

# РОБОТА із ЗОБРАЖЕННЯМИ

Файл: Image\_07\_003

## Перетворення Фур'є для 2D сигналу

Використовуємо бібліотеку [SCIPY](#)

```
%matplotlib inline
```

```
import numpy as np # модуль для операцій со списками и массивами

from scipy.fft import fft # модуль для Фурье
from scipy.fft import ifftn
from scipy.integrate import quad # модуль для интегрирования

import matplotlib.pyplot as plt
import matplotlib.cm as cm

import matplotlib.pyplot as plt # модуль для графиков
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D
```

## ФУР'Є ПЕРЕТВОРЕННЯ ЗОБРАЖЕННЯ

```
frec_max = 2 # Герц ?!
points = 51
x = np.linspace(-frec_max , frec_max, points )
Y = np.linspace(-frec_max, frec_max, points )

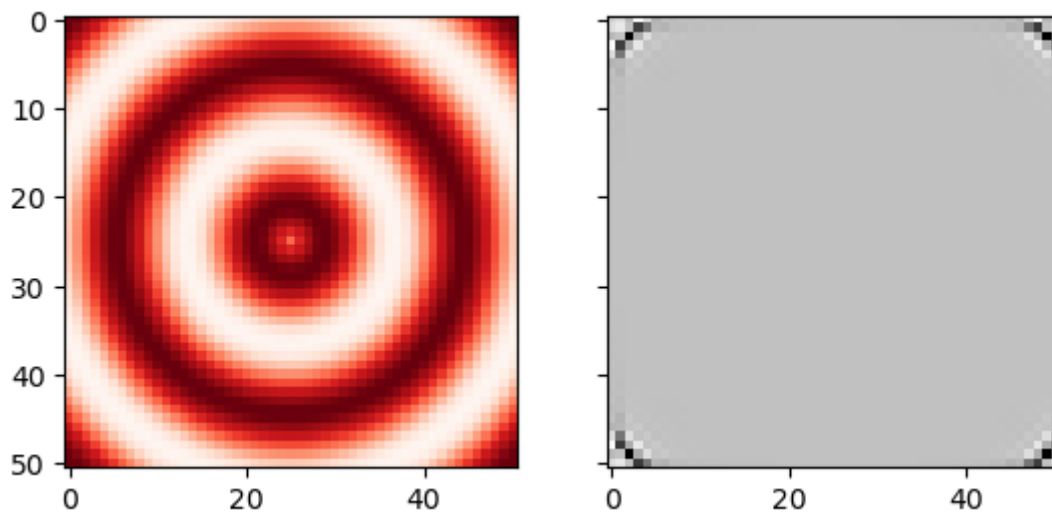
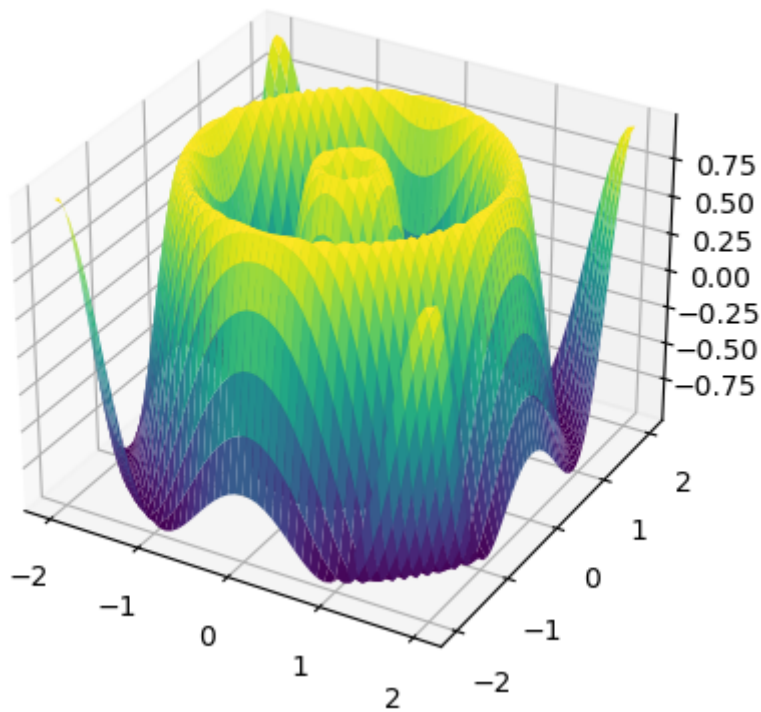
X, Y = np.meshgrid(X, Y)
func = np.sqrt((X/5)**2 + (Y/5)**2)
twoDsin = np.sin(func*8*np.pi)

# fig = plt.figure()
fig = plt.figure(figsize=plt.figaspect(0.5))
ax = fig.add_subplot(1, 1, 1, projection='3d')
ax.plot_surface(X, Y, twoDsin, rstride=1, cstride=1, cmap=cm.viridis)
plt.show()

f, ((ax1), (ax2)) = plt.subplots(1, 2, sharex='col', sharey='row')

Z = ifftn(twoDsin )
ax1.imshow(twoDsin, cmap=cm.Reds)
ax2.imshow(np.real(Z), cmap=cm.Greys)

plt.show()
```



```

N = 200
Tau = 1
T = 100
f, ((ax1), (ax2)) = plt.subplots(1, 2, sharex='col', sharey='row')

xf = np.zeros((N,N))
suar_puls = np.ones((Tau,Tau))
for i in range (0, N, T):
    for j in range (0, N, T):
        xf[i : i+Tau :1, j : j + Tau :1 ] = suar_puls[:,:]

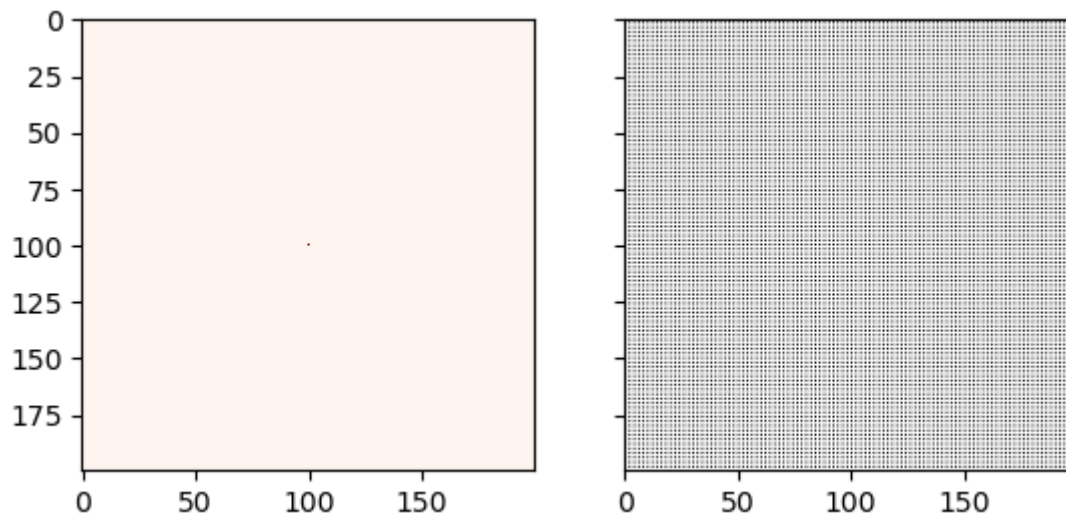
```

```

Z = ifftn(xf)
ax1.imshow(xf, cmap=cm.Reds)
ax2.imshow(np.real(Z), cmap=cm.Greys)

plt.show()

```



# ПРИКЛАД 3  
<https://docs.scipy.org/doc/scipy/reference/tutorial/fft.html#and-n-d-discrete-fourier-transforms>

```

N = 30
f, ((ax1, ax2, ax3), (ax4, ax5, ax6)) = plt.subplots(2, 3, sharex='col',
sharey='row')
xf = np.zeros((N,N))
xf[0, 5] = 1
xf[0, N-5] = 1
Z = ifftn(xf)
ax1.imshow(xf, cmap=cm.Reds)
ax4.imshow(np.real(Z), cmap=cm.Greys)
xf = np.zeros((N, N))
xf[5, 0] = 1
xf[N-5, 0] = 1
Z = ifftn(xf)
ax2.imshow(xf, cmap=cm.Reds)
ax5.imshow(np.real(Z), cmap=cm.Greys)
xf = np.zeros((N, N))
xf[5, 10] = 1
xf[N-5, N-10] = 1
Z = ifftn(xf)
ax3.imshow(xf, cmap=cm.Reds)
ax6.imshow(np.real(Z), cmap=cm.Greys)
plt.show()

```

