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)

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

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

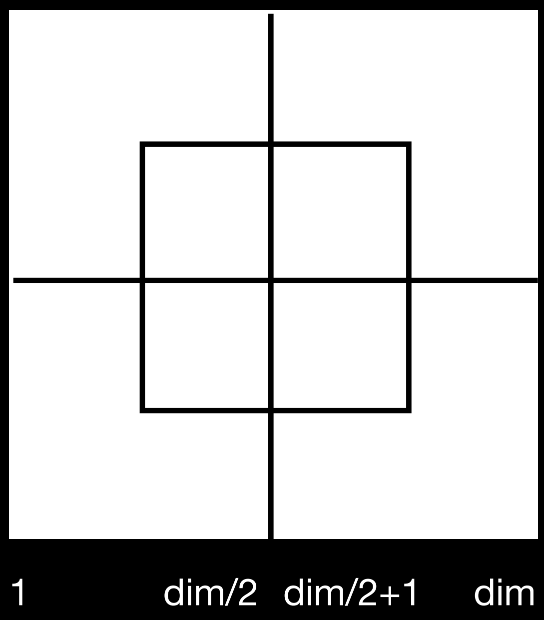
Remarque 2 : Les portes en x et y peuvent décrire le fait qu'une image est de taille *a x b* (en pixels).

Remarque 3 : ∏(x/a) = porte de largeur *a* en *x*.
sinc(u a) de largeur inversement prop. à *a*.

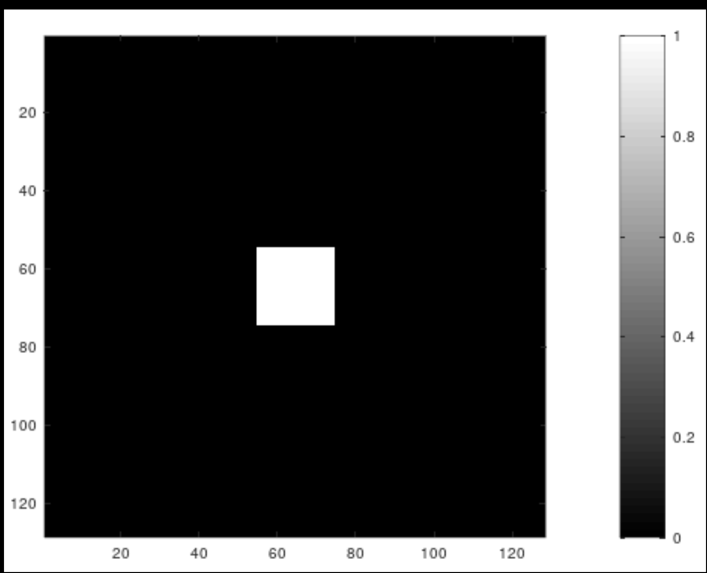
Quand *a* augmente, le pic du *sinc* augmente, mais sa largeur diminue => plus la porte est large, plus le *sinc* dans Fourier est étroit.

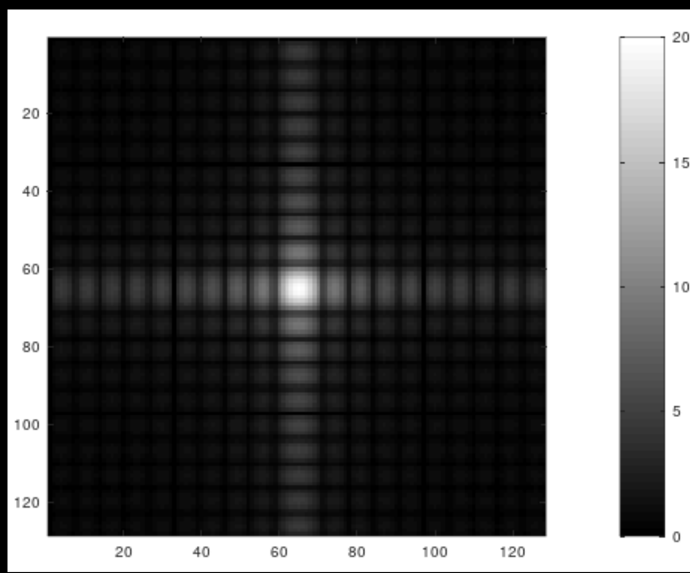
sous Matlab/Octave :

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




```
>>  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
```

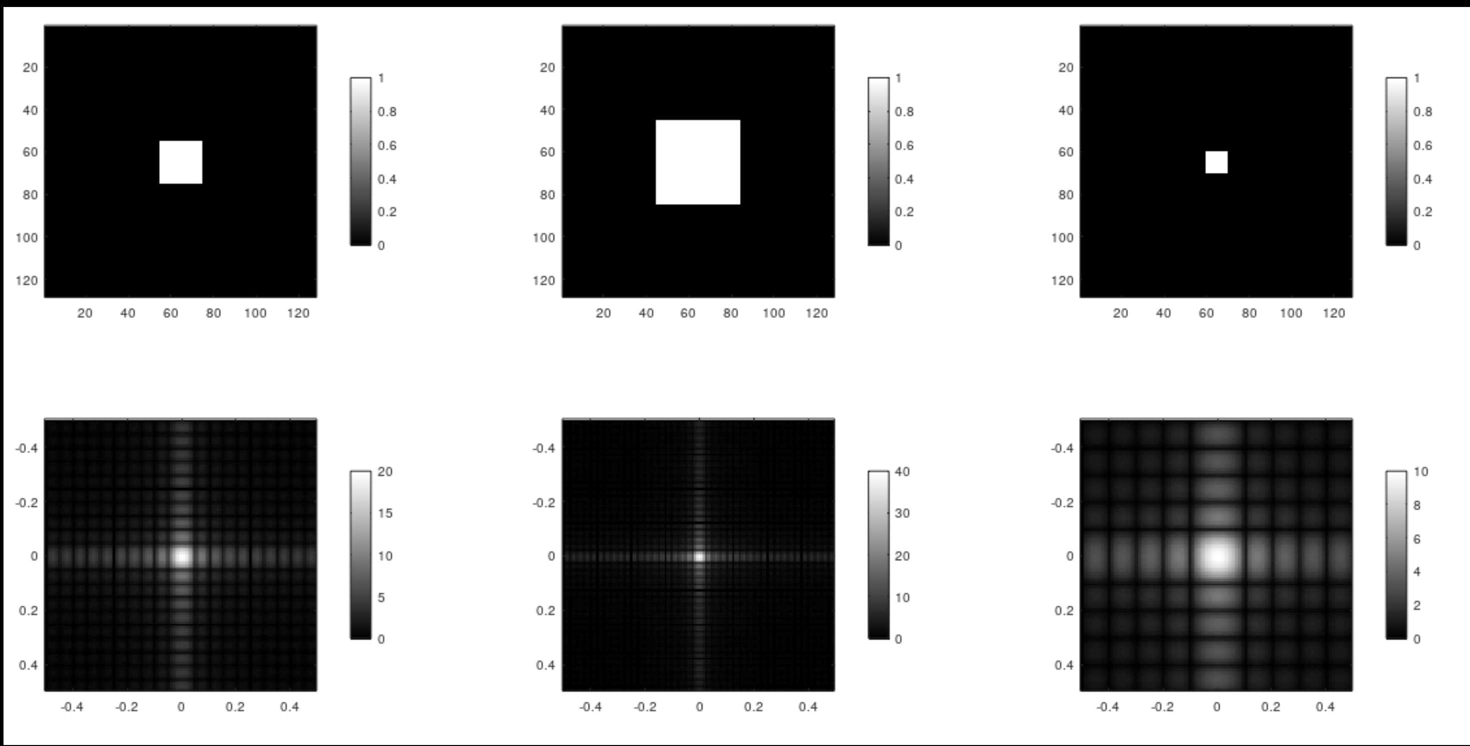


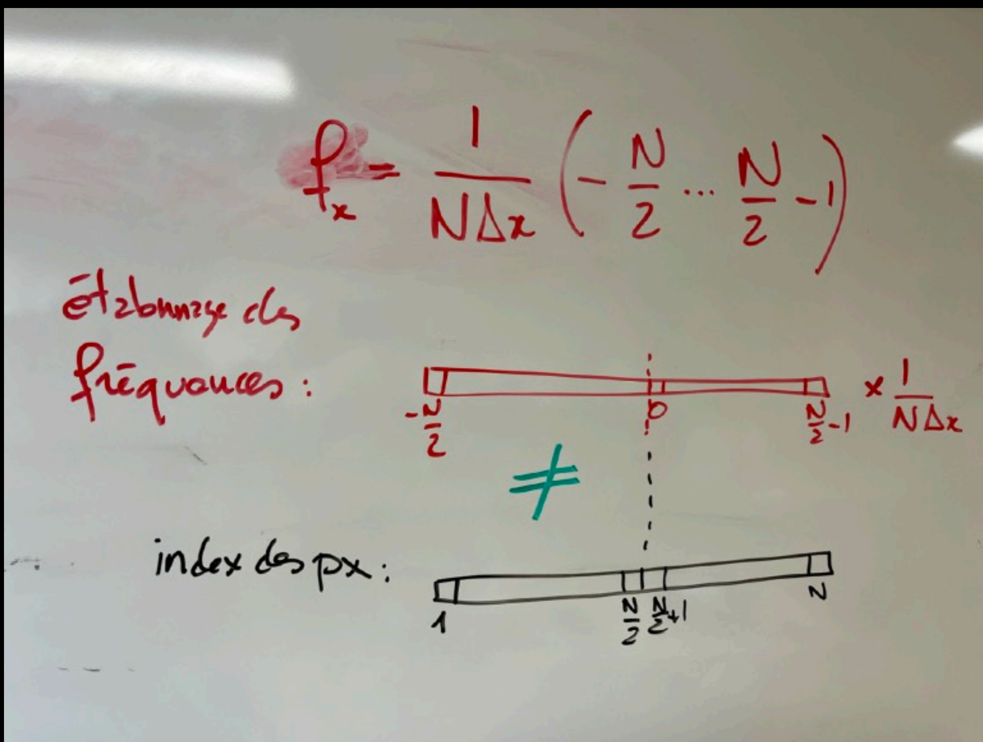


<u>Exercice 2</u> : Reprendre l'exemple précédent et augmenter/diminuer *a* et *b*. Que remarque-t-on ? (Prendre par exemple 3 valeurs : 20, 10, 40) (Mettre aussi les bonnes échelles de fréquences spatiales dans Fourier…)

```
clear
close all

dim=128; fx=1/(dim*1)*(-dim/2:dim/2-1); fy=fx;

% 1er exemple
P20=zeros(dim,dim); a=20; b=20;
P20(dim/2-a/2+1:dim/2+a/2, dim/2-b/2+1:dim/2+b/2)=1.0;
Pchapmod20=abs(fftshift(fft2(P20)));

% 2me exemple
P40=zeros(dim,dim); a=40; b=40;
P40(dim/2-a/2+1:dim/2+a/2, dim/2-b/2+1:dim/2+b/2)=1.0;
Pchapmod40=abs(fftshift(fft2(P40)));

% 3me exemple
P10=zeros(dim,dim); a=10; b=10;
P10(dim/2-a/2+1:dim/2+a/2, dim/2-b/2+1:dim/2+b/2)=1.0;
Pchapmod10=abs(fftshift(fft2(P10)));

```

```
% figure finale
figure, colormap('gray')

subplot(2,3,1), imagesc(P20)
colorbar, axis('square')
title('Porte(x/20,y/20)'), xlabel('x'), ylabel('y')

subplot(2,3,2), imagesc(P40)
colorbar, axis('square')
title('Porte(x/40,y/40)'), xlabel('x'), ylabel('y')

subplot(2,3,3), imagesc(P10)
colorbar, axis('square')
title('Porte(x/10,y/10)'), xlabel('x'), ylabel('y')

subplot(2,3,4), imagesc(fx,fy,Pchapmod20.^.5)
colorbar, axis('square')
title('sqrt(|FFT(Porte_{20})|(f_x,f_y))'), xlabel('f_x'), ylabel('f_y')

subplot(2,3,5), imagesc(fx,fy,Pchapmod40.^.5)
colorbar, axis('square')
title('sqrt(|FFT(Porte_{40})|(f_x,f_y))'), xlabel('f_x'), ylabel('f_y')

subplot(2,3,6), imagesc(fx,fy,Pchapmod10.^.5)
colorbar, axis('square')
title('sqrt(|FFT(Porte_{10})|(f_x,f_y))'), xlabel('f_x'), ylabel('f_y')
```





$f_x = \frac{1}{N\Delta x}\left(-\frac{N}{2} \cdots \frac{N}{2}-1\right)$
étalonnage des fréquences :
$-\frac{N}{2}$
0
$\frac{N}{2}-1$
$\times \frac{1}{N\Delta x}$
≠
index des px :
1
$\frac{N}{2}$
$\frac{N}{2}+1$
N

—> Gaussienne bidimensionnelle

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

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

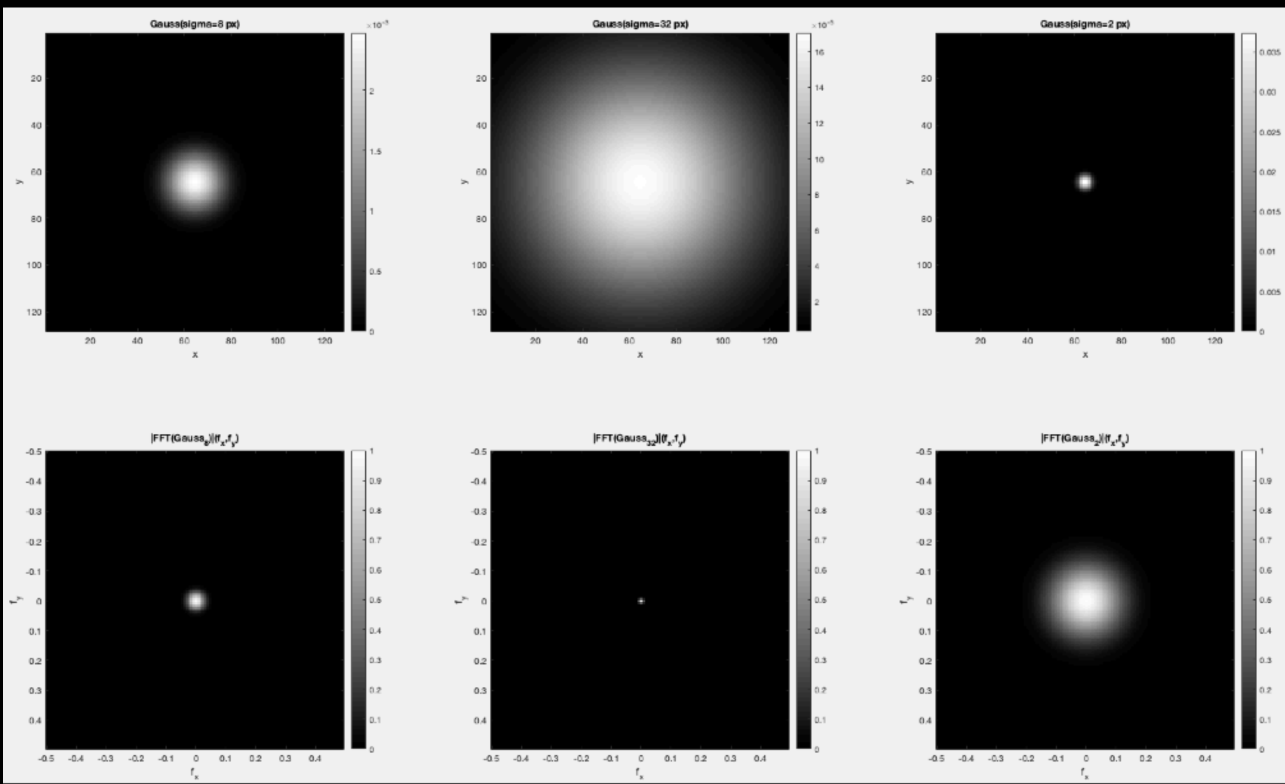
sous Matlab/Octave :

>> 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.)

```
clear
close all
pkg load image

dim=128; fx=1/(dim*1)*(-dim/2:dim/2-1); fy=fx;

% 1er exemple
a=8;
G8=fspecial('gaussian',dim,a);
Gchapmod8=abs(fftshift(fft2(G8)));

% 2me exemple
a=32;
G32=fspecial('gaussian',dim,a);
Gchapmod32=abs(fftshift(fft2(G32)));

% 3me exemple
a=2;
G2=fspecial('gaussian',dim,a);
Gchapmod2=abs(fftshift(fft2(G2)));
```

```matlab
% figure finale
figure, colormap('gray')

subplot(2,3,1), imagesc(G8)
colorbar, axis('square')
title('Gauss(sigma=8 px)'), xlabel('x'), ylabel('y')

subplot(2,3,2), imagesc(G32)
colorbar, axis('square')
title('Gauss(sigma=32 px)'), xlabel('x'), ylabel('y')

subplot(2,3,3), imagesc(G2)
colorbar, axis('square')
title('Gauss(sigma=2 px)'), xlabel('x'), ylabel('y')

subplot(2,3,4), imagesc(fx,fy,Gchapmod8)
colorbar, axis('square')
title('|FFT(Gauss_{8})|(f_x,f_y)'), xlabel('f_x'), ylabel('f_y')

subplot(2,3,5), imagesc(fx,fy,Gchapmod32)
colorbar, axis('square')
title('|FFT(Gauss_{32})|(f_x,f_y)'), xlabel('f_x'), ylabel('f_y')

subplot(2,3,6), imagesc(fx,fy,Gchapmod2)
colorbar, axis('square')
title('|FFT(Gauss_{2})|(f_x,f_y)'), xlabel('f_x'), ylabel('f_y')
```



—> Dirac :

$\partial(x,y)$ —TF—> $\mathbf{1}(u,v)$

—> Continu :

$a\,\mathbf{1}(x,y)$ —TF—> $a\,\partial(u,v)$

—> Peigne de Dirac :

$\underline{III}_{(a)}(x) \,.\, \underline{III}_{(b)}(y)$ —TF—> $\underline{III}_{(1/a)}(u) \,.\, \underline{III}_{(1/b)}(v)$

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

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

=> image discrète (x,y) = image continue (x,y)
• $(\, \underline{III}_{(\Delta x)}(x) \,.\, \underline{III}_{(\Delta x)}(y) \,)$

=> *DFT*(img discrète) (u,v) = *DFT*(img continue) (u,v)
$*\ (\, \underline{III}_{(1/\Delta x)}(u) \,.\, \underline{III}_{(1/\Delta x)}(v) \,)$

(où • décrit le produit et * le produit de convolution)

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

Donc, on a : échantillonnage dans le plan direct => 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

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

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).

<u>Exercice 4</u> : 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 :

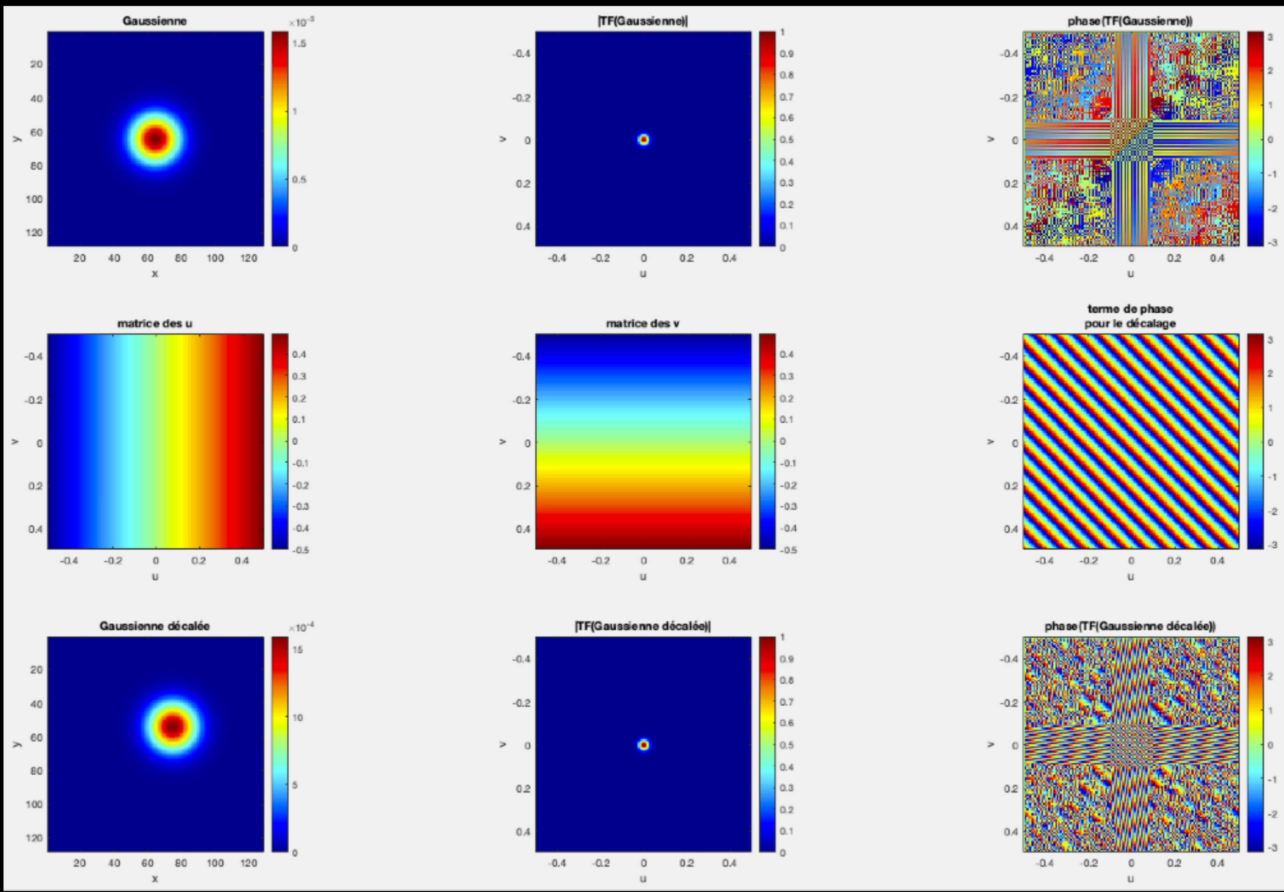
(1) création de la Gaussienne (plan direct)
(2) calculer le terme de phase (plan de Fourier)
(3) appliquer le terme de phase (plan de Fourier)
(4) revenir dans le plan direct

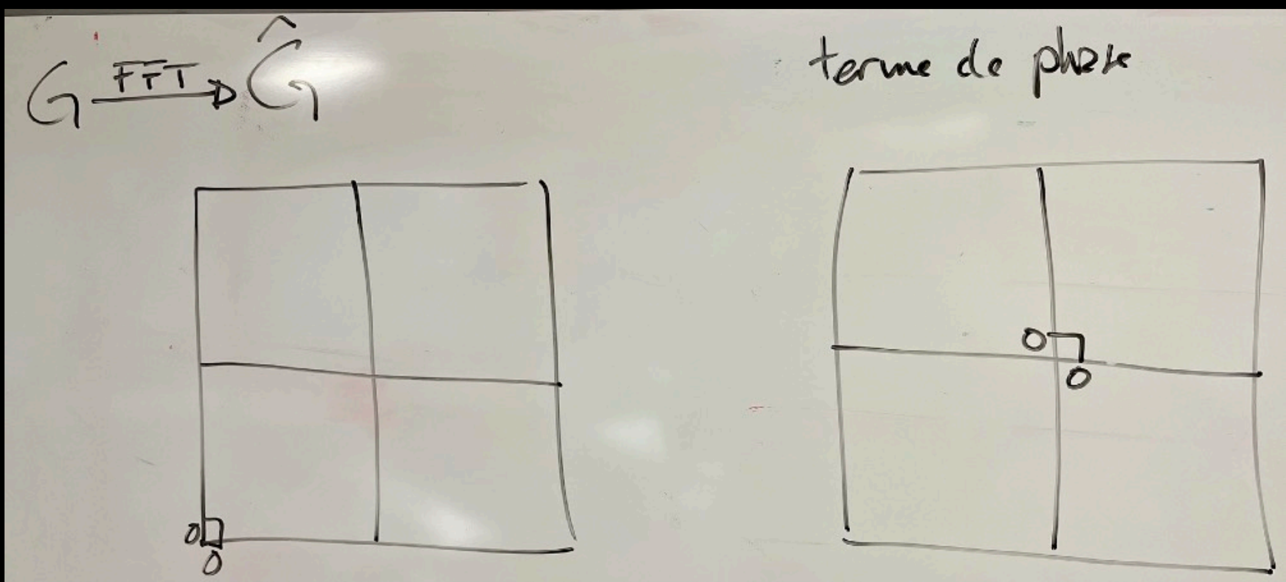
L'étape (2) revient à écrire correctement le terme de phase *exp(-2 i π (xo u + yo v)* dans Fourier… terme qui doit être <u>un tableau</u> de mêmes dimensions que l'image de départ (et donc sa TF et donc la phase de cette TF). Ici *xo* et *yo* sont les décalages respectivement en *x* et en *y*, et ce 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=ones(dim, 1)*fx*, où : *fx=1/(dim*1)*(-dim/2:dim/2-1)*.

Attention : on est ici dans le plan de la **<u>FFT</u>**, 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.

```
% préliminaires
clear
close all
%pkg load image

dim=128;                                % taille des images
fx=1/(dim*1)*(-dim/2:dim/2-1); fy=fx; % vecteur des fréquences :
                                        % 1/(N Δx)*(-N/2 .. 0 .. N/2-1)
                                        % on peut aussi écrire :
                                        % fx=((0:dim-1)-dim/2)/dim; fy=fx;
% Gaussienne
a=10;
G=fspecial('gaussian',dim,a);           % Gaussienne
Gchap = fft2(G);                        % TF(Gaussienne)

% terme de phase
decx= 10.4;                             % décalage en x
decy=-10.4;                             % décalage en y
uo=ones(dim,1)*fx;                      % matrice des u (par colonne)
vo=uo';                                 % matrice des v (par ligne)
u=fftshift(uo);                         % car le plan de la FFT est "shifté"
v=fftshift(vo);                         % d'un demi-tableau en u et en v
decfou=exp(-complex(0,1)*2*pi*(u*decx+v*decy));
                                        % TF(décalage) ; complex(0,1)=i
% décalage
Gdecchap = Gchap.*decfou;               % TF(Gaussienne décalée)
Gdec=real(ifft2(Gdecchap));             % retour dans le plan direct
```

```
% figure finale
figure(1), colormap('jet')              % colormap "jet" (voir help colormap)

subplot(3,3,1), imagesc(G)
colorbar, axis('square')
title('Gaussienne'), xlabel('x'), ylabel('y')

subplot(3,3,2), imagesc(fx, fy, abs(fftshift(Gchap)))
colorbar, axis('square')
title('|TF(Gaussienne)|'), xlabel('u'), ylabel('v')

subplot(3,3,3), imagesc(fx, fy, angle(fftshift(Gchap)))
colorbar, axis('square')
title('phase(TF(Gaussienne))'), xlabel('u'), ylabel('v')

subplot(3,3,4), imagesc(fx, fy, uo)
colorbar, axis('square')
title('matrice des u'), xlabel('u'), ylabel('v')

subplot(3,3,5), imagesc(fx, fy, vo)
colorbar, axis('square')
title('matrice des v'), xlabel('u'), ylabel('v')

subplot(3,3,6), imagesc(fx, fy, angle(fftshift(decfou)))
colorbar, axis('square')
title({'terme de phase'; 'pour le décalage'}), xlabel('u'), ylabel('v')

subplot(3,3,7), imagesc(Gdec)
colorbar, axis('square')
title('Gaussienne décalée'), xlabel('x'), ylabel('y')

subplot(3,3,8), imagesc(fx, fy, abs(fftshift(Gdecchap)))
colorbar, axis('square')
title('|TF(Gaussienne décalée)|'), xlabel('u'), ylabel('v')

subplot(3,3,9), imagesc(fx, fy, angle(fftshift(Gdecchap)))
colorbar, axis('square')
title('phase(TF(Gaussienne décalée))'), xlabel('u'), ylabel('v')
```


Gaussienne
|TF(Gaussienne)|
phase(TF(Gaussienne))
matrice des u
matrice des v
terme de phase
pour le décalage
Gaussienne décalée
|TF(Gaussienne décalée)|
phase(TF(Gaussienne décalée))
x
y
u
v


G —FFT→ Ĝ
terme de phase
0
0
0
0