

```

1 clear
2 close all
3
4 % sinsusoïde
5 dim=100; % dimension linéaire de l'image
6 Dx=1; % pas (taille du pixel)
7 x=0:Dx:(dim-1)*Dx; % vecteur des x
8 whos x
9 y=x; % vecteur des y
10
11 Tx=20; % période de la sinusoïde
12 I=ones(dim,1)*(1+cos(2*pi*x/Tx));
13 whos I % sinusoïde (en fait un cosinus) selon x,
14 % de période Tx et évoluant de 0 à 2
15
16 figure(1), colormap('gray')
17
18 subplot(3,2,1), imagesc(x,y,I)
19 title('Sinusoïde(x,y)'), xlabel('x'), ylabel('y'), colorbar, axis('square')
20
21 subplot(3,2,3), plot(x,I(dim/2+1,:)), axis('square')
22 title('Sinusoïde(x)'), xlabel('x'), ylabel('sin(2*pi*x/Tx)')
23
24 subplot(3,2,5), plot(y,I(:,dim/2+1)), axis('square')
25 title('Sinusoïde(y)'), xlabel('y'), ylabel('constante')
26

```

```

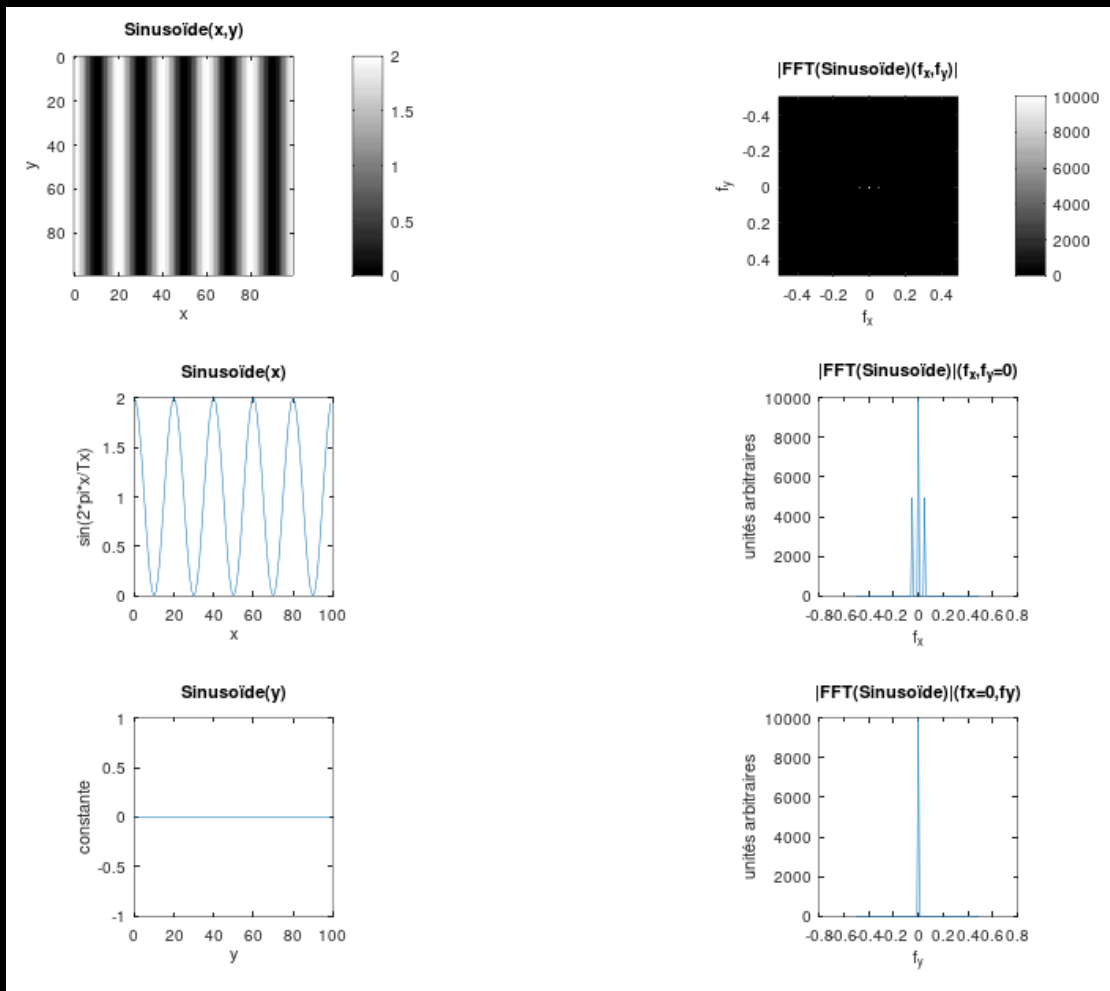
27 % FFT(sinusoïde)
28 Ichap=fft2(I); % FFT(image)
29 whos Ichap
30 Ichapmod=abs(fftshift(Ichap));
31 whos Ichapmod % module de FFT(image)
32
33 fx=1/(dim*Dx)*(-dim/2:dim/2-1); fy=fx;
34 % vecteurs des fréquences spatiales
35
36 subplot(3,2,2), imagesc(fx,fy,Ichapmod)
37 title('|FFT(Sinusoïde)(f_x,f_y)|'), xlabel('f_x'), ylabel('f_y')
38 colorbar, axis('square')
39
40 subplot(3,2,4), plot(fx,Ichapmod(dim/2+1,:)), axis('square'),
41 title('|FFT(Sinusoïde)|(f_x,f_y=0)|'), xlabel('f_x'), ylabel('unités arbitraires')
42
43 subplot(3,2,6), plot(fy,Ichapmod(:,dim/2+1)), axis('square')
44 title('|FFT(Sinusoïde)|(f_x=0,f_y)|'), xlabel('f_y'), ylabel('unités arbitraires')
45

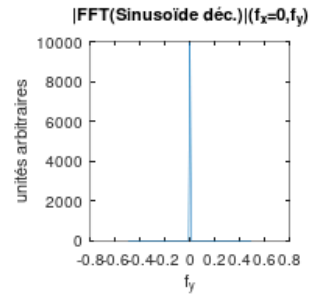
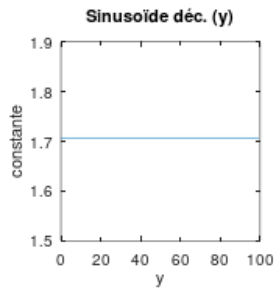
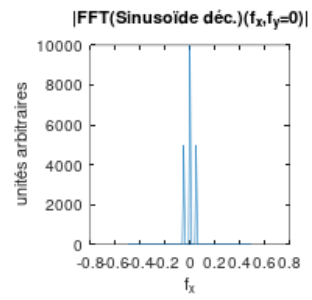
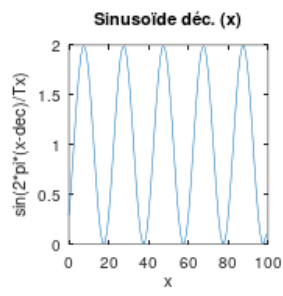
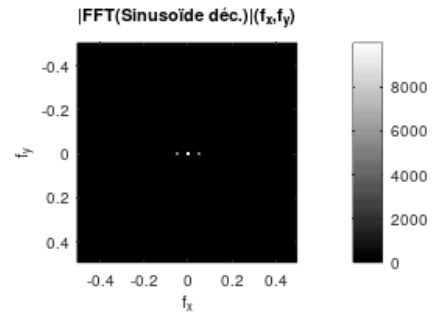
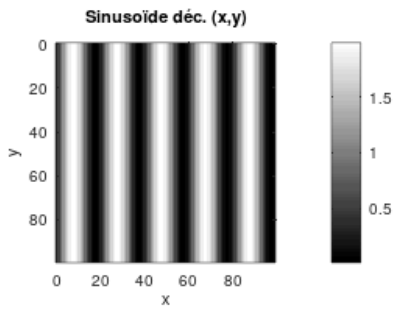
```

```

46 % cas de la sinusoïde décalée
47 dec=7.5;
48 Idec=ones(dim,1)*(1+cos(2*pi*(x-dec)/Tx));
49 Idecchap=fft2(Idec);
50 Idecchapmod=abs(fftshift(Idecchap));
51
52 figure(2), colormap('gray')
53
54 subplot(3,2,1), imagesc(x,y,Idec)
55 colorbar, axis('square')
56 title('Sinusoïde déc. (x,y)'), xlabel('x'), ylabel('y')
57
58 subplot(3,2,2), imagesc(fx,fy,Idecchapmod)
59 axis('square'), colorbar
60 title('|FFT(Sinusoïde déc.)|(f_x,f_y)'), xlabel('f_x'), ylabel('f_y')
61
62 subplot(3,2,3), plot(x,Idec(dim/2+1,:))
63 axis('square')
64 title('Sinusoïde déc. (x)'), xlabel('x'), ylabel('sin(2*pi*(x-dec)/Tx)')
65
66 subplot(3,2,4), plot(fx,Idecchapmod(dim/2+1,:))
67 axis('square')
68 title('|FFT(Sinusoïde déc.)(f_x,f_y=0)|')
69 xlabel('f_x'), ylabel('unités arbitraires')
70
71 subplot(3,2,5), plot(y,Idec(:,dim/2+1))
72 axis('square')
73 title('Sinusoïde déc. (y)'), xlabel('y'), ylabel('constante')
74
75 subplot(3,2,6), plot(fy,Ichapmod(:,dim/2+1))
76 axis('square')
77 title('|FFT(Sinusoïde déc.)|(f_x=0,f_y)')
78 xlabel('f_y'), ylabel('unités arbitraires')

```





Autres exemples communs/utiles

—> Porte bidimensionnelle

$$f(x, y) = \Pi\left(\frac{x}{a}\right) \Pi\left(\frac{y}{b}\right)$$

TF(porte en x) = sinc(u a) ; TF(porte en y) = sinc(v b)

$$\hat{f}(u, v) = \text{sinc}(u a) \text{sinc}(v b)$$

rmq 1 : $\Pi(x/a)$ et $\Pi(y/b)$ sont des fonctions séparables en x et en y => pas de convolution dans le plan de Fourier ici !

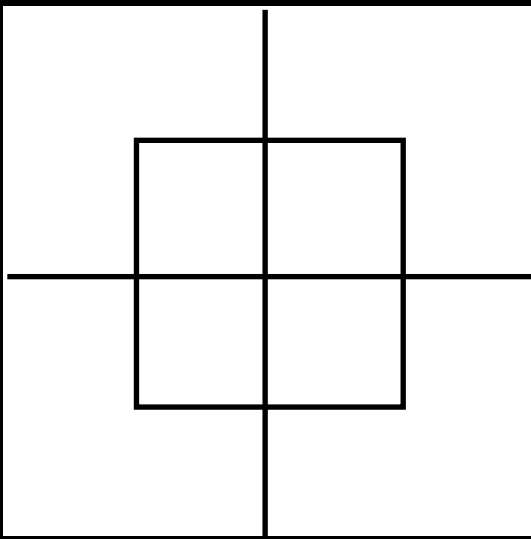
rmq 2 : Les portes en x et en y servent ici à décrire le fait qu'une image est de taille $a \times b$ (en pixels).

rmq 3 : $\Pi(x/a)$ = porte de largeur a en x.
sinc(u a) = largeur inversement prop. à a .

Quand a augmente, le pic du *sinc* augmente, mais sa largeur diminue => plus la porte est large, plus le *sinc* dans Fourier est étroit.

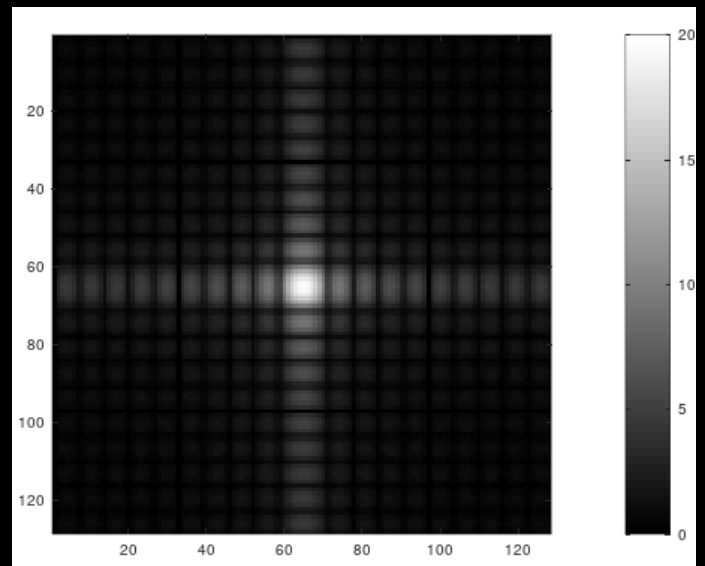
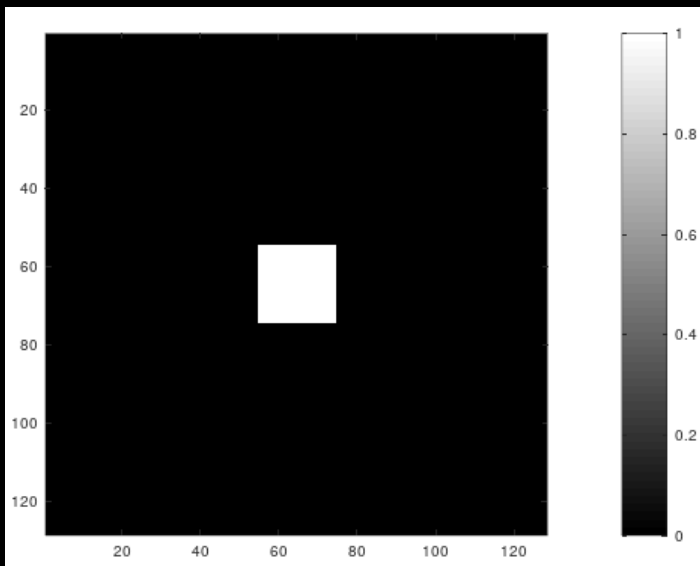
sous Matlab/Octave :

```
>> dim=128; a=20; b=20;  
>> P=zeros(dim,dim);
```



1 dim/2 dim/2+1 dim

```
>> P(dim/2-a/2+1:dim/2+a/2,dim/2-b/2+1:dim/2+b/2)=1;
>> colormap(gray)
>> imagesc(P), axis('square')
>> Pchap=fft2(P);
>> imagesc((abs(fftshift(Pchap))).^0.5), colorbar
```



Exercice 2 : Reprendre l'exemple précédent et augmenter/diminuer a et b . Que remarque-t-on ? (Prendre par exemple 3 valeurs : 20, 10, 40) (Mettre aussi les bonnes échelles de fréquences spatiales dans Fourier...)

```

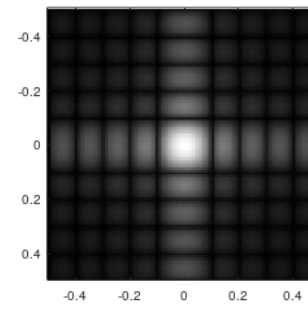
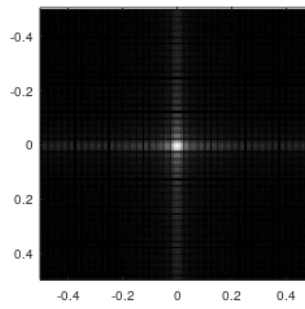
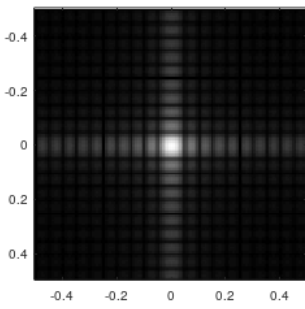
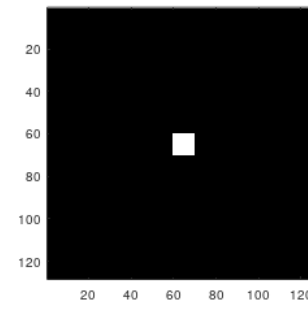
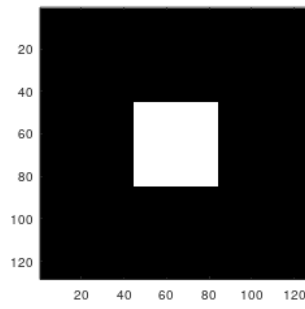
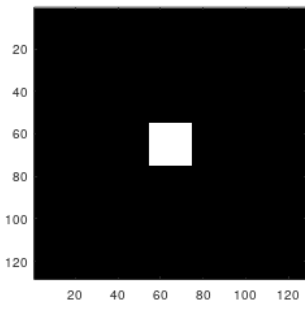
1 clear
2 close all
3
4 dim=128; fx=1/(dim*1)*(-dim/2:dim/2-1); fy=fx;
5
6 % 1er exemple
7 P20=zeros(dim,dim); a=20; b=20;
8 P20(dim/2-a/2+1:dim/2+a/2, dim/2-b/2+1:dim/2+b/2)=1.0;
9 Pchapmod20=abs(ffftshift(ffft2(P20)));
10
11 % 2me exemple
12 P40=zeros(dim,dim); a=40; b=40;
13 P40(dim/2-a/2+1:dim/2+a/2, dim/2-b/2+1:dim/2+b/2)=1.0;
14 Pchapmod40=abs(ffftshift(ffft2(P40)));
15
16 % 3me exemple
17 P10=zeros(dim,dim); a=10; b=10;
18 P10(dim/2-a/2+1:dim/2+a/2, dim/2-b/2+1:dim/2+b/2)=1.0;
19 Pchapmod10=abs(ffftshift(ffft2(P10)));
20

```

```

21 % figure finale
22 figure, colormap('gray')
23
24 subplot(2,3,1), imagesc(P20)
25 colorbar, axis('square')
26 title('Porte(x/20,y/20)'), xlabel('x'), ylabel('y')
27
28 subplot(2,3,2), imagesc(P40)
29 colorbar, axis('square')
30 title('Porte(x/40,y/40)'), xlabel('x'), ylabel('y')
31
32 subplot(2,3,3), imagesc(P10)
33 colorbar, axis('square')
34 title('Porte(x/10,y/10)'), xlabel('x'), ylabel('y')
35
36 subplot(2,3,4), imagesc(fx,fy,Pchapmod20.^5)
37 colorbar, axis('square')
38 title('sqrt(|FFT(Porte_{20})|(f_x,f_y))'), xlabel('f_x'), ylabel('f_y')
39
40 subplot(2,3,5), imagesc(fx,fy,Pchapmod40.^5)
41 colorbar, axis('square')
42 title('sqrt(|FFT(Porte_{40})|(f_x,f_y))'), xlabel('f_x'), ylabel('f_y')
43
44 subplot(2,3,6), imagesc(fx,fy,Pchapmod10.^5)
45 colorbar, axis('square')
46 title('sqrt(|FFT(Porte_{10})|(f_x,f_y))'), xlabel('f_x'), ylabel('f_y')

```



—> Gaussienne bidimensionnelle

$$f(x, y) = \exp \left\{ -\pi \frac{x^2}{a^2} \right\} \exp \left\{ -\pi \frac{y^2}{b^2} \right\}$$

$$\hat{f}(u, v) = a \exp \left\{ -\pi (a u)^2 \right\} b \exp \left\{ -\pi (b v)^2 \right\}$$

sous Matlab/Octave :

```
>> dim=128; sigma=10;  
>> G = fspecial('gaussian', dim, sigma);  
>> ...
```

Exercice 3 : Idem que l'exercice 2 mais avec la Gaussienne...

(En prenant 3 valeurs de sigma et en étalonnant correctement les fréquences spatiales)

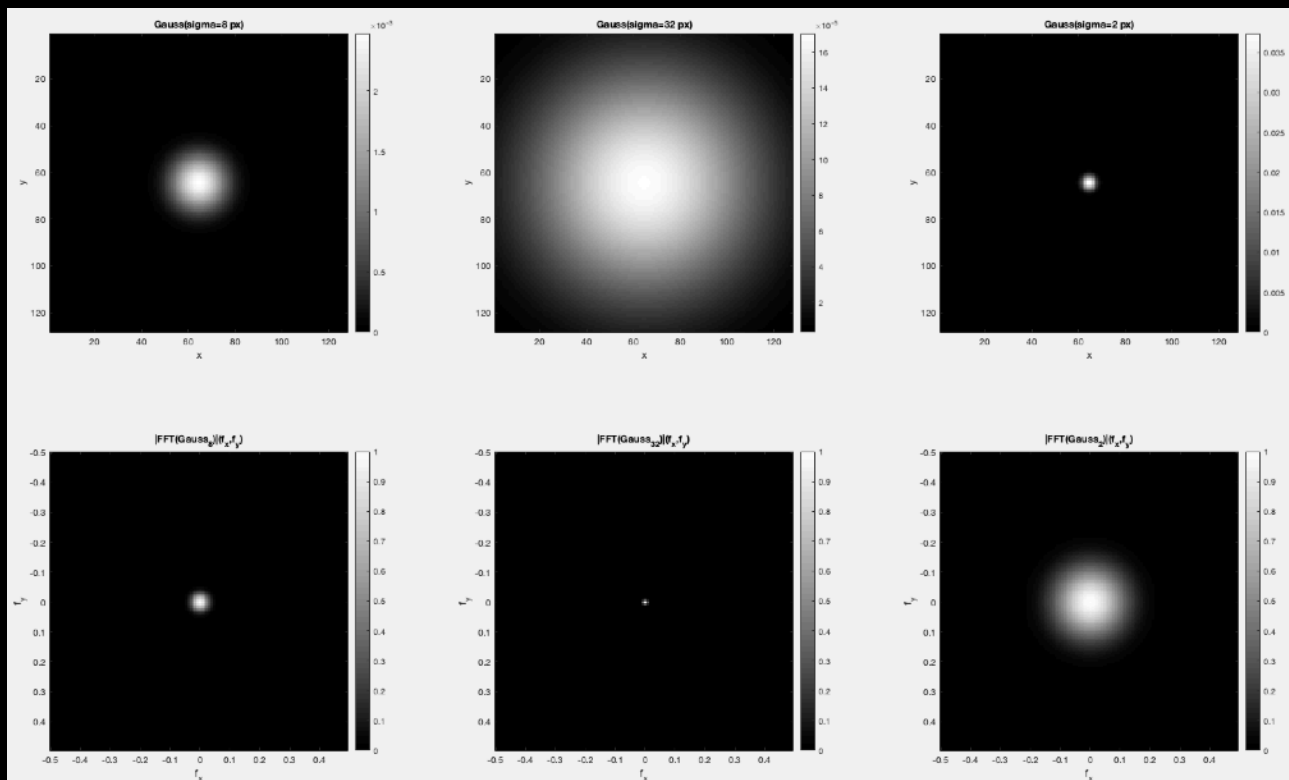
```
1 clear  
2 close all  
3 pkg load image  
4  
5 dim=128; fx=1/(dim*1)*(-dim/2:dim/2-1); fy=fx;  
6  
7 % 1er exemple  
8 a=8;  
9 G8=fspecial('gaussian',dim,a);  
10 Gchapmod8=abs(fftshift(fft2(G8)));  
11  
12 % 2me exemple  
13 a=32;  
14 G32=fspecial('gaussian',dim,a);  
15 Gchapmod32=abs(fftshift(fft2(G32)));  
16  
17 % 3me exemple  
18 a=2;  
19 G2=fspecial('gaussian',dim,a);  
20 Gchapmod2=abs(fftshift(fft2(G2)));  
21
```



```

22 % figure finale
23 figure, colormap('gray')
24
25 subplot(2,3,1), imagesc(G8)
26 colorbar, axis('square')
27 title('Gauss(sigma=8 px)'), xlabel('x'), ylabel('y')
28
29 subplot(2,3,2), imagesc(G32)
30 colorbar, axis('square')
31 title('Gauss(sigma=32 px)'), xlabel('x'), ylabel('y')
32
33 subplot(2,3,3), imagesc(G2)
34 colorbar, axis('square')
35 title('Gauss(sigma=2 px)'), xlabel('x'), ylabel('y')
36
37 subplot(2,3,4), imagesc(fx,fy,Gchapmod8)
38 colorbar, axis('square')
39 title('|FFT(Gauss_{8})|(f_x,f_y)'), xlabel('f_x'), ylabel('f_y')
40
41 subplot(2,3,5), imagesc(fx,fy,Gchapmod32)
42 colorbar, axis('square')
43 title('|FFT(Gauss_{32})|(f_x,f_y)'), xlabel('f_x'), ylabel('f_y')
44
45 subplot(2,3,6), imagesc(fx,fy,Gchapmod2)
46 colorbar, axis('square')
47 title('|FFT(Gauss_{2})|(f_x,f_y)'), xlabel('f_x'), ylabel('f_y')

```



—> Dirac :

$$\delta(x,y) \xrightarrow{TF} \mathbf{1}(u,v)$$

—> Continu :

$$a \mathbf{1}(x,y) \xrightarrow{TF} a \delta(u,v)$$

—> Peigne de Dirac :

$$\text{III}(a)(x) \cdot \text{III}(b)(y) \xrightarrow{TF} \text{III}(1/a)(u) \cdot \text{III}(1/b)(v)$$

La fonction *Sha* de période a en x ($\text{III}(a)(x)$) et de période b en y ($\text{III}(b)(y)$) décrit, notamment, l'échantillonnage d'une image (les pixels!)

(on prend dans la suite $\Delta y = \Delta x =$ taille du pixel)

$$\Rightarrow \text{image discrète}(x,y) = \text{image continue}(x,y) \cdot (\text{III}(\Delta x)(x) \cdot \text{III}(\Delta x)(y))$$

$$\Rightarrow DFT(\text{img discrète})(u,v) = DFT(\text{img continue})(u,v) * (\text{III}(1/\Delta x)(u) \cdot \text{III}(1/\Delta x)(v))$$

(où $*$ décrit le produit de convolution)

Or : $1/\Delta x =$ largeur de toute la *DFT*

Donc, on a : échantillonnage dans le plan direct \Rightarrow périodisation dans le plan de Fourier !

• 3 propriétés remarquables de la TF

—> Dilatation

$$f\left(\frac{x}{a}, \frac{y}{b}\right) \text{ — par TF —} \rightarrow |a| |b| \hat{f}(a u, b v)$$

Ce qui est étroit dans l'espace direct est large dans l'espace de Fourier, et vice versa.

—> Translation

$$f(x + x_0, y + y_0) \text{ — par TF —} \rightarrow \hat{f}(u, v) \exp\{-2i\pi(x_0 u + y_0 v)\}$$

car on a au passage...

$$f(x + x_0, y + y_0) = f(x, y) * \delta((x + x_0, y + y_0))$$

rmq 1 : ce qui constitue un excellent moyen de **décaler** ou **interpoler** une image... (simple multiplication de la TF par un terme de phase.)

rmq 2 : le module est inchangé.

—> Convolution

$$f(x, y) * g(x, y) \text{ — TF —} \rightarrow \hat{f}(u, v) \cdot \hat{g}(u, v)$$

$$f(x, y) \cdot g(x, y) \text{ — TF —} \rightarrow \hat{f}(u, v) * \hat{g}(u, v)$$

C'est la propriété que l'on va utiliser pour le filtrage (... et qui est au cœur de la déconvolution...)

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

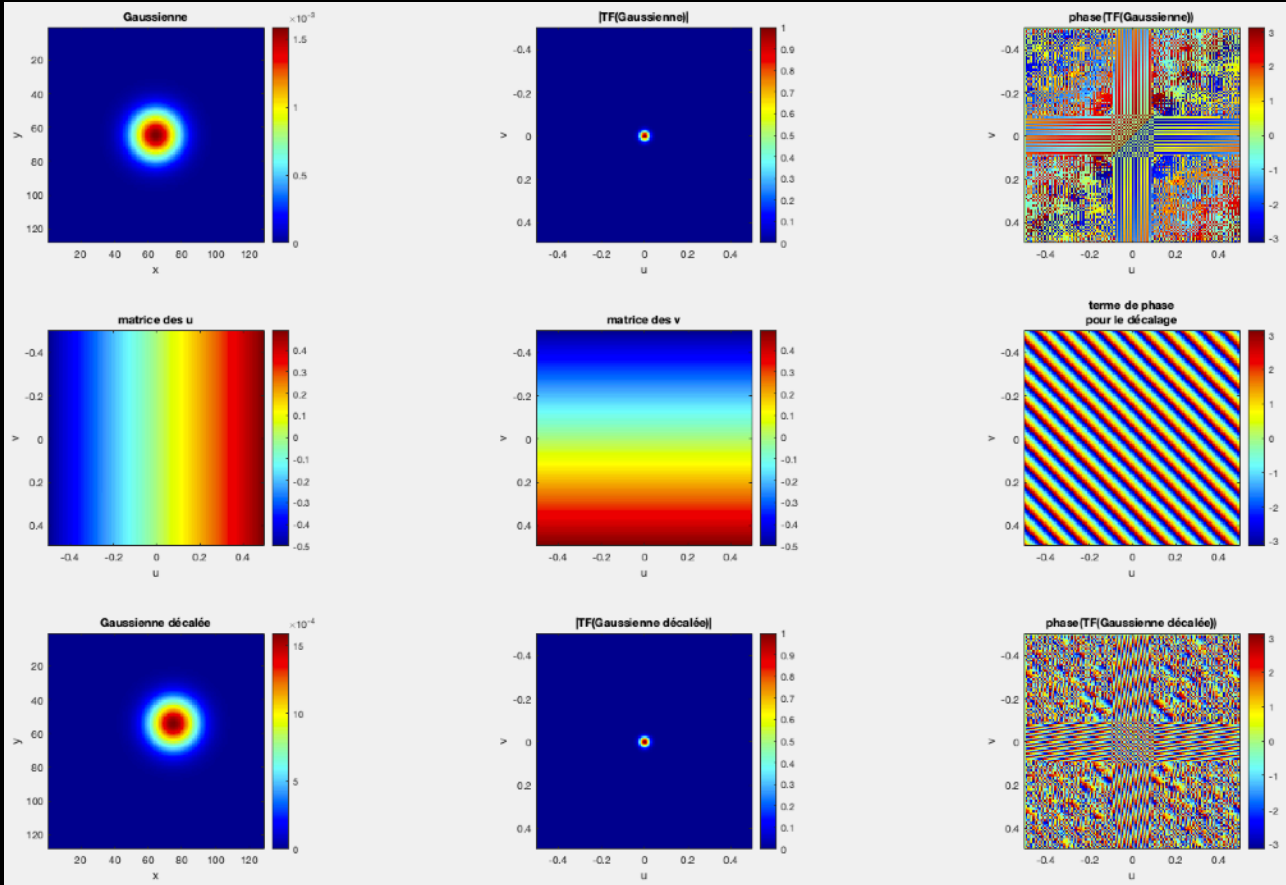
L'étape (2) revient à écrire correctement le terme de phase $\exp(-2 i \pi (x_0 u + y_0 v))$ dans Fourier... terme qui doit être un tableau de mêmes dimensions que l'image de départ (et donc sa TF et donc la phase de cette TF). Ici x_0 et y_0 sont les décalages respectivement en x et en y, et ce sont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

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; fy=fx;
11
12 % Gaussienne
13 a=10;
14 G=fspecial('gaussian',dim,a); % Gaussienne
15 Gchap = fft2(G); % TF(Gaussienne)
16
17 % terme de phase
18 decx= 10.4; % décalage en x
19 decy=-10.4; % décalage en y
20 uo=ones(dim,1)*fx; % matrice des u (par colonne)
21 vo=uo'; % matrice des v (par ligne)
22 u=fftshift(uo); % car le plan de la FFT est "shifté"
23 v=fftshift(vo); % d'un demi-tableau en u et en v
24 decfou=exp(-complex(0,1)*2*pi*(u*decx+v*decy)); % TF(décalage) ; complex(0,1)=i
25
26 % décalage
27 Gdecchap = Gchap.*decfou; % TF(Gaussienne décalée)
28 Gdec=real(ifft2(Gdecchap)); % retour dans le plan direct
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, uo)
46 colorbar, axis('square')
47 title('matrice des u'), xlabel('u'), ylabel('v')
48
49 subplot(3,3,5), imagesc(fx, fy, vo)
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 :

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

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

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

h = 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('|ITF(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('|ITF(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')

```

