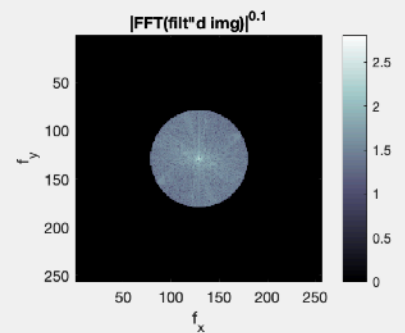
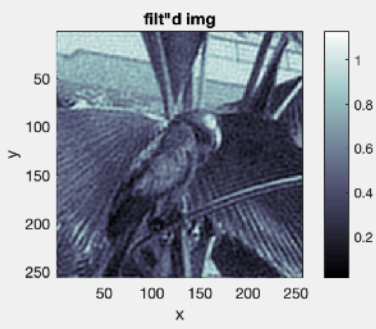
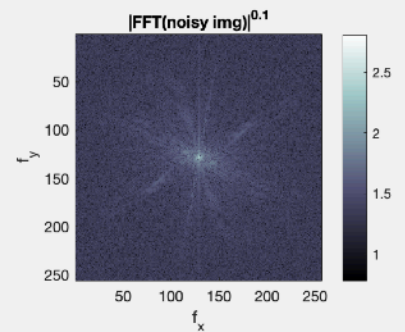
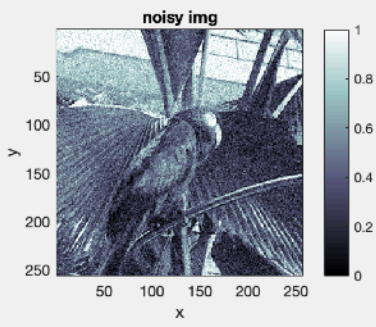
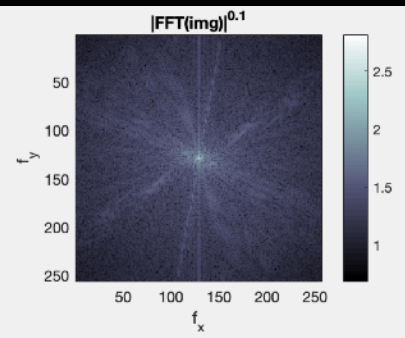
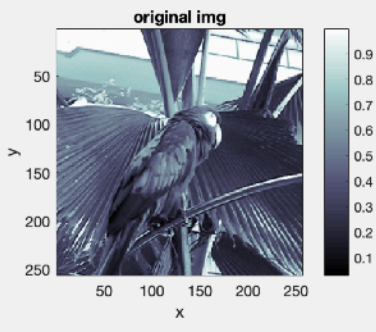


## 3 - Restoration d'images

- **Exercice 13** : Bruiter (bruit Gaussien additif de variance 1%) une image de votre choix (e.g. *bird.jpeg*), puis atténuer ce bruit à l'aide d'un filtre passe-bas adapté par vos soins. Quelle est la fréquence de coupure optimale pour ce filtre ?

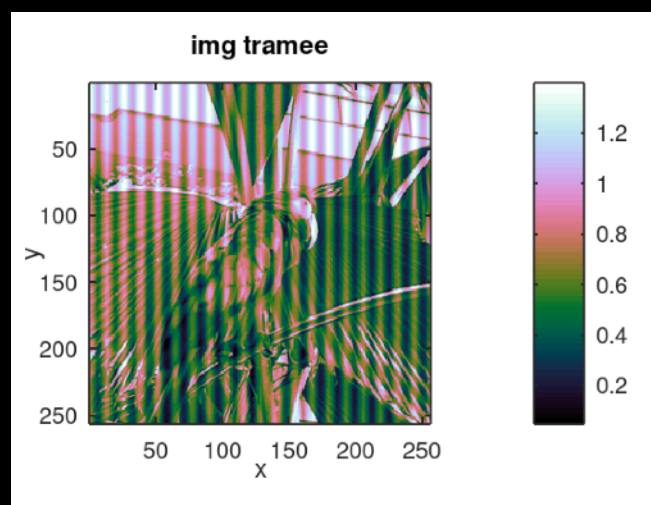
```
1      %%% préliminaires
2      clear
3      close all
4      %pkg load image
5
6      %%% préparation image
7      img=imread('./0-images/bird.jpg');
8      img=rgb2gray(img); img=double(img)/255.;
9      dim=size(img); dimx=dim(1); dimy=dim(2);
10     fx=((0:dimx-1)-dimx/2)/dimx;
11     fy=((0:dimy-1)-dimy/2)/dimy;
12
13     %%% FFT(image) - pour comparaison
14     img_chap = fft2(img);
15     img_chapmod = abs(fftshift(img_chap));
16
17     %%% figure
18     figure, colormap('bone')
19     |
20     subplot(3,2,1)
21     imagesc(img), colorbar, axis('image')
22     title('original img'), xlabel('x'), ylabel('y')
23
24     subplot(3,2,2)
25     imagesc(img_chapmod.^1), colorbar, axis('image')
26     title('|FFT(img)|^{0.1}'), xlabel('f_x'), ylabel('f_y')
27
28     %%% bruitage
29     varg = 0.01;
30     imgn = imnoise(img, 'gaussian', 0., varg);
31
32     %%% FFT(image bruitée)
33     imgn_chap=fft2(imgn);
34     imgn_chapmod=abs(fftshift(imgn_chap));
35
36     subplot(3,2,3)
37     imagesc(imgn), colorbar, axis('image')
38     title('noisy img'), xlabel('x'), ylabel('y')
39
40     subplot(3,2,4)
41     imagesc(imgn_chapmod.^1), colorbar, axis('image')
42     title('|FFT(noisy img)|^{0.1}'), xlabel('f_x'), ylabel('f_y')
43
44     %%% filtrage passe-bas (débruitage)
45     hchap=zeros(dim);
46     rr=50; hh=fspecial('disk',rr); hh=hh/max(max(hh));
47     hchap(dimx/2+1-rr:dimx/2+1+rr,dimy/2+1-rr:dimy/2+1+rr)=hh;
48
49     imgn_chap_filt=fftshift(hchap).*imgn_chap;
50     imgn_chap_filtmod=abs(fftshift(imgn_chap_filt));
51     imgn_filt=real(ifft2(imgn_chap_filt));
52
53     subplot(3,2,5)
54     imagesc(imgn_filt), colorbar, axis('image')
55     title('filt"d img'), xlabel('x'), ylabel('y')
56
57     subplot(3,2,6)
58     imagesc(imgn_chap_filtmod.^1), colorbar, axis('image')
59     title('|FFT(filt"d img)|^{0.1}'), xlabel('f_x'), ylabel('f_y')
```



- **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 un cosinus tel que :  
 $tramage = 1 + \cos(2\pi x/T) * coeff$ ,  $T=12.8 \text{ px}$ ,  $coeff=0.2$ ,  
 puis faire :  $image = image + tramage$ ]



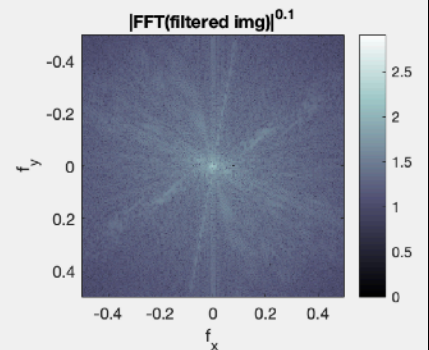
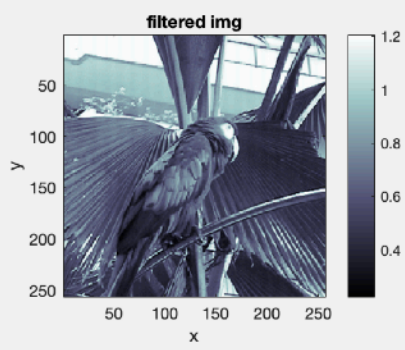
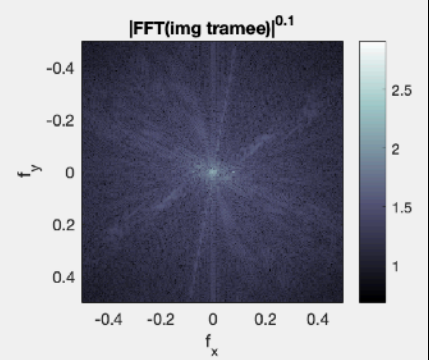
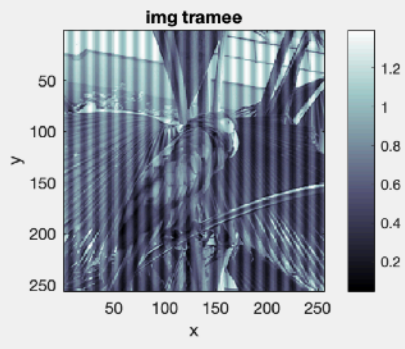
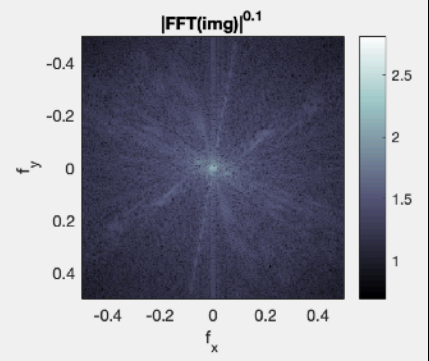
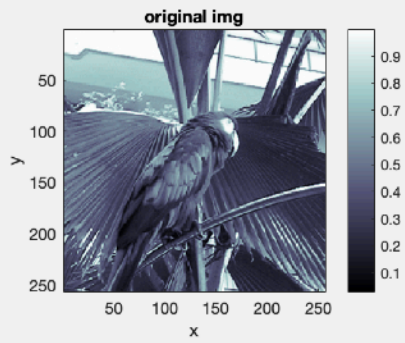
```

1  %%% (0) PRÉLIMINAIRES
2
3  %%% préliminaires
4  clear
5  close all
6  %pkg load image
7
8  %%% préparation image
9  img=imread('/Users/marcel/Documents/MATLAB/GBM/0-images/bird.jpg');
10 img=rgb2gray(img); img=double(img)/255.;
11 dim=size(img); dim=dim(1);
12 fx=((0:dim-1)-dim/2)/dim; fy=((0:dim-1)-dim/2)/dim;
13
14 %%% FFT(image) - pour comparaison
15 img_chap = fftshift(fft2(img)); % on se place dans un plan de Fourier
16 % cette fois-ci ré-ordonné !
17
18 %%% (1) TRAMAGE
19
20 Tx=12.8; % Tx=12.8 [px]
21 % => fc [en px^-1] = 1/Tx ≈ 0.08 px^-1
22 % => fc [en frequels] = 1/Tx*dim = 20 frequels
23 x=0:(dim-1); coeff=.2;
24 tram=ones(dim,1)*(1+cos(2*pi*x/Tx))*coeff;
25 imgt=img+tram;
26 imgt_chap = fftshift(fft2(imgt));
    
```

```

28     %%% (2) DÉTRAMAGE
29
30     %%% filtrage coupe-fréquel (détramage)
31     imgt_chap_filt=imgt_chap;
32     fc = dim/Tx;
33     imgt_chap_filt(dim/2+1,dim/2+1+fc)=0.;
34     imgt_chap_filt(dim/2+1,dim/2+1-fc)=0.;
35     imgt_filt=real(iff2(iffshift(imgt_chap_filt)));
36                                     % à cause du ré-ordonnage dans le plan de
37                                     % Fourier, on est obligé d'utiliser
38                                     % iffshift avant de faire la FFT inverse
39
40     %%% résultat tramage/détramage
41     figure, colormap('bone')
42
43     subplot(3,2,1)
44     imagesc(img), colorbar, axis('square')
45     title('original img'), xlabel('x'), ylabel('y')
46
47     subplot(3,2,2)
48     imagesc(fx,fy,abs(img_chap).^1), colorbar, axis('square')
49     title('|FFT(img) |^{0.1}'), xlabel('f_x'), ylabel('f_y')
50
51     subplot(3,2,3)
52     imagesc(imgt), colorbar, axis('square')
53     title('img tramee'), xlabel('x'), ylabel('y')
54
55     subplot(3,2,4)
56     imagesc(fx,fy,abs(imgt_chap).^1), colorbar, axis('square')
57     title('|FFT(img tramee) |^{0.1}'), xlabel('f_x'), ylabel('f_y')
58
59     subplot(3,2,5)
60     imagesc(imgt_filt), colorbar, axis('square')
61     title('filtered img'), xlabel('x'), ylabel('y')
62
63     subplot(3,2,6)
64     imagesc(fx,fy,abs(imgt_chap_filt).^1), colorbar, axis('square')
65     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 :

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

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

=> solution : 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 :

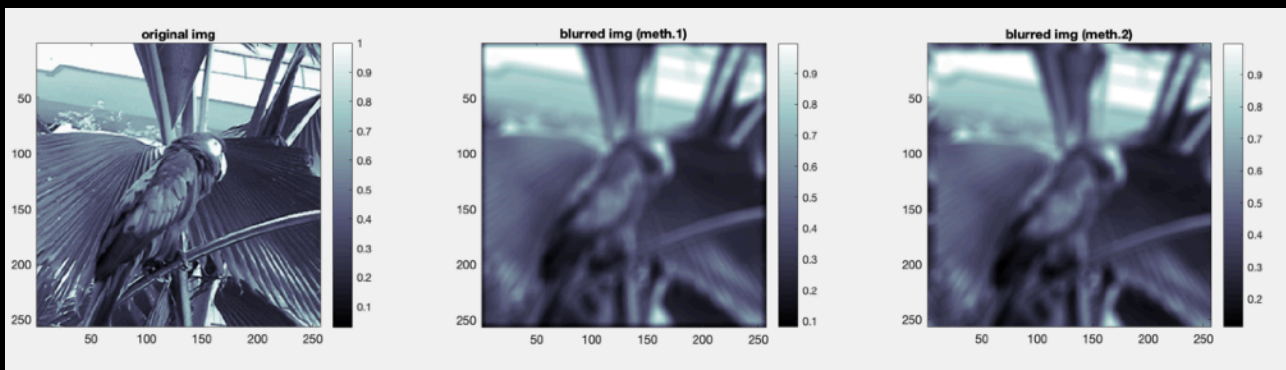
- $|\hat{G}| \geq s$  :  $|\hat{G}| \text{ ok} \Rightarrow \hat{G} \text{ ok.}$
- $|\hat{G}| < s$  :  $|\hat{G}| = \text{seuil} \Rightarrow \hat{G} = \text{seuil} \times \exp\{\iota\phi_G\}$

Il s'agit d'un **filtre inverse**.

- **Exercice 15** : Prendre l'image précédente, la flouter en la convoluant par un disque de rayon 3, puis 6, puis 12px, puis appliquer un *filtre inverse* afin d'opérer une tentative de reconstruction de l'image initiale. Jouer avec le seuil.

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% FILTRAGE INVERSE %%
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5  %% préliminaires
6  clear
7  close all
8
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 %% PREMIÈRE PARTIE : SIMULATION %%
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12
13 %% préparation image
14 img=imread('/Users/marcel/Documents/MATLAB/GBM/0-images/bird.jpg');
15 img=rgb2gray(img); img=double(img)/255.;
16 dim=size(img); dimx=dim(1); dimy=dim(2);
17 img_chap = fft2(img); % pour comparaison
18
19 %% floutage (convolution par un disque)
20 rr=3; gag=fspecial('disk',rr);
21 %% convolution dans le plan direct
22 gau=zeros(dim);
23 gau(dimx/2+1-rr:dimx/2+1+rr, dimy/2+1-rr:dimy/2+1+rr)=gag;
24 imgg=convn(img,gau,'same');
25 %% solution alternative (dans le plan de Fourier)
26 gag_chap=fft2(gag,dimx,dimy);
27 imgg_chap=gag_chap.*img_chap;
28 imgg2 = real(ifft2(imgg_chap));
29 %% affichage floutage
30 figure(1), colormap('bone')
31 subplot(1,3,1), imagesc(img), colorbar
32 title('original img'), axis('image')
33 subplot(1,3,2), imagesc(imgg), colorbar
34 title('blurred img (meth.1)'), axis('image')
35 subplot(1,3,3), imagesc(imgg2), colorbar
36 title('blurred img (meth.2)'), axis('image')
```

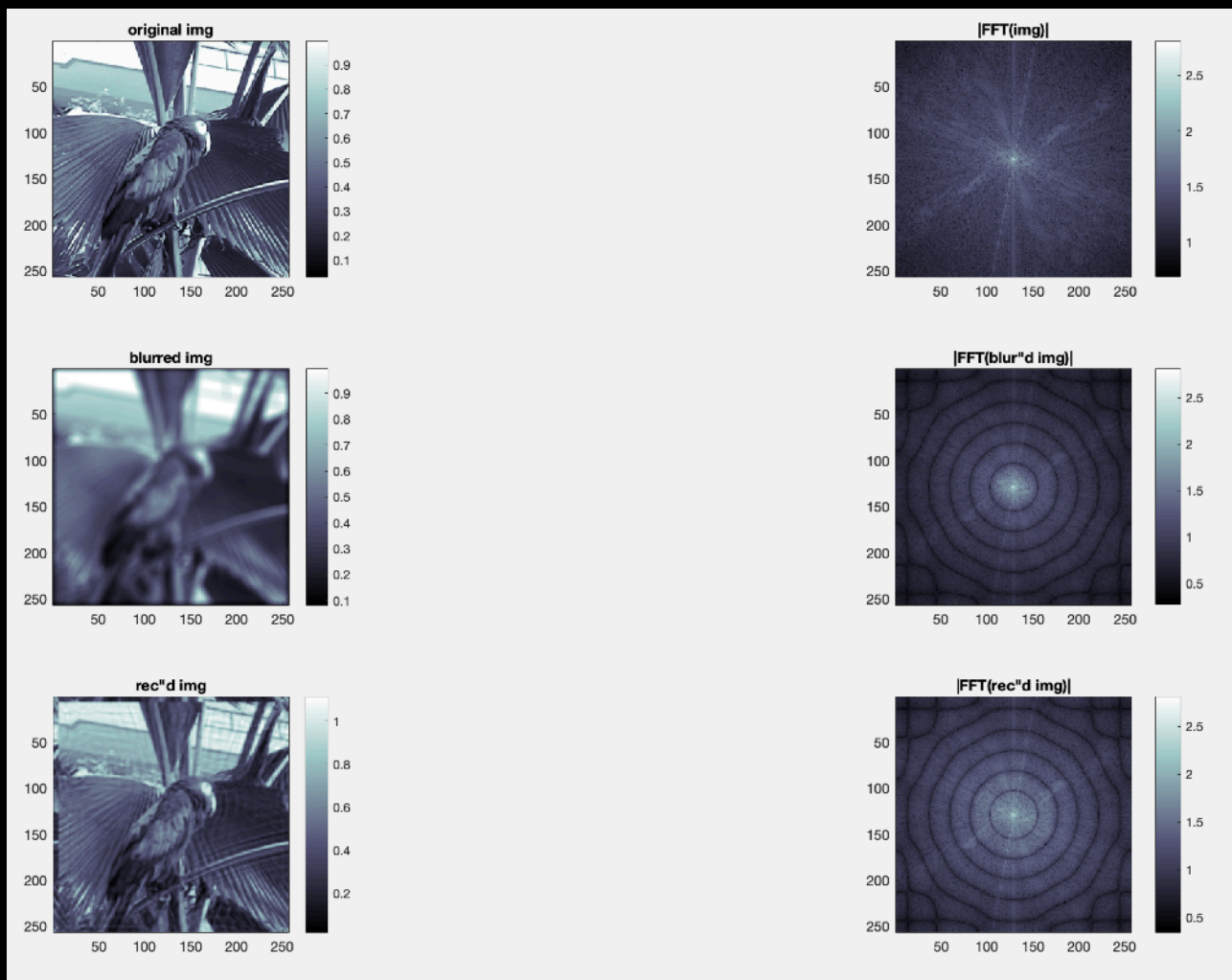




```

38 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39 %%% DEUXIÈME PARTIE : TRAITEMENT %%%
40 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
41
42 %%% défloutage par filtrage inverse
43 imgg_chap=fft2(imgg2);
44 gau_chap=fft2(gau);
45 seuil = .01; % on peut jouer ici avec le seuil pour un résultat optimal
46 idx=find(abs(gau_chap) < seuil);
47 gau_chap(idx)=seuil*exp(complex(0,1)*angle(gau_chap(idx)));
48 img_rec_chap=imgg_chap./gau_chap;
49 img_rec=real(ifftshift(ifft2(img_rec_chap)));
50
51 %%% affichage filtrage inverse
52 figure(2), colormap('bone')
53 subplot(3,2,1), imagesc(img), colorbar
54 title('original img'), axis('image')
55 subplot(3,2,2), imagesc(abs(fftshift(img_chap)).^1), colorbar
56 title('|FFT(img)|'), axis('image')
57 subplot(3,2,3), imagesc(imgg), colorbar
58 title('blurred image'), axis('image')
59 subplot(3,2,4), imagesc(abs(fftshift(imgg_chap)).^1), colorbar
60 title('|FFT(blurred img)|'), axis('image')
61 subplot(3,2,5), imagesc(img_rec), colorbar
62 title("rec'd image"), axis('image')
63 subplot(3,2,6), imagesc(abs(fftshift(img_rec_chap)).^1), colorbar
64 title("|FFT(rec'd image)|"), axis('image')

```



—> on remarque des artefacts dus aux bords...  
=> solution : zero-padding !

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

```

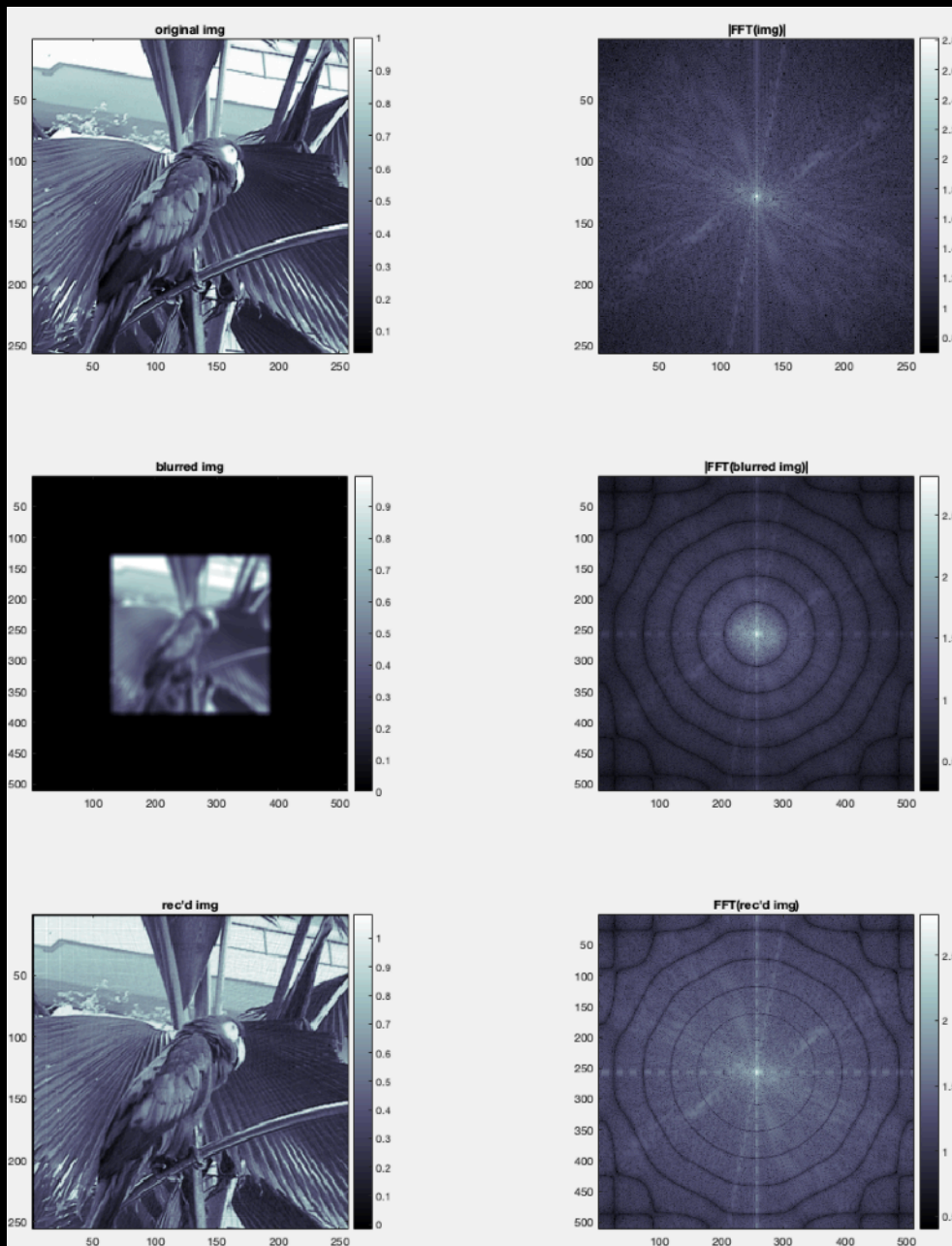
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% FILTRAGE INVERSE AVEC ZERRO_PADDING %%
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5  %% préliminaires
6  clear
7  close all
8
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 %% PREMIÈRE PARTIE : SIMULATION %%
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12
13 %% préparation image
14 img=imread('/Users/marcel/Documents/MATLAB/GBM/0-images/bird.jpg');
15 img=rgb2gray(img); img=double(img)/255.;
16 dim=size(img); dimx=dim(1); dimy=dim(2);
17 img_chap = fft2(img,2*dimx,2*dimy); % FFT (+zero-padding au passage)
18
19 %% floutage (convolution par un disque) (+zero-padding au passage)
20 rr=6; gag=fspecial('disk',rr);
21 gau=zeros(dim); gau(dimx/2+1-rr:dimx/2+1+rr,dimy/2+1-rr:dimy/2+1+rr)=gag;
22 dummy=conv2(img,gau,'full'); imgg=zeros(2*dimx,2*dimy); imgg(2:end,2:end)=dummy;

```

```

24 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25 %% DEUXIÈME PARTIE : TRAITEMENT %%
26 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27
28 %% défloutage par filtrage inverse
29 gau_chap=fft2(gau,2*dimx,2*dimy);
30 imgg_chap=fft2(imgg);
31 seuil = .01; % on peut jouer ici avec le seuil pour un résultat optimal
32 idx=find(abs(gau_chap) < seuil);
33 gau_chap(idx)=seuil*exp(complex(0,1)*angle(gau_chap(idx)));
34 img_rec_chap=imgg_chap./gau_chap;
35 img_rec=real(iff2(img_rec_chap));
36 img_rec=img_rec(1:dimx,1:dimy); % pas besoin de réordonner les quadrants
37 % ici, on prend directement le bon
38 %% affichage filtrage inverse avec zero padding
39 figure(), colormap('bone')
40 subplot(3,2,1), imagesc(img), colorbar
41 title('original img'), axis('image')
42 subplot(3,2,2), imagesc(abs(fftshift(fft2(img))).^1), colorbar
43 title('|FFT(img)|'), axis('image')
44 subplot(3,2,3), imagesc(imgg), colorbar
45 title('blurred img'), axis('image')
46 subplot(3,2,4), imagesc(abs(fftshift(imgg_chap)).^1), colorbar
47 title('|FFT(blurred img)|'), axis('image')
48 subplot(3,2,5), imagesc(img_rec), colorbar
49 title("rec'd img"), axis('image')
50 subplot(3,2,6), imagesc(abs(fftshift(img_rec_chap)).^1), colorbar
51 title("FFT(rec'd img)"), axis('square')

```



- Préférer à partir de maintenant la méthode de convolution en passant par Fourier pour le floutage.

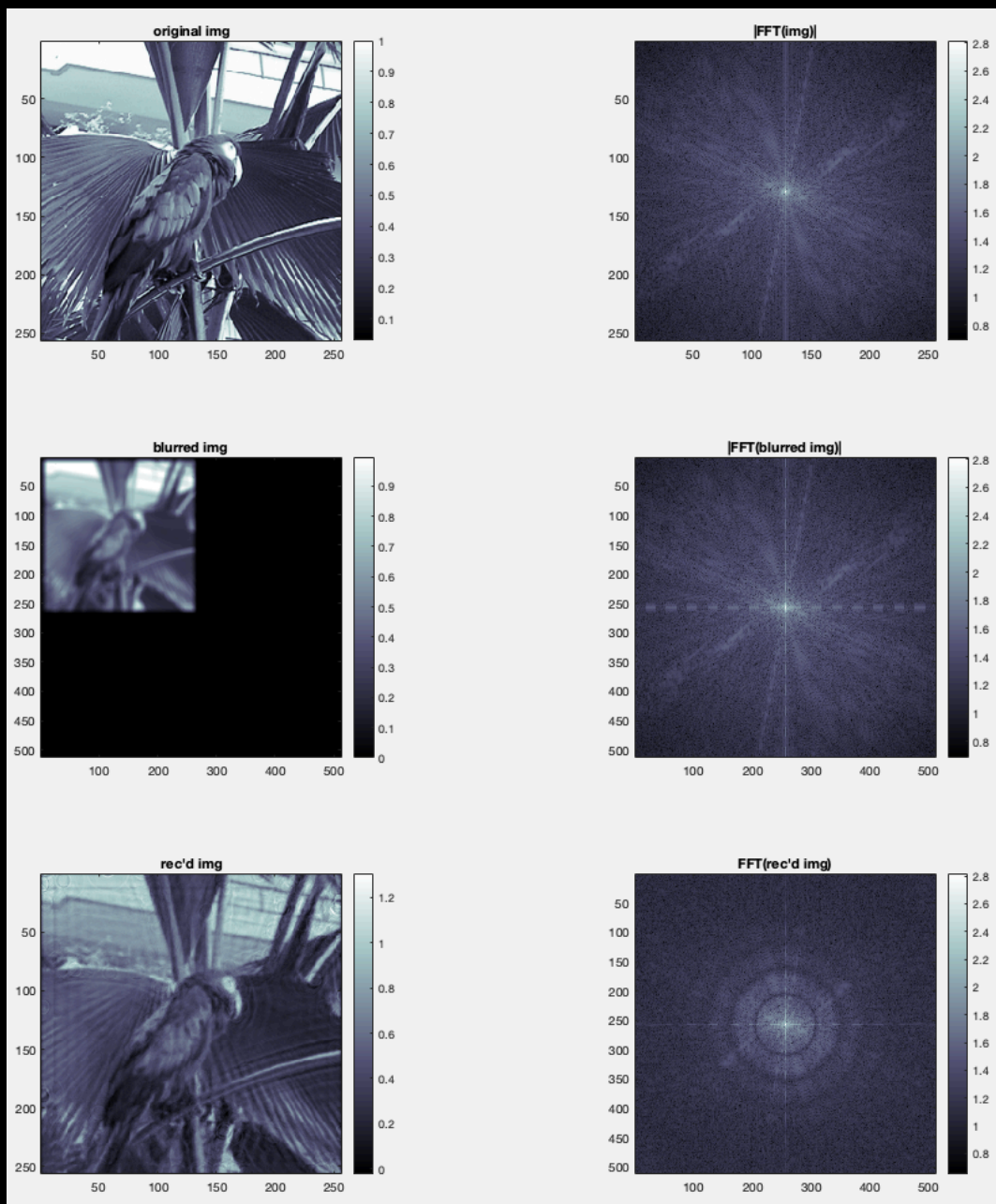
- **Exercice 17** : Reprendre l'exercice en lui appliquant un bruit poivre et sel affectant 2% des pixels. Après avoir tenter une reconstruction de l'image floutée par inversion, penser à tenter une réduction du bruit poivre et sel avant, puis jouer avec le seuil pour trouver un équilibre entre artefacts et propagation du bruit.

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%% FILTRAGE INVERSE EN PRÉSENCE DE BRUIT P&S %%%
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5  %%% préliminaires
6  clear
7  close all
8
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 %%% PREMIÈRE PARTIE : SIMULATION %%%
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12
13 %%% préparation image
14 img=imread('/Users/marcel/Documents/MATLAB/GBM/0-images/bird.jpg');
15 img=rgb2gray(img); img=double(img)/255.;
16 dim=size(img); dimx=dim(1); dimy=dim(2);
17 img_chap = fft2(img,2*dimx,2*dimy); % FFT (+zero-padding au passage)
18
19 %%% floutage (convolution par un disque) avec zero-padding
20 %%% solution alternative (dans le plan de Fourier)
21 rr=6; gag=fspecial('disk',rr); gag_chap=fft2(gag,2*dimx,2*dimy);
22 imgg_chap=gag_chap.*img_chap; imgg = real(ifft2(imgg_chap));
23
24 %%% bruitage poivre et sel à 2%
25 imgg=imnoise(imgg, 'salt & pepper',0.02);
```

```

27 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28 %% DEUXIÈME PARTIE : TRAITEMENT %%
29 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
30
31 %% filtre médian pour contrer le bruit p&s
32 imgg=medfilt2(imgg);
33
34 %% défloutage par filtrage inverse
35 gau_chap=gag_chap;
36 imgg_chap=fft2(imgg);
37 seuil = .1; % on a ici monté le seuil (à optimiser!)
38 idx=find(abs(gau_chap) < seuil);
39 gau_chap(idx)=seuil*exp(complex(0,1)*angle(gau_chap(idx)));
40 img_rec_chap=imgg_chap./gau_chap;
41 img_rec=real(ifft2(img_rec_chap));
42 img_rec=img_rec(1:dimx,1:dimy);
43
44 %% affichage filtrage inverse avec zero padding en présence de bruit p&s
45 figure(), colormap('bone')
46 subplot(3,2,1), imagesc(img), colorbar
47 title('original img'), axis('image')
48 subplot(3,2,2), imagesc(abs(fftshift(fft2(img))).^1), colorbar
49 title('|FFT(img)|'), axis('image')
50 subplot(3,2,3), imagesc(imgg), colorbar
51 title('blurred img'), axis('image')
52 subplot(3,2,4), imagesc(abs(fftshift(img_chap)).^1), colorbar
53 title('|FFT(blurred img)|'), axis('image')
54 subplot(3,2,5), imagesc(img_rec), colorbar
55 title("rec'd img"), axis('image')
56 subplot(3,2,6), imagesc(abs(fftshift(img_rec_chap)).^1), colorbar
57 title("FFT(rec'd img)"), axis('image')

```



- **Exercice 18 :** Reprendre l'exercice en lui appliquant toujours un bruit poivre et sel (2% des pixels), et de surcroît un bruit Gaussien d'écart type relatif 0.05. En plus du filtre non-linéaire permettant de réduire au maximum le bruit poivre et sel, jouer avec le seuil pour trouver un nouvel équilibre entre artefacts et propagation des bruits.

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%% FILTRAGE INVERSE AVEC P&S ET RON %%%
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5  %%% préliminaires
6  clear
7  close all
8
9  %%% préparation image
10 img=imread('/Users/marcel/Documents/MATLAB/GBM/0-images/bird.jpg');
11 img=rgb2gray(img); img=double(img)/255.;
12 dim=size(img); dimx=dim(1); dimy=dim(2);
13 img_chap = fft2(img,2*dimx,2*dimy); % FFT (+zero-padding au passage)
14
15 %%% floutage (convolution par un disque) (+zero-padding au passage)
16 rr=6; gag=fspecial('disk',rr); gag_chap=fft2(gag,2*dimx,2*dimy);
17 imgg_chap=gag_chap.*img_chap; imgg = real(ifft2(imgg_chap));
18
19 %%% bruitage poivre et sel à 2%
20 imgg=imnoise(imgg, 'salt & pepper', 0.02);
21
22 %%% bruitage gaussien additif (sigma=0.05)
23 imgg=imnoise(imgg, 'gaussian', 0., 0.05^2);
24
25 %%% filtre médian pour contrer le bruit p&s
26 imgg_mf=medfilt2(imgg);

```

```

28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 %%% DEUXIÈME PARTIE : TRAITEMENT %%%
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31
32 %%% ajouter éventuellement une étape de réduction du bruit gaussien...
33
34 % défloutage par filtrage inverse
35 imgg_mf_chap=fft2(imgg_mf);
36 gau_chap=gag_chap;
37 seuil = .5; % on a ici jouer avec le seuil (à optimiser!)
38 idx=find(abs(gau_chap) < seuil);
39 gau_chap(idx)=seuil*exp(complex(0,1)*angle(gau_chap(idx)));
40 img_rec_chap=imgg_mf_chap./gau_chap;
41 img_rec=real(ifft2(img_rec_chap)); img_rec=img_rec(1:dimx,1:dimy);
42
43 %%% affichage filtrage inverse avec zero padding
44 figure(), colormap('bone')
45 subplot(3,2,1), imagesc(img), colorbar
46 title('original img'), axis('image')
47 subplot(3,2,2), imagesc(abs(fftshift(fft2(img))).^1), colorbar
48 title('|FFT(img)|'), axis('image')
49 subplot(3,2,3), imagesc(imgg), colorbar
50 title('blurry+noisy img'), axis('image')
51 subplot(3,2,4), imagesc(abs(fftshift(imgg_chap)).^1), colorbar
52 title('|FFT(blurred img)|'), axis('image')
53 subplot(3,2,5), imagesc(img_rec), colorbar
54 title('restaured img'), axis('image')
55 subplot(3,2,6), imagesc(abs(fftshift(img_rec_chap)).^1), colorbar
56 title('FFT(rec"d img)'), axis('image')

```



