

python-spect

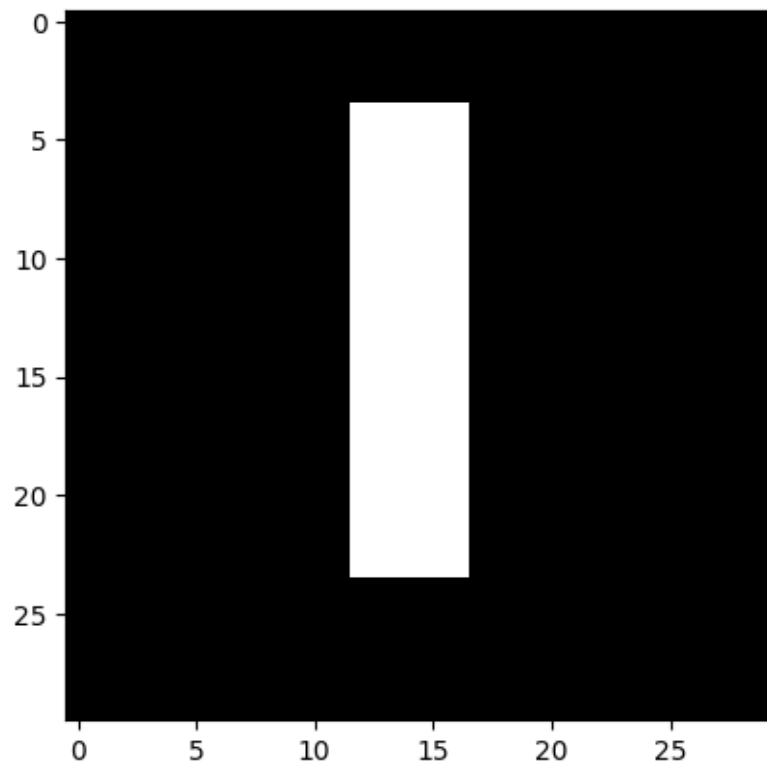
September 4, 2024

0.1 spect.m

```
[ ]: import numpy as np
import matplotlib.pyplot as plt
```

```
[ ]: # Prepare image
f = np.zeros((30, 30))
f[4:24, 12:17] = 1 # Note: Python uses 0-based indexing so 5:24 converts into
↳ 4:24 for example
plt.figure()
plt.imshow(f, cmap='gray')
```

```
[ ]: <matplotlib.image.AxesImage at 0x701e3c0dd420>
```



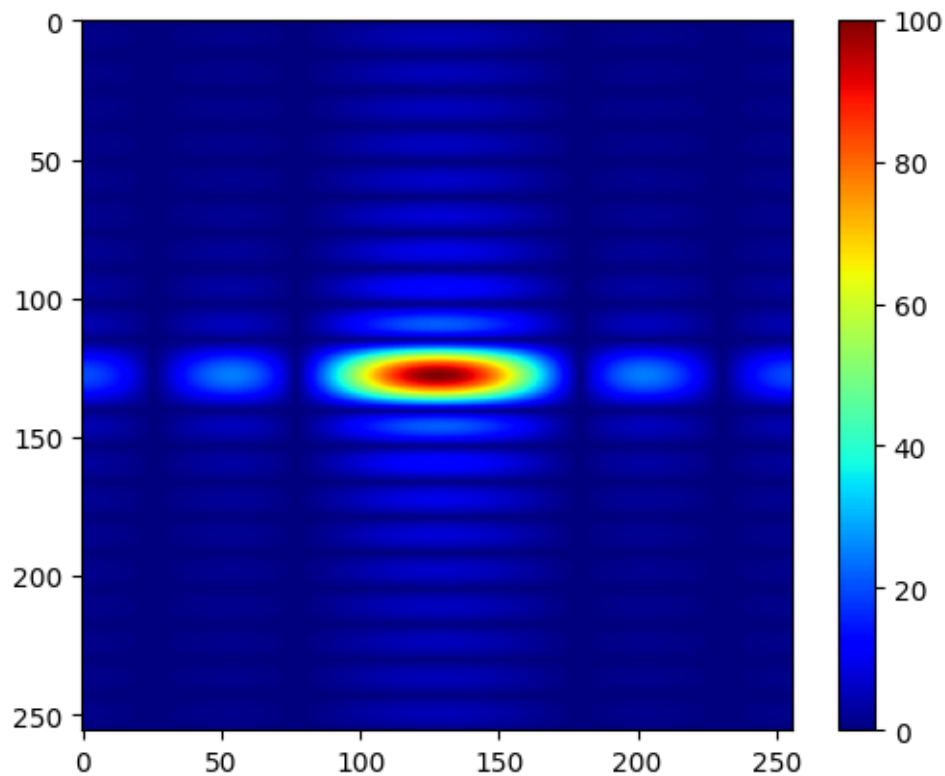
```
[ ]: # Compute Fourier Transform
F = np.fft.fft2(f, (256, 256))
F = np.fft.fftshift(F) # Center FFT
```

```
[ ]: # Measure the minimum and maximum value of the transform amplitude
print(f"Min amplitude: {np.min(np.abs(F))}")
print(f"Max amplitude: {np.max(np.abs(F))}")
```

Min amplitude: 0.0
Max amplitude: 100.0

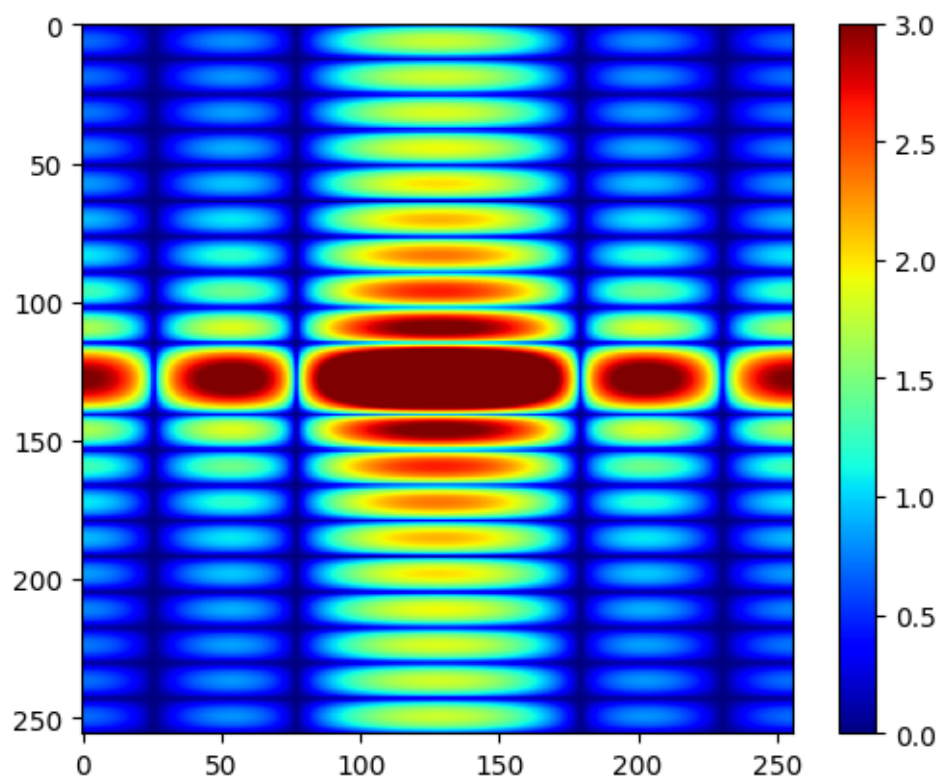
```
[ ]: # Show amplitude
plt.figure()
plt.imshow(np.abs(F), cmap='jet', vmin=0, vmax=100)
plt.colorbar()
```

```
[ ]: <matplotlib.colorbar.Colorbar at 0x701e3373f1c0>
```



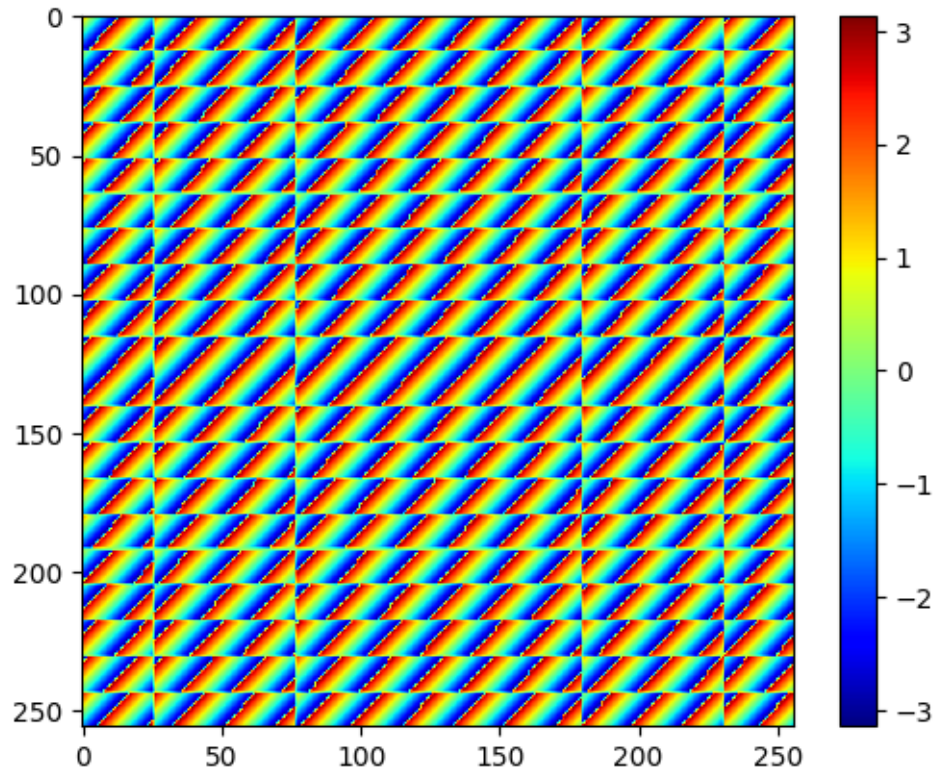
```
[ ]: # Show log amplitude
plt.figure()
plt.imshow(np.log1p(np.abs(F)), cmap='jet', vmin=0, vmax=3)
plt.colorbar()
```

```
[ ]: <matplotlib.colorbar.Colorbar at 0x701e3150fd90>
```



```
[ ]: # Show phases
plt.figure()
plt.imshow(np.angle(F), cmap='jet', vmin=-np.pi, vmax=np.pi)
plt.colorbar()
```

```
[ ]: <matplotlib.colorbar.Colorbar at 0x701e315d3dc0>
```



Lo que se analiza es una simple imagen de un rectangulo vertical blanco en el centro de un cuadrado negro a la cual se le calcula la FFT y se grafica el modulo y el logaritmo de la misma.

Para explicar lo que sucede al graficar el espectro del módulo se debe primero recorrer la imagen original de forma horizontal y se puede apreciar un pulso de duración 4 en la zona blanca que si le hacemos la FFT es una sinc. Luego lo recorremos de forma vertical para ver otro pulso ahora de duración 19 por lo que el ancho del lóbulo principal de la sinc será mas angosto en esta dirección.

Luego se plotea el logaritmo de la FFT para poder conseguir mejor rango dinámico y poder diferenciar aún más los cambios.