

Transformée de Fourier Discrète

Arthur Leclaire

Dans tout ce cours, pour $a, b \in \mathbb{Z}$, $a \leq b$, on notera $[a : b] = \{a, a + 1, \dots, b - 1, b\}$.
 Pour $N \in \mathbb{N}^*$, on notera $\mathbb{Z}_N = \mathbb{Z}/N\mathbb{Z}$ l'ensemble des entiers modulo N .
 Une suite $(u(x))_{x \in \mathbb{Z}}$ indexée par \mathbb{Z} est identifiée à une fonction $u : \mathbb{Z} \rightarrow \mathbb{C}$.

1 Transformée de Fourier Discrète 1D

1.1 L'espace $\mathbb{C}^{\mathbb{Z}_N}$

Définition 1. On note $\mathbb{C}^{\mathbb{Z}_N}$ l'ensemble des suites $u : \mathbb{Z} \rightarrow \mathbb{C}$ qui sont N -périodiques, c'est-à-dire qui vérifient

$$\forall x \in \mathbb{Z}, \quad u(x + N) = u(x).$$

$\mathbb{C}^{\mathbb{Z}_N}$ est un \mathbb{C} -espace vectoriel. De plus, il est muni d'un produit scalaire et d'une norme définis par

$$\langle u, v \rangle = \sum_{x=0}^{N-1} u(x) \overline{v(x)} \quad , \quad \|u\| = \sqrt{\langle u, u \rangle} = \left(\sum_{x=0}^{N-1} |u(x)|^2 \right)^{\frac{1}{2}}.$$

Le produit scalaire vérifie les propriétés suivantes : pour tous $\alpha, \beta \in \mathbb{C}$, $u, u', v, v' \in \mathbb{C}^{\mathbb{Z}_N}$,

$$\begin{aligned} \langle \alpha u + \beta u', v \rangle &= \alpha \langle u, v \rangle + \beta \langle u', v \rangle \\ \langle u, \alpha v + \beta v' \rangle &= \bar{\alpha} \langle u, v \rangle + \bar{\beta} \langle u, v' \rangle \\ \langle v, u \rangle &= \overline{\langle u, v \rangle} \\ \langle u, u \rangle &\geq 0 \\ \langle u, u \rangle = 0 &\iff u = 0. \end{aligned}$$

Remarque 1. Remarquons qu'une suite $u : \mathbb{Z} \rightarrow \mathbb{C}$ qui est N -périodique s'identifie à une fonction $U : \mathbb{Z}_N \rightarrow \mathbb{C}$. En effet, si u est N -périodique, alors on peut définir

$$\forall x \in \mathbb{Z}, \quad U(x \bmod N) = u(x)$$

car le membre de droite ne dépend que de la classe de x modulo N . Ceci explique la notation $\mathbb{C}^{\mathbb{Z}_N}$ qui sera utilisée tout au long du cours.

Proposition 1. L'espace $\mathbb{C}^{\mathbb{Z}_N}$ s'identifie à \mathbb{C}^N via l'isomorphisme

$$\begin{aligned} \varphi : \mathbb{C}^{\mathbb{Z}_N} &\longrightarrow \mathbb{C}^N \\ u &\longmapsto (u(x))_{x \in [0 : N-1]} \end{aligned}$$

Démonstration. L'application φ correspond à la restriction d'une suite N -périodique à l'intervalle $[0 : N - 1]$. Donc on voit aisément que φ est linéaire. De plus, on peut définir sa réciproque ψ explicitement : si $v \in \mathbb{C}^N$, on définit $u = \psi(v)$ comme le prolongement par N -périodicité de la suite finie v . Plus précisément, pour tout $x \in \mathbb{Z}$, il existe un unique couple $(q, r) \in \mathbb{Z} \times [0 : N - 1]$ tel que $x = Nq + r$ (division euclidienne), et on pose alors $u(x) = v(r)$. On vérifie immédiatement que $\varphi(\psi(v)) = v$ et que $\psi(\varphi(u)) = u$. \square

Exemple 1. Dans la pratique, on peut utiliser cet isomorphisme pour écrire explicitement des vecteurs de $\mathbb{C}^{\mathbb{Z}_N}$. Par exemple, on écrira

$$u = (2, i, 0, 1, \pi) \in \mathbb{C}^{\mathbb{Z}_5}$$

pour dire que u est la suite 5-périodique définie par

$$u : \begin{array}{l} 0 \mapsto 2 \\ 1 \mapsto i \\ 2 \mapsto 0 \\ 3 \mapsto 1 \\ 4 \mapsto \pi \end{array} .$$

Mais u étant 5 périodique, on en déduit immédiatement que $u(5) = 2, u(6) = i, u(9) = u(4) = \pi$ ou encore que $u(2020) = u(5 \times 404) = u(0) = 2$.

Remarque 2. On prendra garde au fait que l’isomorphisme précédent adopte une convention : on restreint aux indices $[0 : N - 1]$. On aurait pu utiliser tout autre intervalle de N entiers consécutifs, par exemple $[1 : N]$. Mais dans l’étude de la transformée de Fourier discrète, il est plus naturel de commencer à l’indice $x = 0$ et pas à l’indice $x = 1$. Dans la pratique, on utilisera aussi souvent un intervalle (quasiment) centré en zéro : par exemple $[-\frac{N-1}{2} : \frac{N-1}{2}]$ quand N est impair.

1.2 Transformée de Fourier Discrète

Définition 2. Si $u \in \mathbb{C}^{\mathbb{Z}_N}$, on définit la transformée de Fourier discrète $\hat{u} \in \mathbb{C}^{\mathbb{Z}_N}$ de u en posant

$$\forall \xi \in \mathbb{Z}, \quad \hat{u}(\xi) = \sum_{x=0}^{N-1} u(x) e^{-2i\pi \frac{\xi x}{N}} .$$

Remarquons que \hat{u} ainsi définie est bien N -périodique car

$$\begin{aligned} \forall \xi \in \mathbb{Z}, \quad \hat{u}(\xi + N) &= \sum_{x=0}^{N-1} u(x) e^{-2i\pi \frac{(\xi+N)x}{N}} = \sum_{x=0}^{N-1} u(x) e^{-2i\pi \frac{\xi x}{N} - 2i\pi x} \\ &= \sum_{x=0}^{N-1} u(x) e^{-2i\pi \frac{\xi x}{N}} e^{-2i\pi x} = \sum_{x=0}^{N-1} u(x) e^{-2i\pi \frac{\xi x}{N}} = \hat{u}(\xi) . \end{aligned}$$

On s’autorisera à abrégier en écrivant TFD (ou DFT en anglais), et on écrira parfois $\hat{u} = \text{DFT}(u)$.

Définition 3. Si $v \in \mathbb{C}^{\mathbb{Z}_N}$, on définit $\check{v} \in \mathbb{C}^{\mathbb{Z}_N}$ de v en posant

$$\forall x \in \mathbb{Z}, \quad \check{v}(x) = \frac{1}{N} \sum_{\xi=0}^{N-1} v(\xi) e^{2i\pi \frac{\xi x}{N}} .$$

On peut voir de la même manière que \check{v} est aussi N -périodique. Notons que $\check{v}(x) = \frac{1}{N} \hat{v}(-x)$.

Théorème 1. La TFD est un isomorphisme de $\mathbb{C}^{\mathbb{Z}_N}$ sur lui-même, et sa réciproque est $v \mapsto \check{v}$. Autrement dit, pour tout $u \in \mathbb{C}^{\mathbb{Z}_N}$, on a $u = \check{\check{u}}$, c’est-à-dire

$$\forall x \in \mathbb{Z}, \quad u(x) = \frac{1}{N} \sum_{\xi=0}^{N-1} \hat{u}(\xi) e^{2i\pi \frac{\xi x}{N}} .$$

L’application $\text{DFT}^{-1} : v \mapsto \check{v}$ sera donc appelée TFD inverse.

Démonstration. D'abord, la TFD est linéaire par linéarité de la somme. Pour le reste, il suffit de montrer la formule proposée. Soit $u \in \mathbb{C}^{\mathbb{Z}_N}$. Fixons d'abord $x \in [0 : N - 1]$. On a

$$\begin{aligned}\check{u}(x) &= \frac{1}{N} \sum_{\xi=0}^{N-1} \hat{u}(\xi) e^{2i\pi \frac{\xi x}{N}} = \frac{1}{N} \sum_{\xi=0}^{N-1} \sum_{y=0}^{N-1} u(y) e^{-2i\pi \frac{\xi y}{N}} e^{2i\pi \frac{\xi x}{N}} \\ &= \frac{1}{N} \sum_{\xi=0}^{N-1} \sum_{y=0}^{N-1} u(y) e^{2i\pi \frac{\xi(x-y)}{N}} = \frac{1}{N} \sum_{y=0}^{N-1} u(y) \sum_{\xi=0}^{N-1} \left(e^{\frac{2i\pi(x-y)}{N}} \right)^\xi.\end{aligned}$$

Or, pour x, y fixés dans $[0 : N - 1]$, $\omega = e^{\frac{2i\pi(x-y)}{N}}$ est une racine N -ième de l'unité car

$$\omega^N = \left(e^{\frac{2i\pi(x-y)}{N}} \right)^N = e^{2i\pi(x-y)} = 1;$$

de plus $\omega = 1$ si et seulement si $x - y = 0 \pmod{N}$, et dans le cas contraire, on a $\sum_{\xi=0}^{N-1} \omega^\xi = \frac{1 - \omega^N}{1 - \omega} = 0$. Il en résulte que pour tout $x \in [0 : N - 1]$,

$$\check{u}(x) = \frac{1}{N} \sum_{y=0}^{N-1} u(y) N \mathbf{1}_{x=y} = u(x).$$

Les deux membres étant des fonctions N -périodiques de x , cette égalité se prolonge à \mathbb{Z} par périodicité. \square

Proposition 2. Soient $u, v \in \mathbb{C}^{\mathbb{Z}_N}$. Alors $\langle \hat{u}, v \rangle = N \langle u, \check{v} \rangle$.

Démonstration. \rightarrow TD \square

Théorème 2. La TFD préserve la norme et le produit scalaire à constante près :

$$\forall u, v \in \mathbb{C}^{\mathbb{Z}_N}, \quad \|u\|^2 = \frac{1}{N} \|\hat{u}\|^2, \quad \langle u, v \rangle = \frac{1}{N} \langle \hat{u}, \hat{v} \rangle.$$

Démonstration. \rightarrow TD. \square

Remarque 3. Cette conservation de norme est extrêmement importante. C'est l'analogue pour la TFD de l'égalité de Bessel-Parseval valable dans les espaces de Hilbert. D'un point de vue physique, la norme ℓ^2 de u représente l'énergie du signal u . Le théorème précédent assure qu'il y a conservation de l'énergie (à constante près) lorsque l'on prend la TFD du signal.

1.3 La base de Fourier discrète

Théorème 3. Pour $\xi \in \mathbb{Z}$, notons

$$e_\xi(x) = e^{2i\pi \frac{\xi x}{N}}.$$

Alors $(e_\xi)_{\xi \in [0:N-1]}$ est une base orthogonale de $\mathbb{C}^{\mathbb{Z}_N}$, c'est-à-dire que

$$\forall \xi, \zeta \in [0 : N - 1], \quad \langle e_\xi, e_\zeta \rangle = N \mathbf{1}_{\xi=\zeta}.$$

On l'appelle base de Fourier discrète modulo N (ou juste base de Fourier) et on la notera \mathcal{E} .

Démonstration. Le théorème 1 montre que tout $u \in \mathbb{C}^{\mathbb{Z}_N}$ se décompose de manière unique sur \mathcal{E} . De plus

$$\langle e_\xi, e_\zeta \rangle = \sum_{x=0}^{N-1} e^{2i\pi \frac{(\xi-\zeta)x}{N}}$$

et comme dans la preuve du Théorème 1, ceci vaut N si $\xi = \zeta \pmod{N}$ et 0 sinon. \square

Remarque 4.

- Le théorème 1 donne la décomposition de $u \in \mathbb{C}^{\mathbb{Z}_N}$ sur la base orthogonale \mathcal{E} , qu'on peut réécrire

$$u = \sum_{\xi=0}^{N-1} \frac{\langle u, e_\xi \rangle}{\|e_\xi\|^2} e_\xi.$$

- En normalisant par $\|e_\xi\| = \sqrt{N}$, on a que $(\frac{e_\xi}{\sqrt{N}})_{\xi \in [0:N-1]}$ est une base orthonormée de $\mathbb{C}^{\mathbb{Z}_N}$.

Exemple 2. Dans $\mathbb{C}^{\mathbb{Z}_N}$ pour $N > 7$, considérons $u(x) = \sin(\frac{6\pi x}{N})$. On a

$$u(x) = \frac{1}{2i} \left(e^{2i\pi \frac{3x}{N}} - e^{-2i\pi \frac{3x}{N}} \right) = \frac{1}{2i} \left(e^{2i\pi \frac{3x}{N}} - e^{2i\pi \frac{(N-3)x}{N}} \right) = \frac{1}{N} \left(\frac{N}{2i} e_3(x) - \frac{N}{2i} e_{N-3}(x) \right).$$

Comme $N > 7$, e_3 et e_{N-3} sont bien deux éléments (distincts) de la base de Fourier. Par unicité de la décomposition sur \mathcal{E} , on peut identifier les coefficients, ce qui donne directement la TFD de u , à savoir $\hat{u}(\xi) = 0$ si $\xi \notin \{3, N-3\}$ et

$$\hat{u}(3) = \frac{N}{2i} \quad \text{et} \quad \hat{u}(N-3) = -\frac{N}{2i}.$$

1.4 Spectre de Fourier

Proposition 3. Si $u \in \mathbb{R}^{\mathbb{Z}_N}$, alors pour tout $\xi \in \mathbb{Z}$, $\hat{u}(-\xi) = \overline{\hat{u}(\xi)}$ et en particulier $|\hat{u}|$ est pair.

Démonstration. → TD. □

Définition 4. Soit $u \in \mathbb{C}^{\mathbb{Z}_N}$. On appelle spectre (d'amplitude) de u la fonction $S_u \in \mathbb{R}_+^{\mathbb{Z}_N}$ définie par

$$S_u(\xi) = |\hat{u}(\xi)|.$$

On appelle (spectre de) phase de u la fonction φ_u à valeurs dans $\mathbb{R}/2\pi\mathbb{Z}$ définie aux points ξ tels que $\hat{u}(\xi) \neq 0$ par

$$\varphi_u(\xi) = \arg(\hat{u}(\xi)) \pmod{2\pi}.$$

En pratique, l'outil informatique adopte souvent une convention arbitraire pour l'argument de 0.

Remarque 5.

- Le spectre de u indique la répartition des composantes fréquentielles dans le signal u . Le spectre de phase quant à lui indique le positionnement relatif de chaque composante fréquentielle (mais il est plus difficile à interpréter).
- Pour un signal u à valeurs réelles, la proposition ci-dessus indique que le spectre est pair. En fait, pour des signaux réguliers, on observera souvent que le spectre présente un grand pic près de la fréquence nulle et décroît assez vite quand $|\xi|$ grandit (modulo la périodicité bien sûr).
- Pour cette raison, en pratique on représentera souvent le spectre en échelle logarithmique, c'est-à-dire qu'on représentera $\log(1 + |\hat{u}(\xi)|)$, de façon à ne pas trop écraser les hautes fréquences.
- Le spectre d'un signal réel étant pair, il est agréable de le visualiser sur un intervalle centré en zéro. Lorsque N est impair, $N = 2K + 1$, on prendra les fréquences $\xi \in [-K : K]$. Lorsque N est pair, $N = 2K$, on peut prendre les fréquences $\xi \in [-K : K-1]$ ou bien $\xi \in [-K+1 : K]$. Dès lors, les fréquences ξ près de 0 seront appelées *basses fréquences*, et les fréquences ξ près de $\frac{N}{2}$ (i.e. de K) seront appelées *hautes fréquences*. Il faut bien garder en tête que l'on travaille modulo N . Dans les deux cas, la fenêtre $[-K : K]$ avec $K = \lfloor \frac{N}{2} \rfloor$ est suffisante pour décrire tout le spectre (quitte à recopier un coefficient). On pourra aussi utiliser seulement la partie $[0 : K]$ des fréquences positives, comme sur la Fig. 1.

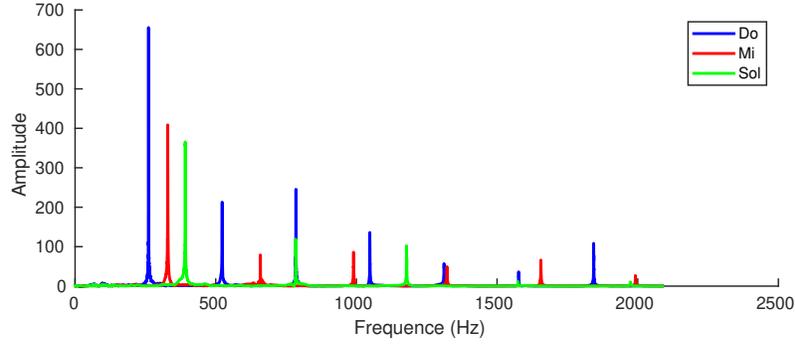


FIGURE 1 – **Spectre d'un accord de do majeur (voir TP)**. On montre ici les spectres de Fourier de signaux sonores obtenus par enregistrement des trois notes Do, Mi, Sol (jouées au piano). Ces spectres étant symétriques, on ne représente que la partie associée aux fréquences positives. Pour chaque note, on constate la présence de plusieurs pics situés sur les multiples d'une fréquence dite fondamentale (qui est en fait la fréquence propre de la corde frappée pour générer le son). On constate aussi que les trois notes partagent certains de leurs pics, signe qu'elles ont des harmoniques communes.

Exemple 3. La visualisation du spectre permet d'analyser des signaux sonores courts. Un son est un signal de pression $p(t)$ dépendant du temps. Un microphone permet de convertir le signal de pression en un signal électrique (tension) $s(t)$. Ensuite, ce signal s est échantillonné suivant une grille temporelle régulière : pour $x \in [0 : N - 1]$, on prendra

$$u(x) = s(x\tau)$$

où τ est la période d'échantillonnage. Pour la plupart des systèmes d'acquisition, la fréquence d'échantillonnage est fixée à $f_s = 44.1\text{kHz}$, et alors $\tau = \frac{1}{f_s} \approx 2.27 \cdot 10^{-5}$ seconde. Un signal sonore de 3" contiendra donc plus de 130000 échantillons.

L'intérêt de l'analyse de Fourier pour les signaux sonores vient du fait qu'ils se décomposent naturellement en oscillations sinusoidales amorties, car les variations de pression observées sont causées par des (micro)-vibrations d'objets ayant une fréquence propre. Plus précisément, on pourra lire dans les pics du spectre les différentes notes (ou harmoniques) qui forment le son, voir Fig. 1. Ceci constitue la base du traitement du signal sonore.

1.5 Transformée de Fourier rapide

A priori, la méthode de calcul naïve de la DFT revient à une multiplication matricielle, et a donc une complexité $\mathcal{O}(N^2)$. En pratique, les logiciels de calcul scientifique utilisent un algorithme beaucoup plus rapide (qui a joué un grand rôle dans le succès de l'utilisation de l'analyse de Fourier en traitement d'images). Cet algorithme consiste à séparer u en parties plus petites, et à se ramener à la DFT de ces signaux plus petits.

Considérons par exemple le cas où $N = 2K$ pour $K \in \mathbb{N}^*$. Notons pour tout $y \in [0 : K - 1]$,

$$\begin{cases} v(y) = u(2y) \\ w(y) = u(2y + 1) \end{cases}$$

Ainsi, pour tout $\xi \in \mathbb{Z}$,

$$\begin{aligned} \hat{u}(\xi) &= \sum_{x=0}^{N-1} u(x) e^{-2i\pi \frac{\xi x}{N}} = \sum_{y=0}^{K-1} u(2y) e^{-2i\pi \frac{\xi 2y}{N}} + \sum_{y=0}^{K-1} u(2y + 1) e^{-2i\pi \frac{\xi(2y+1)}{N}} \\ &= \sum_{y=0}^{K-1} u(2y) e^{-2i\pi \frac{\xi y}{K}} + e^{-2i\pi \frac{\xi}{N}} \sum_{y=0}^{K-1} u(2y + 1) e^{-2i\pi \frac{\xi y}{K}} = \hat{v}(\xi) + e^{-2i\pi \frac{\xi}{N}} \hat{w}(\xi). \end{aligned}$$

Ainsi, pour obtenir les N coefficients de \hat{u} , il suffit de faire les DFT de v et w séparément, puis de les combiner avec N additions et N multiplications (complexes). En notant C_N le nombre d'opérations (maximales) nécessaires au calcul d'une DFT de taille N , on a donc

$$C_{2K} = C_N \leq 2C_K + 2N = 2(C_K + N)$$

Si N est une puissance de 2, $N = 2^p$ avec $p \in \mathbb{N}$, on peut itérer ce procédé et on obtient

$$C_N \leq 2C_{2^{p-1}} + 2^{p+1} \leq 2(2C_{2^{p-2}} + 2^p) + 2^{p+1} = 4C_{2^{p-2}} + 2 \times 2^{p+1} \leq 8C_{2^{p-3}} + 3 \times 2^{p+1} \leq \dots \leq 2^p C_1 + p2^{p+1}.$$

Comme $C_1 = 1$, on obtient $C_N \leq (1+2p)2^p = (1+2 \log_2 N)N$. Dans le cas $N = 2^p$, on a donc un algorithme avec complexité $O(N \log N)$.

Théorème 4. *Pour tout $N \in \mathbb{N}^*$, il existe un algorithme de calcul de la DFT de vecteurs de taille N avec complexité $O(N \log N)$.*

Démonstration. Nous avons traité ci-dessus le cas où N est une puissance de 2. Lorsque N admet un diviseur non trivial M , on peut adapter la technique précédente en séparant le vecteur u en $\frac{N}{M}$ morceaux dans $\mathbb{C}^{\mathbb{Z}_M}$. Par contre, lorsqu'on tombe sur un nombre premier, il faut faire appel à d'autres techniques calcul dans les groupes finis. Pour les curieux, aller voir la *Proceeding letter* de Charles M. Rader intitulée "Discrete Fourier Transforms when the Number of Data Samples is Prime". \square

2 Convolution périodique

2.1 Définition

Définition 5. Soient $u, v \in \mathbb{C}^{\mathbb{Z}_N}$. On définit la convolution périodique de u et v , notée $u * v$ par

$$\forall x \in \mathbb{Z}, \quad u * v(x) = \sum_{y=0}^{N-1} u(y)v(x-y).$$

Proposition 4. *La convolution est commutative et associative.*

Démonstration. Pour la commutativité, on fait le changement d'indice $z = x - y$, ce qui donne

$$u * v(x) = \sum_{z=x-N+1}^x u(x-z)v(z)$$

et comme la suite $z \mapsto u(x-z)v(z)$ est N -périodique, on peut sommer sur les $z \in [0 : N-1]$. Pour l'associativité, il suffit de faire les calculs et d'échanger les sommes... \square

2.2 Lien avec la TFD

Théorème 5. Soient $u, v \in \mathbb{C}^{\mathbb{Z}_N}$. On a $\widehat{u * v} = \hat{u} \hat{v}$.

Autrement dit, la TFD transforme le produit de convolution en produit.

Démonstration. Pour tout $\xi \in \mathbb{Z}$, on a

$$\begin{aligned} \widehat{u * v}(\xi) &= \sum_{x=0}^{N-1} u * v(x) e^{-2i\pi \frac{\xi x}{N}} = \sum_{x,y=0}^{N-1} u(y)v(x-y) e^{-2i\pi \frac{\xi x}{N}} \\ &= \sum_{z,y=0}^{N-1} u(y)v(z) e^{-2i\pi \frac{\xi(z+y)}{N}} = \left(\sum_{y=0}^{N-1} u(y) e^{-2i\pi \frac{\xi y}{N}} \right) \left(\sum_{z=0}^{N-1} v(z) e^{-2i\pi \frac{\xi z}{N}} \right) = \hat{u}(\xi) \hat{v}(\xi). \end{aligned}$$

\square

Corollaire 1. Soient $u, v \in \mathbb{C}^{\mathbb{Z}_N}$. On a $\widehat{uv} = \frac{1}{N} \hat{u} * \hat{v}$.

Démonstration. → TD. □

Remarque 6. Le théorème précédent est extrêmement important en pratique. En effet, pour deux signaux $u, v \in \mathbb{C}^{\mathbb{Z}_N}$, le calcul de la convolution $u * v$ a une complexité $\mathcal{O}(N^2)$. Mais avec l'algorithme FFT, et la propriété précédente, cette complexité est réduite à $\mathcal{O}(N \log N)$.

2.3 Filtrage

Si $k \in \mathbb{C}^{\mathbb{Z}_N}$ est fixé, l'opération de convolution $\gamma_k : u \mapsto k * u$ est parfois appelé filtrage par le noyau k . Le théorème de la partie précédente traduit l'effet du filtrage sur le spectre :

$$\forall \xi \in \mathbb{Z}, \quad |\widehat{k * u}(\xi)| = |\hat{k}(\xi)| |\hat{u}(\xi)|.$$

On parlera alors de

- filtre passe-bas lorsque les plus grandes valeurs de $\hat{k}(\xi)$ sont concentrées dans les basses fréquences,
- filtre passe-haut lorsque les plus grandes valeurs de $\hat{k}(\xi)$ sont concentrées dans les hautes fréquences,
- filtre passe-bande lorsque les plus grandes valeurs de $\hat{k}(\xi)$ sont concentrées sur une bande de fréquences.

En règle générale, un filtre passe-bas a un effet lissant car il va atténuer les hautes fréquences. À l'opposé, un filtre passe-haut a tendance à mettre en valeur les irrégularités car il va amplifier les hautes fréquences.

Dans beaucoup de cas pratiques, on convole le signal u par une fonction $k \in \mathbb{R}^{\mathbb{Z}_N}$ (appelée *noyau* ou *filtre*) dont le support $\omega \subset \mathbb{Z}_N$ est concentré près de 0 (modulo N). On a alors

$$k * u(x) = \sum_{y \in \omega} k(y) u(x - y).$$

Autrement dit, convoler u par le noyau k revient à faire, en chaque point x une combinaison des valeurs de u au voisinage de x pondérées par les valeurs $k(y)$. Lorsque le support ω correspond aux indices $[a : b]$ (avec $0 \in [a : b]$), on adopte souvent une écriture en tableau pour le noyau renversé

$$\tilde{k} = \left[\begin{array}{|c|c|c|c|c|c|c|} \hline k(b) & k(b-1) & \cdots & k(0) & \cdots & k(a-1) & k(a) \\ \hline \end{array} \right].$$

Cette écriture signifie que pour calculer la convolution $k * u$, il suffit de calculer des produits scalaires en glissant ce noyau renversé sur tout le domaine du signal u . On prendra garde néanmoins au fait que cette écriture laisse une ambiguïté sur la position de l'origine (mais on peut, au besoin, la préciser).

On observera un effet régularisant dès que l'on convole par un noyau k à valeurs positives et de somme 1. Par contre, on verra plus tard des noyaux (filtres) permettant de faire ressortir différentes caractéristiques d'un signal (par exemple les discontinuités, ou certaines bandes de fréquences).

Exemple 4. Pour s entier, $1 \leq s < \frac{N}{2}$, on définit un filtre moyenne $k_s \in \mathbb{C}^{\mathbb{Z}_N}$ en posant

$$\forall y \in \left[-\left\lfloor \frac{N}{2} \right\rfloor : \left\lfloor \frac{N}{2} \right\rfloor \right], \quad k_s(y) = \begin{cases} \frac{1}{2s+1} & \text{si } |y| \leq s \\ 0 & \text{sinon} \end{cases}$$

Dès lors,

$$k_s * u(x) = \frac{1}{2s+1} \sum_{|y| \leq s} u(x - y).$$

La valeur $k_s * u(x)$ est donc une moyenne des valeurs de u au voisinage de x . Ce moyennage a un effet lissant, dans le sens où une discontinuité est remplacée par une rampe linéaire (qui sera d'autant plus douce que s est grand), voir la première ligne de la Fig. 2.

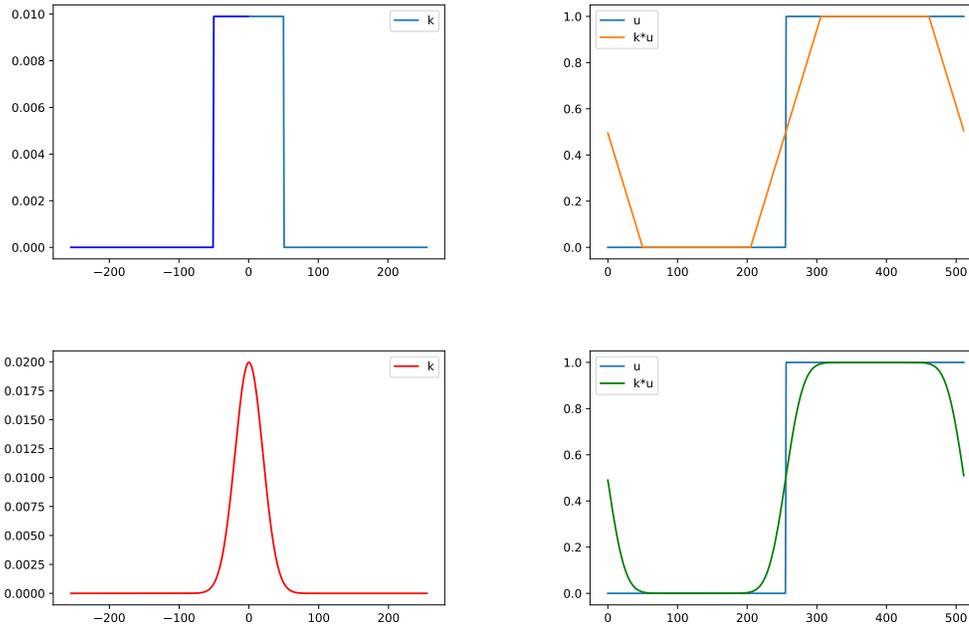


FIGURE 2 – Filtrage avec conditions de bords périodiques. En haut : filtre moyenne, en bas : filtre gaussien.

Exemple 5. Considérons l’opérateur de dérivation discrète (avec conditions de bords périodique) défini sur $\mathbb{C}^{\mathbb{Z}_N}$ par

$$\forall x \in \mathbb{Z}, \quad \partial u(x) = u(x+1) - u(x).$$

On peut voir que cet opérateur s’écrit $\partial u = k * u$ où $k \in \mathbb{C}^{\mathbb{Z}_N}$ est tel que $k(0) = -1$, $k(-1) = 1$, et $k(x) = 0$ si $x \in [1 : N-2]$. Autrement dit, ∂ est l’opérateur de convolution associé au noyau $k = \delta_{-1} - \delta_0$. La TFD de ce filtre s’écrit

$$\forall \xi \in \mathbb{Z}, \quad \hat{k}(\xi) = e^{2i\pi \frac{\xi}{N}} - 1 = e^{i\pi \frac{\xi}{N}} (e^{i\pi \frac{\xi}{N}} - e^{-i\pi \frac{\xi}{N}}) = 2ie^{i\pi \frac{\xi}{N}} \sin\left(\pi \frac{\xi}{N}\right).$$

On a donc $|\hat{k}(\xi)| = 2 \left| \sin\left(\pi \frac{\xi}{N}\right) \right|$. En traçant le spectre de k , on voit que ce filtre est passe-haut. En tableau, on écrit ici $\tilde{k} = \begin{bmatrix} -1 & 1 \end{bmatrix}$.

Exemple 6. Une discrétisation de la dérivée seconde est donnée par

$$\partial^2 u(x) = \partial u(x+1) - \partial u(x) = u(x+2) - 2u(x+1) + u(x).$$

Là encore, on peut voir que ∂^2 est l’opérateur de convolution par le noyau $h = \delta_0 - 2\delta_{-1} + \delta_{-2}$. Le spectre est donné par

$$|\hat{h}(\xi)| = |\hat{k}(\xi)|^2 = 4 \sin^2\left(\pi \frac{\xi}{N}\right).$$

En tableau, on écrit ici $\tilde{k} = \begin{bmatrix} -1 & 2 & 1 \end{bmatrix}$.

Remarque 7. Comme vous le voyez sur les exemples précédents, dans beaucoup de cas de filtrage explicite, le filtre k a un petit support. Pour un tel filtrage, le choix du cadre périodique fait dans ce cours impacte le résultat de la convolution seulement sur les bords. On parle donc de conditions de bords périodiques. Vous pourrez noter que dans les fonctions de Python permettant de calculer des convolutions, diverses conditions de bords sont disponibles (constante, symétrique, etc). Cette terminologie est en lien avec les conditions de bords utilisées dans la résolution d’EDP (conditions de Dirichlet, Neumann, etc).

3 Transformée de Fourier Discrète 2D et Images

Dans cette partie on considérera des fonctions périodiques de deux variables. On notera $\mathbb{Z}_{M,N} = \mathbb{Z}_M \times \mathbb{Z}_N$.

3.1 TFD et Convolution 2D

Définition 6. L'espace $\mathbb{C}^{\mathbb{Z}_{M,N}}$ est l'espace des fonctions $u : \mathbb{Z}^2 \rightarrow \mathbb{C}$ qui sont (M, N) -périodiques, c'est-à-dire que

$$\forall (x, y) \in \mathbb{Z}^2, \quad u(x, y + N) = u(x + M, y) = u(x, y).$$

Définition 7. La transformée de Fourier discrète 2D est définie pour tout $u \in \mathbb{C}^{\mathbb{Z}_{M,N}}$ par

$$\forall (\xi, \zeta) \in \mathbb{Z}^2, \quad \hat{u}(\xi, \zeta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} u(x, y) \exp\left(-2i\pi\left(\frac{\xi x}{M} + \frac{\zeta y}{N}\right)\right).$$

Remarque 8. En adoptant des notations vectorielles $\mathbf{x} = (x, y) \in \mathbb{Z}^2$, $\boldsymbol{\xi} = (\xi, \zeta) \in \mathbb{Z}^2$, on peut aussi écrire avec des notations plus légères

$$\forall \boldsymbol{\xi} \in \mathbb{Z}^2, \quad \hat{u}(\boldsymbol{\xi}) = \sum_{\mathbf{x} \in \Omega} u(\mathbf{x}) e^{-i\langle \boldsymbol{\xi}, \mathbf{x} \rangle}$$

où l'on a posé $\langle \boldsymbol{\xi}, \mathbf{x} \rangle = 2\pi\left(\frac{\xi x}{M} + \frac{\zeta y}{N}\right)$ et $\Omega = [0 : M-1] \times [0 : N-1]$. La plupart des théorèmes et propriétés rencontrés s'adaptent alors aisément au cas 2D et on les énoncera ici sans redémonstration.

Théorème 6. La TFD 2D réalise un isomorphisme de $\mathbb{C}^{\mathbb{Z}_{M,N}}$ sur lui-même. En effet, on a la formule d'inversion de Fourier discrète 2D suivante : pour tout $u \in \mathbb{C}^{\mathbb{Z}_{M,N}}$,

$$\forall (x, y) \in \mathbb{Z}^2, \quad u(x, y) = \frac{1}{MN} \sum_{\xi=0}^{M-1} \sum_{\zeta=0}^{N-1} \hat{u}(\xi, \zeta) \exp\left(2i\pi\left(\frac{\xi x}{M} + \frac{\zeta y}{N}\right)\right).$$

De plus, pour toute $u \in \mathbb{C}^{\mathbb{Z}_{M,N}}$, on a $\|\hat{u}\|^2 = MN\|u\|^2$.

Définition 8. Si $u, v \in \mathbb{C}^{\mathbb{Z}_{M,N}}$, la convolution de u et v est la fonction $u * v \in \mathbb{C}^{\mathbb{Z}_{M,N}}$ définie par

$$\forall (x, y) \in \mathbb{Z}^2, \quad u * v(x, y) = \sum_{a=0}^{M-1} \sum_{b=0}^{N-1} u(x-a, y-b)v(a, b).$$

Théorème 7. Pour toutes $u, v \in \mathbb{C}^{\mathbb{Z}_{M,N}}$, $\widehat{u * v} = \hat{u}\hat{v}$.

3.2 Modélisation d'images

Une image en niveaux de gris de taille $M \times N$ est identifiée à une fonction

$$u : \Omega \longrightarrow \mathbb{R}$$

où $\Omega = [0 : M-1] \times [0 : N-1]$. La valeur $u(x, y)$ représente le niveau de gris du pixel $(x, y) \in \Omega$. En informatique, la plupart des formats d'images sont quantifiés sur 256 niveaux de gris, ce qui signifie que u est à valeurs dans $[0 : 255]$. La valeur 255 représente le blanc, 0 le noir. Néanmoins, beaucoup d'algorithmes sont basés sur des calculs en nombres réels. En Python, on ramènera donc souvent l'échelle $[0 : 255]$ sur $[0, 1]$ par proportionnalité (dans ce cas, 1 = blanc, et 0 = noir). D'ailleurs, certains calculs que l'on effectuera peuvent créer des valeurs < 0 et > 1 . En général, on seuillera les valeurs en dehors de $[0, 1]$ (en remplaçant simplement u par $\min(\max(u, 0), 1)$). La gamme de valeurs prises par u est parfois appelée *dynamique de l'image* (notamment par les testeurs d'appareils photo).

Mentionnons dès maintenant que la perception humaine du contenu géométrique d'une image n'est pas basée sur la valeur absolue des niveaux de gris, mais sur les différences de contrastes locales. Par

conséquent, ajouter une constante à u ne modifie pas le contenu géométrique perçu (tant que l'on reste dans l'intervalle $[0, 1]$). Ainsi, lorsque l'on cherchera à observer des détails fins d'une image à un endroit, on pourra extraire une image à cet endroit, et réhausser le contraste en appliquant une fonction affine aux valeurs, voir Fig. 3.

On notera la convention inhabituelle utilisée ici pour les coordonnées : l'axe des x est en vertical de haut en bas, et l'axe des y en horizontal de gauche à droite. Cette convention vient de la représentation en Python avec des `numpy.array`, et est donc la même convention utilisée avec les indices matriciels. Lorsqu'on manipule des images, on prendra toujours garde à la convention de coordonnées, selon le langage et la librairie utilisés.

Dans ce cours, on ne pourra pas étudier en détail les procédés d'acquisition d'images en jeu. Mentionnons simplement que dans un appareil photo, les valeurs des pixels sont acquises grâce à l'utilisation d'un capteur CCD. Ce capteur a la forme d'une matrice, qui va compter, pour chaque pixel, le nombre de photons arrivant sur la cellule carrée correspondante pendant la période d'obturation. Le comportement aléatoire des photons fait que l'image acquise sera déjà affectée par un bruit. L'image pourra aussi être victime d'autres artefacts, par exemple : flou (de focus, ou de bouger), aberrations optiques, etc.

En Python, il existe plusieurs librairies de manipulations d'images. La librairie `matplotlib` contient déjà des fonctions basiques pour la manipulation d'images (dans `matplotlib.plot`, et dans `matplotlib.image`). En ce qui concerne le filtrage (voir parties suivantes), on utilisera dans ce cours la librairie `scipy.ndimage`. Une librairie plus complète est donnée par la *Python Image Library* (PIL). Aussi, une librairie plus récente (construite autour de concepts issus de l'apprentissage machine) est donnée par `scikit-image`.

En TP, vous aurez l'occasion d'afficher des images avec `matplotlib.pyplot.imshow`, et vous verrez que certaines conventions de visualisation sont appliquées. D'abord, pour afficher effectivement en niveaux de gris, il faudra préciser une `colormap` avec l'option `cmap=gray`. Cette fonction renormalise alors la dynamique automatiquement de façon à exploiter toute la gamme des valeurs de l'image ; on peut contourner cela en précisant les options `vmín=0` et `vmax=1` (ou d'autres valeurs si besoin). On notera aussi que les axes sont inversés par rapport à la convention indicielle de Python. Ainsi, une image de dimension 1024×768 sera représentée par un `numpy.array` de taille $(768, 1024)$. **Néanmoins**, dans ce cours, pour simplifier, on restera proche de la représentation en tableaux utilisée dans Python (sans chercher à respecter la convention standard). Autrement dit, la valeur `u[x, y]` désignera la valeur de l'image à la ligne x et à la colonne y .

Terminons par une remarque sur la représentation des images en couleurs. La représentation la plus simple des images couleurs (adoptée dans plusieurs formats de fichier standard) consiste à utiliser trois canaux séparés pour les composantes Rouge, Vert et Bleu. L'oeil humain (hors cas particulier, de daltonisme par exemple) comporte des récepteurs dont la sensibilité peut d'une certaine manière se décomposer sur ces trois couleurs. On parlera donc d'images RGB (pour *Red, Green, Blue*). En Python, une image couleurs RGB sera donc représentée par un `numpy.array` de taille $(M, N, 3)$. Mentionnons qu'il existe d'autres espaces (comme l'espace YUV) permettant de représenter les couleurs d'une façon différente (en séparant par exemple les composantes de teinte, luminance et chrominance), mais on ne les utilisera pas dans ce cours. En fait, trouver une représentation mathématique de la couleur qui soit perceptuellement satisfaisante est une question de recherche toujours ouverte.

3.3 TFD d'images

À l'image $u : \Omega \rightarrow \mathbb{R}$ de taille $M \times N$, on associera naturellement son prolongement par périodicité $u \in \mathbb{C}^{\mathbb{Z}^{M,N}}$ que l'on continuera à noter u . On pourra donc appliquer les outils développés dans la Partie 3 et calculer notamment la TFD bidimensionnelle de l'image u , que l'on notera bien sûr \hat{u} . Le spectre (d'amplitude) de u est encore défini par $|\hat{u}|$ (que l'on visualisera souvent en échelle logarithmique, c'est-à-dire en regardant $\log(1 + |\hat{u}|)$ plutôt que $|\hat{u}|$).

De façon analogue au cas 1D, la TFD d'une image $u \in \mathbb{C}^{\mathbb{Z}^{M,N}}$ est encore une suite $u \in \mathbb{C}^{\mathbb{Z}^{M,N}}$. On peut donc la représenter entièrement en prenant n'importe quel rectangle $\Theta \subset \mathbb{Z}^2$ de taille $M \times N$. Néanmoins, on privilégiera souvent deux choix possibles

- $\Theta_0 = \Omega = [0 : M - 1] \times [0 : N - 1]$,
- $\Theta_c = [\lceil -\frac{M}{2} \rceil : \lfloor \frac{M-1}{2} \rfloor] \times [\lceil -\frac{N}{2} \rceil : \lfloor \frac{N-1}{2} \rfloor]$.

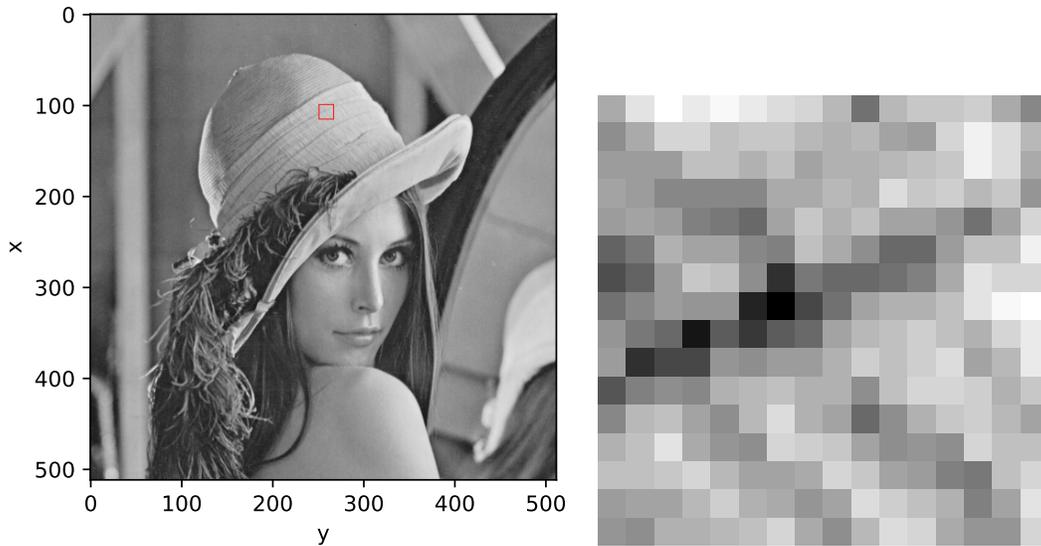


FIGURE 3 – À gauche, on montre une image numérique u de taille 512×512 . À droite, on montre l’image (dite aussi *patch*) de taille 16×16 localisée en position $(100, 250)$ et encadrée en rouge dans l’image de gauche. On notera bien que l’axe des y est renversé par rapport aux représentations habituelles.

En visualisation avec Θ_0 , la fréquence $(0, 0)$ est placée en haut à gauche, alors qu’avec Θ_c , elle est placée au milieu du domaine. Puisque les spectres d’images naturelles sont paires et sont concentrés près de zéro, on privilégiera la deuxième solution avec Θ_c , de façon à observer l’essentiel du spectre en plein milieu de l’image. Pour faire la différence, on parlera de spectre lorsqu’on visualise sur Θ_0 et de spectre centré lorsqu’on visualise sur Θ_c . En pratique, pour visualiser le spectre centré, il suffit d’appliquer la fonction `fftshift` qui applique la bonne translation périodique à la TFD.

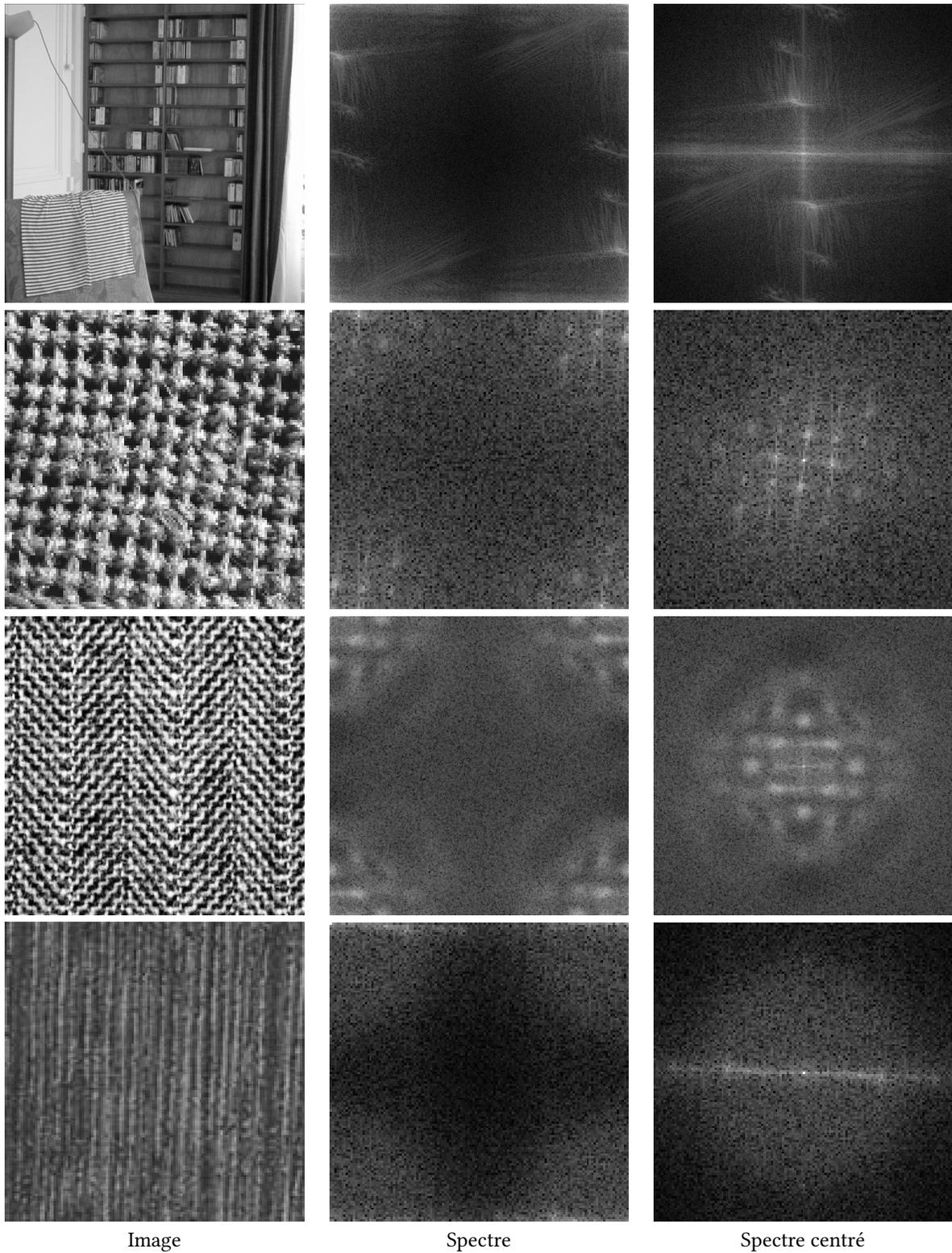
On pourra observer les spectres de plusieurs images sur la Fig. 4. Ces images ont été choisies car elles présentent des oscillations (rayures) à un endroit dans l’image à une fréquence très particulière. On notera que dans ce cas, le spectre présentera un pic correspondant à la direction orthogonale aux rayures. La position du pic dans le domaine fréquentiel ne dépend que de la direction et de la fréquence d’oscillation des rayures. Elle ne dépend pas de l’endroit où sont localisées ces rayures dans l’image. Par conséquent, il est parfois difficile de lire le spectre, puisque l’information de l’image s’y mélange : ceci est simplement dû à la définition même de la TFD, qui effectue la somme sur tout le domaine Ω . Néanmoins, on pourra s’entraîner à chercher à quels endroits de l’image correspondent les pics observés dans le domaine fréquentiel.

On notera également que pour des images naturelles, la visualisation du spectre proposée conduira à l’apparition de deux lignes horizontales et verticales passant par la fréquence $(0, 0)$. Celles-ci sont essentiellement dues au fait que l’on calcule la TFD en supposant que l’image est (M, N) -périodique. C’est donc la discontinuité d’un bord à l’autre de l’image qui crée ces deux lignes à forte énergie dans le spectre. Il existe des moyens d’atténuer cet artefact en modifiant un peu l’image en amont.

3.4 Filtres 2D

De même que dans le cas 1D, on va utiliser les opérations de convolution pour filtrer l’image. Ceci s’explique par le fait que de nombreuses transformations d’images (dérivées, flou, etc) peuvent être modélisées par des convolutions. Dans ce paragraphe, on définit quelques filtres, dont vous pourrez constater l’effet en TP.

Exemple 7. Considérons les opérateurs de dérivation discrète (avec conditions de bords périodiques)



Image

Spectre

Spectre centré

FIGURE 4 – Pour chaque ligne, de gauche à droite, on montre une image numérique, le spectre, et le spectre centré. On observera qu’il existe une relation entre les oscillations présentes dans l’image et les pics que l’on peut voir dans le spectre.

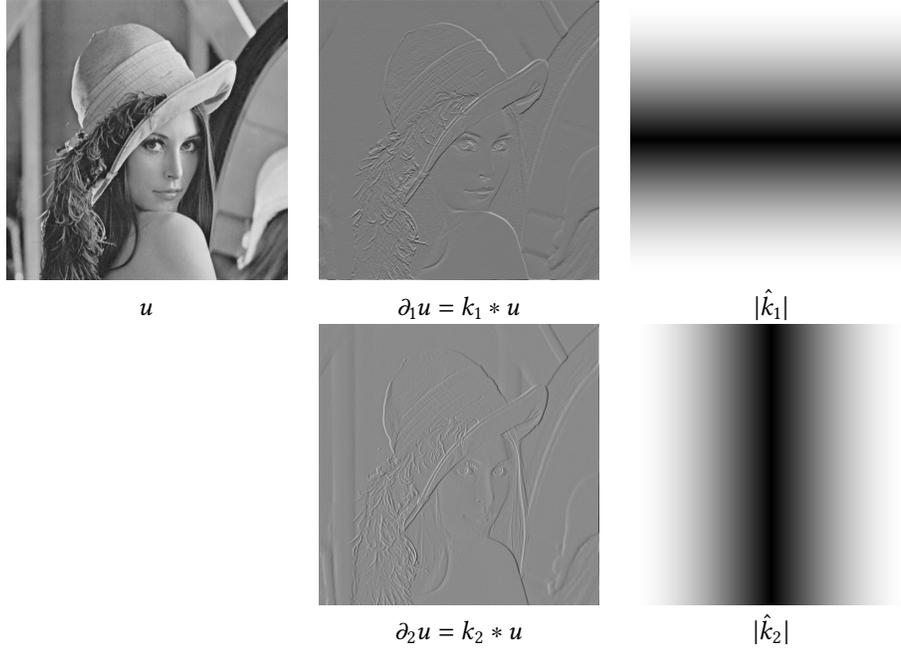


FIGURE 5 – **Dérivées discrètes de l'image à l'ordre 1.** Sur la première colonne, on montre une image u . Sur la deuxième colonne, on montre les images $\partial_1 u$, $\partial_2 u$. Sur la troisième colonne, on montre les spectres centrés $|\hat{k}_1|$, $|\hat{k}_2|$ des filtres correspondants. On notera que le filtre ∂_1 (resp. ∂_2) permet de faire ressortir des contours horizontaux (resp. verticaux). On notera aussi que l'on conserve le signe de la discontinuité, c'est-à-dire que dans les images $\partial_1 u$ et $\partial_2 u$, on voit du blanc aux endroits où l'on passe du foncé au clair.

définis sur $\mathbb{C}^{\mathbb{Z}_{M,N}}$ par

$$\forall (x, y) \in \mathbb{Z}^2, \quad \begin{cases} \partial_1 u(x, y) = u(x+1, y) - u(x, y) \\ \partial_2 u(x, y) = u(x, y+1) - u(x, y) \end{cases}$$

On peut voir que cet opérateur s'écrit $\partial_1 u = k_1 * u$ où $k_1 \in \mathbb{C}^{\mathbb{Z}_{M,N}}$ qui vaut 0 sur tout Ω , sauf en deux points

$$k_1(0, 0) = -1, \quad k_1(-1, 0) = 1.$$

On peut montrer que le spectre de ce filtre s'écrit

$$\forall (\xi, \zeta) \in \mathbb{Z}^2, \quad |\hat{k}_1(\xi, \zeta)| = 2 \left| \sin \left(\frac{\pi \xi}{M} \right) \right|.$$

Ce filtre permet de faire ressortir les contours horizontaux, voir Fig. 5.

Là encore, on adoptera une écriture en tableau pour décrire le noyau renversé :

$$\tilde{k}_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \tilde{k}_2 = \begin{bmatrix} -1 & 1 \end{bmatrix}.$$

On prendra garde au fait que ces écritures décrivent bien des noyaux bidimensionnels, avec des tableaux à double entrée.

Exemple 8. Une discrétisation des dérivées secondes est donnée par

$$\partial_1^2 u(x, y) = u(x+2, y) - 2u(x+1, y) + u(x, y).$$

$$\partial_2^2 u(x, y) = u(x, y+2) - 2u(x, y+1) + u(x, y).$$

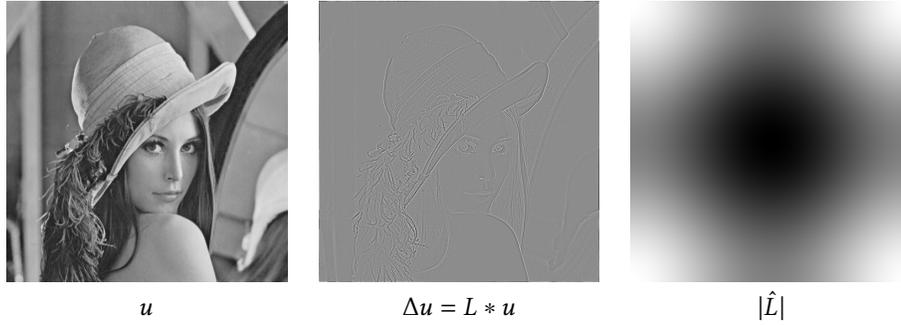


FIGURE 6 – **Laplacien discret de l'image**. De gauche à droite, on montre une image u , son laplacien discret Δu ainsi que le spectre centré du filtre correspondant $|\hat{L}|$. Sur le spectre, on peut voir que $|\hat{L}|$ est un filtre passe-haut. On remarquera que $|\Delta u|$ est grand près des contours de l'image u . On notera aussi que le bruit de l'image est réhaussé (et donc l'extracteur de contours correspondant doit être amélioré si l'on veut le rendre moins sensible au bruit).

Les représentations en tableau de ces deux filtres sont données respectivement par

$$\begin{array}{|c|} \hline 1 \\ \hline -2 \\ \hline 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline \end{array}$$

Le spectre est donné par

$$\forall (\xi, \zeta) \in \mathbb{Z}^2, \quad |\hat{h}_1(\xi)| = 4 \sin^2 \left(\frac{\pi \xi}{M} \right).$$

En ajoutant les deux dérivées secondes et en recentrant, on obtient une discrétisation du laplacien

$$\Delta u(x, y) = \partial_1^2 u(x-1, y) + \partial_2^2 u(x, y-1).$$

C'est aussi un opérateur de convolution par un noyau L , symétrique, qui s'écrit en tableau

$$L = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}.$$

On peut voir que la TFD de ce filtre s'écrit

$$\hat{L}(\xi) = -4 \left(\sin^2 \left(\frac{\pi \xi}{M} \right) + \sin^2 \left(\frac{\pi \zeta}{N} \right) \right) = -|\hat{k}_1(\xi, \zeta)|^2 - |\hat{k}_2(\xi, \zeta)|^2.$$

Vous pourrez voir en TP que ce filtre permet d'extraire les contours de l'image, voir Fig. 6. Néanmoins, il reste assez sensible au bruit en comparaison d'autres extracteurs de contours plus sophistiqués.

Exemple 9. Terminons par l'exemple du filtre gaussien $g_s \in \mathbb{C}^{\mathbb{Z}_{M,N}}$ défini pour $s > 0$ par

$$\forall |\xi| \leq \frac{M}{2}, \forall |\zeta| \leq \frac{N}{2}, \quad g_s(x, y) = c \exp \left(-\frac{x^2 + y^2}{2s^2} \right),$$

où la constante $c > 0$ est choisie de sorte que la somme des coefficients de g_s soit égale à 1 (ce qui revient à imposer $\hat{g}_s(0) = 1$). On verra en TP que ce filtre permet de modéliser un opérateur de flou isotrope sur l'image. Le paramètre s permet de régler l'étalement du noyau de flou, voir Fig. 7.



FIGURE 7 – **Flou gaussien.** Sur la première colonne, on montre une image u . Sur la deuxième colonne, on montre l'image floutée $g_s * u$ où g_s est le noyau gaussien d'écart-type s . Sur la troisième colonne, on montre le noyau g_s et sur la quatrième colonne son spectre centré. Plus s est grand, plus le noyau g_s est étalé en espace et plus $g_s * u$ paraît flou. À l'inverse, lorsque s grandit, le spectre $|\hat{g}_s|$ est moins étalé (et donc le filtrage par g_s détruit une plus grande partie du spectre).