Maxime CHUPIN — 2022

Introduction au TRAITEMENT DU SIGNAL

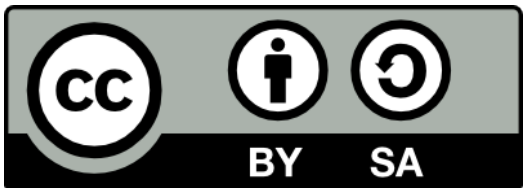


Table des matières

Introduction

Sommaire du chapitre

Ce cours concerne le traitement du signal. C'est un vaste sujet tant les signaux sont présents tout autour de nous, proviennent de sources très diverses. Le signal est le support de l'information. Le traitement du signal traite des opérations possibles sur ces signaux, et elles sont nombreuses : la transmission, le filtrage, avec notamment la réduction du *bruit* qui vient perturber le signal, la compression, le contrôle, etc. Si cette discipline trouve son origine dans les sciences de l'ingénieur avec l'électronique et l'automatique, elle fait aujourd'hui largement appel à un large spectre de domaines mathématiques. Dans ce cours, on essaiera de balayer un certain nombre de thématiques tout en essayant d'être suffisamment précis et rigoureux.

Énormément de choses qui nous entourent peuvent être considérées comme des signaux. On distingues deux grands types de signaux :

— **les signaux analogiques** qui sont des signaux physiques devenus la plupart du temps électriques à l'aide de capteurs. On peut penser au signal à la sortie d'un microphone, aux senseurs thermiques, optique, etc.

— **Les signaux numériques** eux sont issus d'ordinateurs et de terminaux.

Les signaux à traiter proviennent de sources très diverses : le son, les images, la vidéo, les échanges sur internet, le Wi-Fi, etc. La plupart d'entre eux, pour être modifiés ou transportés, sont soit des signaux numériques, soit transformés en signaux numériques (par numérisation).

Le traitement du signal peut avoir de multiples finalités comme la détection d'un signal, l'estimation de grandeurs à mesurer, le codage, et la compressions des signaux en vue du stockage et de la transmission, l'amélioration de sa qualité, la modification du signal (effets sonores par exemple).

0.1 Objectifs du cours

Le but principal de ce cours est de mieux comprendre les techniques, méthodes et outils *mathématiques* utilisés dans ce qu'on appelle habituellement le traitement numérique du signal.

Ce faisant, il doit permettre de mieux comprendre les différents problèmes scientifiques rencontrés. Plus généralement, l'objectif est aussi de familiariser le lecteur et la lectrice à un domaine qui occupe de nos jours une place importante dans la plupart des objets techniques de la vie courante.

Parmi les multiples finalités du traitement du signal, nous nous intéresseront principalement à son échantillonnage (transformation d'un signal analogique en signal numérique), à son codage, à son traitement et sa compressions.

Mis à part le cours d'intégration et l'analyse de Fourier, peu de prérequis sont nécessaires pour l'aborder. Les outils mathématiques abordés dans ce cours ont souvent une portée qui dépasse le simple cadre du traitement numérique du signal, si bien que le cours a aussi vocation à ouverture scientifique et culturelle.

0.2 Chaîne de transmission numérique

Prenons l'exemple de la transmission d'une conversation téléphonique, et voyons quelles sont les différentes procédures que subit le signal entre son émission et sa réception.

Conversion analogique-numérique

Lorsque l'on parle dans le microphone d'un téléphone, les vibrations acoustiques sont transformées en une tension oscillante par l'intermédiaire d'une membrane et d'un électro-aimant qui convertit ainsi le signal mécanique en signal électrique. C'est à ce moment qu'intervient la conversion analogique-numérique, c'est-à-dire le passage du continu au discret, de l'analogique au numérique, de $\mathbf{R} \to \mathbf{R}$ à $\mathbf{Z} \to \mathbf{Z}$ en quelque sorte. Le temps et les valeurs prises par la tension sont des valeurs continues, à valeur dans $\mathbf{R}$. Pour obtenir un signal complètement numérique, la tension est donc discrétisée en temps et en valeur. Concrètement, cela revient à prendre une série de photos instantanées des valeurs de la tension, puis de projeter les valeurs obtenues sur une grille fixe. Dans le vocabulaire du traitement numérique du signal, ces deux étapes sont respectivement appelées *échantillonnage* et *quantification*. L'ensemble de ces deux étapes constitue la conversion analogique-numérique, souvent notée C.A.N.

Codage et compression

Revenons à notre exemple de conversation téléphonique. Les normes suivies dans les télécommunications sont fixées au niveau international par l'IUT, l'union internationale de télécommunications, c'est-à-dire un important (en taille et en influence) groupement d'ingénieur·e·s et d'expert·e·s des télécommunications. Dans notre exemple, le signal analogique (la tension issue de la conversion signal mécanique en signal électrique) est échantillonné à 8 kHz et quantifié sur 8 bits. Cela signifie que l'on relève la valeur de la tension 8000 fois par seconde, et que la valeur obtenue est remplacée par une valeur choisie sur une grille en comportant $256 = 2^8$. Si l'on voulait transmettre telle quelle la liste de 0 et de 1 obtenue après ces deux étapes, il faudrait donc disposer d'un canal de transmission de débit 64 kBits par seconde.

Au début des années 50, on pensait que ce débit était très peu compressible et qu'il fallait concevoir des dispositifs de transmission supportant de tels débits. À l'heure actuelle, on transmet correctement une conversation téléphonique à l'aide d'un canal de débit de 4 kBits par seconde. Comment a-t-on fait pour réduire autant (plus de 10 fois !) le volume d'informations à transmettre ?

On a fait appel à des techniques de compression. L'ensemble de ces méthodes, appelées *codage source* développées à partir des années 50, peut être divisé en deux grandes catégories : la compression avec et la compression sans perte. Un représentant célèbre de la première catégorie est le format *mp3*, et un représentant célèbre de la seconde catégorie est le logiciel *7-zip*. Toujours dans l'exemple de la conversation téléphonique, ces deux techniques sont utilisées pour obtenir le débit de 4 kBits par seconde. Ce débit prend en compte un autre traitement qu'a subi le signal à transmettre, le *codage canal* qui, au contraire de la compression, ajoute des redondances pour permettre de corriger les erreurs éventuelles qui vont intervenir lors de la transmission. Signalons que c'est également ce type de techniques qui permet de lire correctement des *CD* ou des *DVD* rayés.

Modulation et transmission dans le canal

Le signal — codé — peut alors être envoyé dans le canal de transmission qui peut être selon le téléphone utilisé, un câble en cuivre (cas courant du téléphone fixe), une fibre optique, l'atmosphère (cas du téléphone portable), l'espace (cas des communications par satellite)… Le but est alors d'adapter les symboles numériques, *i.e.* la séquence de 0 et de 1, au canal de transmission choisi. Concrètement, si l'on souhaite transmettre 1 bit en Δt seconde, il s'agit de construire un signal analogique qui garde une caractéristique constante pendant Δt seconde. Toute cette étape est généralement désignée sous le terme

de *modulation*. Dans le canal, le signal codé subit des altérations de nature très variées, que le codage canal va pouvoir corriger.

Réception et décodage

À la réception en sortie de canal, interviennent séquentiellement les opérations inverses de celles présentées ci-dessus. On effectue donc une démodulation, un décodage canal, un décodage source et finalement une conversion numérique-analogique.

L'ensemble de cette chaîne de transmission est reproduite dans la figure 1.

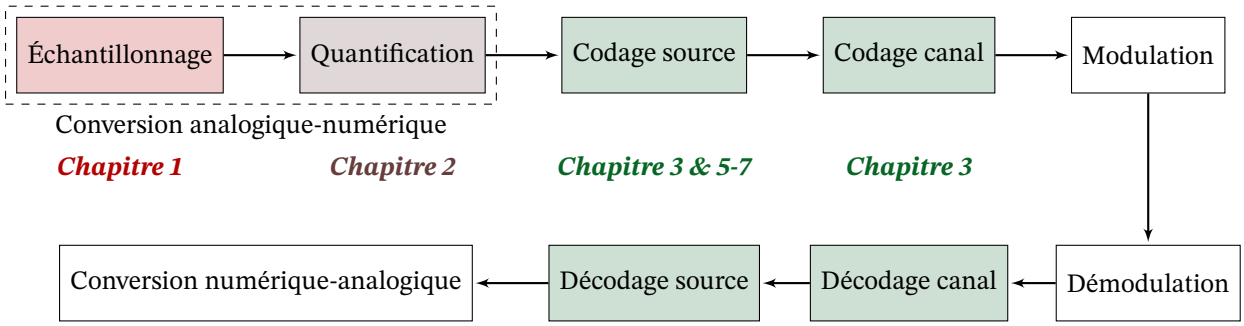


FIGURE 1 – La chaîne de transmission du signal et chapitres correspondants

0.3 LES SONS ET LES IMAGES

Les signaux sonores et les images sont de bonnes illustrations pour les méthodes exposées dans ce cours.

0.3.1 Le son

Un son est un ébranlement élastique de l'air, d'un fluide ou d'un solide qui se manifeste par des variations de pression autour de la pression moyenne du milieu. Lorsque le milieu est homogène, l'onde sonore se propage à vitesse constante c appelée célérité, célérité qui va dépendre du milieu.

Ces variations de pression sont modélisées par une fonction, et cette fonction peut-être décomposée, sous certaines hypothèses, en somme de cosinus et de sinus (série de Fourier). Pour les sons simples, il suffit d'ailleurs de peu de fonctions sinus et cosinus pour décrire correctement le son.

En reliant un microphone à un oscilloscope, on peut observer l'évolution temporelle de la pression acoustique lorsqu'une note est jouée avec un instru-

ment de musique. Une note produite peut être décomposée en quatre phases successives :

— la phase d'attaques ;
— le déclin ;
— la phase de maintien ;
— la chute.

Durant la phase de maintien, le signal est quasi périodique dont la forme sur une période est propre à l'instrument. C'est la périodicité de cette partie qui constitue la fréquence principale et donc la note que l'on définit comme la *hauteur d'une note*. Le *timbre* d'une note est directement lié à la forme du signal pendant la phase de maintien.

0.3.2 Les images

Une onde. La lumière est une onde électromagnétique. Nos yeux détectent les ondes électromagnétiques, mais une toute petite part d'entre elles, celles dont la longueur d'onde est comprise entre 380 et 760 nm. Ce sont ces ondes qu'on appelle les *ondes lumineuses*. Les fréquences correspondantes sont très élevées, environ 1 million de milliards d'oscillations par seconde (c'est-à-dire 1000 milliards de fois les fréquences des sons). Là encore, comme il y a onde, l'analyse de Fourier est très importante.

Mathématiquement, une image est une fonction $f : (x, y) \in \mathbf{R}^2 \mapsto f(x, y)$ où $f(x, y)$ peut être un réel, appelé niveau de gris, pour une image monochromatique, ou bien un vecteur de $\mathbf{R}^3$ (ou $\mathbf{R}^4$) pour des images couleurs. L'espace sans doute le plus connu est l'espace *rouge-vert-bleu* (RVB), où chaque composante du vecteur $f(x, y) \in \mathbf{R}^3$ correspond à l'intensité de rouge, de vert ou de bleu de la couleur à produire en (x, y). Il y a d'autres codages de la couleur, par exemple dans $\mathbf{R}^4$ avec l'espace *cyan-magenta-jaune-noir* (CMJN).

Ces décompositions sont cependant assez éloignées de la représentation dîte *psychovisuelle* qui décrit une couleur selon les trois attributs :

La luminosité : (ou luminance) qui traduit le niveau énergétique de l'observation.

La teinte : qui indique une position dans la palette des couleurs visibles. Elle correspond à la longueur d'onde d'un rayonnement monochromatique provoquant une sensation dans le même ton coloré.

La saturation : qui traduit le degré de pureté de la teinte. En effet, le rayonnement observé peut être considéré comme un mélange de lumière blanche et de lumière de teinte pure. La saturation est le rapport entre la luminosité de teinte pure et de celle du rayonnement total.

0.4 Plan du cours

Le plan du cours suit en partie les différentes étapes de la chaîne de transmission qui vient d'être décrite. Les chapitres 1 et 2 portent sur les conditions d'échantillonnage et les techniques de quantification [1]. Pour établir le résultat principal du chapitre 1, nous introduirons les outils d'analyse de Fourier, outils mathématiques fondamentaux dans le traitement du signal. Des techniques de codage source sans perte de codage canal sont présentées au chapitre 3. Les chapitres suivants permettent d'introduire la compression avec perte. Celle-ci repose presque systématiquement sur la décomposition du signal. Dans le cadre de ce cours, cette décomposition s'effectue suivant les composantes de Fourier du signal à transmettre. On commence donc par introduire la transformation de Fourier discrète ainsi qu'un algorithme très efficace permettant son calcul, la *transformée de Fourier rapide*, encore appelée FFT au chapitre 4. On aborde ensuite la notion de filtre numérique, outil indispensable à la décomposition d'un signal numérique au chapitre 5. Enfin, dans le chapitre 6, nous aborderons le traitement dit « temps/fréquence » où nous introduirons la transformée de Fourier à fenêtre ainsi que la transformée en ondelettes.

1. Le chapitre 2 ne fait pas partie du contenu du cours, il est là, à titre d'information pour répondre à la curiosité des plus investi·e·s.

CHAPITRE 1

Formule de Shannon-Nyquist et échantillonnage

SOMMAIRE DU CHAPITRE

Dans ce chapitre, nous établissons un théorème fondamental du traitement du signal. En effet, comme discuté dans l'introduction, les signaux analogiques, pour leur traitement, sont discrétisés, et donc, il est nécessaire d'être capable, à

partir de relevés discrets d'un signal, de le reconstruire entièrement. Ceci se fait bien entendu au prix de quelques hypothèses de régularité sur le signal que nous allons détailler. Ce théorème est connu sous le nom du *théorème d'échantillonnage*, du *théorème de Shannon* ou bien encore du *théorème de Nyquist-Shannon*. [1]

La démonstration du théorème, ainsi que bon nombre de résultat de traitement du signal reposent sur la théorie de la Transformation de Fourier [2], théorie au cœur du traitement du signal. Ainsi ce chapitre débute par quelques rappels de résultats bien connus.

1.1 Transformées de Fourier des fonctions L^1

Tout d'abord, donnons la définition de la transformée de Fourier.

1.1.1 Définitions et notations

Introduisons tout d'abord quelques notations. On note :

$$L^1 \stackrel{\text{def}}{=} \left\{ f \,:\, \mathbf{R} \to \mathbf{C},\ \int_{\mathbf{R}} |f| < +\infty \right\}.$$

Pour ce qui nous concerne, nous désignerons par *signal stable* une fonction de L^1.

Étant donnée une partie A de $\mathbf{R}$, on note $\mathbf{1}_A$ sa fonction indicatrice, c'est-à-dire la fonction de $\mathbf{R}$ dans $\mathbf{R}$ qui vaut 1 sur la partie A et 0 ailleurs.

Une fonction est dite *à support borné* si elle est nulle en dehors d'une partie bornée de $\mathbf{R}$. Venons-en à la définition qui nous intéresse.

Définition 1.1 — (Transformée de Fourier dans L^1) Soit $s \in L^1(\mathbf{R})$. La transformée de Fourier de s, notée $\hat{s}$ est définie sur $\mathbf{R}$ par la formule :

$$\hat{s}(\nu) \stackrel{\text{def}}{=} \int_{\mathbf{R}} s(t) \mathrm{e}^{-2\mathrm{i}\pi\nu t} \mathrm{d}t.$$

1. (Wikipédia) À partir des années 1960, le théorème d'échantillonnage est souvent appelé théorème de Shannon, du nom de l'ingénieur qui en a publié la démonstration en posant les bases de la théorie de l'information chez *Bell Laboratories* en 1949. Quelques années plus tard, on joint à ce nom celui de Nyquist, de la même entreprise, qui avait ouvert la voie dès 1928.

2. Joseph Fourier (21 mars 1768, Auxerre - 16 mai 1830, Paris) est un mathématicien et physicien français, connu pour ses travaux sur la décomposition de fonctions périodiques en séries trigonométriques convergentes appelées séries de Fourier.

La transformée de Fourier se note aussi $\mathcal{F}(s) = \hat{s}$.

Remarque 1.1 :

Plusieurs conventions sont possibles pour la définition de la transformée de Fourier : sans constante dans l'exponentielle (2π), soit avec une constante $1/2\pi$ facteur de l'intégrale. Le seul changement dans les résultats se situe au niveau des constantes dans les formules.

Remarque 1.2 :

— Le cadre fonctionnel $L^1(\mathbf{R})$ est un cadre *naturel* pour la transformée de Fourier. En effet, avec ce cadre là, la transformée $\hat{s}$ est bien définie pour tout ν dans $\mathbf{R}$ puis que $(x \mapsto s(x)\mathrm{e}^{-2\mathrm{i}\pi\nu x}) \in L^1$. Nous verrons que dans ce cadre fonctionnel, la transformée de Fourier n'est pas toujours inversible.
— La transformée de Fourier peut être étendue aux fonctions L^2 par un passage à la limite, cadre dans lequel elle est un isomorphisme et une isométrie (théorème de Plancherel). On adoptera ce cadre fonctionnel au chapitre 6.
— La transformée de Fourier peut être considéré dans l'espace des *distributions tempérées* de Laurent Schwartz qui contient tous les L^p et dans lequel elle est encore inversible. Beaucoup de références de traitement du signal se placent dans ce cadre là. Nous ne l'aborderons presque pas ici.

Exercice 1.1 :

Montrer que pour $T > 0$, la transformée de Fourier d'une fonction fenêtre, encore appelée *signal rectangulaire*,définie par

$$\forall t \in \mathbf{R},\ \mathrm{rect}_T(t) \stackrel{\text{def}}{=} \mathbf{1}_{[-T/2,T/2]}(t)$$

est donnée par :

$$\widehat{\mathrm{rect}_T}(\nu) \stackrel{\text{def}}{=} T\,\mathrm{sinc}(\pi\nu T) = T\frac{\sin(\pi\nu T)}{\pi\nu T}.$$

Voir figure 1.1 pour un tracé.

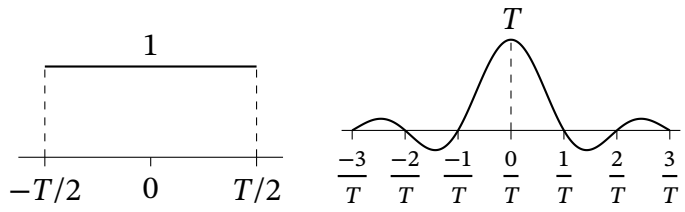


FIGURE 1.1 – Fonction fenêtre et sa transformée de Fourier

Exercice 1.2 :

Soit f une fonction de $L^1(\mathbf{R})$ paire. Montrez que :

$$\forall \nu \in \mathbf{R}, \quad \mathcal{F}(f)(\nu) = 2\int_0^{+\infty} f(t)\cos(2\pi\nu t)\mathrm{d}t.$$

Exercice 1.3 :

Soit $a > 0$ et

$$f : t \in \mathbf{R} \mapsto \mathrm{e}^{-a|t|}.$$

Montrez que

$$\forall \nu \in \mathbf{R}, \quad \mathcal{F}(f)(\nu) = \frac{2a}{a^2 + 4\pi^2\nu^2}.$$

On montre par le calcul la proposition suivante :

Proposition 1.1 : *Soit s un signal stable. Alors, on a :*

Transformations	f	$\widehat{f}$
Délais	$f : t \mapsto s(t - t_0)$	$\widehat{f} : \nu \mapsto \mathrm{e}^{-2\mathrm{i}\pi\nu t_0}\hat{s}(\nu)$
Modulation	$f : t \mapsto \mathrm{e}^{2\mathrm{i}\pi\nu_0 t}s(t)$	$\widehat{f} : \nu \mapsto \hat{s}(\nu - \nu_0)$
Op. linéaires	$f : t \mapsto s(at)$	$\widehat{f} : \nu \mapsto \frac{1}{\lvert a\rvert}\hat{s}\left(\frac{\nu}{a}\right)$
	$f : t \mapsto \lambda s(t)$	$\widehat{f} : \nu \mapsto \lambda\hat{s}(\nu)$
	$f : t \mapsto s^*(t)$	$\widehat{f} : \nu \mapsto \hat{s}(-\nu)^*$

Exercice 1.4 :

Démontrer les résultats de la proposition 1.1.

Exercice 1.5 :

Montrer que la transformée de Fourier est linéaire et continue dans $L^1(\mathbf{R})$.

1.1.2 Convolution

Une loi de composition est généralement associée à la transformée de Fourier, c'est le *produit de convolution* [3]. Nous verrons que cette loi de composition donne les clés pour comprendre et analyser les problèmes soulevés par la conversion d'un signal analogique en signal digital (CAN : conversion analogique-numérique) et aussi pour la conversion inverse, d'un signal digital en un signal analogique (CNA : convertion numérique-analogique).

DÉFINITION 1.2 — (Convolution) Soit a, b deux fonctions de L^1. On définit le produit de convolution ou *convolée* de a et b par la formule :

$$\forall t \in \mathbf{R}, \quad a \star b(t) = \int_{\mathbf{R}} a(t-u)b(u)\mathrm{d}u.$$

Cette définition a bien un sens car d'après le théorème de Tonelli (voir annexe A), on a :

$$\int_{\mathbf{R}}\int_{\mathbf{R}} |a(t-u)||b(u)|\mathrm{d}u\ \mathrm{d}t = \left(\int_{\mathbf{R}} |a|\right) \times \left(\int_{\mathbf{R}} |b|\right),$$

et donc :

$$\int_{\mathbf{R}} |a(t-u)||b(u)|\mathrm{d}u < +\infty \qquad p.p.,$$

par conséquent $t \mapsto \int_{\mathbf{R}} a(t-u)b(u)\mathrm{d}u$ est définie presque partout.

Attention, ici le vocabulaire diverge entre physicien·ne·s et mathématicien·ne·s. En mathématiques, on parle de *convolée* de deux fonctions, alors que les physicien·ne·s emploient le terme *convoluée*.

3. qui elle aussi peut être généralisée à des cadres plus larges que celui des fonctions de $L^1(\mathbf{R})$.

On vérifie aisément que cette loi de composition est commutative et associative. Le lien avec la transformée de Fourier est donné par le lemme suivant.

LEMME 1.1 Soit a, b deux fonctions de L^1. On a :

$$\widehat{a \star b} = \hat{a}\hat{b}.$$

Autrement dit, la transformée de Fourier change le produit de convolution en simple produit.

Preuve : La fonction $a \star b$ étant intégrable, d'après le théorème de Fubini nous avons :

$$\begin{aligned}\int_{\mathbf{R}} \left(\int_{\mathbf{R}} a(t-u)b(u)\mathrm{d}u \right) \mathrm{e}^{-2\mathrm{i}\pi\nu t}\mathrm{d}t &= \int_{\mathbf{R}} b(u) \left(\int_{\mathbf{R}} a(t-u)\mathrm{e}^{-2\mathrm{i}\pi\nu(t-u)}\mathrm{d}t \right) \mathrm{e}^{-2\mathrm{i}\pi u}\mathrm{d}u \\ &= \hat{a}\hat{b},\end{aligned}$$

ce qui achève la preuve. ■

Exercice 1.6 :

Soit a de classe $\mathcal{C}^1$, stable de dérivée a' stable, et soit b un signal stable. Montrer que

$$\forall t \in \mathbf{R}, \frac{\mathrm{d}}{\mathrm{d}t}(a \star b)(t) = a' \star b(t).$$

1.1.3 Le lemme de Riemann-Lebesgue

Dans les démonstrations que nous verrons dans la suite, nous ferons souvent appel à des passages à la limite sous le signe intégral. Nous aurons en particulier besoin du résultat suivant, généralement appelé Lemme de Riemann-Lebesgue.

LEMME 1.2 — (Lemme de Riemann-Lebesgue) La transformée de Fourier d'un signal s stable vérifie :

$$\lim_{|\nu|\to+\infty} |\hat{s}(\nu)| = 0.$$

Preuve : On procède en trois étapes.

1. Pour un signal rectangulaire, nous avons vu à la section 1.1.1 qu'il existe un $K > 0$ tel que :
$$|\hat{s}(\nu)| \leq \frac{K}{|\nu|}.$$
Le résultat est donc vrai dans ce cas.

2. Ce résultat s'étend aux combinaisons linéaires finies de signaux rectangulaires, c'est-à-dire aux fonctions étagées.
3. Soit maintenant un signal stable s et un réel ν. Par densité des fonctions étagées dans L^1, nous savons qu'il existe une suite de fonctions étagées $(s_n)_{n\in\mathbf{N}}$ vérifiant :
$$\lim_{n\to\infty}\int_{\mathbf{R}}|s_n(\nu)-s(\nu)\mathrm{d}\nu|=0.$$
avec, d'après le point précédant,
$$|\widehat{s_n}(\nu)|\leq\frac{K_n}{|\nu|}.$$
Par linéarité de l'intégral :
$$\forall\nu\in\mathbf{R},\int_{\mathbf{R}}s(t)\mathrm{e}^{-2\mathrm{i}\pi\nu t}\mathrm{d}t=\int_{\mathbf{R}}(s(t)-s_n(t))\,\mathrm{e}^{-2\mathrm{i}\pi\nu t}\mathrm{d}t+\int_{\mathbf{R}}s_n(t)\mathrm{e}^{-2\mathrm{i}\pi\nu t}\mathrm{d}t,$$
puis par inégalité triangulaire, nous obtenons :
$$\begin{aligned}|\hat{s}(\nu)|&\leq|\widehat{s_n}(\nu)|+\int_{\mathbf{R}}|s(t)-s_n(t)|\mathrm{d}t\\&\leq\frac{K_n}{|\nu|}+\int_{\mathbf{R}}|s(t)-s_n(t)|\mathrm{d}t,\end{aligned}$$
qui peut être rendu arbitrairement petit, sous réserve que $|\nu|$ soit suffisamment grand. ■

1.1.4 La formule d'inversion

Les résultats précédents nous indiquent que la transformée de Fourier d'un signal stable tend vers 0. On peut également montrer qu'elle est uniformément continue et bornée.

Par contre, elle ne constitue pas un signal stable. En effet, nous avons vu que la transformée de Fourier du signal rectangulaire rect_T pour $T\in\mathbf{R}$ est proportionnelle au sinus cardinal et que celui ci n'appartient pas à $L^1(\mathbf{R})$.

Exercice 1.7 :

Montrer que la fonction $t\mapsto\frac{\sin t}{t}$ n'appartient pas à $L^1(\mathbf{R})$.

Ainsi, la transformée de Fourier d'un signal stable n'est pas nécessairement un signal stable.

L'inversion de la transformée de Fourier ne peut donc se faire que sur un *sous espace vectoriel* de L^1.

THÉORÈME 1.1 — (Formule d'inversion de la transformée de Fourier) Soit s un signal stable, tel que sa transformée de Fourier $\hat{s}$ soit également stable. Alors, pour presque tout t :

$$s(t) = \int_{\mathbf{R}} \hat{s}(\nu) \mathrm{e}^{2\mathrm{i}\pi\nu t} \mathrm{d}\nu.$$

Preuve : (hors programme) On procède de nouveau en trois étapes.

1. Par un calcul simple, sans problème de convergence, on montre le résultat pour les fonctions $f_{\alpha,\beta}$, α et β deux réels, définies par :

$$\forall t \in \mathbf{R}, \quad f_{\alpha,\beta}(t) = \mathrm{e}^{-\alpha t^2 + \beta t}.$$

2. On considère ensuite la fonction h_σ définie par :

$$h_\sigma(t) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{t^2}{2\sigma^2}},$$

dont la transformée de Fourier est :

$$\widehat{h_\sigma}(\nu) = \mathrm{e}^{-2\pi^2\sigma^2\nu^2}.$$

Étant donné un signal stable s, la formule d'inversion est vraie pour $s \star h_\sigma$. En effet :

— on a successivement :

$$\begin{aligned} s \star h_\sigma(t) &= \int_{\mathbf{R}} s(u) h_\sigma(t-u) \mathrm{d}u \\ &= \int_{\mathbf{R}} s(u) \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(t-u)^2}{2\sigma^2}} du \\ &= \int_{\mathbf{R}} s(u) h_\sigma(u) f_{\frac{1}{\sigma\sqrt{2}}, \frac{u}{\sigma^2}}(t) \mathrm{d}u, \end{aligned}$$

— de plus :

$$\begin{aligned} \hat{s}(\nu)\widehat{h_\sigma}(\nu) &= \int_{\mathbf{R}} \int_{\mathbf{R}} s(u) h_\sigma(u) f_{\frac{1}{\sigma\sqrt{2}}, \frac{u}{\sigma^2}}(t) \mathrm{e}^{-2\pi\mathrm{i}\nu t} \mathrm{d}t \mathrm{d}u \\ &= \int_{\mathbf{R}} s(u) h_\sigma(u) \underbrace{\int_{\mathbf{R}} f_{\frac{1}{\sigma\sqrt{2}}, \frac{u}{\sigma^2}}(t) \mathrm{e}^{-2\pi\mathrm{i}\nu t} \mathrm{d}t}_{\widehat{f}_{\frac{1}{\sigma\sqrt{2}}, \frac{u}{\sigma^2}}(\nu)} \mathrm{d}u, \end{aligned}$$

— donc :

$$\begin{aligned}\int_{\mathbf{R}} \hat{s}(\nu)\widehat{h_\sigma}(\nu)\mathrm{e}^{2\mathrm{i}\pi\nu t}\mathrm{d}\nu &= \int_{\mathbf{R}}\int_{\mathbf{R}} s(u)h_\sigma(u)\widehat{f}_{\frac{1}{\sigma\sqrt{2}},\frac{u}{\sigma^2}}(\nu)\mathrm{e}^{2\mathrm{i}\pi\nu t}\mathrm{d}u\mathrm{d}\nu\\ &= \int_{\mathbf{R}} s(u)h_\sigma(u)f_{\frac{1}{\sigma\sqrt{2}},\frac{u}{\sigma^2}}(t)\mathrm{d}u\\ &= s\star h_\sigma(t),\end{aligned}$$

ce qui constitue la conclusion recherchée.

3. Enfin, nous avons :

$$\lim_{\sigma\to 0}\hat{h}_\sigma(\nu) = \lim_{\sigma\to 0}\mathrm{e}^{2\pi^2\sigma^2\nu^2} = 1.$$

Par convergence dominée, ceci donne :

$$\lim_{\sigma\to 0}\int_{\mathbf{R}}\hat{s}(\nu)\widehat{h_\sigma}(\nu)\mathrm{e}^{2\mathrm{i}\pi\nu t}\mathrm{d}\nu = \int_{\mathbf{R}}\hat{s}(\nu)\mathrm{e}^{2\mathrm{i}\pi\nu t}\mathrm{d}\nu.$$

Il reste donc à montrer :

$$\lim_{\sigma\to 0} s\star h_\sigma = s,$$

dans L^1. Pour ce faire, en utilisant le fait que $\int_{\mathbf{R}} h_\sigma(u)\mathrm{d}u = 1$, notons que :

$$\int_{\mathbf{R}}|s\star h_\sigma - s| = \int_{\mathbf{R}}\left|\int_{\mathbf{R}}\big(s(t-u)-s(t)\big)h_\sigma(u)\mathrm{d}u\right|\mathrm{d}t.$$

On pose alors $\phi(u) = \int_{\mathbf{R}}|s(t-u)-s(t)|\mathrm{d}t$. On a que $\phi\in L^1(\mathbf{R})$ (bornée par $2\|s\|_{L^1}$) et nous obtenons :

$$\int_{\mathbf{R}}|s\star h_\sigma - s| \le \int_{\mathbf{R}}\phi(u)h_\sigma(u)\mathrm{d}u = \int_{\mathbf{R}}\phi(\sigma u)h_1(u)\mathrm{d}u = I_\sigma.$$

Il reste à montrer que $\lim_{\sigma\to 0} I_\sigma = 0$.

— Si s est à support compact et continue, alors par uniforme-continuité, pour toute suite u_n tendant vers 0

$$\lim_{n\to+\infty}|s(t-u_n)-s(t)| = 0.$$

Donc, puisque ϕ est intégrable, on obtient par convergence dominée :

$$\lim_{\sigma\to 0} I_\sigma = 0.$$

— On utilise la densité des fonctions continue à support compact dans L^1 : ainsi pour s stable, il existe une suite de fonctions s_n à support compact qui converge vers s dans L^1. On obtient alors le résultat cherché par l'intermédiaire de cette approximation. ■

Exercice 1.8 :

En utilisant le résultat de l'exercice 1.3, et la transformée de Fourier inverse, montrer que la transformée de Fourier de la fonction, pour $b \in \mathbf{R}$, définie par

$$f : t \in \mathbf{R} \mapsto \frac{1}{b^2 + t^2}$$

est

$$\mathcal{F}(f) : \nu \mapsto \frac{\pi}{b} \mathrm{e}^{-2\pi b t}.$$

1.1.5 Dérivation et transformée de Fourier

THÉORÈME 1.2 — (Dérivation) Soit $n \in \mathbf{N}$.

1. Soit s un signal dans $\mathcal{C}^n$ tel que toutes ses dérivées k-ième pour $k \in \{1, \dots, n\}$ sont L^1, alors
$$\mathcal{F}(s^{(k)}) = \left(\nu \mapsto (2\mathrm{i}\pi\nu)^k \hat{s}(\nu)\right).$$
2. Soit $s \in L^1$ tel que $t \mapsto t^k s(t) \in L^1$ pour tout $k \in \{1, \dots, n\}$ où $n \in \mathbf{N}$, alors sa transformée de Fourier est dans $\mathcal{C}^n$ et pour tout k, on a
$$\mathcal{F}(t \mapsto (-2\mathrm{i}\pi t)^k s(t)) = \left(\nu \mapsto \hat{s}^{(k)}(\nu)\right).$$

Preuve : Montrons tout d'abord la deuxième assertion. Avec la définition de la transformée de Fourier, et grâce au théorème de différentiation sous l'intégrale où intervient l'hypothèse $t \mapsto t^k s(t) \in L^1$ pour dominer la dérivée, on peut dérivée k fois sous le signe intégral et obtenir :

$$\forall \nu \in \mathbf{R}, \quad \hat{s}^{(k)}(\nu) = \int_{\mathbf{R}} (-2\mathrm{i}\pi t)^k s(t) \mathrm{e}^{-2\mathrm{i}\pi\nu t} \mathrm{d}t.$$

La première assertion se prouve par récurrence. On prouve le résultat pour $n = 1$ puis on itère le procédé.

Tout d'abord, on a que $\lim_{|a|\to\infty} s(a) = 0$. En effet, soit $a > 0$, on a :

$$s(a) = s(0) + \int_0^a s'(t)\mathrm{d}t,$$

et puisque $s' \in L^1$, la limite existe et est finie. Cette limite est alors 0 puis que s est intégrable. Ensuite, on a

$$\int_{\mathbf{R}} \mathrm{e}^{-2\mathrm{i}\pi\nu t} s'(t)\mathrm{d}t = \lim_{|a|\to\infty} \int_{-a}^{+a} \mathrm{e}^{-2\mathrm{i}\pi\nu t} s'(t)\mathrm{d}t.$$

Une simple intégration par partie donne

$$\int_{-a}^{+a} \mathrm{e}^{-2\mathrm{i}\pi\nu t} s'(t)\mathrm{d}t = \left[\mathrm{e}^{-2\mathrm{i}\pi\nu t} s(t)\right]_{-a}^{+a} + \int_{-a}^{+a} (2\mathrm{i}\pi\nu)\mathrm{e}^{-2\mathrm{i}\pi\nu t} s(t)\mathrm{d}t.$$

Il suffit alors de faire tendre a vers $+\infty$ pour obtenir le résultat. ■

Exercice 1.9 :

Montrer que la transformée de Fourier d'une gaussienne est une gaussienne :

$$\mathcal{F}(f : t \mapsto \mathrm{e}^{-b^2t^2}) = \left(\nu \mapsto \frac{\sqrt{\pi}}{|b|}\mathrm{e}^{-\frac{\pi^2}{b^2}\nu^2}\right).$$

On pourra partir de l'équation différentielle que satisfait f, et d'en faire la transformée de Fourier. On rappellera que

$$\int_{\mathbf{R}} \mathrm{e}^{-b^2t^2}\mathrm{d}t = \frac{\sqrt{\pi}}{|b|}.$$

Exercice 1.10 :

Résoudre par transformée de Fourier l'équation de la chaleur suivante

$$\frac{\partial\theta}{\partial t} = \kappa\frac{\partial^2\theta}{\partial^2 x},$$

où

- pour tout $x \in \mathbf{R}$ et $t \in \mathbf{R}$, $\theta(x,t)$ est la température au temps t au point x d'une barre unidimensionnelle de conductance $\kappa \in \mathbf{R}$;
- à $t = 0$, on a pour tout $x \in \mathbf{R}$, $\theta(x,0) = f(x)$ avec $f \in L^1(\mathbf{R})$ et $\widehat{f} \in L^1(\mathbf{R})$.

On supposera l'application partielle $t \mapsto \theta(x,t)$ suffisamment régulière et intégrable, et on supposera l'application partielle $x \mapsto \theta(x,t) \in L^1(\mathbf{R})$.

1.1.6 Transformée de Fourier à deux dimensions

La transformée de Fourier dans $\mathbf{R}^n$ est une extension simple de la transformée de Fourier à une dimension dont nous venons de rappeler les résultats qui nous seront utiles dans ce cours.

Nous allons ici nous restreindre au cas bidimensionnel utile au traitement des images par exemple.

DÉFINITION 1.3 — (Transformée de Fourier dans $L^1(\mathbf{R}^2)$) Soit $s \in L^1(\mathbf{R}^2)$. On notera $x = (x_1, x_2) \in \mathbf{R}^2$ et $\nu = (\nu_1, \nu_2) \in \mathbf{R}^2$. La transformée de Fourier de s, notée $\hat{s}$ est définie sur $\mathbf{R}^2$ par la formule :

$$\forall \nu \in \mathbf{R}^2, \quad \hat{s}(\nu_1, \nu_2) \stackrel{\text{def}}{=} \int_{\mathbf{R}} \int_{\mathbf{R}} s(x_1, x_2) \mathrm{e}^{-2\mathrm{i}\pi(x_1\nu_1 + x_2\nu_2)} \mathrm{d}x_1 \, \mathrm{d}x_2.$$

Remarque 1.3 :

On peut remarquer que $x_1\nu_1 + x_2\nu_2 = x \cdot \nu$, et qu'avec la notation avec le produit scalaire, il est assez immédiat d'étendre la transformée de Fourier à $\mathbf{R}^n$.

On a les résultats analogues aux précédents que nous synthétisons dans la proposition suivante.

Proposition 1.2 : — *Si s et $\hat{s}$ sont deux fonctions de $L^1(\mathbf{R}^2)$ alors on a la formule d'inversion pour tout $x \in R^2$:*

$$s(x) = \iint_{\mathbf{R}^2} \hat{s}(\nu) \mathrm{e}^{\mathrm{i}2\pi(\nu \cdot x)} \mathrm{d}\nu.$$

— *Si s et h sont deux fonctions de $L^1(\mathbf{R}^2)$, alors la transformée de Fourier de la convolution g définie pour tout $x \in \mathbf{R}^2$ par*

$$g(x) = s \star h(x) = \iint_{\mathbf{R}^2} s(u) h(x - u) \mathrm{d}u,$$

est, pour tout $\nu \in \mathbf{R}^2$:

$$\hat{g}(\nu) = \hat{s}(\nu) \hat{h}(\nu).$$

1.2 Séries de Fourier des signaux L^1_{loc}

Un signal périodique n'est ni stable ($L^1(\mathbf{R})$), ni d'énergie finie ($L^2(\mathbf{R})$). Pour ce type de signal, nous allons considérer leurs *séries de Fourier*. Celles-ci peuvent être vues comme des transformées de Fourier particulières : il s'agit en effet de transformées de Fourier de fonctions périodiques. Dans la suite, L^1_{loc} désigne l'espace des signaux localement stable, *i.e.* :

$$L^1_{\text{loc}} \stackrel{\text{def}}{=} \left\{ s;\ \forall A \text{ compact} \subset \mathbf{R},\ \mathbf{1}_A \cdot s \in L^1 \right\}.$$

1.2.1 Définitions, coefficients de Fourier

On rappelle le cadre fonctionnel considéré.

Définition 1.4 — (Signal périodique) Soit $T > 0$. Un signal s est dit T-périodique si :

$$\forall t \in \mathbf{R}, s(t+T) = s(t).$$

Un signal s T-périodique est dit de plus *localement stable* si $s \in L^1_{\text{loc}}$.

Et dans le cas des fonctions périodiques, la transformée de Fourier est définie sur un ensemble discret, c'est la suite des coefficients de Fourier.

Définition 1.5 — (Transformée de Fourier dans L^1_{loc}) La transformée de Fourier $(\hat{s}_n)_{n\in\mathbf{Z}}$ d'un signal localement stable s, T-périodique, est définie par la formule :

$$\hat{s}_n \stackrel{\text{def}}{=} \frac{1}{T}\int_0^T s(t)\mathrm{e}^{-2\mathrm{i}\pi\frac{n}{T}t}\mathrm{d}t.$$

Étant donné $n \in \mathbf{Z}$, $\hat{s}_n$ est appelé n-ième coefficient de Fourier du signal s.

Les coefficients de Fourier permettent d'écrire les fonctions périodiques localement intégrable comme une série :

$$\forall t \in [0,T], \quad s(t) = \sum_{n=-\infty}^{+\infty} \hat{s}_n \mathrm{e}^{2\mathrm{i}\pi t\frac{n}{T}}.$$

La convergence de ces séries de Fourier est un sujet à part entière et nous n'établirons uniquement ici qu'un résultat de convergence presque partout,

résultat analogue à la formule d'inversion dans le cas de la transformée de Fourier dans L^1 (voir théorème 1.1).

On notera

$$S_n(t) = \sum_{k=-n}^{n} \hat{s}_k e^{2i\pi t \frac{k}{T}} \tag{1.1}$$

la somme partielle.

Bien entendu d'autres résultats de convergence pour les séries de Fourier existent et nous renvoyons à [5, 3] pour plus de détails.

Illustration graphique

Le vidéaste *3Blue1Brown* [4], mathématicien et vulgarisateur mathématique sur YouTube, a réalisé une vidéo où il illustre la décomposition de Fourier d'un chemin fermé par des animations de mécanismes d'engrenages de cercles mis bout à bout. Le résultat est magnifique et envoûtant (voir figure 1.2).

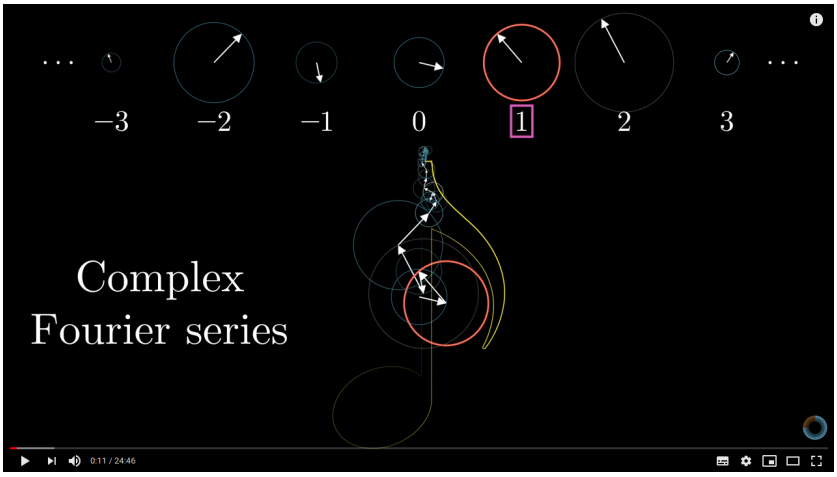


FIGURE 1.2 – Vidéo *Mais qu'est-ce qu'une série de Fourier ? Du transfert thermique à des dessins avec des cercles* du vidéaste *3Blue1Brown* sur YouTube.

Cette animation est une application directe de la décomposition en séries de Fourier. Ici, on considèrera une fonction périodique dans $\mathbf{R}^2$ (ou dans $\mathbf{C}$).

Géométriquement, on peut voir la relation (1.1) comme une somme de vecteurs de $\mathbf{R}^2$ (donc mis bout à bout) de norme le module du nombre complexe $\hat{s}_k e^{2i\pi t \frac{k}{T}}$ et, comme orientation, son argument. Ainsi, quand t parcourt

4. https://www.youtube.com/watch?v=-qgreAUpPwM

l'intervalle $[0, T]$, ces vecteurs bout à bout tournent et l'extrémité du dernier vecteur dessine la courbe fermée [5] que l'on a décomposée.

En figure 1.3, vous pouvez voir une illustration de ceci appliqué au contour de la lettre f. Si vous visionnez le PDF de ce document avec Acrobat Reader, vous pourrez la voir sous forme d'animation [6].

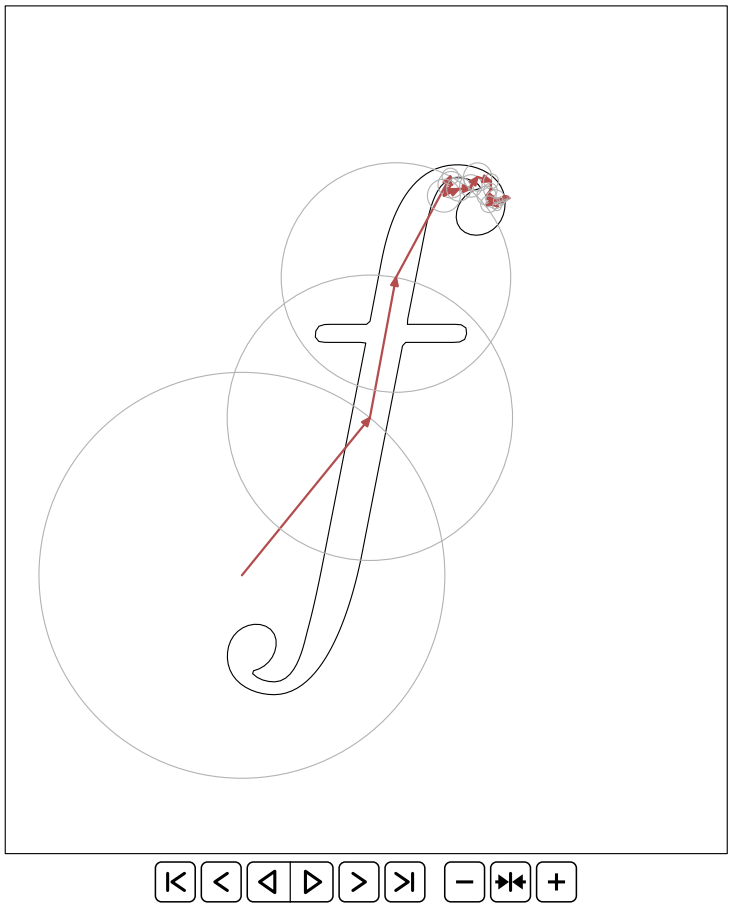


FIGURE 1.3 – Illustration de la décomposition en série de Fourier d'un contour de lettre.

1.2.2 Convolution

Puisque le produit de convolution est une opération essentiellement algébrique, elle s'applique au cadre précédemment défini. On peut même convoler

5. Ou plutôt une approximation du contour.
6. Cette décomposition a été réalisée avec LuaLaTeX !

des fonctions périodiques et des fonctions L^1, comme l'explique le résultat suivant.

LEMME 1.3 Soit s un signal localement stable T-périodique et h un signal stable. Alors la fonction g définie par la formule :

$$\forall t \in \mathbf{R}, \quad g(t) \stackrel{\text{def}}{=} \int_{\mathbf{R}} h(t-u)s(u)\mathrm{d}u$$

est T-périodique, définie presque partout et localement stable.
De plus son n-ième coefficient de Fourier est donné par la formule :

$$\forall n \in \mathbf{Z}, \quad \hat{g}_n = \hat{h}\left(\frac{n}{T}\right)\hat{s}_n.$$

Cette dernière formule est à mettre en relation avec le fait que la transformée de Fourier change un produit de convolution en produit.

Preuve : (hors programme) Par changement de variable, on a :

$$\int_{\mathbf{R}} h(t-u)s(u)\mathrm{d}u = \int_0^T \tilde{h}_T(t-u)s(u)\mathrm{d}u,$$

où l'on a noté $\tilde{h}_T(u) = \sum_{n\in\mathbf{Z}} h(u+nT)$. De plus, on peut montrer que g est périodique de période T partout où elle est définie. En effet, en utilisant le fait que $h \star s = s \star h$, on a

$$g(t+T) = \int_{\mathbf{R}} h(u)s(t+T-u)\mathrm{d}u = \int_{\mathbf{R}} h(u)s(t-u)\mathrm{d}u = g(t).$$

Comme :

$$\int_0^T |\tilde{h}_T(u)|\mathrm{d}u \leq \int_{\mathbf{R}} |h(u)|\mathrm{d}u < +\infty,$$

on a en utilisant le théorème de Fubini (voir annexe A)

$$\begin{aligned}\int_0^T |g(t)|\,\mathrm{d}t &\leqslant \int_0^T \int_0^T |\tilde{h}_T(t-u)||s(u)|\mathrm{d}u\mathrm{d}t \\ &\leqslant \|h\|_{L^1} \int_0^T |s(u)|\,\mathrm{d}u \\ &< +\infty\end{aligned}$$

Ainsi pour presque tout $t \in [0,T]$, $t \mapsto \int_{\mathbf{R}} h(t-u)s(u)\mathrm{d}u$ est définie, et par T-périodicité

pour presque tout $t \in \mathbf{R}$. On vérifie ensuite aisément que g est localement stable et enfin, on montre :

$$\begin{aligned}\hat{g}_n &= \frac{1}{T}\int_0^T g(t)\mathrm{e}^{2\mathrm{i}\pi\frac{n}{T}t}\mathrm{d}t \\ &= \frac{1}{T}\int_0^T\int_0^T \tilde{h}_T(t-u)s(u)\mathrm{e}^{2\mathrm{i}\pi\frac{n}{T}t}\mathrm{d}u\mathrm{d}t \\ &= \frac{1}{T}\int_0^T\int_0^T \left(\tilde{h}_T(t-u)\mathrm{e}^{2\mathrm{i}\pi\frac{n}{T}(t-u)}\right)s(u)\mathrm{e}^{-2\mathrm{i}\pi\frac{n}{T}u}\mathrm{d}u\mathrm{d}t \\ &= \hat{h}\left(\frac{n}{T}\right)\hat{s}_n,\end{aligned}$$

ce qui achève la preuve. ■

La formule que nous venons d'obtenir sur les coefficients peut sembler sophistiquée, mais elle nous sera utile par la suite.

1.2.3 Formule d'inversion et formule de Poisson faible

Dans cette section, nous établissons la formule de Poisson qui permet de faire le lien entre transformée de Fourier et série de Fourier, et qui est une des briques élémentaires de la preuve du Théorème de Shannon [7]-Nyquist [8] qui, rappelons le, est le but de ce chapitre.

Noyau de Poisson **

On introduit tout d'abord le noyau de Poisson et quelques-unes de ses propriétés.

DÉFINITION 1.6 — (Noyau de Poisson) Étant donné un réel r appartenant à $]-1,1[$ et $T>0$, on appelle *noyau de Poisson* la fonction P_r définie

7. Claude Elwood Shannon (30 avril 1916 Gaylord, Michigan - 24 février 2001) est un ingénieur électricien et mathématicien américain. Il est l'un des pères, si ce n'est le père fondateur, de la théorie de l'information.

8. Harry Nyquist (7 février 1889 Nilsby, Suède- 4 avril 1976, Harlingen, Texas) a été un important contributeur à la théorie de l'information et à l'automatique. Ses travaux théoriques sur la détermination de la bande passante nécessaire à la transmission d'information, publiés dans l'article *Certain factors affecting telegraph speed* posent les bases des recherches de Claude Shannon.

sur $\mathbf{R}$ et à valeur dans $\mathbf{C}$ par :

$$P_r(t) \stackrel{\text{def}}{=} \sum_{n\in\mathbf{Z}} r^{|n|} \mathrm{e}^{2\mathrm{i}\pi\frac{n}{T}t}.$$

Certaines propriétés de ce noyau sont indiquées dans la proposition suivante que nous ne démontrerons pas.

Proposition 1.3 : *La fonction P_r vérifie :*

1. $P_r(t) = \sum_{n\in\mathbf{N}} r^{|n|}\mathrm{e}^{2\mathrm{i}\pi\frac{n}{T}t} + \sum_{n\in\mathbf{N}} r^{|n|}\mathrm{e}^{-2\mathrm{i}\pi\frac{n}{T}t} - 1 = \frac{1-r^2}{\left|1-r\mathrm{e}^{2\mathrm{i}\pi\frac{t}{T}}\right|^2}$.
2. *Pour tout réel t, on a :*
$$P_r(t) \geqslant 0.$$
3. *La fonction est normalisée :*
$$\frac{1}{T}\int_{-T/2}^{T/2} P_r(t)\mathrm{d}t = 1.$$
4. *Pour tout $\varepsilon > 0$, on a :*
$$\frac{1}{T}\int_{[-T/2,T/2]-[-\varepsilon,\varepsilon]} P_r(t)\mathrm{d}t \leqslant \frac{1-r^2}{\left|1-r\mathrm{e}^{2\mathrm{i}\pi\frac{\varepsilon}{T}}\right|^2} \xrightarrow[r\to 1]{} 0.$$

Formule d'inversion

Les propriétés 2, 3, 4 correspondent à ce que l'on appelle une *identité approchée*. On a déjà rencontré cette notion, sans le signaler, avec la fonction h_σ de la preuve du théorème 1.1. Les identités approchées permettent de régulariser les fonctions par convolution et d'utiliser ensuite des formules du type :

$$\lim_{r\to 1}\frac{1}{T}\int_0^T \varphi(t)P_r(t)\mathrm{d}t = \varphi(0), \tag{1.2}$$

lorsque l'on souhaite revenir à la fonction initiale. C'est ce raisonnement que nous allons maintenant effectuer avec le noyau de Poisson pour obtenir une formule d'inversion.

THÉORÈME 1.3 — (Formule d'inversion) Soit s un signal T-périodique localement stable, tel que

$$\sum_{n\in\mathbf{Z}} |\hat{s}_n| < +\infty.$$

Alors, pour presque tout $t \in \mathbf{R}$,

$$s(t) = \sum_{n\in\mathbf{Z}} \hat{s}_n e^{2i\pi\frac{n}{T}t}.$$

Autrement dit, si $\sum_{n\in\mathbf{Z}} |\hat{s}_n| < +\infty$, la fonction s est connue dès lors que ses coefficients de Fourier sont connus.

Remarque 1.4 :

Si on suppose en outre que s est continue, alors la formule d'inversion devient vraie pour tout $t \in \mathbf{R}$.

Preuve : (hors programme) Par le calcul, on a :

$$\frac{1}{T}\int_0^T s(u)P_r(t-u)\mathrm{d}u = \sum_{n\in\mathbf{Z}} \hat{s}_n r^{|n|} e^{2i\pi\frac{n}{T}t},$$

puisque toutes les intégrales considérées sont convergentes. De plus :

$$\lim_{r\to 1}\int_0^T \left|\frac{1}{T}\int_0^T s(u)P_r(t-u)\mathrm{d}u - s(t)\right| \mathrm{d}t = 0,$$

d'après (1.2) et le théorème de convergence dominée. Par conséquent la suite de fonctions

$$t \mapsto \varphi_r(t) \overset{\text{def}}{=} \sum_{n\in\mathbf{Z}} \hat{s}_n r^{|n|} e^{2i\pi\frac{n}{T}t}$$

(indexée par r) converge dans L^1_{loc} vers s. D'autre part, puisque $\sum_{n\in\mathbf{Z}} |\hat{s}_n| < +\infty$, la fonction $\varphi_r(t)$ tend vers $\sum_{n\in\mathbf{Z}} \hat{s}_n \exp(2i\pi\frac{n}{T}t)$ ponctuellement. On utilise alors le résultat du cours d'intégration suivant :

« Si une suite de fonction f_n converge vers f dans L^p et vers g presque partout, alors $f = g$ presque partout »

pour conclure que pour presque tout $t \in \mathbf{R}$:

$$s(t) = \sum_{n\in\mathbf{Z}} \hat{s}_n e^{2i\pi\frac{n}{T}t}.$$

Au passage, on en déduit le résultat suivant.

Corollaire 1.1 : *Deux signaux T-périodiques stable ayant les mêmes coefficients de Fourier sont égaux presque partout.*

Formule de Poisson faible

Le théorème suivant établit un lien entre séries de Fourier et transformée de Fourier.

THÉORÈME 1.4 — (Formule de Poisson faible) Soit s un signal stable et $T > 0$. La série $\sum_{n\in\mathbf{Z}} s(t+nT)$ converge presque partout vers une fonction ϕ T-périodique, localement intégrable et dont le n-ième coefficient de Fourier est donné par la formule :

$$\widehat{\phi}_n = \frac{1}{T}\hat{s}\left(\frac{n}{T}\right).$$

Formellement, ce théorème nous dit que ϕ est telle que :

$$\forall t \in \mathbf{R}, \quad \phi(t) = \sum_{n\in\mathbf{Z}} s(t+nT).$$

De plus, ϕ est T-périodique et sa série de Fourier formelle S_f est :

$$S_f(t) = \frac{1}{T}\sum_{n\in\mathbf{Z}} \hat{s}\left(\frac{n}{T}\right) \mathrm{e}^{2\mathrm{i}\frac{\pi}{T}t}.$$

On utilise ici le mot formelle car nous ne disons rien sur la convergence de cette série de Fourier.

Remarque 1.5 :

Si on arrive à montrer $\phi(0) = S_f(0)$, alors on aura la formule de Poisson forte : pour tout $s \in L^1$,

$$T\sum_{n\in\mathbf{Z}} s(nT) = \sum_{n\in\mathbf{Z}} \hat{s}\left(\frac{n}{T}\right).$$

Ce résultat faible nous suffira pour établir le théorème de Shannon-Nyquist dans la section suivante.

Preuve (du théorème 1.4) : On montre aisément que ϕ est T-périodique. Notons en-

suite que ϕ est bien définie, puisque :

$$\begin{aligned}\int_0^T \sum_{n\in\mathbf{Z}} |s(t+nT)|\mathrm{d}t &= \sum_{n\in\mathbf{Z}} \int_0^T |s(t+nT)|\mathrm{d}t \\ &= \sum_{n\in\mathbf{Z}} \int_{nT}^{(n+1)T} |s(t)|\mathrm{d}t \\ &= \int_{\mathbf{R}} |s(t)|\mathrm{d}t < +\infty,\end{aligned} \tag{1.3}$$

et que donc $\sum_{n\in\mathbf{Z}} |s(t+nT)| < +\infty$ presque partout.

La formule (1.3) montre en outre que ϕ est localement intégrable. On peut donc calculer son coefficient de Fourier. On obtient :

$$\begin{aligned}\widehat{\phi}_n =& \frac{1}{T}\int_0^T \phi(t)\mathrm{e}^{-2\mathrm{i}\pi\frac{n}{T}t}\mathrm{d}t \\ =& \frac{1}{T}\int_0^T \left(\sum_{k\in\mathbf{Z}} s(t+kT)\right)\mathrm{e}^{-2\mathrm{i}\pi\frac{n}{T}t}\mathrm{d}t \\ =& \frac{1}{T}\int_0^T \left(\sum_{k\in\mathbf{Z}} s(t+kT)\mathrm{e}^{-2\mathrm{i}\pi\frac{n}{T}(t+kT)}\right)\mathrm{d}t \\ =& \frac{1}{T}\int_{\mathbf{R}} s(t)\mathrm{e}^{-2\mathrm{i}\pi\frac{n}{T}t}\mathrm{d}t \\ =& \frac{1}{T}\hat{s}\left(\frac{n}{T}\right),\end{aligned}$$

ce qui achève la preuve. ■

Pour aller un peu plus loin vers la version forte, on peut énoncer le résultat suivant.

Proposition 1.4 : *Soit s un signal stable tel que $\sum_{n\in\mathbf{Z}} \left|\hat{s}\left(\frac{n}{T}\right)\right| < +\infty$. Alors :*

$$\sum_{n\in\mathbf{Z}} s(t+nT) = \frac{1}{T}\sum_{n\in\mathbf{Z}} \hat{s}\left(\frac{n}{T}\right)\mathrm{e}^{2\mathrm{i}\pi\frac{n}{T}t}.$$

Ce résultat est en fait une simple application de la formule d'inversion obtenue au théorème 1.3.

1.3 Reconstruction des signaux périodiques

Dans cette dernière section nous allons répondre aux deux questions suivantes :

1. Étant donné un signal analogique, quels critères doivent-être appliqués pour en extraire une suite discrète représentative ?
2. Comment reconstruire un signal analogique à partir d'un échantillon discret de valeurs ?

Les réponses à ces deux questions sont en partie données par le théorème de Shannon-Nyquist, lui-même obtenu à partir de la formule de Poisson vue à la section précédente.

1.3.1 Le théorème de Shannon-Nyquist

Avant d'énoncer le théorème, on montre deux résultats préliminaires.

LEMME 1.4 Soit s, un signal stable et continu, de transformée de Fourier $\hat{s}$ stable et un réel $B > 0$. Supposons que :

$$\sum_{n\in\mathbf{Z}} \left|s\left(\frac{n}{2B}\right)\right| < +\infty. \tag{1.4}$$

Alors :

$$\sum_{j\in\mathbf{Z}} \hat{s}(\nu + 2jB) = \frac{1}{2B} \sum_{n\in\mathbf{Z}} s\left(\frac{n}{2B}\right) \mathrm{e}^{-2\mathrm{i}\pi\nu\frac{n}{2B}},$$

pour presque tout $\nu \in \mathbf{R}$.

Preuve : D'après la formule de Poisson faible, la fonction $\nu \mapsto \phi(\nu) = \sum_{j\in\mathbf{Z}} \hat{s}(\nu + 2jB)$ est localement intégrable et son n-ième coefficient de Fourier vaut :

$$\widehat{\phi}_n = \frac{1}{2B} \int_{\mathbf{R}} \hat{s}(\nu) \mathrm{e}^{-2\mathrm{i}\pi\frac{n\nu}{2B}} \mathrm{d}\nu.$$

Mais puisque $\hat{s}$ est stable, on peut appliquer la formule d'inversion du théorème 1.1. Dans notre cas, celle-ci s'écrit :

$$\int_{\mathbf{R}} \hat{s}\,(\nu)\, \mathrm{e}^{-2\mathrm{i}\pi\frac{n\nu}{2B}} \mathrm{d}\nu = s\left(-\frac{n}{2B}\right).$$

En combinant les deux formules, nous obtenons donc formellement :

$$\phi(\nu) = \frac{1}{2B} \sum_{n\in\mathbf{Z}} s\left(\frac{n}{2B}\right) \mathrm{e}^{-2\mathrm{i}\pi\frac{n\nu}{2B}}.$$

Mais l'hypothèse (1.4), permet l'utilisation du théorème d'inversion 1.3 et donc justifie rigoureusement cette dernière formule. ■

Montrons maintenant un second lemme technique. .

LEMME 1.5 Soit s un signal vérifiant les hypothèses du lemme 1.4 et h un signal de la forme :

$$h(t) = \int_{\mathbf{R}} \chi(\nu) \mathrm{e}^{2\mathrm{i}\pi\nu t} \mathrm{d}\nu, \tag{1.5}$$

où χ est stable. Alors, le signal :

$$\tilde{s}(t) = \frac{1}{2B} \sum_{n\in\mathbf{Z}} s\Big(\frac{n}{2B}\Big) h\Big(t - \frac{n}{2B}\Big) \tag{1.6}$$

admet la représentation :

$$\tilde{s}(t) = \int_{\mathbf{R}} \left(\sum_{j\in\mathbf{Z}} \hat{s}(\nu + 2jB) \right) \chi(\nu) \mathrm{e}^{2\mathrm{i}\pi\nu t} \mathrm{d}\nu.$$

Preuve : Notons tout d'abord que $\tilde{s}$ est borné et continu, en tant que somme d'une série normalement convergente (en plus des hypothèses sur s, h est bornée, et uniformément continue).

En remplaçant h par sa valeur dans (1.6), on obtient :

$$\begin{aligned}\tilde{s}(t) &= \frac{1}{2B} \sum_{n\in\mathbf{Z}} s\Big(\frac{n}{2B}\Big) \int_{\mathbf{R}} \chi(\nu) \mathrm{e}^{2\mathrm{i}\pi\nu\left(t - \frac{n}{2B}\right)} \mathrm{d}\nu \\ &= \int_{\mathbf{R}} \sum_{n\in\mathbf{Z}} \frac{1}{2B} s\Big(\frac{n}{2B}\Big) \mathrm{e}^{-\frac{2\mathrm{i}\pi\nu n}{2B}} \chi(\nu) \mathrm{e}^{2\mathrm{i}\pi\nu t} \mathrm{d}\nu,\end{aligned}$$

où on a utilisé le théorème de Fubini, applicable car :

$$\int_{\mathbf{R}} \sum_{n\in\mathbf{Z}} \Big|s\Big(\frac{n}{2B}\Big)\Big| \cdot |\chi(\nu)| \, \mathrm{d}\nu = \left(\sum_{n\in\mathbf{Z}} \Big|s\Big(\frac{n}{2B}\Big)\Big| \right) \cdot \left(\int_{\mathbf{R}} |\chi(\nu)| \, \mathrm{d}\nu \right) < +\infty.$$

Donc :

$$\tilde{s}(t) = \int_{\mathbf{R}} g(\nu) \chi(\nu) \mathrm{e}^{2\mathrm{i}\pi\nu t} \mathrm{d}\nu,$$

avec :

$$g(\nu) = \sum_{n\in\mathbf{Z}} \frac{1}{2B} s\Big(\frac{n}{2B}\Big) \mathrm{e}^{-\frac{2\mathrm{i}\pi\nu n}{2B}},$$

ou encore, d'après le lemme 1.4 :

$$g(\nu) = \sum_{j\in\mathbf{Z}} \hat{s}(\nu + 2jB),$$

ce qui conduit au résultat annoncé. ■

On peut alors énoncer le fameux théorème de Shannon-Nyquist.

THÉORÈME 1.5 — (de Shannon-Nyquist) Soit s, un signal stable et continu dont la transformée de Fourier $\hat{s}$ s'annule en dehors d'un intervalle $[-B, B]$. Supposons également que (1.4) soit vérifiée. Alors,

$$s(t) = \sum_{n\in\mathbf{Z}} s\Big(\frac{n}{2B}\Big)\operatorname{sinc}\Big((2Bt-n)\pi\Big). \tag{1.7}$$

Dans le cadre temporel, ce théorème nous dit que si un signal n'a pas de fréquence plus haute que B hertz, alors il est complètement déterminé par les valeurs discrètes séparées de $1/2B$ seconde.

Ainsi le théorème de Shannon-Nyquist nous indique comment choisir une fréquence d'échantillonnage lorsqu'on a des informations sur le support de la transformée de Fourier du signal [9] considérée et comment le reconstruire.

Démontrons ce théorème.

Preuve : Définissons χ par la formule :

$$\chi = \mathbf{1}_{[-B,B]},$$

de telle sorte que la fonction h de lemme 1.5 soit définie par :

$$\forall t \in \mathbf{R}, \quad h(t) = 2B\operatorname{sinc}(2\pi Bt) = 2B\frac{\sin(2\pi Bt)}{2\pi Bt}.$$

Alors :

$$\Big(\sum_{j\in\mathbf{Z}} \hat{s}(\nu + 2jB)\Big)\chi(\nu) = \hat{s}(\nu)\chi(\nu) = \hat{s}(\nu),$$

d'après le choix de χ et l'hypothèse faite sur le support de $\hat{s}$.

Le lemme 1.5 donne alors :

$$\tilde{s}(t) = \int_{\mathbf{R}} \hat{s}(\nu)\mathrm{e}^{2\mathrm{i}\pi\nu t}\mathrm{d}\nu = \frac{1}{2B}\sum_{n\in\mathbf{N}} s\Big(\frac{n}{2B}\Big)h\Big(t-\frac{n}{2B}\Big),$$

puis, par le théorème d'inversion 1.1, on obtient :

$$\int_{\mathbf{R}} \hat{s}(\nu)\mathrm{e}^{2\mathrm{i}\pi\nu t}\mathrm{d}\nu = s(t),$$

ce qui achève la démonstration. ■

La fonction χ joue donc finalement le rôle d'une troncature, puisqu'elle tronque artificiellement le support de la fonction considérée.

9. Les ingénieurs parlent plutôt de spectre d'un signal.

Échantillonnage, distribution de Dirac et illustration **

Lorsqu'on utilise le cadre des distributions tempérées (voir [2, 12]), alors l'échantillonnage d'un signal s revient à multiplier ce signal par un *peigne de Dirac*.

DÉFINITION 1.7 — (Distribution de Dirac) La distribution de Dirac, notée δ_0, peut être vue comme la forme linéaire continue de norme 1 définie par :

$$\delta_0 : \begin{array}{ccc} \mathcal{D}(\mathbf{R}) & \to & \mathbf{R} \\ \phi & \mapsto & \langle \delta_0, \phi \rangle \stackrel{\text{def}}{=} \phi(0) \end{array}$$

où $\mathcal{D}(\mathbf{R})$ est l'ensemble des fonctions $\mathcal{C}^\infty$ à support compact.

Par abus de langage, cette distribution est souvent expliquée comme étant une « fonction » qui vaut zéro partout, sauf en 0 où sa valeur infinie correspond à une « masse » de 1.

À partir de cette distribution, on peut définir un outil important de traitement du signal.

DÉFINITION 1.8 — (Peigne de Dirac) Le peigne de Dirac III est défini par

$$\mathrm{III}_T \stackrel{\text{def}}{=} \sum_{k=-\infty}^{\infty} \delta_{kT}$$

où pour $a \in \mathbf{R}$, δ_a est la masse de Dirac translatée en a, *i.e.* :

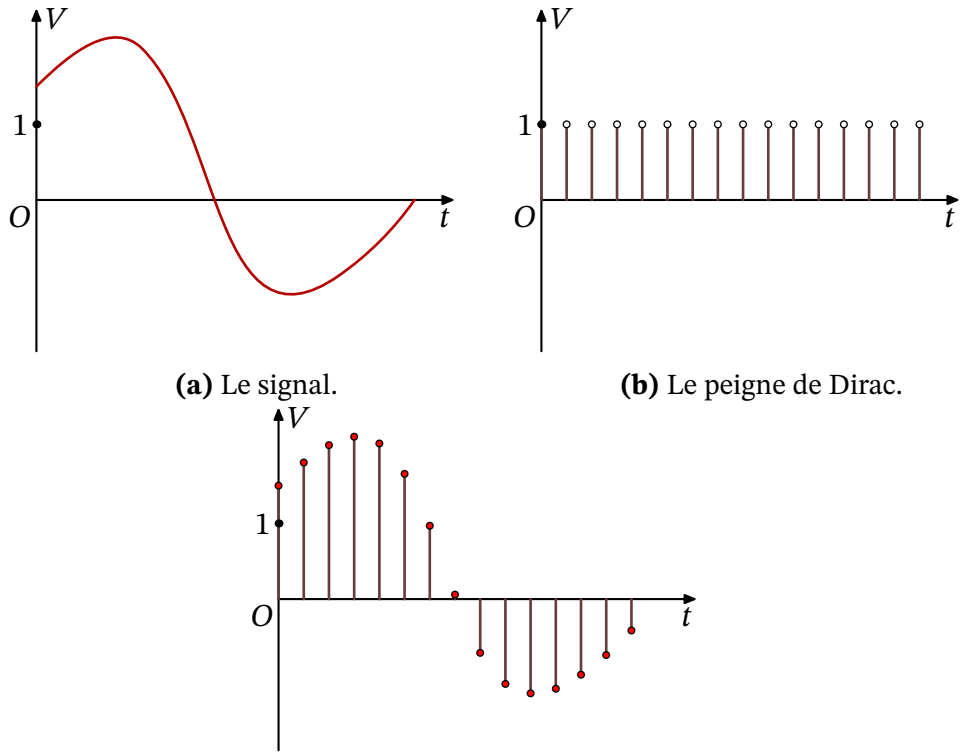
$$\delta_a : \begin{array}{ccc} \mathcal{D}(\mathbf{R}) & \to & \mathbf{R} \\ \phi & \mapsto & \langle \delta_a, \phi \rangle \stackrel{\text{def}}{=} \phi(a) \end{array}.$$

Grâce au peigne de Dirac, l'échantillonnage d'un signal s de fréquence $1/T$, noté s_T, est alors simplement sa multiplication avec le peigne de Dirac.

$$s_T(t) = \sum_{n=-\infty}^{+\infty} s(nT)\delta_{nT}(t),$$

avec l'abus de notation $\delta_a(\cdot)$ comme « fonction ».

Cette formalisation donne une bonne vision de l'échantillonnage qui correspond au fait de prendre des points sur le graph du signal, voir figure 1.4.



(a) Le signal.

(b) Le peigne de Dirac.

(c) Le signal échantillonné.

FIGURE 1.4 – Procédé d'échantillonnage d'un signal vu comme multiplication du signal avec un peigne de Dirac.

1.3.2 Phénomène de recouvrement de spectre

Si le support de la transformée de Fourier du signal, ou encore le spectre du signal, est inclus dans l'intervalle $[-B, B]$, alors, une fréquence d'échantillonnage correcte est $f_e = 1/2B$, parfois appelée *fréquence de Nyquist*. Mais que se passe-t-il lorsqu'on *sous-échantillonne*, c'est-à-dire, lorsque l'hypothèse faite sur le support n'est plus valide ?

Sans passer par l'analyse de Fourier, on peut se convaincre qu'il faut une certaine fréquence d'échantillonnage $1/T$ pour être capable de reconstruire le signal, sans même le traiter. La figure 1.5 montre un signal sinusoïdale, un échantillonnage de celui-ci, et le signal sinusoïdale qu'on peut reconstruire à partir des échantillon.

Supposons donc que l'affirmation $\mathrm{supp}(\hat{s}) \subset [-B, B]$ soit fausse. Après multiplication de $\hat{s}$ par $\mathbf{1}_{[-B,B]}$, c'est-à-dire convolution par $h(t) = 2B\,\mathrm{sinc}(2\pi Bt)$,

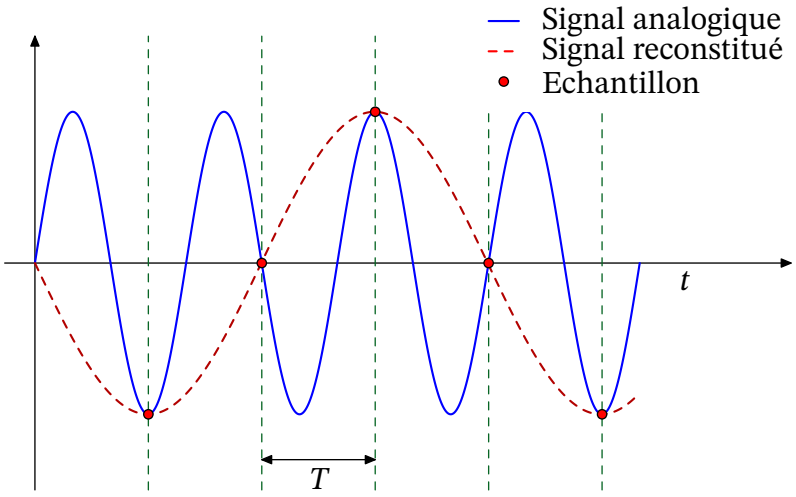


FIGURE 1.5 – Phénomène de sous-échantillonnage.

on se ramène au cadre précédent et on a :

$$\tilde{s}(t) = \sum_{n\in\mathbf{Z}} s\Big(\frac{n}{2B}\Big)\operatorname{sinc}\big((2Bt-n)\pi\big).$$

Quelle est la transformée de Fourier de ce signal ?

Proposition 1.5 : *Soit s un signal stable et continu tel que la condition* (1.4) *soit satisfaite. Alors le signal :*

$$\tilde{s}(t) = \sum_{n\in\mathbf{N}} s\Big(\frac{n}{2B}\Big)\operatorname{sinc}\big((2BT-n)\pi\big),$$

admet la représentation :

$$\tilde{s}(t) = \int_{\mathbf{R}} \hat{\tilde{s}}(\nu)\mathrm{e}^{2\mathrm{i}\pi\nu t}\mathrm{d}\nu,$$

avec :

$$\hat{\tilde{s}}(\nu) = \Big(\sum_{j\in\mathbf{Z}} \hat{s}(\nu+2jB)\Big)\mathbf{1}_{[-B,B]}(\nu). \tag{1.8}$$

La démonstration de ce résultat s'obtient en reprenant ligne à ligne celle du théorème 1.5. La figure 1.6 donne une illustration de ce phénomène.

En conséquence, on voit que les supports des fonctions dans la somme (1.8) vont se recouvrir sur les bords de l'intervalle $[-B,B]$, ce qui altérera le spectre du signal reconstruit et par conséquent, le signal lui-même. Ce phénomène est

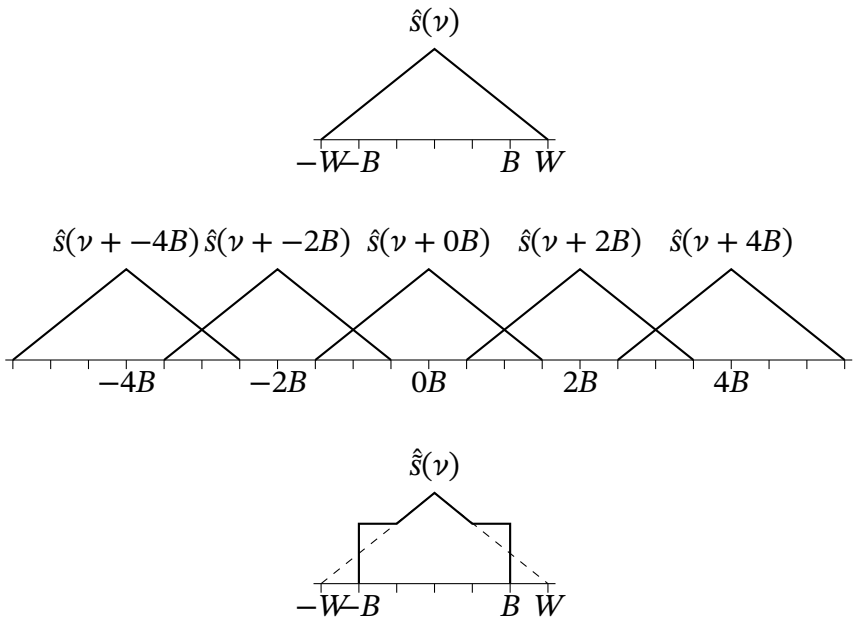


FIGURE 1.6 – Illustration du phénomène de recouvrement de spectre, phénomène connu sous le nom d'*aliasing*.

appelé *recouvrement du spectre* ou encore, en anglais, *aliasing*.

La conclusion pratique que l'on peut tirer de cette proposition est que, lorsque l'on souhaite discrétiser [10] un signal, il faut tout d'abord en tronquer le spectre — à l'aide d'un filtre, voir les chapitres 5 — puis l'échantillonner à une fréquence au moins égale à celle de Nyquist.

Un exemple simple est celui de la numérisation de signaux sonores. Les fréquences audibles pour un nourrisson vont de 20 Hz à 20 kHz. Pour concevoir le format de codage des CD audio, les ingénieur·e·s ont choisi une fréquence d'échantillonnage de 44 kHz, c'est-à-dire à peu près le double de la fréquence maximale audible, ce qui correspond bien à la fréquence de Nyquist. Les 4 kHz supplémentaires correspondent à des corrections d'erreurs qui seront présentées au chapitre 3. Pour les conversations téléphoniques, l'échantillonnage se fait à la cadence de 8 kHz, qui est plus faible que celle des CD audio, car les spectres considérés (que l'on souhaite transmettre) sont beaucoup plus étroits. La qualité du rendu n'est plus un critère prédominant.

10. En utilisant un vocabulaire plus proche de celui des ingénieurs, on dirait « numériser ».

1.3.3 Sur-échantillonnage

On peut maintenant se poser la question inverse. Quelles sont les conséquences d'un échantillonnage à une fréquence trop élevée ?

Nous allons voir que le sur-échantillonnage peut permettre d'accélérer la vitesse de convergence. En effet, on peut remarquer que dans la formule (1.7) la série obtenue converge très lentement, grosso-modo en $1/n$. En effet, la quantité $\operatorname{sinc}(2Bt - n)$ est d'ordre $1/n$ et de signe alterné. Elle n'est donc pas satisfaisante d'un point de vue pratique.

Supposons que l'on sur-échantillonne un signal s : soit W tel que $\operatorname{supp}(\hat{s}) \subset [-W, W]$, et choisissons la fréquence d'échantillonnage $1/2B$ telle que $B = (1+\alpha)W$, pour un certain $\alpha > 0$.

Choisissons notre fonction de troncature χ de telle sorte que :

$$\forall \nu \in [-W, W], \qquad \chi(\nu) = 1,$$

et

$$\forall \nu \in \mathbf{R} - [-B, B], \qquad \chi(\nu) = 0,$$

où nous ne disons rien entre sur $]-B, -W[$ et sur $]W, B[$.

Alors, le théorème de Shannon-Nyquist s'applique et en en reproduisant la preuve, on obtient :

$$s(t) = \frac{1}{2B} \sum_{n \in \mathbf{Z}} s\Big(\frac{n}{2B}\Big) h\Big(t - \frac{n}{2B}\Big),$$

où la fonction h est la transformée inverse de χ donnée par la formule (1.5). On met ainsi en évidence un nouveau degré de liberté : par un choix judicieux de h, i.e. de χ sur $]-B, -W[$ et sur $]W, B[$, on peut accélérer la vitesse de convergence. Ceci se fait en particulier en augmentant la régularité de la fonction χ.

1.4 Note bibliographique

Pour plus détails sur ce chapitre, avec le même cadre fonctionnel, on consultera les parties A et B de la référence [3]. Pour approfondir la théorie de la transformée de Fourier, on pourra consulter [5]. Enfin, pour les résultats analogues dans le cadre de la théorie des distributions, on consultera [2, 12].

CHAPITRE 2

Quantification scalaire des signaux discrets

SOMMAIRE DU CHAPITRE

Remarque 2.1 :

Ce chapitre n'est pas au programme du cours, il est là à titre d'information pour satisfaire la curiosité des plus curieux et curieuses.

2.1 Introduction

Après avoir échantillonné un signal analogique, on dispose d'un signal *discret*, c'est-à-dire une suite de nombres réels. À ce stade, la conversion analogique-numérique (souvent désignée par *CAN*) n'est pas encore achevée. Il nous faut encore transformer une suite de nombres réels en une suite de 0 et de 1. C'est l'objet de ce chapitre.

On considère donc une *source* [1] émettant un signal discret en temps. La transformation des nombres émis en suite de nombres binaires consiste en fait en deux opérations : le *codage* et le *décodage*.

Coder, ou *encoder* un signal, c'est appliquer à chacun des termes de la suite qu'il constitue une fonction de la forme :

$$Q : \begin{array}{ccc} [-M,M] & \to & \{I_n\}_{0\leq n\leq N} \\ s(t) & \mapsto & I_k \end{array} \tag{2.1}$$

où :

- l'ensemble $\{I_n\}_{0\leq n\leq N}$ est une famille d'intervalles disjoints formant une partition de l'intervalle $[-M,M]$ (N est un entier naturel),
- $s(t)$ est un réel, représentant la valeur du signal s à l'instant t, supposé appartenir à l'intervalle $[-M,M]$ (M est un réel, qui peut être inconnu).

L'encodage est donc une opération irréversible car faisant intervenir une fonction non-injective, qui entraîne une perte d'information.

Un exemple courant est la quantification sur 3 bits. Dans ce cas, l'intervalle $[-M,M]$ est partitionné en $N = 2^3 = 8$ intervalles.

Décoder un signal, c'est réaliser l'opération inverse, c'est-à-dire appliquer à chacun des termes d'une suite d'intervalles une fonction de la forme :

$$I_n \to y_n,$$

où y_n est un réel représentant l'intervalle I_n, par exemple la valeur de son milieu.

1. Ce terme sera très largement employé dans la suite et désigne un dispositif émettant des signaux à partir de maintenant supposés discrets en temps.

Remarque 2.2 :

Les valeurs y_n sont ensuite elles-même codées par des entiers binaires.

La figure 2.1 représente un signal avant et après quantification.

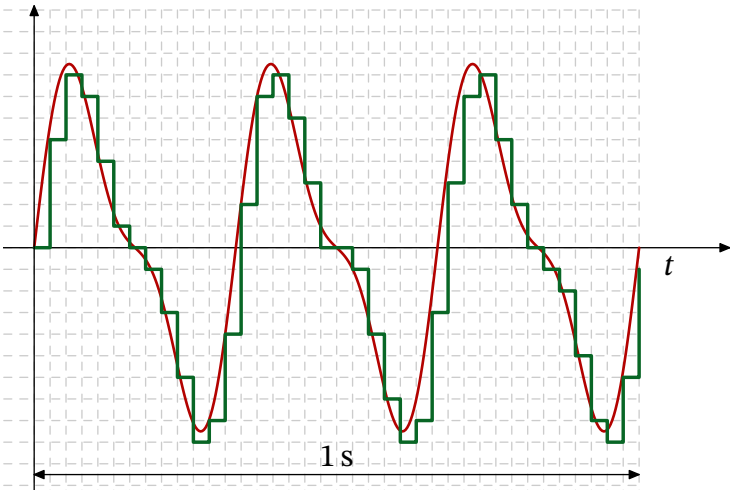


FIGURE 2.1 – Exemple de quantification d'un signal.

2.2 FORMULATION MATHÉMATIQUE

Dans cette section, on formalise mathématiquement le problème de la quantification. Nous allons voir que le problème de quantification fait intervenir beaucoup de paramètre, et nous allons formuler plusieurs problèmes d'optimisation qui permettent de déterminer la quantification. Comme il est courant dans ce genre de modélisation, les problèmes qui en découlent sont souvent trop difficile à attaquer, et il faut passer par des approximations (fixer *a priori* des paramètres, simplifier le modèle, utiliser des méthodes numériques pour résoudre, etc.).

2.2.1 Cadre

On considère donc une source émettant un signal *aléatoire* S de densité de probabilité $f_S \in L^1(\mathbf{R})$. Pour construire une quantification, il faut donc définir une famille d'intervalles (*i.e.* la famille $\{I_n\}_{0\leq n\leq N}$ de l'introduction), donc de frontières, et de valeurs d'assignation (*i.e.* les valeurs y_n de l'introduction).

Pour fixer les idées conservons les notations de l'introduction et désignons par N le nombre d'intervalles constituant la partition et donc également le nombre de valeurs d'assignations. Le nombre de valeurs frontières, aussi appelées *valeurs de décision*, que l'on note b_i est alors $N+1$. On pose *a priori* $b_0 = -\infty$ et $b_N = +\infty$ et on signalera dans la suite les cas où l'on adopte une autre convention. La fonction de quantification s'écrit :

$$Q(x) = y_n,$$

pour $x \in]b_{n-1}, b_n]$.

2.2.2 Erreur de distorsion de quantification

DÉFINITION 2.1 — (Erreur de distorsion de quantification) On définit l'erreur quadratique moyenne de quantification par :

$$\begin{aligned} \sigma_q^2(B,Y) &= \int_{-\infty}^{+\infty} (x - Q(x))^2 f_S(x)\mathrm{d}x \\ &= \sum_{n=1}^{N} \int_{b_{n-1}}^{b_n} (x - y_n)^2 f_S(x)\mathrm{d}x, \end{aligned} \tag{2.2}$$

où l'on a noté $B = \{b_n\}_{0\leq n\leq N}$ et $Y = \{y_n\}_{1\leq n\leq N}$. La quantité σ_q est aussi appelée *erreur de distorsion de quantification*.

Une formalisation possible de l'erreur de quantification consiste à considérer son effet comme celui d'un bruit externe affectant le signal S.

Le problème est alors le suivant : étant donné f_S et N le nombre d'intervalles de quantification, trouver les valeurs b_n et y_n solution du problème

$$\min_{B,Y} \sigma_q^2(B,Y).$$

En conclusion, l'erreur de distorsion de quantification dépend à la fois de la partition choisie et du choix du représentant.

2.2.3 Taux de quantification

DÉFINITION 2.2 — (Taux de quantification) On considère un codage des y_n en entiers binaires de *taille variable* ℓ_n, la taille moyenne des symboles après codage, appelé taux de quantification, est :

$$R = \sum_{n=1}^{N} \ell_n p(y_n), \tag{2.3}$$

où l'on a noté $p(y_n)$ la probabilité d'apparition après codage du symbole correspondant à y_n.

Attention, la valeur R dépend des frontières car :

$$p(y_n) = \int_{b_{n-1}}^{b_n} f_S(x)\mathrm{d}x,$$

et donc :

$$R = \sum_{n=1}^{N} \ell_n \int_{b_{n-1}}^{b_n} f_S(x)\mathrm{d}x.$$

On considère donc selon les cas l'une ou l'autre des reformulations du problème suivantes.

Problème 1

Si l'on se donne une contrainte de distorsion de quantification

$$\sigma_q \leq \sigma^\star, \tag{2.4}$$

où $\sigma^\star$ est un réel fixé, trouver le nombre N, les frontières B et les représentants Y minimisant le taux R, défini par (2.3) et satisfaisant (2.4).

Problème 2

Si l'on se donne une contrainte sur le taux

$$R \leq R^\star, \tag{2.5}$$

où $R^\star$ est un réel fixé, trouver le nombre N, les frontières B et les représentants Y minimisant la distorsion quantification et satisfaisant (2.5).

Remarque 2.3 :

L'une où l'autre de ces formulations constitue un problème plus général que celui que nous allons considérer dans ce chapitre. On se restreint en effet dans la suite à de codage de tailles fixes, si bien que R est constant par rapport à B et Y. Le problème du codage en taille variable sera quant à lui traité au chapitre 3.

2.3 Quantification uniforme

La solution de quantification la plus simple est la quantification uniforme. Dans ce cadre, tout les intervalles ont la même longueur, que l'on note dans la suite Δ. On parle dans ce cas de *quantification uniforme*.

On suppose que l'on adopte une stratégie de codage où N le nombre d'intervalles est pair, c'est-à-dire que $0 \in B$.

2.3.1 Quantification uniforme des sources uniformes

Supposons que pour tout t, $s(t) \in [-S_{\max}, S_{\max}]$ avec une probabilité uniforme. Dans ce cas, on fixe cette fois-ci $b_0 = -S_{\max}$ et $b_N = S_{\max}$. Alors $\Delta = \frac{2S_{\max}}{N}$ et

$$\begin{aligned}\sigma_q^2(B,Y) &= 2\sum_{n=1}^{N/2}\int_{(n-1)\Delta}^{n\Delta}\left(x-\frac{2n-1}{2}\Delta\right)^2\frac{1}{2S_{\max}}\mathrm{d}x\\ &= \frac{\Delta^2}{12}.\end{aligned}$$

Afin d'évaluer l'effet de l'augmentation du nombre d'intervalle sur la qualité du codage, on considère alors le rapport signal/bruit défini par :

$$SNR = 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_q^2}\right),$$

où σ_x désigne l'écart type du signal S.

Puisque S est une variable aléatoire de loi uniforme à valeurs dans $[-S_{\max}, S_{\max}]$, on a alors :

$$\sigma_x^2 = \sqrt{\mathbb{E}[X^2] - \underbrace{\mathbb{E}[X]^2}_{=0}} = \frac{1}{2S_{\max}}\int_{-S_{\max}}^{S_{\max}} x^2\mathrm{d}x = \frac{S_{\max}^2}{3}.$$

Par conséquent, le rapport signal/bruit vaut :

$$\begin{aligned} SNR &= 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_q^2}\right) \\ &= 10\log_{10}\left(\frac{(2S_{\max})^2}{12}\times\frac{12}{\left(\frac{2S_{\max}}{N}\right)^2}\right) \\ &= 10\log_{10}(N^2) \\ &= 20\log_{10}(2^k) \\ &= 6.02k \text{ dB}, \end{aligned} \tag{2.6}$$

où k représente le nombre de bits utilisés pour le codage binaire des représentants y_n.

DÉFINITION 2.3 — (Décibel) Le décibel (dB) est une unité de grandeur sans dimension définie comme dix fois le logarithme décimal du rapport entre deux puissances, utilisé dans les télécommunications, l'électronique et l'acoustique.

La conclusion du calcul précédent est que l'ajout d'un bit de quantification augmente le rapport signal/bruit d'approximativement 6 dB. C'est un résultat très standard et qui est utilisé comme indication du gain maximal possible lorsqu'on augmente le taux de quantification. Ceci n'est qu'une indication car les hypothèses sur le signal sont très fortes et rarement vérifiées en pratique.

2.3.2 Quantification uniforme des sources non-uniformes

Pour introduire ce qui va suivre, considérons l'exemple d'une source émettant dans $[-100, 100]$ dont 95% des valeurs sont dans $[-1, 1]$. Supposons que l'on ait adopté la stratégie de quantification uniforme de la section précédente et que le codage soit effectué sur $k = 3$ (codage sur 3 bits), ce qui revient à considérer 8 intervalles de longueur 25. On a donc, dans 95% des cas une erreur minimum de 11,5. Cet exemple est représentatif des sources gaussiennes et montre le défaut de la quantification uniforme telle que présentée dans la section précédente.

Si l'on souhaite rester dans le cadre de la quantification uniforme, une solution consiste à utiliser la fonction de répartition pour optimiser la taille des intervalles considérés. Concrètement cela revient à considérer σ_q^2 comme une fonction de Δ et à résoudre :

$$\min_{\Delta} \sigma_q^2(\Delta).$$

Explicitons le début du calcul menant à la résolution de ce problème. On a :

$$\sigma_q^2(\Delta) = 2\sum_{n=1}^{N/2-1}\int_{(n-1)\Delta}^{n\Delta}\left(x-\frac{2n-1}{2}\Delta\right)^2 f_S(x)\mathrm{d}x + 2\int_{(N/2-1)\Delta}^{+\infty}\left(x-\frac{N-1}{2}\Delta\right)^2 f_S(x)\mathrm{d}x.$$

Exercice 2.1 :

Montrer que :

$$\frac{\partial\sigma_q^2(\Delta)}{\partial\Delta} = -\sum_{n=1}^{N/2-1}(2n-1)\int_{(n-1)\Delta}^{n\Delta}\left(x-\frac{2n-1}{2}\Delta\right)f_S(x)\mathrm{d}x - (N-1)\int_{(N/2-1)\Delta}^{+\infty}\left(x-\frac{N-1}{2}\Delta\right)f_S(x)\mathrm{d}x.$$

Pour trouver un extremum de σ_q, il faut donc annuler cette dernière valeur. L'équation en découlant n'est pas résoluble algébriquement dans le cas général. On la résout donc approximativement par des méthodes numériques. Il existe aussi des tableaux indiquant les solutions dans les cas de densités de probabilités classiques.

Remarque 2.4 :

Dans ce cas, on peut en fait distinguer deux types d'erreurs de quantification :

1. L'*erreur de surcharge*, qui correspond aux erreurs se produisant dans les deux intervalles extrêmes. On parle également de *bruit de saturation*.
2. L'*erreur granulaire*, qui correspond à l'erreur de quantification dans les autres intervalles. On parle alors de *bruit granulaire*.

Il arrive que l'on ne dispose que d'information partielles sur la source et sa densité de probabilité f_S. Cette carence conduit à des erreurs de modélisation. La fonction f_S^{approx} considérée ne coïncide pas avec la vraie fonction f_S. Il faut donc en fait prévoir une série de tests sur la source pour estimer ses paramètres.

2.4 Quantification adaptative

Une solution simple aux problèmes évoqués dans la section précédente consiste à fonder la stratégie de codage sur des intervalles de longueurs variables. On parle dans ce cas de *quantification adaptative.* Le but est bien sûr d'adapter les paramètres de quantification à la source considérée.

2.4.1 Approches *online* et *offline*

Deux approches sont généralement considérées, l'approche *online* conduisant à une adaptation de la quantification *rétrograde*, c'est-à-dire effectuée en même temps que la quantification elle-même et une adaptation *offline*, où les paramètres de la quantification sont calculés de manière *directe*, c'est-à-dire en fixant *a priori* les paramètres de la quantification.

2.4.2 Quantification adaptative directe – *offline*

Dans cette approche le signal émis est divisé en blocs temporels et chaque bloc est analysé avant la quantification. Les paramètres de la quantification sont calculés en fonction de l'analyse. On a donc, dans ce cas, besoin d'adjoindre à chaque bloc codé un bloc d'information supplémentaire permettant d'indiquer au décodeur les paramètres de quantification qui ont finalement été retenus. Deux problèmes apparaissent dans cette approche.

1. Un problème de synchronisation, car deux types de données sont transmis : le code du signal et les blocs d'informations.
2. Un problème de choix de la taille des bloc de codage considérés. Il faut alors trouver un compromis entre des bloc grands, qui seront alors plus grossièrement décrits par les paramètres de quantifications retenus, et des blocs petits, qui conduiront à de nombreux blocs d'information.

Donnons maintenant un exemple de procédure d'estimation des paramètres d'un bloc à quantifier. Étant donné un bloc de taille M, un estime sa variance [2] au voisinage du temps n par

$$\hat{\sigma}_x^2 = \frac{1}{M} \sum_{i=0}^{M-1} x_{i+n}^2,$$

où on a supposé que notre signal d'entrée avait une valeur moyenne nulle.

2. Il s'agit en fait de ce qu'on appelle en probabilité la variance empirique.

Pour quantifier le bloc considéré, une stratégie possible est alors de considérer une densité de probabilité f_S pour le bloc considéré du signal qui par exemple peut être une gaussienne avec pour variance la variance empirique précédemment calculée et utiliser la stratégie uniforme indiquée à la section 2.3.2.

Évidemment d'autres raffinements sont possible mais dépasse le cadre de cette introduction à la quantification.

Remarque 2.5 :

Attention, il faudra aussi quantifier les informations des blocs d'information, car tout doit être *in fine* sous forme binaire !

2.4.3 Quantification adaptative rétrograde – *online*

Dans cette second approche, l'adaptation se fait à la sortie du quantificateur. Elle ne nécessite pas de bloc d'information supplémentaire. Ici, seulement les valeurs quantifiés passées sont accessible pour adapter la quantification.

La méthode consiste à fixer le modèle, par exemple gaussien, et à adapter au cours du temps la valeur de Δ en observant les histogrammes issus de la quantification. Au bout d'un temps long, on aura de *bonnes* informations sur le signal, mais comment adapter la quantification avant ?

Évidemment, l'idée est de n'avoir ni un paramètre Δ trop petit, ni trop grand.

Nous allons présenter ici une idée de quantification adaptative qui n'utilise que la dernière valeur du signal passé. Il s'agit de la quantification de JAYANT. L'idée derrière cet algorithme est très simple : pour un nombre fixé d'intervalles N, on considère des multiplicateurs M_k pour chaque intervalle. On numérotera de 0 à $N/2-1$ les intervalles positifs, avec la relation

$$M_k = M_{k+N/2},$$

pour tout k entre 0 et $N/2-1$. On aura donc deux intervalles pour $k = N/2-1$ et $k = N-1$ qui seront de la forme $[b_k, +\infty[$ et $]-\infty, -b_k]$ et qui sont des intervalles en dehors de la zone de quantification.

Partant de là, si la valeur à quantifier est dans les intervalles en dehors de la zone de quantification alors, le paramètre Δ doit être augmenter (pour tenter de faire rentrer la valeur dans la zone à l'intérieur de la quantification). Au contraire, si la valeur à quantifier est à l'intérieur de la zone de quantification, alors on peut essayer de diminuer la valeur de Δ.

On aura alors que pour les intervalles *internes*, $M_k \leq 1$ et pour les deux intervalles *externes* $M_k > 1$. L'algorithme est alors :

— la n-ième valeur est dans l'intervalle k,
— le paramètre de quantification est mise à jour par $\Delta_n = M_k * \Delta_{n-1}$,
— et les intervalles $(I_k^n)_{k\in[0,N-1]}$ sont recalculés avec le nouveau Δ_n.

Exercice 2.2 :

Appliquer l'algorithme pour :
— $M_0 = M_4 = 0.8$, $M_1 = M_5 = 0.9$, $M_2 = M_6 = 1$ et $M_3 = M_7 = 1.2$;
— la valeur initiale du paramètre de quantification $\Delta_0 = 0.5$;
— la séquence à quantifier $\{0.1, -0.2, 0.2, 0.1, -0.3, 0.1, 0.2, 0.5, 0.9, 1.5\}$.

Le choix des multiplicateurs est un choix crucial pour cette algorithme. Il est assez libre, mais plusieurs méthodes ont été proposées pour le faire. Nous renvoyons à [10] pour une description d'une d'entre elles.

2.5 Quantification non-uniforme

La dernière solution envisagée habituellement consiste à postuler une densité de probabilité, à considérer ensuite σ_q comme une fonction des $2N + 1$ variables réelles contenues dans les variables Y et B et finalement à optimiser σ_q globalement. La taille des intervalles de quantification $\Delta_i = b_{i+1} - b_i$ n'est alors plus constant.

Nous présentons dans cette section un algorithme connu sous le nom d'algorithme de Lloyd ou de Lloyd-Max. Cette méthode a initialement été développée pour les problèmes de quantification (en 1957) mais a désormais des domaines d'application très variés notamment pour calculer le diagramme de Voronoï de k points, pour la résolution d'EDP, ou même désormais en *machine-learning*.

Un calcul de différentielle partielle de (2.2) conduit à l'algorithme suivant.

avec $b_0 = -\infty$ et $b_N = +\infty$. Celles-ci permettent de définir un algorithme de point fixe. En effet, le calcul des y_n dépend des b_n, et le calcul des b_n dépend des y_n. On peut montrer rigoureusement que l'algorithme converge.

La méthode est donc d'itérer le calcul des jeux de variables (y_n) et (b_n) jusqu'à convergence.

De même que ce qu'à la section 2.3.2, ce système n'est, dans le cas général, par résoluble algébriquement. Il faut donc mettre en oeuvre une méthode numérique pour le résoudre approximativement.

Une preuve de convergence de l'algorithme de Lloyd peut se lire ici [13], mais a été largement étendue depuis.

Algorithme 1 : Algorithme de LLOYD-MAX

Require : $b_1^0, \dots, b_{N-1}^0, y_1^0, \dots, y_N^0$ et f_S

1 : Pour tout i, $b_0^i = -\infty$, $b_N^i = +\infty$

2 : $i = 0$

3 : **repeat**

4 : Calculer :

$$y_n^{i+1} = \frac{\int_{b_{n-1}^i}^{b_n^i} x f_S(x)\mathrm{d}x}{\int_{b_{n-1}^i}^{b_n^i} f_S(x)\mathrm{d}x}, \quad \forall n \in \{1, \dots, N\}$$

$$b_n^{i+1} = \frac{y_{n+1}^{i+1} + y_n^{i+1}}{2}, \quad \forall n \in \{1, \dots, N-1\}$$

5 : **until** Convergence

2.6 NOTE BIBLIOGRAPHIQUE

Pour plus de détails sur les différentes techniques de quantification des signaux, on consultera le chapitre 9 de la référence [10].

CHAPITRE 3

Codage sans perte de l'information

SOMMAIRE DU CHAPITRE

Nous continuons de suivre le cheminement du signal. Après avoir été échantillonné puis quantifié, il est maintenant représenté par une suite de 0 et de 1. Nous allons voir comment on peut préparer sa transmission dans de bonnes conditions. Plus précisément, deux raffinements très utiles vont être présentés dans ce chapitre : tout d'abord ce qui est parfois appelé le *codage source*, *i.e.* une méthode de compression sans perte d'information du signal, comme le fait par exemple la compression *zip*, ensuite, ce qui est parfois appelé le *codage canal*, *i.e.* l'ajout judicieux de *bits de correction* qui vont permettre de corriger les erreurs éventuelles qui vont affecter le signal au cours de sa transmission. Nous regarderons un codage particulier appelé le code de Hamming qui permet

de détecter deux erreurs par mot transmis et d'en corriger une.

3.1 CODAGE SOURCE ET COMPRESSION SANS PERTE

On appelle donc codage source l'ensemble des techniques permettant de compresser avec ou sans perte d'information un signal numérique. Ceci revient en fait à coder de manière astucieuse les symboles émis par une source de manière à réduire le nombre total de bits utilisés.

Dans cette section on démontre le théorème fondamental du codage source, qui indique une limite théorique de compression sans perte et on présente un algorithme, dû à Huffman [1], permettant d'approcher arbitrairement cette limite.

3.1.1 Définitions

On commence par poser deux définitions.

DÉFINITION 3.1 — (Source de symboles m-aire) On appelle source de symboles m-aire tout dispositif émettant des *mots* construits par concaténations de symboles pris dans un *alphabet* de taille m.

Par exemple en informatique, l'alphabet est constitué de deux symboles 0 et 1, il s'agit d'un alphabet 2-aire, ou encore *binaire*. L'anglais utilise quant à lui un peu plus de 26 symboles (car on peut ajouter aux 26 lettres de l'alphabet latin [2] quels autres symboles comme les signes de ponctuation).

On appelle *source sans mémoire* une source pour laquelle la probabilité d'émission d'un symbole ne dépend pas des symboles précédemment émis. Dans ce cours, on ne va considérer que des sources sans mémoire, mais la plupart des résultats présentés dans ce chapitre, et en particulier un nouveau théorème de Shannon, peuvent être étendus à des cas beaucoup plus généraux.

DÉFINITION 3.2 — (Extension d'ordre k) On appelle extension d'ordre k d'une source S, la source S_k dont l'alphabet est obtenu par concaténation de k symboles consécutif de la source S.

1. David Albert Huffman (9 août 1925 — 7 octobre 1999, Ohio), chercheur américain pionnier dans le domaine de l'informatique.

2. L'alphabet français lui comporte l'alphabet latin morderne de 26 lettres auxquelles il faut ajouter les lettres accentuées au nombre de 13, le c cédille et deux ligutares, e-dans-l'a et e-dans-l'o, pour un total de 42 !

Remarque 3.1 :

Notons que le débit de la source S_k est k fois moins élevé que celui de la source initiale S. De plus, si S est une source de symboles m-aire, l'alphabet de la source S_k sera de taille m^k.

Exemple

Soit $A = \{s_1, \dots, s_m\}$ un alphabet et S une source emettant dans cet alphabet, c'est-à-dire un enchaînement de s_i de l'alphabet, par exemple :

$$S = s_2 s_4 s_8 s_5 s_1 s_9 s_2 s_2 s_3 \dots$$

L'extension d'ordre 2 prendra ses valeurs dans l'alphabet $A_2 = \{(s_i s_j)_{i,j \in \{0,\dots,m\}}\}$ de taille 2^m et donc, sur l'exemple précédent, on considèrera

$$S_2 = (s_2 s_4)(s_8 s_5)(s_1 s_9)(s_2 s_2)(s_3 \dots$$

3.1.2 Entropie et mesure de la quantité d'information

Pour pouvoir compresser efficacement les symboles émis par une source, il est utile de savoir mesurer la quantité d'information apportée par les symboles qu'elle produit. Intuitivement, un symbole qui apparaît rarement, par exemple le *w* dans les textes en français, apporte plus d'information qu'un symbole très fréquent, par exemple le *e* toujours dans les textes en français [3].

Quantité d'information par symbole

Étant donné une source S émettant des symboles d'un alphabet A, on commence par calculer la probabilité d'apparition des symboles de A. Ce calcul peut-être fait *offline*, c'est-à-dire *a posteriori*, en calculant la fréquence d'apparition de chaque symbole si l'on dispose de l'ensemble du signal émis, ou bien *online*, c'est-à-dire *a priori*, si l'on connaît certaines propriétés de la source, par exemple si l'on sait que la source émet des mots dans une certaine langue écrite.

Notons h la quantité d'information — que nous n'avons pas encore définie — apportée par un symbole $x \in A$. Faisons une liste des propriétés de h attendues.

3. à l'exception de *La disparition* de G. Perec, un livre ne contenant pas — dans un but bien précis — la lettre *e*.

1. Tout d'abord, on souhaite que la quantité d'information apportée par l'émission d'un symbole x augmente lorsque la probabilité d'émission de x, notée $p(x)$ dans la suite, diminue, ce qui peut se modéliser par

$$h(x) = f\left(\frac{1}{p(x)}\right),$$

où f est un fonction *croissante*.

2. Si la probabilité d'émission d'un symbole est 1, alors celui-ci n'apporte aucune information, on peut alors imposer

$$f(1) = 0.$$

3. On souhaite enfin que la quantité d'information apportée par deux messages *indépendants* soit la somme des quantités d'information apportées par chacun des messages. On peut alors demander à f de vérifier

$$f\left(\frac{1}{p(x \text{ et } y)}\right) = f\left(\frac{1}{p(x) \cdot p(y)}\right) = f\left(\frac{1}{p(x)}\right) + f\left(\frac{1}{p(y)}\right).$$

Ces différentes propriétés impliquent que la fonction h doit être choisie de la forme :

$$h(x) = -\log_q(p(x)).$$

La base q n'est pas fixée *a priori*. Nous verrons plus tard que on la choisira en fonction de la « dimension » de codage que on va utiliser.

Entropie d'une source

On souhaite maintenant définir la quantité moyenne d'information apportée par la source S. Cette quantité est appelée *entropie*[4] et est naturellement définie par la formule :

$$H(S) \stackrel{\text{def}}{=} \sum_{x \in A} p(x)h(x) = -\sum_{i=1}^{m} p_i \log_q(p_i),$$

où l'on a noté p_i la probabilité d'émission du i-ème symbole de l'alphabet A.

4. L'entropie est une notion très variable suivant les disciplines mais s'il existe des liens entre les différentes formulation. Ici, nous ne parlerons que de l'entropie de Shannon : https://fr.wikipedia.org/wiki/Entropie_de_Shannon.

Borne sur l'entropie

On peut montrer que l'entropie est bornée supérieurement. Pour ce faire, on introduit le résultat préliminaire suivant.

LEMME 3.1 — (Inégalité de Gibbs) Soit n nombres p_i et q_i tels que :

$$0 < p_i < 1, \qquad 0 < q_i < 1,$$

et

$$\sum_{i=1}^{n} p_i = 1, \qquad \sum_{i=1}^{n} q_i \leq 1.$$

Alors :

$$\sum_{i=1}^{n} p_i \log_q \left(\frac{q_i}{p_i}\right) \leq 0,$$

avec égalité si et seulement si

$$\forall i,\ 0 \leq i \leq n, \qquad p_i = q_i.$$

Exercice 3.1 :

Montrer ce lemme pour le logarithme népérien.

Cette inégalité est également appelée *inégalité de Jensen* [5] dans d'autres contextes.

Proposition 3.1 : *L'entropie vérifie l'inégalité suivante*

$$H(S) \leq \log_q(m),$$

avec égalité dans le cas où tous les symboles de S sont équiprobables, c'est-à-dire que pour tout $i \in \{1, \ldots, m\}$, $p_i = 1/m$.

5. Johan Ludwig William Valdemar Jensen, surtout connu comme Johan Jensen, (8 mai 1859, Nakskov, Danemark - 5 mars 1925, Copenhague, Danemark) était un mathématicien et ingénieur danois. Il est surtout connu pour sa fameuse inégalité de Jensen. En 1915, il démontra également la formule de Jensen en analyse complexe.

Exercice 3.2 :

Montrer cette proposition.

3.1.3 Propriétés d'un codage source

L'objectif de la compression est de construire un codage minimisant la taille moyenne des nouveaux symboles obtenus :

$$\bar{\ell} = \sum_{i=1}^{m} n_i p_i,$$

où n_i est la taille du nouveau code du i-ème symbole de A.

Supposons que la source S (après quantification) soit constituée de m états, c'est-à-dire, prenant des valeur dans l'alphabet $A = \{s_1, \dots, s_m\}$. À chacun de ces états s_i, on va associer une suite de n_i symboles d'un nouvel alphabet q-aire. Ainsi le code source sera une suite de concaténation de n_i symboles d'un alphabet $C = \{c_1, \dots, c_q\}$ (un mot) correspondant à s_i, pour tout $i \in \{1, \dots, m\}$.

Comme ici, on considère deux alphabets, on appellera les symboles de l'alphabet constituant la source S qu'on cherche à coder les *états*.

DÉFINITION 3.3 — (Déchiffrabilité) Un code est appelé un code *déchiffrable*, ou encore à décodage unique, si toute suite de mots de code ne peut être interprétée que de manière unique.

Il y a plusieurs manières d'avoir un code déchiffrable parmi lesquels :
- définir un code à longueur fixe, c'est-à-dire avec des mots de longueur fixe, qu'on peut décoder sans ambiguïté (par exemple : le code ASCII) ;
- consacrer un symbole de l'alphabet de destination comme séparateur de mot ;
- on évite qu'un mot du code soit identique au début d'un autre mot.

C'est à cette dernière proporiété que nous allons nous intéresser dans ce cours. On va alors ajouter une autre contrainte sur le nouvel alphabet. On souhaite que celui-ci soit *auto-ponctué*, c'est-à-dire que les nouveaux symboles soient émis par simple concaténation. On parle aussi de code *préfixe* ou code *irréductible*. Ceci est formalisé par la définition suivante.

DÉFINITION 3.4 — (Code irréductible) On appelle code *irréductible*, ou code *préfixe*, ou code *auto-ponctué*, un ensemble de mots tel qu'aucun mot ne soit le début d'un autre.

De la sorte, le récepteur d'un message sait toujours comment découper la suite de symboles qu'il reçoit, car il sait identifier les moments où l'on passe d'un symbole à un autre.

Tous les codes utilisés dans la vie courante ne sont pas irréductibles. Les langues écrites ne le sont pas en général, le code morse non plus par exemple.

Arbre $q-$aire

Les codes auto-ponctués sont de manière naturelle associés à la notion d'arbre q-aire, que l'on introduit maintenant.

DÉFINITION 3.5 — (Arbre q-aire) On appelle arbre q-aire un arbre dont chaque nœud est soit une feuille, soit possède au plus q descendants.

Par exemple un arbre binaire est obtenu en considérant l'arbre généalogique des ascendants d'une personne (et en se fixant une limite dans le temps). La notion d'arbre q-aire est le bon concept pour représenter les codes auto-ponctués. Le nombre q représente le nombre de symboles du nouvel alphabet de codage utilisés. Les mots sont donnés par l'ensemble des feuilles. Le découpage du message reçu se fait donc par parcours successifs de l'arbre, avec retour à la racine à chaque fois que l'on obtient une feuille.

Si l'alphabet est un alphabet binaire (composé de `0` et de `1`) alors, l'arbre correspondant est un arbre binaire très utile en algorithme et pour lesquels il existe beaucoup de résultats théoriques et de bibliothèques pour nombreux langages de programmation.

Les arbres q-aires vérifient l'inégalité de Kraft [6] qui nous servira dans la suite.

6. Cette inégalité fut publiée pas Leon Kraft en 1949. Dans l'article correspondant Kraft ne considère que les codes auto-ponctués et attribue l'analyse qu'il présente pour obtenir son résultat à Raymond M. Redheffer. Cette inégalité est aussi parfois appelée « Théorème de Kraft-McMillan » après que Brockway McMillan l'a découverte indépendamment en 1956. McMillan prouve le résultat pour une classe plus large de codes et attribue quant à lui la version correspondant aux codes auto-ponctués de Kraft à des observations de Joseph Leo Doob en 1955.

LEMME 3.2 — (Inégalité de Kraft et réciproque) Les longueurs $(n_i)_{1\leq i\leq m}$ des m chemins allant vers les m feuilles d'un arbre q-aire vérifient :

$$\sum_{i=1}^{m} q^{-n_i} \leq 1. \qquad \text{(inégalité de Kraft)}$$

Réciproquement, si l'on se donne un entier q et m entiers $(n_i)_{1\leq i\leq m}$ vérifiant l'inégalité de Kraft, alors on peut construire un arbre q-aire ayant m feuilles situées à des profondeurs $(n_i)_{1\leq i\leq m}$.

Exercice 3.3 :

Montrer cette inégalité pour les code irréductibles.

Théorème fondamental du codage source

On dispose maintenant de tous les outils pour énoncer et démontrer le théorème fondamental du codage source, qui est également un théorème de Shannon.

THÉORÈME 3.1 — (Théorème fondamental du codage source) Dans le codage d'une source S auto-ponctué sans mémoire à l'aide d'un alphabet de taille q, la longueur moyenne $\bar{\ell}$ des mots du code utilisé pour coder un symbole de S vérifie :

$$H(S) \leq \bar{\ell},$$

et l'on peut approcher cette limite aussi près que l'on veut, quitte à coder les extensions de S au lieu de S elle-même.

Preuve : On considère m états de la source S codés par des mots de longueurs $(n_i)_{1\leq i\leq m}$ dans un alphabet de taille q. Puisqu'on considère un code auto-ponctué, ces mots sont obtenus comme les feuilles d'un arbre q-aire et leurs longueurs vérifient :

$$Q \stackrel{\text{def}}{=} \sum_{i=1}^{m} q^{-n_i} \leq 1.$$

D'après l'inégalité de Gibbs, on a également :

$$\sum_{i=1}^{m} p_i \log_q \left(\frac{q^{-n_i}}{p_i}\right) \leq 0,$$

où p_i est la probabilité d'émission du i-ème symbole de S avec n_i la longueur du nouveau code du i-ème symbole. On en déduit :

$$-\sum_{i=1}^{m} p_i \log_q p_i \leq \log_q q \times \sum_{i=1}^{m} p_i n_i = 1 \times \sum_{i=1}^{m} p_i n_i.$$

Or :

— $H(S) = -\sum_{i=1}^{m} p_i \log_q p_i$,
— $\sum_{i=1}^{m} p_i = 1$,
— $\sum_{i=1}^{m} p_i n_i = \bar{\ell}$, la longueur moyenne d'un mot du nouveau code.

Finalement, on trouve :

$$H(S) \leq \bar{\ell}.$$

Nous avons donc obtenu la borne théorique de compression annoncée dans l'introduction de cette section.

Montrons maintenant qu'on peut s'approcher aussi près que l'on veut de cette borne. On cherche à minimiser $\bar{\ell}$. Pour ce faire, on peut essayer d'approcher le cas d'égalité dans les inégalités de Kraft et de Gibbs utilisées, c'est-à-dire à avoir :

$$Q = 1,\ q^{-n_i} = p_i,$$

qui se traduit pour n_i par :

$$n_i = -\log_q p_i.$$

Mais cette dernière fraction n'est pas forcément égale à un entier, on choisit donc de définir n_i par :

$$-\log_q p_i \leq n_i \leq -\log_q p_i + 1.$$

En conséquence, on a :

$$\sum_{i=1}^{m} q^{-n_i} \leq \sum_{i=1}^{m} p_i = 1,$$

et l'inégalité de Kraft est vérifiée. Il existe donc un arbre q-aire correspondant aux longueurs n_i, dont on peut déduire les mots (de longueurs n_i) du nouveau codage. De plus on a :

$$-p_i \log_q p_i \leq p_i n_i \leq -p_i \log_q p_i + p_i,$$

d'où l'on déduit par sommation :

$$H(S) \leq \bar{\ell} \leq H(S) + 1.$$

Pour approcher arbitrairement la limite théorique, une stratégie judicieuse est de considérer non plus la source S comme source initiale, mais son extension d'ordre k, S_k. Si on considère les symboles comme des variables aléatoires prenant des valeurs dans $A = \{s_1, \dots, s_m\}$, alors

$$\bar{\ell}_k = \sum_{x_1,\dots,x_k} p(x_1, \dots, x_k) n(x_1, \dots, x_k)$$

où la somme est sur toutes les combinaisons possibles, et p est la probabilité d'avoir la combinaison et n la longueur de codage de cette suite de symboles. Pour cette source là, on applique le théorème que l'on vient d'obtenir, et on obtient :

$$H(S_k) \leq \bar{\ell}_k \leq H(S_k) + 1,$$

où $\bar{\ell}_k$ représente la longueur moyenne des mots du nouveau codage de S_k. On divise alors ces inégalités par k :

$$\frac{H(S_k)}{k} \leq \frac{\bar{\ell}_k}{k} \leq \frac{H(S_k)}{k} + \frac{1}{k}.$$

La quantité $\frac{\bar{\ell}_k}{k}$ est alors notre longueur moyenne pour la source S et notons $\mathcal{H} = \frac{H(S_k)}{k}$. On a alors

$$\begin{aligned}\mathcal{H} &= \frac{1}{k} \sum_{x_1,\dots,x_k} p(x_1,\dots,x_k) \log_q\left(\frac{1}{p(x_1,\dots,x_k)}\right) \\ &= \frac{1}{k}\mathbf{E}\left[\log_q\left(\frac{1}{p(X_1,\dots,X_k)}\right)\right]\end{aligned}$$

où on a noté X_k les variables aléatoires du processus et $\mathbf{E}(X) = \sum_i x_i p_i$. Dans ce cours, on ne considère que des sources sans mémoire, et donc les symboles sont indépendants, ainsi $p(x_1,\dots,x_k) = p(x_1)p(x_2)\cdots p(x_k)$ et donc

$$\begin{aligned}\mathcal{H} &= \frac{1}{k}\mathbf{E}\left[\log_q\left(\prod_{i=1}^{k}\frac{1}{p(X_i)}\right)\right] \\ &= \frac{1}{k}\sum_{i=1}^{k}\mathbf{E}\left[\log_q\left(\frac{1}{p(X_i)}\right)\right] \\ &= \frac{1}{k}\sum_{i=1}^{k} H \\ &= H.\end{aligned}$$

■

Remarque 3.2 :

Nous avons démontrer ce résultat dans le cas d'une source sans mémoire, mais on peut aussi montrer que $\mathcal{H} \leq H$ dans le cas d'une source avec mémoire.

Remarque 3.3 :

Évidemment, approcher cette limite a un certain coût ! En effet, la technique de codage considérée implique à un moment ou à un autre la transmission d'un dictionnaire permettant le décodage pour retrouver les symboles émis initialement. Si l'on code les extensions de S au lieu de S elle-même, la taille du dictionnaire va augmenter et ce exponentiellement par rapport à l'extension considérée.

3.1.4 Algorithme de Huffman

Problème d'optimisation *

On cherche à construire un code préfixé de longueur moyenne minimale. La compression est d'autant plus forte que la longueur moyenne des mots de code est faible. Trouver un tel code revient donc à choisir les longueurs des mots de codes, par rapport à la distribution de probabilité des symboles de source, afin de rendre la longueur moyenne minimale. Il faut donc minimiser la longueur moyenne du code $\bar{\ell}$ sous les conditions de l'inégalité de Kraft (qui donne une condition nécessaire et suffisante pour qu'un code possède un code préfixé équivalent). Mathématiquement cela revient à résoudre le problème de minimisation de la longueur moyenne

$$\bar{\ell} = \sum_{i=1}^{m} p_i n_i$$

sur l'ensemble des longueurs $\{n_i\}_{i\in\{1,\dots,m\}}$ et sous la contrainte donnée par l'inégalité de Kraft :

$$\sum_{i=1}^{m} q^{-n_i} \leq 1.$$

Par la méthode des *multiplicateurs de Lagrange*, on définit le lagrangien J :

$$J = \sum_{i=1}^{m} p_i n_i + \lambda \left(\sum_{i=1}^{m} q^{-n_i} - 1 \right)$$

que l'on différentie par rapport aux n_i. Un rapide calcul donne les longueurs optimales $n_i^* = -\log_2(p_i)$, c'est-à-dire une longueur moyenne égale à l'entropie. Bien évidemment, ces longueurs optimales ne sont pas des longueurs entières. Le but de cette section est de présenter un algorithme qui permet d'approcher cette limite.

L'algorithme

Il nous reste à donner une méthode permettant la mise en pratique du résultat théorème 3.1 (qui n'est pas constructif). C'est l'algorithme 2, dit de Huffman, qui fournit ce résultat à partir d'une source de m symboles, dont on connaît les probabilités $(p_i)_{1\leq i\leq m}$ d'émission. En pratique cet algorithme explique comment construire un *arbre de codage*.

Algorithme 2 : Algorithme de Huffman d'un code q-aire de m symboles

Require : $(s_i)_{i\in\{1,\dots,m\}}$ symboles de distribution de probabilité $(p_i)_{i\in\{1,\dots,m\}}$

1 : Définir m nœuds actifs correspondant aux m symbole
2 : Calculer r comme le reste de la division de $1-m$ par $q-1$ *(si on a un code binaire, alors $r=0$)*
3 : **while** il reste plus d'un nœud actif **do**
4 : Grouper, un tant que fils d'un nœud nouvellement créé, les $q-r$ nœuds actifs les moins probables et les r nœuds inutilisés
5 : Marquer les $q-r$ nœuds actifs comme *non-actifs* et le nœud nouvellement crée comme *actif*
6 : Assigner au nœud nouvellement créé une probabilité égale à la somme des probabilités des $q-r$ nœuds qui viennent d'être désactivés.
7 : Poser $r=0$
8 : **end while**

Soit T un arbre binaire de n feuilles dont les poids respectifs sont notés p_i et les profondeurs ℓ_i. Montrons que l'algorithme de Huffman minimise la longueur moyenne $\sum_i \ell_i p_i$ dans le cas d'un arbre binaire.

DÉFINITION 3.6 — (Arbre optimal) On appelera un arbre T de n feuilles aux poids et profondeurs respectives p_i et ℓ_i un *arbre optimal*, un arbre dont le placement des feuilles minimise la quantité $\sum_{i=1}^{n} \ell_i p_i$.

Pour cela, établissons le lemme suivant.

LEMME 3.3 Soit T un arbre optimal, alors l'arbre T n'a aucun nœud qui n'a qu'un seul descendant.

Preuve : Supposons qu'un nœud de l'arbre T a un descendant vide, son descendant de gauche par exemple. En supprimant ce nœud et en recollant à la place le fils droit (non

vide), alors on obtient un arbre T' dont la longueur est strictement inférieur. Impossible puisque l'arbre T est optimal. ■

Nous avons besoin d'un second lemme.

LEMME 3.4 Soit x et y deux symboles de l'alphabet à coder de plus petit poids (probabilités). Il y a un abre optimal pour lequel ces deux symboles sont des feuilles sœurs d'un même nœud de profondeur maximale.

Preuve : Soit T un arbre optimal. Soient b et c deux feuilles sœurs de profondeur maximale (deux telles feuilles existent car l'arbre est optimal). Supposons par exemple que $p_b \leqslant p_c$ et $p_x \leqslant p_y$. Comme x et y ont les deux plus petits poids, $p_x \leqslant p_b$ et $p_y \leqslant p_c$. De plus b et c sont à une profondeur maximal dans l'arbre T, donc $\ell_x \leqslant \ell_b$ et $\ell_y \leqslant \ell_c$. Échangeons x et b. En notant $B(T) = \sum_{\alpha \in A} p_\alpha \ell_\alpha$, on obtient un noudel arbre T' tel que :

$$\begin{aligned} B(T') &= B(T) - p_x\ell_x - p_b\ell_b + p_x\ell_b + p_b\ell_x \\ &= B(T) - (p_b - p_x)(\ell_b - \ell_x) \\ &\leqslant B(T). \end{aligned}$$

Comme l'arbre T est optimal, $B(T') = B(T)$, ainsi T' est aussi optimal. On fait de même en échangeant c et y et on obtiont un nouvel arbre T'', toujours optimal, qui vérifie la condition voulue. ■

On peut alors établir le théorème suivant.

THÉORÈME 3.2 Un arbre construit avec l'algorithme de Huffman minimise la longueur moyenne $\sum_i \ell_i p_i$.

Preuve : On démontre ce résultat par récurrence.

Pour le cas $n = 2$, il n'y a que deux arbres possibles, tout deux avec la même longueur moyenne.

Ensuite, considéront un arbre de Huffman T avec $n \geqslant 3$ feuille. Supposons donc la propriété d'optimalité vraie pour les alphabets ayant $n-1$ symboles.

Considérons un alphabet A de taille n. Les $n-1$ dernières itérations de l'algorithme de Huffman ne sont autres que l'algorithme de Huffmann appliqub à l'alphabet $A' = A \cup \{z\} \backslash \{x, y\}$ où $p_z = p_x + p_y$ et les poids des autres symboles sont les même pour A et A'. Par l'hypothèse de récurrence, l'algorithme de Huffman renvoie un arbre H optimal pour A'. Il nous reste à voir que cet arbre est optimal pour l'alphabet A.

On a alors :

$$B_A(H) = B_{A'}(H) - p_x\ell_x - p_y\ell_y + p_z\ell_z.$$

Mais $\ell_z = \ell_x - 1 = \ell_y - 1$. Ainsi $B_A(H) = B_{A'}(H) - (p_x + p_y)$.

Soit maintenant T un arbre optimal pour l'alphabet A avec les symboles x et y sur des feuilles sœurs de profondeurs maximales. T peut aussi être vu comme un arbre sur l'alphabet A'. Or H est optimal pour A'. Donc $B_{A'}(T) \geqslant B_{A'}(H)$. Mais $B_{A'}(T) = B_A(T) - (p_x + p_y)$ (même calculs que ceux faits pour H), donc

$$B_A(T) - (p_x + p_y) \geqslant B_{A'}(H) = B_A(H) - (p_x + p_y).$$

On en déduit que $B_A(T) \geqslant B_A(H)$ et donc que $B_A(T) = B_A(H)$ (puisque T est optimal), et donc H est optimal pour l'alphabet A. ■

Exercice 3.4 :

On considère une source S composée des états $\{s_1, s_2, \dots, s_8\}$ avec la distribution de fréquence suivante que l'on veut coder *en binaire* :

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
0.4	0.18	0.1	0.1	0.07	0.06	0.05	0.04

1. Calculer l'entropie de la source d'information S pour un codage en base 2.
2. Calculer un codage de Huffman binaire de cette source puis la longueur moyenne de ce codage. Comparer avec la longueur moyenne à l'entropie.

Remarque 3.4 :

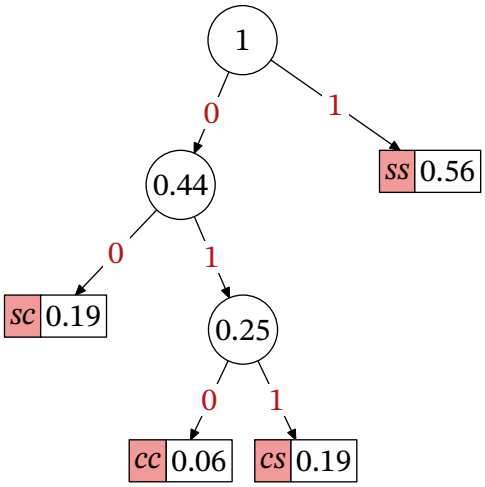
En ayant calculé l'entropie de la source S, on peut établir une prescription d'un objectif pour la taille moyen du code $\bar{\ell}$. Si l'objectif de longueur moyenne n'est pas atteint, alors il suffit de reprendre l'algorithme avec une extension d'ordre plus élevée du code.

Exemple : Réduction de la longueur moyenne par extension

Considérons un code simpliste : soit une source S prenant valeur dans les états $C = (c, s)$ de probabilité d'émission $P = (p_c, p_s) = (1/4, 3/4)$. Le codage de Huffman est alors juste un arbre à un nœud et deux feuilles qui donne $c \to 0$ et $s \to 1$ de taille respective $n_c = n_s = 1$, de longueur moyenne

$$\bar{\ell} = 1 \times 1/4 + 1 \times 3/4 = 1.$$

Considérons désormais l'extension d'ordre 2, c'est-à-dire S^2 d'alphabet $C^2 = (cc, cs, sc, ss)$ et d'émission de probabilité $P^2 = (1/16, 3/16, 3/16, 9/16)$. Cette extension d'ordre 2 donne l'arbre suivant.



$cc \to 010$, $cs \to 011$, $sc \to 00$ et $ss \to 1$.

La longueur moyenne est alors de

$$\bar{\ell}_2 = 3 \times 1/16 + 3 \times 1/16 + 2 \times 3/16 + 9/16 = 1.6875,$$

ce qui donne

$$\bar{\ell}^* = 0.5 \times \bar{\ell}_2 < \bar{\ell}.$$

3.2 CODAGE CANAL ET CORRECTION D'ERREUR

On considère maintenant uniquement des signaux binaires même si de nombreuses définitions et résultats présentés ici peuvent être généralisés. Une fois le signal — c'est-à-dire une suite de 0 et de 1 — compressé, il va maintenant s'agir de le transmettre. Le medium utilisé pour la transmission est appelé *canal*. Évidemment, les canaux peuvent être de nature très variée. Il peut s'agir d'une onde électromagnétique envoyée dans une fibre optique, dans l'atmosphère ou dans l'espace dans le cas de la communication par satellite, d'impulsions électriques envoyées dans un réseau, ou encore d'une clef USB, d'un disque dur ou d'un disque optique pour les CD ou pour les DVD.

Chacun de ces canaux rend intrinsèquement possible l'introduction d'erreurs. Il y a concrètement toujours un risque que le médium transforme un 0 en 1 et réciproquement. Cela peut ne pas avoir d'effet grave si le code supporte une image ou un son, mais peut être beaucoup plus lourd de conséquence s'il s'agît du texte d'un programme exécutable par exemple. Dans ce cas, la transmission échouera si le moindre symbole du programme est modifié.

Il apparaît donc nécessaire de mettre en place une stratégie permettant au moins de détecter, lors de la réception, les erreurs de transmission, voir de les corriger. De nombreuses méthodes atteignant cet objectif ont été conçues depuis une cinquantaine d'années. Elles reposent toutes sur l'ajout de *bits de correction*, ou *bits de contrôle*, c'est-à-dire qu'elles ont toujours pour effet négatif de grossir la taille du signal à envoyer. Un compromis doit donc être trouvé entre qualité de correction et alourdissement du signal.

Dans cette section, on présente des outils permettant de quantifier le facteurs intervenant dans ce compromis et une stratégie générale de codage canal ainsi qu'une de ses réalisations concrètes.

Dans la suite on désignera indifféremment de *code correcteur*, *codage canal* ou encore *code* la stratégie de correction considérée.

3.2.1 Une approche naïve

Une idée simple que l'on peut avoir est de répéter un certain nombre de fois chaque symbole à transmettre. Si on choisit par exemple de doubler le symbole, c'est-à-dire d'appliquer la fonction :

$$0 \to 00,$$

$$1 \to 11,$$

on pourra facilement détecter une erreur (si les symboles reçus ne coïncident pas deux à deux). Par contre, on ne pourra corriger l'erreur correctement qu'une fois sur deux en moyenne, en choisissant arbitrairement de remplacer la séquence erronée détectée par un 0, par exemple.

Dans cette démarche, on a ajouté un bit de correction par bit transmis. Si on choisit maintenant d'en ajouter deux, en adoptant une stratégie de triplement, on constate que la correction sera plus efficace : on remplacera une séquence erronée, par exemple 001, par le signe apparaissant majoritairement [7] dans le triplet, 0 dans notre exemple. Cette méthode permet de détecter deux erreurs par bloc de trois bits et d'en corriger une : en effet, si deux erreurs affectent, lors de la transmission la séquence de trois bits, alors la stratégie choisie donnera lieu,

7. On parle parfois de *stratégie du vote majoritaire*.

lors du décodage, à une erreur. Pour aller plus loin, on peut regarder l'exemple d'un canal introduisant 5% d'erreurs, avec lequel on applique la stratégie du triplement. Puisqu'une erreur est corrigée, la probabilité qu'une séquence de trois bits soit transmise correctement ou avec une seule erreur est de :

$$p = 0,95^3 + 3 \times 0,95^2 \times 0,005 = 0,99275.$$

Ainsi le taux de succès est passé de 95% à un taux supérieur 99%. Évidemment, ceci a un coût ! Il a en effet fallu tripler la taille du signal à transmettre.

On va voir dans la suite comment « optimiser » l'usage des bits de correction.

3.2.2 Codes linéaires par blocs

Découpage en bloc et structure algébrique des blocs

Un préalable souvent nécessaire à la correction d'erreur est le découpage du signal à transmettre en blocs de taille fixe, disons k. L'ensemble sur lequel on va travailler est donc B^k où $B = \{0, 1\}$. En munissant celui-ci de l'addition composante par composante dans le groupe $(\mathbf{Z}/2\mathbf{Z}, +)$, ainsi que du corps des scalaires (fini) $(\mathbf{Z}/2\mathbf{Z}, +, \times)$, on voit que l'on peut doter B^k d'une structure d'espace vectoriel (fini) que l'on notera $\mathbf{F}_2^k$.

Remarque 3.5 :

— Le plus petit corps fini est noté $\mathbf{F}_2$. Il est composé de deux éléments distincts, 0 qui est l'élément neutre pour l'addition, et 1 qui est élément neutre pour la multiplication. Ceci détermine les tables de ces deux opérations en dehors de $1 + 1$ qui ne peut alors être que 0, car 1 doit avoir un opposé. On vérifie alors qu'elles définissent bien un corps commutatif.

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

Le corps $\mathbf{F}_2$ peut s'interpréter diversement. C'est l'anneau $\mathbf{Z}/2\mathbf{Z}$, les entiers pris modulo 2, c'est-à-dire que 0 représente les entiers pairs, 1 les entiers impairs (c'est le reste de leur division par 2), et les opérations se déduisent de celles sur $\mathbf{Z}$.

C'est aussi l'ensemble des valeurs de vérité classiques, 0 pour le faux, et 1 pour le vrai. L'addition est le « ou exclusif » (xor), la multiplication le « et ».

— Une généralisation naturelle de $\mathbf{F}_2 = \mathbf{Z}/2\mathbf{Z}$ est, pour p premier, le corps $\mathbf{Z}/p\mathbf{Z}$ à p éléments, noté aussi $\mathbf{F}_p$. Pour que l'anneau $\mathbf{Z}/p\mathbf{Z}$

soit un corps, il faut et il suffit que p soit premier.

Théorie des codes correcteurs

Introduisons quelques définitions pour pouvoir travailler mathématiquement avec ces concepts.

DÉFINITION 3.7 — (Poids d'un mot) On définit le poids d'un mot m de $\mathbf{F}_2^k$ comme le nombre de composantes non nulles du mot. On le notera $w(m)$.

On définit alors une distance [8] sur cet espace vectoriel.

DÉFINITION 3.8 — (Distance de Hamming) La distance de Hamming entre deux mots m_1 et m_2 de $\mathbf{F}_2^k$ est le nombre de composantes distinctes de m_1 et m_2. On la note $d(m_1, m_2)$.

On peut montrer, mais nous ne le ferons pas ici, que cette distance de Hamming [9] est bien une distance au sens mathématique du terme.

Proposition 3.2 : *La* distance de Hamming *est une distance sur* $\mathbf{F}_2^k$.

On a par exemple :

$$d(111, 101) = 1.$$

Une propriété intéressante de cette distance est qu'elle vérifie :

$$d(m_1, m_2) = d(m_1 + m_2, 0_{\mathbf{F}_2^n}), \tag{3.1}$$

où :

$$0_{\mathbf{F}_2^n} = (\underbrace{0, \dots, 0}_{n \text{ fois}}).$$

8. On appelle distance sur un ensemble E toute application d définie sur le produit $E^2 = E \times E$ et valeur dans $\mathbf{R}^+$ vérifiant les propriétés suivantes :

(symétrie) $\forall (a, b) \in E^2,\ d(a, b) = d(b, a)$,

(séparation) $\forall (a, b) \in E^2,\ d(a, b) = 0 \Leftrightarrow a = b$,

(inégalité triangulaire) $\forall (a, b, c) \in E^3, d(a, c) \leqslant d(a, b) + d(b, c)$.

Un ensemble muni d'une distance est un *espace métrique*.

9. Richard Wesley Hamming (11 février 1915, Chicago - 7 janvier 1998, Monterey, Californie) est un mathématicien américain surtout connu pour son algorithme de correction d'erreur, le Code de Hamming. Il travailla avec Claude Shannon entre 1946 et 1976 aux laboratoires Bell.

Autrement dit, pour connaître la distance entre deux mots, il suffit de les additionner et de compter le nombre de composantes égales à 1 dans le résultat. Derrière cette notion de distance, se cache une stratégie de correction : étant donné que les mots du code ne forment qu'une sous-partie de $\mathbf{F}_2^n$, lorsqu'on reçoit en sortie de canal un mot ne figurant pas dans l'ensemble des mots possible, on le remplace par le mot du code le plus proche.

Code

Dans le cadre que l'on vient d'introduire, l'ajout de r bits de correction peut être vu comme l'application d'une certaine fonction *injective* :

$$g : \mathbf{F}_2^k \to \mathbf{F}_2^n,$$

où $n = k + r$. La stratégie de correction est dite *linéaire*, lorsque g est une application linéaire (entre les espaces $\mathbf{F}_2^k$ et $\mathbf{F}_2^n$). Si on note m un mot de taille k et m' son image par le codage, on a donc :

$$m' = mG,$$

où G est la transposée de la matrice de g de dimensions (n, k) et à coefficients dans $\mathbf{Z}/2\mathbf{Z}$.

Puisque $n > k$, l'image de g est un sous-espace vectoriel de $\mathbf{F}_2^n$.

DÉFINITION 3.9 — (Code linéaire) On appelle codage linéaire le sous-espace vectoriel Im(g) que l'on note $C(k, n)$. En tant que sous-espace de dimension k (g est injective), il est caractérisé par $n - k = r$ équations linéaires traduisant les dépendances entre les bits des blocs codés.

Exemple

Un exemple particulièrement simple est le bit de parité : on ajoute à la fin du message un bit correspondant à la parité de la somme des éléments du message. Si la somme est paire, alors le bit vaut 0 et 1 sinon. Pour un mot de longueur k, le code associé a donc pour paramètres $(k, k + 1)$.

DÉFINITION 3.10 — (Matrice génératrice) La matrice G est appelée matrice génératrice du code $C(k, n)$.

DÉFINITION 3.11 — (Distance minimale d'un code) On appelle distance minimale d'un code C (de correction linéaire par bloc) la plus petite distance entre deux mots de C :

$$d_C = \min\{d(m_1, m_2);\ m_1 \in C, m_2 \in C, m_1 \neq m_2\}.$$

Proposition 3.3 : *Pour un code linéaire C, on a*

$$d_C = \min\{w(m);\ m \in C, m \neq 0\}.$$

Exercice 3.5 :

Écrire la preuve.

Proposition 3.4 : *Soit C un code linéaire de paramètre (n, k, d_C), alors*

$$d_C \leq n - k + 1.$$

C'est la borne de Singleton.

Exercice 3.6 :

Écrire la preuve (indication : faire un raisonnement sur les dimensions des espaces).

Décodage par maximum de vraisemblance

Une classe de décodage importante est la classe des décodage à distance minimale que nous définissons maintenant.

DÉFINITION 3.12 — (Décodage à distance minimale) On dit qu'un code C utilise un *décodage à distance minimale* chaque fois que la décision de décodage, noté $\mathcal{D}$ consiste pour un mot reçu m^*, à choisir le ou un des

mots de code le plus proche

$$\mathcal{D}(m^*) = \arg\min_{m \in C} d(m, m^*).$$

Le lemme suivant donne une méthode simple pour calculer la distance d'un code.

LEMME 3.5 La distance d'un code est égale au nombre de 1 contenus dans le mot non nul du code qui en contient le moins.

Preuve : Ce lemme est une conséquence directe de la formule (3.1). En effet, puisque $C(k, n)$ est un espace vectoriel, on a :

$$C(k, n) - \{0_{\mathbf{F}_2^n}\} = \{m_1 + m_2;\ m_1 \neq m_2,\ m_1 \in C(k, n),\ m_2 \in C(k, n)\}.$$ ■

3.2.3 Détection et correction d'erreur

Si la distance d'un code est 1, un mot avec un bit erroné peut également être un mot du code. On ne pourra donc pas à coup sûr détecter 1 erreur. Si la distance est 2, on détectera à coup sûr une erreur, mais on ne pourra pas la corriger dans tous les cas. En effet, il se peut que le mot entaché d'une erreur soit équidistant de deux mots distincts du code, ce qui fait qu'on ne pourra choisir qu'arbitrairement et sans garantie le mot du code qui lui correspondait avant l'erreur. Enfin, si la distance est 3, alors, par un raisonnement analogue, on pourra détecter 2 erreurs et corriger correctement 1 erreur. Tout ceci se généralise dans le théorème suivant.

THÉORÈME 3.3 Un code par bloc de longueur n utilisant un décodage à distance minimale peut, pour toute paire d'entiers $t \in \{0, \dots, n\}$ et $s \in \{0, \dots, n - t\}$

— corriger les mots contenant t erreurs ou moins,
— détecter les mots contenant de $t + 1$ à $t + s$ erreurs,

si et seulement si

$$d_C > 2t + s.$$

Avant de prouver ce résultat, on introduira la formalisation suivante. Comme stipulé précédemment, la transmission à travers le canal peut donner lieu à

une erreur affectant certains bits du mot. Le mot m^* recueilli en sortie de canal sera donc de la forme :

$$m^* = m' + e,$$

où e est un vecteur de $\mathbf{F}_2^n$ comportant des 1 sur les composantes correspondant à celles affectées et des 0 ailleurs.

Preuve : (hors programme) On démontre ce théorème en démontrant la contraposée, c'est-à-dire : C ne peut corriger tous les schémas à t erreurs *ou* ne peut détecter tous les schémas à $t+1$,...,$t+s$ erreurs si et seulement si $d_C \leqslant 2t+s$.

Supposons tout d'abord que le code C ne peut pas corriger tous les schémas à t erreurs *ou* ne peut détecter tous les schémas à $t+1$,...,$t+s$ erreurs.

Dans un premier temps, si le code C ne peut pas corriger tous les schémas de moins de t erreurs (inclues), cela signifie qu'il existe au moins un mot de code m_1 et un schéma d'erreur e de poids inférieur à t tels que le décodage $\mathcal{D}(m_1+e)$ ne soit pas m_1. Notons m_2 le mot décodé au lieu de m_1, c'est-à-dire $m_2 = \mathcal{D}(m_1+e)$. Grâce à l'égalité triangulaire, on a

$$d(m_1, m_2) \leqslant d(m_1, m_1+e) + d(m_1+e, m_2).$$

De plus, on a $d(m_1, m_1+e) = w(e) \leqslant t$ et $d(m_1+e, m_2) \leqslant d(m_1+e, m_1)$ car le code utilise un décodage à distance minimale. Ainsi

$$d(m_1, m_2) \leqslant t + t \leqslant 2t+s,$$

et donc d_C qui est inférieur ou égale à $d(m_1, m_2)$ est aussi inférieure ou égale à $2t+s$.

Ensuite, si le code peut corriger tous les schémas de moins de t erreurs mais ne peut pas détecter tous les schémas à $t+1, \dots, t+s$ il existe au moins un mot de code m_1 et schéma d'erreur e de poids entre $t+1$ et $t+s$ qui ne sont pas détectés mais décodé en un autre mot du code $m_2 = \mathcal{D}(m_1+e)$. En introduisant l'erreur $e' = m_2 + (m_1+e)$, nous avons aussi $\mathcal{D}(m_2+e') = m_2$, c'est-à-dire que e' est une erreur qui est corrigée lorsqu'elle est appliquée à m_2. Comme $w(e') = d(m_2+e', m_2) = d(m_1+e, m_2) \leqslant d(m_1+e, m_1) = w(e)$ (puisque le décodage est à distance minimale), nous avons $w(e') \leqslant t+s$. Comme e' est une erreur qui est corrigée, alors $w(e') \leqslant t$. On conclut alors

$$d(m_1, m_2) \leqslant d(m_1, m_1+e) + d(m_1+e, m_2) \leqslant (t+s) + t = 2t+s,$$

et donc $d_C \leqslant 2t+s$.

Réciproquement, supposons que $d_C \leqslant 2t+s$. Il existe donc deux mots du code m_1 et m_2 tels que $d(m_1, m_2) \leqslant 2t+s$. Ceci signifie aussi que le poids du vecteur m_1+m_2 est aussi inférieur à $2t+s$. Tout vecteur m de poids inférieur ou égal à $2t+s$ peut s'écrire comme la somme de deux vecteurs e et f tels que $w(e) \leqslant t$ et $w(f) \leqslant t+s$: prendre les premières composantes de m jusqu'à $2t$ complétées par des zéros pour constituer e, et inversement, $2t$ composantes nulles complétées par le reste des composantes de

m pour f. Par exemple `011010` peut être écrit comme 010000 + 001010 pour $t = 1$ et $s = 1$.

Ainsi $m = m1 + m2 = e + f$, c'est-à-dire (puis que en binaire $+e = -e$), $m_1 + f = m_2 + e$. Ceci veut dire que deux mots de code distincts et deux schémas d'erreur seront décodés de la même façon. Ceci implique que (au moins) $m_1 + f$ n'est pas corrigé ($\mathcal{D}(m_1 + f) \neq m_1$) ou que $m_2 + e$ n'est pas détecté ($\mathcal{D}(m_2 + e) = m_1$).

Ainsi, tous les schémas de moins de t erreurs ne peuvent être corrigés, ou tous les schémas à $t + 1, \dots, t + s$ erreurs ne peuvent être détectées. ■

Pour illustrer ce résultat, considérons un code par bloc ayant une distance minimale de 8. Un tel code peut être utilisé pour l'un ou l'autre des points suivants :

— corriger tous les schémas d'erreur de moins que 3 (inclus) erreurs et détecter tous les schémas à 4 erreurs ($t = 3, s = 1$);
— corriger tous les schémas d'erreur de moins que 2 erreurs et détecter tous les schémas de 3 à 5 erreurs ($t = 2, s = 3$);
— corriger tous les schémas d'erreur de 1 erreur et détecter tous les schémas de 2 à 6 erreurs ($t = 1, s = 5$);
— détecter tous les schémas de moins que 7 (inclus) erreurs ($t = 0, s = 7$).

Un code par bloc ayant une distance minimale de 7 peut (l'un ou l'autre) :

— corriger tous les schémas d'erreur de moins que 3 (inclus) erreurs ($t = 3$, $s = 0$);
— corriger tous les schémas d'erreur de moins que 2 erreurs et détecter tous les schémas de 3 à 4 erreurs ($t = 2, s = 2$);
— corriger tous les schémas d'erreur de 1 erreur et détecter tous les schémas de 2 à 5 erreurs ($t = 1, s = 4$);
— détecter tous les schémas de moins que 6 (inclus) erreurs ($t = 0, s = 6$).

Du théorème précédent, on obtient les deux résultats suivants.

Corollaire 3.1 (Capacité maximale de détection d'erreurs) : *Un code par bloc C utilisant le décodage à distance minimale peut être utilisé pour détecter tous les schémas d'erreur de $d_C - 1$ erreurs, ou moins.*

Preuve : Utiliser $t = 0$ et $s = d_C - 1$ dans le théorème précédent. ■

Corollaire 3.2 (Capacité maximale de correction d'erreurs) : *Un code par bloc C utilisant le décodage à distance minimale peut-être utilisé pour corriger tous les schémas d'erreur de maximum $(d_C - 1)/2$ (division euclidienne) erreurs ou moins, mais ne peut pas corriger les schémas d'erreur de $1 + (d_C - 1)/2$ erreurs.*

Preuve : Utiliser $t = (d_C - 1)/2$ et $s = 0$, et le t est maximal car pour $t^* = 1 + (d_c - 1)/2$ on a $2t^* = 1 + d_C \geqslant d_C$. ■

3.2.4 Syndrôme et matrice de vérification

Notion de syndrome

On peut caractériser l'appartenance au code $C(k,n)$ grâce à la notion de matrice de vérification :

$$m' \in C(k,n) \Leftrightarrow Hm' = 0,$$

où H est une matrice de dimensions (r,n).

> **Remarque 3.6 :**
> La théorie de l'algèbre linéaire montre qu'une telle application existe, il suffit par exemple de considérer un projecteur [a] sur un sous-espace supplémentaire de $C(k,n)$ parallèlement à $C(k,n)$. Notons qu'un code linéaire donné pourrait avoir plusieurs matrices de vérification différentes : toute matrice dont les lignes sont une base de l'espace vectoriel orthogonal au code linéaire est une matrice de vérification pour ce code.

DÉFINITION 3.13 — (Matrice de vérification) La matrice H est appelée *matrice de vérification*, *matrice de contrôle* ou matrice de test du code linéaire $C(k,n)$

On a $C(k,n) = \mathrm{Ker}(H)$ et la relation :

$$Hm^* = He,$$

puisque $m \in \mathrm{Ker}(H)$.

DÉFINITION 3.14 — (Syndrome) Le vecteur $Hm^* = He$ est appelé *syndrome* de m^*.

a. Soient F un sous-espace vectoriel de E et G un supplémentaire de F dans E. N'importe quel vecteur x de E peut s'écrire d'une façon unique comme somme d'un vecteur de F et d'un vecteur de G : $\forall x \in E, \exists!(x', x'') \in F \times G, x = x' + x''$. La projection sur F parallèlement à G est alors l'application :

$$p : \begin{array}{ll} E = F \oplus G & \to E \\ x = x' + x'' & \mapsto x'. \end{array}$$

Remarque 3.7 :

— Si seul une erreur s'est produite sur la composante i de m', alors le syndrome sera égal à la i-ème colonne de H.
— On note ici Hm au lieu de mH^T car cela facilitera l'écriture du prochain théorème.
— Les définitions précédentes ne disent pas comment trouver en pratique les matrices de vérifications ce qui sera l'objet de ce qui suit.

Introduisons maintenant les matrices génératrices sous forme systématique.

DÉFINITION 3.15 — (Forme systématique) Une matrice génératrice G d'un code linéaire (k, n) est dite sous forme systématique si elle est de la forme

$$G = [I_k \; P],$$

où I_k est la matrice identité de taille k et P est une matrice $k \times (n-k)$ souvent appelée *matrice de parité*.

On peut montrer que lorsqu'elle existe pour un code donné, la matrice génératrice systématique est unique.

Exemple

Pour les messages binaires, le *bit de vérification de parité* est le bit qui correspond à la parité du message, c'est-à-dire à la somme (binaire) du message.

Le code par bit de parité consiste simplement à envoyer d'abord le message tel quel, suivi de son bit de parité. En termes de codes, ceci correspond au code binaire linéaire $(k, k+1)$ dont la matrice génératrice est

$$G = \begin{bmatrix} & 1 \\ I_k & \vdots \\ & 1 \end{bmatrix}$$

qui est sous forme systématique.

THÉORÈME 3.4 Pour un code linéaire $C(k,n)$ dont la matrice génératrice est sous forme systématique est

$$G = [I_k \; P]$$

la matrice

$$H = [-P^\top \; I_{n-k}]$$

est une matrice de vérification.

Preuve : Pour tout message $m \in \mathbf{F}_2^k$, le mot de code $z \in \mathbf{F}_2^n$ correspondant est

$$z = m \cdot G = m \cdot [I_k \; P],$$

c'est-à-dire

$$\begin{cases} (z_1, \dots, z_k) = m, \\ (z_{k+1}, \dots, z_n) = m \cdot P. \end{cases}$$

Ainsi

$$(z_{k+1}, \dots, z_n) = (z_1, \dots, z_k) \cdot P,$$

ce qui sous forme matricielle donne la matrice de contrôle H.

Réciproquement, on montre que tout code $z \in \mathbf{F}_2^n$ tel que $H \cdot z = 0$ vérifie

$$z = (z_1, \dots, z_k) \cdot G$$

et se trouve être un mot de code. ■

Remarque 3.8 :

On notera qu'il est fait mention de la matrice $-P^T$. Le signe $-$ est là car, même s'il n'a pas été défini sur l'espace $\mathbf{F}_2$, il provient de la preuve, et le résultat s'étend à des espaces vectoriels autre que $\mathbf{F}_2$. Pour notre cas, c'est-à-dire le cas binaire, le signe $-$ est comme le signe $+$, en effet $1-1 = 0 = 1+1$ en binaire, et $0-1 = 0+1 = 1$.

Exercice 3.7 :

Soit le code C dont la matrice génératrice est

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Trouver une matrice de vérification pour le code C.

Matrice de vérification et distance minimale

THÉORÈME 3.5 — (Matrice de vérification et distance minimale) Si H est une matrice de vérification pour un code linéaire $C(k,n)$ alors la distance minimale d_C de ce code est égale au plus petit nombre de colonnes linéairement dépendantes de H.

Preuve : Pour tout vecteur $z \in \mathbf{F}_2^n$, $H \cdot z$ est une combinaison linéaire de $w(z)$ colonnes de H. Par définition, $z \in C(k,n)$ si et seulement si $H \cdot z = 0$. Ainsi, si $z \in C(k,n)$, il existe $w(z)$ colonnes de H qui sont linéairement dépendantes. Réciproquement, si q colonnes de H sont linéairement dépendantes, il existe un mot du code de poids q.

Donc d_C est le nombre minimal de colonnes de H qui sont linéairement dépendantes en vertu de la proposition 3.3. ■

3.2.5 Codes de Hamming

Pour achever la description de la stratégie de correction, on aimerait être capable de construire les matrices G et H. C'est l'objet de cette section. On va prendre l'exemple de codes très répandus : les codes de Hamming.

L'idée est de disposer d'un code dont le syndrome indique directement la position de l'erreur, par exemple sous forme de son code binaire.

Choix des paramètres

On se place dans le cadre simple où *au plus une erreur peut affecter les mots (de longueur n) du code*. Il y a donc $n+1$ scenari possibles : soit il n'y a pas eu d'erreur pendant la transmission, soit une erreur s'est produite sur l'un des n bits. Or, d'après les dimensions des espaces considérés, 2^r syndromes possibles, puisque $He \in \mathbf{F}_2^r$. Si l'on veut pouvoir faire correspondre de manière sûre, c'est-à-dire par le biais d'une injection, les vecteurs syndromes observés et le scénario dont il est la conséquence, il faut nécessairement :

$$2^r \geqslant n+1.$$

On va ici se restreindre aux paramètres vérifiants l'égalité suivante :

$$2^r = n+1. \tag{3.2}$$

D'autre part, on a :

$$n = k+r. \tag{3.3}$$

Enfin, on prend $k \geqslant 1$, car on travaille sur des blocs de longueur non nulle ! En cherchant quels sont les k pour lesquels (3.2) et (3.3) ont des solutions (k,n) entières, on trouve par exemple :

n	k	r
3	1	2
7	4	3
15	11	4
31	26	5
63	57	6

Le premier exemple correspond à la stratégie de triplement, décrite à la section 3.2.1. Le second correspond à un code largement utilisé, souvent appelé $C(4,7)$, ce qui correspond aux notations utilisées dans cette section.

L'idée du code de Hamming est d'exploiter le fait que lorsqu'il n'y a qu'une erreur, le syndrôme nous donne exactement la position de l'erreur. En exploitant le fait que le produit He nous donne la i^e colonne de H, on définit le code de Hamming comme il suit.

DÉFINITION 3.16 — (Code de Hamming) Un code de Hamming est un code linéaire (k,n) binaire dont la matrice de vérification est

$$H_r = [b_r(1)^\top \; b_r(2)^\top \cdots b_r(n)^\top]$$

où $n = 2^r - 1$ et $k = 2^r - r - 1$, pour $r \geq 2$, et $b_r(i)$ est la représentation binaire de i sur r bits.

Par construction, ces matrices donnent bien un code linéaire et sont bien de rang plein puisqu'il est aisé de construire la matrice indentité I_{n-k} à partir des colonnes. La dimension du noyau est alors k.

Pour construire la matrice de génération du code, il suffirait par exemple de mettre la matrice H sous forme systématique pour extraire la matrice P et construire la matrice G. Cependant, cela nous ferait perdre la propriété que le syndrôme donne la position de l'erreur à corriger. Nous n'allons donc pas suivre cette stratégie.

La matrice de contrôle définit totalement la géométrie du code, il suffit donc, pour terminer l'implémentation de trouver une matrice génératrice G. L'application linéaire associée doit vérifier deux conditions : elle est injective, et son image est le noyau de H. Il suffit donc de trouver une matrice de rang k ($n = k + r$) tel que $H \cdot G^T = 0$.

THÉORÈME 3.6 — (Distance minimum du code de Hamming) La distance minimale des codes de Hamming est toujours 3. Ainsi de tels codes peuvent corriger tous les schémas à une erreur.

Preuve : Calculons la distance minimale. Les matrices de vérification n'ont jamais de colonne nulle ni deux fois la même colonne. De plus, les trois premières colonnes (représentation binaires de 1, 2, et 3) sont toujours linéairement dépendantes. La distance minimale est donc toujours 3. ■

Exemple

Soit la matrice de syndrôme du code de Hamming $C(4,7)$:

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Pour trouver une matrice génératrice, nous cherchons 4 vecteurs linéairement indépendant z_i tels que $Hz_i^\top = 0$, par exemple :

$$\begin{aligned} z_1 &= 1110000 \\ z_2 &= 1101001 \\ z_3 &= 1000011 \\ z_4 &= 1111111 \end{aligned}$$

ce qui donne

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

Supposons maintenant devoir envoyer le message $m = 1001$. Il est codé par G en $z = 0001111$. Faisons maintenant l'hypothèse qu'une erreur soit survenue sur le troisième bit, et donc que $\hat{z} = 0011111$ soit reçu.

Le syndrome d'un tel code reçu sera alors $s = 011$, c'est-à-dire 3 en code binaire, ce qui indique que l'erreur est apparue sur le troisième bit. Le résultat de décodage avec correction d'erreur est alors

$$z = \hat{z} - 0010000.$$

Remarque 3.9 :

On peut montrer que pour un canal ayant une probabilité d'erreur de 5%, la probabilité de transmettre correctement 4 bits est $.95^4 \approx 81\%$.

L'utilisation de $C(4,7)$ permet de faire passer cette valeur à $(1-p)^7+7p(1-p)^6 \approx 95\%$. Ce résultat, comparable à la stratégie par triplement, est en fait bien meilleur, car la taille des mots du code n'a même pas doublé. Notons enfin qu'on peut rendre la probabilité d'erreur aussi faible que l'on veut en appliquant un code correcteur plusieurs fois de suite.

3.3 NOTES BIBLIOGRAPHIQUES

Ce chapitre s'inspire de notes de cours gracieusement fournies par Yvan Pigeonnat. Pour plus d'informations sur le codage source, on pourra consulter les références [10], [4]. Pour plus d'informations sur les codes correcteurs (de type Hamming), on pourra également consulter [4], mais bien d'autres documents relatifs à la correction d'erreur se trouvent facilement sur Internet. Enfin, pour des références très complète sur le codage et pour découvrir des codages autres que le codage de Hamming (il en existe énormément), on pourra consulter [11, 8] et [1]. Pour les structures de données, les listes, les arbres, l'algorithme de Huffman, on pourra consulter [7].

CHAPITRE 4

Transformation de Fourier discrète

SOMMAIRE DU CHAPITRE

Ce chapitre est un préliminaire aux chapitres qui suivent. Avant d'aborder les filtres, qui permettent de travailler et de modifier la transformée de Fourier, en d'autres termes le *spectre* d'un signal, il est nécessaire d'indiquer comment calculer effectivement ce spectre. Puisque les signaux auxquels on s'intéresse sont discrets (car numériques), on se concentre ici sur la transformée de Fourier discrète. Son calcul, si il est fait naïvement, peut être très coûteux. Il peut cependant être accéléré considérablement en utilisant une méthode due à Tuckey [1] et Cooley [2], le fameux algorithme *FFT* , dont le nom est l'acronyme correspondant à *Fast Fourier Transform*.

1. John Wilder Tukey (16 juin 1915, New Bedford, Massachusetts - 26 juillet 2000, New Brunswick, New Jersey.) était un statisticien américain.
2. James Cooley (1926-2016) était un mathématicien americain.

4.1 La transformée de Fourier discrète (TFD)

On commence par rappeler le cadre discret où l'on se place.

4.1.1 Cadre et problèmes

On considère un signal échantillonné, fini :

$$s = \{s_0, s_1, \dots, s_{N-2}, s_{N-1}\},$$

obtenu à partir d'un signal analogique. également noté s, par une formule du type :

$$s_n = s(n\Delta t),$$

où Δt est le pas de l'échantillonnage, relié à la fréquence d'échantillonnage $2B$ du chapitre 1 par l'équation :

$$\Delta t = \frac{1}{2B}.$$

On ne tient pas compte ici du fait que le signal peut avoir été quantifié, c'est-à-dire que les s_n sont considérés *a priori* comme réels.

Définition 4.1 — (Transformée de Fourier discrète (TFD)) On appelle *transformée de Fourier discrète*, en abrégé TFD, de la suite finie $s = \{s_0, s_1, \dots, s_{N-2}, s_{N-1}\}$, la suite finie

$$\hat{s} = \{\hat{s}_0, \hat{s}_1, \dots, \hat{s}_{N-2}, \hat{s}_{N-1}\}$$

définie par la formule :

$$\hat{s}_k = \sum_{n=0}^{N-1} s_n \mathrm{e}^{-2\mathrm{i}\pi k \frac{n}{N}}.$$

Comme toute transformée de Fourier, on voit que la TFD est une application linéaire. Sa matrice est une *matrice pleine*, c'est-à-dire ne comportant pas de 0, de type matrice de Vandermonde. On voit de plus que le pas de temps utilisé n'apparaît pas dans la formule. On peut en quelque sorte considérer que celui-ci est égal à 1. Ceci est en fait naturel : un signal discret est une suite de nombres et seules des informations supplémentaires sur ce qu'elle représente permettent de connaître la fréquence qui a été utilisée pour obtenir l'échantillon.

On va s'intéresser à deux questions dans la suite :

1. Quel est le coût de calcul de la TFD ? et comment l'améliorer ?
2. Quelle est la qualité de l'approximation de la transformée de Fourier du signal analogique initial ?

4.1.2 Propriétés de la TFD et signaux périodiques discrets

Pour simplifier l'exposé, on introduit de nouvelles notations. Soit une suite finie $a = \{a_0, a_1, ..., a_{N-2}, a_{N-1}\}$, et $\omega_N = \mathrm{e}^{-\mathrm{i}2\pi/N}$. Alors la suite $A = \{A_0, A_1, ..., A_{N-2}, A_{N-1}\}$, définie par

$$A_m = \sum_{n=0}^{N-1} a_n \omega_N^{nm} \tag{4.1}$$

est la TFD de a. Sous forme matricielle, cela s'écrit :

$$\begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ \vdots \\ A_{N-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & \cdots & 1 \\ 1 & \omega_N^1 & \omega_N^2 & \cdots & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \cdots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega_N^N & \omega_N^{2(N-1)} & \cdots & \cdots & \omega_N^{(N-1)^2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ a_{N-1} \end{pmatrix},$$

et l'on notera la matrice de Vandermonde Ω_N.

Dans le cas de la transformée de Fourier discrète, on peut, sans autre hypothèse, inverser la transformée pour obtenir le signal initial.

THÉORÈME 4.1 — (Formule d'inversion de la TFD) En gardant les notations précédentes, on a :

$$a_n = \frac{1}{N} \sum_{m=0}^{N-1} A_m \omega_N^{-nm}.$$

Exercice 4.1 :

Prouver ce résultat.

Évidemment, on retrouve dans la TFD et sa formule d'inversion les analogies habituelles entre les transformées de Fourier et leurs inverses. Dans le cas

présent, on retrouve simplement que l'inverse d'une matrice de Vandermonde est une matrice de Vandermonde [3], c'est-à-dire $\Omega_N^{-1} = 1/N \times \overline{\Omega_N}$.

Remarque 4.1 :

Cette formule reste valable si on considère *les extensions périodiques* de a et A, définies par les formules :

$$a_{n+kN} = a_n, \quad A_{m+kN} = A_m.$$

C'est maintenant comme cela que l'on envisagera les choses. Tous les signaux considérés seront vus dorénavant comme des signaux discrets périodiques.

Par conséquent, l'ordre de sommation peut-être choisi arbitrairement. En supposant par exemple que $N = 2M + 1$, on obtient :

$$A_m = \sum_{n=0}^{N-1} a_n \omega_N^{nm}, \quad a_n = \frac{1}{2M+1} \sum_{m=-M}^{M} A_m \omega_N^{-nm}.$$

Pour simplifier encore, on notera dans la suite ces relations en abrégé par :

$$(a_n) \overset{\mathcal{F}_N}{\longleftrightarrow} (A_m).$$

THÉORÈME 4.2 Si $(a_n) \overset{\mathcal{F}_N}{\longleftrightarrow} (A_m)$ et $(b_n) \overset{\mathcal{F}_N}{\longleftrightarrow} (B_m)$, alors :

$$\left(\sum_{k=0}^{N-1} a_k b_{n-k} \right) \overset{\mathcal{F}_N}{\longleftrightarrow} (A_m B_m).$$

Exercice 4.2 :

Montrer ce résultat.

Ce théorème est une variante de ceux que nous avions déjà vus au chapitre 1 sur le lien entre séries de Fourier et convolution.

3. proportionnelle à une matrice de Vandermonde, pour être plus précis.

4.2 L'algorithme FFT

4.2.1 Nombres d'opérations pour le calcul de la TFD

En appliquant la formule (4.1), on est conduit à calculer la somme suivante :

$$A_m = a_0 + a_1\omega_N^m + a_2\omega_N^{2m} + \ldots + a_{N-1}\omega_N^{m(N-1)}.$$

Considérons que la suite $(\omega_N^{km})_{k=0,\ldots,N-1}$ a été pré-calculée [4], pour un élément A_m, on est conduit à effectuer $N-1$ multiplications et N additions. Le calcul complet de la suite A nécessite donc $N(N-1)$ multiplications et N^2 additions , ce qui n'est pas satisfaisant du tout dès lors que N devient « assez » grand.

On va montrer comment réduire à $N\log_2 N$ le nombre de multiplications.

4.2.2 L'algorithme de Tuckey et Cooley

Publié en 1965 dans un article devenu depuis célèbre, Tuckey et Cooley ont donné une méthode permettant de réduire considérablement le temps de calcul de la suite (A_m) précédente.

Première itération

On se place dans le cas où la taille des suites considérées est paire et on considère

$$(a_n) \overset{\mathcal{F}_{2N}}{\longleftrightarrow} (A_m).$$

On introduit alors deux suites $(b_n)_{n=0,\ldots,N-1}$ et $(c_n)_{n=0,\ldots,N-1}$ définies par $b_n = a_{2n}$ et $c_n = a_{2n+1}$ et on note :

$$(b_n) \overset{\mathcal{F}_N}{\longleftrightarrow} (B_m), \quad (c_n) \overset{\mathcal{F}_N}{\longleftrightarrow} (C_m).$$

On remarque alors que :

— $A_m = B_m + \omega_{2N}^m C_m$, pour $m = 0, \ldots, 2N-1$,

— $B_{m+N} = B_m$, $C_{m+N} = C_m$ et $\omega_{2N}^{m+N} = -\omega_{2N}^m$.

Exercice 4.3 :

Le prouver.

4. C'est-à-dire calculée une fois pour toute, en supposant que l'entier N est fixé.

On en déduit :

$$A_m = B_m + \omega_{2N}^m C_m \qquad m = 0, \dots, N-1 \tag{4.2}$$
$$A_{m+N} = B_m - \omega_{2N}^m C_m \qquad m = 0, \dots, N-1. \tag{4.3}$$

On voit ainsi que le nombre nécessaire de multiplication pour calculer la suite (A_m) est égal à deux fois le nombre de multiplications pour calculer la suite (B_m) ou (C_m) (qui ont la même taille) plus N multiplications (les calculs de $\omega_{2N}^m C_m$).

De même, le nombre nécessaire d'additions pour calculer la suite (A_m) est égal à deux fois le nombre d'additions pour calculer la suite (B_m) ou (C_m) (qui ont la même taille) plus $2N$ additions (les additions et soustrations de $\omega_{2N}^m C_m$).

Cas général

On peut aller encore plus loin ! Il suffit pour cela d'appliquer récursivement le raisonnement précédent. Supposons pour ce faire que $N = 2^M$. Comme N est de cette forme on pourra reproduire la division en deux sous suites pour les suite (B_m) et (C_m) et ainsi de suite, M fois.

Notons $\mathcal{M}(j)$ et $\mathcal{A}(j)$ respectivement le nombre de multiplications et d'additions à l'étape j, j allant de 1 à M.

Regardons tout d'abord ce qu'il se passe quand on a une suite de deux éléments. Calculons la suite (A_0, A_1) à partir de la suite (a_0, a_1). On a

$$\begin{cases} A_0 = a_0 + a_1 \\ A_1 = a_0 - a_1 \end{cases}$$

Ainsi $\mathcal{M}(1) = 0$ et $\mathcal{A}(1) = 2$.

Avec notre raisonnement sur (4.2) et (4.3), on a les relations de récurrence suivantes, pour tout j allant de 1 à $M-1$:

$$\mathcal{M}(j+1) = 2\mathcal{M}(j) + 2^j \tag{4.4}$$
$$\mathcal{A}(j+1) = 2\mathcal{A}(j) + 2 \times 2^j \tag{4.5}$$

En multipliant (4.4) et (4.5) par $2^{M-(j+1)}$ et en sommant pour j de 1 à $M-1$ respectivement les équations sur les multiplications et sur les additions, on obtient :

$$\mathcal{M}(M) = 2^{M-1}\mathcal{M}(1) + (M-1)2^{M-1}$$
$$\mathcal{A}(M) = 2^{M-1}\mathcal{A}(1) + (M-1) \times 2^M$$

Ainsi, puisque $\mathcal{M}(1) = 0$ et $\mathcal{A}(1) = 2$, on obtient le théorème suivant.

THÉORÈME 4.3 En notant $\mathcal{A}(M)$ et $\mathcal{M}(M)$ le nombre d'additions et de multiplications nécessaires pour calculer l'ensemble des coefficients de Fourier de la suit (a_n) de taille $N = 2^M$, on a :

$$\mathcal{M}(M) = (M-1)2^{M-1} = \frac{1}{2}(N\log_2 N - N)$$
$$\mathcal{A}(M) = M2^M = N\log_2 N$$

Pour illustrer ce résultat, considérons une suite de $N = 1024$ termes. Par la méthode naïve expliquée en introduction de cette section, le calcul nécessite $(N)^2 = 1\,048\,576$ additions et $N(N-1) = 1\,047\,552$ multiplications. Par l'algorithme FFT de Tuckey et Cooley, il nécessite moins de $N\log_2 N = 10240$ additions et 4608 multiplications. Le rapport du temps de calcul entre les deux méthodes est donc d'à peu près 140. Autrement dit, ce qui nécessitait presque 5 mois ne requiert maintenant qu'une journée.

Si la taille de l'échantillon n'est pas une puissance de 2, ce qui n'arrive pas en traitement numérique du signal, on complète généralement l'échantillon par des zéros pour pouvoir appliquer l'algorithme.

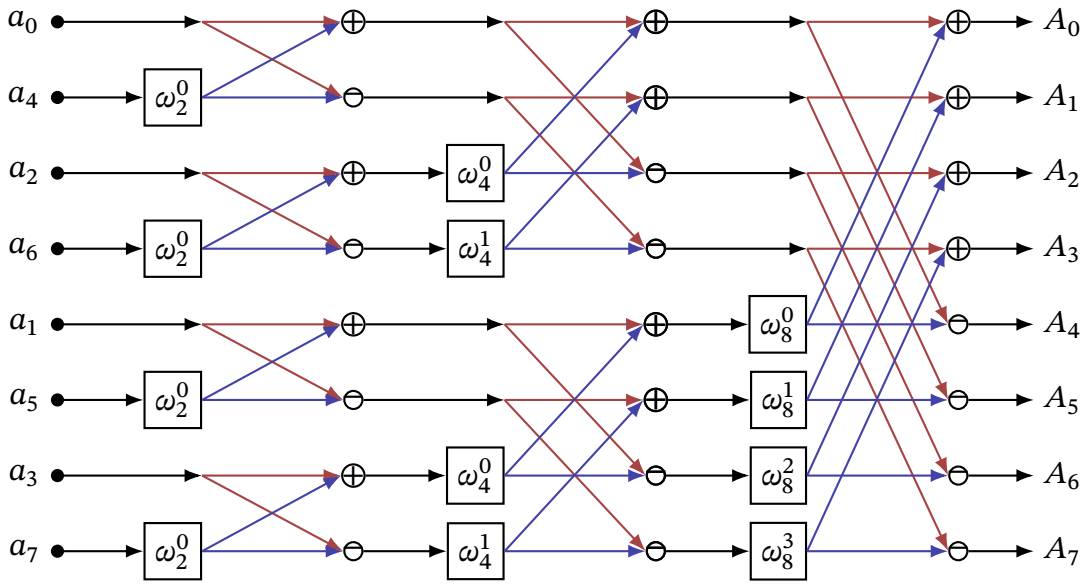


FIGURE 4.1 – Schéma illustratif de l'algorithme de la FFT.

4.3 Analyse numérique

À ÉCRIRE

4.4 Notes bibliographiques

Ce chapitre s'inspire de la présentation de la FFT faite dans la référence [3], qui ne contient pas l'application aux polynômes. On pourra également consulter les chapitres 7 et 8 de la référence [6] pour une autre présentation de l'algorithme, moins mathématique. Ici encore, de nombreux textes et documents concernant la FFT sont disponibles sur Internet.

CHAPITRE 5

Filtres numériques

SOMMAIRE DU CHAPITRE

On aborde dans ce chapitre, le filtrage des signaux numériques. Une fois le signal échantillonné et quantifié, il est utile pour de nombreuses applica-

tions d'en définir et calculer le contenu fréquentiel, c'est-à-dire son spectre. L'algorithme de la FFT nous permet de réaliser efficacement cet objectif.

L'étape suivante, à laquelle on s'attaque maintenant, est d'agir sur ce spectre de manière à en atténuer, amplifier, sélectionner ou occulter certaines fréquences, ou certaines plages de fréquences. Cette démarche s'appelle le *filtrage* et peut être réalisée, pour ce qui concerne les signaux discrets, par des *filtres numériques* [1]. Dans ce chapitre, on donne les bases permettant de définir un filtre et ses propriétés.

5.1 Filtres linéaires

Un filtre numérique peut-être vu de plusieurs manières, selon le domaine dans lequel on travaille. Certains le définiront par un circuit électronique, d'autres par un assemblage de portes logiques… Notre penchant pour les mathématiques nous conduit plutôt à les assimiler à des algorithmes de calculs. Mais on parle en tout cas de la même chose.

5.1.1 Définitions

On donne donc une définition mathématique d'un filtre numérique.

Définition 5.1 Un filtre numérique est une application de $\ell^2(\mathbf{Z}) \to \ell^2(\mathbf{Z})$ où $\ell^2(\mathbf{Z})$, noté aussi ℓ^2, est définie par

$$\ell^2(\mathbf{Z}) = \left\{(x_n)_{n\in\mathbf{Z}};\ \sum_{n\in\mathbf{Z}} |x_n|^2 < +\infty\right\}.$$

Cet espace est un espace vectoriel muni de la norme définie pour tout $x = (x_n) \in \ell^2$,

$$\|x\|_2 = \left(\sum_{n\in\mathbf{Z}} |x_n|^2\right)^{1/2}.$$

Un filtre numérique est donc un système qui produit un signal discret, que l'on notera dans la suite $y = (y_n)_{n\in\mathbf{Z}}$ à partir d'un signal reçu en entrée $x = (x_n)_{n\in\mathbf{Z}}$. On appelle parfois *excitation* ou *entrée* du filtre la suite x et

1. Il existe aussi des filtres analogiques, qui agissent sur des fonctions L^1, par exemple, et dont les outils sont proches de ceux introduit au chapitre 1. On en trouve dans la nature, par exemple l'oreille humaine, ou plus généralement tout système physique répondant à une excitation. Du point de vue de l'activité humaine, l'essor massif des technologies numériques les place plutôt sur la liste des espèces en voie de disparition.

réponse ou *sortie* du filtre la suite y.

On schématisera cette définition par la figure 5.1.

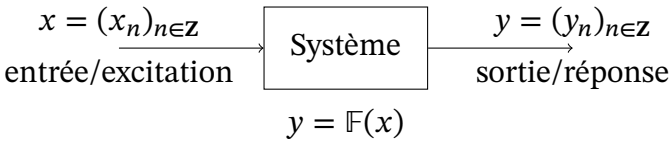


FIGURE 5.1 – Schématisation d'un filtre numérique.

Dans la pratique, les signaux rencontrés sont finis, si bien que le fait de travailler dans $\ell^2(\mathbf{Z})$ plutôt que dans un autre $\ell^p(\mathbf{Z})$ n'a pas beaucoup d'importance. Le fait de choisir cet espace plutôt qu'un autre vient du fait que la norme 2 d'un signal (discret ou pas) est souvent égale à son énergie et que d'un point de vue physique, il est raisonnable de ne considérer que des signaux d'énergie finie.

Remarque 5.1 :

Dans toute la suite, on ne considère –– sauf mention contraire –– que des filtres qui élaborent la réponse en un temps fixé n en fonction d'un nombre *fini* de termes de x ou de y.

Dans la réalité, les dispositifs dont on dispose ne permettent de toute façon que de travailler dans le cadre fixé par cette remarque. Donnons un exemple de filtre. La formule :

$$y_n = x_{n-1}$$

définit un filtre qui effectue un décalage unitaire. On fera souvent appel dans la suite à ce filtre élémentaire, si bien qu'on fixe d'ores-et-déjà sa notation.

DÉFINITION 5.2 — (Décalage unitaire) On appelle décalage unitaire et on note τ le filtre défini par la relation

$$y_n = x_{n-1}.$$

La suite y peut être cependant définie de manière plus compliquée, par exemple de manière récursive, comme c'est le cas pour :

$$y_n = \frac{1}{4}y_{n-1} + \frac{1}{2}x_n + \frac{1}{2}x_{n-1}. \tag{5.1}$$

On parle alors de *filtre récursif*[2]. Attention, un même filtre peut avoir une définition récursive et une définition non récursive. Par exemple :

$$y_n = \frac{1}{2} y_{n-1} + x_n$$

produit le même filtre que :

$$y_n = \sum_{k \in \mathbf{N}} \left(\frac{1}{2}\right)^k x_{n-k}.$$

Par contre, cette dernière formule n'entre pas dans le cadre des filtres non-récursifs que l'on considère dans la suite de ce cours, puisqu'on s'est placé dans le champs de la remarque 5.1.

Réponse impulsionelle

Dans la suite, on considérera souvent l'image par les filtres de l'entrée *impulsion*, c'est-à-dire le signal spécifique :

$$\delta^0_n = \begin{cases} 1 & \text{si} \quad n = 0, \\ 0 & \text{sinon.} \end{cases}$$

DÉFINITION 5.3 — (Réponse impulsionelle) On appelle réponse impulsionnelle d'un filtre numérique l'image par ce filtre du signal d'entrée $\delta^0 = (\delta^0_n)_{n \in \mathbf{Z}}$.
La réponse impulsionelle sera notée $h = (h_n)_{n \in \mathbf{Z}}$.

5.1.2 Propriétés des filtres

On passe maintenant en revue les premières propriétés des filtres numériques. Pour simplifier les définitions, on note $\mathbb{F}$ le filtre, considéré comme une fonction.

DÉFINITION 5.4 — (Invariance temporelle) Un filtre est dit invariant en temps, si l'image par $\mathbb{F}$ de la suite décalée $(x_{n-n_0})_{n \in \mathbf{Z}}$, où n_0 est un entier relatif fixé, est la suite décalée de $y = (y_n)_{n \in \mathbf{Z}}$, c'est-à-dire $(y_{n-n_0})_{n \in \mathbf{Z}}$.

De manière équivalente, un filtre est invariant en temps si sa fonction $\mathbb{F}$ commute avec la fonction décalage τ.

2. On verra dans la suite du chapitre des définitions plus précises.

Exercice 5.1 :

Montrer que le filtre défini par la relation :

$$y_n = nx_n,$$

ne définit pas un filtre invariant en temps.

Remarque 5.2 :

Tous les filtres que nous considérerons à partir de maintenant seront *invariants en temps*.

DÉFINITION 5.5 — (Linéarité) Un filtre est dit *linéaire* si la relation de dépendance de y à x est linéaire.

Autrement dit, un filtre est dit linéaire si la fonction $\mathbb{F}$ est une application linéaire.

Un premier théorème permet de caractériser simplement les filtres linéaires invariants en temps.

THÉORÈME 5.1 Un filtre linéaire invariant ou *système linéaire invariant* (SLI) en temps est entièrement déterminé par la donnée de sa réponse impulsionnelle, c'est-à-dire, en notant $(h_n)_{n\in\mathbf{Z}}$ la réponse impulsionnelle, on a pour une entrée $(x_n)_{n\in\mathbf{Z}}$, une sortie

$$\forall n \in \mathbf{Z},\ y_n = \sum_{k\in\mathbf{Z}} x_k h_{n-k}. \tag{5.2}$$

Remarque 5.3 :

Nous ne considérerons à partir de maintenant que les systèmes linéaires invariants.

Remarque 5.4 :

La propriété du théorème 5.1, à savoir d'être entièrement déterminé par la donnée de sa réponse impulsionnelle peut avoir une grande utilité. Imaginons que l'on dispose d'un filtre devant nous dont nous ne connaissons

pas les propriétés, mais auquel nous pouvons soumettre une entrée et récupérer la sortie correspondante. Supposons alors que c'est un SLI. Il suffit alors d'envoyer une entrée le signal *impulsion* et de récupérer la réponse impulsionnelle (sortie) pour, si notre hypothèse SLI est valide, pouvoir prédire la réponse à n'importe quel signal d'entrée.

Preuve : Notons $h = (h_n)_{n\in\mathbf{Z}}$ la réponse impulsionnelle d'un filtre défini par une fonction $\mathbb{F}$.

On a :

$$\begin{aligned}
(y_n)_{n\in\mathbf{Z}} &= \mathbb{F}\left((x_n)_{n\in\mathbf{Z}}\right)\\
&= \mathbb{F}\left(\left(\sum_{k\in\mathbf{Z}} x_k\delta^0_{n-k}\right)_{n\in\mathbf{Z}}\right)\\
&= \sum_{k\in\mathbf{Z}} x_k\mathbb{F}\left((\delta^0_{n-k})_{n\in\mathbf{Z}}\right) \qquad (\mathbb{F}\text{ linéaire})\\
&= \sum_{k\in\mathbf{Z}} x_k\mathbb{F}\left(\tau^{-k}(\delta^0_n)_{n\in\mathbf{Z}}\right) \qquad (\text{Déf. de } \tau)\\
&= \sum_{k\in\mathbf{Z}} x_k\tau^{-k}\mathbb{F}\left((\delta^0_n)_{n\in\mathbf{Z}}\right) \qquad (\mathbb{F}\text{ invariante en temps})\\
&= \sum_{k\in\mathbf{Z}} x_k\tau^{-k}(h_n)_{n\in\mathbf{Z}} \qquad (\text{Déf. de } (h_n)_{n\in\mathbf{Z}})\\
&= \sum_{k\in\mathbf{Z}} x_k(h_{n-k})_{n\in\mathbf{Z}} \qquad (\text{Déf. de } \tau)\\
&= \left(\sum_{k\in\mathbf{Z}} x_k h_{n-k}\right)_{n\in\mathbf{Z}}
\end{aligned}$$

On voit donc que le calcul de y nécessite seulement la connaissance de la suite $h = (h_n)_{n\in\mathbf{Z}}$. ■

La formule (5.2) est une formule de convolution. On voit ainsi une relation algébrique très simple entre l'entrée et la sortie d'un filtre linéaire, invariant en temps :

$$y = h \star x,$$

qui montre que de tels filtres ne font, *in fine*, que réaliser des produits de convolution.

Causalité

Dans les exemples que nous avons traités jusqu'à présent, l'indice n des signaux numériques représente le temps. Dans ce cadre, il est impossible qu'un

filtre élabore une réponse en fonction de données du futur. C'est ce qui motive l'introduction de la définition suivante.

DÉFINITION 5.6 — (Causalité) Un filtre est dit causal si, à n fixé, y_n ne dépend que des valeurs x_k, pour $k \leq n$ et y_k, pour $k \leq n-1$.

La réponse impulsionnelle h d'un tel filtre est donc nécessairement à support dans $\mathbf{N}$. On a alors un équivalent du théorème 5.1.

THÉORÈME 5.2 — (Filtre linéaire, invariant en temps et causal)
Un filtre linéaire, invariant en temps et causal vérifie :

$$y_n = \sum_{k \leq n} h_{n-k} x_k = \sum_{k \in \mathbf{N}} h_k x_{n-k},$$

où l'on a noté $h = (h_n)_{n \in \mathbf{N}}$ la réponse impulsionelle du filtre.

Remarque 5.5 :

Il existe cependant des cas où il est nécessaire de considérer des filtres non causaux. En traitement d'image par exemple, où les signaux – les images donc – sont de dimension 2, les filtres appliquent à chaque pixel de l'image une fonction de l'ensemble des pixels de l'image considérée. La notion de causalité dans ce cas n'a pas vraiment de sens.

5.2 STABILITÉ

On introduit maintenant une notion importante pour la conception des filtres.

De manière imagée, on parle de *filtre stable* si le filtre n'explose pas pour certaines entrées. La plupart des filtres que l'on considère sont évidemment stables. Les filtres instables donnent cependant parfois lieu à des phénomènes de saturation, qui peuvent être mis à profit dans certaines applications [3].

3. Par exemple en électronique, où l'utilisation des Amplificateurs Opérationnels en régime saturé permet de concevoir des oscillateurs et donc des horloges.

DÉFINITION 5.7 Un filtre est dit stable si il laisse stable le sous-espace $\ell^\infty(\mathbf{Z})$.

Autrement dit, pour toute entrée bornée, la sortie est bornée.

Remarque 5.6 :

Cette définition est valable pour tout filtre, même non-linéaire.

Dans le cas des filtres linéaires, invariants en temps, on a une caractérisation simple de la stabilité.

LEMME 5.1 Un filtre linéaire, invariant en temps est stable si et seulement si sa réponse impulsionnelle est une suite de $\ell^1(\mathbf{Z})$.

Preuve : Condition suffisante : supposons $h = (h_n)_{n\in\mathbf{Z}} \in \ell^1(\mathbf{Z})$ et $x = (x_n)_{n\in\mathbf{Z}} \in \ell^\infty(\mathbf{Z})$. Alors :

$$|y_n| = |(h \star x)_n| \leq \sum_{k\in\mathbf{Z}} |h_k| \cdot |x_{n-k}| \leq \|x\|_\infty \cdot \sum_{k\in\mathbf{Z}} |h_k| = \|x\|_\infty \|h\|_1,$$

ce qui montre que y est bornée, donc dans $\ell^\infty(\mathbf{Z})$.

Condition nécessaire : supposons, pour simplifier, que h soit à valeurs réelles. Posons $x = \big(\text{signe}(h_{-n})\big)_{n\in\mathbf{Z}}$. Puisque le système est stable, $h \star x \in \ell^\infty(\mathbf{Z})$, or :

$$(h \star x)_0 = \sum_{k\in\mathbf{Z}} h_k x_{-k} = \sum_{k\in\mathbf{Z}} |h_k|.$$

On en déduit le résultat. ■

Ce lemme n'est qu'une reformulation du fait que le dual de $\ell^1(\mathbf{Z})$ est $\ell^\infty(\mathbf{Z})$.

5.3 FILTRES LINÉAIRES RÉCURSIFS CAUSAUX

On définie précisément ici ce que nous avons introduit à la remarque 5.1.

DÉFINITION 5.8 — (Filtres linéaires récursifs causaux) Les filtres linéaires récursifs causaux élaborent leur réponse au temps n par une formule du type :

$$y_n = \sum_{k=1}^{N} a_k y_{n-k} + \sum_{k=0}^{M} b_k x_{n-k}, \tag{5.3}$$

où les (a_k) et les (b_k) sont des constantes.

L'intérêt de tels filtres est qu'ils sont très faciles à implémenter en pratique. On dit que ces systèmes sont définis par *une équation aux différences*.

5.3.1 Réponse impulsionnelle finie (RIF) et infinie (RII)

Les filtres linéaires invariants en temps et causaux peuvent maintenant être divisés en deux catégories.

DÉFINITION 5.9 — (RIF et RII) Un filtre linéaire, invariant en temps et causal est dit :

— *à réponse impulsionnelle finie* (en abrégé *RIF* ou *FIR*) si $h = (h_n)_{n\in\mathbf{Z}}$ est à support fini. Ces filtres sont un cas particulier des filtres récursifs causaux, où les a_k sont nuls, c'est-à-dire :

$$y_n = \sum_{k=0}^{M} b_k x_{n-k}.$$

— Il est dit à *réponse impulsionnelle infinie* (en abrégé RII ou IIR) dans le cas où il existe un ou des a_k non nuls.

5.3.2 Transformée en z

La réponse impulsionnelle d'un filtre n'est pas forcément très parlante. Nous avons donc besoin d'outils pour caractériser les filtres et savoir construire la réponse impulsionnelle d'un filtre en fonction des caractéristiques que l'on désire pour lui. Un de ces outil est ce qu'on appelle *la transformée en z*.

DÉFINITION 5.10 — (Transformée en z) Étant donné un signal $x = (x_n)_{n\in\mathbf{Z}}$, on *appelle transformée en z* de x, la fonction :

$$X(z) = \sum_{n\in\mathbf{Z}} x_n z^{-n},$$

pour tout $z \in \mathbf{C}$ dans le domaine de convergence

$$D_x = \{z \in \mathbf{C};\ R_1 < |z| < R_2\},$$

R_2 pouvant valoir l'infini.

Traditionnellement, on utilise des lettres majuscules pour noter la transformée en z.

Remarque 5.7 :

1. La transformée en z est une série entière dite *série de Laurent*.
2. Le domaine de convergence de ces séries est un sujet en soit, faisant appel à l'analyse complexe. Nous ne l'aborderons pas ici.
3. Il y a un lien entre la transformée en z et la transformée de Fourier discrète : la TFD est la transformée en z restreinte au cercle unité $\left(z = \mathrm{e}^{2\mathrm{i}\pi/N}\right)$ à condition qu'il appartienne à la couronne de convergence.
4. La transformée en z est à la TFD ce que la transformée de Laplace est à la transformée de Fourier.

On peut montrer les quelques propriétés suivantes sur le domaine de convergence.

Proposition 5.1 : — *Si (x_n) est de durée finie, D_x est le plan complexe tout entier, sauf peut-être en $z = 0$.*
— *Si (x_n) est nulle à gauche ($x_n = 0$ pour tout $n < N$), D_x s'étend à l'infini.*
— *Si (x_n) est nulle à droite ($x_n = 0$ pour tout $n > N$), D_x contient l'origine.*
— *Si (x_n) s'étend à droite et à gauche, alors D_x est une couronne.*

Exercice 5.2 :

Calculer la transformée en z de (x_n) définie par

$$\forall n \in \mathbf{N}, \quad x_n = a^n$$

où $a \in \mathbf{R}$, $x_n = 0$ pour $n < 0$.

La transformée en z possède aussi d'autre propriétés qui nous seront utiles.

Proposition 5.2 : *La transformée en z satisfait les propriétés suivantes :*

Propriété	*suite*	*Trans. en z*	*Domaine*
Linéarité	$(x_n) + \lambda(y_n)$	$X(z) + \lambda Y(z)$	$D_x \cap D_y$
Retournement temporel	(x_{-n})	$X(1/z)$	$1/D_x$
Retard	(x_{n-n_0})	$X(z)z^{-n_0}$	D_x
Convolution	$((x \star y)_n)$	$X(z)Y(z)$	$D_x \cap D_y$
Produit	$(x_n y_n)$	$X \star Y(z)$	$D_x \times D_y$

Exercice 5.3 :

Démontrer la propriété de retard de la proposition 5.2.

Calculer les transformées inverses

Comme souvent, il est intéressant d'avoir la transformée en *z inverse*. Pour la calculer, il faut encore faire appel à l'analyse complexe et intégrer sur un contour de Cauchy. En pratique, les transformées en z qui nous intéresserons seront des fractions rationnelles (voir exercice 5.2) que l'*on décomposera en éléments simples*.

Nous rappelons ici le théorème de décomposition en éléments simples sur le corps des complexes.

THÉORÈME 5.3 — (Décomposition en éléments simples) Toute fraction rationnelle $F = P/Q \in \mathbf{C}(x)$ admet une unique décomposition en éléments simples, c'est-à-dire comme somme d'un polynôme T et de fractions $a/(x–z)^k$ avec a et z complexes et k entier supérieur ou égal à 1. Si Q admet la factorisation

$$Q = (x - z_1)^{n_1}(x - z_2)^{n_2} \dots (x - z_p)^{n_p},$$

alors la décomposition de F est de la forme

$$F = \frac{P}{Q} = T + F_1 + \dots + F_p \quad \text{et} \quad F_i = \frac{a_{i,1}}{x - z_i} + \frac{a_{i,2}}{(x - z_i)^2} + \dots + \frac{a_{i,n_i}}{(x - z_i)^{n_i}},$$

c'est-à-dire que les seuls $a/(x–z)^k$ avec a non nul qui risquent d'apparaître sont pour z égal à un pôle de F et k inférieur ou égal à son ordre. T est appelé *la partie entière* de F.

Exercice 5.4 :

Déterminer la transformée inverse de :

$$X(z) = \frac{1}{1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}}.$$

5.3.3 Fonction de transfert

Puisque cette nouvelle transformée s'applique à tout signal numérique, on peut en particulier l'appliquer à la réponse impulsionnelle.

DÉFINITION 5.11 — (Fonction de transfert) On appelle fonction de transfert la transformée en z de la réponse impulsionnelle :

$$H(z) = \sum_{i \in \mathbf{Z}} h_i z^{-i}.$$

On peut alors établir un lien entre les transformées en z de cette dernière, de l'entrée et de la sortie.

LEMME 5.2 Soit x et y l'entrée et la sortie d'un filtre linéaire récursif causal défini par (5.3) et de réponse impulsionnelle h. On a :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}.$$

Preuve : Commençons par établir la deuxième égalité. On a :

$$\begin{aligned}
Y(z) &= \sum_{n \in \mathbf{Z}} y_n z^{-n} \\
&= \sum_{n \in \mathbf{Z}} \left(\sum_{k=1}^{N} a_k y_{n-k} \right) z^{-n} + \sum_{n \in \mathbf{Z}} \left(\sum_{k=0}^{M} b_k x_{n-k} \right) z^{-n} \\
&= \sum_{k=1}^{N} a_k z^{-k} \left(\sum_{n \in \mathbf{Z}} y_{n-k} z^{-(n-k)} \right) + \sum_{k=0}^{M} b_k z^{-k} \left(\sum_{n \in \mathbf{Z}} x_{n-k} z^{-(n-k)} \right) \\
&= \sum_{k=1}^{N} a_k z^{-k} Y(z) + \sum_{k=0}^{M} b_k z^{-k} X(z),
\end{aligned}$$

d'où le résultat.

La première égalité provient quant à elle du fait que la transformée en z, comme le font les différentes transformées de Fourier, transforme le produit de convolution en produit. ■

On voit facilement que la fonction de transfert d'un filtre récursif causal est une fraction rationnelle. La réciproque n'est pas vraie. On peut par exemple voir que le filtre défini par la formule :

$$y_n = x_{n+1}$$

à pour fonction de transfert $H(z) = z$.

DÉFINITION 5.12 — (Forme normalisée) On appelle *forme normalisée* de la fonction de transfert d'un filtre linéaire récursif causal est une fraction rationnelle

$$H(z) = \frac{\prod_{k=1}^{M}(1 - z_k z^{-1})}{\prod_{k=0}^{N}(1 - p_k z^{-1})},$$

où les coefficients $z_k,\ k \in \{1, \dots, M\}$ et $p_k,\ k \in \{0, \dots, M\}$ sont appelés respectivement zéros et pôles de la fonction transfert.

Dans la définition précédente, on suppose bien entendu que la fraction rationnelle est irréductible, c'est-à-dire qu'aucun facteur n'apparaît à la fois au numérateur et au dénominateur.

Exercice 5.5 :

Calculer la fonction transfert des filtres définis par :

— $y_n = x_{n-1}$;
— $y_n = \frac{1}{4}y_{n-1} + \frac{1}{2}x_n + \frac{1}{2}x_{n-1}$.

5.3.4 Stabilité

La fonction de transfert introduite à la section précédente permet de formuler un nouveau critère de stabilité des filtres linéaires récursifs, causaux.

THÉORÈME 5.4 Un filtre linéaire, récursif, causal est stable si et seulement si les pôles de sa fonction de transfert (mise sous forme irréductible) sont situés à l'intérieur du disque unité.

Preuve : (partielle) Puisque l'on considère un filtre causal, on sait que le domaine de convergence de sa fonction de transfert est l'extérieur d'un cercle,

Par hypothèse de rationalité de la fonction de transfert, on peut la réduire en éléments simples, c'est-à-dire la mettre sous la forme, où on a supposé pour simplifier que les pôle étaient simples

$$H(z) = \sum_{i\in\{1,\dots,N\}} \frac{\alpha_i}{1-p_i z^{-1}} =: \sum_i H_i(z).$$

Le développement en série entière par rapport au paramètre z^{-1} de chacun des éléments de cette somme vaut :

$$H_i(z) = \sum_{k\in\mathbf{N}} p_i^k z^{-k}.$$

Par identification, on voit que $H_i(z)$ est la transformée en z de la réponse impulsionnelle h^i définie par $h^i_n = p_i^n$, qui appartient à $\ell^1(\mathbf{Z})$ si et seulement si $|p_i| < 1$. ■

Il existe de nombreux critères permettant de déterminer rapidement par calcul si les racines d'un polynôme donné sont situées à l'intérieur ou à l'extérieur du disque unité. On peut citer par exemple :

— le critère de Schur,
— le critère de Routh,
— le critère de Jary,
— le critère de Nyquist,
— le critère d'Evans,
— le critère du revers,
— le critère du contour de Hall.

On n'aborde ici ni les énoncés ni les démonstrations de la validité de ces critères. Une fois un critère choisi, il est facile de décider si un filtre est stable ou non.

5.3.5 Représentation en schéma-bloc

Le schéma fonctionnel, appelé aussi schéma-bloc, schéma de principe ou en anglais *block diagram*, est la représentation graphique simplifiée d'un procédé relativement complexe impliquant plusieurs unités ou étapes. Il est composé de blocs connectés par des lignes d'action. Il est utilisé principalement en automatique, en traitement du signal, en génie chimique et en fiabilité.

Il est très pratique de représenter les filtres sous forme de *schéma-bloc*. C'est à la fois très intuitif, facile à former à partir de l'équation de définition du filtre et proche de ce qui serait un circuit électronique réalisant effectivement le filtre.

Dans notre contexte, on utilisera des blocs élémentaires de décalage unitaire et des opérations de sommation (ou soustraction). L'entrée du schéma est x_n et

la sortie est y_n. Avec ces simples éléments, on est alors capable de représenter graphiquement les filtres récursifs causaux de ce cours.

Considérons de nouveau l'exemple de filtre défini par (5.1), que nous rappelons ici :

$$y_n = \frac{1}{4}y_{n-1} + \frac{1}{2}x_n + \frac{1}{2}x_{n-1}.$$

La figures 5.2 est une des représentations possibles sous forme de *schéma-bloc*.

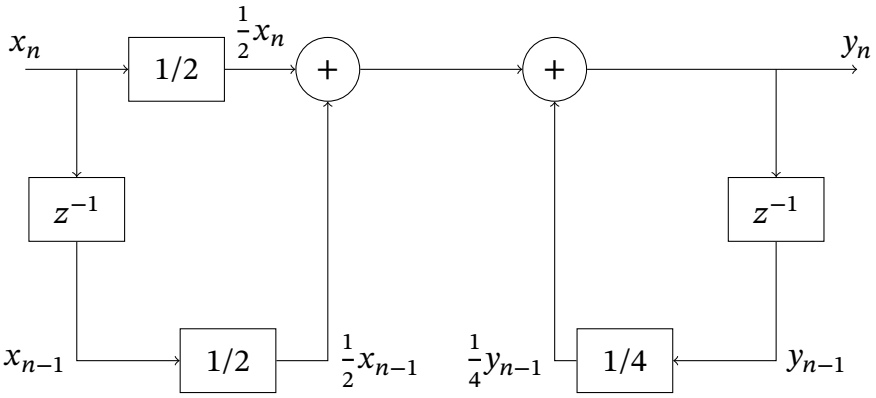


FIGURE 5.2 – Une représentation du filtre défini par $y_n = \frac{1}{4}y_{n-1} + \frac{1}{2}x_n + \frac{1}{2}x_{n-1}$.

N'importe quel filtre récursif causal peut-être représenté par un schéma-bloc, cependant, la représentation n'est évidemment pas unique.

5.4 Réponse fréquentielle

On va maintenant présenter un lien entre transformée en z et série de Fourier. Ce lien avait déjà été entrevu lorsque l'on avait utiliser un argument de convolution dans la preuve du lemme 5.2

5.4.1 Définition

La *réponse fréquentielle* d'un filtre (linéaire, récursif ou non, causal ou non) donne des informations sur la manière dont le filtre réagit aux excitations périodiques.

DÉFINITION 5.13 On appelle réponse fréquentielle d'un filtre la fonction :

$$\begin{array}{rcl} [-\pi,\pi] & \to & \mathbf{C} \\ \omega & \mapsto & H(\mathrm{e}^{\mathrm{i}\omega}). \end{array}$$

Cette fonction est souvent, par abus de notations, simplement notée $H(\omega)$.

On remarque que $H(\mathrm{e}^{i\omega})$ est la série de Fourier dont les coefficients sont ceux de la réponse impulsionnelle. La réponse en fréquence est donc la transformée de Fourier de la suite $(h_n)_{n\in\mathbf{Z}}$.

On se place dans le cadre d'application du théorème d'échantillonnage de Shannon, et on suppose que le contenu fréquentiel du signal d'entrée est inclus dans $[-f_e/2, f_e/2]$ où f_e est la fréquence d'échantillonnage.

De plus, comme on se place après la phase d'échantillonnage, bien qu'il n'apparaisse plus, le temps entre x_n et x_{n+1} est $1/f_e$ (et doit être transmis).

On a alors la correspondance suivante :

$$\omega = 2\pi\frac{f}{f_e}$$

qui appartient à $[-\pi,\pi]$ quand $f \in [-f_e/2, f_e/2]$. Le passage par ω permet de normaliser la réponse fréquentiel. Le retour à la réponse fréquentielle non normalisée se fait par une simple multiplication pour la fréquence d'échantillonnage f_e.

Remarque 5.8 :

— On note que $\omega \mapsto H(\mathrm{e}^{\mathrm{i}\omega})$ est 2π-périodique.

— Sans rentrer dans le détail mathématique, dans le cas où l'anneau de convergence inclut le cercle unité, l'intégrale d'inversion de la transformée en z peut se faire sur le cercle unité. On a alors

$$x_k = \frac{1}{2\pi}\int_0^{2\pi} X(\mathrm{e}^{\mathrm{i}\omega})\mathrm{e}^{\mathrm{i}k\omega}\mathrm{d}\omega.$$

5.4.2 Modification spectrale du signal d'entrée

Grâce à ce formalisme mathématique, on arrive à l'idée du traitement du signal par modification spectral. En effet, pour les filtre linéaires récursifs causaux, la relation :

$$H(z) = \frac{Y(z)}{X(z)} \Leftrightarrow Y(z) = H(z)X(z),$$

nous donne dans le domaine fréquentiel

$$Y(\mathrm{e}^{\mathrm{i}\omega}) = H(\mathrm{e}^{\mathrm{i}\omega})X(\mathrm{e}^{\mathrm{i}\omega}).$$

Ainsi, dans le domaine fréquentiel, le spectre de Y est le produit des deux spectres de H et de X. Ainsi, pour réaliser les traitements que l'on souhaite, il faudra être capable de construire la fonction H correspondante.

Exercice 5.6 :

Soit le filtre numérique défini par

$$y_n = \sum_{k\in\mathbf{N}} a^k x_{n-k}, \tag{5.4}$$

où la suite (x_n) est le signal d'entrée et (y_n) le signal de sortie, et où $0 < a < 1$.

1. Montrer que le filtre peut se mettre sous forme récursive. On pourra passer par la transformer en z du filtre (c'est-à-dire de l'égalité (5.4)).
2. Déterminer la réponse impulsionnelle du filtre, c'est-à-dire la réponse $(h_n)_{n\geq 0}$ au signal d'entrée $(x_n)_{n\geq 0}$ défini par $x_0 = 1$ et $x_n = 0$ pour tout $n > 0$.
3. À quelle condition sur la réponse impulsionnelle le filtre est-il stable ? Est-ce que cela est vérifié ?
4. À quelle condition sur la fonction de transfert le filtre est-il stable ? Est-ce que cela est vérifié ?
5. Pour $a = 1/2$, étudier la réponse fréquentielle du filtre. On rappelle que la réponse fréquentielle du filtre est définie à partir de sa fonction de transfert $H(z)$ par $|H(\mathrm{e}^{\mathrm{i}\omega})|$ où ω est la pulsation normalisée, c'est-à-dire que $\omega = 2\pi f/f_e$ où f est la fréquence et f_e la fréquence d'échantillonnage. On supposera que le signal a été échantillonné en validant les hypothèses du théorème de Shannon. Déterminer alors la nature du filtre.

5.4.3 Réponse à phase linéaire

Dans les applications, il est utile que la *phase* du filtre, c'est-à-dire l'argument de la fonction de transfert, soit une fonction affine de ω. Le filtre est alors appelé abusivement *filtre à phase linéaire*. Expliquons rapidement pourquoi. On considère un signal périodique simple, de la forme $t \mapsto \mathrm{e}^{i\omega t}$. Au cours du temps, on voit que la phase varie de manière linéaire, c'est-à-dire que l'effet du temps sur la phase du signal est un déphasage proportionnel à t. Lorsque l'on transmet ce signal, on ne peut espérer qu'à la réception le déphasage soit nul, ce qui correspondrait au cas idéal d'une réception en temps réel. En pratique, on reçoit un signal de la forme $t \mapsto \mathrm{e}^{i\left(\omega t+\phi(\omega)\right)} = \mathrm{e}^{i\omega\left(t-\tau(\omega)\right)}$, où τ est un retard engendré par le canal de transmission. Notons $\phi(\omega) = \arg\Big(H(\mathrm{e}^{i\omega})\Big)$ la phase d'un filtre donné (de fonction de transfert H) et considérons deux signaux $t \mapsto \mathrm{e}^{i\omega_1 t}$ et $t \mapsto \mathrm{e}^{i\omega_2 t}$. L'effet du temps sur le déphasage entre les deux signaux est d'introduire un déphasage proportionnel à $\omega_2 - \omega_1$. Dans de nombreuses situations, on souhaite qu'un filtre n'affecte pas plus la phase que ne le ferait le temps, c'est-à-dire que le déphasage entre deux signaux qu'il reçoit soit proportionnel à la différence de leurs phases, ce qui conduit à l'équation :

$$\phi(\omega_2) - \phi(\omega_1) = \alpha(\omega_2 - \omega_1),$$

autrement dit, la fonction de transfert du filtre doit être de la forme :

$$H(\mathrm{e}^{i\omega}) = |H(\mathrm{e}^{i\omega})|\mathrm{e}^{i(\alpha\omega+\beta)}.$$

Dans le cas où le filtre n'est pas à phase linéaire, la déformation qu'il engendre sur le signal d'entrée est appelée *distorsion de phase*. Ce défaut ne doit pas être confondu avec la *distorsion harmonique* qui intervient lorsque le filtre n'est pas linéaire.

Exercice 5.7 :

Étudier la réponse fréquentielle du filtre défini par la réponse impulsionnelle suivante :

$$h = \{0,5; 0; 0; 0; 0; 0; 0; 0; 0; 0,5\}.$$

5.4.4 Diagramme de Bode

Le diagramme de Bode est un moyen de représenter la réponse en fréquence d'un système dont la fonction de transfert se met sous la forme d'une fraction

rationnelle. Il est adapté à la représentation des filtres récursifs causaux.

Le diagramme de Bode d'un système de réponse fréquentielle $H(\omega)$ se compose de deux tracés :

— le gain (ou amplitude) en décibels (dB). Sa valeur est calculée à partir de $20 \log_{10} |H(\omega)|$. ;

— la phase en degré, donnée par $\arg(H(\omega)$.

L'échelle des pulsations est logarithmique et est exprimée en rad/s (radian par seconde). L'échelle logarithmique permet un tracé très lisible, car construit à partir de tronçons de ligne droite.

Par exemple, le diagramme de Bode de la fonction de l'exercice 5.4 :

$$H(z) = \frac{1}{1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}} \tag{5.5}$$

est donné en figure 5.3.

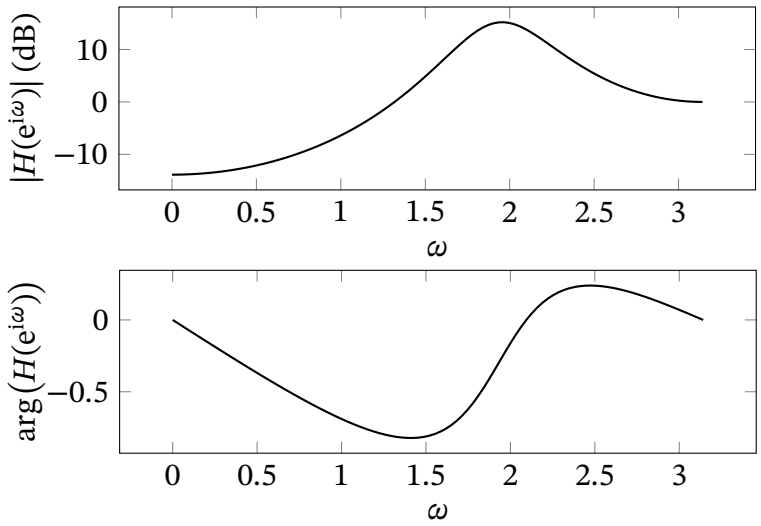


FIGURE 5.3 – Diagramme de Bode de la fonction de transfert définie par l'équation 5.5.

Suivant les puissances des polynômes du dénominateur et du numérateur, il y a des classifications standards de ces filtres.

5.5 SYNTHÈSE DE FILTRE

Dans cette section, nous allons effleurer la thématique de la synthèse de filtre, c'est-à-dire la conception de filtres numériques.

5.5.1 Filtres idéaux et gabarits

On considérera ici que trois types de filtre :

— les filtres *passe-bas*, qui ne laissent passer que les basses fréquences ;
— les filtres *passe-haut*, qui ne laissent passer que les hautes fréquences ;
— les filtre *passe-bande*, qui ne laissent passer qu'une bande fréquentielle.

On appelle *fréquence de coupure* toute fréquence autour de laquelle le filtre change de comportement. La qualité d'un filtre se mesure bien souvent à ses qualité de coupure, où l'objectif est souvent de pour passer idéalement de 1 à 0 instantanément (en fréquence).

DÉFINITION 5.14 — (Gabarit) On appelle *gabarit* l'ensemble des contraintes que l'on souhaite pour le filtre à réaliser.

En figure 5.4, un exemple gabarit type d'un filtre passe-bas est présenté. On a en rouge le filtre idéal que l'on souhaite réaliser avec des paramètres de tolérance :

— Δf est la *zone* de coupure autour de la fréquence de coupure souhaité, notée f_c, pendant laquelle la transition doit être effectuée ;
— Δe_1 et Δe_2 sont les tolérances autour de respectivement les valeurs 1 et 0.

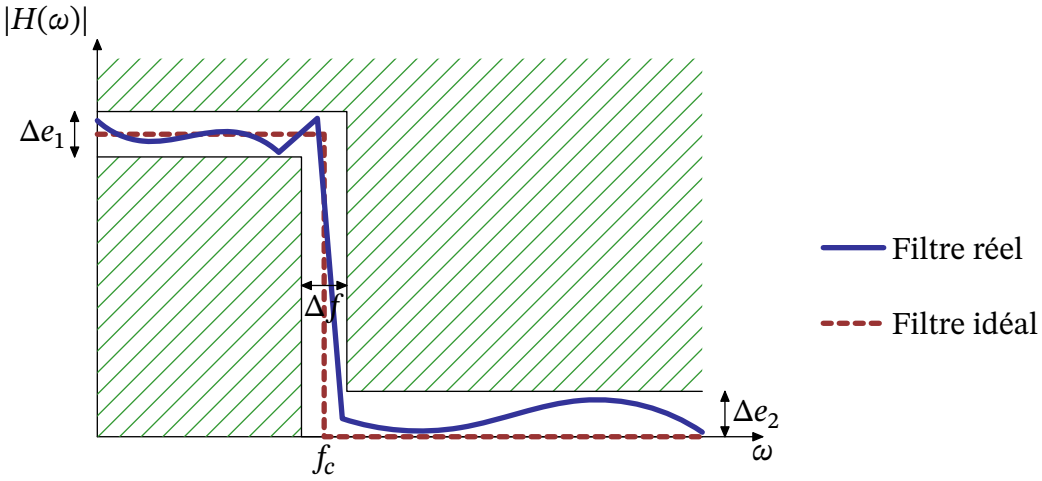


FIGURE 5.4 – Illustration d'un gabarit d'un filtre et ses paramètres de tolérance.

Nous n'aborderons dans ce cours que la synthèse de filtre à réponse impulsionnelle finie.

5.5.2 Conception de filtres à RIF

Dans cette section, nous abordons rapidement la conception de filtres à réponse impulsionnelle finie (RIF). Pour ce type de filtre, $h = (h_n)_{n\in\mathbf{Z}}$ est à support fini, donc, un tel filtre est nécessairement stable.

Méthode de la fenêtre

La méthode de la fenêtre est la suivante. On choisit une réponse fréquentielle *idéale* notée $H^*(\omega)$. Ce choix peut-être fait par exemple par le graphe de la fonction dans le domaine fréquentiel.

Par ailleurs, on a

$$H^*(\omega) = \sum_{n\in\mathbf{Z}} h_n^* \mathrm{e}^{\mathrm{i}\omega n}$$

avec, grâce à la formule de transformée inverse :

$$h_n^* = \frac{1}{2\pi}\int_{-\pi}^{\pi} H^*(\omega)\mathrm{e}^{\mathrm{i}\omega n}\mathrm{d}\omega.$$

Malheureusement, rien ne garantit que la suite (h_n^*) correspondante soit à support fini. On va donc tronquer la suite de coefficients (h_n^*) à un certain rang $M-1$, c'est-à-dire multiplier (h_n^*) par une suite *fenêtre* (g_k) défini par :

$$g_k = \begin{cases} 1 \text{ si } k = 0, \dots, M-1 \\ 0 \text{sinon.} \end{cases}.$$

Ce procédé revient à définir un nouveau filtre (h_k) par l'opération :

$$h_k = h_k^* g_k.$$

Pour avoir la réponse fréquentielle du filtre réel obtenue, on prend la transformée de Fourier du produit :

$$H(\omega) = H^* \star G(\omega),$$

c'est-à-dire

$$H(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H^*(\omega)G(\omega-\nu)\mathrm{d}\nu,$$

et

$$G(\omega) = \sum_{k=0}^{M-1} \mathrm{e}^{-\mathrm{i}k\omega}.$$

Malheureusement, si cette méthode est assez simple à comprendre et à mettre en place, elle souffre du phénomène de Gibbs (voir figure 5.5).

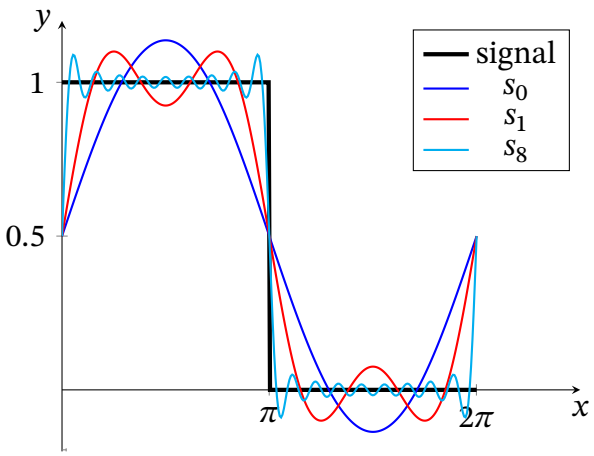


FIGURE 5.5 – Illustration du phénomène de Gibbs sur la fonction créneau.

Remarque 5.9 :

Pour améliorer cette méthode, il est possible d'utiliser des fonctions fenêtre plus régulières comme les fenêtre de Hamming, Barlett, etc.

5.5.3 Synthèse par transformation de Fourier discrète

5.6 NOTES BIBLIOGRAPHIQUES

Pour une introduction plus complète des filtres numériques, on pourra consulter les chapitres 2, 3, 4, 5 et 9 de la référence [6]. Pour un exposé plus court et en français, on pourra se référer à [12].

CHAPITRE 6

L'analyse temps/fréquences

SOMMAIRE DU CHAPITRE

Dans ce chapitre nous aborderons l'analyse temps/fréquences. Nous avons vu jusque là que l'outil d'analyse de Fourier permet d'obtenir beaucoup d'informations sur un signal donnée. Nous avons vu cela dans le cadre de l'espace fonctionnel L^1, dit des signaux stables. Nous allons ici voir que cette transformée s'étend à l'espace L^2 dans lequel elle est une bijection. La théorie de Fourier est mathématiquement très belle et très utile mais il n'est pas entièrement satisfaisant pour des applications importantes du traitement du signal.

En effet, lorsqu'on considère un signal sonore, par exemple un morceau de musique, les notes sont à la fois localisées en temps et en fréquence. La transformée de Fourier n'est pas adapté à cette analyse, car pour une fréquence

donnée (une note), elle est reliée à l'énergie totale de *toutes les occurrences* de cette note dans tout le morceau.

Après avoir donner rapidement des résultats sur la transformée de Fourier dans l'espace fonctionnel L^2, on introduira la transformée de Fourier à fenêtre, puis nous introduirons la transformée en ondelettes.

6.1 Transformées de Fourier des fonctions L^2

Comme nous avons dit en remarque 1.2, on peut considérer plusieurs cadre fonctionnel pour la transformée de Fourier. Dans cette section, nous nous intéresserons aux résultats propre à l'espace de fonction L^2. On ne démontrera que peu de ces résultats et nous renvoyons à [**ma102**, 3] pour plus de détails.

6.1.1 Définition et propriétés de L^2

On désignera par Ω un ouvert de $\mathbf{R}^n$.

Définition 6.1 — (Espace $L^2(\Omega)$) Une fonction définie sur Ω à valeurs complexes est dite de carré intégrable si u est mesurable et $u^2 \in L^1(\Omega)$. On pose alors

$$\|u\|_{L^2} = \left(\int_\Omega |u(x)|^2 \, \mathrm{d}x\right)^{1/2}.$$

On note $L^2(\Omega)$ l'ensemble des fonctions de carré intégrable sur Ω.

Proposition 6.1 : *L'ensemble $L^2(\Omega)$ satisfait les propriétés suivantes :*

1. *$L^2(\Omega)$ est un espace vectoriel et $\|\cdot\|_{L^2}$ est une norme sur $L^2(\Omega)$. En particulier, si $u \in L^2(\Omega)$ et $v \in L^2(\Omega)$, alors la somme $u + v$ appartient à $L^2(\Omega)$ et on a l'*inégalité de Minkowski

$$\|u + v\|_{L^2} \leq \|u\|_{L^2} + \|v\|_{L^2}.$$

2. *Si $u \in L^2(\Omega)$ et $v \in L^2(\Omega)$, alors le produit uv appartient à $L^1(\Omega)$ et on a l'inégalité de Cauchy-Schwarz :*

$$\int_\Omega |u(x)v(x)| \, \mathrm{d}x \leqslant \left(\int_\Omega |u(x)|^2 \, \mathrm{d}x\right)^{1/2} \left(\int_\Omega |v(x)|^2 \, \mathrm{d}x\right)^{1/2}.$$

3. *L'application*

$$\langle u,\ v\rangle_{L^2} : (u,v) \mapsto \int_\Omega u(x)\overline{v}(x)\mathrm{d}x$$

est un produit scalaire sur $L^2(\Omega)$.

On peut alors montrer le résultat suivant :

THÉORÈME 6.1 Soit (f_n) une suite de Cauchy dans $L^2(\Omega)$. Alors il existe $f \in {}^2(\Omega)$ tel que

$$\|f - f_n\|_{L^2} \to 0 \text{ quand } n \to +\infty.$$

L'espace $L^2(\Omega)$ est dont complet pour la topologie déduite de la norme :

$$\|f\|_{L^2} = \langle f,\ f\rangle_{L^2}^{1/2}.$$

Ce théorème montre que $L^2(\Omega)$ est un espace de Hilbert.

6.1.2 Transformation de Fourier

Dans le cadre de ce cours, Ω sera $\mathbf{R}$ ou $\mathbf{R}^2$ (mais tous les résultats s'étendent aisément à $\mathbf{R}^n$). Remarquons tout d'abord qu'une fonction f de carré intégrable dans $\mathbf{R}$ appartient à L^1_{loc}. En effet, pour tout compact $A \in \mathbf{R}$, on a, d'après l'inégalité de Cauchy-Schwarz :

$$\int_A |f(x)|\,\mathrm{d}x = \int_{\mathbf{R}} \mathbf{1}_A(x)\,|f(x)|\,\mathrm{d}x \leqslant \|f\|_{L^2}\,|A|\,.$$

En revanche, on peut trouver des fonction L^2 qui n'appartiennent pas à L^1, et donc, les définitions établies au chapitre 1, section 1.1.1 ne s'appliquent pas.

Pour définir la transformée de Fourier dans L^2 on utilise le théorème suivant (qu'on ne démontrera pas).

THÉORÈME 6.2 Soit $u \in L^1 \cap L^2$, alors $\mathcal{F}(u) \in L^2$ et

$$\int_{\mathbf{R}} |u(t)|^2\,\mathrm{d}t = \int_{\mathbf{R}} |\mathcal{F}(u)(\nu)|^2\,\mathrm{d}\nu.$$

Grâce à ce théorème, l'application $\phi : u \mapsto \mathcal{F}(u)$ de $L^1 \cap L^2$ dans L^2 est linéaire et est une isométrie. Ensuite, par *densité* de $L^1 \cap L^2$ dans L^2, cette

isométrie linéaire peut être étendue de manière unique à L^2 tout entier. On continuera à noter $\hat{s}$ ou $\mathcal{F}(s)$ la transformée de Fourier sur L^2.

L'isométrie dont on vient juste de parler est donnée par le théorème suivant.

THÉORÈME 6.3 — (de Plancherel) Si s_1 et s_2 sont deux fonctions de L^2, alors

$$\int_{\mathbf{R}} s_1(t)\overline{s_2}(t)\mathrm{d}t = \int_{\mathbf{R}} \widehat{s_1}(\nu)\overline{\widehat{s_2}}(\nu)\mathrm{d}\nu,$$

ou encore

$$\langle s_1,\ s_2\rangle_{L^2} = \langle \widehat{s_1},\ \widehat{s_2}\rangle_{L^2}.$$

Et on peut aussi démontrer un résultat concernant le produit de convolution.

THÉORÈME 6.4 Soit $h \in L^1$ et $x \in L^2$, alors

$$y(t) = \int_{\mathbf{R}} h(t-s)x(s)\mathrm{d}s$$

est défini presque partout et appartient à L^2. De plus, sa transformée de Fourier est

$$\forall \nu \in \mathbf{R}, \hat{y}(\nu) = \hat{h}(\nu)\hat{x}(\nu).$$

On peut montrer alors le résultat suivant.

THÉORÈME 6.5 La transformation de Fourier $\mathcal{F}$ est une isométrie bijective de L^2 dans lui-même, d'inverse $\overline{\mathcal{F}}$.

Remarque 6.1 :

Dans la construction de la transformée de Fourier dans L^2 on passe par l'ensemble $L^1 \cap L^2$. En pratique, pour calculer la transformée de Fourier d'une fonction de L^2 qui n'est pas dans L^1, on utilisera le résultat qui dit que pour $f \in L^2$, on a $\widehat{f}$ comme limite dans L^2 lorsque $n \to +\infty$ de

$$\int_{|t|\leqslant n} f(t)\mathrm{e}^{-2\mathrm{i}\pi\nu t}\mathrm{d}t.$$

6.2 La transformée de Fourier à fenêtre

Comme expliqué en introduction de ce chapitre, la problématique à laquelle on s'attaque ici est de pouvoir obtenir de l'information en temps/fréquence mais localisée. L'amplitude de transformée de Fourier nous donne, dans le cas d'un morceau de piano par exemple, pour une note de fréquence ν_f sera reliée à toutes les occurrences de cette note dans tout le morceau.

Ceci a amené Dennis Gabor [1] à proposer l'idée d'une transformée de Fourier à fenêtre.

6.2.1 Fonction fenêtre

Pour un signal f à analyser, l'information locale *autour* du temps $t = b$ est contenue dans le signal *localisé en temps*

$$\forall t, \quad t \mapsto f(t)w(t-b) \tag{6.1}$$

où w est une *fonction fenêtre* temporelle *paire* qui est nulle, ou *suffisamment* proche de zéro, en dehors d'un petit intervalle autour de zéro. Donnons deux exemples classiques pour se fixer les idées.

La fenêtre rectangulaire : $w : t \mapsto w(t) = \mathbf{1}_{[-\alpha,\alpha]}(t)$ où $\alpha > 0$;

La fenêtre gaussienne : $w : t \mapsto w(t) = \mathrm{e}^{-\alpha t^2}$ où $\alpha > 0$.

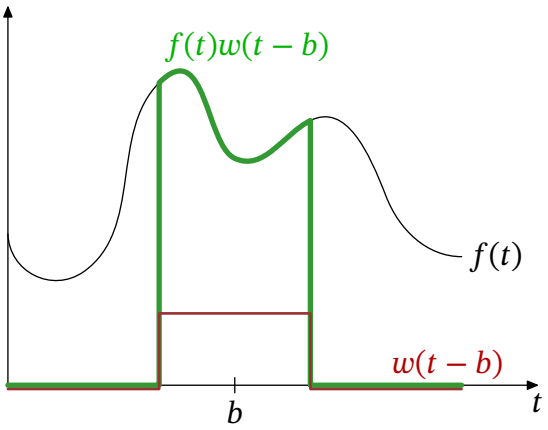


FIGURE 6.1 – Illustration de l'action d'une fenêtre rectangulaire sur une fonction.

1. Dennis Gabor (5 juin 1900 à Budapest, Hongrie - 8 février 1979 à Londres) est un ingénieur et physicien hongrois. Il est notamment connu pour l'invention de l'holographie pour laquelle il a reçu le prix Nobel de physique de 1971.

Ayant une fonction fenêtre w, l'information spectrale locale au temps b est obtenue en calculant la transformée de Fourier de (6.1) que l'on notera :

$$W_f(\nu, b) = \int_{\mathbf{R}} f(t)w(t-b)\mathrm{e}^{-2\mathrm{i}\pi\nu t}\mathrm{d}t. \tag{6.2}$$

DÉFINITION 6.2 — (Transformée de Fourier à fenêtre) Soient w et f deux fonctions L^2, où w est une fonction fenêtre. On appelle la fonction $W_f : \mathbf{R}\times\mathbf{R} \to \mathbf{C}$ définie par (6.2) la transformée de Fourier à fenêtre de f associée à la fenêtre w.
Si la fenêtre est la fenêtre rectangulaire, on parlera de *transformée de Fourier en temps court*, et si la fenêtre est gaussienne, on parlera de *la transformée de Gabor* de f.

6.2.2 Formule d'inversion

On énoncera ici un théorème de reconstruction du signal à partir de la transformée de Fourier à fenêtre, dont la preuve dépasse le cadre de ce cours (voir [3, 5]).

THÉORÈME 6.6 Supposons :
- $w \in L^1 \cap L^2$;
- $\int_{\mathbf{R}} |w(t)|^2 \,\mathrm{d}t = 1$;
- $|\widehat{w}|$ est une fonction paire.

On a alors, pour tout $f \in L^2$, la formule de la conservation de l'énergie

$$\int_{\mathbf{R}}\int_{\mathbf{R}} |W_f(\nu, b)|^2 \,\mathrm{d}\nu\mathrm{d}b = \int_{\mathbf{R}} |f(t)|^2 \,\mathrm{d}t.$$

De plus, on a la formule de reconstruction (ou formule d'inversion) :

$$\lim_{A\to+\infty} \int_{\mathbf{R}} \left| f(t) - \int_{|\nu|\leqslant A}\int_{\mathbf{R}} W_f(\nu, b)\overline{w}(t-b)\mathrm{e}^{2\mathrm{i}\pi\nu t}\mathrm{d}\nu\mathrm{d}b \right|^2 \mathrm{d}t.$$

Preuve : (difficile) À ÉCRIRE ■

Ce résultat montre que pour la transformée de Fourier à fenêtre dans L^2, nous avons des formules analogues à celles pour la transformée de Fourier dans L^2 : la conservation de l'énergie, et la formule d'inversion.

Remarque 6.2 :

— En notant
$$w_{\nu,b}(t) = \overline{w}(t-b)\mathrm{e}^{2\mathrm{i}\pi\nu t},$$
on a alors $W_f(\nu, b) = \langle f,\ w_{\nu,b}\rangle_{L^2}$.

— D'après le théorème de Plancherel, on a alors
$$W_f(\nu, b) = \langle f,\ w_{\nu,b}\rangle_{L^2} = \langle \widehat{f},\ \widehat{w}_{\nu,b}\rangle_{L^2}.$$
Ce résultat nous dit que nous aurions pu aussi définir la fonction W_f à partir de la transformée de Fourier de f.

6.2.3 Le principe d'incertitude

Nous allons dans un premier temps voir ce qui nous permettra de *quantifier* la notion intuitive de « largeur » d'une fonction, à la fois dans le domaine temporel et dans le domaine spectrale.

Nous verrons ensuite ce que ces notions nous permettent de comprendre sur la transformée de Fourier à fenêtre et ses limitations.

Localisation temps-fréquence

Commençons par définir le centre d'une fonction.

DÉFINITION 6.3 — (Centre d'une fonction) On définit et on note m_w le centre d'une fonction w comme la valeur, quand elle est finie, de :
$$m_w = \frac{1}{\|w\|_{L^2}^2} \int_{\mathbf{R}} t\,|w(t)|^2\,\mathrm{d}t.$$

Remarque 6.3 :

Quand la fenêtre w est paire, et donc $m_w = 0$, $w_{\nu,b}(t) = \overline{w}(t-b)\mathrm{e}^{2\mathrm{i}\pi\nu t}$ est centré sur b. Le produit $f\ \overline{w}_{\nu,b}$ isole donc les composantes de f au voisinage de b.

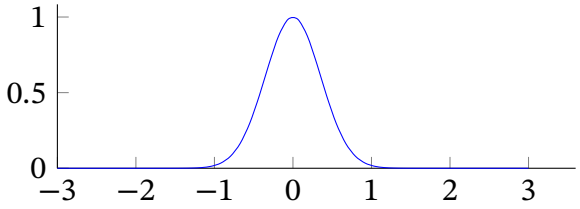
On définit la notion d'étalement en temps.

DÉFINITION 6.4 — (Étalement en temps) Avec les notations qui précèdent on définit l'étalement en temps de la fonction fenêtre par

$$\sigma_w = \left(\frac{1}{\|w\|_{L^2}^2} \int_{\mathbf{R}} (t - m_w)^2 \, |w(t)|^2 \, \mathrm{d}t \right)^{1/2}.$$

Exemple

Pour la gaussienne $w : t \mapsto \mathrm{e}^{-4t^2}$, on a $\|w\|_{L^2}^2 = \frac{1}{4}\sqrt{2\pi}$, et $\sigma_w = \frac{1}{4}$.



Il est assez aisé de vérifier que dans le cas d'une fenêtre centrée en 0, on a :

$$\forall b \in \mathbf{R}, \quad \sigma_w = \left(\frac{1}{\|w\|_{L^2}^2} \int_{\mathbf{R}} (t - b)^2 \, |w_{\nu,b}(t)|^2 \, \mathrm{d}t \right)^{1/2}.$$

Exercice 6.1 :

Montrer que

$$\widehat{w}_{\nu,b}(\omega) = \mathrm{e}^{-2\mathrm{i}\pi b(\omega+\nu)} \widehat{w}(\omega + \nu).$$

On définit la notion analogue d'étalement en fréquence.

DÉFINITION 6.5 — (Étalement en fréquence) Avec les notations qui précèdent on définit l'étalement en fréquence de la transformée de Fourier de fonction fenêtre par

$$\sigma_{\widehat{w}} = \left(\frac{1}{\|\widehat{w}\|_{L^2}^2} \int_{\mathbf{R}} (\nu - m_{\widehat{w}})^2 \, |\widehat{w}(\nu)|^2 \, \mathrm{d}\nu \right)^{1/2},$$

où $m_{\widehat{w}}$ est le centre de $\widehat{w}$.

Là encore, il est assez aisé de vérifier que, dans le cas où $m_{\widehat{w}} = 0$ et que

$$\forall b \in \mathbf{R}, \quad \sigma_{\widehat{w}} = \left(\frac{1}{\|\widehat{w}\|_{L^2}^2} \int_{\mathbf{R}} (\nu - \omega)^2 \left| \widehat{w}_{\nu,b}(\nu) \right|^2 \mathrm{d}\nu \right)^{1/2}.$$

Boîte de Heisenberg. Dans le plan temps-fréquence (t, ω), on représente $w_{\nu,b}$ par une *boîte de Heisenberg* de taille $\sigma_w \times \sigma_{\widehat{w}}$, centré en (b, ν). La taille de la boîte ne dépend pas de (b, ν), ce qui veut dire que la résolution de la transformée de Fourier à fenêtre reste constante sur tout le plan temps-fréquence. On peut voir une illustration de ce concept en figure 6.2.

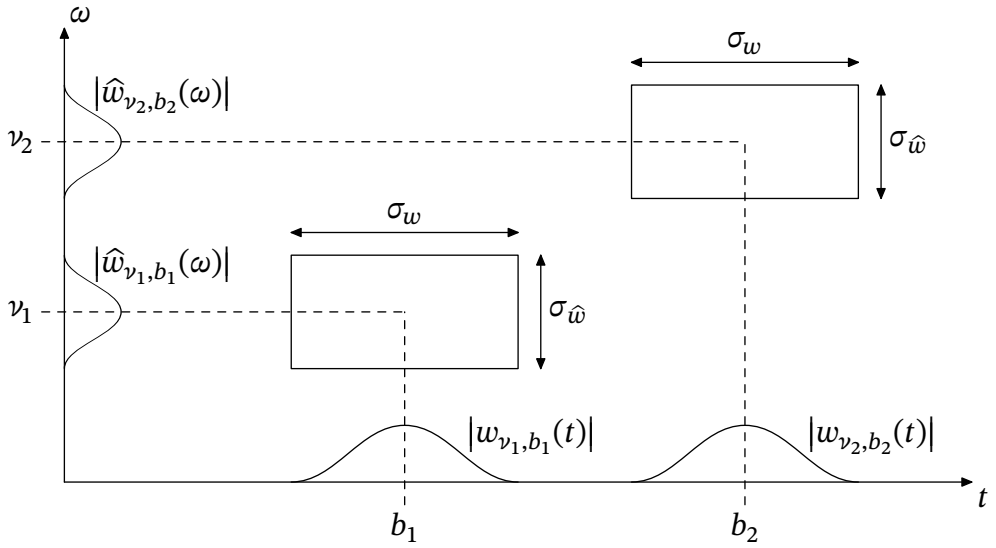


FIGURE 6.2 – Illustration du concept de boîte de Heisenberg pour la transfourmée de Fourier à fenêtre.

Incertitude de Heisenberg

Pour mesurer les composantes de la fonction f à étudier dans les petits voisinages de b et ν, il faut construire une fenêtre w qui est bien localisée dans le temps, et dont l'énergie de la transformée de Fourier est concentrée dans un petit domaine fréquentiel. Nous avons vu avec l'exemple de la transformée de Fourier du créneau (exercice 1.1), un phénomène général : moins une fonction temporelle est étalée en temps, plus sa transformée de Fourier l'est en fréquence. Ce phénomène peut aussi être illustré par le fait d'opérer un changement d'échelle de temps d'un facteur $a > 1$, c'est-à-dire considérer la fonction w_a : $t \mapsto w(at)$ dont la transformée de Fourier est, d'après la proposition 1.1 du

chapitre 1, $\widehat{w}_a : \nu \mapsto \frac{1}{a}\widehat{w}\left(\frac{\nu}{a}\right)$. Il y a donc un compromis entre la localisation en temps et celle en fréquence.

Les concentration en temps et en fréquences sont limitées par le principe d'Heisenberg [2]. Le théorème suivant montre que le produit $\sigma_w \times \sigma_{\widehat{w}}$ ne peut être arbitrairement petit.

THÉORÈME 6.7 — (Incertitude de Heisenberg) On suppose que $w \in L^2$ est une fonction centrée en 0 et dont la transformée de Fourier est aussi centrée en 0 :

$$\int_{\mathbf{R}} t\,|w(t)|^2\,\mathrm{d}t = \int_{\mathbf{R}} \nu\,|\widehat{w}(\nu)|^2\,\mathrm{d}\nu = 0.$$

Alors on a l'inégalité de Heisenberg suivante :

$$\sigma_w\,\sigma_{\widehat{w}} \geqslant \frac{1}{4\pi}. \tag{6.3}$$

Preuve : Nous ne démontrerons ici le résultat que pour les fonctions fenêtre w $\mathcal{C}^\infty$ à support compact, le résultat s'obtenant dans le cas général par des outils de densité, qui dépasse le cadre de ce cours.

On doit montrer que

$$\|tw\|_{L^2}\,\|\nu\widehat{w}\|_{L^2} \geqslant \frac{1}{4\pi}\,\|w\|_{L^2}^2\,.$$

On a d'après le théorème 1.2 (qui s'étend aux fonction L^2) :

$$\widehat{w'}(\nu) = 2\mathrm{i}\pi\nu\widehat{w},$$

et donc, en utilisant le théorème 6.3,

$$\|\nu\widehat{w}\|_{L^2}^2 = \frac{1}{4\pi^2}\left\|\widehat{w'}\right\|_{L^2}^2 = \frac{1}{4\pi^2}\,\|w'\|_{L^2}^2\,.$$

Il reste alors à montrer que

$$\|tw\|_{L^2} \times \|w'\|_{L^2} \geqslant \frac{1}{2}\,\|w\|_{L^2}^2\,. \tag{6.4}$$

Par l'inégalité de Cauchy-Schwarz, on a :

$$\|tw\|_{L^2}\,\|w'\|_{L^2} \geqslant |\langle tw,\ w'\rangle| \geqslant |\mathrm{Re}\{\langle tw,\ w'\rangle\}|\,.$$

On a alors :

$$\begin{aligned} 2\,\mathrm{Re}\{\langle tw,\ w'\rangle\} = \langle tw,\ w'\rangle + \langle w',\ tw\rangle &= \int_{\mathbf{R}} t(w\overline{w'} + w'\overline{w})\mathrm{d}t \\ &= \left[t\,|w(t)|^2\right]_{-\infty}^{+\infty} - \int_{\mathbf{R}} |w(t)|^2\,\mathrm{d}t = 0 - \|w\|_{L^2}^2\,, \end{aligned}$$

2. Ce principe d'incertitude a un rôle crucial en mécanique quantique.

ce qui permet de démontrer 6.4 et donc l'inégalité de Heisenberg pour les fenêtres $\mathcal{C}^\infty$ à support compact. ■

On a alors la proposition suivante.

Proposition 6.2 : *L'égalité dans le théorème 6.7 est obtenu si et seulement si la fenêtre w est proportionnelle à un signal gaussien $t \mapsto \mathrm{e}^{-ct^2}$ avec $c > 0$.*

La fenêtre de Gabor est donc optimal au sens où elle minimise l'incertitude $\sigma_w\ \sigma_{\widehat{w}}$.

Preuve : Pour qu'il y ait égalité dans la preuve du théorème 6.7, il faut et il suffit qu'il y ait égalité dans l'inégalité de Cauchy-Schwarz, c'est-à-dire que $tw(t)$ et $w'(t)$ soient proportionnelles :

$$\exists c \in \mathbf{R}, \quad w'(t) = -ctw(t),$$

ce qui donne

$$w(t) = A\mathrm{e}^{-ct^2}$$

où $c > 0$ car $w \in L^2$. Le reste de la preuve se poursuit sans difficulté. ■

6.2.4 Spectrogramme

DÉFINITION 6.6 — (Spectrogramme) On peut associer à la transformée de Fourier à fenêtre une densité d'énergie qu'on appelle *spectrogramme* que l'on définit comme suit. Pour fonction f et toute fenêtre w satisfaisant les hypothèses précédentes, on appelle spectrogramme, et on note $P_W f$ la fonction :

$$(\nu, b) \mapsto \left|W_f(\nu, b)\right|^2 = \left|\int_{\mathbf{R}} f(t)w(t-b)\mathrm{e}^{-2\mathrm{i}\pi\nu t}\mathrm{d}t\right|^2.$$

Le spectrogramme mesure l'énergie de f et $\widehat{f}$ dans le voisinage temps-fréquence ou l'énergie de $w_{\nu,b}$ est concentrée. Le code python suivant, grâce à la bibliothèque `librosa`, permet de tracer le spectrogramme obtenu par la calcul de la transformée de Fourier à fenêtre avec une fenêtre à temps court (*Short Time Fourier Transform* : `stft`).

```
import librosa
import matplotlib.pyplot as plt
import librosa.display
from scipy.io import wavfile
```

```
x, sr = librosa.load('musique.wav', sr=None)
X = librosa.stft(x)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(14, 5))
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='
    hz')

plt.savefig('spectrogramA.pdf',bbox_inches='tight',
    transparent=True, pad_inches=0.0 )
```

Ce code produit l'image présentée en figure 6.3 pour un morceau de musique classique (provenant d'une conversion d'une fichier `mp3` vers le format `wav`).

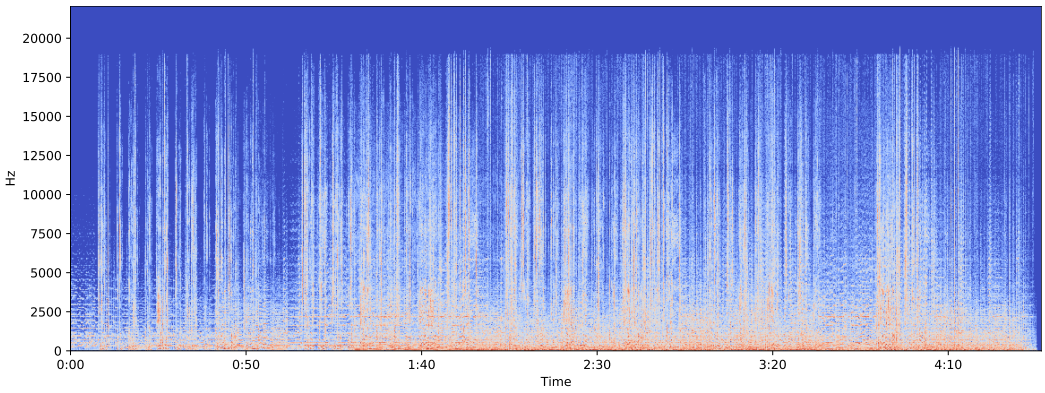


FIGURE 6.3 – Tracé d'un spectrogramme en python grâce à la bibliothèque `librosa` d'un morceau de musique avec une fenêtre à temps court.

Remarque 6.4 :

Nous venons de présenter rapidement la transformée de Fourier à fenêtre sur le plan théorique et *analytique*. Il faudrait, mais cela ferait trop pour ce cours introductif, rentrer dans le détails de la version discrète et des aspects numériques de cette transformée de Fourier à fenêtre.

6.3 LA TRANSFORMATION EN ONDELETTES

On a vu que la transformée de Fourier à fenêtre présente un inconvénient majeur : la fenêtre est de longueur fixe, et cet handicap est important lorsqu'on

veut traiter des signaux dont les variations peuvent avoir des ordres de grandeur très variables. C'est le cas notamment pour le traitement du son : l'attaque d'une note est une phase courte composée de hautes fréquences et caractéristiques de l'instrument et de l'interprète, tandis que le reste de la note contient des fréquences relativement plus basses.

Le géophysicien J. Morlet [3] a constaté ces inconvénients en prospection pétrolière pour l'analyse des signaux sismiques. Ceci l'a amené à proposer en 1983 une méthode nouvelle où la fenêtre varie par translation mais aussi par dilatation ou contraction. En anglais, cette transformée en ondelettes est appelée *wavelet transform*.

6.3.1 L'idée de base

L'idée de la transformée en ondelettes est de remplacer la famille de fonctions de la transformée de Fourier à fenêtre :

$$w_{\nu,b}(t) = \overline{w}(t-b)\mathrm{e}^{2\mathrm{i}\pi\nu t}, \quad b \in \mathbf{R}, \nu \in \mathbf{R},$$

par une famille de fonctions élémentaires, dîtes *ondelettes*, construite à partir d'une *ondelette-mère* ψ :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right), \quad a, b \in \mathbf{R}, a \neq 0. \tag{6.5}$$

On ne rentrera pas dans le détails des choix de l'ondelette mère mais celle-ci, doit satisfaire les hypothèses données dans la définition suivante.

DÉFINITION 6.7 — (Transformée en ondelettes) Soit $\psi \in L^1 \cap L^2$ une ondelette-mère vérifiant

$$\int_{\mathbf{R}} |\psi(t)|^2 \, \mathrm{d}t = 1,$$

et

$$\int_{\mathbf{R}} \frac{\left|\widehat{\psi}(\nu)\right|^2}{|\nu|} \mathrm{d}\nu = K < \infty.$$

La transformée en ondelettes d'une fonction $f \in L^2$ est la fonction C_f :

3. Jean Morlet (né à Fontenay-sous-Bois le 13 janvier 1931, mort à Nice le 27 avril 2007), ancien élève de l'École polytechnique (X1952), est un géophysicien français qui est le pionnier dans le domaine de l'analyse des ondelettes en collaboration avec Alex Grossmann. Morlet invente le mot « ondelette » pour décrire des équations similaires à celles existant depuis environ les années 1930. (Wikipedia)

$(\mathbf{R} - \{0\}) \times \mathbf{R} \to \mathbf{C}$ définie par :

$$C_f(a,b) = \langle f,\ \psi_{a,b} \rangle = \int_{\mathbf{R}} f(t)\overline{\psi}_{a,b}(t)\mathrm{d}t,$$

où la famille $\psi_{a,b}$ est définie par (6.5).

On peut tout de même citer quelques ondelettes mères.

Ondelette de Haar. Une ondelette mère simple qui permet de bien visualiser les effets de translation et de dilatation est l'ondelettede Haar [4], ondelette considérée comme la première ondelette connue.

La fonction-mère des ondelettes de Haar est une fonction constante par morceaux :

$$\psi(t) = \begin{cases} 1 & \text{pour } -\frac{1}{2} \leq t < 0, \\ -1 & \text{pour } 0 \leq t < \frac{1}{2}, \\ 0 & \text{sinon.} \end{cases}$$

En figure 6.4, sont représenter l'ondelette mère de Haar ainsi que deux ondelettes construites à partir d'elle.

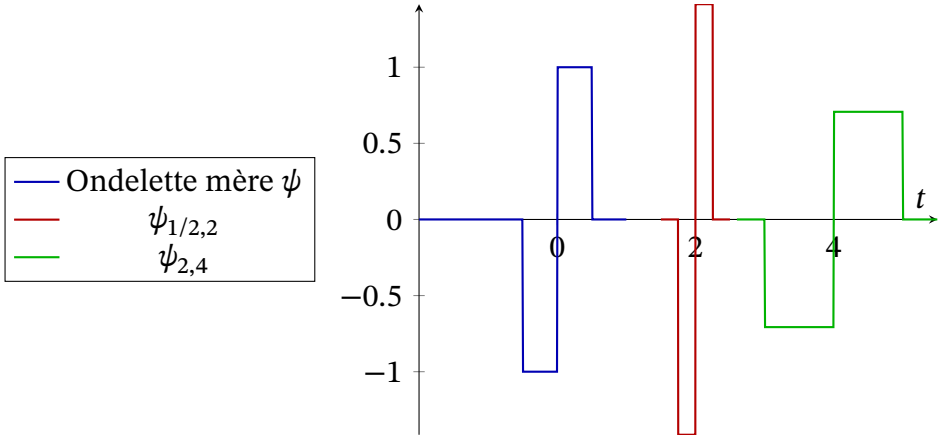


FIGURE 6.4 – Représentation de l'ondelette mère de Haar ainsi que deux ondelettes filles qui permettent d'observer les phénomènes de décalage temporel et de dilatation.

4. Alfréd Haar (né le 11 octobre 1885 à Budapest et mort le 16 mars 1933 à Szeged) est un mathématicien hongrois.

La transformée de Fourier de l'ondelette de Haar est imaginaire pure, et vaut

$$\widehat{\psi}(\nu) = \mathrm{i}\frac{\cos(\pi\nu) - 1}{\pi\nu}.$$

Nous la représentons en figure 6.5.

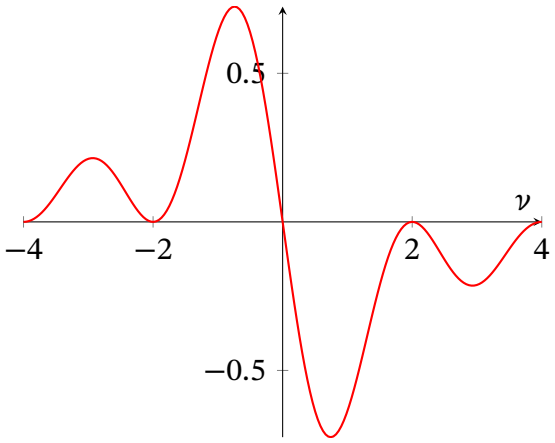


FIGURE 6.5 – Partie imaginaire de la transformée de Fourier de l'ondelette de Haar, sa partie réelle étant nulle.

L'ondelette de Morlet. l' ondelette de Morlet [5] est une ondelette composée d'une exponentielle complexe (porteuse) multipliée par une fenêtre gaussienne (enveloppe). Cette ondelette est étroitement liée à la perception humaine, à la fois auditive et visuelle.

En figure 6.6, on a représenté l'ondelette réelle mère de Morlet suivante :

$$\psi(t) = \mathrm{e}^{-t^2/2}\cos(5t),$$

ainsi que sa transformée de Fourier.

On a un équivalent du théorème 6.6 obtenu pour la transformée de Fourier à fenêtre pour la transformée en ondelettes. Le théorème suivant donne une formule d'inversion, c'est-à-dire une formule qui permet d'obtenir une fonction L^2 à partir de sa transformée en ondelettes. De plus, la transformée en ondelettes préserve l'énergie.

5. Jean Morlet (13 janvier 1931 - 27 avril 2007) était un géophysicien français qui a été un pionnier dans le domaine de l'analyse des ondelettes. Il a inventé le terme ondelette pour décrire les fonctions qu'il utilisait.

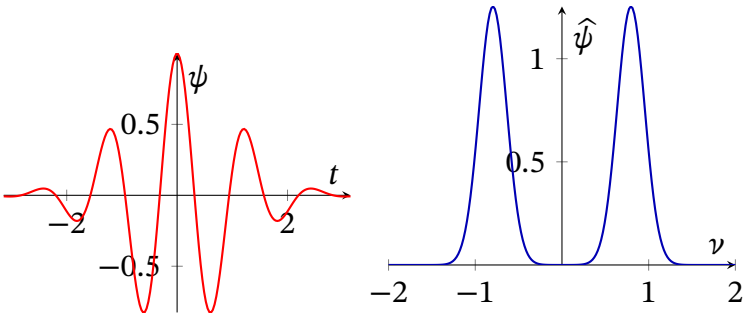


FIGURE 6.6 – Ondelette mère de Morlet et sa transformée de Fourier.

THÉORÈME 6.8 Soit ψ une ondelette-mère qui satisfait les hypothèses de la définition 6.7 et la famille d'ondelettes $\psi_{a,b}$ définie par (6.5). Pour toute fonction $f \in L^2$, on a :

$$\int_{\mathbf{R}} |f(t)|^2 \, \mathrm{d}t = \frac{1}{K} \int_{\mathbf{R}-\{0\}} \int_{\mathbf{R}} |C_f(a,b)|^2 \, \frac{\mathrm{d}a \, \mathrm{d}b}{a^2},$$

et

$$f(t) = \frac{1}{K} \int_{\mathbf{R}-\{0\}} \int_{\mathbf{R}} C_f(a,b) \psi_{a,b}(t) \frac{\mathrm{d}a \, \mathrm{d}b}{a^2}.$$

Preuve : (difficile) À ÉCRIRE. ■

6.3.2 Localisation temps-fréquence

Nous considérerons dans cette introduction à la théorie des ondelettes que des ondelettes mères ψ est centré en 0. Ainsi, il est facile de vérifier par le calcul que $\psi_{a,b}$ est centrée en b.

Dans un soucis de généralité, nous ne présupposerons rien concernant le centre de la transformée de Fourier de ψ que nous noterons classiquement $m_{\widehat{\psi}}$.

On peut montrer que :

$$\widehat{\psi}_{a,b}(\nu) = |a|^{1/2} \, \mathrm{e}^{-2\mathrm{i}\pi\nu b} \widehat{\psi}(a\nu), \tag{6.6}$$

et grâce à cette relation, que :

$$m_{\widehat{\psi}_{a,b}} = \frac{1}{a} m_{\widehat{\psi}}. \tag{6.7}$$

Exercice 6.2 :

Montrer les égalités (6.6) et (6.7).

Concernant l'étalement en temps et en fréquence, en adoptant les même conventions et définitions que pour la transformée de Fourier à fenêtre, on peut montrer qu'à partir des définitions données à la section précédentes :

$$\sigma_{\psi_{a,b}} = a\sigma_{\psi}. \tag{6.8}$$

De même on peut prouver que

$$\sigma_{\widehat{\psi}_{a,b}} = \frac{1}{a}\sigma_{\widehat{\psi}}. \tag{6.9}$$

Exercice 6.3 :

Montrer les égalités (6.8) et (6.9).

On voit alors que $C_f(a,b)$ est le résultat de l'analyse de la fonction f dans la boîte d'Heisenberg :

$$[b - \frac{a}{2}\sigma_{\psi},\ b + \frac{a}{2}\sigma_{\psi}] \times \left[\frac{m_{\widehat{\psi}}}{a} - \frac{\sigma_{\widehat{\psi}}}{2a},\ \frac{m_{\widehat{\psi}}}{a} + \frac{\sigma_{\widehat{\psi}}}{2a}\right].$$

On peut alors, pour se fixer les idées, représenter ce phénomène comme un pavage du plan temps-fréquence comme fait à la figure 6.7. On constate que plus la fréquence est haute, c'est-à-dire a grand, plus l'étalement fréquentiel est grand, et moins l'étalement temporel est grand, et réciproquement.

6.4 Aspects numériques et compression

À ÉCRIRE

6.5 Notes bibliographiques

Pour l'analyse temps-fréquence on renverra aux références classiques [3, 5, 9].

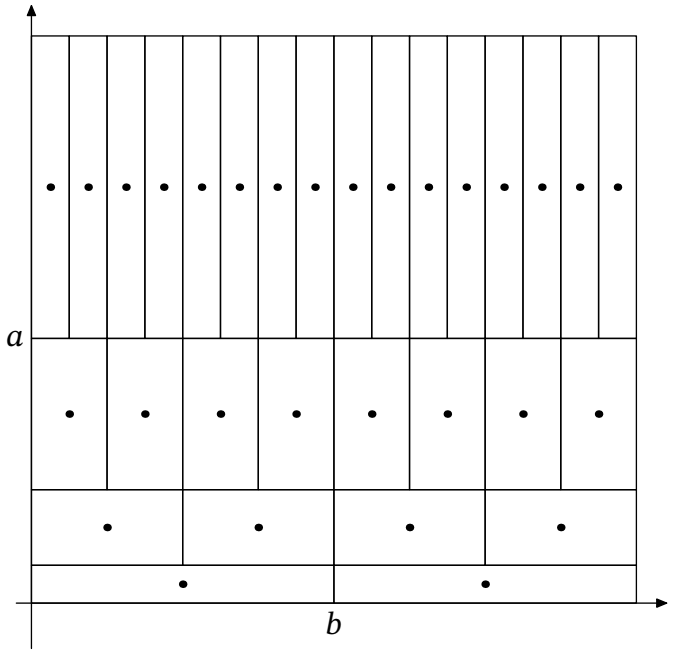


FIGURE 6.7 – Pavage du plan temps-fréquence pour l'analyse en ondelettes.

ANNEXE A

Rappels d'intégration

A.1 Théorèmes de convergence

Dans les deux théorème qui suivent, on considèrera (Ω, B, μ) un espace mesuré (on pourra penser à un ouvert de $\mathbf{R}^n$).

Théorème A.1 — (de convergence monotone)

1. Soit $(f_n)_{n\geqslant 0}$ une suite *croissante* de fonctions *mesurables* sur Ω. On a :
$$\lim_{n\to+\infty} \int f_n \, \mathrm{d}\mu = \int \lim_{n\to+\infty} f_n \, \mathrm{d}\mu.$$
2. Soit $(f_n)_{n\geqslant 0}$ une suite de fonctions *mesurables positives*. On a :
$$\int \sum_{n=0}^{\infty} f_n \, \mathrm{d}\mu = \sum_{n=0}^{\infty} \int f_n \, \mathrm{d}\mu.$$

THÉORÈME A.2 — (de convergence dominée (LEBESGUE)) Soit $(f_n)_{n\geqslant 0}$ une suite de fonction *mesurables* de Ω dans $\mathbf{C}$ et f une fonction *mesurable* de Ω dans $\mathbf{C}$. On suppose que :

- **(limite)** pour μ-presque tout $x \in \Omega$, $f_n(x) \underset{n\to\infty}{\to} f(x)$;
- **(domination)** il existe une fonction $\varphi : \Omega \to \mathbf{R}[+]$ mesurable telle que $\int \varphi \mathrm{d}\mu < \infty$ et pour tout $n \in \mathbf{N}$, pour μ-presque tout $x \in \Omega$, $|f_n(x)| \leqslant \varphi(x)$.

On a alors que f est intégrable, de même que pour tout $n \in \mathbf{N}$ f_n est intégrable, et

$$\int |f_n - f| \,\mathrm{d}\mu \underset{n\to\infty}{\to} 0 \quad \text{et} \quad \int f_n \,\mathrm{d}\mu \underset{n\to\infty}{\to} \int f \,\mathrm{d}\mu.$$

THÉORÈME A.3 — (de continuité sous l'intégrale) Soit $f : (t, x) \mapsto f(t, x)$ une fonction de $I \times \Omega$ dans $\mathbf{C}$ où I est un intervalle de $\mathbf{R}$. On suppose que :

- **(mesurabilité)** pour tout $t \in I$, $x \mapsto f(t, x)$ est mesurable ;
- **(continuité)** pour presque tout $x \in \Omega$, $t \mapsto f(t, x)$ est continue sur I ;
- **(domination)** il existe une fonction $\varphi : \Omega \to \mathbf{R}_+$ mesurable telle que $\int \varphi \mathrm{d}\mu < \infty$ et que pour tout $t \in I$, pour presque tout $x \in \Omega$, $|f(t, x)| \leqslant \varphi(x)$.

Alors, la fonction

$$F : t \mapsto F(t) = \int f(t, x) \mathrm{d}\mu(x)$$

est bien définie pour tout $t \in I$ et est continue sur I.

THÉORÈME A.4 — (de dérivation sous l'intégrale) Soit $f : (t,x) \mapsto f(t,x)$ une fonction de $I \times \Omega$ dans $\mathbf{C}$ où I est un intervalle de $\mathbf{R}$. On suppose que :

- **(existence de F)** pour tout $t \in I$, $x \mapsto f(t,x)$ est intégrale ;
- **(dérivabilité)** pour presque tout $x \in \Omega$, $t \mapsto f(t,x)$ est dérivable sur I, de dérivée notée $\frac{\partial f}{\partial t}$;
- **(domination de la dérivée)** il existe une fonction $\varphi : \Omega \to \mathbf{R}_+$ mesurable telle que $\int \varphi \mathrm{d}\mu < \infty$ et que pour tout $t \in I$, pour presque tout $x \in \Omega$, $\left|\frac{\partial f}{\partial t}(t,x)\right| \leqslant \varphi(x)$.

Alors, la fonction

$$F : t \mapsto F(t) = \int f(t,x)\mathrm{d}\mu(x)$$

est dérivable sur I et pour tout $t \in I$

$$F'(t) = \int \frac{\partial f}{\partial t}(t,x)\mathrm{d}\mu(x).$$

A.2 THÉORÈMES DE FUBINI ET TONELLI

Les théorèmes de Fubini et de Tonelli sont des théorèmes qui permettent de changer les ordres d'intégration dans les calculs d'intégrales de fonctions dépendant de plusieurs variables. Il en existe différentes versions.

Soient (Ω, B, μ) et (Ω', B', ν) des espaces mesurés σ-fini (on pourra penser à des ouverts de $\mathbf{R}^n$ par exemple).

THÉORÈME A.5 — (de TONELLI) Soit $f : \Omega \times \Omega' \to [0, +\infty[$ mesurable. Alors :

- pour tout $x \in \Omega, y \mapsto f(x,y)$ est B'-mesurable et $x \mapsto \int_{\Omega'} f(x,y)\mathrm{d}\nu(y)$ est B-mesurable.
- Pour tout $y \in \Omega', x \mapsto f(x,y)$ est B-mesurable et $y \mapsto \int_{\Omega} f(x,y)\mathrm{d}\mu(x)$ est B'-mesurable.

En outre, on a :

$$\begin{aligned}\int_{\Omega\times\Omega'} f(x,y)\mathrm{d}(\mu\otimes\nu)(x,y) &= \int_{\Omega}\left(\int_{\Omega'} f(x,y)\mathrm{d}\nu(y)\right)\mathrm{d}\mu(x)\\ &= \int_{\Omega'}\left(\int_{\Omega} f(x,y)\mathrm{d}\mu(x)\right)\mathrm{d}\nu(y)\end{aligned}$$

où $\mu\otimes\nu$ est la mesure produit.

THÉORÈME A.6 — (de FUBINI) Soit $f \in L^1(\Omega\times\Omega')$ mesurable. Alors :

- la fonction $x \mapsto \int_{\Omega'} f(x,y)\mathrm{d}\nu(y)$ estdéfinie pour presque tout x et est dans $L^1(\Omega)$.
- La fonction $y \mapsto \int_{\Omega} f(x,y)\mathrm{d}\mu(x)$ est définie pour presque tout y et est dans $L^1(\Omega')$.

En outre, on a :

$$\begin{aligned}\int_{\Omega\times\Omega'} f(x,y)\mathrm{d}(\mu\otimes\nu)(x,y) &= \int_{\Omega}\left(\int_{\Omega'} f(x,y)\mathrm{d}\nu(y)\right)\mathrm{d}\mu(x)\\ &= \int_{\Omega'}\left(\int_{\Omega} f(x,y)\mathrm{d}\mu(x)\right)\mathrm{d}\nu(y)\end{aligned}$$

où $\mu\otimes\nu$ est la mesure produit.

Bibliographie

[1] F. BAVAUD, J.-C. CHAPPELIER et J. KOHLAS. *Introduction à la Théorie de l'Information et ses applications*. 2008. URL : https://icwww.epfl.ch/~chappeli/it/pdf/FullCourseEPFL-FR.pdf.

[2] Maurice BELLANGER. *Traitement numérique du signal Cours et exercices corrigés 9eme edition*. 9eme. Dunod, 2012. ISBN : 2100588648 9782100588640.

[3] Pierre BRÉMAUD. *Mathematical Principles of Signal Processing : Fourier and Wavelet Analysis*. 1re éd. Springer-Verlag New York, 2002. ISBN : 978-1-4419-2956-3,978-1-4757-3669-4.

[4] Thomas M. COVER et Joy A. THOMAS. *Elements of Information Theory*. 2e éd. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, 2006. ISBN : 0-471-24195-4,978-0-471-24195-9.

[5] Claude GASQUET et Patrick WITOMSKI. *Analyse de Fourier et applications : Filtrage, calcul numérique, ondelettes*. Dunod, 1995. ISBN : 2-225-85426-2.

[6] V.K. INGLE et J.G. PROAKIS. *Digital Signal Processing Using Matlab : Version 4*. 1999.

[7] Donald E. KNUTH. *The Art of Computer Programming, Volume 1 (3rd Ed.) : Fundamental Algorithms*. USA : Addison Wesley Longman Publishing Co., Inc., 1997. ISBN : 0201896834.

[8] F. J. MACWILLIAMS et N. J. A. SLOANE. *The Theory of Error-Correcting Codes*. 1st. North-Holland Mathematical Library 16. North-Holland, 1977. ISBN : 9780444850102,0444850104.

[9] Stephane MALLAT. *A wavelet tour of signal processing the Sparse way*. 3e éd. Academic Press, 2008. ISBN : 9780123743701,0123743702.

[10] Khalid SAYOOD. *Introduction to data compression*. 3rd ed. Morgan Kaufmann series in multimedia information and systems. Elsevier, 2006. ISBN : 012620862X,9780126208627,9780080509259.

[11] Lin SHU et Daniel J. COSTELLO. *Error control coding : fundamentals and applications*. Cambridge University Press, 2004. ISBN : 0130179736.

[12] E. TISSERAND, J.F. PAUTEX et P. SCHWEITZER. *Analyse et traitement des signaux - 2e éd. : Méthodes et applications au son et à l'image*. Sciences de l'ingénieur. Dunod, 2009. ISBN : 9782100539840.

[13] X. Wu. “On convergence of Lloyd’s method I”. In : *IEEE Transactions on Information Theory* 38.1 (jan. 1992), p. 171-174. ISSN : 0018-9448. DOI : 10.1109/18.108266.

Index