MT26 TP
séances numéro 5 et 6

UTBM-année 2000/2001
Printemps 2001

TRANSFORMÉES DE FOURIER DISCRÈTES ET SÉRIES DE FOURIER

1. Rappels et Introduction

On rappelle que si f est une fonction de $\mathbb{R}$ dans $\mathbb{R}$, 2π-périodique, alors, sous certaines conditions de régularité, on a

$$\forall t \in [0, 2\pi[, \quad f(t) = \sum_{k=-\infty}^{+\infty} c_k e^{ikt}, \tag{1}$$

où les coefficients de Fourier de f sont définis par

$$\forall k \in \mathbb{Z}, \quad c_k = \frac{1}{2\pi}\int_0^{2\pi} f(t)e^{-ikt}dt. \tag{2}$$

La relation (1) permet de calculer les valeurs ponctuelles de f en fonction des coefficients de Fourier c_k de f. Réciproquement, la relation (2) permet de calculer les coefficients de Fourier de f en fonction des valeurs ponctuelles de f.

Pour tout ce TP, on supposera que f possède un nombre fini de coefficients de Fourier non nul et que f admet le développement en série de Fourier suivant :

$$\forall t \in [0, 2\pi[, \quad f(t) = \sum_{k=-n}^{n} c_k e^{ikt}, \tag{3}$$

où n est un entier naturel. En notation réelle, (4) est équivalent à

$$\forall t \in [0, 2\pi[, \quad f(t) = a_0 + \sum_{k=1}^{n} a_k \cos(kt) + b_k \sin(kt). \tag{4}$$

Dans un premier temps (section 3), on se donnera $N = 2n+1$ valeurs réelles $(\alpha_l)_{0\leq l\leq N-1}$ et on montrera qu'il existe une unique fonction f du type (3) ou (4) telle que

$$\forall l \in \{0, ..., N-1\}, \quad f\left(\frac{2\pi l}{N}\right) = \alpha_l. \tag{5}$$

Ce problème correspond à celui de l'interpolation : par un nuage de points, faire passer la fonction f (cf. figure 1).

Comme dans le cas général où f est définie par (1), nous montrerons dans la section 3 que la données des valeurs ponctuelles $(\alpha_l)_{0\leq l\leq N-1}$ suffit à définir les coefficients de Fourier $(c_k)_{-n\leq k\leq n}$

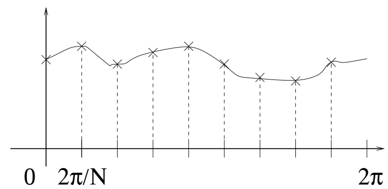

Figure 1. Le problème d'interpolation de Fourier

de f. Réciproquement, nous montrerons que les coefficients de Fourier $(c_k)_{-n\leq k\leq n}$ de f définissent totalement les valeurs ponctuelles de f. Pour ce, nous utiliserons la notion de transformation discrète de Fourier, rappelée dans la section 2.

Dans la section 4, nous verrons comment l'analyse spectrale d'un signal (c'est à dire la recherche des coefficients c_k de la fonction f qui interpole le nuage de points) permet de connaître l'importance des fréquences qui constituent ce signal ; les fréquences graves (correspondant aux coefficients c_k pour les petites valeurs de k) portent la partie régulière du signal et les fréquences aiguës (correspondant aux coefficients c_k pour les grandes valeurs de k) portent la partie oscillante du signal (voir figure 2). Cette

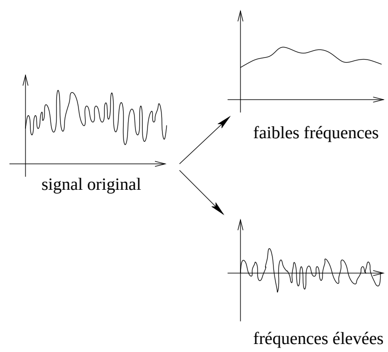

Figure 2. Séparation des fréquences d'un signal

séparation des fréquences permettra de filtrer un signal et d'atténuer par exemples les oscillations correspondant à des parasites ou un bruit (cf. section 5).

2. Transformée de Fourier discrète

Soit N appartenant à $\mathbb{N}^*$. La transformée de Fourier discrète est l'application de $\mathbb{C}^N$ dans $\mathbb{C}^N$ qui, aux N nombres complexes $(x_k)_{0\leq k\leq N-1}$, associe les N nombres complexes $(y_l)_{0\leq l\leq N-1}$ définis par

$$\forall l \in \{0, ..., N-1\}, \quad y_l = \sum_{k=0}^{N-1} x_k e^{-\frac{2ikl\pi}{N}}. \tag{6}$$

On notera

$$Y = tfd_N(X). \tag{7}$$

Question 1

Montrer que la formule (6) *s'inverse de la façon suivante*

$$\forall k \in \{0, ..., N-1\}, \quad x_k = \frac{1}{N} \sum_{l=0}^{N-1} y_l e^{\frac{2ikl\pi}{N}}. \tag{8}$$

L'application de $\mathbb{C}^N$ dans $\mathbb{C}^N$ qui, aux N nombres complexes $(y_l)_{0 \leq l \leq N-1}$, associe les N nombres complexes $(x_k)_{0 \leq k \leq N-1}$ définis par (8) s'appelle la transformée de Fourier discrète inverse. On notera

$$X = tfdinv_N(Y). \tag{9}$$

Pour calculer la transformée de Fourier discrète directe ou inverse d'un vecteur de $\mathbb{C}^N$, on peut utiliser les expressions (6) ou (8) ; cependant, on peut montrer que ces formules sont très lentes d'emploi en pratique. Pour utiliser chacune d'elle, il faut exécuter N^2 additions et multiplications. Depuis 1964, on utilise l'algorithme rapide mis au point par Cooley et Tuckey ; celui-ci réduit notablement le temps de calcul, permet de faire du traitement de signal en temps réel et est implanté dans les chaînes hi-fi, les ordinateurs ... Les fonctions *fft* et *ifft* de matlab (comme fast Fourier transform et inverse fast Fourier transform) réalisent ces transformées de Fourier rapides (voir aussi partie facultative). Pour toute la suite du TP, on utilisera systématiquement ces deux fonctions.

Question 2

Testez ces deux fonctions sur des vecteurs à quelques éléments et vérifier qu'elles sont inverses l'une de l'autre en calculant, pour X vecteur complexe fft(ifft(X)) et ifft(fft(X)).

3. Interpolation d'un nuage de points

Soient $n \in \mathbb{N}$, $N = 2n+1$ et les N valeurs réelles $(\alpha_l)_{0 \leq l \leq N-1}$. On cherche f de $\mathbb{R}$ dans $\mathbb{R}$, $2\pi-$périodique, définie par (3) ou (4), qui interpole ce nuage de point, c'est-à-dire qu'elle vérifie (5).

3.1. Passage des valeurs ponctuelles aux coefficients de Fourier

Question 3

Montrer que (3) *et* (5) *impliquent*

$$\forall l \in \{0, ..., N-1\}, \quad \alpha_l e^{-\frac{2iln\pi}{N}} = \sum_{k=0}^{N-1} \overline{c_{k-n}} e^{-\frac{2ikl\pi}{N}}. \tag{10}$$

Ainsi, par transformée de Fourier inverse, on peut exprimer les coefficients c_k à partir des valeurs ponctuelles α_k : l'équation (10) donne par transformée de Fourier inverse :

$$\forall k \in \{0, ..., N-1\}, \quad \overline{c_{k-n}} = \frac{1}{N} \sum_{l=0}^{N-1} \alpha_l e^{-\frac{2iln\pi}{N}} e^{\frac{2ikl\pi}{N}}. \tag{11}$$

Question 4

Montrer, à partir de (11), *que*

$$\forall k \in \{0, ..., n\}, \quad c_{-k} = \overline{c_k}. \tag{12}$$

et que

$$\forall k \in \{0, ..., n\}, \quad \overline{c_k} = \frac{1}{N} \sum_{l=0}^{N-1} \alpha_l e^{\frac{2ikl\pi}{N}}. \tag{13}$$

On constate donc que les nombres complexes c_k étant conjugués deux à deux, il suffit de connaître la moitié des coefficients de Fourier de f, c'est-à-dire les valeurs de c_k pour $k \in \{0, ..., n\}$. L'équation (13) nous montre que ces valeurs sont données par transformation de Fourier discrète inverse des réels α_l. L'algorithme de passage des réels α_l aux complexes c_k s'écrit (avec $N = 2n + 1$) :

$$(\alpha_l)_{0 \leq l \leq N-1} \longrightarrow (d_k)_{0 \leq k \leq N-1} = tfdinv_N \left((\alpha_l)_{0 \leq l \leq N-1}\right) \longrightarrow \begin{cases} c_0 = \operatorname{Re}(d_0), \\ \forall k \in \{1, ..., n\}, \quad c_k = \overline{d_k}. \end{cases} \tag{14}$$

La connaissance des $N = 2n + 1$ valeurs réelles α permet de déterminer les N valeurs réelles qui correspondent au nombre réel c_0 et aux n nombres complexes c_k.

Question 5

À partir de l'algorithme (14), *écrire une fonction matlab* ***coefficientfourierdiscret*** *qui, à un vecteur de N réels, contenant les valeurs* $(\alpha_l)_{0 \leq l \leq N-1}$, *renvoie un vecteur de* $n+1$ *complexes, correspondant aux valeurs* $(c_k)_{0 \leq k \leq n}$, *la valeur* c_0 *étant réelle pure, en utilisant la fonction* ifft *de matlab.*

3.2. Passage des coefficients de Fourier aux valeurs ponctuelles

Réciproquement, on cherche maintenant à passer des coefficients de Fourier aux valeurs ponctuelles. On admet que l'algorithme inverse de (14) est (avec $N = 2n + 1$) :

$$(c_k)_{0 \leq k \leq n} \text{ (où } c_0 \in \mathbb{R}) \longrightarrow \begin{cases} d_0 = c_0, \\ \forall k \in \{1, ..., n\}, \quad d_k = \overline{c_k}, \\ \forall k \in \{n+1, ..., 2n\}, \quad d_k = c_{2n-k+1}, \end{cases}$$
$$\longrightarrow (\alpha_l)_{0 \leq l \leq N-1} = tfd_N \left((d_k)_{0 \leq k \leq N-1}\right). \tag{15}$$

En matlab, pour programmer cet algorithme, on pourra utiliser la fonction *fliplr* qui retourne un vecteur.

Question 6

À partir de l'algorithme (15), *écrire une fonction matlab* ***valeurfourierdiscret*** *qui, à un un vecteur de* $n+1$ *complexes, correspondant aux valeurs* $(c_k)_{0 \leq k \leq n}$, *la valeur* c_0 *étant réelle pure, renvoie un vecteur de N réels, contenant les valeurs* $(\alpha_l)_{0 \leq l \leq N-1}$, *en utilisant la fonction* fft *de matlab.*

Ces deux fonctions programmées ne fonctionnent que pour des valeurs impaires de N. Dans la partie facultative est expliquée une méthode pour écrire ces deux algorithme si N est un entier pair.

4. Analyse spectrale

Nous allons maintenant utiliser la fonction *coefficientfourierdiscret* programmée dans la section 3 pour calculer les coefficients c_k d'une fonction passant par un nuage de points donné, c'est-à-dire faire l'analyse des différentes fréquences constituant cette fonction ; ces différentes valeurs constituent le spectre de la fonction (ou, par abus de langage, du nuage de points).

Par ailleurs, on rappelle qu'un script n'est pas une fonction : c'est une suite d'instructions, contenant éventuellement des entrées-sorties de données (avec les fonctions *disp* et *input*) et ne commençant pas par la ligne *function* ..... On rappelle aussi les commandes *stem* et *plot* (voir l'aide de matlab) qui permettent de faire des graphiques.

Question 7

Écrire un petit script matlab qui trace les modules des différents coefficients c_k correspondant à un nuage de point donné sous la forme d'un vecteur de $N = 2n + 1$ réels.

Question 8

Utilisez ce script pour représenter graphiquement le spectre des nuages définis par les valeurs suivantes : pour $l \in \{0, ..., N-1\}$, $\alpha_l = g\left(\frac{2\pi l}{N}\right)$ où

$$g(x) = \cos(x), \quad 3 + \cos(2x), \quad \cos(x) + 3\sin(4x) + 4\cos(20x) \ \textit{pour } N \in \{101, 3001\}.$$

Si p est un entier naturel, la fonction $rand(1,p)$ de matlab renvoie une matrice de taille $(1,p)$ dont chacune des composantes est un nombre aléatoirement choisi dans l'intervalle $[0,1]$.

Question 9

Déterminer le spectre du nuage défini par

$$N \in \{101, 3001\}, \quad \alpha = 2rand(1,N) - 1.$$

Question 10

Déterminer le spectre du nuage défini par

$$N = 3001, \quad \alpha = \beta\big(2rand(1,N) - 1\big) + \gamma,$$

où β est un paramètre appartenant à $\{0,1; 0,5; 1; 2; 10\}$ et γ est un vecteur de taille N dont les composantes sont définies par

$$\forall l \in \{0, ..., N-1\}, \quad \gamma_l = g\left(\frac{2\pi l}{N}\right),$$
$$\textit{où } g(x) = \cos(x), \quad 3 + \cos(2x), \quad \cos(x) + 3\sin(4x) + 4\cos(20x).$$

Comparez ces spectres à ceux de la question 8. Qu'en pensez-vous ?

5. Filtrage d'un signal bruité

Dans la section 4, on a vu comment la séparation des petites fréquences et des grandes fréquences permettait de séparer les petites oscillations des grandes oscillations. Nous allons appliquer ce principe pour filtrer le signal défini par la fonction suivante :

$$g(x) = f(x) + \beta\delta(x),$$

où f est une fonction $2\pi-$périodique oscillant «lentement» (par exemple une somme de quelques sinusoïdales), β est un paramètre positif et δ est un bruit, c'est-à-dire, une suite de valeurs aléatoirement choisies dans $[-1, 1]$. Ce cas de figure peut se présenter quand f est un signal émis, auquel s'ajoute des parasites représentés par δ (avec une importance représentées par β) ; le récepteur désire, à partir du signal g, reconstituer le signal f. Pour cela, on détermine le spectre de la fonction g :

$$\forall t \in [0, 2\pi[, \quad g(t) = a_0 + \sum_{k=1}^{n} a_k \cos(kt) + b_k \sin(kt),$$

et on considère le signal filtré, pour lequel les fréquences aiguës, correspondant à $k \in \{n_c, n\}$ ont été annulées (où la fréquence de coupure n_c appartient à $\{0, ..., n\}$)

$$\forall t \in [0, 2\pi[, \quad \widetilde{g}(t) = a_0 + \sum_{k=1}^{n_c-1} a_k \cos(kt) + b_k \sin(kt). \tag{16}$$

De façon pratique, si on connaît un nuage de point $(\alpha_l)_{0\leq l\leq N-1}$ (avec $N = 2n + 1$), on détermine la fonction g telle que

$$\forall l \in \{0, ..., N-1\}, \quad g\left(\frac{2\pi l}{N}\right) = \alpha_l,$$

puis on détermine son spectre et on considère la fonction $\widetilde{g}$ définie par (16). Les valeurs filtrées correspondent aux valeurs $(\widetilde{\alpha}_l)_{0\leq l\leq N-1}$ telle que

$$\forall l \in \{0, ..., N-1\}, \quad \widetilde{\alpha}_l = \widetilde{g}\left(\frac{2\pi l}{N}\right). \tag{17}$$

Le filtre choisi est un filtre passe-bas, puisque l'on a annulé les fréquences les plus élevées.

Question 11

En utilisant les fonctions coefficientfourierdiscret *et* valeurfourierdiscret*, écrire une fonction matlab* ***filtrepassebas***(n_c, α) *qui, à* α *un vecteur de* $2n + 1$ *réels et à* $n_c \in \{0, ..., n\}$*, associe le vecteur* $\widetilde{\alpha}$ *de* $2n + 1$ *réels correspondant aux valeurs filtrées définies par* (17).

Question 12

Filtrer les différents nuages de la question 10, c'est-à-dire représenter le signal filtré. On essayera plusieurs fréquences de coupures.

Question 13

Filtrer les différents nuages définis par

$$N = 3001, \quad \alpha = \beta\big(2rand(1, N) - 1\big) + \gamma,$$

où l'amplitude β *du bruit appartient à* $\{0,1;0,5;1;2;10\}$ *et où* γ *est aussi un signal aléatoire oscillant «lentement». On pourra le définir en matlab par*

$$\gamma = \textit{interp1}(2 * pi * (0 : N_b - 1)/N_b, 2 * rand(1, N_b) - 1, 2 * pi * (0 : N - 1)/N, \textit{'spline'}),$$

où, par exemple, $N_b = 70$. *Que fait cette instruction ? On essayera plusieurs fréquences de coupures. On pourra prendre un nombre plus élevé de points d'échantillonnage (par exemple* $N = 8001$, $N_b = 100$, $\beta \in \{2;5;10\}$*) ; on pourra représenter sur le même graphe le signal pur et le signal filtré.*