Analyse de Fourier numérique

Le traitement d'image ou de signal par transformée de Fourier numérique est très similaire à l'étude des réseaux pour représenter les cristaux puisque les données informatiques sont stockées de manière discrète, tels nos noeuds ponctuels. Bien que les télévisions à ultra-haute définition (UHDTV) offrent maintenant des images avec des millions de pixels, nous sommes toutefois encore loin de la densité d'atomes dans un cristal de l'ordre de 10^{22} cm^{-3} résultant en des niveaux discrets d'énergie si rapprochés qui s'approxime en une densité d'états continue permettant les calculs analytiques. Sans cette approximation pour un nombre de pixels fini dans une image, les calculs numériques avec des processeurs contemporains peuvent toutefois de traiter stocker toutes les données.

La transformation de Fourier par ordinateur nécessite d'être représentée en matrices discrètes finies autant en entrée qu'en sortie. Ceci n'est pas inclus dans les définitions habituelles impliquant des intégrales et sommations infinies permettant d'aussi bien représenter des fonctions continues. Toutefois, la table 1 met en évidence les cas périodiques qui permettent de simplifier en série de Fourier dont la séquence des coefficients peut se représenter en matrice discrète. Qui plus est, la discrétisation de l'entrée en intervalles réguliers rend la sortie spectrale périodique, d'où les sommations pourront être tronquées à une période avec un nombre fini de termes (au sens strict si les conditions du théorème d'échantillonnage de Nyquist-Shannon sont satisfaites). Cette approche est dite une transformation de Fourier discrète (*discrete Fourier transform*, DFT) en analyse numérique.

	Entrée		Sortie	
	Périodicité	Discrétisation	Périodicité	Discrétisation
Transformée de Fourier	facultative	continue	NON périodique	continue
	facultative	discrète	périodique	continue
Série de Fourier	périodique	continue	NON périodique	discrète
	périodique	discrète	périodique	discrète

Table 1 – Caractéristiques des fonctions en entrée et sortie de la transformation de Fourier.

En physique de l'état solide, le principe de la DFT rejoint celui des conditions frontières périodiques de Born–von Karman appliquées à un réseau de Bravais qui "boucle" le cristal sur lui-même pour former une période. Le nombre de mailles formant un cristal est donc l'analogue 3D du nombre de pixels formant une image 2D, ou encore du nombre d'échantillons d'un signal 1D en fonction du temps par exemple. Plusieurs propriétés spectrales dans la discussion qui suit sont donc aussi applicables à l'espace des vecteurs d'ondes $\vec{k}$ et la périodicité de son réseau réciproque $\vec{K}$.

Domaines et échantillonnage numérique

Rappelons d'abord que la fonction sortante de la transformation de Fourier est duale à la fonction d'entrée. Le domaine de la fonction de sortie correspond à une variable indépendante fréquentielle qui hérite de l'inverse des unités du domaine en entrée. Par exemple, la transformation d'un signal d'entrée variant dans le temps nous donnera de l'information sur les fréquences temporelles, telles les notes de musique en Hz pour une variable indépendante d'entrée en secondes. La transformation d'un signal d'entrée variant dans l'espace mènera quant à elle à des fréquences spatiales en nm^{-1} ou autre inverse d'unité de longueur. Finalement, il ne faut pas oublier que la fonction de sortie est complexe même si la variable indépendante fréquentielle demeure réelle. Cette variable dépendante peut s'exprimer en forme cartésienne (Re, Im) ou en forme polaire avec les diagrammes de Bode, ce qui est plus commun avec la notion de spectre pour la norme (module) de la fonction, mais sans oublier l'argument. Ce dernier indique la phase relative, généralement en radians, des oscillations sinusoïdales correspondant à chaque fréquence du domaine de Fourier.

La discrétisation numérique en un nombre entier N d'intervalles réguliers Δx, ou échantillonnage, du domaine $\left[\frac{-L}{2}, \frac{L}{2}\right]$ de la fonction d'entrée donne un vecteur (ou matrice selon le nombre de dimensions) étalonné [1] ainsi

$$x = \left[\tfrac{-N}{2}\Delta x \quad \tfrac{-N+1}{2}\Delta x \ldots \quad -\Delta x \quad 0 \quad \Delta x \quad 2\Delta x \ldots \quad n\Delta x \ldots \quad \tfrac{N-1}{2}\Delta x \quad \tfrac{N}{2}\Delta x\right] \quad \text{où} \quad n \in \left[\frac{-N}{2}, \frac{N}{2}\right],$$

montrant que la taille du domaine couvre $L = N\Delta x$. Plusieurs algorithmes numériques qui implémentent la DFT, notamment *Fast Fourier Transform*, traitent seulement la matrice représentant la variable dépendante laissant le soin à l'utilisateur d'étalonner correctement le domaine de sortie en fréquence. Il faut éviter le piège de penser que la résolution Δx ne change pas et se rappeler que les domaines sont des inverses en gardant le même nombre d'intervalles :

$$N = \frac{L}{\Delta x} = \frac{1/\Delta x}{1/L} := \frac{f_s}{\Delta f}$$

où le domaine fréquentiel devient $\left[\frac{-1}{2\Delta x}, \frac{1}{2\Delta x}\right]$ avec $\frac{1}{\Delta x}$ s'interprétant comme une fréquence d'échantillonnage f_s.

La division en intervalles réguliers $\Delta f = \frac{1}{L} = \frac{1}{N\Delta x}$ donne la résolution fréquentielle qui permet d'étalonner le vecteur de fréquences ainsi

$$f = \left[\tfrac{-N}{2}\Delta f \quad \tfrac{-N+1}{2}\Delta f \ldots \quad -\Delta f \quad 0 \quad \Delta f \quad 2\Delta f \ldots \quad n\Delta f \ldots \quad \tfrac{N-1}{2}\Delta f \quad \tfrac{N}{2}\Delta f\right] \quad \text{où} \quad n \in \left[\frac{-N}{2}, \frac{N}{2}\right].$$

Pour terminer cette discussion superficielle de l'analyse de Fourier numérique, notons que les fréquences $\pm\frac{N\Delta f}{2}$ aux extrémités du domaine correspondant à $\frac{f_s}{2}$ sont dites fréquences de Nyquist. Si le signal contient des fréquences supérieures à celles-ci, l'information correspondante ne sera pas correctement traitée par l'échantillonnage et la DFT.

Exemple 1D

Le signal à la gauche de la figure 1 ne présente pas d'oscillations sinusoïdales apparentes ni périodicité, mais il peut tout de même se représenter par une superposition d'oscillations grâce à la transformée de Fourier, dont la partie réelle en sortie est aussi illustrée. Le maximum à fréquence nulle indique bien qu'il y décalage vertical de la moyenne du signal au-dessus de zéro. Pour ce signal d'entrée réel, la symétrie paire de la partie réelle est telle qu'attendue dans le domaine de Fourier, qui présenterait aussi une anti-symétrie impaire correspondante pour la partie imaginaire. En effet, l'opération est hermitienne $\mathrm{DFT}(f) = \mathrm{DFT}^*(-f)$, donnant la possibilité de traiter seulement les fréquences positives. Il faut donc porter attention à l'index correspondant à la fréquence nulle selon le logiciel utilisé et s'il y a lieu, s'assurer que toutes les conditions pour inverser la DFT sont remplies.

1. Un domaine temporel serait préférable sans temps négatif, ce qui n'est pas problématique grâce à une translation menant à $n \in [0, N]$ qui ne change pas l'analyse.

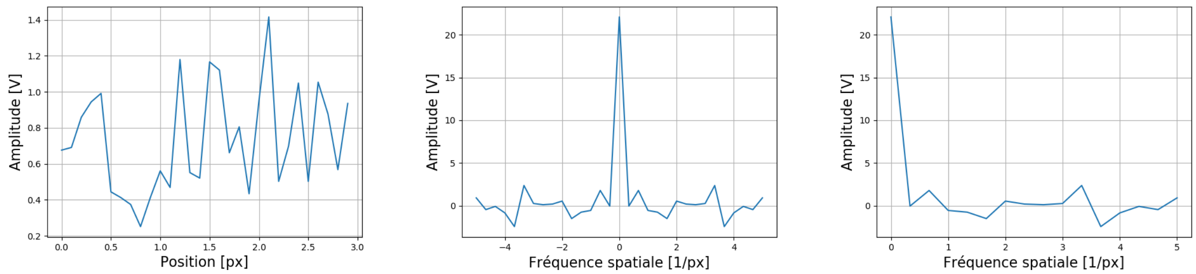

Figure 1 – De gauche à droite : Signal d'entrée quasi-aléatoire ; Partie réelle de la transformée de Fourier sans étalonnage avec une fréquence nulle centrée ; Ibidem gardant seulement les fréquences positives.

Exemple 2D

Lorsque le signal d'entrée est une fonction à deux variables indépendantes donnée par une image ou autre graphique, elle est s'associe à une matrice via les niveaux de gris ou autre échelle de couleur qui donne la correspondance à une valeur pour chaque pixel. Certains logiciels vont convertir le domaine de sortie en coordonnées polaires où l'angle est associé à l'orientation d'éventuelles oscillations dans l'image. La fréquence nulle est alors placée au centre du domaine de Fourier puis les fréquences croissent linéairement vers l'extérieur le long du rayon.

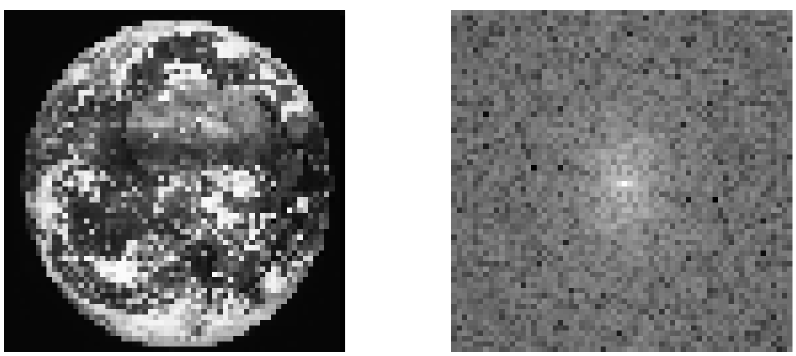

Figure 2 – Image 64px × 64px à gauche et la partie réelle de sa transformée de Fourier à droite.

Les algorithmes implémentant la DFT préfèrent des images en entrée aussi hautes que larges et vont même parfois jusqu'à étirer une dimension des images rectangulaires avant de faire la transformation. Une fois dans le domaine fréquentiel pour ce dernier cas, l'image sera carrée en conservant la plus grande dimension de l'image d'origine. Par exemple, une image 256px × 512px en entrée deviendra 512px × 512px en sortie et la résolution fréquentielle conserve la même définition en utilisant la plus grande dimension en entrée. Qui plus est, les calculs numériques sont plus rapides si cette dimension est divisée en une puissance de 2 en pixels (256, 512, 1024, ...), d'où certains algorithmes referont même l'échantillonnage pour obtenir un tel nombre de pixels. Bref, la morale est qu'il faut prendre le temps de lire les instructions spécifiques à un logiciel donné.

Annexe

Logiciels proposant des algorithmes évaluant la DFT

ImageJ

ImageJ est un programme en accès libre d'analyse d'image simple et efficace qui fonctionne sur tous les systèmes d'exploitation (OS). Simplement accéder au site web *www.imagej.net* pour le télécharger et en lire la documentation. Il sera nécessaire de télécharger Fiji, qui contient ImageJ, pour des récentes raisons de mise à niveau.

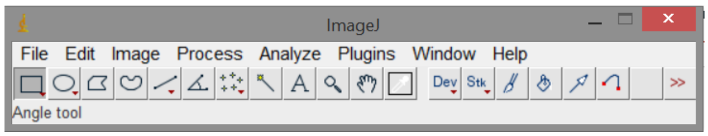

FIGURE 3 – Interface d'ImageJ.

Pour manipuler une image, on commence par aller dans File → Open, puis on sélectionne l'image à analyser. L'image va apparaître dans un fenêtre séparée. Pour manipuler l'image, simplement sélectionner la fenêtre de l'image, puis utiliser les différentes fonctions de la barre d'outils d'origine. ImageJ garde en mémoire la dernière fenêtre sélectionnée.

La transformée de Fourier se trouve sous Process → FFT et on peut lire le rayon, l'angle et la valeur d'un pixel en le pointant avec le curseur et en observant la valeur sous la barre d'outils.

Il est même possible de créer des librairies imageJ et d'importer des fonctionnalités. On peut même travailler sur de grandes quantités d'images en écrivant des scripts séquentiels d'analyse.

Python

Python contient plusieurs librairies pouvant permettre de faire des transformées de Fourier.

Numpy Cette librairie bien connue contient tous les outils mathématiques pour créer ses propres fonctions d'analyse, en autant que l'utilisateur soit bien à l'aise avec le langage. Sa librairie *numpy.fft* contient plusieurs fonctions permettant d'effectuer différents types de FFT. Il est important d'aller voir la documentation de ces fonctions (facile à trouver suite à une recherche sur le web) puisque plusieurs FFT ne prennent en argument qu'une seule série de données, d'où le domaine de Fourier est adimensionnel *a priori* tel que discuté précédemment. Il faut donc déterminer le domaine *a posteriori*.

Pycroscopy Pycroscopy est un exemple de librairie python spécialisée en analyse d'image. Celle-ci est écrite spécialement pour la micrographie, mais, comme toute librairie Python, peut s'avérer utile dans d'autres domaines. La classe *pycroscopy.processing.fft* contient les outils nécessaire pour traiter plusieurs images simultanément.

Matlab

Matlab contient à la base plusieurs fonctions pouvant servir à l'analyse d'image. Dans ce language, tous les éléments sont considérés comme vectoriels, ce qui simplifie le code lors du traitement d'image. Les fonctions *fft* et *fft2* permettent de faire une FFT rapidement. Par contre, cette FFT prend elle aussi une seule série de données avec un domaine de sortie adimensionnel.

IDL

Le logiciel de traitement d'images IDL est couramment utilisé en astrophysique et offre probablement des outils d'analyse numérique de Fourier. La NASA héberge un dépôt de librairies IDL complémentaires à cette adresse *https ://idlastro.gsfc.nasa.gov/homepage.html*.