TP3 - IMN359 - Transformée de Fourier Discrète, Convolution & Échantillonnage

Maxime Descoteaux

18 novembre 2024

Vous devez rédiger un rapport avec les solutions et me remettre un zip avec votre code Python. Commentez le code et assurez-vous que je puisse reproduire vos résultats et figures. Séparez votre code en différents fichiers pour faciliter la lecture. Des points seront attribués pour la qualité du document et ses figures (5 points), et la qualité du code Python (5 points).

La date de remise sera déterminée en classe.

1. **TFD et Convolution [20 pts]**
   La transformée de Fourier de la Gaussienne $g(t) = \exp(-t^2/(2\sigma^2))$ est aussi une Gaussienne $G(f) = \sqrt{2\sigma^2\pi}\exp(-2\pi^2\sigma^2 f^2)$. (https ://mathworld.wolfram.com/FourierTransformGaussian.html)

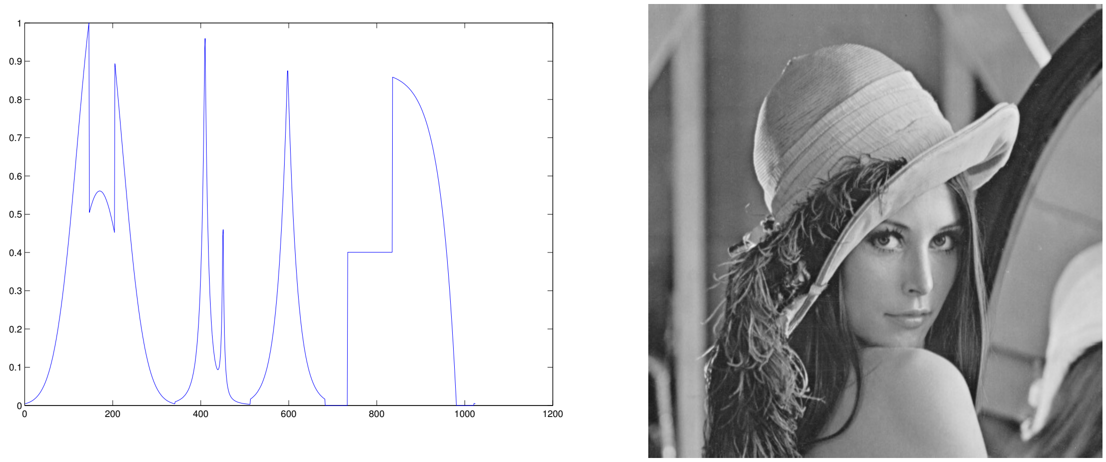
   (a) En Python, construiser et illustrer la Gaussienne 1D et sa TF de la partie (a), utilisant un $\sigma = 6$ et 1024 points discrets de temps t.

   (b) En utilisant la *fft* de Python, calculer et illustrer sa transformée de Fourier $G(f)$.
   Visualiser le résultat et assurez-vous que le résultat théorique en b) correspond à votre résultat trouvé numériquement.

   (c) Charger le signal 1D 'piece-regular' (piece-regular.mat). En utilisant la fonction *rand*, ajouter-y un bruit additif avec un écart type de votre choix (piece-regular + τ·rand(1024,1)). Si $\tau = 0.1$, vous ajoutez 10% de bruit. Illustrer le signal et sa version bruité en faisant un subplot.

   (d) Illustrer de façon pratique le théorème de convolution. $(f * g)(t) = \mathcal{TF}^{-1}(F(w)G(w))$.

      i. D'abord, calculer la convolution du signal bruité et du filtre Gaussien. Illustrer le résultat.
      ii. Ensuite, faites la multiplication des transformées de Fourier respectives dans l'espace de Fourier et calculer la transformée de Fourier inverse. Illustrer le résultat.
      iii. Que remarquez-vous ?
      iv. Montrer que le théorème de Plancherel est respecté (conservation d'énergie).



Signal 1D 'piece-regular' Image de Lena

FIGURE 1 – Image et signal classiques utilisés en imagerie.

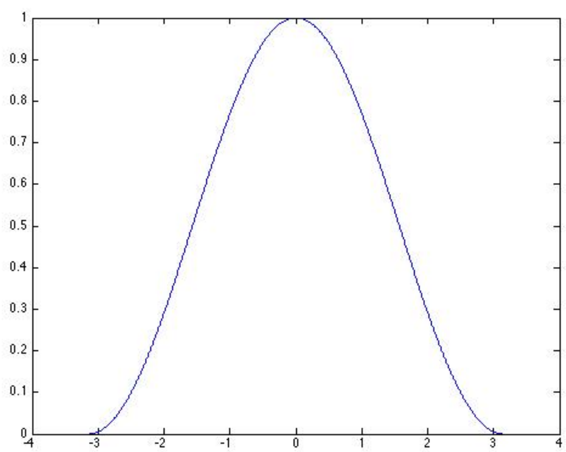


FIGURE 2 – Exemple d'un filtre cosinusoidal sur l'intervalle $[-\pi, \pi]$.

2. **TFD 2D [20 pts]**
   (a) Construiser un filtre cosinusoidal 1D avec un support de $[-\pi, \pi]$. Illustrer-le.
   (Indice : Votre filtre devrait ressembler à une fonction porte centrée à l'origine avec des transitions moins abrutes à cause du cosinus tel qu'illustré en Figure 2)

   (b) Généraliser ce filtre en 2D. Illustrer-le. (Explorer la fonction *mesh* de Python pour l'illustrer.)

(c) Charger l'image de Lena (lena.mat) et calculer sa transformée Fourier 2D (*fft2*). Illustrer Lena et l'image du spectre dans l'espace de Fourier sur une échelle logarithmique. (Faites un subplot) Que remarquez vous dans le contenu fréquentiel de Lena ?

(d) En traitement d'image, pour estimer le vrai contenu fréquentiel d'une image et non pas ses artéfactes, on *multiplie* souvent l'image par une fonction de fenêtrage. Utiliser le filtre cosinusoidal créé en b) à cette fin. Masquer Lena par la fonction de fenêtrage cosinusoidal (une simple multiplication dans le domaine image) et recalculer la transformée de Fourier sur l'image de Lena fenêtrée. Illustrer le résultat de fenêtrage et sa transformée de Fourier (toujours sur une échelle log). Que remarquez-vous ?

(e) Directement dans l'espace de Fourier, appliquer un filtre passe-bas sur le spectre de Lena. Votre filtre passe-pas devrait garder seulement les basses fréquences dans un carré autour de l'origine du spectre. Définisser votre filtre de telle sorte qu'il garde seulement M fréquences basses ($\sqrt{M}$ fréquences en x et $\sqrt{M}$ fréquences en y). Illustrer l'image du spectre filtré pour 3 valeurs de M. Faites un subplot.
(Conseil : Faites cette partie proprement. Vous en aurez besoin dans le TP4)

(f) Reconstruire l'image de Lena à partir des 3 spectres filtrés. Illustrer les résultats sur un subplot. Rapporter l'erreur quadratique moyenne de chaque image reconstruite et le pourcentage de points utilisés pour ces reconstructions. Que remarquez-vous ?

3. **Échantillonnage et TFD [20 pts]**
IRM.mat contient le contenu fréquentiel d'un cerveau de chat *ex-vivo* passé en imagerie par résonance magnétique sur l'IRM 7 Tesla du Centre d'Imagerie Moléculaire de Sherbrooke. C'est vraiment ce qui sort de l'acquistion IRM (k-space dans le jargon IRM) avant que le module de reconstruction de l'imageur produise l'image en faisant la iFFT (transformée de Fourier inverse).

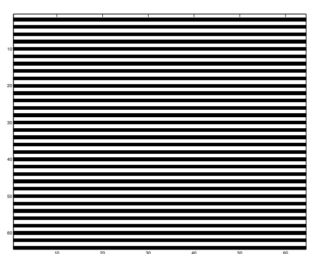
(a) À partir de l'image k-space, mettez une ligne sur N ($N = 2$, 3, 4) à zéro. C'est-à-dire, remplacer par des zéros une ligne sur N du k-space. C'est comme si vous aviez échantillonné ou mesuré le signal IRM avec une ligne sur N en moins.
Illustrez les images reconstruites en faisant la iFFT de l'image du k-space et rapporter le pourcentage de points utilisés dans le titre de la figure. Que remarquez-vous ?

(b) À partir de l'image k-space, faites la même chose sur les colonnes et illustrez la reconstruction en rapportant le pourcentage de points utilisez. Que remarquez-vous ?

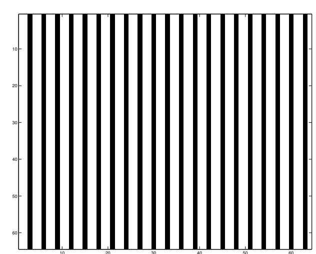
(c) Ajouter des zéros à l'image de k-space pour qu'elle soit de taille, 600x600, 850x850, 1024x1024. Illustrez vos reconstructions. Que remarquez-vous ? Cette technique s'appelle le *zero padding.* D'après vous, le zero padding donne une image super-résolution ?
Bonus : Analyser mathétiquement ce qui se passe en faisant du zero padding.

(d) Maintenant, faites un échantillonnage radial de l'image de k-space. C'est-à-dire, tracez des droites traversant l'origine à chaque angle θ (voir la Figure 2). Faites-le pour 3 θ différents, illustrez les reconstructions en rapportant le pourcentage de points utilisez. Que remarquez-vous ?

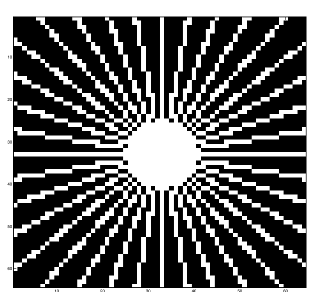
(e) Enfin, faites une échantillonnage aléatoire de l'image de k-space en gardant un certain pourcentage P des points du k-space. Faites-le pour 3 P différents, illustrez les reconstructions en rapportant le pourcentage de points utilisez. Que remarquez-vous ?



50% des points (1 ligne sur 2)



67% des points (1 ligne sur 3 éliminée)



30% des points (18 rayons)

40% des points

Figure 3 – Échantillonnage du k-space.