Chapitre 10

Transformée de Fourier rapide (FFT)[1]

1 Définition

La FFT (Fast Fourier Transform) est un algorithme de calcul rapide de la transformée de Fourier discrète. La transformée de Fourier discrète est une application linéaire de $\mathbb{C}^N$ dans $\mathbb{C}^N$ avec $\underline{N = 2^M}$. On note I, l'imaginaire tel que $I^2 = -1$. On définit cette application linéaire par :

$$s : g \longmapsto G$$

où

$$G_j = \sum_{k=0}^{2^M-1} g_k * W_N^{j*k}$$

avec $W_N = \exp^{\frac{2I\pi}{N}}$. On associe à cette application s la matrice S telle que $Sg = G$ avec $S = (W_N^{jk})$.
Le problème est de réduire le temps de calcul de ce produit matrice vecteur qui est en $O(N^2)$.

2 La permutation miroir

On définit la permutation miroir en fonction de M par :

Soit n un entier $\in [0, 2^M - 1]$, on associe $r_M(n)$. On peut décomposer n en base 2 : $n = \sum_{i=0}^{M-1} \lambda_i * 2^i$. On aura alors $r_M(n) = \sum_{i=0}^{M-1} \lambda_{M-1-i} * 2^i$. On a donc échangé l'ordre des bits (la plus grande puissance étant à gauche) :

$$(\lambda_{M-1}, \lambda_{M-2}, \cdots, \lambda_1, \lambda_0)_2 \longmapsto (\lambda_0, \lambda_1, \cdots, \lambda_{M-2}, \lambda_{M-1})_2$$

Cette permutation miroir a un rôle majeur. En effet, elle permet de transformer la matrice S en une matrice T possédant des propriétés intéressantes. On remarque que r_m est involutive : $r_M(r_M(n)) = n$.

3 Transformation du problème

Nous allons appliquer cette permutation miroir aux lignes de G, c'est à dire que l'on va changer l'ordre des lignes :
on permute la i^{eme} ligne et la $r_M(i)^{eme}$ de G (<u>on numérote les lignes de 0 à $N-1$</u>). Nous avons :

$$G = Sg \longmapsto G^* = PSg = T^{(M)}g$$

avec P la matrice de permutation. L'avantage d'effectuer cette opération est que $T^{(M)}$ vérifie :

$$T^{(M)} = \begin{pmatrix} T^{(M-1)} & T^{(M-1)} \\ T^{(M-1)} * L^{(M-1)} & -T^{(M-1)} * L^{(M-1)} \end{pmatrix}$$

1. Je remercie Antoine Tonnoir, élève GM3 (2008-2009) de la première transcription en TEX des notes du cours ANA sur la FFT (hors application).

$L^{(M-1)}$ est une matrice diagonale. En effet, soit $(\delta_{i,j})$ le bloc inférieur gauche de la matrice. Nous avons :

$$\delta_{i,j} = S_{r_M(i),j} = W_N^{r_M(i)*j} = \exp^{\frac{2I\pi}{2^M}*\sum_{l=0}^{M-1}\lambda_{M-1-l}*2^l*j}$$

$$\iff \delta_{i,j} = \exp^{\frac{2I\pi}{2^{M-1}}*\sum_{l=0}^{M-2}\lambda_{M-2-l}*2^l*j+\frac{2I\pi}{2^M}*\lambda_{M-1}*j}$$

$$\iff \delta_{i,j} = \exp^{\frac{2I\pi}{2^{M-1}}*\sum_{l=0}^{M-2}\lambda_{M-2-l}*2^l*j} * \exp^{\frac{2I\pi}{2^M}*\lambda_{M-1}*j}$$

Pour les lignes de numéros impairs nous avons le dernier bit à 1 donc $r_M(i)$ est grand car son bit de point fort sera à 1. On a donc dans δ $\lambda_{M-1} = 1$ d'ou :

$$\delta_{i,j} = \exp^{\frac{2I\pi}{2^{M-1}}*\sum_{l=0}^{M-2}\lambda_{M-2-l}*2^l*j} * \exp^{\frac{2I\pi}{2^M}*j}$$

Nous sommes dans le bloc δ donc, la ligne i a pour indice : $i \geq 2^{M-1}$. On a donc $\sum_{l=0}^{M-2}\lambda_{M-2-l} * 2^l = r_M(i) - 2^{M-1}$. D'où $\delta_{i,j} = T^{(M-1)}_{i-2^{M-1},j} * L^{(M-1)}_{j,j}$. Nous remarquons que $L^{(M-1)}$ est une matrice diagonale avec $L^{(M-1)}_{j,j} = \exp^{\frac{2I\pi}{2^M}*j}$.

En notant $u_0 = g$, nous avons :

$$T^{(M)}g = T^{(M)}u_0 = \begin{pmatrix} T^{(M-1)} & 0 \\ 0 & T^{(M-1)} \end{pmatrix} * \begin{pmatrix} I^{(M-1)} & I^{(M-1)} \\ L^{(M-1)} & -L^{(M-1)} \end{pmatrix} * u_0$$

avec $I^{(M-1)}$ la matrice identité de dimension 2^{M-1}. Nous obtenons ainsi une formule récurrente réduisant notre indice M. Ceci revient au principe *divide and conquer*. En effet, on réduit le problème à des produits matrice vecteur simples. Nous remarquons que la matrice composé de $I^{(M-1)}$ et de $L^{(M-1)}$ est creuse, car ces deux matrices sont diagonales. On peut ainsi continuer l'écriture du produit matriciel sous la forme :

$$\begin{pmatrix} \begin{pmatrix} T^{(M-2)} & 0 \\ 0 & T^{(M-2)} \end{pmatrix} & 0 \\ 0 & \begin{pmatrix} T^{(M-2)} & 0 \\ 0 & T^{(M-2)} \end{pmatrix} \end{pmatrix} * \begin{pmatrix} \begin{pmatrix} I^{(M-2)} & I^{(M-2)} \\ L^{(M-2)} & -L^{(M-2)} \end{pmatrix} & 0 \\ 0 & \begin{pmatrix} I^{(M-2)} & I^{(M-2)} \\ L^{(M-2)} & -L^{(M-2)} \end{pmatrix} \end{pmatrix} * \begin{pmatrix} I^{(M-1)} & I^{(M-1)} \\ L^{(M-1)} & -L^{(M-1)} \end{pmatrix}$$

On peut ainsi définir une suite u_n avec :

$$u_1 = \begin{pmatrix} I^{(M-1)} & I^{(M-1)} \\ L^{(M-1)} & -L^{(M-1)} \end{pmatrix} * u_0$$

En faisant le produit, on obtient :

$$u_1(j) = u_0(j) + u_0(j + 2^{M-1})$$

pour $j \in [0, 2^{M-1}[$

et

$$u_1(j) = \exp^{\frac{2I\pi * j}{2^M}} * (u_0(j) - u_0(j + 2^{M-1}))$$

pour $j \in [2^{M-1}, 2^M[$

On peut faire de même pour le k^{eme} terme de la suite :

$$u_k = \begin{pmatrix} \begin{pmatrix} I^{(M-k)} & I^{(M-k)} \\ L^{(M-k)} & -L^(M-k) \end{pmatrix} \ddots & & 0 \\ & & \\ 0 & & \ddots \begin{pmatrix} I^{(M-k)} & I^{(M-k)} \\ L^{(M-k)} & -L^{(M-k)} \end{pmatrix} \end{pmatrix} * u_{k-1}$$

On a donc :

$$u_k(q * 2^{M-k+1} + j) = u_{k-1}(q * 2^{M-k+1} + j) + u_{k-1}(q * 2^{M-k+1} + j + 2^{M-k})$$

pour $q \in [0, 2^{k-1} - 1]$ et $j \in [0, 2^{M-k} - 1]$

et

$$u_k(q * 2^{M-k+1} + j + 2^{M-k}) = \exp^{\frac{2I\pi * j}{2^{M-k+1}}} * (u_{k-1}(q * 2^{M-k+1} + j) - u_{k-1}(q * 2^{M-k+1} + j + 2^{M-k}))$$

pour $q \in [0, 2^{k-1} - 1]$ et $j \in [0, 2^{M-k} - 1]$

L'indice q a pour rôle de se déplacer d'un bloc $\begin{pmatrix} I^{(M-k)} & I^{(M-k)} \\ L^{(M-k)} & -L^{(M-k)} \end{pmatrix}$ au suivant. Les 2^{k-1} blocs diagonaux de taille 2^{M-k+1} sont composés de matrices diagonales. Pour obtenir le produit $T^{(M)} * u_0$, nous calculons les termes de la suite jusqu'à u_M. Nous avons :

$$u_M = T^{(M)} * u_0 = \begin{pmatrix} \begin{pmatrix} I^{(0)} & I^{(0)} \\ L^{(0)} & -L^{(0)} \end{pmatrix} \ddots & & 0 \\ & & \\ 0 & & \ddots \begin{pmatrix} I^{(0)} & I^{(0)} \\ L^{(0)} & -L^{(0)} \end{pmatrix} \end{pmatrix} * u_{M-1} = \begin{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \ddots & & 0 \\ & & \\ 0 & & \ddots \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{pmatrix} * u_{M-1}$$

On remarque que $T^{(1)} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

Donc pour calculer la transformée de Fourier rapide d'un vecteur, nous avons juste à utilisé les **deux formules de récurrence énoncés précédemment**. L'immense avantage est le coût calcul (en nombre d'opérations élémentaires en arithmétique complexe) de cet algorithme qui est en $N \log_2 N$ alors que la méthode directe est en N^2 ! Il est intéressant de noter que la TFD (transformée de Fourier discrète) est très utilisé en informatique et dans de nombreux domaines (traitement d'image, calcul rapide...).

4 Exemple M=2

Nous allons décrire l'algorithme sur un exemple simple pour trouver la transformée de Fourier de $v = (1, 1, 1, 1)$. Nous allons le faire en utilisant les formules de récurrence :

$$u_1(0) = u_0(0) + u_0(2) = 1 + 1 = 2$$

$$u_1(1) = u_0(1) + u_0(2 + 1) = 1 + 1 = 2$$

$$u_1(2) = \exp^{\frac{2I\pi*0}{4}} *(u_0(0) - u_0(0+2)) = 1*(1-1) = 0$$

$$u_1(3) = \exp^{\frac{2I\pi*1}{4}} *(u_0(1) - u_0(1+2)) = 1*(1-1) = 0$$

Ensuite, on calcule u_2, le dernier terme de la suite :

$$u_2(0) = u_1(0) + u_1(1) = 2+2 = 4$$

$$u_2(2) = u_1(2) + u_1(3) = 0+0 = 0$$

$$u_2(1) = \exp^{\frac{2I\pi*0}{2}} *(u_1(0) - u_1(0+1)) = 1*(2-2) = 0$$

$$u_2(3) = \exp^{\frac{2I\pi*0}{2}} *(u_1(2) - u_1(2+1)) = 1*(0-0) = 0$$

On a le résultat $T^{(2)}v = (4,0,0,0)^t$. On peut le vérifier en calculant directement le produit Sg :

$$S = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

Le produit $Sv = (4,0,0,0)^t$ est le résultat de la TFD appliquée à v alors que le calcul précédent n'est pas exactement le résultat de la TFD appliquée à v, mais $T^{(2)}v = PSv$. En prenant $PT^{(2)}v = Sv$ (P matrice de permutation) c'est-à-dire en appliquant la transformation mirroir ($r_2(0) = 0, r_2(1) = 2, r_2(2) = 1, r_2(3) = 3$) on permute les lignes 1 et 2 (la numérotation commence à 0) du vecteur $T^{(2)}v$ et on obtient le résultat de la TFD appliquée à v.

5 Remarques

Cette algorithme a été décrit pour des vecteurs de taille $N = 2^M$, mais il est possible de faire de même avec des vecteurs de taille différentes (par exemple $N = 3^M$). Le plus simple est en général d'ajouter des coordonnées nulles au vecteur afin d'obtenir une taille égale à 2^M.

Cet algorithme a été découvert par deux mathématiciens dans les années 1960 : Cooley et Tuckey qui ont donné leurs noms à cet algorithme.

6 Application

Coefficients de Fourier d'une fonction périodique de période T :

$$c_k(f) = \frac{1}{T}\int_0^T f(t)e^{\frac{-2i\pi kt}{T}}dt, \ \ -\infty < k < +\infty$$

On considère la série de Fourier $\sum\limits_{k=-\infty}^{+\infty} c_k(f)e^{\frac{2i\pi kt}{T}}$

On prend un pas $\underline{\Delta t = T/n}$ et on approxime l'intégrale par les sommes de Riemann :

$$c_k(f) \simeq \frac{1}{n}\sum_{p=0}^{n-1} f(p\Delta t)e^{\frac{-2i\pi pk}{n}}$$

On pose $x_k = f(k\Delta t) \quad -\infty < k < +\infty$

Cette suite (x_k) est périodique, de période n, de transformée discrète :

$$\widehat{x_k} = \sum_{p=0}^{n-1} x_p e^{\frac{-2i\pi kp}{n}}, \ \ k = 0, 1, \cdots, n-1$$

On prend la suite $(\widehat{x_k})$ périodique, de période n.

Pour que l'approximation de l'intégrale soit correcte on prend $\underline{|k| < n/2}$

On a :

$$c_k(f) \simeq \frac{1}{n}\widehat{x_k}, \ \ c_{-k}(f) \simeq \frac{1}{n}\widehat{x_{n-k}}, \ \ 0 \leq k < n/2$$

Si f est une fonction numérique réelle de période T, on a :

$$\sum_{k=-\infty}^{+\infty} c_k(f)e^{\frac{2i\pi kt}{T}} = c_0(f) + \sum_{k\geq 1} c_k(f)e^{\frac{2i\pi kt}{T}} + c_{-k}(f)e^{\frac{-2i\pi kt}{T}}$$

$$= c_0(f) + \sum_{k\geq 1}(c_k(f) + c_{-k}(f))cos\frac{2\pi kt}{T} + i(c_k(f) - c_{-k}(f))sin\frac{2\pi kt}{T}$$

$$= c_0(f) + \sum_{k\geq 1} a_k(f)cos\frac{2\pi kt}{T} + b_k(f)sin\frac{2\pi kt}{T}, \ avec \ c_0(f) = \frac{1}{T}\int_0^T f(t)dt$$

$$a_k(f) = c_k(f) + c_{-k}(f) = \frac{1}{T}\int_0^T f(t)(e^{\frac{-2i\pi kt}{T}} + e^{\frac{2i\pi kt}{T}})dt = \frac{2}{T}\int_0^T f(t)cos\frac{2\pi kt}{T}dt$$

$$b_k(f) = i(c_k(f) - c_{-k}(f)) = \frac{i}{T}\int_0^T f(t)(e^{\frac{-2i\pi kt}{T}} - e^{\frac{2i\pi kt}{T}})dt = \frac{2}{T}\int_0^T f(t)sin\frac{2\pi kt}{T}dt$$

On prend $a_0(f) = 2c_0(f)$ et $b_0(f) = 0$ et on obtient la série trigonométrique :

$$\frac{a_0(f)}{2} + \sum_{k\geq 1} a_k(f)cos\frac{2\pi kt}{T} + b_k(f)sin\frac{2\pi kt}{T}$$