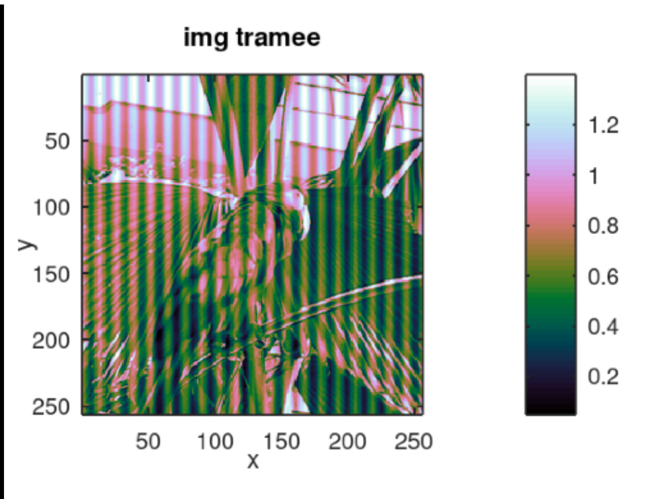
Exercice 14 : Détramage

Tramer (avec une haute fréquence), puis atténuer le tramage dans l'image de votre choix (ou a priori *bird.jpeg*) en filtrant dans le plan de Fourier.

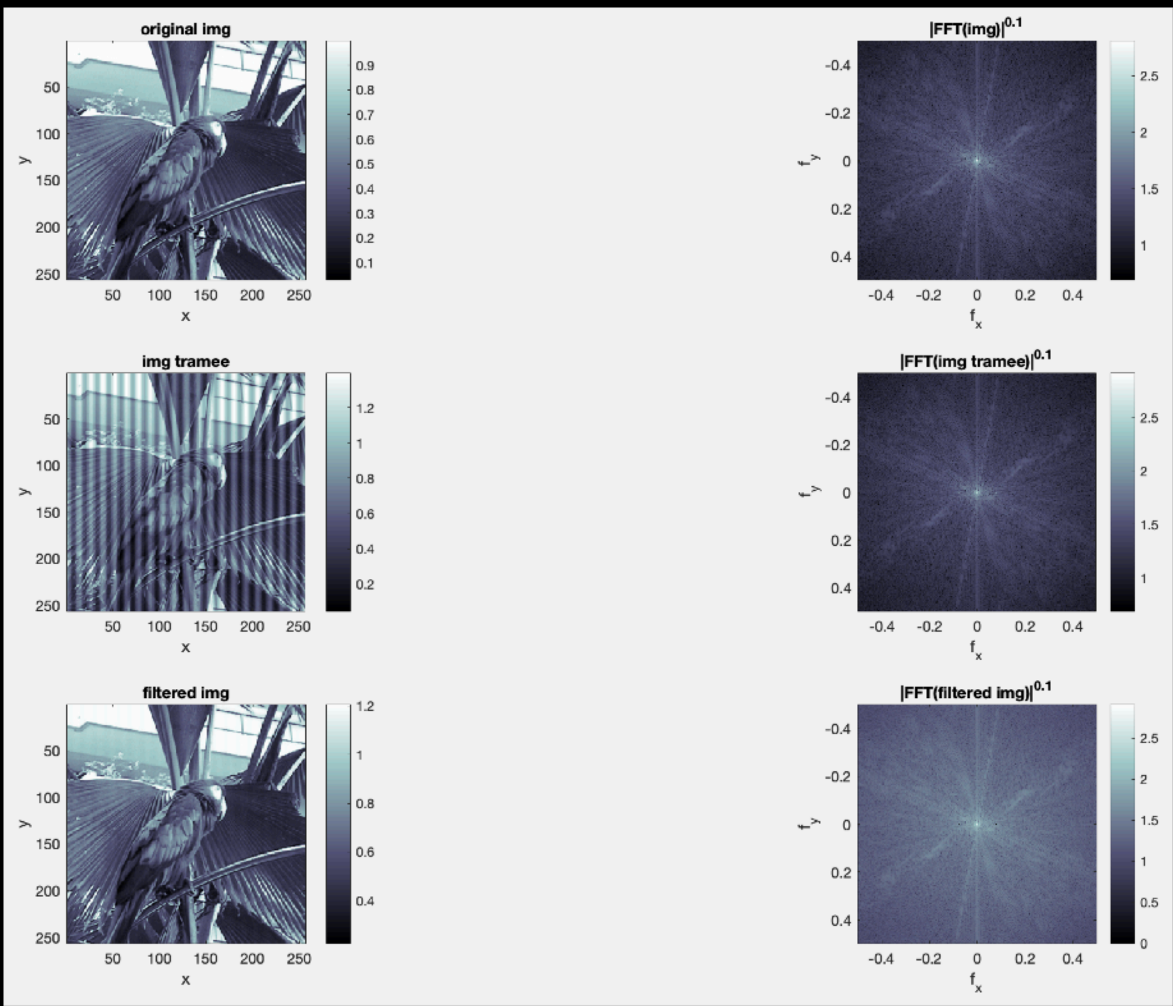
[Tramage avec une sinusoïde, tel que :
*tramage = (1 + cos(2πx/T))*coeff, T=12.8 px, coeff=0.2,*
puis : *image = image+tramage*]

Rmq : fréq.[en frequels] x $1/N\Delta x$ = fréq.[en px^{-1}]

```matlab
%%% (0) PRÉLIMINAIRES

%%% préliminaires
clear
close all
%pkg load image

%%% préparation image
img=imread('/Users/marcel/Documents/MATLAB/GBM/0-images/bird.jpg');
img=rgb2gray(img); img=double(img)/255.;
dim=size(img); dim=dim(1);
fx=((0:dim-1)-dim/2)/dim; fy=((0:dim-1)-dim/2)/dim;

%%% FFT(image) - pour comparaison
img_chap = fftshift(fft2(img)); % on se place dans un plan de Fourier
                                % cette fois-ci ré-ordonné !

%%% (1) TRAMAGE

Tx=12.8;                        % Tx=12.8 [px]
                                % => fc [en px^-1] = 1/Tx ≈ 0.08 px^(-1)
                                % => fc [en frequels] = 1/Tx*dim = 20 frequels
x=0:(dim-1); coeff=.2;
tram=ones(dim,1)*(1+cos(2*pi*x/Tx))*coeff;
imgt=img+tram;
imgt_chap = fftshift(fft2(imgt));
```



```matlab
%%% (2) DÉTRAMAGE

%%% filtrage coupe-fréquel (détramage)
imgt_chap_filt=imgt_chap;
fc = dim/Tx;
imgt_chap_filt(dim/2+1,dim/2+1+fc)=0.;
imgt_chap_filt(dim/2+1,dim/2+1-fc)=0.;
imgt_filt=real(ifft2(ifftshift(imgt_chap_filt)));
                                 % à cause du ré-ordonnage dans le plan de
                                 % Fourier, on est obligé d'utiliser
                                 % ifftshift avant de faire la FFT inverse

%%% résultat tramage/détramage
figure, colormap('bone')

subplot(3,2,1)
imagesc(img), colorbar, axis('square')
title('original img'), xlabel('x'), ylabel('y')

subplot(3,2,2)
imagesc(fx,fy,abs(img_chap).^.1), colorbar, axis('square')
title('|FFT(img)|^{0.1}'), xlabel('f_x'), ylabel('f_y')

subplot(3,2,3)
imagesc(imgt), colorbar, axis('square')
title('img tramee'), xlabel('x'), ylabel('y')

subplot(3,2,4)
imagesc(fx,fy,abs(imgt_chap).^.1), colorbar, axis('square')
title('|FFT(img tramee)|^{0.1}'), xlabel('f_x'), ylabel('f_y')

subplot(3,2,5)
imagesc(imgt_filt), colorbar, axis('square')
title('filtered img'), xlabel('x'), ylabel('y')

subplot(3,2,6)
imagesc(fx,fy,abs(imgt_chap_filt).^.1), colorbar, axis('square')
title('|FFT(filtered img)|^{0.1}'), xlabel('f_x'), ylabel('f_y')
```



4 - Reconstruction d’image

Il s’agit de « déflouter » (en fait *déconvoluer*) une image, rendue floue par ailleurs.

$$I_{\text{floue}} = I \otimes G$$

où $\otimes$ est l’opérateur de la *convolution* et G la fonction qui rend floue l’image I.

$$\Rightarrow \hat{I}_{\text{floue}} = \hat{I} \times \hat{G}$$

où $\times$ est ici une multiplication élément par élément.

Une idée (qui est naturelle mais pas forcément très bonne en présence de bruit, ne serait-ce que numérique) serait de poser :

$$\hat{I} = \hat{I}_{\text{floue}} \; / \; \hat{G}$$

Mais on doit déjà considérer le fait que l’on est en présence de nombres complexes, et donc on a :

$$\hat{I} = |\hat{I}| \; \exp\{i\phi_I\}$$

$$\hat{G} = |\hat{G}| \; \exp\{i\phi_G\}$$

$$\hat{I}_{\text{floue}} = |\hat{I}_{\text{floue}}| \; \exp\{i\phi_{I_{\text{floue}}}\}$$

d'où :

$$\hat{I} = \frac{|\hat{I}_{\text{floue}}|}{|\hat{G}|} \ \exp\left(\imath \ \left(\phi_{I_{\text{floue}}} - \phi_G\right)\right)$$

où la division des deux modules est a priori particulièrement délicate à cause des valeurs de $|\hat{G}|$ quand elles sont proches de zéro.

=> <u>solution</u> : appliquer un seuil à $|\hat{G}|$ afin d'éviter les points qui vont diverger vers l'infini à cause de la division…

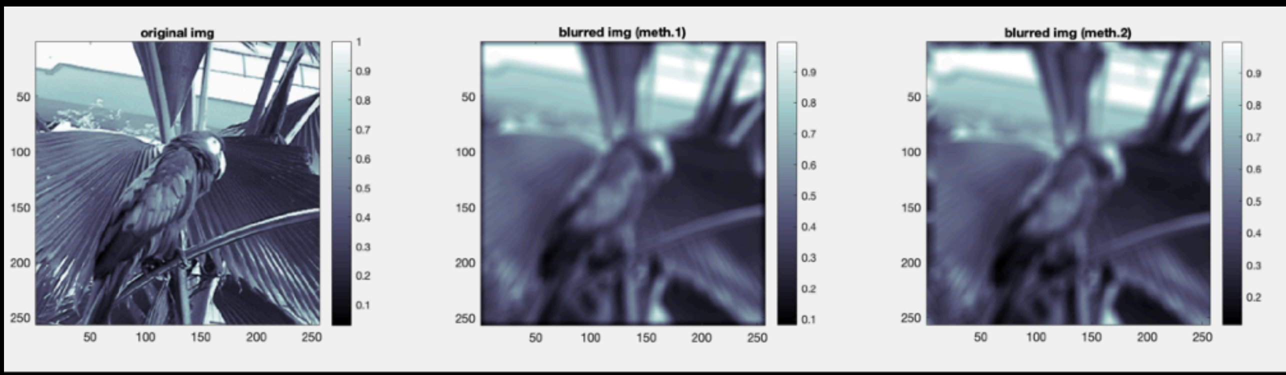
… et ce en appliquant :

- $|\hat{G}| \geq s$: $|\hat{G}|$ ok $\Rightarrow \hat{G}$ ok.
- $|\hat{G}| < s$: $|\hat{G}| = \text{seuil} \Rightarrow \hat{G} = \text{seuil} \times \exp\{\imath\phi_G\}$

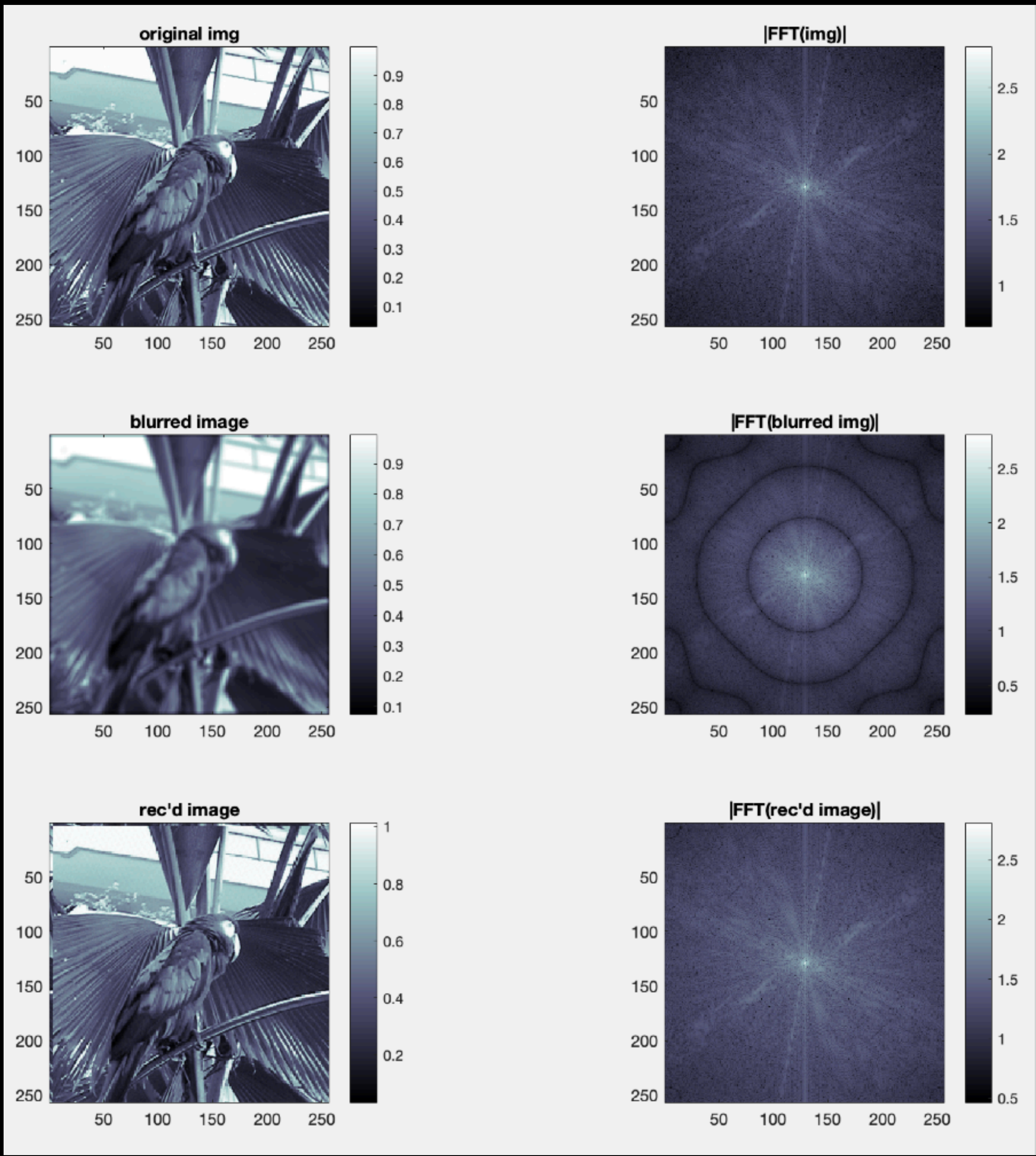
Il s'agit d'un ***<u>filtre inverse</u>***.

- **Exercice 15 :** Prendre l’image précédente, la flouter en la convoluant par un disque de rayon 3px, puis appliquer un *filtre inverse* afin d’opérer une tentative de reconstruction de l’image initiale. Jouer avec le seuil.

```
%%%%%%%%%%%%%%%%%%%%%%%%
%%% FILTRAGE INVERSE %%%
%%%%%%%%%%%%%%%%%%%%%%%%

%%% préliminaires
clear
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% PREMIÈRE PARTIE : SIMULATION %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% préparation image
img=imread('/Users/marcel/Documents/MATLAB/GBM/0-images/bird.jpg');
img=rgb2gray(img); img=double(img)/255.;
dim=size(img); dimx=dim(1); dimy=dim(2);
img_chap = fft2(img);   % pour comparaison

%%% floutage (convolution par un disque)
rr=3; gag=fspecial('disk',rr);
%%% convolution dans le plan direct
gau=zeros(dim);
gau(dimx/2+1-rr:dimx/2+1+rr, dimy/2+1-rr:dimy/2+1+rr)=gag;
imgg=convn(img,gau,'same');
%%% solution alternative (dans le plan de Fourier)
gag_chap=fft2(gag,dimx,dimy);
imgg_chap=gag_chap.*img_chap;
imgg2 = real(ifft2(imgg_chap));
%%% affichage floutage
figure(1), colormap('bone')
subplot(1,3,1), imagesc(img), colorbar
title('original img'), axis('image')
subplot(1,3,2), imagesc(imgg), colorbar
title('blurred img (meth.1)'), axis('image')
subplot(1,3,3), imagesc(imgg2), colorbar
title('blurred img (meth.2)'), axis('image')
```




```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% DEUXIÈME PARTIE : TRAITEMENT %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% défloutage par filtrage inverse
imgg_chap=fft2(imgg2);
gau_chap=fft2(gau);
seuil = .001;   % on peut jouer ici avec le seuil pour un résultat optimal
idx=find(abs(gau_chap) < seuil);
gau_chap(idx)=seuil*exp(complex(0,1)*angle(gau_chap(idx)));
img_rec_chap=imgg_chap./gau_chap;
img_rec=real(ifftshift(ifft2(img_rec_chap))); % real() nécessaire ici

%%% affichage filtrage inverse
figure(2), colormap('bone')
subplot(3,2,1), imagesc(img), colorbar
title('original img'), axis('image')
subplot(3,2,2), imagesc(abs(fftshift(img_chap)).^.1), colorbar
title('|FFT(img)|'), axis('image')
subplot(3,2,3), imagesc(imgg), colorbar
title('blurred image'), axis('image')
subplot(3,2,4), imagesc(abs(fftshift(imgg_chap)).^.1), colorbar
title('|FFT(blurred img)|'), axis('image')
subplot(3,2,5), imagesc(img_rec), colorbar
title("rec'd image"), axis('image')
subplot(3,2,6), imagesc(abs(fftshift(img_rec_chap)).^.1), colorbar
title("|FFT(rec'd image)|"), axis('image')
```



—> on remarque des artefacts aux bords…
=> <u>solution</u> : zero-padding !

- **<u>Exercice 16 :</u>** Reprendre l'exercice 15 en appliquant un pavage de zéro (zero-padding) dans des tableaux deux fois plus grands (linéairement, i.e. quatre fois plus grands en surface!).