

VI Transformée de Fourier discrète et filtrage

- Revisite du filtrage *linéaire* vu précédemment, mais sous l'angle du filtrage « directement » dans le plan de Fourier, i.e. le plan des fréquences spatiales (par le biais de la Transformée de Fourier (TF) bidimensionnelle).

=> de nouveaux horizons vont s'ouvrir à nous en termes d'amélioration/de restauration d'image...

- Illustrations/applications :
 - filtrage passe-bas (afin d'atténuer les hautes fréquences, de « lisser » une image),
 - filtrage passe-haut (afin de faire ressortir les hautes fréquences, par exemple des contours),
 - filtrage passe-bande (pour enlever par exemple des biais périodiques, comme un tramage).
- D'autres applications abordées à la fin de ce chapitre (restauration/reconstruction d'image => déconvolution).

(1) TF discrète (TFD, DFT en anglais) bidimensionnelle

- Soit $f(x, y)$, avec : $x=0, 1, 2, \dots, M-1$, et : $y=0, 1, 2, \dots, N-1$, une image digitale de taille $M \times N$.

La TFD de $f(x, y)$, $\hat{f}(u, v)$, s'écrit :

$$\hat{f}(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \{-i2\pi(ux/M + vy/N)\}$$

Remarque : dans le monde continu, on a :

$$\hat{f}(u, v) = \int \int f(x, y) \exp \{-i2\pi(ux + vy)\} dx dy$$

- x et y sont les coordonnées spatiales des pixels de l'image $f(x, y)$
- u et v sont les coordonnées fréquentielles des *frequels* (*frequels* = « *frequency elements* » = les pixels dans le plan de Fourier) de $\hat{f}(u, v)$.
- La TFD inverse s'écrit :

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{f}(u, v) \exp \{i2\pi(ux/M + vy/N)\}$$

Même remarque :

$$f(x, y) = \int \int \hat{f}(u, v) \exp \{i2\pi(ux + vy)\} du dv$$

- $\hat{f}(0,0)$ = « composante continu » (vient de l'électronique) = intégrale (somme dans le monde discret) de $f(x, y)$
- Même si $f(x, y)$ est réelle, $\hat{f}(u, v)$ est a priori complexe.
- On peut donc a priori exprimer $\hat{f}(u, v)$ comme :

$$\hat{f}(u, v) = |\hat{f}(u, v)| \exp \{i\phi(u, v)\}$$

où $|\hat{f}(u, v)|$ est le module de $\hat{f}(u, v)$ et $\phi(u, v)$ son argument.

$$|\hat{f}(u, v)| = \sqrt{Re^2(\hat{f}(u, v)) + Im^2(\hat{f}(u, v))}$$

$$\phi(u, v) = \arctan \frac{Im(\hat{f}(u, v))}{Re(\hat{f}(u, v))}$$

- Densité spectrale :

$$|\hat{f}(u, v)|^2 = Re^2(\hat{f}(u, v)) + Im^2(\hat{f}(u, v))$$

- Quelques propriétés de la TFD

- f réelle $\Rightarrow \text{Re}(\hat{f})$ paire et $\text{Im}(\hat{f})$ impaire

- Et $|\hat{f}(u, v)| = |\hat{f}(-u, -v)|$ module toujours paire
 $\Rightarrow$ Densité spectrale centro-symétrique

- f paire $\Rightarrow \text{Im}(\hat{f}) = 0$

- La TFD est circulaire (i.e. périodique de période M en u et de période N en v)

- La TFD inverse aussi (de période M en x et N en y)
 $\Rightarrow$ l'image obtenue par TFD inverse est périodisée !

- Ces deux derniers points : car les images et leurs TFD sont échantillonnées.

Échantillonnage dans le plan direct (respectivement dans le plan de Fourier) $\Leftrightarrow$ périodisation dans le plan de Fourier (respectivement dans le plan direct).

- Calculer la TFD sous Matlab/Octave : en fait calcul d'une FFT (*Fast Fourier Transform*) :

$\gg \hat{f} = \text{fft2}(f)$

- Mais la FFT produit un décalage de $(M/2, N/2) \Rightarrow$ *fftshift* sous Matlab/Octave pour réordonner le plan.

- Quand on va devoir placer l'image initiale dans un tableau plus grand pour le filtrage, on utilisera une syntaxe plus complète pour la FFT :

```
>>  $\hat{f}$  = fft2(f, P, Q)
```

où P et Q plus grands que M et N (*zero padding*).

- Module de $\hat{f}(u, v)$ obtenu avec la commande *abs* :

```
>> mod = abs( $\hat{f}$ )
```

- Visualiser $|FFT|$ correctement (*frequel* central au centre du tableau visualisé, pas dans les coins) :

```
>> imagesc(abs(fftshift(fft2(f))))
```

- Pour mieux voir (dynamique souvent limitée par rapport à la dynamique de la TF) :

```
>> imagesc((abs(fftshift(fft2(f))))^0.5)
```

- Opération inverse à *fftshift* : *ifftshift*, ainsi :

```
>>  $\hat{f}$  = ifftshift(fftshift( $\hat{f}$ ))
```

- Pour calculer la phase (ou argument) de $\hat{f}$: *atan2(Im(f),Re(f))*, qui est un tableau d'angles entre $-\pi$ et π . Autrement dit :

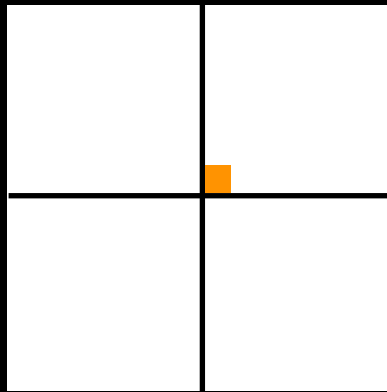
```
>> phi = atan2(imag(f_hat), real(f_hat))
```

ou plus directement avec *angle* (ou même *arg*) :

```
>> phi = angle(f_hat)
```

(Et on a donc : $\hat{f} = \text{abs}(\hat{f}) \cdot \exp(i * \text{angle}(\hat{f}))$)

- Attention, pour la TFD : le centre se situe ici $[M/2+1, N/2+1]$ si M et N sont paires :



(si nb impairs (M ou N) : $\text{floor}(M/2)+1, \text{floor}(N/2)+1$)

- La FFT inverse s'obtient grâce à *ifft2* :

```
>> f = ifft2(f_hat)
```

- Attention : *fft2* convertit au passage en classe « double »...

f de type *uint8* $\rightarrow$ $\hat{f}$ de type *double* (valeurs réelles en double précision, mais entre 0.0 et 255.0)

=> pour éviter les problèmes : convertir dès le début les images en réels double précision entre 0.0 et 1.0 !

- Unités dans le plan de Fourier :

- image de longueur $L=N \Delta x$, avec N le nb linéaire de pixels et Δx la taille du pixel

=> pas en fréquence $\Delta u = 1/L = 1/(N \Delta x)$, en u — et $\Delta v = 1/(N \Delta y)$ en v ... mais normalement $\Delta x=\Delta y$!

=> vecteur de fréquences spatiales u (ou v) :

$$u=1/(N\Delta x) [-N/2...0...N/2-1]$$

=> plus grande (haute!) fréquence spatiale disponible = $1/(N\Delta x) \times N/2 = 1/2 \ 1/\Delta x$



→
TFD



$$\Delta x=L/N$$
$$L = N \Delta x = N L/N$$

$$\Delta u=1/L$$
$$N \Delta u = N/L = 1/\Delta x$$

- Exemple sous Matlab :

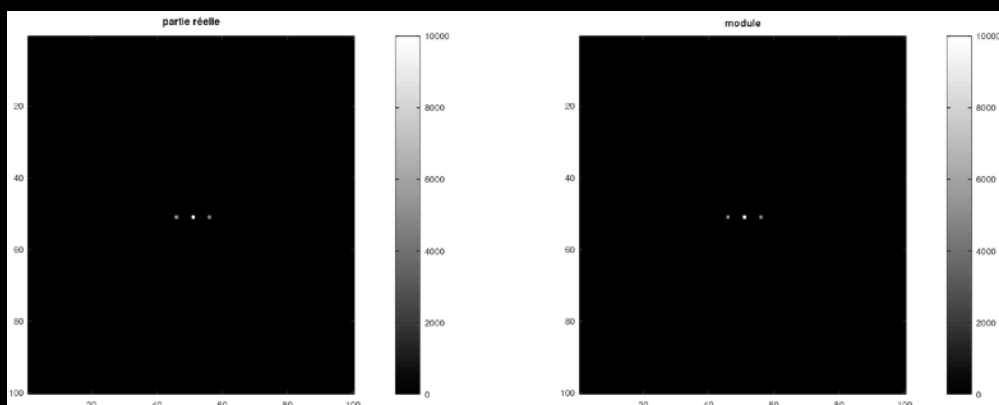
—> FFT bidimensionnelle d'une sinusoïde :

```
>> Tx=20;    —> période de 20 pixels (tôle ondulée)
>> x=0:99;
>> whos x    —> vecteur de 100 valeurs de 0 à 99
>> I = ones(100,1) * (1 + cos(2*pi*x/Tx));
>> imagesc(I), colorbar
```

(>> *improfile* sous Matlab, sinon un *plot* pour vérifier la sinusoïde selon l'axe des x)

```
>> Ichap = fft2(I);
>> whos Ichap
>> figure
>> subplot(1,2,1), imagesc(I), title('Sinusoïde'),
colorbar, axis('square')
>> subplot(1,2,2), imagesc(abs(fftshift(Ichap))), title('|
FFT(Sinusoïde)|'), colorbar, axis('square')
```

Remarque : ici, la fonction cosinus étant paire, la partie imaginaire de sa TF est nulle, et donc on aurait pu représenter de manière équivalente la partie réelle plutôt que le module.



- **Exercice 1** : Reprendre l'exemple ci-dessus. Décaler la sinusoïde de 7.5 pixels. Que se passe-t-il au niveau (du module) de la TF ?

```

1 clear
2 close all
3
4 % sinusoïde
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

```

```

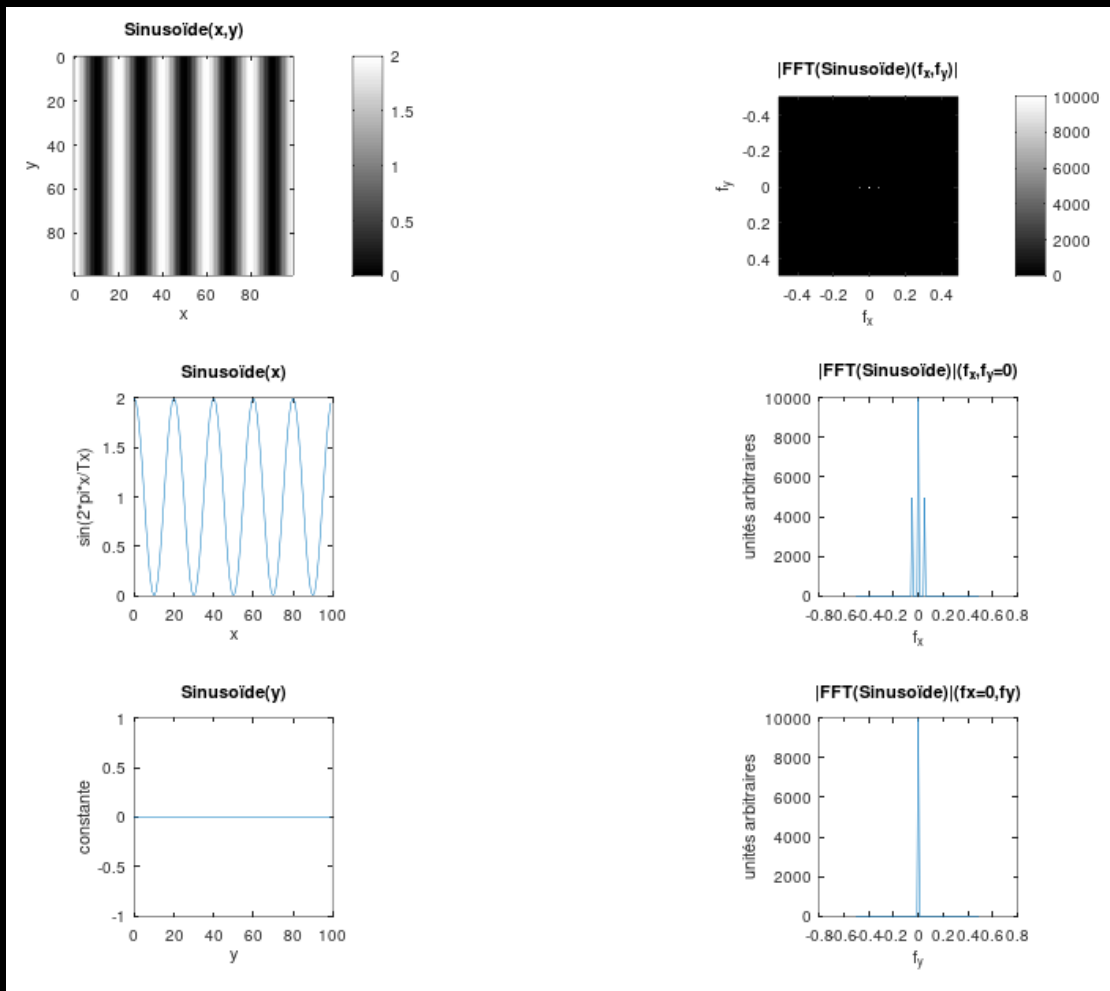
27 % FFT(sinusoïde)
28 Ichap=fft2(I); % FFT(image)
29 whos Ichap
30 Ichapmod=abs(fftshift(Ichap));
31 whos Ichapmod % module de FFT(image)
32
33 fx=1/(dim*Dx)*(-dim/2:dim/2-1); fy=fx;
34 % vecteurs des fréquences spatiales
35
36 subplot(3,2,2), imagesc(fx,fy,Ichapmod)
37 title('IFFT(Sinusoïde)(f_x,f_y)|'), 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('IFFT(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('IFFT(Sinusoïde)|f_x=0,f_y'), xlabel('f_y'), ylabel('unités arbitraires')
45

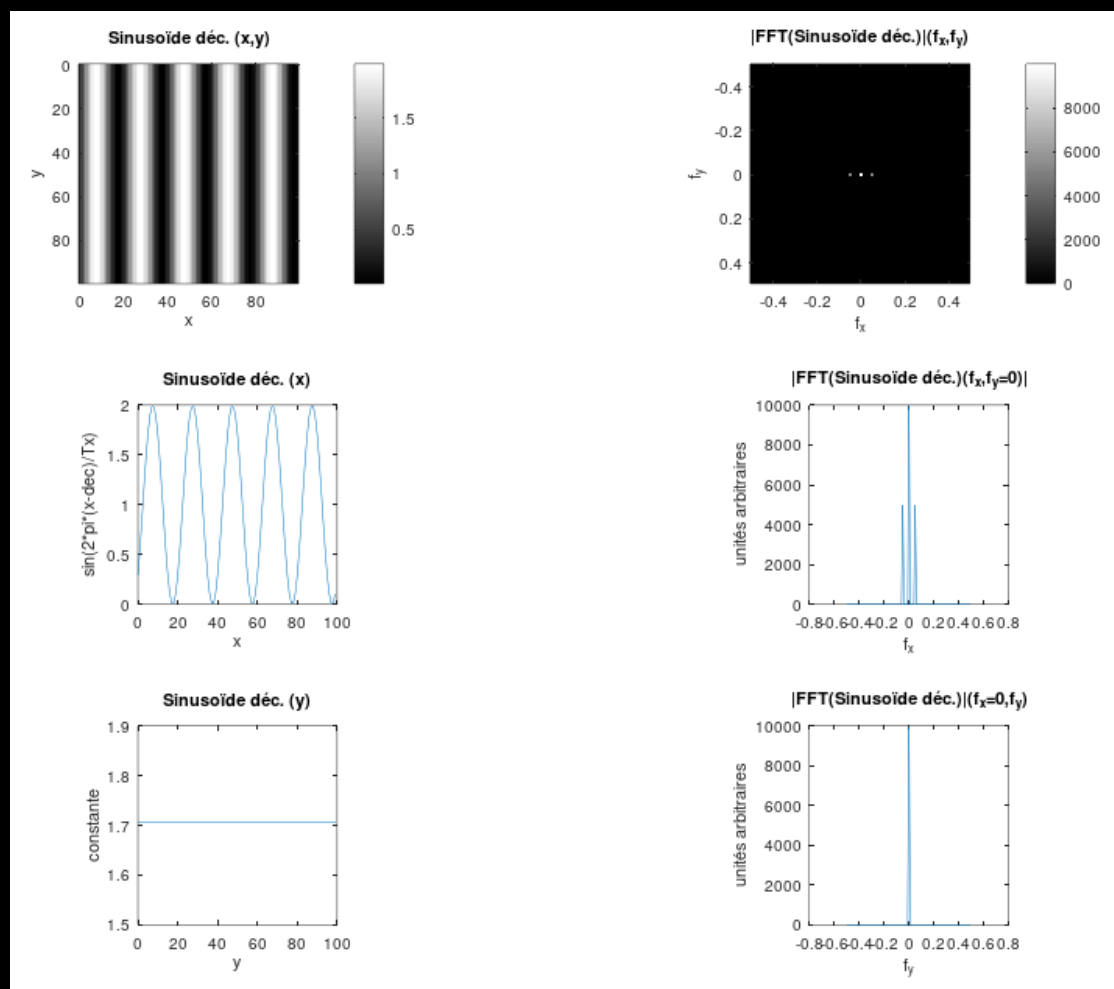
```

```

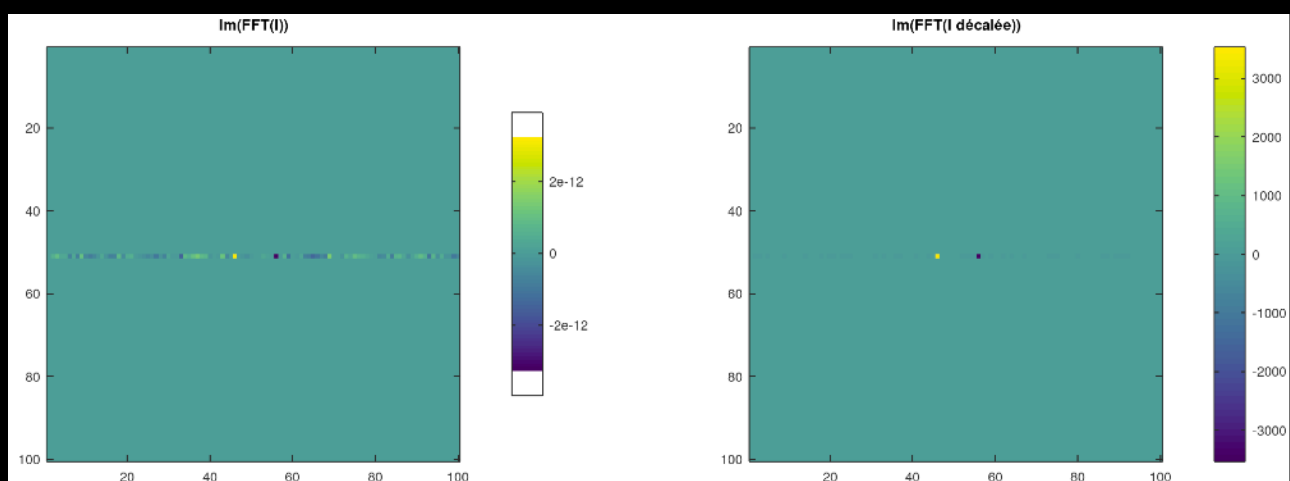
46 % cas de la sinusoïde décalée
47 dec=7.5;
48 Idec=ones(dim,1)*(1+cos(2*pi*(x-dec)/Tx));
49 Idecchap=fft2(Idec);
50 Idecchapmod=abs(fftshift(Idecchap));
51
52 figure(2), colormap('gray')
53
54 subplot(3,2,1), imagesc(x,y,Idec)
55 colorbar, axis('square')
56 title('Sinusoïde déc. (x,y)'), xlabel('x'), 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')

```





Remarque (sur les parties imaginaires) :



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)

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

Remarque 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 !

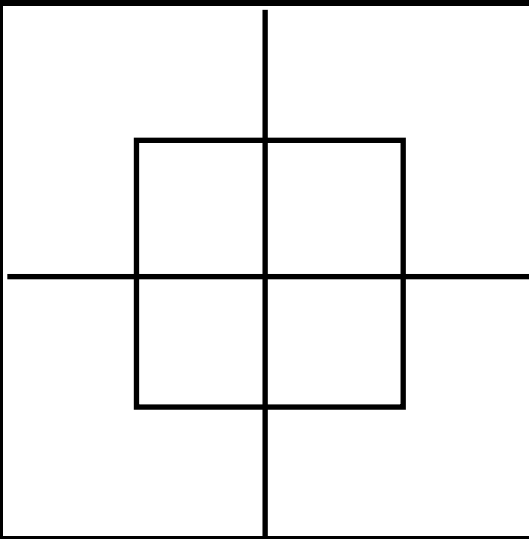
Remarque 2 : Les portes en x et y peuvent décrire le fait qu'une image est de taille $a \times b$ (en pixels).

Remarque 3 : $\Pi(x/a)$ = porte de largeur a en x.
sinc(u a) de 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.

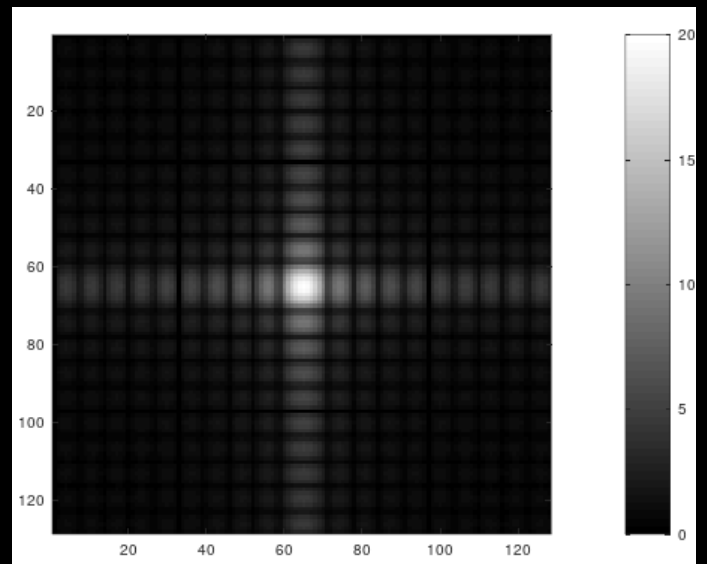
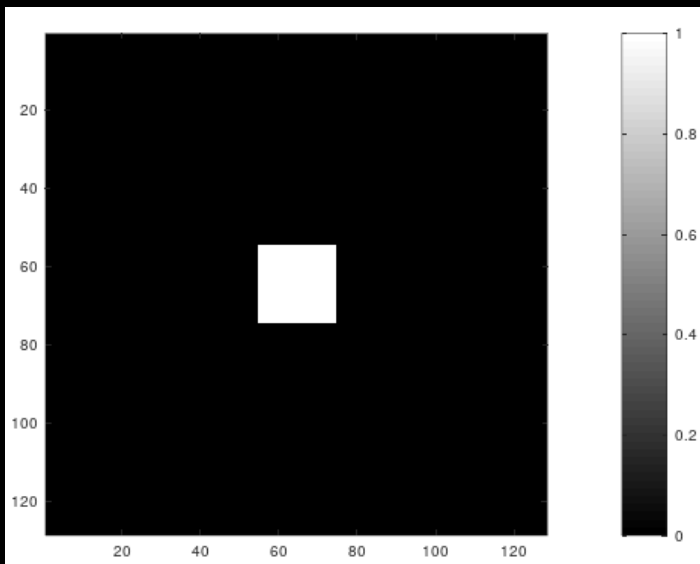
sous Matlab/Octave :

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



1 $\text{dim}/2$ $\text{dim}/2+1$ dim

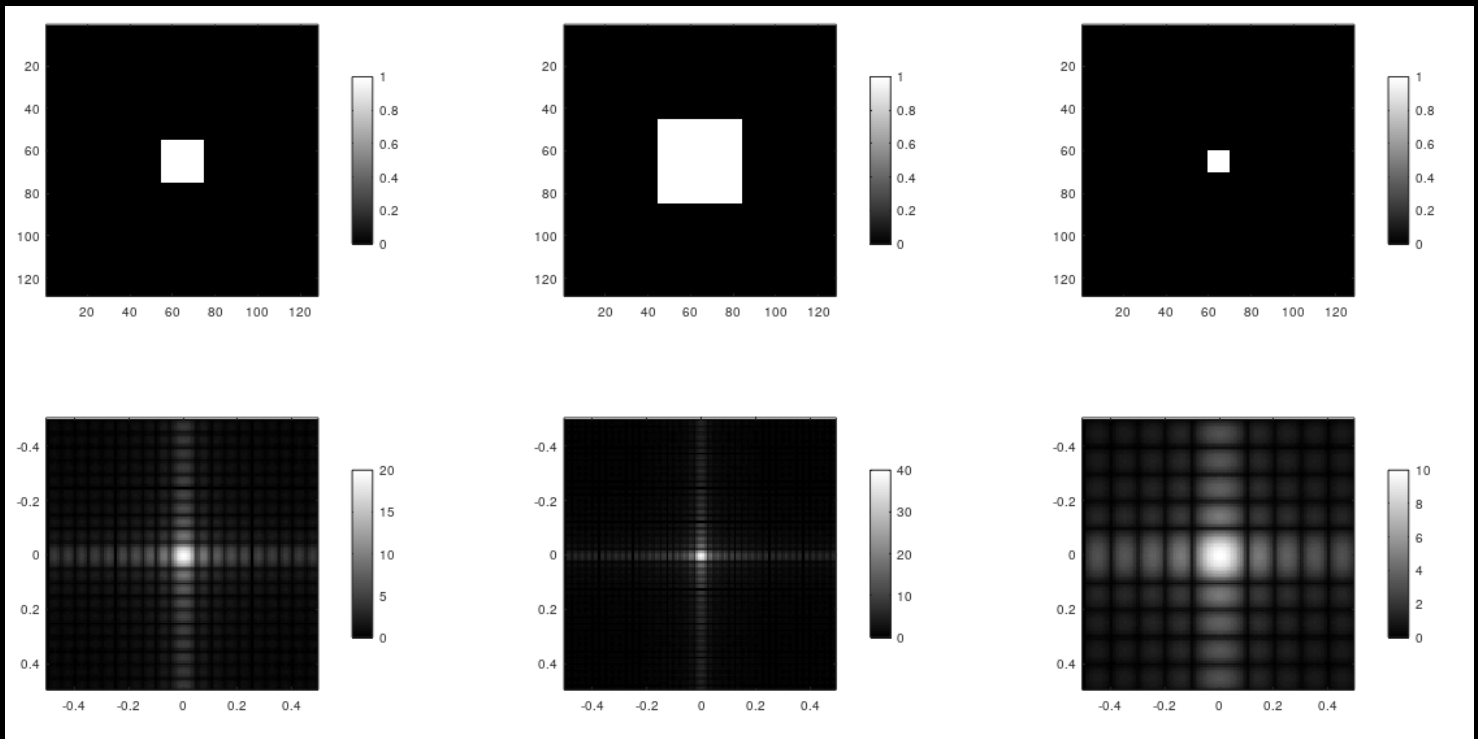
```
>> P(dim/2-a/2+1:dim/2+a/2,dim/2-b/2+1:dim/2+b/2)=1;
>> colormap(gray)
>> imagesc(P), axis('square')
>> Pchap=fft2(P);
>> imagesc((abs(fftshift(Pchap))).^0.5), colorbar
```



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(ffshift(ffshift2(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(ffshift(ffshift2(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(ffshift(ffshift2(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')
```



$$f_x = \frac{1}{N\Delta x} \left(-\frac{N}{2} \dots \frac{N}{2} - 1 \right)$$

et l'ensemble des
 fréquences :

$$- \frac{N}{2} \leq f_x \leq \frac{N}{2} - 1 \quad \times \frac{1}{N\Delta x}$$

$\neq$

index des px :

$$1 \leq n \leq N$$

—> 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 : Comme l'exercice 2 mais avec une Gaussienne...

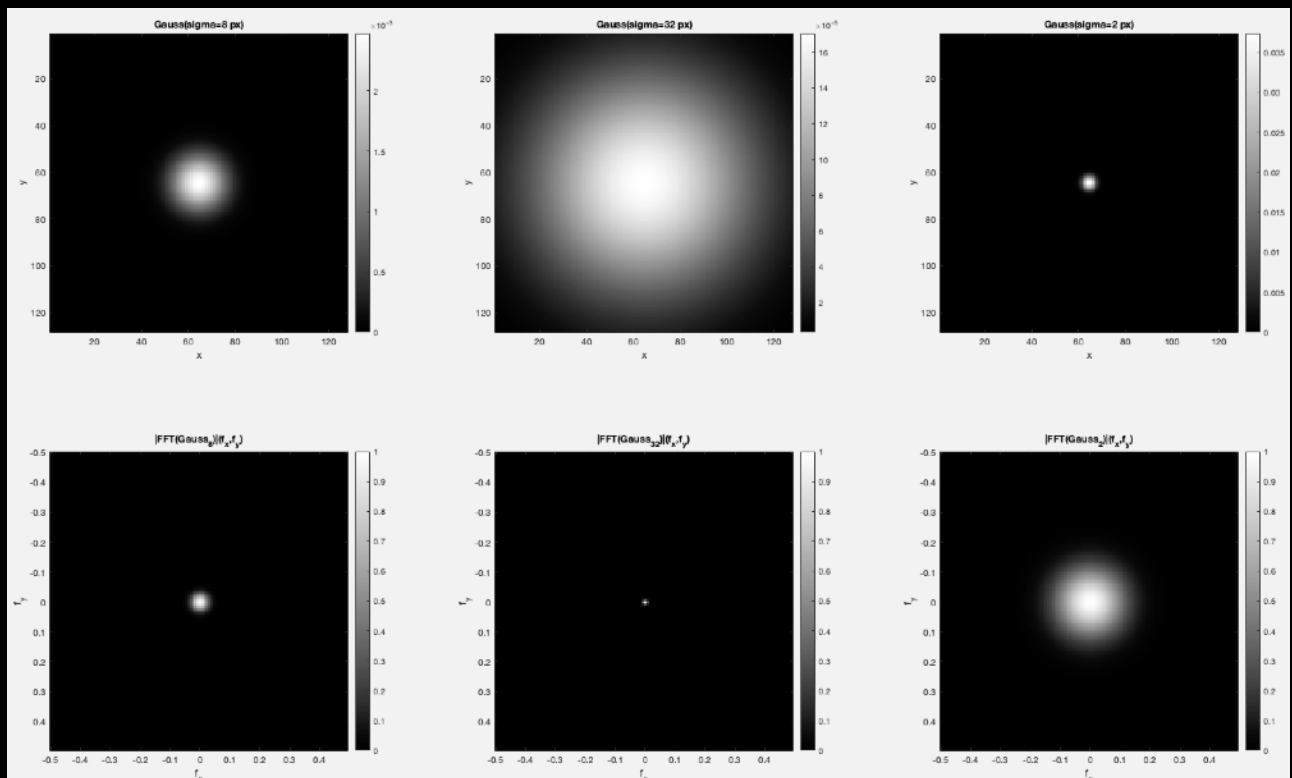
(En prenant trois valeurs de sigma et toujours en étalonnant correctement les fréquences spatiales.)

```
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('IFFT(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('IFFT(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('IFFT(Gauss_{2})|(f_x,f_y)'), xlabel('f_x'), ylabel('f_y')

```



—> Dirac :

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

—> Continu :

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

—> 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 $\Delta y = \Delta x = \text{taille du pixel}$)

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

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

(où $\cdot$ décrit le produit et $*$ le produit de convolution)

Or : $1/\Delta x = \text{largeur de toute la DFT}$

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

- Trois 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...

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

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

Remarque 2 : le module est inchangé.

—> Convolution

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

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

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 : $u=\text{ones}(\text{dim}, 1)*fx$, 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.