

РОБОТА із ЗОБРАЖЕННЯМИ

Файл: Image_07_005

Частина 2. Фільтрація 2D сигналу (зображення) у частотному просторі

Використовуємо бібліотеку [SCIPY](#)

```
%matplotlib inline
```

```
import matplotlib.pyplot as plt
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline

import numpy as np
# Імпорт функцій прямого та зворотнього перетворення Фурє + функція центрування спектру
from scipy.fftpack import fft2, ifft2, fftshift
import skimage.io as io
# Імпорт функцій роботи із зображеннями
from skimage import img_as_float # Перетворення зображення ф формат FLOAT
from skimage.color import rgb2gray # рьчислення компоненти яскравості з RGB зображення
from skimage.io import imread
from skimage.data import astronaut # імпорт стандартного зображення (512, 512, 3)
uint8 ndarray
from skimage.filters import window
```

```
C:\ProgramData\Anaconda3\lib\site-packages\paramiko\transport.py:219:
CryptographDeprecationWarning: Blowfish has been deprecated
"__class__": algorithms.Blowfish,
```

ПРЯМЕ та ЗВОРОТНЄ 2D ПЕРЕТВОРЕННЯ ФУР'Є

```
# Завантажуємо Зображення

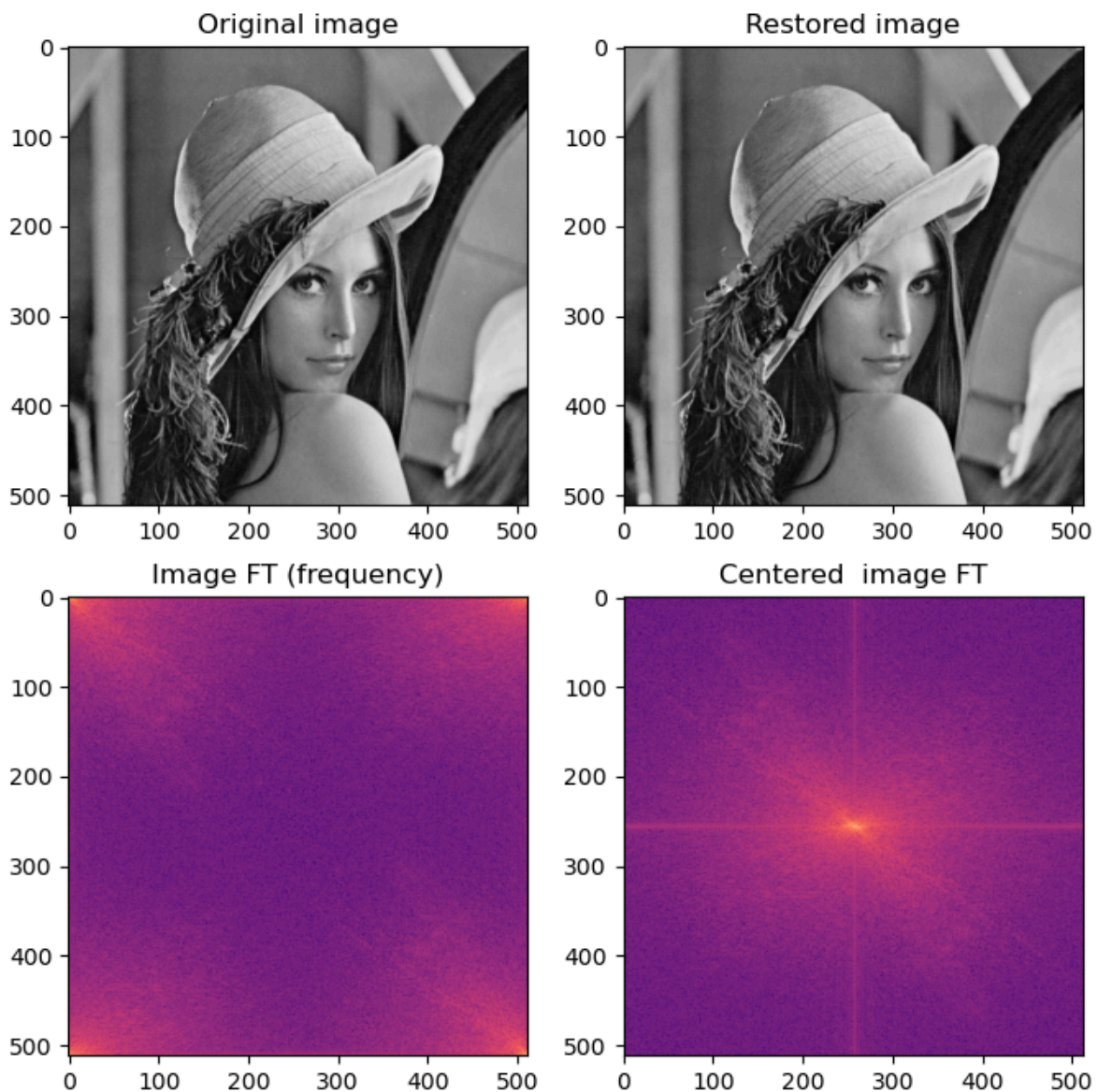
#image = img_as_float(rgb2gray(astronaut()))

filename = 'Lenna.png'
lena_im = imread(filename)
image = img_as_float(rgb2gray(lena_im))
#image = img_as_float(rgb2gray(astronaut()))

# Прямє перетворення
image_f = fft2(image) # ПРЕОБРАЗОВАНИЕ ФУРЬЕ
image_f_shft = np.abs(fftshift(image_f)) # СМЕЩАЕМ В ЦЕНТР ЧАСТНОЙ ОБЛАСТИ
# Зворотнє перетворення (відновлення)
```

```
wimage_restored = np.abs(iff2(image_f)) # С ПОМОЩЬЮ ОБРАТНОГО ПРЕОБРАЗОВАНИЯ  
ВОССТАНАВЛИВАЕМ ИЗОБРАЖЕНИЕ
```

```
fig, axes = plt.subplots(2, 2, figsize=(8, 8))  
ax = axes.ravel()  
ax[0].set_title("Original image")  
ax[0].imshow(image, cmap='gray')  
ax[1].set_title("Restored image")  
ax[1].imshow(wimage_restored, cmap='gray')  
ax[2].set_title("Image FT (frequency)")  
ax[2].imshow(np.log(np.abs(image_f)), cmap='magma')  
ax[3].set_title("Centered image FT")  
ax[3].imshow(np.log(image_f_shft), cmap='magma')  
plt.show()
```



Відображення спектру як поверхні

```
X = range(0, 512, 1)  
Y = range(0, 512, 1)  
X, Y = np.meshgrid(X, Y)
```

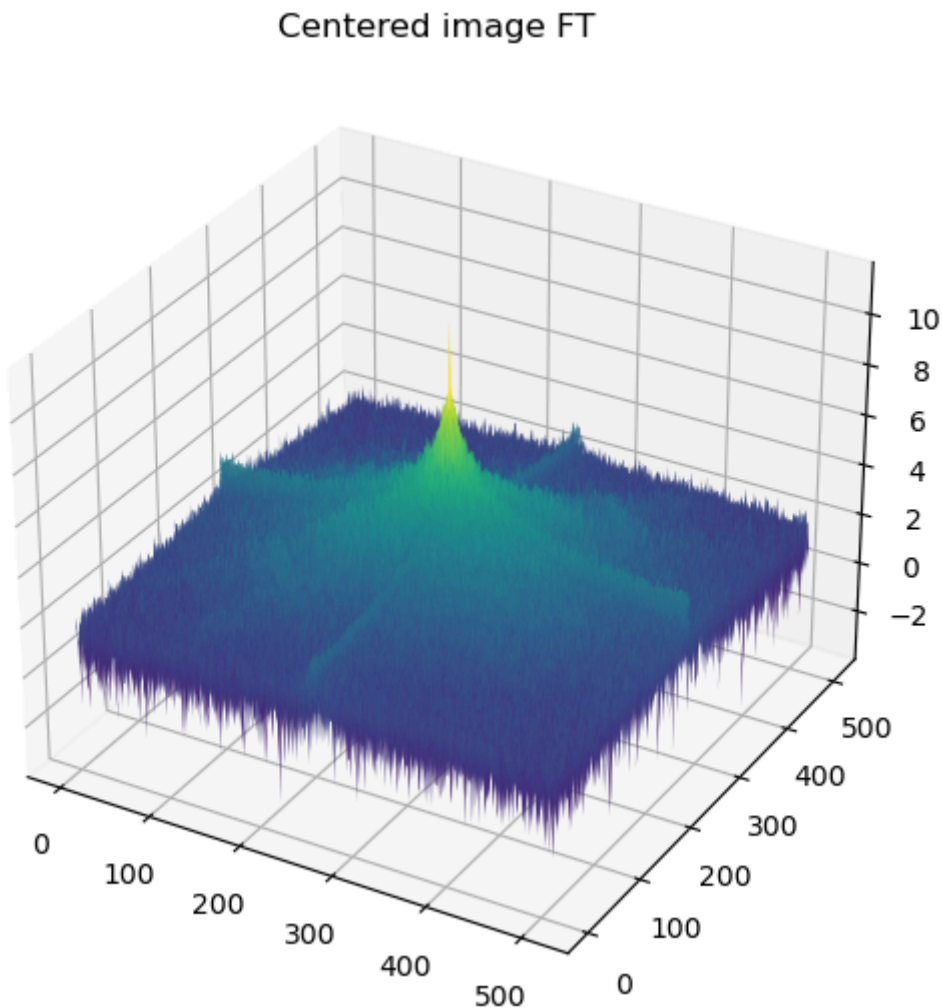
```

#snc = np.sinc(func)
ft_func = np.log(image_f_shft)

fig = plt.figure()
ax = Axes3D(fig, auto_add_to_figure=False)
fig.add_axes(ax)
ax.set_title("Centered image FT")
ax.plot_surface(X, Y, ft_func, rstride=1, cstride=1, cmap=cm.viridis)

plt.show()

```



Формування низко частотного фільтру

```

# Определяем АЧХ фильтра

filter_koeff = np.zeros ((512,512), dtype = np.complex128)

flwids = 25
for i in range (512):
    for j in range (512):
        filter_koeff[i,j] = .0001 + 0.0j
for i in range (0,flwids,1):
    for j in range (0,flwids,1):

```

```

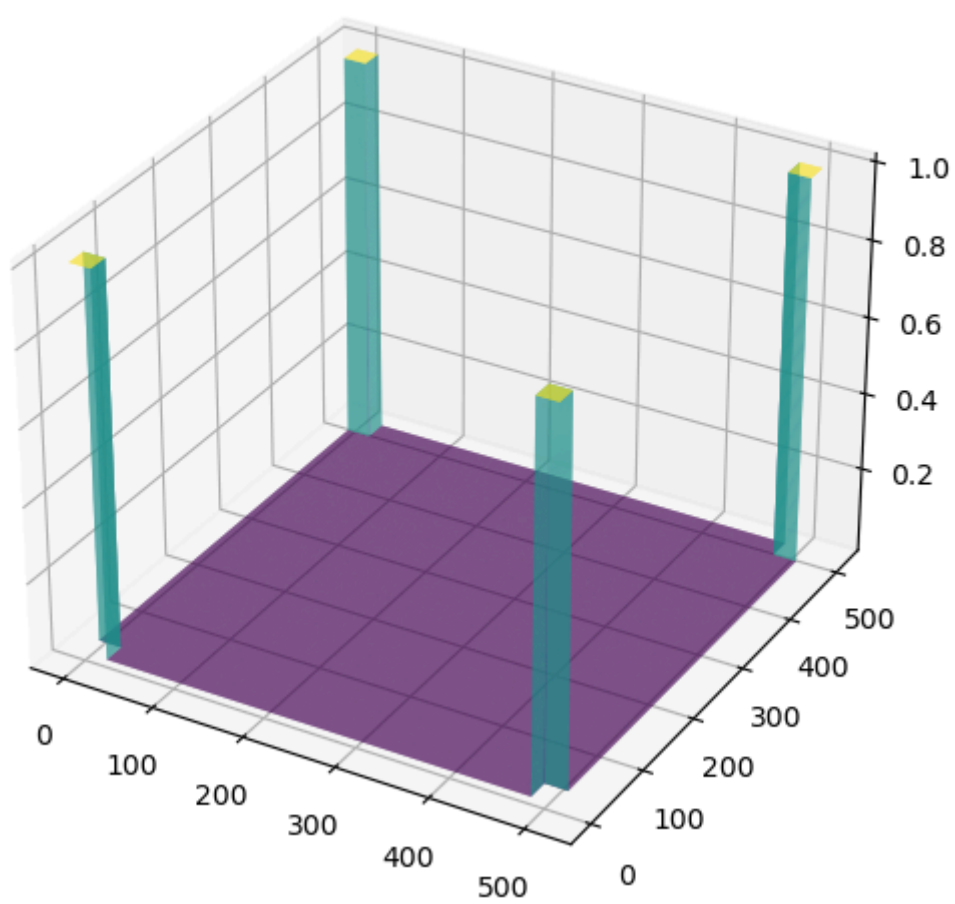
        filter_koeff[i,j] = 1.0 + 0.0j
    for i in range (512-flwids,512,1):
        for j in range (0,flwids,1):
            filter_koeff[i,j] = 1.0 + 0.0j
    for i in range (0,flwids,1):
        for j in range (512-flwids,512,1):
            filter_koeff[i,j] = 1.0 + 0.0j
    for i in range (512-flwids,512,1):
        for j in range (512-flwids,512,1):
            filter_koeff[i,j] = 1.0 + 0.0j

fig = plt.figure()
ax = Axes3D(fig, auto_add_to_figure=False)
fig.add_axes(ax)
ax.set_title("NOT centered AFR")
ax.plot_surface(X, Y, np.abs(filter_koeff), rstride=1, cstride=1,
cmmap=cm.viridis)
plt.show()

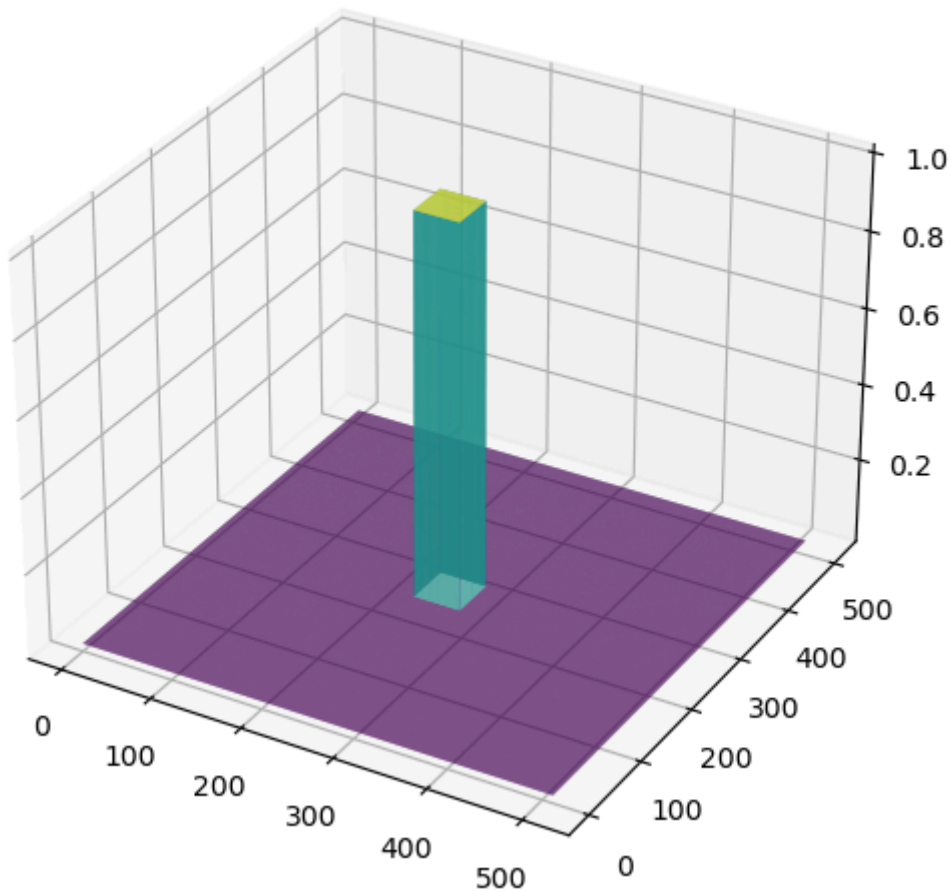
filter_koeff_shft = np.abs(fftshift(filter_koeff)) # СМЕЩАЕМ В ЦЕНТР ЧАСТНОЙ
ОБЛАСТИ
fig = plt.figure()
ax = Axes3D(fig, auto_add_to_figure=False)
fig.add_axes(ax)
ax.set_title("Centered AFR")
ax.plot_surface(X, Y, np.abs(filter_koeff_shft), rstride=1, cstride=1,
cmmap=cm.viridis)
plt.show()

```

NOT centered AFR



Centered AFR

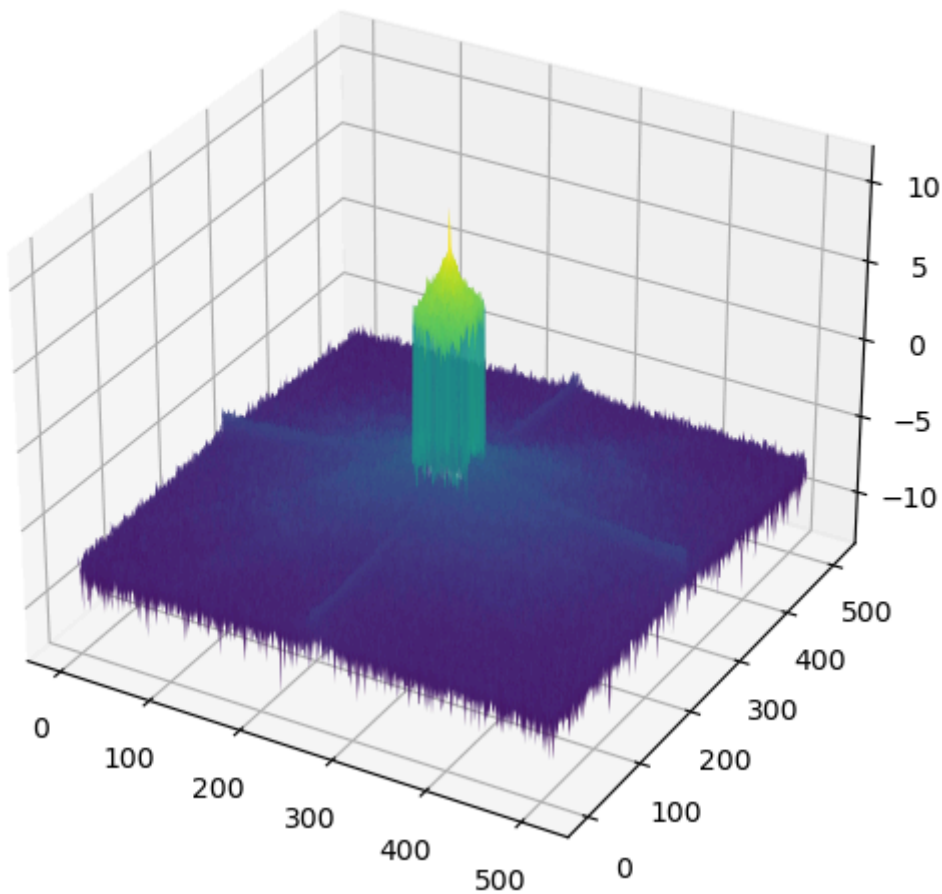


Спектр фільтрованого зображення (поверхня)

```
filter_image_f = image_f * filter_koeff
filter_image_f_shft = np.abs(fftshift(filter_image_f)) # СМЕЩАЕМ В ЦЕНТР ЧАСТНОЙ
ОБЛАСТИ

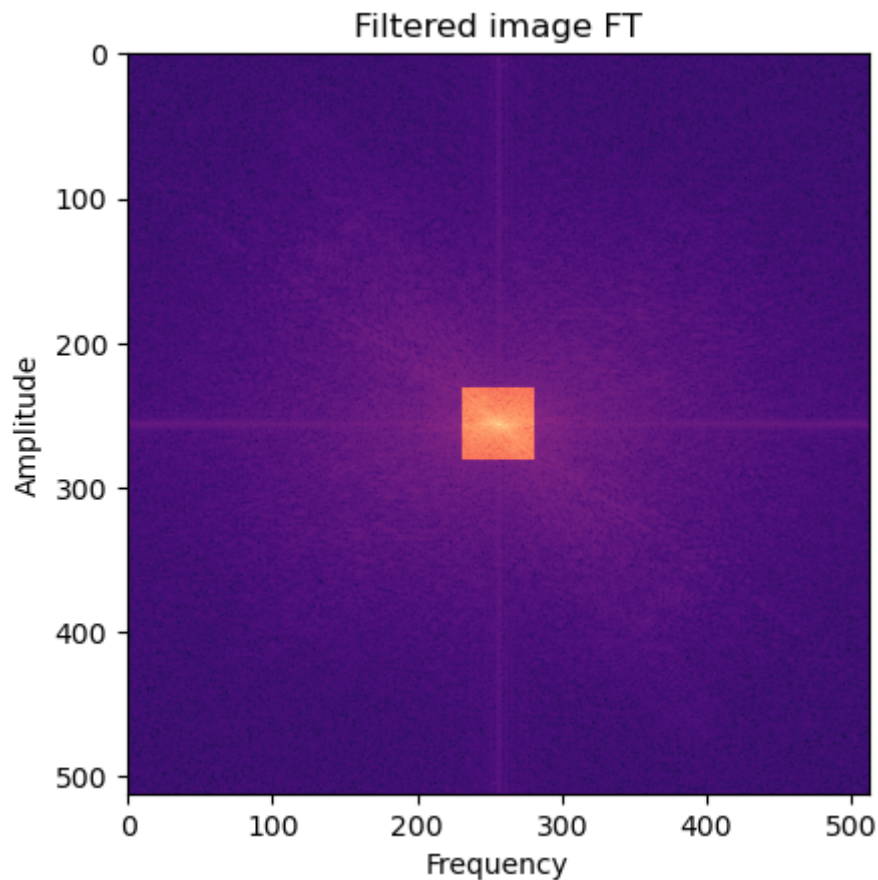
fig = plt.figure()
ax = Axes3D(fig, auto_add_to_figure=False)
fig.add_axes(ax)
ax.set_title("Filtered image FT")
ax.plot_surface(X, Y, np.log(np.abs(filter_image_f_shft)), rstride=1, cstride=1,
               cmap=cm.viridis)
plt.show()
```

Filtered image FT



Спектр фільтрованого зображення

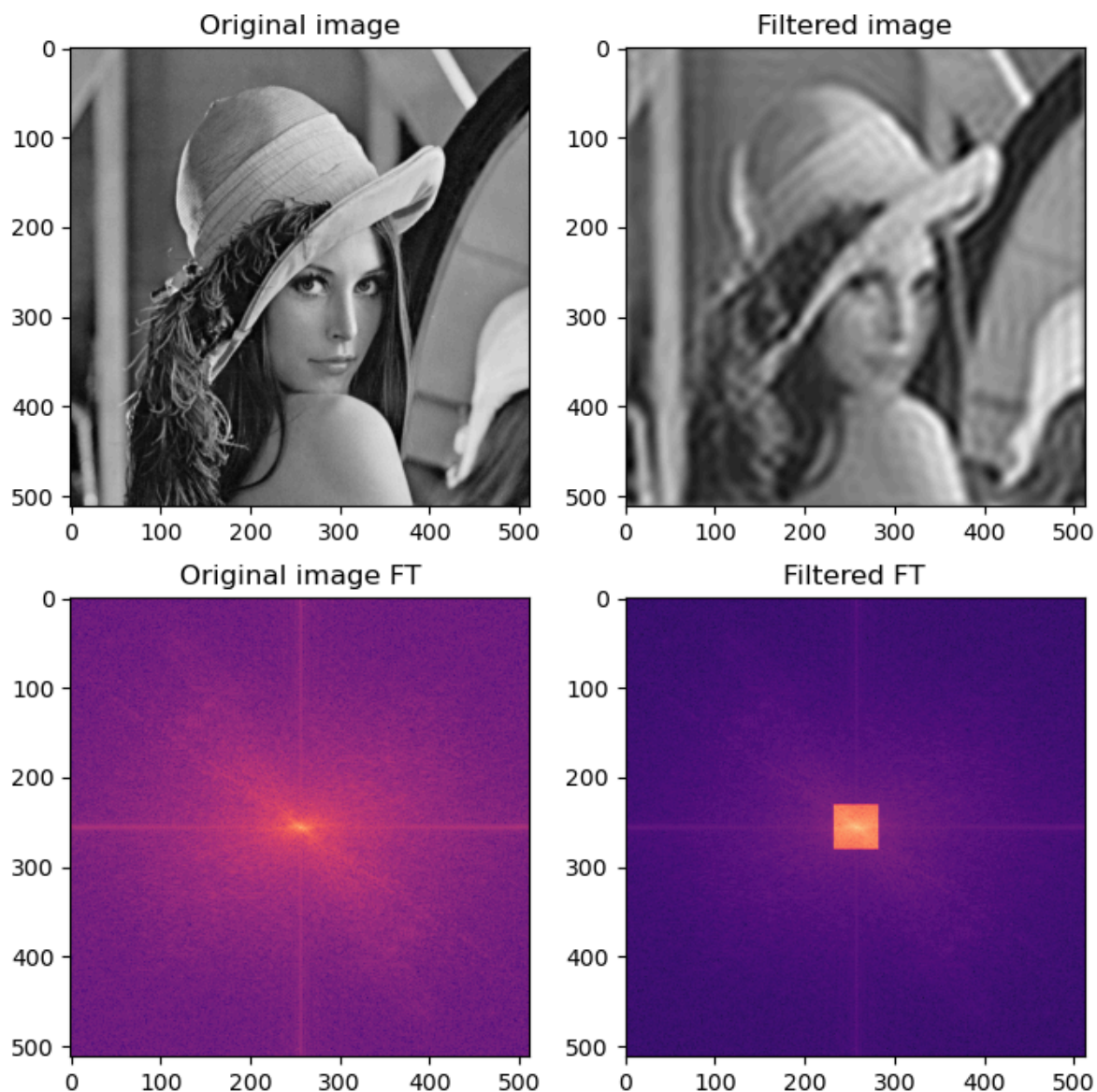
```
fig = plt.figure()
plt.imshow(np.log(np.abs(filter_image_f_shft)), cmap='magma')
plt.title("Filtered image FT")
plt.xlabel('Frequency')
plt.ylabel('Amplitude')
plt.show()
```



Відфільтроване ФНЧ зображення

```
wimage_filter_restored = np.abs(iff2(filter_image_f)) # С ПОМОЩЬЮ ОБРАТНОГО
ПРЕОБРАЗОВАНИЯ ВОССТАНАВЛИВАЕМ ИЗОБРАЖЕНИЕ
```

```
fig, axes = plt.subplots(2, 2, figsize=(8, 8))
ax = axes.ravel()
ax[0].set_title("Original image")
ax[0].imshow(image, cmap='gray')
ax[1].set_title("Filtered image")
ax[1].imshow(wimage_filter_restored, cmap='gray')
ax[2].set_title("Original image FT")
ax[2].imshow(np.log(np.abs(image_f_shft)), cmap='magma')
ax[3].set_title("Filtered FT")
ax[3].imshow(np.log(np.abs(filter_image_f_shft)), cmap='magma')
plt.show()
```

Формування високочастотного фільтру

```
# Определяем АЧХ фильтра

filter_koeff_ = np.zeros ((512,512), dtype = np.complex128)
for i in range (512):
    for j in range (512):
        filter_koeff_[i,j] = 1.0 + 0.0j

flwids = 25
for i in range (0,flwids,1):
    for j in range (0,flwids,1):
        filter_koeff_[i,j] = .0001 + 0.0j
for i in range (512-flwids,512,1):
    for j in range (0,flwids,1):
        filter_koeff_[i,j] = .0001 + 0.0j
for i in range (0,flwids,1):
    for j in range (512-flwids,512,1):
        filter_koeff_[i,j] = .0001 + 0.0j
for i in range (512-flwids,512,1):
```

```

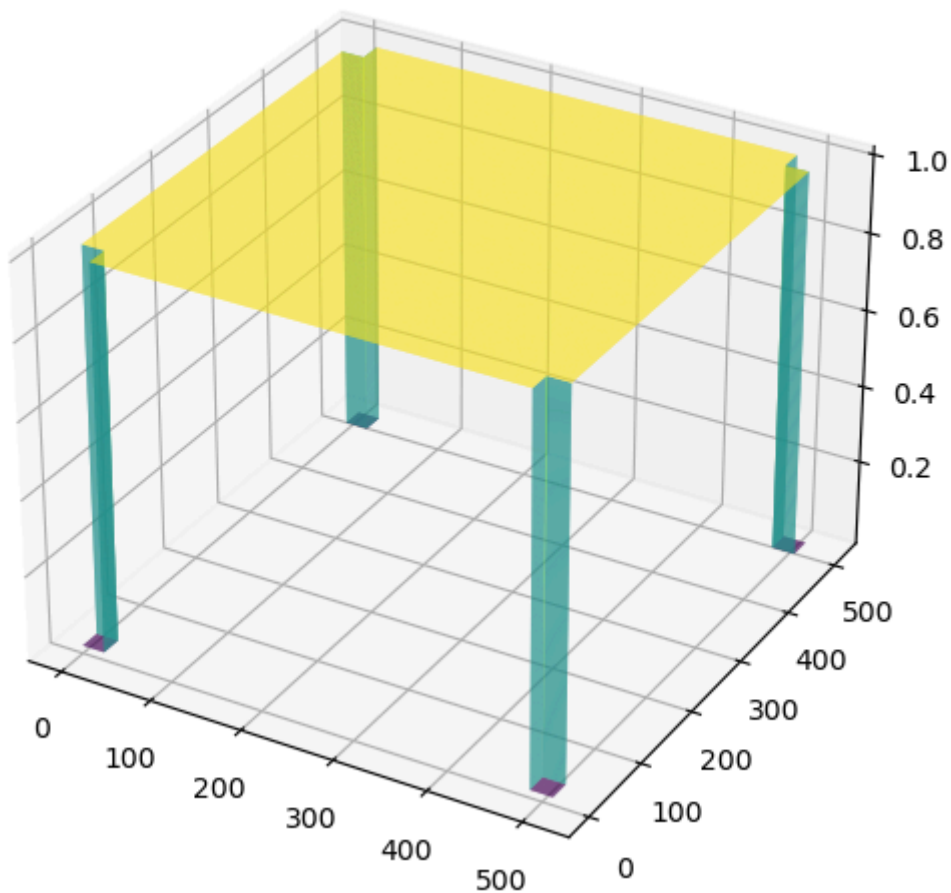
for j in range (512-flwids,512,1):
    filter_koeff_[i,j] = .0001 + 0.0j

fig = plt.figure()
ax = Axes3D(fig, auto_add_to_figure=False)
fig.add_axes(ax)
ax.set_title("NOT centered AFR")
ax.plot_surface(X, Y, np.abs(filter_koeff_), rstride=1, cstride=1,
               cmap=cm.viridis)
plt.show()

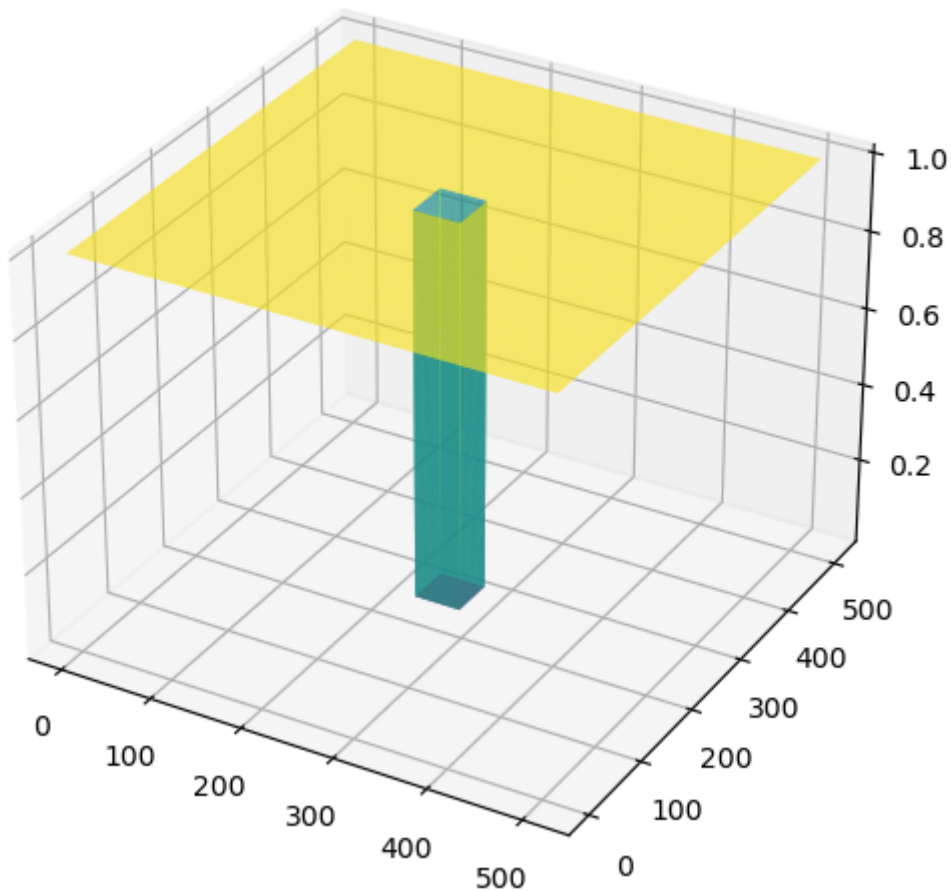
filter_koeff_shft_ = np.abs(fftshift(filter_koeff_)) # СМЕЩАЕМ В ЦЕНТР ЧАСТНОЙ
ОБЛАСТИ
fig = plt.figure()
ax = Axes3D(fig, auto_add_to_figure=False)
fig.add_axes(ax)
ax.set_title("Centered AFR")
ax.plot_surface(X, Y, np.abs(filter_koeff_shft_), rstride=1, cstride=1,
               cmap=cm.viridis)
plt.show()

```

NOT centered AFR



Centered AFR



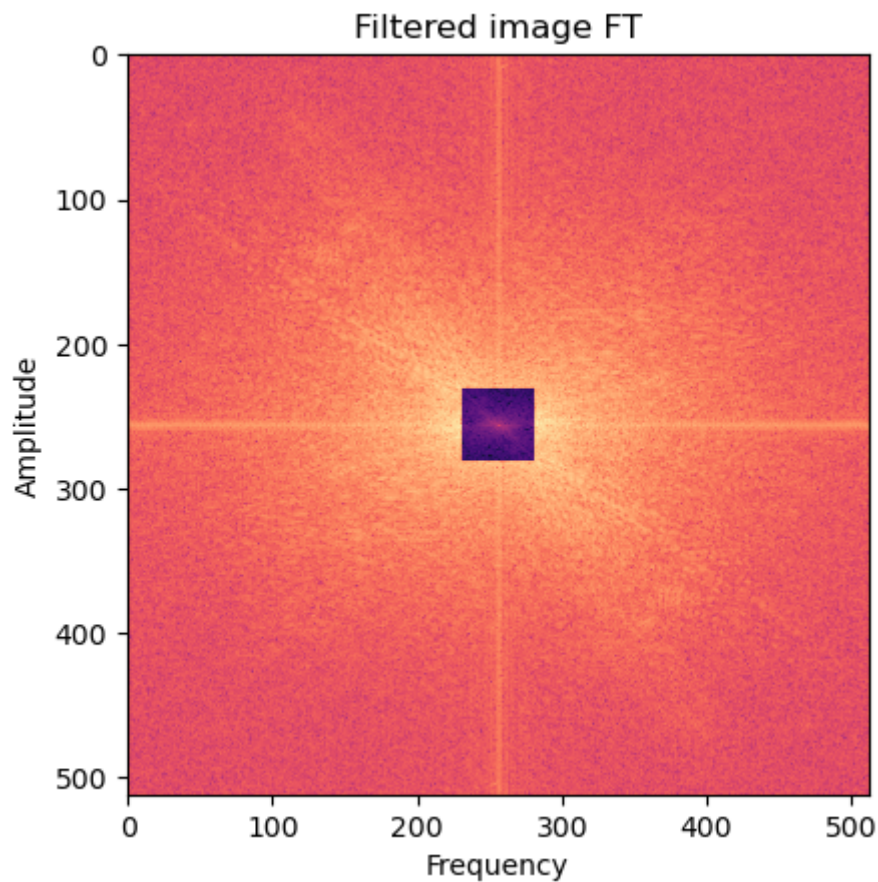
Спектр фільтрованого зображення

```
filter_image_f_ = image_f * filter_koeff_  
filter_image_f_shft_ = np.abs(fftshift(filter_image_f_)) # СМЕЩАЕМ В ЦЕНТР ЧАСТНОЙ  
ОБЛАСТИ
```

```
fig = plt.figure()  
ax = Axes3D(fig)  
ax.set_title("Filtered image FT")  
ax.plot_surface(X, Y, np.abs(filter_image_f_shft_), rstride=1, cstride=1,  
               cmap=cm.viridis)  
plt.show()
```

```
fig = plt.figure()  
plt.imshow(np.log(np.abs(filter_image_f_shft_)), cmap='magma')  
plt.title("Filtered image FT")  
plt.xlabel('Frequency')  
plt.ylabel('Amplitude')  
plt.show()
```

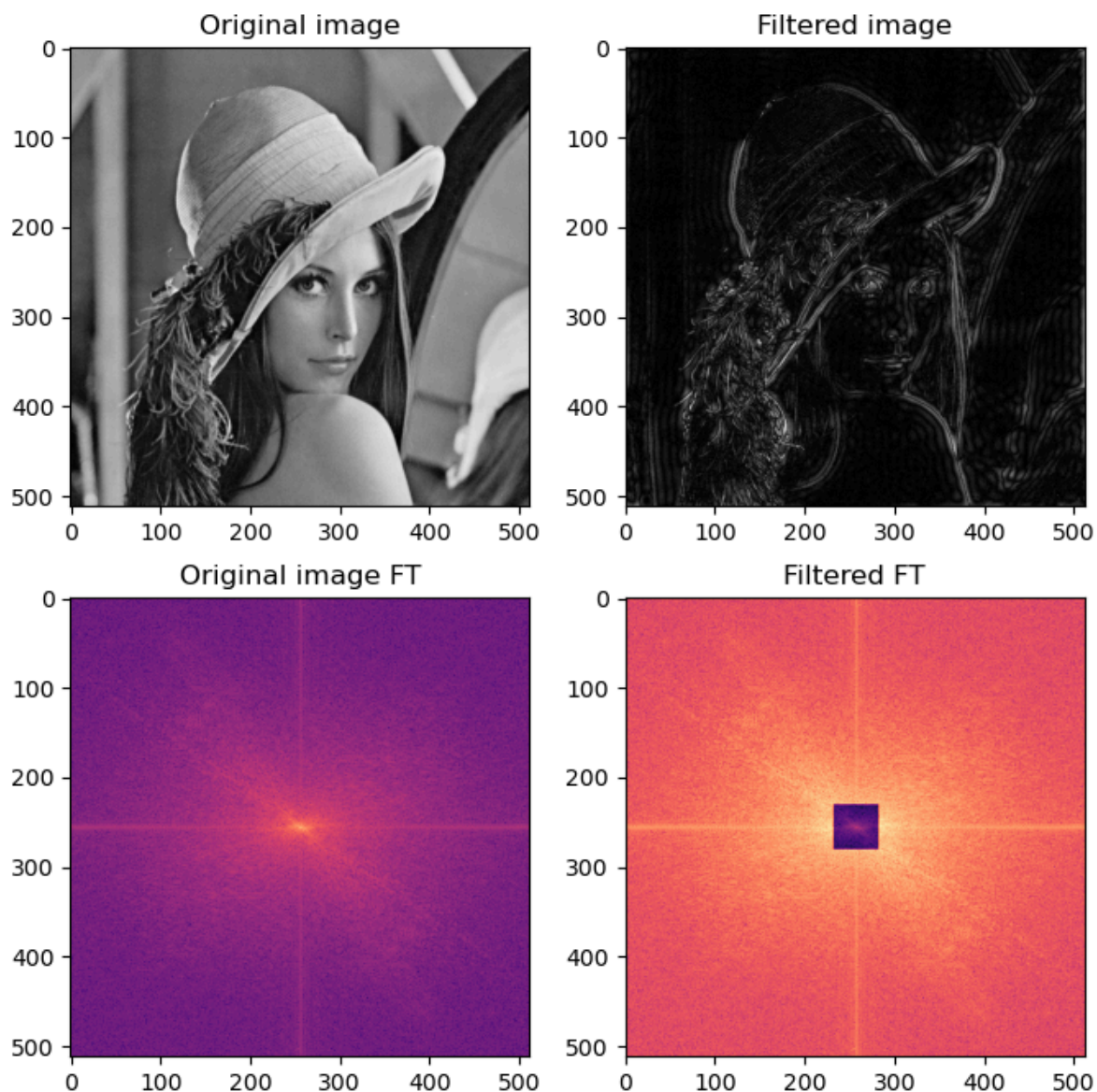
<Figure size 640x480 with 0 Axes>



Відфільтроване ФВЧ зображення

```
wimage_filter_restored_ = np.abs(iff2(filter_image_f_)) # С ПОМОЩЬЮ ОБРАТНОГО
ПРЕОБРАЗОВАНИЯ ВОССТАНАВЛИВАЕМ ИЗОБРАЖЕНИЕ
```

```
fig, axes = plt.subplots(2, 2, figsize=(8, 8))
ax = axes.ravel()
ax[0].set_title("Original image")
ax[0].imshow(image, cmap='gray')
ax[1].set_title("Filtered image")
ax[1].imshow(wimage_filter_restored_, cmap='gray')
ax[2].set_title("Original image FT")
ax[2].imshow(np.log(np.abs(image_f_shft)), cmap='magma')
ax[3].set_title("Filtered FT")
ax[3].imshow(np.log(np.abs(filter_image_f_shft_)), cmap='magma')
plt.show()
```



Формування довільного фільтру

```
# Определяем АЧХ фильтра

filter_koeff_2 = np.zeros ((512,512), dtype = np.complex128)

for i in range (256):
    for j in range (256):
        koeff = 0.002762*np.sqrt((i-256)*(i-256)+(j-256)*(j-256)) + 0.001j
        filter_koeff_2[i,j] = koeff + 0.0j
for i in range (256,512):
    for j in range (256):
        koeff = 0.002762*np.sqrt((i-256)*(i-256)+(j-256)*(j-256)) + 0.001j
        filter_koeff_2[i,j] = koeff + 0.0j
for i in range (256):
    for j in range (256,512):
        koeff = 0.002762*np.sqrt((i-256)*(i-256)+(j-256)*(j-256)) + 0.001j
        filter_koeff_2[i,j] = koeff + 0.0j
for i in range (256,512):
    for j in range (256,512):
        koeff = 0.002762*np.sqrt((i-256)*(i-256)+(j-256)*(j-256)) + 0.001j
        filter_koeff_2[i,j] = koeff + 0.0j
```

```

    koeff = 0.002762*np.sqrt((i-256)*(i-256)+(j-256)*(j-256)) + 0.001j
    filter_koeff_2[i,j] = koeff + 0.0j

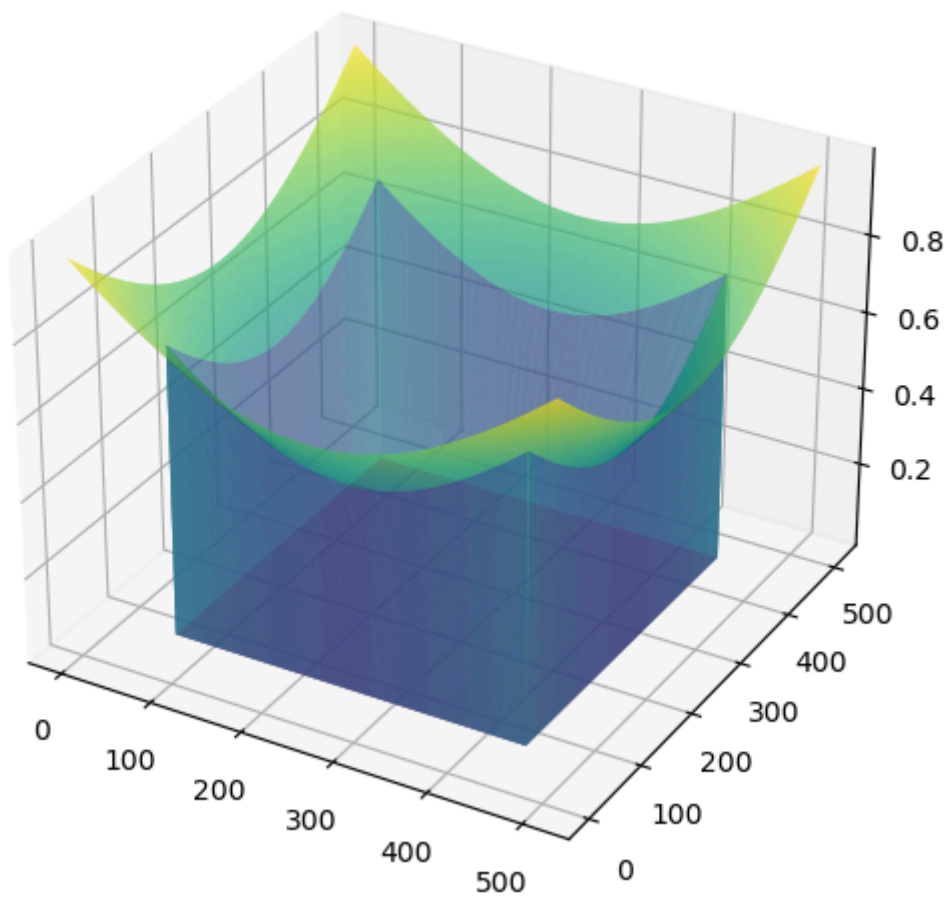
for i in range (64,448):
    for j in range (64,448):
        filter_koeff_2[i,j] = .0001 + 0.0j

fig = plt.figure()
ax = Axes3D(fig, auto_add_to_figure=False)
fig.add_axes(ax)
ax.set_title("NOT centered AFR")
ax.plot_surface(X, Y, np.abs(filter_koeff_2), rstride=1, cstride=1,
               cmap=cm.viridis)
plt.show()

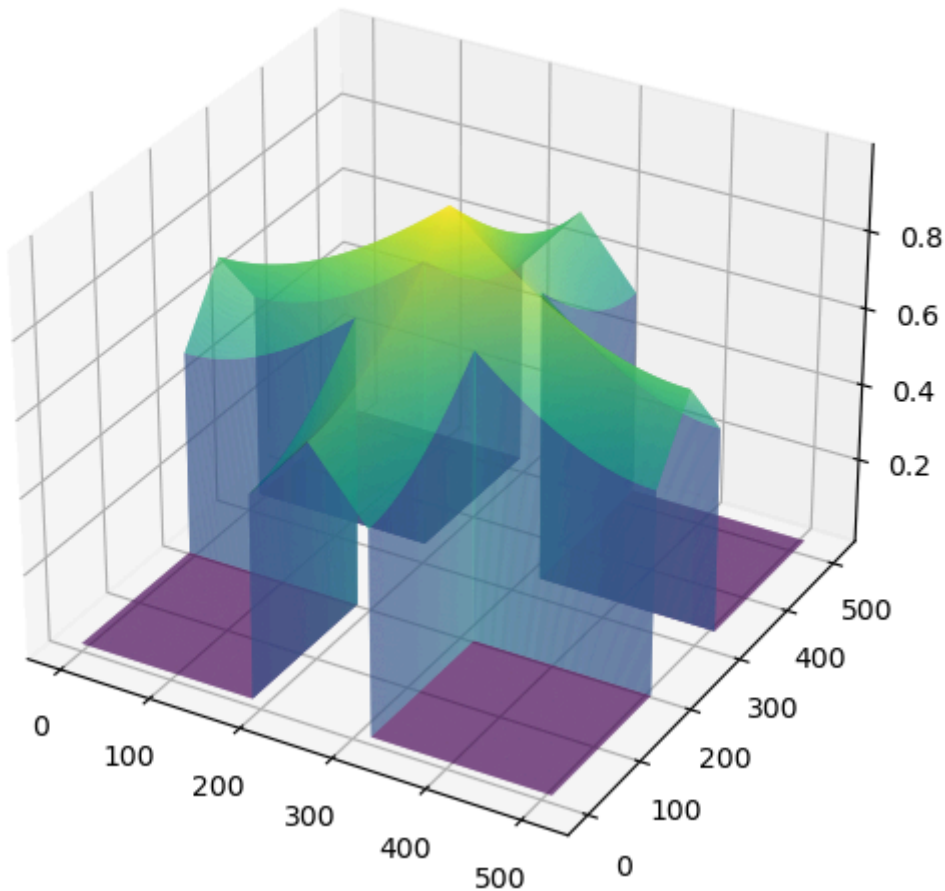
filter_koeff_2_shft_ = np.abs(fftshift(filter_koeff_2)) # СМЕЩАЕМ В ЦЕНТР ЧАСТНОЙ
ОБЛАСТИ
fig = plt.figure()
ax = Axes3D(fig, auto_add_to_figure=False)
fig.add_axes(ax)
ax.set_title("Centered AFR")
ax.plot_surface(X, Y, np.abs(filter_koeff_2_shft_), rstride=1, cstride=1,
               cmap=cm.viridis)
plt.show()

```


NOT centered AFR



Centered AFR



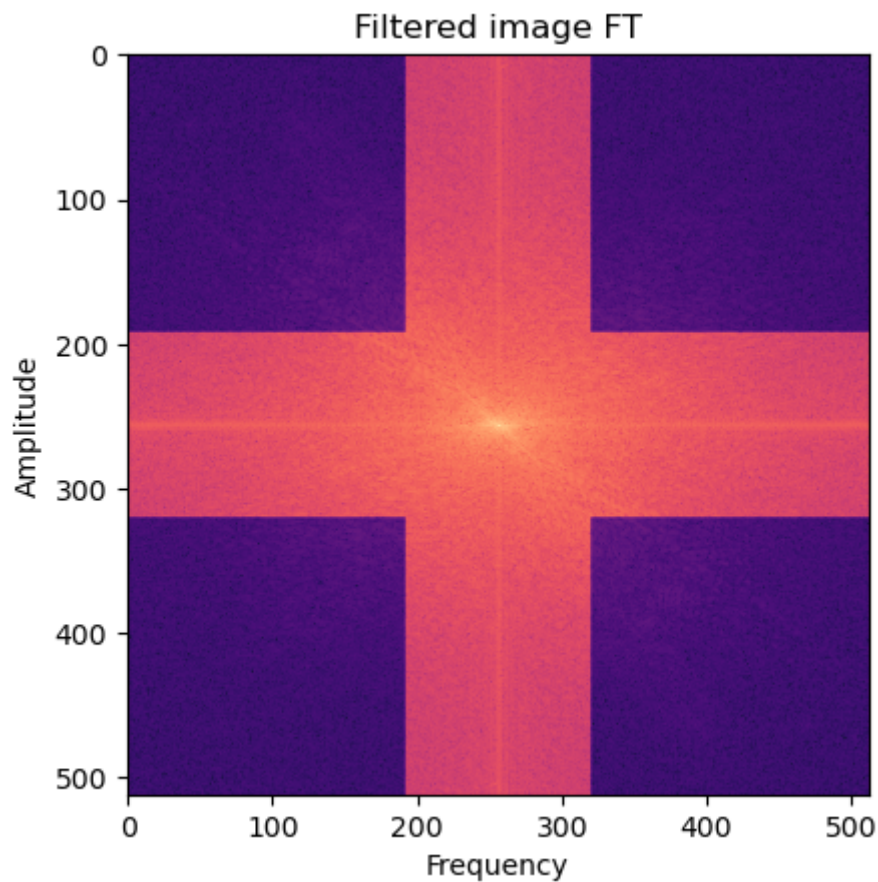
Спектр відфільтрованого зображення

```
filter_image_2_f_ = image_f * filter_koeff_2
filter_image_2_f_shft_ = np.abs(fftshift(filter_image_2_f_)) # СМЕЩАЕМ В ЦЕНТР ЧАСТНОЙ ОБЛАСТИ
```

```
fig = plt.figure()
ax = Axes3D(fig)
ax.set_title("Filtered image FT")
ax.plot_surface(X, Y, (np.log(np.abs(filter_image_2_f_shft_))), rstride=1,
               cstride=1, cmap=cm.viridis)
plt.show()
```

```
fig = plt.figure()
plt.imshow(np.log((np.abs(filter_image_2_f_shft_))), cmap='magma')
plt.title("Filtered image FT")
plt.xlabel('Frequency')
plt.ylabel('Amplitude')
plt.show()
```

<Figure size 640x480 with 0 Axes>



Відфільтроване зображення

```
wimage_filter_2_restored_ = np.abs(iff2(filter_image_2_f_)) # С ПОМОЩЬЮ ОБРАТНОГО
ПРЕОБРАЗОВАНИЯ ВОССТАНАВЛИВАЕМ ИЗОБРАЖЕНИЕ
```

```
fig, axes = plt.subplots(2, 2, figsize=(8, 8))
ax = axes.ravel()
ax[0].set_title("Original image")
ax[0].imshow(image, cmap='gray')
ax[1].set_title("Filtered image")
ax[1].imshow(wimage_filter_2_restored_, cmap='gray')
ax[2].set_title("Original image FT")
ax[2].imshow(np.log(np.abs(image_f_shft)), cmap='magma')
ax[3].set_title("Filtered FT")
ax[3].imshow(np.log(np.abs(filter_image_2_f_shft_)), cmap='magma')
plt.show()
```

