La transformée de Fourier Discrète

La seule transformée de Fourier calculable par un ordinateur

Yohann Tendero*

La transformée de Fourier discrète est un moyen de calculer les coefficients de Fourier d'une fonction a-périodique u à partir de ses N échantillons $u(\frac{ka}{N})$, $N = 0, \dots, N-1$ Cela n'est possible que si la fonction présente un nombre de fréquences inférieur ou égal à N (en un sens qui sera précisé dans la suite).

Pour coller à la pratique numérique, nous supposerons toujours dans cette section que N est pair. En général, N est une puissance de 2 et est donc pair. Néanmoins, tous les résultats énoncés s'adaptent sans difficulté au cas N impair.

Soit $u(x)$ une fonction réelle ou complexe de période a, et N un entier. On cherche un polynôme trigonométrique de la forme

$$P(x) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_k e^{\frac{2i\pi kx}{a}} \tag{1}$$

qui soit égal à u aux points $\frac{ka}{N}$, $N = 0, \dots, N-1$. On dira dans la suite qu'une fonction P telle que (1) a pour degré $\frac{N}{2}$.

Pourquoi choisir un polynôme trigonométrique ? La raison est physique : tous les dispositifs d'acquisition de signaux (sons) ou images ont une bande passante, c'est-à-dire un intervalle de fréquences captées par le dispositif d'enregistrement; les autres fréquences sont perdues ou tellement atténuées qu'on les néglige: on suppose donc ici que la "bande passante" est contenue dans l'intervalle $[-\frac{N}{2}, \frac{N}{2}-1]$. On reconnaît au passage ici une condition très similaire au théorème de Shannon. Il n'y a par contre aucune raison de supposer que le signal ou l'image soit périodique. Cette hypothèse est donc imposée à la donnée de manière abusive, et provoque une distorsion près des bords de l'image que l'on peut voir et évaluer : c'est le phénomène de Gibbs. Vous pouvez le voir p. ex. slide 24 du cours 2 sur le bord droit particulièrement,vous le verrez à nouveau dans vos TPs et nous verrons un moyen de l'atténuer. Si la fonction u dont on possède les N échantillons n'avait pas une bande de fréquence contenue dans $[\frac{-N}{2}, \frac{N}{2}-1]$, son interpolation par un polynôme trigonométrique de degré $\frac{N}{2}$ provoque une autre distorsion, très grave: l'aliasing que nous avons observé, décrit qualitativement et que nous analyserons au cours de la séance prochaine.

Que calcule la TFD ? On commence par rappeler la définition de la TFD:

Definition On pose $u_k = u(\frac{ka}{N})$ et, pour $n = 0, \dots, N-1$,

$$\hat{u}_n = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} u_k e^{\frac{-2i\pi kn}{N}}. \tag{2}$$

Les N coefficients $\hat{u}_n$ sont appelés transformée de Fourier discrète des échantillons de u.

On a

Proposition 0.1. *Les coefficients $\hat{u}_n$ définis par* (2) *sont les uniques coefficients tels que le polynôme trigonométrique P* (1) *vérifie $P(\frac{ka}{N}) = u_k$ pour tout $k = 0, \dots, N-1$.*

Autrement dit, lorsqu'on dispose d'une suite d'échantillons, ces échantillons peuvent être vus comme la donnée d'un polynôme trigonométrique. Cela signifie aussi:

*yohann.tendero@telecom-paristech.fr

Proposition 0.2. *Si u est un polynôme trigonométrique $u(x) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_k e^{\frac{2i\pi kx}{a}}$, les coefficients $\hat{u}_k$ sont obtenus par* (2)*. Ces coefficients sont les coefficients de la série de Fourier de u.*

Autrement dit La proposition 0.2 explique ce qu'il se passe lorsqu'on effectue des calculs en utilisant

$$\begin{tikzcd}
u(x) \arrow[rr, "Fourier\ series"] \arrow[dr, "Sampling"'] & & \hat{u}_n \\
& u_k \arrow[ur, "Discrete\ Fourier\ transform"'] &
\end{tikzcd}$$

Figure 1: La TFD calcule les coefficients de Fourier de u si u est un polynôme trigonométrique. On se souviendra que si on a N échantillons il faut que u soit de degré $\frac{N}{2}$ afin d'éviter l'aliasing.

la TFD: c'est en fait un polynôme trigonométrique que l'on manipule (que l'on convole, etc.). En fait, même si le paysage continu observé n'est pas un polynome trigonométrique on a

Proposition 0.3. *Soit u une fonction continue et a-périodique alors les $\hat{u}_k$ sont des approximations des $c_k(u)$ pour $k = -\frac{N}{2}, \ldots, \frac{N}{2}$.*

La proposition 0.3 est en principe toujours applicable: le signal observé passe à travers un filtre passe-bas avant d'être échantillonné (voir Théorème de Shannon-Whittaker) qui ce qui rend continu. On conclue ce document en rappelant que la TFD se calcule rapidement à l'aide d'un algorithme de FFT dont nous expliquons brièvement le fonctionnement.

L'algorithme de la FFT Comme nous l'avons vu plus haut, le calcul des coefficients de Fourier $\hat{u}_k$ demande l'évaluation d'un certain polynôme aux racines N-ièmes de l'unité. L'évaluation classique (ex. méthode de Hörner) d'un polynôme quelconque de degré $N-1$ en un point prend $\mathcal{O}(N)$ opérations, ce qui fait donc $\mathcal{O}(N^2)$ opérations pour les N racines de l'unité. La FFT permet de résoudre cette question en $\mathcal{O}(N \log N)$.
Pour ceux qui ne seraient pas convaincus par l'utilité de ce gain la différence est de $2.8h$ pour la méthode directe contre $60ms$ pour la FFT. (J'ai tiré ces chiffres de Wikipedia et vous pouvez les vérifier `https://fr.wikipedia.org/wiki/Analyse_de_la_complexit%C3%A9_des_algorithmes`, la taille d'une image on est plutot dans la colonne de droite du tableau.)

Soit $C(N)$ le cout (additions et multiplications) demandé pour l'évaluation d'un polynome trigonométrique de degré $N-1$ aux racines N-ièmes de l'unité. On se place dans le cas $N = 2^n$ (c'est presque toujours le cas en pratique, sinon on peut toujour prendre la puissance de 2 suivante). Soit

$$P(X) = \sum_{k=0}^{N-1} a_k X^k.$$

On sépare les termes d'odre pairs et impairs

$$Q(X) = \sum_{k=0}^{\frac{N}{2}-1} a_{2k} X^{2k},$$

$$R(X) = \sum_{k=0}^{\frac{N}{2}-1} a_{2k+1} X^{2k+1}.$$

D'une part on a

$$P(e^{\frac{2i\pi k}{N}}) = Q\left((e^{\frac{2i\pi k}{N}})^2\right) + e^{\frac{2i\pi k}{N}} R\left((e^{\frac{2i\pi k}{N}})^2\right). \tag{3}$$

D'autre part, si N est pair alors les $\left(e^{\frac{2i\pi k}{N}}\right)^2$ sont exactement les racines d'ordre $\frac{N}{2}$ de l'unité. Ce problème, plus petit, se calcule en $C(\frac{N}{2})$. On obtient donc

$$C(N) = 2C\left(\frac{N}{2}\right) + 2N,$$

d'où on tire facilement que

$$C(N) = \mathcal{O}(N \log N).$$

On fera attention à ne pas surinterpréter les complexités *asymptotiques* "$\mathcal{O}$" qui négligent tous les termes d'ordres inférieur comme on ne fait pour évaluer le comportement d'une fonction à $\pm\infty$. Les termes constants peuvent être gros. Cela signifie, qu'*a priori* un algorithme en $\mathcal{O}(N \log N)$ ne "battra" réellement un algorithme en $\mathcal{O}(N^2)$ que pour N suffisamment grand.