

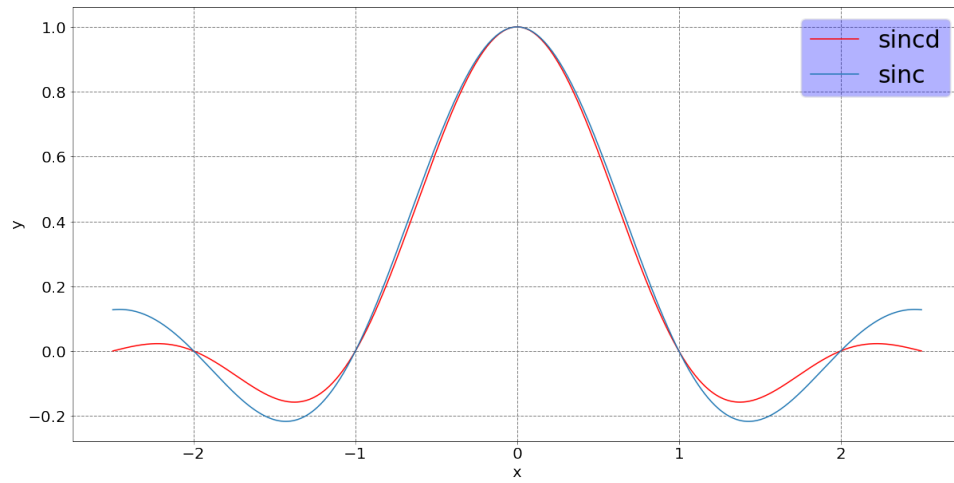
---

# COMPTE-RENDU TP3

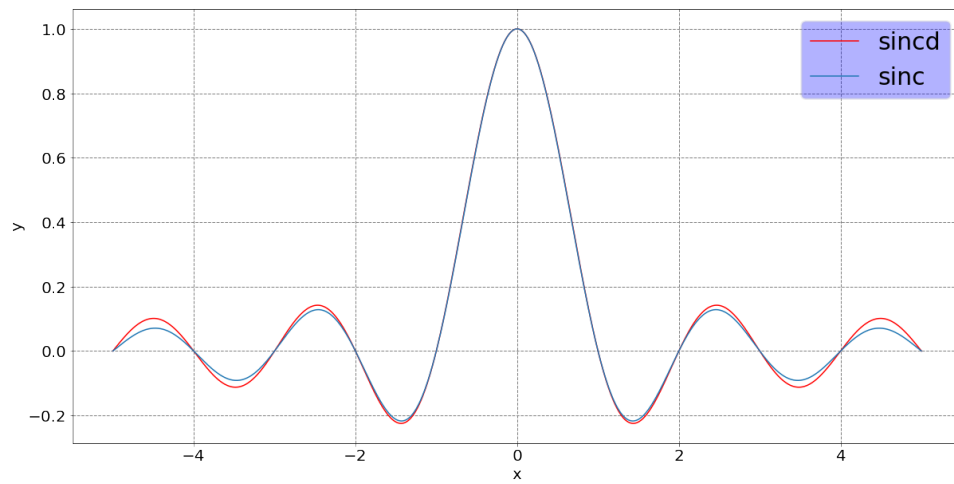
Vincent Matthys  
vincent.matthys@ens-paris-saclay.fr

---

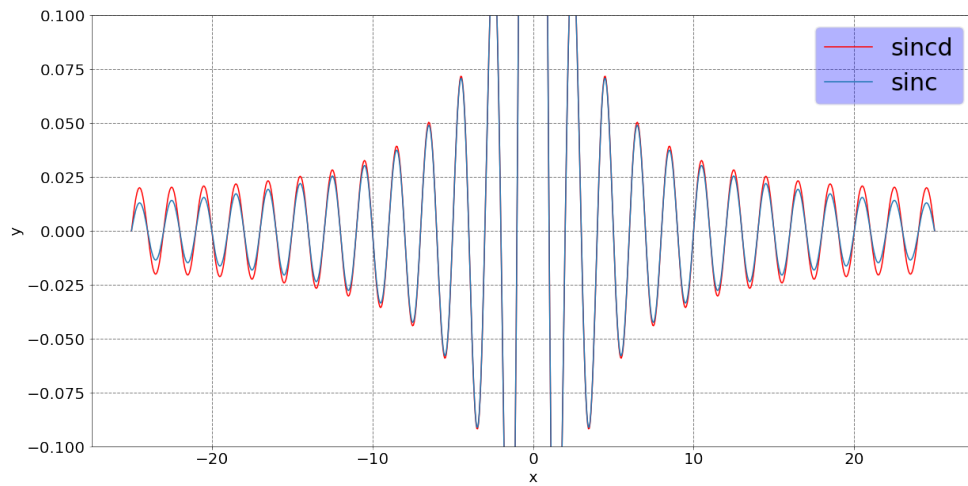
## 6 Exercice 6



(a) Pour  $N = 5$



(b) Pour  $N = 10$



(c) Pour  $N = 50$

FIGURE 1 – Représentation graphique de la convergence du  $\text{sincd}_N$  et du  $\text{sinc}$ , sur  $[-N/2, N/2]$

En figure 1 sont représentés le  $\text{sinc}$  et le  $\text{sincd}_N$ , pour différentes valeurs de  $N$  pour  $x \in [-N/2, N/2]$ . On constate que, pour chaque valeur entière, on a bien égalité, et que,

pour  $N$  assez grand, les différences sont minimales.

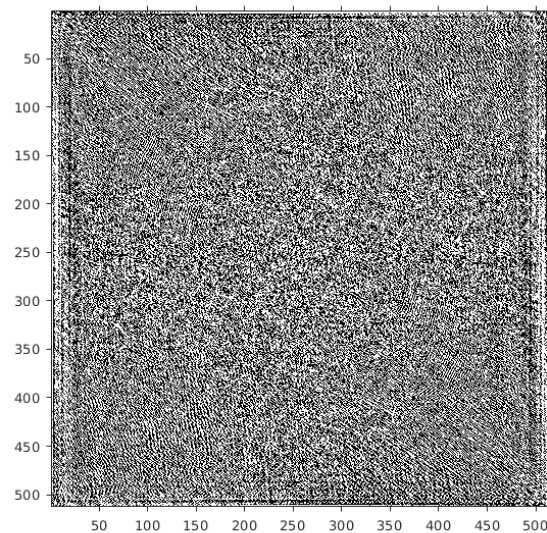
## 7 Exercice 7

```
u = double(imread('lena.pgm'));
```



(a) Image lena

```
f = fft2(u);  
imshow(f, 'InitialMagnification', 100);
```

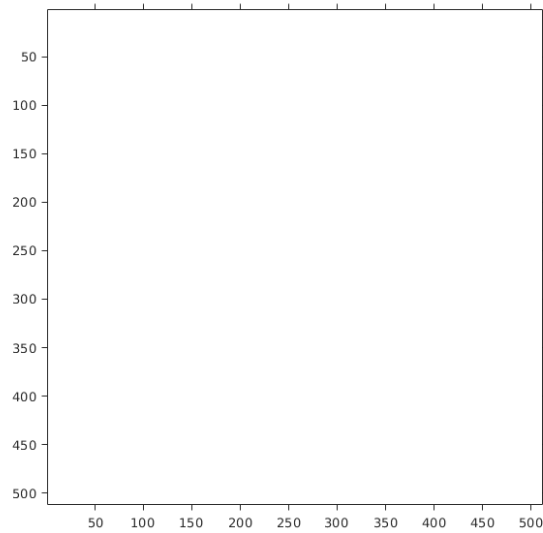


(b) Partie réelle de la transformée de Fourier de lena

FIGURE 2 – Partie réelle de la transformée de Fourier de Lena

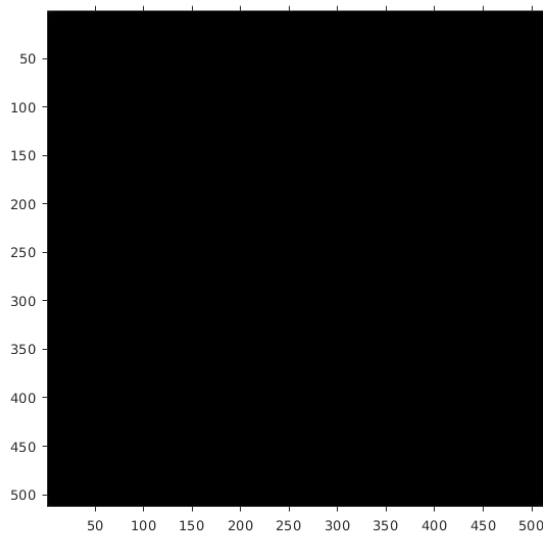
En figure 2, on demande à Matlab d'afficher en niveaux de gris une matrice complexe, résultant en un message d'erreur *Warning : Displaying real part of complex input..* On observe donc uniquement la partie réelle de la transformée de Fourier.

```
imshow(abs(f));
```



(a) Sans l'option [] de imshow

```
imshow(abs(f),[]);
```



(b) Avec l'option [] de imshow

FIGURE 3 – Représentations du module de la transformée de Fourier

En figure 3, la commande 3a est équivalent à :

```
imshow(abs(f) > 1);
```

et on n'obtient que des pixels blancs, puisque `imshow`, quand on lui présente un double (ce qui est le type de `abs(f)`), représente en noir les valeurs à 0, en blanc les valeurs à 1, et en niveaux de gris les valeurs entre ces bornes. Or, on vérifie que `abs(1) > 1` ne contient que des 1. En revance, avec l'option `[ ]`, on rescale  $abs(f) \in [0, 1]$ , avec la plus grande valeur en blanc, et la plus faible en noire. Or, la répartition des valeurs est très asymétrique, et on ne représente quasiment que du noir. On peut s'en convaincre en tracant le logarithme de cette image, qui donne la figure 4.

```
imshow(log(abs(f)), [ ]);
```

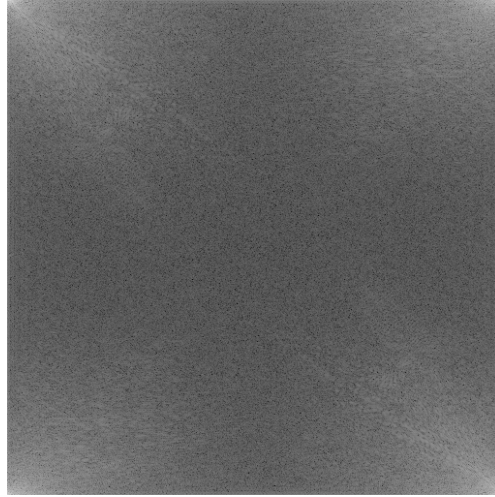
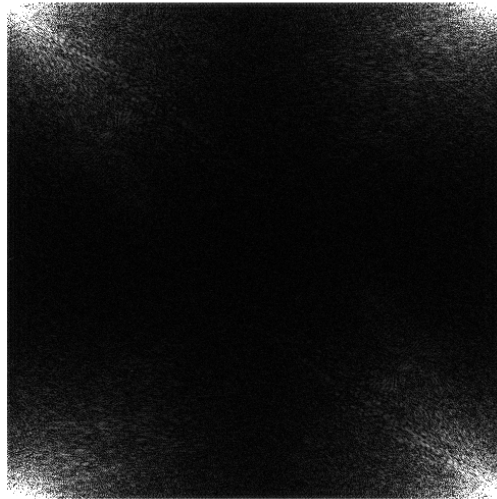


FIGURE 4 – Logarithme de la figure 3b

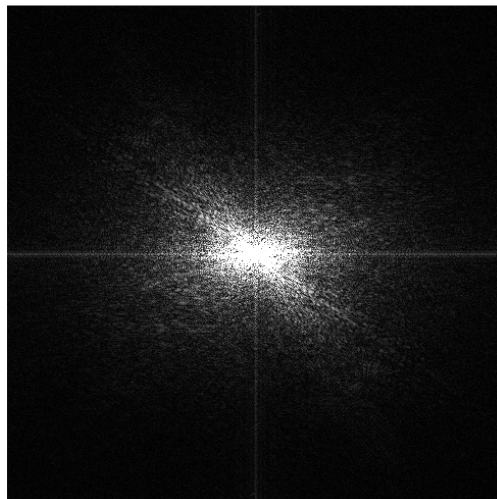
En figure 5, on sature 1 % des valeurs extrémales du module de la transformée, et on redistribue linéairement les valeurs restantes sur tous les niveaux de gris. C'est ce que fait la fonction *normsat*. La fonction *fftshift* permet de centrer l'origine des fréquences, passant du coin supérieur gauche en figure 5a au centre en figure 5b.

```
imshow(normsat(abs(f),1));
```



(a) Avec l'origine des fréquences en haut à gauche de l'image

```
imshow(normsat(fftshift(abs(f)),1));
```



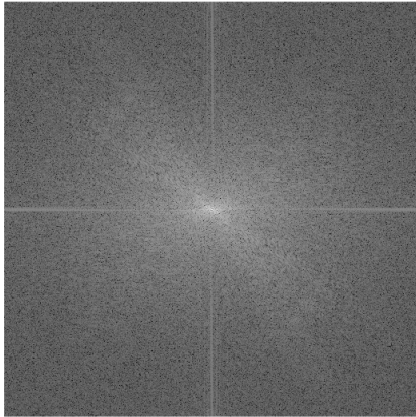
(b) Avec l'origine des fréquences au milieu de l'image

FIGURE 5 – Visualisation du module de la transformée de Fourier après saturation à 1 % des valeurs extrémales et mapping linéaire des valeurs restantes en niveau de gris

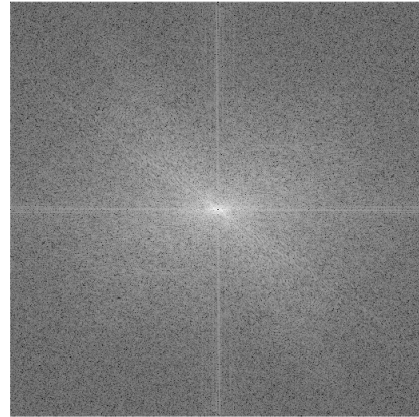
En figure 6d sont présentés les commandes et leur résultat pour visualiser efficacement différentes parties d'une transformée de Fourier.

```
imshow(log(1 +abs(real(fftshift(f)
))), [])
```

```
imshow(log(1 +abs(imag(fftshift(f)
))), [])
```

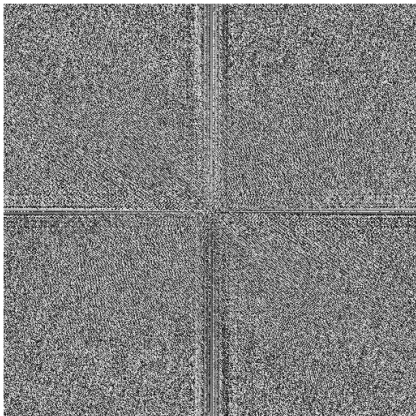


(a) Logarithme de la partie réelle



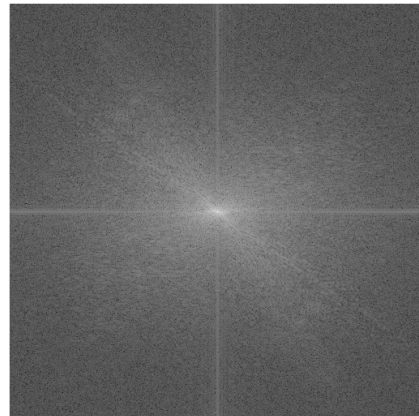
(b) Logarithme de la partie imaginaire

```
imshow(angle(fftshift(f)) + pi,
[])
```



(c) Phase

```
imshow(log(1 +abs(fftshift(f))),
[])
```



(d) Logarithme du module

FIGURE 6 – Visualisation des divers parties d'une transformée de Fourier

## 8 Exercice 8