boucle de calcul)
$x_1(4n_0 + 2k_1 + k_0)$ est le premier signal (ou suite) intermédiaire. Les calculs sont montrés dans le tableau suivant :



x(0)	$x_1(0) = x(0) + x(4)$
x(1)	$x_1(1) = x(1) + x(5)$
x(2)	$x_1(2) = x(2) + x(6)$
x(3)	$x_1(3) = x(3) + x(7)$
x4)	$x_1(4) = x(0) - x(4)$
x(5)	$x_1(5) = x(1) - x(5)$
x(6)	$x_1(6) = x(2) - x(6)$
x(7)	$x_1(7) = x(3) - x(7)$

Pour le deuxième signal obtenu on a :
$x_2(4n_0 + 2n_1 + k_0) = \sum_{k_1=0}^{k_1=1} x_1(4n_0 + 2k_1 + k_0)w^{(2n_1+n_0)2k_1} = x_1(4n_0 + k_0) + x_1(4n_0+2+k_0)w^{(2n_1+n_0)2}$ on somme par rapport à k_1 , d'où l'augmentation de 2 pour l'indice à chaque boucle de calcul , l'exposant de w étant $(2n_1+n_0)2k_1$. Les calculs du deuxième signal sont donnés dans le tableau suivant :



$x_2(0) = x_1(0) + x_1(2)$

$x_2(1) = x_1(1) + x_1(3)$

$x_2(2) = x_1(0) - x_1(2)$

$x_2(3) = x_1(1) - x_1(3)$

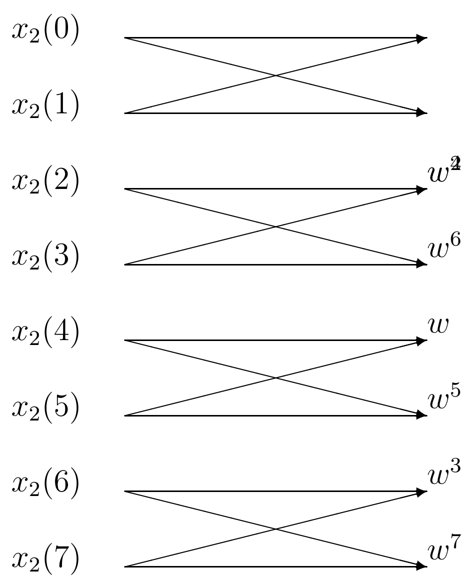
$x_2(4) = x_1(4) + x_1(6)w^2$

$x_2(5) = x_1(5) + x_1(7)w^2$

$x_2(6) = x_1(4) + x_1(6)w^6$

$x_2(7) = x_1(5) + x_1(7)w^6$

La dernière sommation nous donne le dernier signal. Explicitons les calculs.
$x_3(4n_0 + 2n_1 + n_2) = \sum_{k_0=0}^{k_0=1} x_2(4n_0 + 2n_1 + k_0)w^{(4n_2+2n_1+n_0)k_0} = x_2(4n_0 + 2n_1) + x_2(4n_0 + 2n_2 + 1)w^{4n_2+2n_1+n_0}$, on somme par apport à k_0 , donc l'indice augmente de 1 à chaque calcul. Voici le tableau des calculs correspondants :



$x_3(0) = x_2(0) + x_2(1)$

$x_3(1) = x_2(0) + x_2(1)w^4$

$x_3(2) = x_2(2) + x_2(3)w^2$

$x_3(3) = x_2(2) + x_2(3)w^6$

$x_3(4) = x_2(4) + x_2(5)w$

$x_3(5) = x_2(4) + x_2(5)w^5$

$x_3(6) = x_2(6) + x_2(7)w^3$

$x_3(7) = x_2(6) + x_2(7)w^7$

La transformée tfr X(n) du signal x(k) s'obtient à partir du dernier signal intermédiaire $x_3(k)$ par la formule :
$X(4n_2 + 2n_1 + n_0) = x_3(4n_0 + 2n_1 + n_0)$
Autrement dit la pième composante de X s'obtient en écrivant p en base 2 : $p = 4n_2 + 2n_1 + n_0$ en récrivant son écriture dans le sens contraire pour obtenir l'indice de x_3 $k = 4n_0 + 2n_1 + n_2$ .C'est à cette position dans x_3 qu'il faut lire $X(n)$. Il faut renverser les bits de l'écriture de p en base 2.
Donc $X(0) = x_3(0)$; $X(1) = x_3(4)$; $X(2) = x_3(2)$; $X(3) = X(2+1) = x_3(4+2) = x_3(6)$; $X(4) = x_3(1)$; $X(6) = X(4+2) = x_3(2+1) = x_3(3)$; $X(7) = x_3(7)$

6 algorithme itératif de tfr pour $N = 2^r$

Le cas général est tout à fait semblable au cas particulier que nous venons de traiter.

$$X[2^{r-1}n_{r-1}+2^{r-2}n_{r-2}+....+2n_1+n_0) = \sum_{k_0=0}^{k_0=1}\sum_{k_1=0}^{k_1=1}....\sum_{k_{r-1}=0}^{k_{r-1}=1} x(2^{r-1}k_{r-1}+2^{r-2}k_{r-2}+....+2k_1+k_0)w^{nk}$$

avec $n = 2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + ... + 2n_1 + n_0$, $k = 2^{r-1}k_{r-1} + 2^{r-2}k_{r-2} + ... + 2k_1 + k_0$, $w = e^{I2\pi/N}$, $w^N = 1$

En simplifiant le développement de w^{nk} on obtient :
Pour le premier signal intermédiaire :

$$x_1(2^{r-1}n_0+2^{r-2}k_{r-2}+....+2k_1+k_0) = \sum_{k_{(r-1)=0}^{(k_{r-1}=1}} x(2^{r-1}k_{r-1}+2^{r-2}k_{r-2}+....+2k_1+k_0)w^{n_0 2^{r-1}k_{r-1}}$$

Soit :
$x_1(2^{r-1}n_0 + 2^{r-2}k_{r-2} + ... + 2k_1 + k_0) = x(0 + k_{r-2}2^{r-2} + ... + k_1 2 + k_0) + w^{2^{r-1}n_0}x(1 \times 2^{r-1} + 2^{r-2}k_{r-2} + ... + 2k_1 + k_0)$ la somme se fait en faisant varier le premier indice en partant de la gauche.

Pour le deuxième signal intermédiaire :
$x_2(2^{r-1}n_0 + 2^{r-2}n_1 + 2^{r-3}k_{r-3}.... + 2k_1 + k_0) = \sum_{k_{(r-2)=0}^{(k_{r-2}=1}} x_1(2^{r-1}n_0 + 2^{r-2}k_{r-2} + ... + 2k_1 + k_0)w^{(2n_1+n_0)2^{r-2}k_{r-2}}$ Soit :

$x_2(2^{r-1}n_0 + 2^{r-2}n_1 + 2^{r-3}k_{r-3}.... + 2k_1 + k_0) = x_1(2^{r-1}n_0 + 2^{r-2} \times 0 + 2^{r-3}k_{r-3}....+2k_1+k_0)+w^{2^{r-2}(2n_1+n_0)}x_1(2^{r-1}n_0+2^{r-2}\times 1+2^{r-3}k_{r-3}....+2k_1+k_0)$
On somme en faisant varier le deuxième indice en partant de la gauche.

Le troisime signal intermédiaire s'écrit en fonction du deuxiéme :

$x_3(2^{r-1}n_0 + 2^{r-2}n_1 + 2^{r-3}n_3 + 2^{r-4}k_{r-4} + .. + 2k_1 + k_0) = x_2(2^{r-1}n_0 + 2^{r-2}n_1 + 2^{r-3} \times 0 + 2^{r-4}k_{r-4} + .. + 2k_1 + k_0) + w^{2^{r-3}(2^2n_2+2n_1+n_0)}x_2(2^{r-1}n_0 + 2^{r-2}n_1 + 2^{r-3} \times 1 + 2^{r-4}k_{r-4} + .. + 2k_1 + k_0)$.
On somme par rapport au troisiéme indice en partant de la gauche.
- ....

Le m-iéme signal intermédiaire se calcule é partir du précédent par la formule suivante :

$x_m(2^{r-1}n_0 + 2^{r-2}n_1 + 2^{r-3}n_3 + ... + 2^{m-1}n_{m-1} + ... + 2^{r-m-1}k_{m-r-1} + .. + 2k_1 + k_0) = x_{m-1}(2^{r-1}n_0 + 2^{r-2}n_1 + 2^{r-3}n_3 + ... + 2^{m-2}n_{m-2} + 2^{m-1} \times 0 + 2^{r-m-1}k_{m-r-1} + .. + 2k_1 + k_0) + w^{2^{r-m}(2^{m-2}n_{m-2}+...+2n_1+n_0)}x_{m-1}(2^{r-1}n_0 + 2^{r-2}n_1 + 2^{r-3}n_3 + ... + 2^{m-2}n_{m-2} + 2^{m-1} \times 1 + 2^{r-m-1}k_{m-r-1} + .. + 2k_1 + k_0)$
La somme se fait en faisant varier le m-iéme indice en partant de la gauche.

Le dernier signal intermédiaire est :
$x_r(2^{r-1}n_0 + 2^{r-2}n_1 + ... + 2n_{r-2} + n_{r-1}) = \sum_{k_0=0}^{k_0=1} x_{r-1}(2^{r-1}n_0 + 2^{r-2}n_1 + ... + 2n_{r-2} + n_0)w^{(2^{r-1}n_{r-1}+2^{r-2}n_{r-2}+..2n_1+n_0)k_0}$ soit :
$x_r(2^{r-1}n_0 + 2^{r-2}n_1 + ... + 2n_{r-2} + n_{r-1} = x_{r-1}(2^{r-1}n_0 + 2^{r-2}n_1 + ... + 2n_{r-2} + 0) + w^{2^{r-1}n_0+2^{r-2}n_{r-2}+...+2n_1+n_0}x_{r-1}r2^{r-1}n_0 + 2^{r-2}n_1 + ... + 2n_{r-2} + 1)$
La tfr se déduit du dernier signal intermédiaire par renversement de ses indices.

D'oùl'algorithme :
début
n=la dimension du tableau $ll[k]$ contenant les valeurs de x(k), k variant de 1 à n.
m=n/2
tant que $m \geq 1$ faire :
z=1
(c'est la premiére valeur de w^k)

$w = e^{-I\pi/m}$ (c'est la valeur de w)
Pour k de 1 jusqu'à m faire :
Pour h de 1 jusqu'à $n/(m*2)$ faire (c'est la boucle de calcul du signal intermédiaire x_k)
$i = k + 2 \times (h-1) \times m$
$j = i + m$
$tmp = (ll[i] - ll[j]) \times z$
$ll[i] = ll[i] + ll[j]$
$ll[j] = tmp$
fin faire
$z = zw$
fin faire
$m = m/2$
fin de tant que
Remettre le tableau x_m dans le bon ordre en renversant les indices écrits en base deux. On commence par leur soustraire 1 pour qu'ils varient de 0 à n-1. Après l'écriture dans l'autre sens , on rajoute 1 pour revenir dans l'intervalle 1 à n.

Voici ce que cela donne pour la calculatrice Ti 89 ou Ti 92 . Le programme br (= bitreverse) fait le renversement de l'indice.

Fonction fft(ll)
Local n,p,j,k,u,m,z,z0,i,h,tmp
dim(ll) sto n
int(ln(n)/(ln(2))) sto p
n/2 sto m
While $m \geq 1$
1 sto z
$e^{-I\pi/m} sto z_0$
For k,1,m
For h,1,n/(2*m)
k+2*(h-1)*m sto i
i+m sto j
(ll[i]-ll[j])*z sto tmp
ll[i]+ll[j] sto ll[i]
tmp sto ll[j] EndFor
z*z0 sto z EndFor

m/2sto m
EndWhile
For j,1,n/2
br(j-1,3) sto k
ll[k+1] sto u
ll[j] sto ll[k+1]
u sto ll[j]

EndFor
Return ll
EndFunc

fonction br(k,n)
Func
Local a,i,q,r,x
0sto a :k sto x
For i,0,n-1
int(x/2) sto q
x-2*q sto r
$a + r * 2^{(n-i-1)}$sto a
q sto x
EndFor
Return a
EndFunc

7 programme itératif de tfr et de son inverse itfr en Maple

```
> restart:
> fft:=proc(L)
  local h,i,j,k,l,m,n,p,tmp,w,z;
  n:=nops(L); l:=L; m:=n/2;
  while m>=1 do
      z:=1; w:=evalf(exp(-I*Pi/m));
      for k from 1 to m do
```

```
            for h from 1 to n/(2*m) do
                i:=k+2*(h-1)*m;
                j:=i+m;
                tmp:=(l[i]-l[j])*z;
                l[i]:=l[i]+l[j];
                l[j]:=tmp
            od;
            z:=evalf(w*z)
        od;
        m:=m/2
    od;
    j:=1;
    for i from 1 to n do
          if j>i then tmp:=l[j]; l[j]:=l[i]; l[i]:=tmp fi;
          m:=n/2;
          while m>=2 and j>m do j:=j-m; m:=m/2 od;
          j:=j+m
    od;
    RETURN(evalf(evalf(l)))
end:
> fft([1,2,3,4,5,6,7,8],1);


> ifft:=proc(L)
  local h,i,j,k,l,m,n,p,tmp,w,z;
  n:=nops(L); l:=L; m:=n/2;

for j from 1 to n do
l[j]:=l[j]/n

od:
  while m>=1 do
      z:=1; w:=evalf(exp(I*Pi/m));
      for k from 1 to m do
          for h from 1 to n/(2*m) do
              i:=k+2*(h-1)*m;
              j:=i+m;
              tmp:=(l[i]-l[j])*z;
```

```
            l[i]:=l[i]+l[j];
            l[j]:=tmp
         od;
         z:=evalf(w*z)
      od;
      m:=m/2
   od;
   j:=1;
   for i from 1 to n do
         if j>i then tmp:=l[j]; l[j]:=l[i]; l[i]:=tmp fi;
         m:=n/2;
         while m>=2 and j>m do j:=j-m; m:=m/2 od;
         j:=j+m
   od;
   RETURN(evalfl))
end:
```

```
 a:=fft([1,2,3,4,5,6,7,8]);print(a);b:=ifft(a):c:=array(1..nops(b)):
for i to nops(b) do c[i]=round(b[i]) od: print (c);
```

$a = [36.0, -4.000000002{+}9.656854244\,i, -4.0{+}4.0\,i, -3.999999998{+}1.656854248\,i,$
$-4.0, -3.999999998{-}1.656854248\,i, -4.0{-}4.0\,i, -4.000000002{-}9.656854244\,i]$

$c = [1, 2, 3, 4, 5, 6, 7, 8]$

Regardons maintenant le gain de temps obtenu avec la tfr par rapport à la tfd en la faisant opérer sur 256 nombres :

```
> randlist:=proc(n) local f; f:=rand(-50..50); [seq(f(),k=1..n)] end;

> u:=randlist(256);

> s:=time();a:=fft(u);t:=time()-s;
```

L'exécution de la tfr est faite en 0.535 secondes alors que la tfd nécessite 195 secondes!!!.

8 calcul du nombre d'opérations effectuées dans la tfr

D'après ce qui précède , nous constatons que le calcul des N valeurs de chaque signal intermédiaire exige sensiblement N opérations ; D'autre part , le nombre de signaux intermédiaires est $r = log_2(N)$. La tfr exige donc un nombre d'opérations de l' ordre de $Nlog_2(N)$.
Dans l'exemple donné avec $N = 8 = 2^3$ ce nombre est 24 alors qu'un calcul direct par un produit de matrices aurait donné 64 opérations.

9 utilsation de la fft pour multiplier deux polynômes

Nous savons qu'un polynôme P(x) de degré n-1 est parfaitement déterminé par ses valeurs prises sur n valeurs $x(i)$, i=0..n. Soit donc un polynôme P(x) . Calculons par la fft les valeurs prises par P(x) sur les 2n racines 2nièmes de 1. On prend 2n comme une puissance de 2 en complétant éventuellements par des coefficients nuls. Nous obtenons alors une suite c(i) de complexes i=0..2n. Effectuons ensuite la multiplication $d(i) = c(i) \times d(i)$ pour i=0..2n. Cette suite d(i) obtenue représente les valeurs du polynôme $Q(x) = P(x) \times P(x)$ sur les 2n racines 2nièmes de l'unité. Comme le degré de Q(x) est inférieur à 2n-1, il est parfaitement déterminé par ces valeurs d(i). La transformée inverse ifft de ces 2n points nous donne donc Q(x) produit de P(x) par lui même. Le nombre d'opérations est bien moindre que la multiplication terme à terme de P(x).
Avec la fft suivie de ifft on a sensiblement $2nlog_2(n)$ au lieu de n^2 multiplications.
Présentons ce calcul sur un exemple simple à vérifier : $P(x) = (1+x)^{15}$ On calcule les coefficients de P(x) .Ensuite on appelle fft :

```
\\
\begin{verbatim}
> a:=array(1..32):a[1]:=1:for i to 15 do a[i+1]:=(15!/(i!*(15-i)!)) od:
for i to 16 do a[i+16]:=0 od:latex(a);a:=[seq(a[i],i=1..32)];b:=fft(a);
```

$a = [1, 15, 105, 455, 1365, 3003, 5005, 6435, 6435, 5005, 3003, 1365, 455, 105, 15, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,$
Ensuite on multiplie les coefficients b(i) par eux mêmes :

```
> c:=array(1..32):for i to 32 do c[i]:=(b[i]*b[i]) od:a:=[seq(c[i],i=1..32)];
```

On calcule ensuite l'inverse fft de la suite obtenue . C'est la suite des coefficients du polynôme Q(x) cherché.

```
> d:=ifft(a): for i to 32 do d[i]:=round(d[i]) od;latex(d);
```

$1, 30, 435, 4060, 27405, 142506, 593775, 2035800, 5852925, 14307150, 30045015, 54627300, 86493225, 119759850, 145422675, 155117520, 145422675, 119759850, 86493225, 54627300, 30045015, 14307150, 5852925, 2035800, 593775, 142506, 27405, 4060, 435, 30, 1, 0$

Comparons avec le développement fourni par Maple : expand$((1+x)^{30}) =$ $1+435\,x^2+4060\,x^3+27405\,x^4+142506\,x^5+593775\,x^6+2035800\,x^7+5852925\,x^8+14307150\,x^9+30045015\,x^{10}+54627300\,x^{11}+86493225\,x^{12}+119759850\,x^{13}+145422675\,x^{14}+155117520\,x^{15}+145422675\,x^{16}+119759850\,x^{17}+86493225\,x^{18}+54627300\,x^{19}+30045015\,x^{20}+14307150\,x^{21}+5852925\,x^{22}+2035800\,x^{23}+593775\,x^{24}+142506\,x^{25}+27405\,x^{26}+4060\,x^{27}+435\,x^{28}+30\,x^{29}+x^{30}+30\,x$

Maple ne met pas les puissances dans le bon ordre, mais on a bien les coefficients désirés .

10 bibliographie

Numerical Recipes in C : the art of scientific computing (ISBN 0-521-43108-5).
http ://www.essi.fr/ leroux/courssignal/node12.html
An algorithm for the machine calculation of complex Fourier series . by James Cooley and John W .Tukey.Mathematics of computation.
Théorie et applications de la transformation de Fourier rapide. J. Lifermann. Masson.
Cours d'algèbre . Michel Demazure. Nouvelle bibliothèque . Mathématique. Cassini.