

FOURIER ET LES IMAGES

1. Objectif de ce travail :

Le travail proposé ici consiste à manipuler le spectre des images et à découvrir les relations entre les espaces direct et de Fourier.

2. Petite histoire : Jean-Baptiste que l'on appelait Joseph...

Né le 21 mars 1768 à Auxerre – mort le 16 mai 1830 à Paris

Élu académicien libre le 27 mai 1816. Le 29 mai, l'Académie est avisée que le roi Louis XVIII n'approuve pas cette élection. Une nouvelle élection le 12 mai 1817 (pour la section de physique générale) est confirmée par le roi le 23 mai 1817.

Élu Secrétaire perpétuel pour les sciences mathématiques le 18 novembre 1822. Il fut élu Membre de l'[Académie française](#) le 14 décembre 1826.

Mathématicien, Joseph Fourier est l'auteur de travaux fondamentaux sur la théorie de la chaleur. Il est l'un des initiateurs de la théorie mathématique des phénomènes physiques.



Joseph Fourier consacra ses premiers travaux à l'étude de théorèmes généraux relatifs à la résolution d'équations algébriques. Il présenta un mémoire sur ce sujet devant l'Académie des sciences le 9 décembre 1789.

En 1794, il est nommé élève à l'École normale, puis répétiteur à l'École polytechnique, poste qu'il occupa jusqu'en 1797. Recruté par Gaspard Monge, il partit en 1798 avec les scientifiques de l'expédition d'Égypte. Bonaparte créa l'Institut d'Égypte selon un plan inspiré de celui de Paris et le désigna pour en être le Secrétaire perpétuel. Joseph Fourier séjourna en Égypte jusqu'en 1801 où il mena des travaux scientifiques, historiques, administratifs et diplomatiques. Il fut chargé d'écrire la "Préface historique" de l'ouvrage "Description de l'Égypte" (1809) qui regroupe l'ensemble des observations faites au cours de l'expédition.

À son retour en France en 1802, il fut nommé préfet du département de l'Isère. Son nom reste attaché à des grands travaux (assèchement des marais de Bourgoin, ouverture de la route de Grenoble à Turin). Il trouva cependant suffisamment de temps pour effectuer en quelques années l'essentiel de son œuvre scientifique. Le 21 décembre 1807, il soumit à la première Classe de l'Institut national des sciences et des arts un premier mémoire intitulé "Théorie de la propagation de la chaleur dans les solides". La section de géométrie de l'Institut mit au concours la question "Donner la théorie mathématique des lois de la propagation de la chaleur, et comparer le résultat de cette théorie à des expériences exactes". Après une vive polémique comprenant notamment Jean-Baptiste Biot, Louis de Lagrange et Pierre-Simon de Laplace, l'Institut couronna les travaux de Fourier en 1812 par le grand Prix de Mathématiques de l'Institut.

Révoqué de la vie publique en mai 1815, élu Membre de l'Académie des sciences en 1817, puis Secrétaire perpétuel pour la division des mathématiques en 1822, il consacra l'essentiel de son temps à cette fonction tout en continuant à publier des travaux scientifiques ("Mémoire d'analyse indéterminée sur le calcul des conditions d'inégalités", "Sur de nouvelles expériences thermoélectriques", "Analyse des équations déterminées"...).

Il mourut le 16 mai 1830. Son éloge fut prononcé par François Arago.

Joseph Fourier était aussi Membre de la [Royal Society of London](#) (1823).

Référence Site web de l'académie des sciences :

http://www.academie-sciences.fr/MEMBRES/in_memoriam/Fourier/Fourier_oeuvre.htm

$$\text{Transformée Directe } FreqIma[m, n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} Ima[i, j] \cdot e^{-2j \cdot \pi \cdot \frac{(n \cdot i + l \cdot j)}{M \cdot N}}$$

$$\text{Transformée Inverse } Ima[i, j] = \frac{1}{M \cdot N} \cdot \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} FreqIma[m, n] \cdot e^{2j \cdot \pi \cdot \frac{(n \cdot i + l \cdot j)}{M \cdot N}}$$

Ima est l'image et *i, j* sont les coordonnées du *domaine spatial*.

FreqIma est l'image et *m, n* sont les coordonnées du *domaine fréquentiel*.

M, N sont les dimensions de l'image.

3 . MANIPULATIONS :

3.1 . Transformée de Fourier 1D :

Dans cette première partie, on vous propose de revoir rapidement comment est construite la Transformée de Fourier Discrète (FFT) sur un signal 2D :

Pour commencer, ouvrez l'image « porte.bmp » :

```
MaPorte=imread('porte.bmp');
```

Vous savez à présent que l'image est disponible sous la forme d'une matrice de valeurs codées sur 8 bits. Pour certaines opérations, il est important de noter que le résultat obtenu est plus précis, voire très différents, si les calculs sont réalisés sur des valeurs réelles. Pour convertir l'image en « flottant » :

```
MaPorte = double(MaPorte);
```

L'image en niveau de gris est formée de Nlin lignes, Ncol colonnes et 1 seul plan. Ces informations sont récupérables facilement grâce à :

```
[Nlin, Ncol] = size(MaPorte);
```

Pour afficher l'image :

```
figure(1); image(MaPorte);
```

matlab permet de calculer la FFT d'un signal 1D (un vecteur) grâce à la fonction :

```
Freq_MonSignal = fft(MaSignal);
```

Attention : votre image est un signal 2D réel et sa fft est complexe (la fonction abs() donne le module).

Essayez de regarder ce qu'il se passe si vous faites la fft sur la ligne Nlin/2 de votre image. Tracez ce spectre.

Effectuez la même opération sur la première ligne de l'image. Conclusion ?

Sachant que le spectre d'un signal échantillonné est périodique, effectuez les instructions suivantes pour rendre votre espace des fréquences périodique :

```
Freq_periodique=Freq_ligne;
for u=2:16
    Freq_periodique=[Freq_periodique Freq_ligne];
end
```

Tracez les spectres Freq_ligne et Freq_periodique.

Calculez la fft inverse (fonction ifft(MonSpectre)) du nouveau spectre. Conclusion ?

Sur les spectres tracés, vous observez que l'axe des abscisses représente l'indice du point

calculé et non la fréquence. Vous pouvez aussi vous rendre compte que le tracé représente les fréquences entre 0 et $F_{ech}/2$ sur les indices 0 à $N_{lin}/2$ et les fréquences entre $-F_{ech}/2$ et 0 pour les indices $N_{lin}/2$ à N_{lin} .

Changez l'échelle de l'axe des abscisses.

```
plot(abs(Freq_periodique));
v=axis; % récupère les valeurs des échelles sur x et y
axis([Nlin/2 3*Nlin/2 v(3) v(4)]); % modifie les valeurs sur
l'axe x uniquement !
```

Le spectre obtenue doit ressembler à quelque chose que vous connaissez.

Sous matlab, il existe une fonction qui permet d'effectuer ce décalage d'axe. Essayez :

```
figure(3);
FFT_ligne=fft(Im1(Nlin/2,:));
FFT_ligne=fftshift(FFT_ligne); % recentrage de l'origine des
fréquences au centre de la fenêtre
plot(abs(FFT_ligne)); % trace le module du spectre
```

3.2 . La Transformée de Fourier 2D :

Vous avez revue « vos classiques » sur la transformée 1D. Revenons aux images. Matlab permet de calculer la fft d'une image (commande `fft2`, regardez l'aide)

Affichez la transformée de Fourier 2D de 'MaPorte.bmp'.

Sachant que les valeurs du module obtenues peuvent être grandes, vous devrez ajuster les amplitudes du spectre aux valeurs de couleurs disponibles.

```
Image_affiche=FFT_MaPorte.... % mise à l'échelle des niveaux
de gris
image(Image_affiche);
colormap(pink(256)); % meilleure visibilité du spectre
axis('image');
```

Remarque : Vous pouvez utiliser la fonction `imagesc()` de matlab à la place de `image()` qui permet de « recadrer » automatiquement les valeurs dans l'intervalle des couleurs disponibles.

3.3 . Animez votre Transformée de Fourier 2D !

Vous maîtrisez parfaitement les affichages dans le domaine fréquentiel.

On se propose de réaliser une petite application didactique :

- a) Créez une image 512x512 contenant un cosinus aux fréquences $f_{ox}=1/T_{ox}$ suivant x et $f_{oy}=1/T_{oy}$ suivant y qui utilise tous les niveaux de gris disponibles. Affichez l'image créée.
- b) Calculez la fft de cette image et affichez en le module
- Faites une boucle qui reprend les étapes a) et b) en changeant la valeur de la fréquence à chaque passage dans la boucle (utilisez `imshow(Image); drawnow;` pour forcer l'affichage).

```
Tox=0; % définition de la période suivant x
Toy=0; % définition de la période suivant y
while(Tox<=1024)
    Tox=Tox+1;
```

```

    Toy=Toy+1;
    for i=1:128
        for j=1:128
            mon_cos(i,j)=double(128*(1+cos(2*pi*(i/Tox+j/Toy))));
        end
    end
    figure (1);
    imshow((mon_cos-min(min(mon_cos)))*1/(max(max(mon_cos))-min(min(mon_cos))));
    drawnow;
    colormap(pink(256));
    axis('image');

    figure (2);
    colormap(pink(256));
    axis('image');
    FFT_MonCos=fft2(mon_cos);
    FFT_MonCos=fftshift(FFT_MonCos);
    Image_affiche=abs(FFT_MonCos);
    imshow((Image_affiche-min(min(Image_affiche)))*256/(max(max(Image_affiche))-min(min(Image_affiche))));
    drawnow;
end

```

4 . COMPRESSION :

Vous allez maintenant effectuer une compression JPEG. Cette compression utilise la transformée en Cosinus Discret (DCT). Vous verrez que ce type de compression assume une perte d'information lors de l'opération.

4.1 . Principe :

Ce type de compression repose sur plusieurs étapes :

- découper l'image en blocs de taille identique (8x8 en pratique).
- calculer la DCT de chaque bloc.
- représenter le résultat de la DCT 2D en regroupant les pixels de tous les blocs par ordre de coefficient.
- supprimer les blocs de coefficients qui représentent les fréquences les plus élevées.
- calculer les DCT inverse de chaque bloc.
- regrouper les blocs dans une même image.

4.2 . Manipulation :

- Pour commencer, il est intéressant de regarder le résultat obtenu en effectuant la DCT de l'image complète. Utilisez l'image 'Clown.bmp'.

```

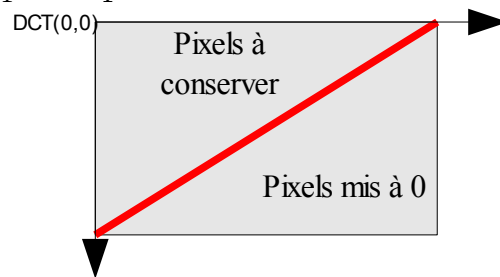
MonClown=imread('clown.bmp');
figure (1);
imagesc(DCT);
colormap(gray(256));
DCT=dct2(MonClown);

```

- Affichez le résultat de cette DCT :

```
% affichage du résultat
figure (2);
imagesc(DCT);
colormap(pink(256));
```

Remarquez que l'origine des fréquences se trouvent en haut à gauche de l'image.
 Ci-dessous, vous trouverez les codes qui vous permettent de réaliser ces différentes étapes.
 Mettre à zéro les pixels qui sont au-delà de la bissectrice rouge figure ci-dessous.



- Calculez la DCT inverse et afficher l'image obtenue (image 'Ma_DCT_inv').
- Estimez l'erreur induite par la mise à zéro des pixels en faisant la différence entre l'image originale et 'Ma_DCT_inv'.

Découpage des blocs & calcul DCT par bloc :

```
tbloc=8; % taille du bloc dans l'image (2,4,8,16,...)

%Découpage des blocs & calcul DCT par bloc
for i=1:round(Ncol/tbloc)
    for j=1:round(Nlin/tbloc)
        % Découpage des blocs
        for k=1:tbloc
            for l=1:tbloc
                %i,j,k,l
                bloc(k,l)=I((i-1)*tbloc+k,(j-1)*tbloc+l);
            end
        end
        % calcul DCT d'un bloc
        dct_image(i,j,,:) = dct2(bloc);
    end
end
```

Premier type d'affichage du résultat :

Dans ce mode de représentation de la DCT2d, on conserve la localisation des blocs de l'image originale : observez l'irrégularité spatiale de la répartition de l'information.

```
% reconstitution des blocs DCT en une image avec préservation
de la position des blocs de l'image originale
for i=1:round(Ncol/tbloc)
    for j=1:round(Nlin/tbloc)
        for k=1:tbloc
            for l=1:tbloc
                DCT1((i-1)*tbloc+k,(j-1)*tbloc+l) = dct_image(i,j,k,l);
            end
        end
    end
end
% affichage du résultat
figure (2);
```

```
imagesc(DCT1)
colormap(gray(256));
```

Deuxième type d'affichage du résultat :

Dans ce mode de représentation, on regroupe maintenant les pixels de tous les blocs par coefficient (tous les premiers pixels de chaque bloc sont regroupés dans un bloc, tous les second pixels sont regroupés dans un second bloc, etc.). Vous utiliserez ce mode d'affichage pour évaluer le niveau de dégradation acceptable de votre compression.

```
% mode de représentation de la DCT2d par regroupement des
% pixels de tous les blocs par coefficient.
for k=1:tbloc
    for l=1:tbloc
        for i=1:Ncol/tbloc
            for j=1:Nlin/tbloc
                DCT2((k-1)*Ncol/tbloc+i,(l-1)*Nlin/tbloc+j) = dct_image(i,j,k,l);
            end
        end
    end
end
% affichage du résultat
figure(3);
imagesc(DCT2)
colormap(gray(256));
```

Reconstruction de l'image : décompression

```
%Reconstruction par DCT inverse de l'image
for i=1:round(Ncol/tbloc)
    for j=1:round(Nlin/tbloc)
        bloc=idct2(masque.*reshape(dct_image(i,j, :, :),tbloc,t
        bloc));
        for k=1:tbloc
            for l=1:tbloc
                ImReconstitue((i-1)*tbloc+k,(j-1)*tbloc+l) = bloc(k,l);
            end
        end
    end
end
% affichage
figure(4);
imagesc(ImReconstitue)
colormap(gray(256));
```

Notez que dans cette dernière opération apparaît une variable nommée « masque ». Il s'agit d'une matrice de dimension (tbloc x tbloc) que vous pouvez remplir de 0 et de 1. Un 0 signifie que vous perdrez cette partie de la zone de fréquences, alors qu'un 1 signifie au contraire que vous souhaitez conserver ces fréquences. Essayez différentes solutions.

Rappel : les basses fréquences sont en haut et à gauche de votre image dans la représentation 2.