

Exercice 4 : Décaler la Gaussienne bidimensionnelle de l'exercice précédent (avec $\sigma=10$ px) de 10,4 px en x et -10,4 px en y, par TF.

Étapes :

- (1) création de la Gaussienne (plan direct)
- (2) calculer le terme de phase (plan de Fourier)
- (3) appliquer le terme de phase (plan de Fourier)
- (4) revenir dans le plan direct

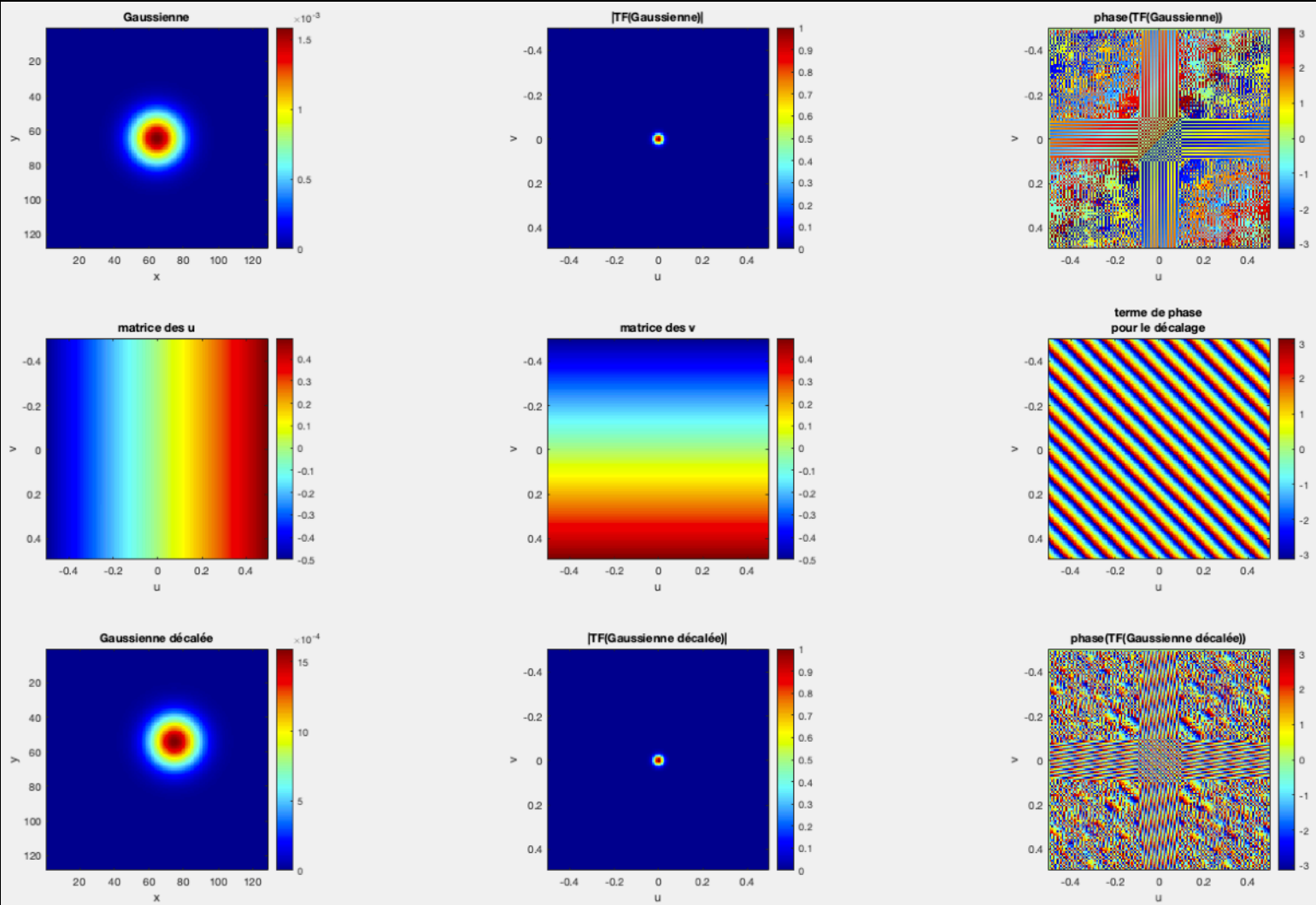
L'étape (2) revient à écrire correctement le terme de phase $\exp(-2 i \pi (x_0 u + y_0 v))$ dans Fourier... terme qui doit être un tableau de mêmes dimensions que l'image de départ (et donc sa TF et donc la phase de cette TF). Ici x_0 et y_0 sont les décalages respectivement en x et en y, et ce sont des nombres, pas des tableaux. Par contre, u et v doivent décrire toutes les valeurs des fréquences dans le plan de Fourier... Ce sont donc des tableaux de mêmes dimensions que l'image de départ. Par exemple : $u=\text{ones}(\text{dim}, 1)*fx$, où : $fx=1/(\text{dim}*1)*(-\text{dim}/2:\text{dim}/2-1)$.

Attention : on est ici dans le plan de la **FFT**, avec son décalage d'un demi-tableau à droite et vers le haut, il faut donc utiliser **fftshift** de manière à avoir les fréquences des tableaux de u et de v en face de celles de la FFT de l'image.

```

1 % préliminaires
2 clear
3 close all
4 %pkg load image
5
6 dim=128; % taille des images
7 fx=1/(dim*1)*(-dim/2:dim/2-1); fy=fx; % vecteur des fréquences :
8 % 1/(N Δx)*(-N/2 .. 0 .. N/2-1)
9 % on peut aussi écrire :
10 % fx=((0:dim-1)-dim/2)/(dim*1);
11 % Gaussienne
12 a=10;
13 G=fspecial('gaussian',dim,a); % Gaussienne
14 Gchap = fft2(G); % TF(Gaussienne)
15
16 % terme de phase
17 x0= 10.4; % décalage en x
18 y0=-10.4; % décalage en y
19 uu=ones(dim,1)*fx; % matrice des u (par colonne)
20 vv=uu'; % matrice des v (par ligne)
21 %vv=fy*ones(1,dim); % matrice des v (par ligne)
22 u=fftshift(uu); % car le plan de la FFT est "shifté"
23 v=fftshift(vv); % d'un demi-tableau en u et en v
24 decfou=exp(-complex(0,1)*2*pi*(u*x0+v*y0)); % TF(décalage) ; complex(0,1)=i
25 % décalage
26 Gdecchap = Gchap.*decfou; % TF(Gaussienne décalée)
27 Gdec=real(ifft2(Gdecchap)); % retour dans le plan direct
28 % (real pas forcément nécessaire)
29
30 % figure finale
31 figure(1), colormap('jet') % colormap "jet" (voir help colormap)
32
33 subplot(3,3,1), imagesc(G)
34 colorbar, axis('square')
35 title('Gaussienne'), xlabel('x'), ylabel('y')
36
37 subplot(3,3,2), imagesc(fx, fy, abs(fftshift(Gchap)))
38 colorbar, axis('square')
39 title('|TF(Gaussienne)|'), xlabel('u'), ylabel('v')
40
41 subplot(3,3,3), imagesc(fx, fy, angle(fftshift(Gchap)))
42 colorbar, axis('square')
43 title('phase(TF(Gaussienne))'), xlabel('u'), ylabel('v')
44
45 subplot(3,3,4), imagesc(fx, fy, uu)
46 colorbar, axis('square')
47 title('matrice des u'), xlabel('u'), ylabel('v')
48
49 subplot(3,3,5), imagesc(fx, fy, vv)
50 colorbar, axis('square')
51 title('matrice des v'), xlabel('u'), ylabel('v')
52
53 subplot(3,3,6), imagesc(fx, fy, angle(fftshift(decfou)))
54 colorbar, axis('square')
55 title({'terme de phase'; 'pour le décalage'}), xlabel('u'), ylabel('v')
56
57 subplot(3,3,7), imagesc(Gdec)
58 colorbar, axis('square')
59 title('Gaussienne décalée'), xlabel('x'), ylabel('y')
60
61 subplot(3,3,8), imagesc(fx, fy, abs(fftshift(Gdecchap)))
62 colorbar, axis('square')
63 title('|TF(Gaussienne décalée)|'), xlabel('u'), ylabel('v')
64
65 subplot(3,3,9), imagesc(fx, fy, angle(fftshift(Gdecchap)))
66 colorbar, axis('square')
67 title('phase(TF(Gaussienne décalée))'), xlabel('u'), ylabel('v')

```



(2) Filtrage dans le plan (complexe) de Fourier

- On utilise le théorème de convolution de la TF :

$$f * h \xrightarrow{TF} \hat{f} \bullet \hat{h}$$

$$f \bullet h \xrightarrow{TF} \hat{f} * \hat{h}$$

avec f = image, h = filtre $\Rightarrow \hat{h}$ = fonction de transfert de h

$h = \text{filtre}$ (et $\hat{h}$ sert de masque dans le plan de Fourier)

L'idée ici est de modeler $\hat{h}$ (à travers son module) afin de modifier la distribution des fréquences spatiales présentes dans l'image f .

- Filtrage passe-bas

On fait diminuer les valeurs de $\hat{h}$ au fur et à mesure que l'on s'éloigne du centre $(0,0)$ des fréquences spatiales, favorisant ainsi les plus basses fréquences en atténuant les plus hautes.

Cette distribution peut-être brutale (on « coupe » alors les fréquences plus hautes qu'une certaine fréquence **de coupure** à l'aide d'une porte de largeur $(2 u_c, 2 v_c)$ autour du frequel $(0,0)$, où : u_c est la fréquence de coupure en u , v_c celle en v .

Applications : supprimer les détails fins, réduire le bruit.

Remarque : on prend en général des fonctions de transfert $\hat{h}$ autour du frequel $(0,0)$.

Exemple : revisite du filtrage Gaussien (où l'on cherche à faire le filtrage Gaussien du chapitre 3 mais en passant par Fourier...)

```
>> dim=128;
>> I=zeros(dim,dim);
>> I(:,1:dim/2) = 0.6;
>> figure, colormap(gray)
>> subplot(3,2,1), imagesc(I), colorbar, title('img')
—
>> Ichap = fft2(I);
>> whos Ichap;
>> subplot(3,2,2), imagesc(abs(fftshift(Ichap)).^0.5),
colorbar, title('|FFT(image)|')
—
>> h = fspecial('gaussian', dim, 0.849);
>> subplot(3,2,3), imagesc(h.^0.5), colorbar,
title('Gauss')
—
>> hchap = fft2(h);
>> whos hchap;
>> subplot(3,2,4), imagesc(abs(fftshift(hchap)).^0.5),
colorbar, title('|FFT(Gauss)|')
—
```

```
>> lchapfilt = hchap.*lchap;  
>> whos lchapfilt  
>> subplot(3,2,6),  
imagesc(abs(fftshift(lchapfilt)).^0.5), colorbar, title(''  
FFT(image filtrée)''
```

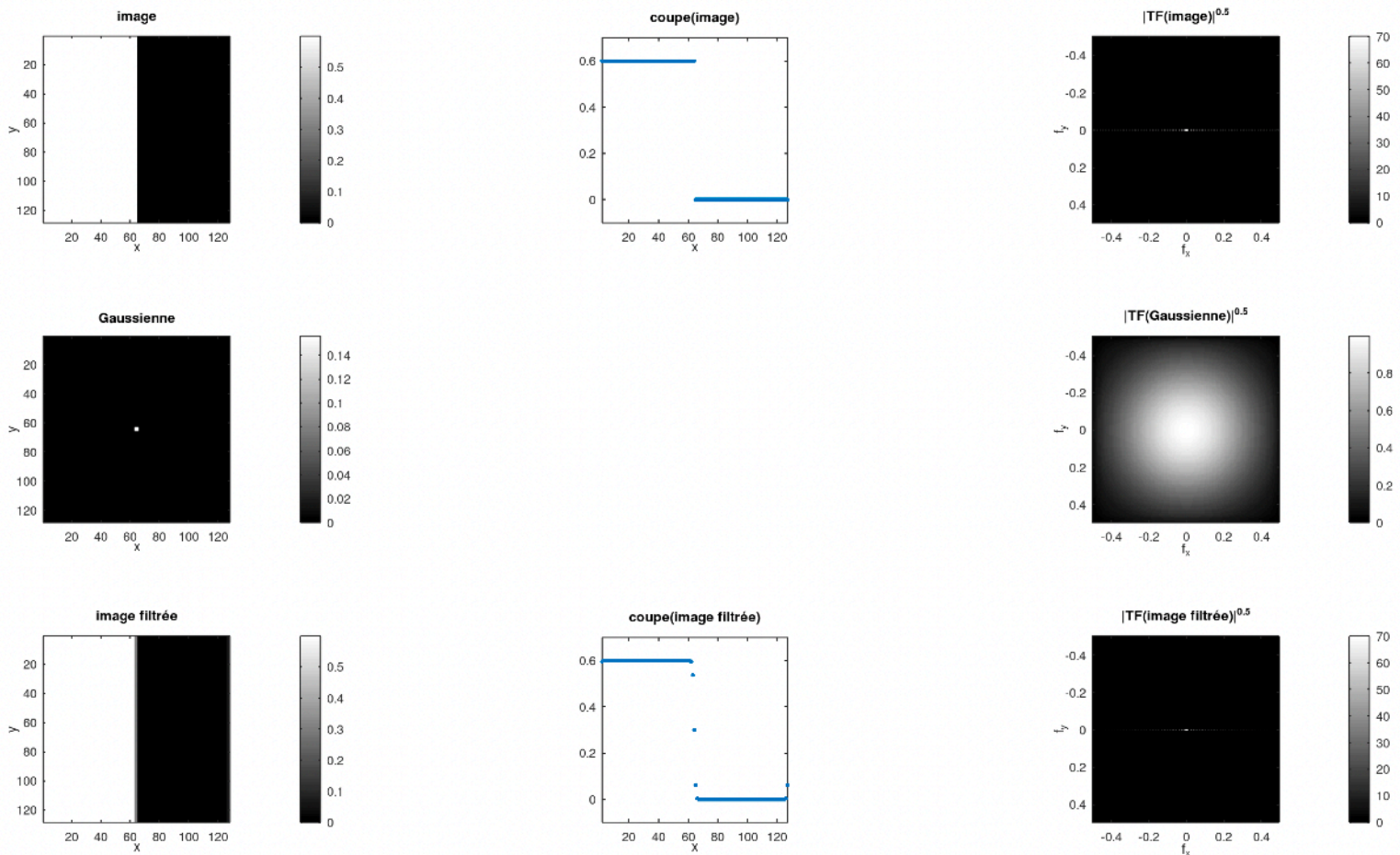
```
>> lfilt = ifftshift(ifft2(lchapfilt));  
>> whos(lfilt)  
>> subplot(3,2,5), imagesc(real(lfilt)), colorbar,  
title('image filtrée')
```

```
1 clear  
2 close all  
3 pkg load image  
4  
5 % image  
6 dim=128; I=zeros(dim,dim); I(:,1:dim/2)=0.6;  
7 fx=((0:dim-1)-dim/2)/dim; fy=fx;  
8  
9 figure, colormap('gray')  
10  
11 subplot(3,3,1), imagesc(I)  
12 colorbar, axis('square')  
13 title('image'), xlabel('x'), ylabel('y')  
14  
15 subplot(3,3,2), plot(I(64,:),'.')  
16 axis('square')  
17 title('coupe(image)'), xlim([2,dim-1]), ylim([-1 .7]), xlabel('x')  
18  
19 % FFT(image)  
20 Ichap=fft2(I); Ichapmod=abs(fftshift(Ichap));  
21 subplot(3,3,3), imagesc(fx,fy,Ichapmod.^5)  
22 colorbar, axis('square')  
23 title('|TF(image)|^{0.5}'), xlabel('f_x'), ylabel('f_y')  
24  
25 % filtre  
26 h=fspecial('gaussian',dim,.849);  
27 subplot(3,3,4), imagesc(h)  
28 colorbar, axis('square')  
29 title('Gaussienne'), xlabel('x'), ylabel('y')  
30  
31 % => fct de transfert  
32 hchap=fft2(h); hchapmod=abs(fftshift(hchap));  
33 subplot(3,3,6), imagesc(fx,fy,hchapmod.^5)  
34 colorbar, axis('square')  
35 title('|TF(Gaussienne)|^{0.5}'), xlabel('f_x'), ylabel('f_y')  
36
```

```

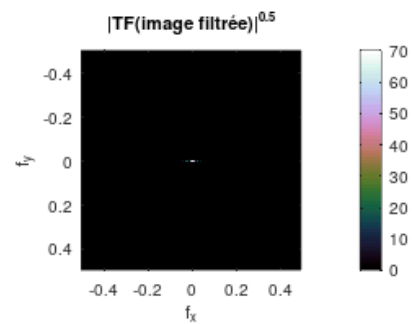
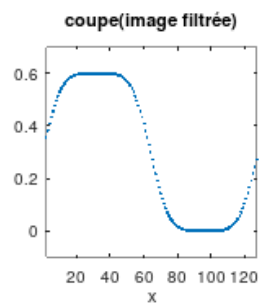
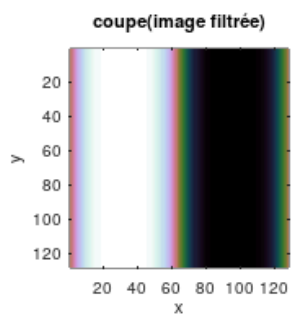
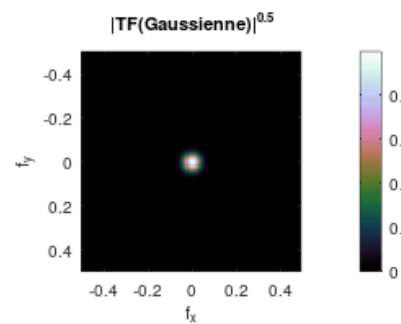
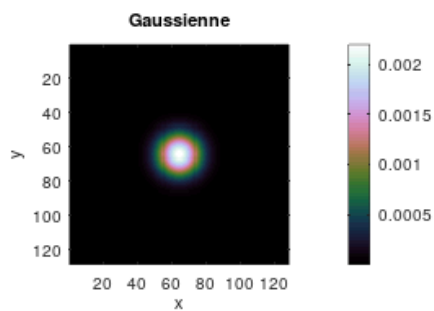
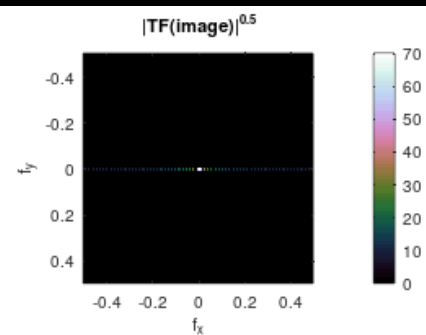
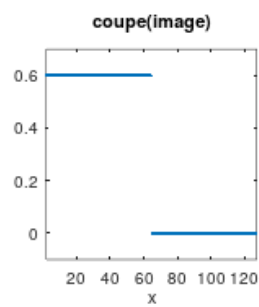
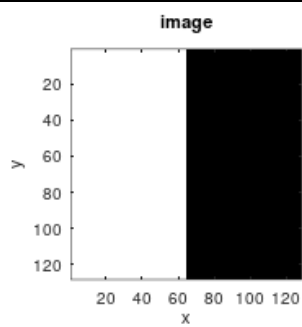
31 % => fct de transfert
32 hchap=fft2(h); hchapmod=abs(fftshift(hchap));
33 subplot(3,3,6), imagesc(fx,fy,hchapmod.^.5)
34 colorbar, axis('square')
35 title('|TF(Gaussienne)|^{0.5}'), xlabel('f_x'), ylabel('f_y')
36
37 % filtrage dans l'espace de Fourier
38 Ichapfilt=hchap.*Ichap; Ichapfiltmod=abs(fftshift(Ichapfilt));
39 subplot(3,3,9), imagesc(fx,fy,Ichapfiltmod.^.5)
40 colorbar, axis('square')
41 title('|TF(image filtrée)|^{0.5}'), xlabel('f_x'), ylabel('f_y')
42
43 % retour vers l'espace réel
44 Ifilt=ifftshift(ifft2(Ichapfilt));
45 subplot(3,3,7), imagesc(real(Ifilt))
46 axis('square'), colorbar
47 title('image filtrée'), xlabel('x'), ylabel('y')
48
49 subplot(3,3,8), plot(real(Ifilt(64,:)),'.')
50 axis('square'), xlim([2,dim-1]), ylim([-1 .7])
51 title('coupe(image filtrée)'), xlabel('x')

```



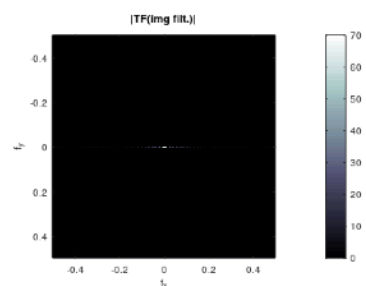
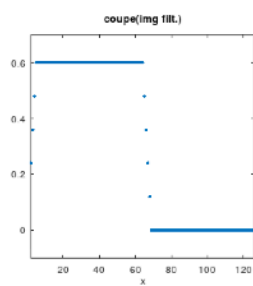
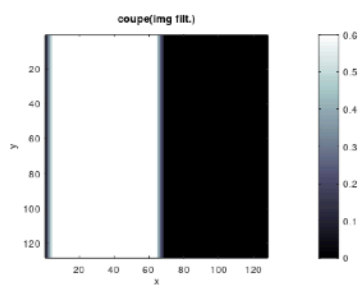
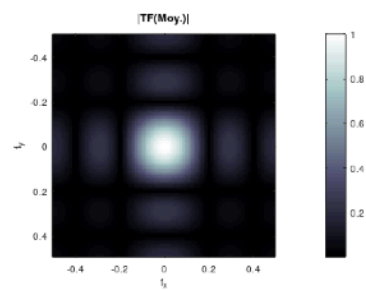
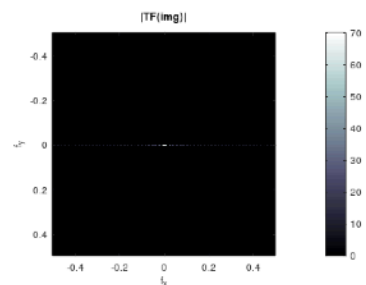
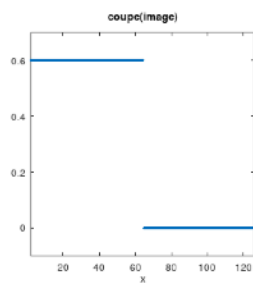
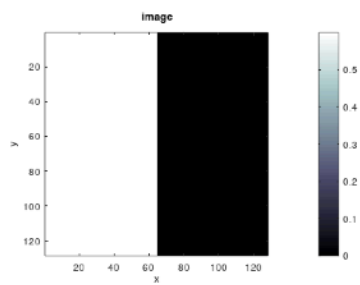
Exercice 5bis : Reprendre l'exemple ci-avant et diminuer la largeur de $\hat{h}$ d'un facteur 10. Que se passe-t-il ?

```
1 clear
2 close all
3 pkg load image
4
5 % image
6 dim=128; I=zeros(dim,dim); I(:,1:dim/2)=0.6;
7 fx=((0:dim-1)-dim/2)/dim; fy=fx;
8
9 figure, colormap('cubehelix')
10
11 subplot(3,3,1), imagesc(I)
12 colorbar, axis('square')
13 title('image'), xlabel('x'), ylabel('y')
14
15 subplot(3,3,2), plot(I(64,:),'.')
16 axis('square')
17 title('coupe(image)'), xlim([2,dim-1]), ylim([-1 .7]), xlabel('x')
18
19 % FFT(image)
20 Ichap=fft2(I); Ichapmod=abs(fftshift(Ichap));
21 subplot(3,3,3), imagesc(fx,fy,Ichapmod.^5)
22 colorbar, axis('square')
23 title('ITF(image)|^{0.5}'), xlabel('f_x'), ylabel('f_y')
24
25 % filtre
26 h=fspecial('gaussian',dim,.849*10);
27 subplot(3,3,4), imagesc(h)
28 colorbar, axis('square')
29 title('Gaussienne'), xlabel('x'), ylabel('y')
30
31 % => fct de transfert
32 hchap=fft2(h); hchapmod=abs(fftshift(hchap));
33 subplot(3,3,5), imagesc(fx,fy,hchapmod.^5)
34 colorbar, axis('square')
35 title('ITF(Gaussienne)|^{0.5}'), xlabel('f_x'), ylabel('f_y')
36
37 % filtrage dans l'espace de Fourier
38 Ichapfilt=hchap.*Ichap; Ichapfiltmod=abs(fftshift(Ichapfilt));
39 subplot(3,3,6), imagesc(fx,fy,Ichapfiltmod.^5)
40 colorbar, axis('square')
41 title('ITF(image filtrée)|^{0.5}'), xlabel('f_x'), ylabel('f_y')
42
43 % retour vers l'espace réel
44 Ifilt=ifftshift(ifft2(Ichapfilt));
45 subplot(3,3,7), imagesc(real(Ifilt))
46 axis('square'), colorbar
47 title('coupe(image filtrée)'), xlabel('x'), ylabel('y')
48
49 subplot(3,3,8), plot(real(Ifilt(64:)),'.')
50 axis('square'), xlim([2,dim-1]), ylim([-1 .7])
51 title('coupe(image filtrée)'), xlabel('x')
```



Exercice 6 : Faire de même avec la moyenne glissante, en 3x3 puis 5x5.

```
1 clear
2 close all
3 pkg load image
4
5 % image
6 dim=128; I=zeros(dim,dim); I(:,1:dim/2)=0.6; % image = marche 0.6/0
7 fx=((0:dim-1)-dim/2)/dim; fy=fx; % étalonnage des fréquences
8
9 figure, colormap('bone') % colormap "bone" (voir help)
10
11 subplot(3,3,1), imagesc(I)
12 colorbar, axis('square')
13 title('image'), xlabel('x'), ylabel('y')
14
15 subplot(3,3,2), plot(I(dim/2,:),'.')
16 title('coupe(image)'), xlabel('x')
17 axis('square'), xlim([2,dim-2]), ylim([-1 .7])
18
19 % FFT(image)
20 Ichap=fft2(I); % FFT(image)
21 Ichapmod=abs(fftshift(Ichap)); % module réordonné
22 subplot(3,3,3), imagesc(fx,fy,Ichapmod.^5)
23 colorbar, axis('square')
24 title('|TF(img)|'), xlabel('f_x'), ylabel('f_y')
25
26 % filtre
27 nn=5; h=fspecial('average',nn); % moyenne glissante 3x3 ou 5x5
28 % on peut utiliser aussi ones(nn,nn).
29 % et on peut ensuite recentrer le filtre h dans un tableau dim*dim "à la main,
30 % ou utiliser la commande "padarray", mais ce n'est pas nécessaire puisque l'on
31 % pourra tout aussi bien faire ça au vol avec la commande FFT2.
32
33 % => fct de transfert
34 hchap=fft2(h,dim,dim); % masque=FFT(filtre)
35 hchapmod=abs(fftshift(hchap)); % module
36 subplot(3,3,6), imagesc(fx,fy,hchapmod)
37 colorbar, axis('square')
38 title('|TF(Moy.)|'), xlabel('f_x'), ylabel('f_y')
39
40 % filtrage dans l'espace de Fourier
41 Ichapfilt=Ichap.*hchap; % filtrage
42 Ichapfiltmod=abs(fftshift(Ichapfilt)); % module du résultat dans Fourier
43 subplot(3,3,9), imagesc(fx,fy,Ichapfiltmod.^5)
44 colorbar, axis('square')
45 title('|TF(img filt.)|'), xlabel('f_x'), ylabel('f_y')
46
47 % retour vers l'espace réel
48 Ifilt=real(ifft2(Ichapfilt)); % image filtrée
49 subplot(3,3,7), imagesc(Ifilt)
50 colorbar, axis('square')
51 title('coupe(img filt.)'), xlabel('x'), ylabel('y')
52
53 subplot(3,3,8), plot(Ifilt(64,:),'.')
54 title('coupe(img filt.)'), xlabel('x')
55 axis('square'), xlim([2,dim-2]), ylim([-1 .7])
```



Exercice 7 : Faire de même avec le filtre passe-haut vu au chapitre 3.

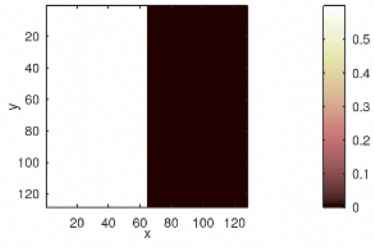
-1 -1 -1

-1 9 -1

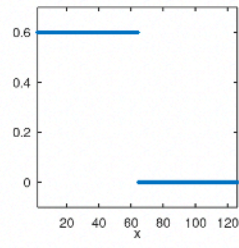
-1 -1 -1

```
1 clear
2 close all
3 pkg load image
4
5 % image
6 dim=128; I=zeros(dim,dim); I(:,1:dim/2)=0.6; % image=marche 0.6/0
7 fx=((0:dim-1)-dim/2)/dim; fy=fx; % étalonnage des fréquences
8
9 figure, colormap('pink')
10
11 subplot(3,3,1), imagesc(I), colorbar, axis('square')
12 title('img'), xlabel('x'), ylabel('y')
13
14 subplot(3,3,2), plot(I(64,:),'.'), axis('square')
15 title('coupe(img)'), xlim([2,dim-2]), ylim([-1 .7]), xlabel('x')
16
17 % FFT(image)
18 Ichap=fft2(I);
19 Ichapmod=abs(fftshift(Ichap));
20
21 subplot(3,3,3), imagesc(fx,fy,Ichapmod.^.5), colorbar, axis('square')
22 title('|TF(img)|^{0.5}'), xlabel('f_x'), ylabel('f_y')
23
24 % filtre => fct de transfert
25 h=ones(3,3); h(2,2)=9; % filtre passe-haut
26 hchap=fft2(h,dim,dim); % fct de transfert corr.
27 hchapmod=abs(fftshift(hchap));
28 subplot(3,3,6), imagesc(fx,fy,hchapmod), colorbar, axis('square')
29 title('|TF(Passe-haut)|'), xlabel('f_x'), ylabel('f_y')
30
31 % filtrage dans l'espace de Fourier
32 Ichapfilt=Ichap.*hchap;
33 Ichapfiltmod=abs(fftshift(Ichapfilt));
34 subplot(3,3,9), imagesc(fx,fy,Ichapfiltmod.^.5), colorbar, axis('square')
35 title('|TF(img filt.)|^{0.5}'), xlabel('f_x'), ylabel('f_y')
36
37 % image filtrée
38 Ifilt=real(ifft2(Ichapfilt));
39 subplot(3,3,7), imagesc(Ifilt), colorbar, axis('square')
40 title('img filt.'), xlabel('x'), ylabel('y')
41
42 subplot(3,3,8), plot(Ifilt(64,:),'.'), title('coupe(img filt.)')
43 axis('square'), xlim([2,dim-2]), ylim([-3 3])
```

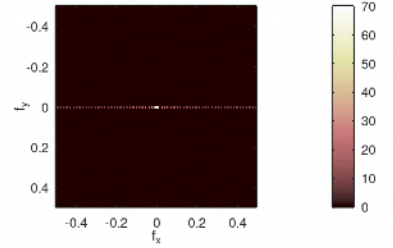
img



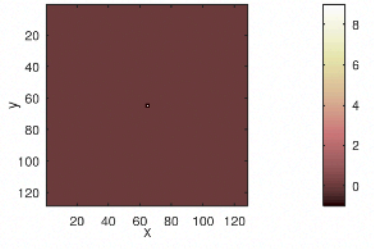
coupe(img)



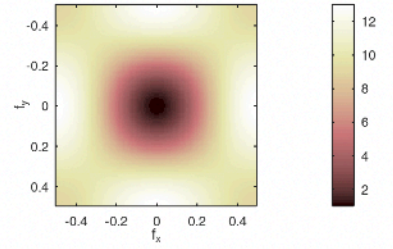
$|\text{TF}(\text{img})|^{0.5}$



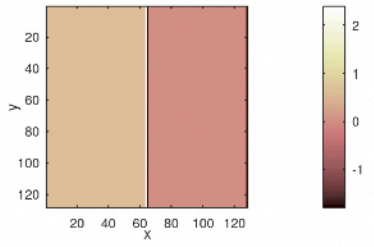
Passe-haut



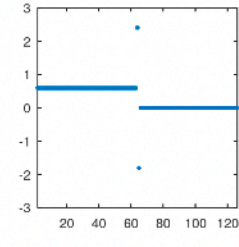
$|\text{TF}(\text{Passe-haut})|$



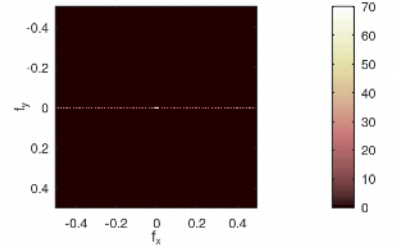
img filt.



coupe(img filt.)



$|\text{TF}(\text{img filt.})|^{0.5}$



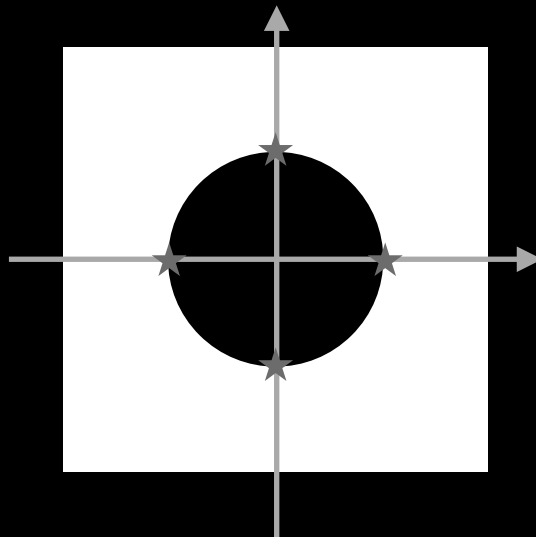
Filtrage passe-haut, passe-bande/coupe-bande, dérivateur — directement dans le plan de Fourier

L'avantage d'être dans le plan de Fourier, c'est que l'on peut modeler les fréquences comme on le désire... Y compris le faire de manière plus simple.

- Cas du filtrage passe-haut : créer $\hat{h}$ tel que $|\hat{h}|$ croît quand la fréquence (spatiale) croît également => on favorise les fréquences les plus élevées => effet contraire du filtre passe-bas.

Application : Atténuer les variations lentes de l'image, pour faire ressortir les variations brusques (contours).

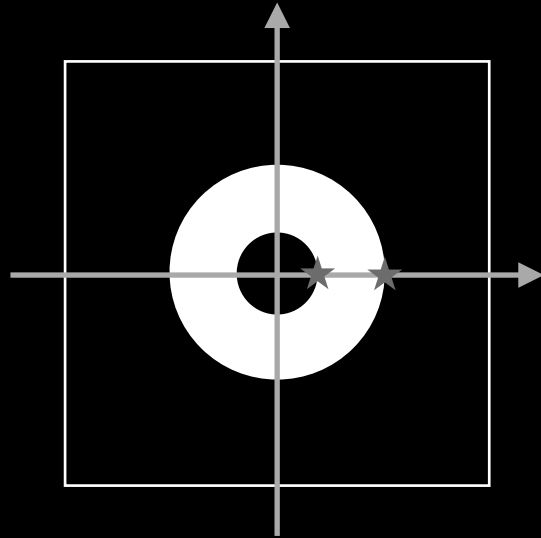
Exemple :



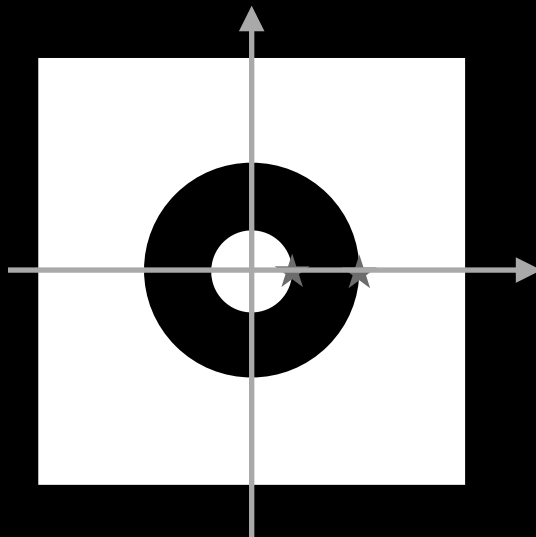
★ fréq. de coupure

(il s'agit bien d'un filtrage directement dans le plan de Fourier, appliqué à la TF de l'image)

- Cas du filtrage passe-bande : on laisse une bande de fréquences entre deux fréquences de coupures.



- Cas du filtrage coupe-bande :



- Filtre dérivateur :

En fait un filtre passe-haut, construit à partir de :

$$f'(x,y) \xrightarrow{TF} 2i\pi u \quad 2i\pi v \quad \hat{f}(u,v)$$

Exercice 8 :

Filtrer (en masquant dans le plan de Fourier : notion de fonction de transfert) avec un filtre passe-bas, carré, de largeur $\text{dim}/2$. Commenter le résultat.

```
1 clear
2 close all
3 %pkg load image
4
5 % image
6 dim=128; I=zeros(dim,dim); I(:,1:dim/2)=0.6; % marche 0.6/0
7 fx=((0:dim-1)-dim/2)/dim; fy=fx; % fréquences
8
9 figure, colormap('colorcube') % colormapp "colorcube"
10
11 subplot(3,3,1), imagesc(I)
12 colorbar, axis('square')
13 title('img'), xlabel('x'), ylabel('y')
14
15 subplot(3,3,2), plot(I(dim/2,:),'.')
16 axis('square'), ylim([-1 .7])
17 title('coupe(img)'), xlabel('x')
18
19 % FFT(image)
20 Ichap=fft2(I,dim,dim);
21 Ichapmod=abs(fftshift(Ichap));
22 subplot(3,3,3), imagesc(fx,fy,Ichapmod.^5)
23 colorbar, axis('square')
24 title('|TF(img)|^{0.5}'), xlabel('f_x'), ylabel('f_y')
25
26 % "filtre" (en fait fct de transfert directement!)
27 hchap=zeros(dim,dim); nn=dim/2;
28 hchap(dim/2-nn/2+1:dim/2+nn/2,dim/2-nn/2+1:dim/2+nn/2)=1.;
29 % plus smooth : le filtre gaussien...
30 % hchap=fspecial('gaussian',dim,nn); hchap=hchap/max(max(hchap));
31
32 subplot(3,3,6), imagesc(fx,fy,hchap)
33 colorbar, axis('square')
34 title('hchap passe-bas'), xlabel('f_x'), ylabel('f_y')
35
36 % filtrage dans l'espace de Fourier
37 Ichapfilt=fftshift(hchap).*Ichap;
38 Ichapfiltmod=abs(fftshift(Ichapfilt));
39 subplot(3,3,9), imagesc(fx,fy,Ichapfiltmod.^5)
40 colorbar, axis('square')
41 title('|TF(img filt.)|^{0.5}'), xlabel('f_x'), ylabel('f_y')
42
43 % retour dans l'espace réel
44 Ifilt=ifft2(Ichapfilt);
45 subplot(3,3,7), imagesc(real(Ifilt))
46 colorbar, axis('square')
47 title('img filt.'), xlabel('x'), ylabel('y')
48
49 subplot(3,3,8), plot(real(Ifilt(64,:)),'.')
50 axis('square'), ylim([-1 .7])
51 title('coupe(img filt.)'), xlim([2,dim-2]), xlabel('x')
```

