

w domu
na listach
to samo

plt

2 np

n_wierszy = ...
n_kolumn

0	0	0
0	0	0

```
def gen_obrazek(n_wierszy, n_kolumn):  
    return
```

w domu
Wypróbować PyCharm

Rozwiązania problemów
VS Code - powershell

1) wymusić cmd zamiast powershell

2) tajemnicza linia kodu w powershell

3) współpraca z githubem

```
obrazek = gen_obrazek(n_wierszy=6, n_kolumn=4)  
obrazek[2,1] = 1
```

```
plt.imshow(obrazek, cmap='binary')
```

```
plt.savefig('inferno')
```

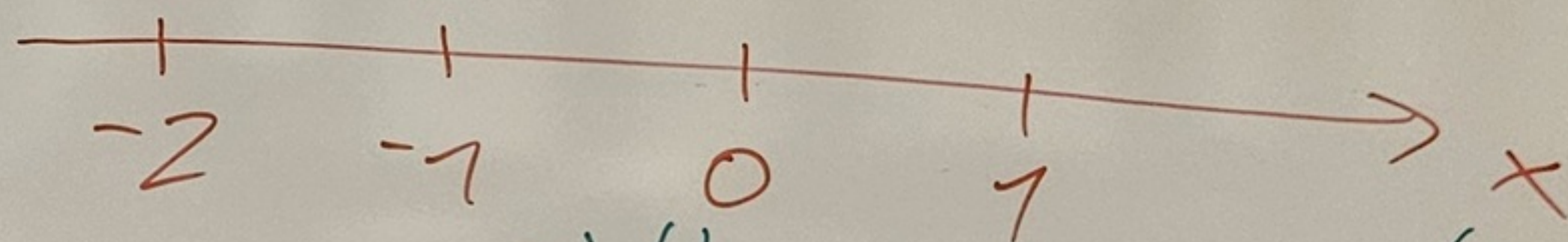
```
plt.show()
```


Przykład sesji użytkownika:
 „Chcę obrazek 800 na 600
 z zakresami $x \in \langle -1, 2 \rangle$
 $y \in \langle 2, 5 \rangle$ i żeby
 kolor był równy części
 rzeczywistej”

W domu
 jeżeli nie
 zdążysz na
 lekcji

def gen_siatka_zesp(wymiary_obrazka,
 zakresy_matemat)

		0	1	2	3
0	4	$-2 + 4j$	$-1 + 4j$	$0 + 4j$	$1 + 4j$
1	3	$-2 + 3j$	$-1 + 3j$	$0 + 3j$	$1 + 3j$
2	2	$-2 + 2j$	$-1 + 2j$	$0 + 2j$	$1 + 2j$
3	1	$-2 + 1j$	$-1 + 1j$	$0 + 1j$	$1 + 1j$
4	0	$-2 + 0j$	$-1 + 0j$	$0 + 0j$	$1 + 0j$
	-1	$-2 - 1j$	$-1 - 1j$	$0 - 1j$	$1 - 1j$



siatka = np.zeros(5, 8, dtype=complex)

$\langle -2, 1 \rangle$

abs
 obrazek = np.real(siatka)
 imag

plt.imshow(obrazek)

$$[1, 2, 3] + [4, 5, 6]$$

W domu
żeby działało jak
jak numpy zip

$$\frac{1}{\text{liczba zesp} - 1}$$

[a + b for ...]

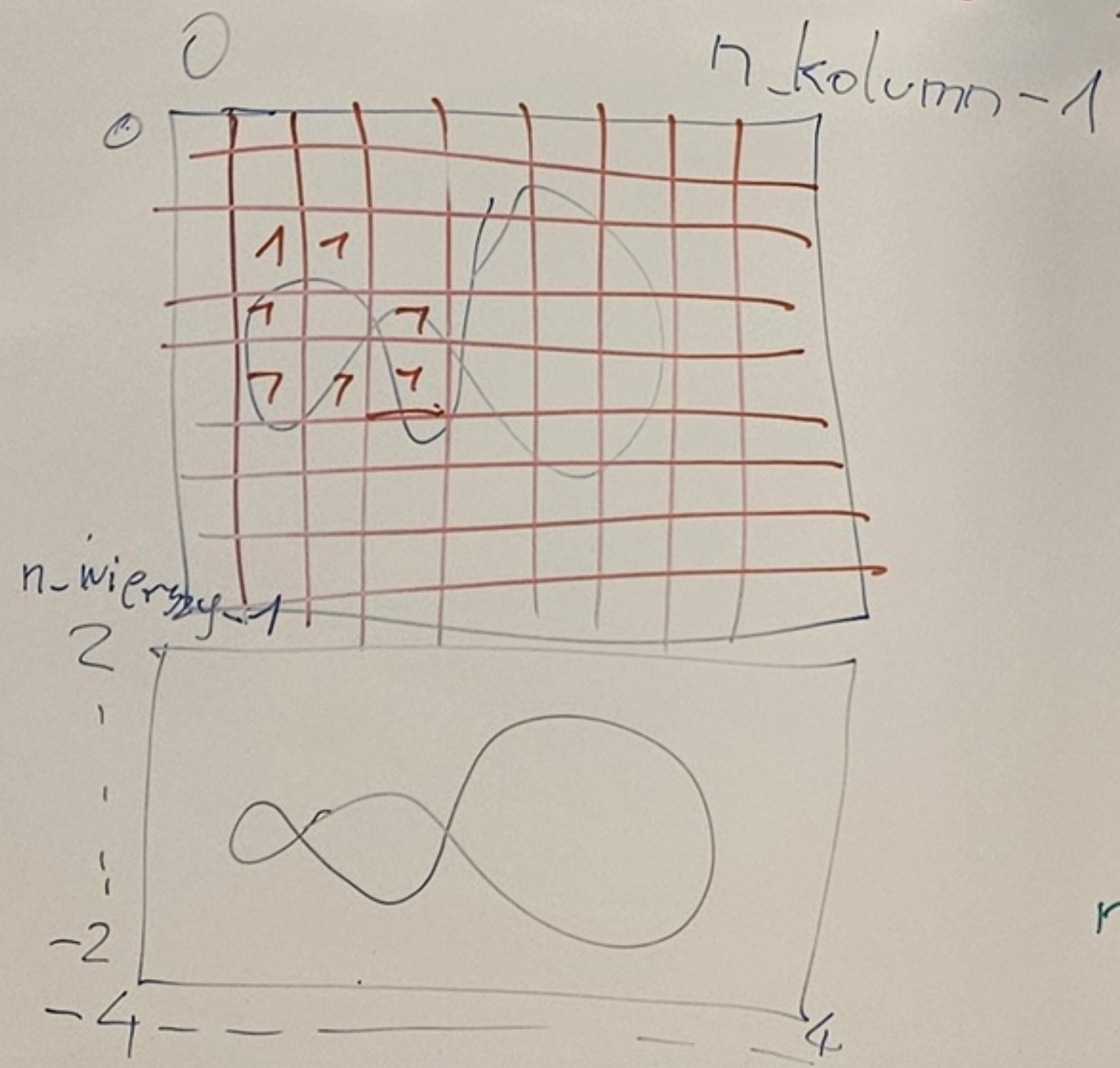
$$[1, 2, 3] ** 2$$

np.zeros((4, 8))

$$[1] * 6$$

np
numpy

$$[1] * 8 \rightarrow [1, 1, 1, 1, \dots, 1]$$



$$t = \dots$$

$$X = \sin(Z * t)$$

$$y = \dots$$

$$\text{np.arange}(\text{start} = -4, \text{stop} = 4, \text{step} = \dots)$$

$$\text{np.linspace}(\text{start} = -4, \text{stop} = 4, \text{num} = \text{n_kolumn})$$