TP 4 - Au delà de Fourier
(DCT, DCT locale, multirésolution et ondelettes)

Maxime Descoteaux

13 novembre 2020

Vous devez rédiger un *Latex* et me remettre un zip avec votre code MATLAB (n'utiliser pas de ToolBox Matlab. Vous pouvez tout faire avec les commandes de bases de Matlab et celles que je vous ai données).

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 latex et ses figures, et la qualité du code MATLAB (10 points).

1. **Approximations dans les bases de Fourier et d'ondelettes** sur une image de votre choix.

5 pts a) **Approximation de Fourier.** L'approximation linéaire garde que les M coefficients de plus basses fréquences dans l'espace de Fourier. Faites l'approximation linéaire dans Fourier gardant que M coefficients, pour 3 M différents et rapportez le SNR des images approximées ainsi que les erreurs relatives d'approximation sur les *plot*. (Vous avez déjà fait cela dans le TP3)

10 pts b) **Approximation avec cosinus discrets et des cosinus discrets locaux.** En premier temps, faites l'approximation linéaire avec une base de cosinus discrets (DCT) et une base de cosinus discrets locaux (locDCT), gardant que les M coefficients de plus basses fréquences, pour 3 M différents et rapportez le SNR des images approximées ainsi que les erreurs relatives d'approximation sur les *plot*. Utilisez les fonctions *dct2.m* et *idct2.m*, données en classe (vous devez étendre ce que j'ai déjà fait en 1D en 2D).

5 pts c) **Approximation d'ondelettes de Haar périodiques**. À partir de votre ondelette de Haar, faites l'approximation linéaire sur cette base en utilisant 3 M différents. Rapportez le SNR des images approximées et les erreurs relatives d'approximation sur les *plot*. Vous pouvez faire la version algébrique (comme en classe) ou celle avec les convolutions circulaires.

BONUS - 20 pts d) **Approximation d'ondelettes de Daubechies**. Utilisez une des ondelettes de Daubechies (e.g. [h, g] = compute_wavelet_filter('Daubechies', 4) pour l'ondelette de Daubechies de support 4). Faites l'approximation linéaire pour 3 M différents. Rapportez le SNR des images approximées et les erreurs relatives d'approximation. Pour celle-ci, vous n'aurez pas le choix que d'implémenter la version avec la convolution.

10 pts e) **Approximations nonlinéaires.** Une approximation nonlinéaire garde que les M coefficients les plus élevés de la décomposition dans une base. Faites les approximations nonlinéaires avec les bases des parties (a) à (d).

10 pts f) Faites un tableau des SNR et des erreurs d'approximations linéaires et nonlinéaires. Que remarquez-vous ?

10 pts g) Tracez la courbe des erreurs d'approximation en fonction de M. Pour un graphe optimal, mettez sur l'axe x, $\log_{10}(M/N)$ et sur l'axe y, $\log_{10}(|f - f_M|^2/|f|^2)$, ou N est le nombre de pixel dans l'image, M le nombre de coefficients, f l'image originale et f_M l'image approximée.
Utilisez plus de 3 M pour tracer cette courbe. Pensez à votre affaire ! Il y a moyen de faire ça très simplement et rapidement.
(indice : Conservation d'énergie)

20 pts h) Faites les approximations nonlinéaires pour les 4 images de la figure 1 (regular3, phantom, lena et le mandrill). Que remarquez-vous ? Les approximations dépendent-elles de la "complexité" de l'image ? Expliquez.

10 pts i) Selon vous, quelle est la meilleure base ? Pourquoi ?

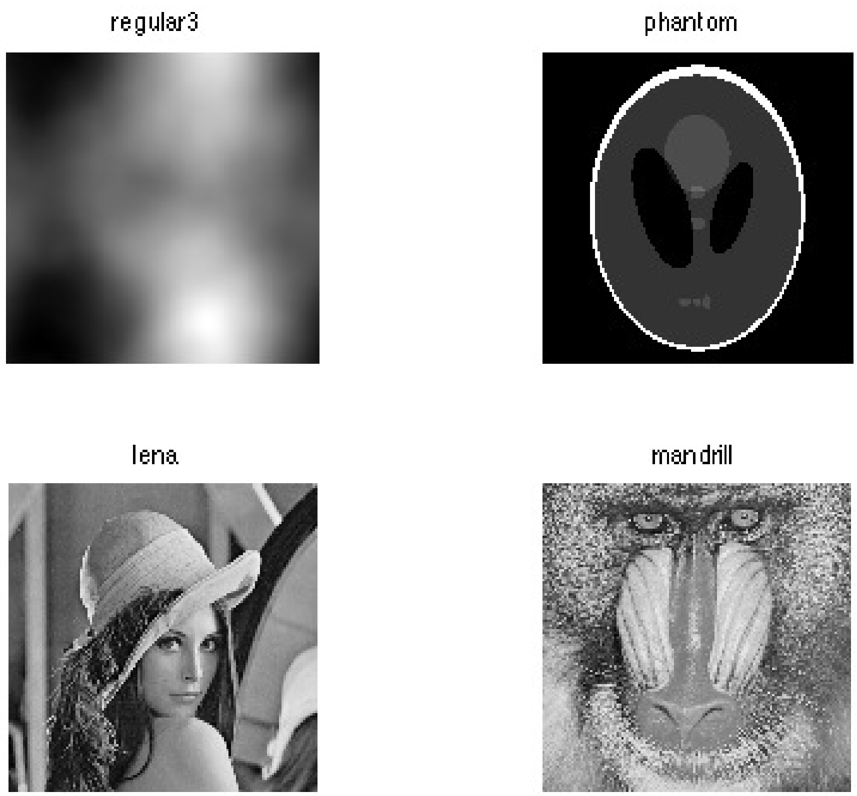

FIGURE 1 – 4 images classiques en traitement d'images à approximer nonlinéairement.