Języki i techniki programowania

Wykład 6

Maciej Rybczyński

newaxis, wszystkie pary

```
import numpy as np

a = np.array([1,2,3,4])
print(a, '\n')

b = np.array([10,100,1000])
b = b[: , np.newaxis]
print(b, '\n')

print(a + b)
```

równoważne z: b = b.reshape(3,1)

spróbuj: a==b, a!=b, a<b, etc.

```
= RESTART: D:/CERNBox/zajeci
[1 2 3 4]

[[  10]
 [ 100]
 [1000]]

[[  11   12   13   14]
 [ 101  102  103  104]
 [1001 1002 1003 1004]]
```

Liczby całkowite w numpy

```
import numpy as np

a = np.array([125,9], dtype=np.int64)

print(a, a.dtype.name, '\n')

print(a**10)

# int16 Integer (-32768 do 32767)
# int32 Integer (-2147483648 do 2147483647)
# int64 Integer (-9223372036854775808 do 9223372036854775807)
```

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danyc
[125   9] int64

[89853709300009348 25          3486784401]
```

Zobacz: https://numpy.org/doc/stable/user/basics.types.html

Liczby całkowite w Pythonie, long

```
a = 2**1256

print(a)
print('\n')
print(type(a))
```

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/lec6_c.py =
1240722202618646512261061707420836155249507695110643482842339557457400329074068104983835153786221049409201337570462347005770631581345017541425216985425629706731334866212513488027236860950181907471771171158276773835622510569995460375185311005644590889421981440223593810622503350380451155265119651781430101614841010833160939339005760608760471109018093451230071413602199639148199936

<class 'int'>
```

long integer!

Long integer w numpy

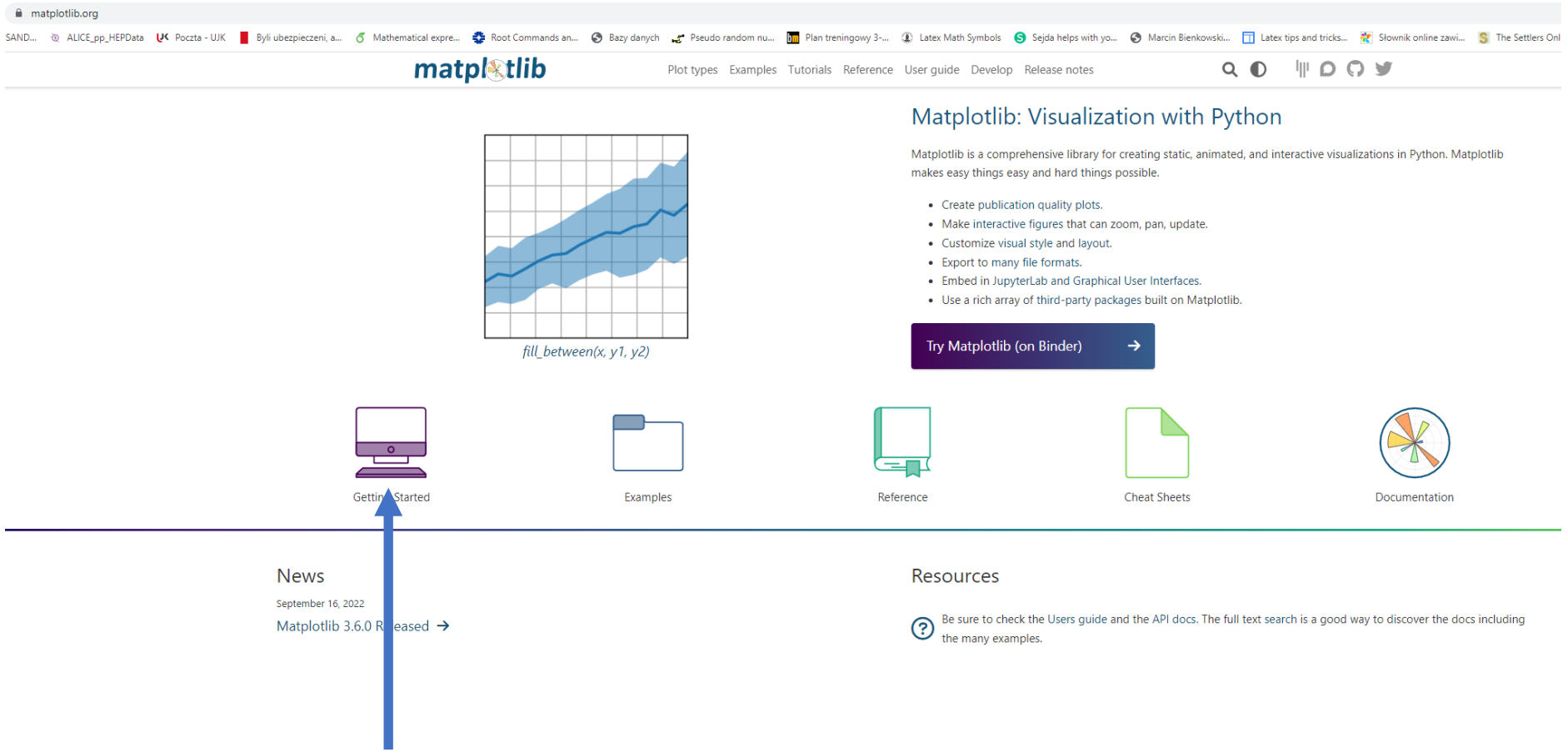
```
import numpy as np

a = np.array([125], dtype=object)

print(a, a.dtype.name, '\n')

print(a**35)
```

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/lec6_d.py
[125] object

[24651903288156618919116517665087069677287701097156968899071216583251953125]
```

MatPlotLib

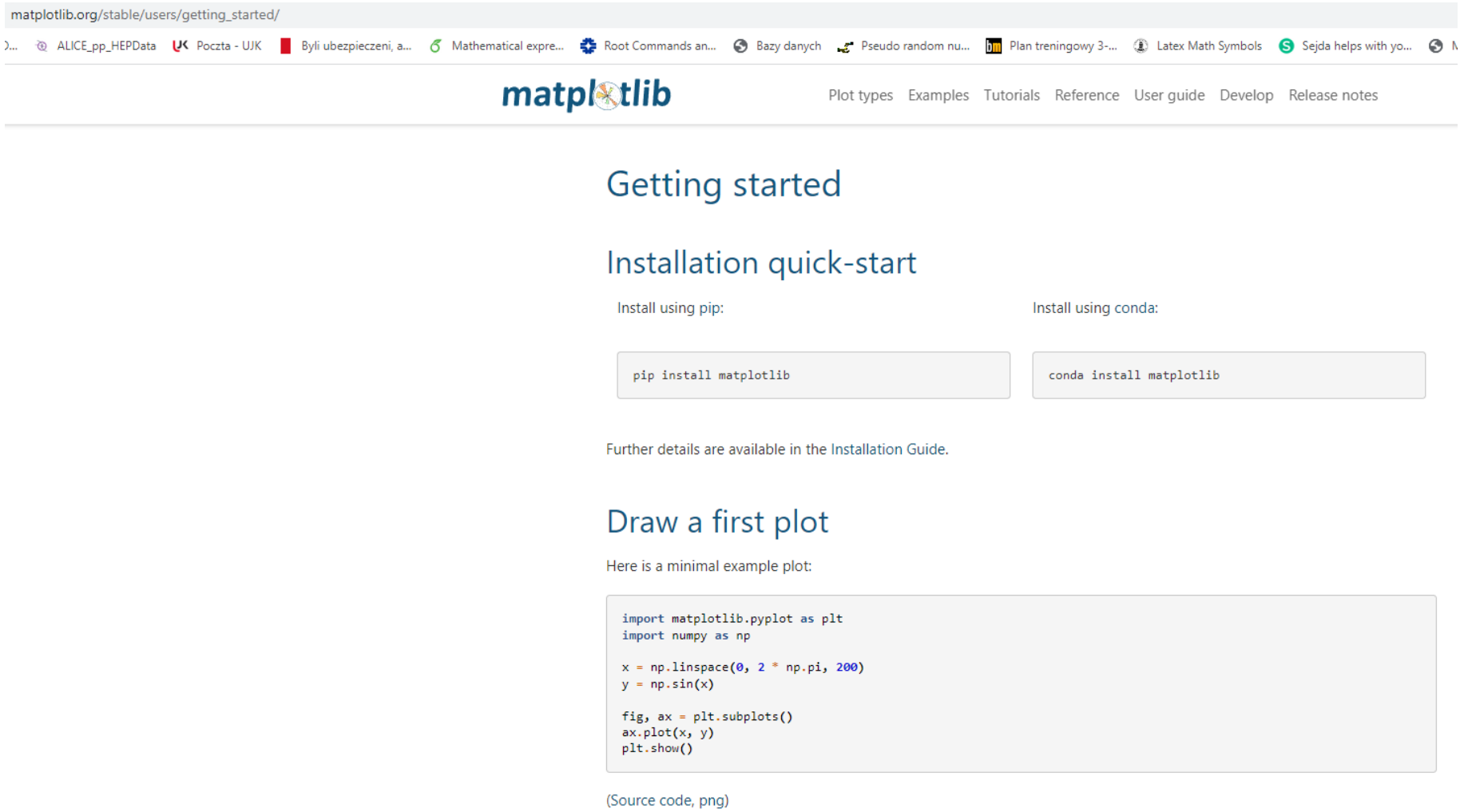
MatPlotLib, biblioteka do kreślenia 2D w Pythonie

https://matplotlib.org/



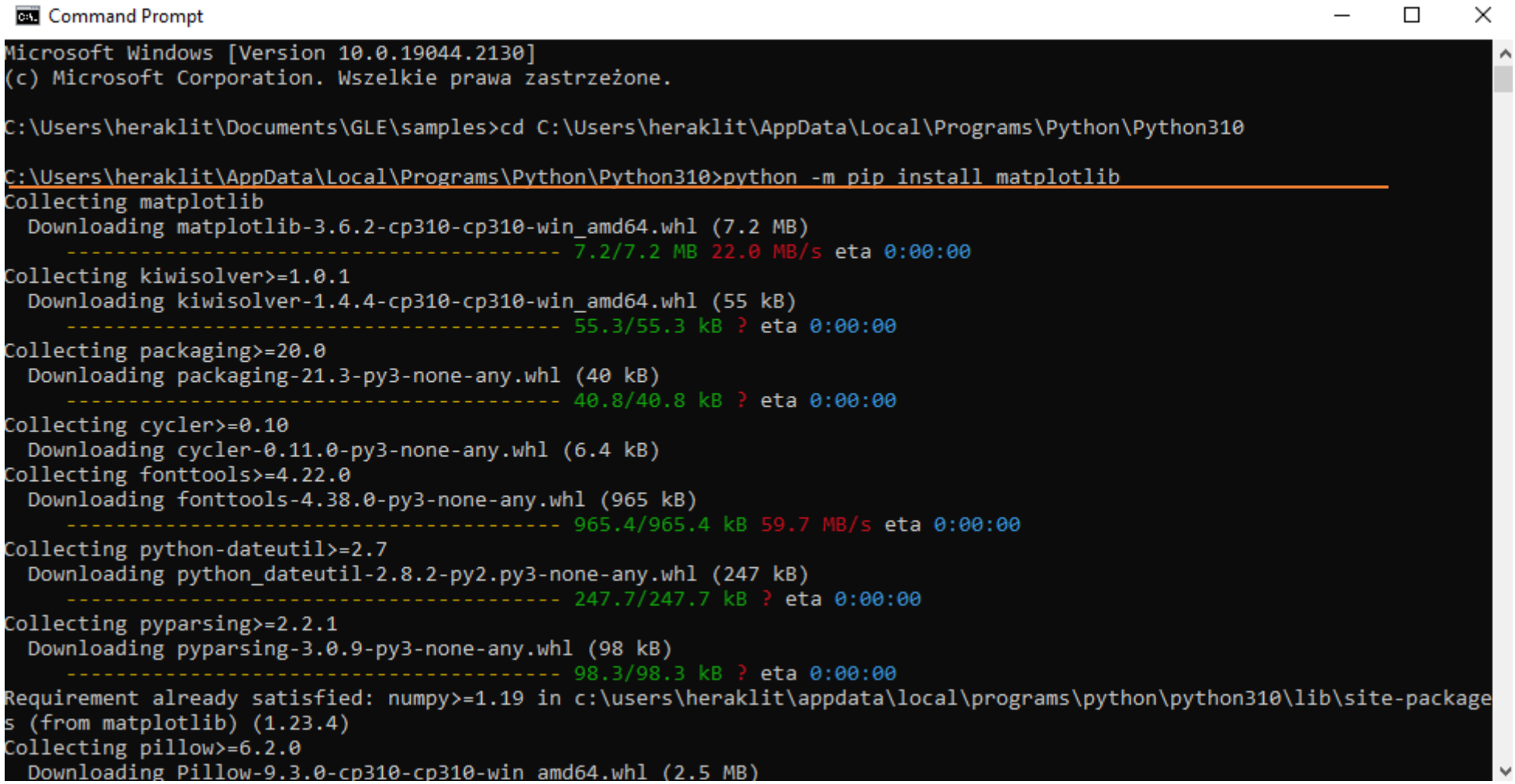
kliknij tutaj

Instalowanie w Windows



Otwórz wiersz poleceń,
przejdź do katalogu gdzie jest zainstalowany python

Łatwa instalacja!



wpisz polecenie:

python –m pip install matplotlib

Wymaganie połączenie z internetem

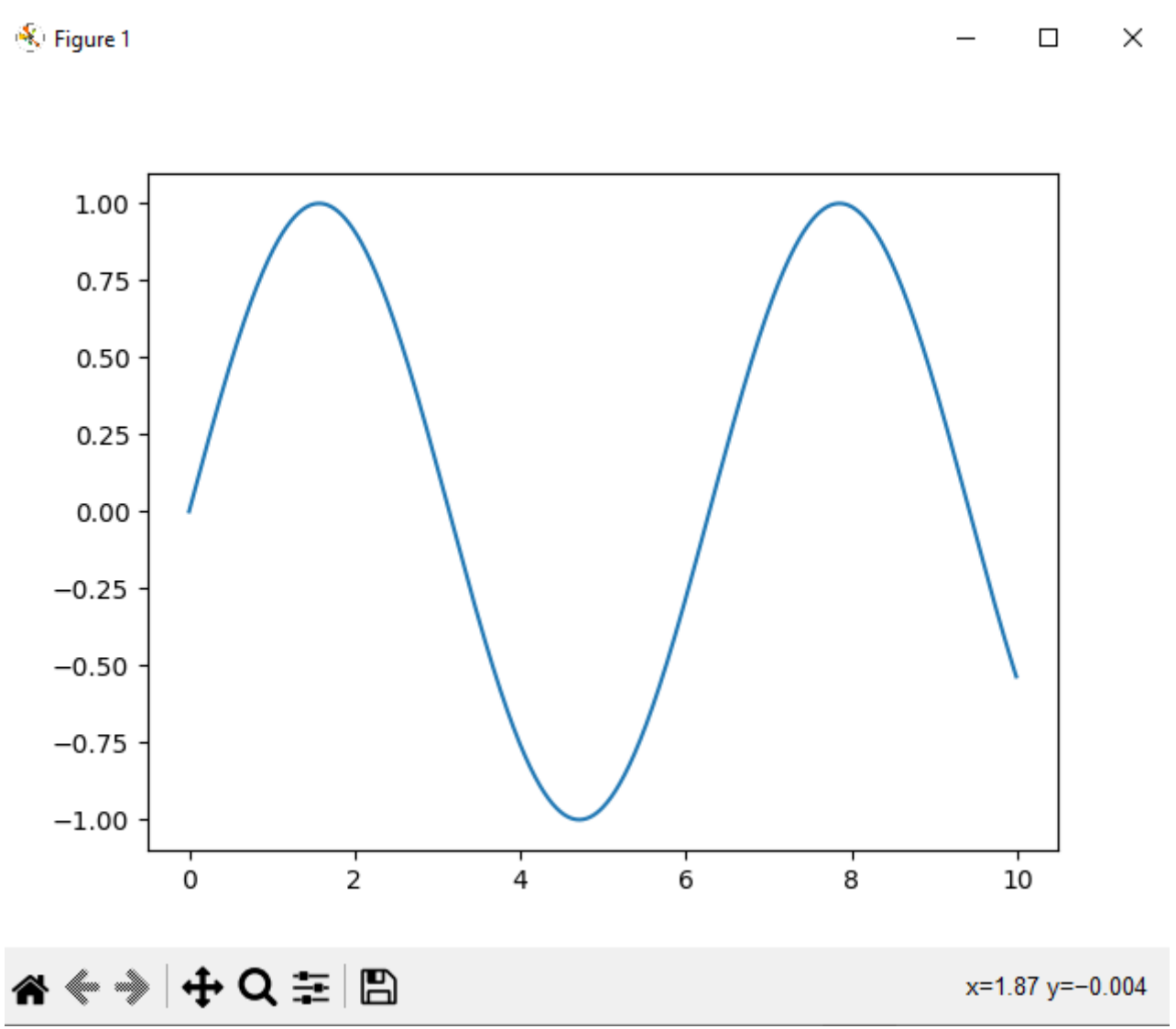
Pierwszy wykres

lec6_e.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/mo

File Edit Format Run Options Window Help

```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0,10,0.01)

y = np.sin(x)

plt.plot(x,y)

plt.show()
```

x = [0, 0.01, 0.02, ..., 9.99] jest tablicą
y = np.sin(x) jest również tablicą



Możesz zapisać wykres jako plik .pdf, .ps, .eps, etc.
Możesz przybliżyć, przesunąć, etc.

Lepszy wykres

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 14          # rozmiar czcionki

x = np.arange(0,10,0.01)
y = np.sin(x)

plt.plot(x, y, '-', color='red', linewidth=2.8)  # '-' ciągła linia

plt.xlabel('x')
plt.ylabel('sin(x)')       # podpisy osi

plt.axis([-2, 12, -1.5, 1.5])    # zakres osi

plt.grid(True)

plt.text(0,1.2, 'jakiś tekst')  # tekst zaczyna się w x=0 i y=1.2

plt.show()
```

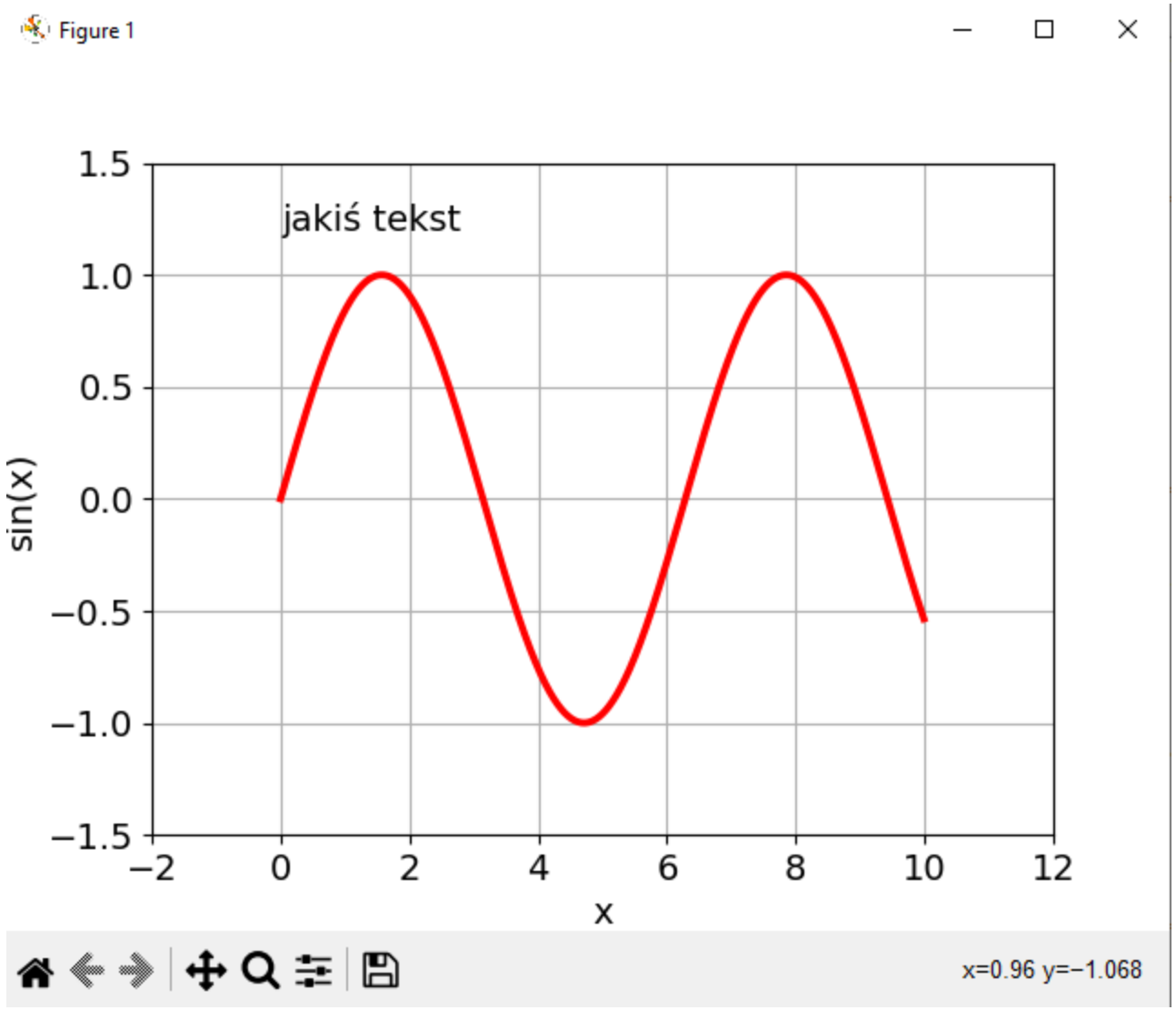


Figure 1
jakiś tekst
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
sin(x)
−2
0
2
4
6
8
10
12
x
x=0.96 y=−1.068

Linie

lec6_g.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/lec6_g.py

File Edit Format Run Options Window Help

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 14
plt.rcParams['lines.linewidth'] = 3   # grubość linii

x = [0,10]
y = np.array([1,1])

plt.plot(x, y, '-')
plt.plot(x, y*2, '--')
plt.plot(x, y*3, '-', dashes=[20,20])
plt.plot(x, y*4, '-', dashes=[15,11])
plt.plot(x, y*5, '-', dashes=[10,8,3,8])
plt.plot(x, y*6, '-', dashes=[10,5,3,3,3,5])
plt.plot(x, y*7, '-', dashes=[15,5,3,3,3,3,3,5])

plt.axis([-1, 11, 0, 8])

plt.show()
```

dashes = [linia, przerwa, linia, przerwa, …]

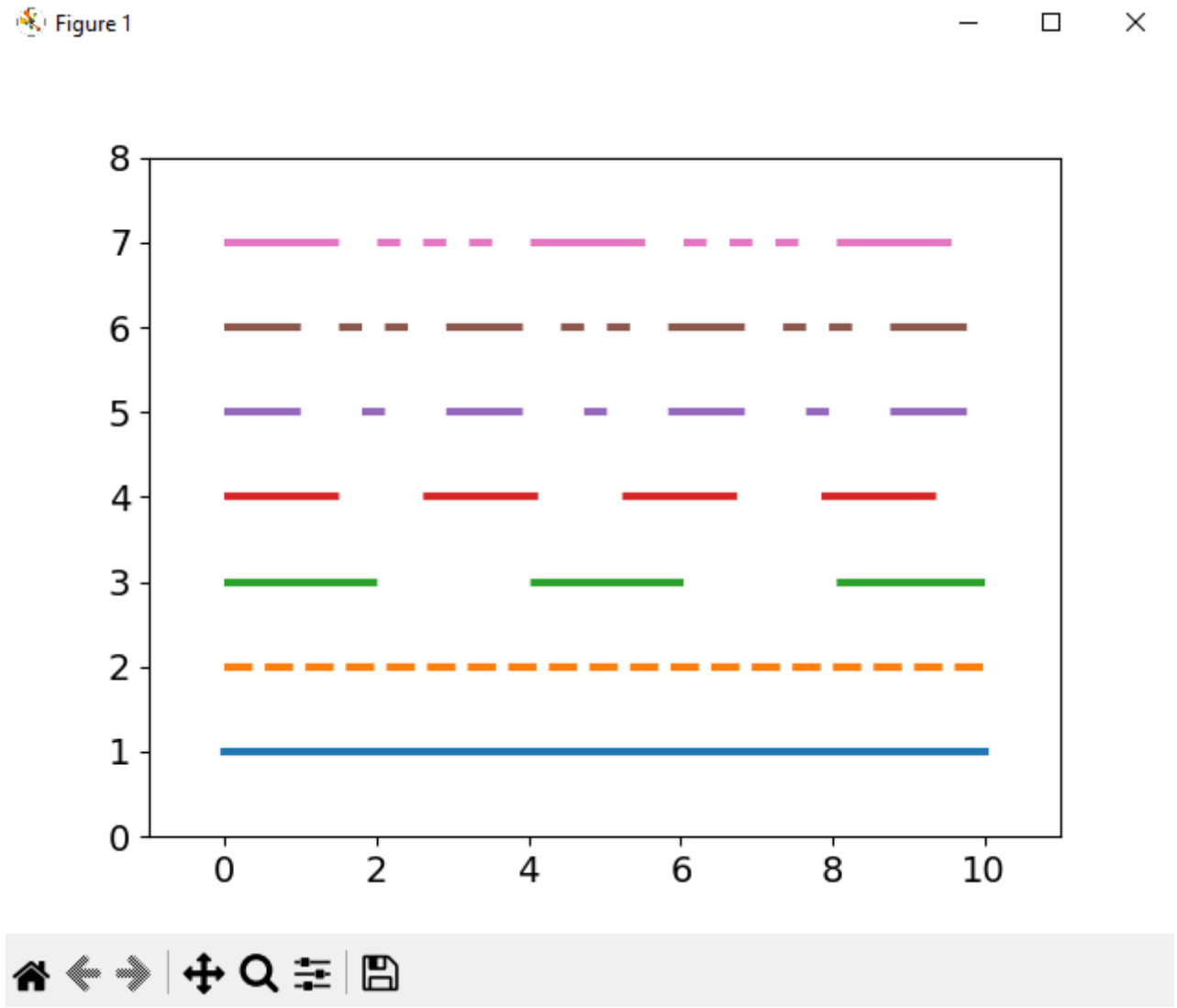


Figure 1
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10

Znaczniki

lec6_h.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/lec6_h.py (3.10.7)

File Edit Format Run Options Window Help

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 14

x = np.linspace(0,10,9)    # 9 liczb od 0 do 10
y = np.ones(9)             # [1,0,1.0,...,1.0]

plt.plot(x, y, 'o', markersize=9)     # kółko
plt.plot(x, y*2, 's', ms=9)           # prostokąt, ms=markersize
plt.plot(x, y*3, '^', ms=9)           # trójkąt
plt.plot(x, y*4, '>', ms=9)           # trójkąt
plt.plot(x, y*5, '<', ms=9)           # trójkąt
plt.plot(x, y*6, 'v', ms=9)           # trójkąt
plt.plot(x, y*7, 'p', ms=9)           # pięciokąt
plt.plot(x, y*8, '*', ms=9)           # gwiazdka
plt.plot(x, y*9, 'x', ms=9)           # x
plt.plot(x, y*10, '+', ms=9)          # +

plt.axis([-1, 11, 0, 11])

plt.show()
```

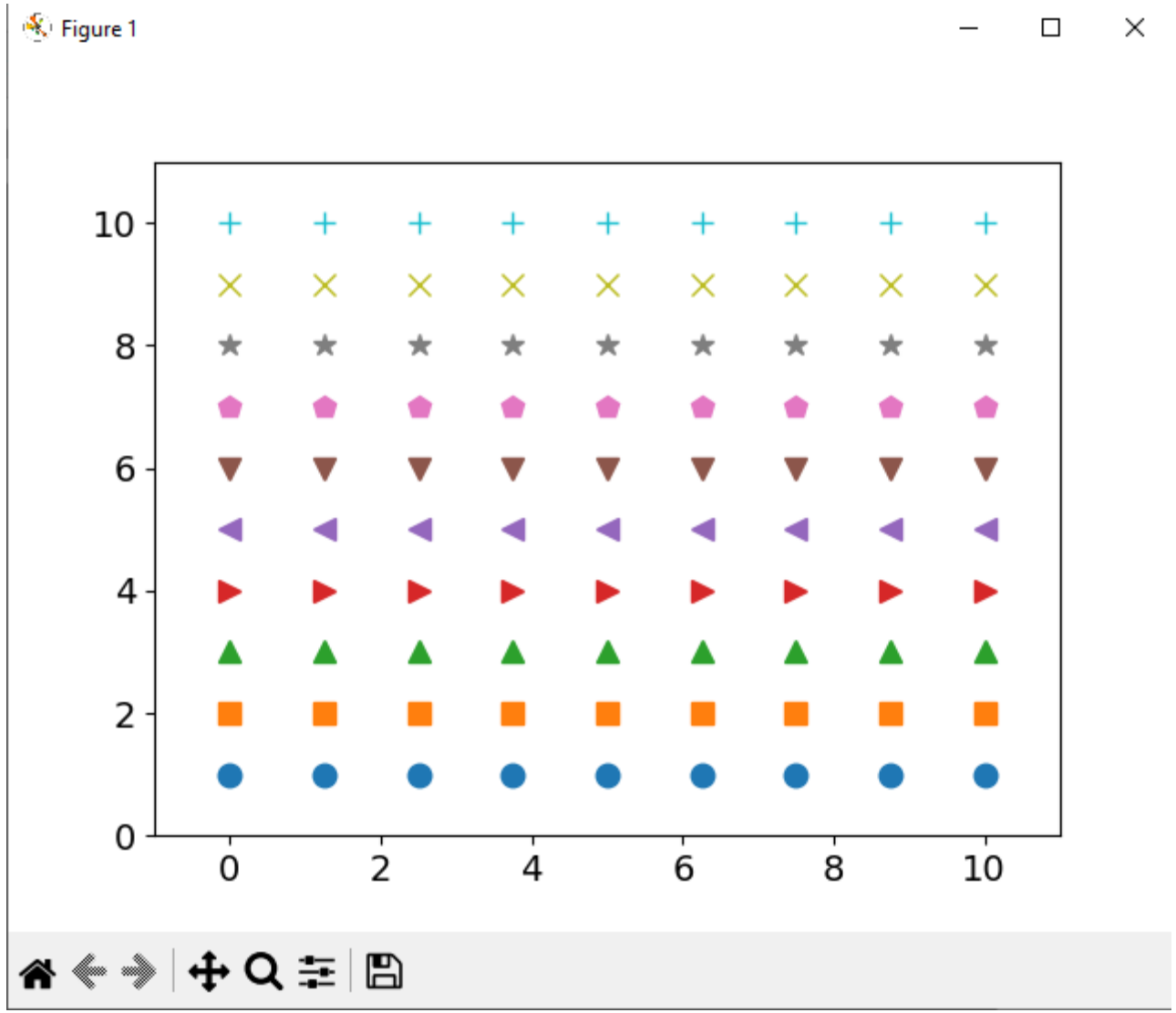


Figure 1
10
8
6
4
2
0
0
2
4
6
8
10

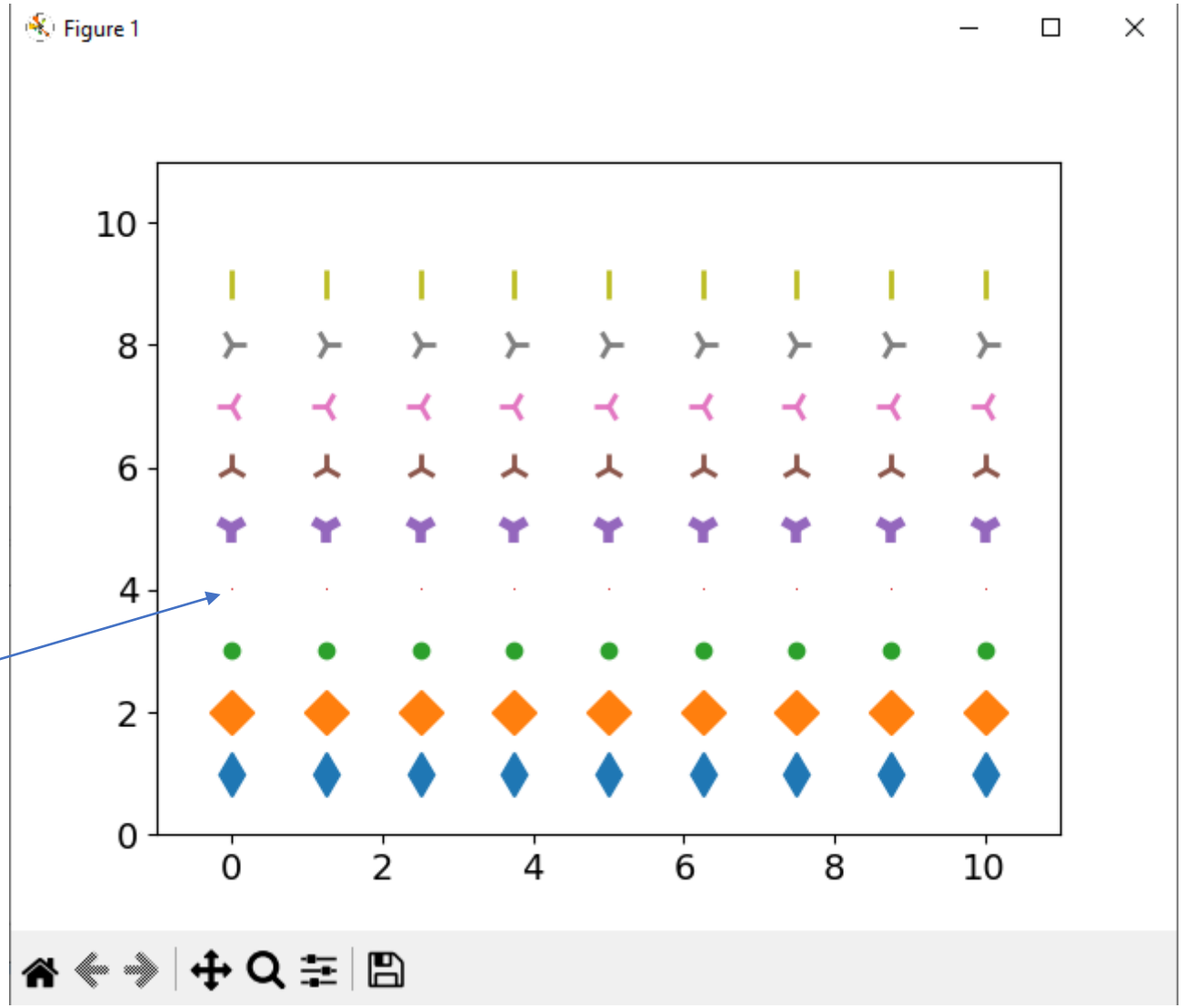
Więcej znaczników

lec6_i.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/lec6_i.py (3.10.7)

File Edit Format Run Options Window Help

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 14
plt.rcParams['lines.markersize'] = 12

x = np.linspace(0,10,9)
y = np.ones(9)

plt.plot(x, y, 'd')           # wąski diament
plt.plot(x, y*2, 'D')         # diament
plt.plot(x, y*3, '.')         # punkt
plt.plot(x, y*4, ',')         # piksel
plt.plot(x, y*5, '1', mew=4) # mew = marker edge width
plt.plot(x, y*6, '2', mew=2)
plt.plot(x, y*7, '3', mew=2)
plt.plot(x, y*8, '4', mew=2)
plt.plot(x, y*9, '|', mew=2)

plt.axis([-1, 11, 0, 11])

plt.show()
```

piksel



Kolory

lec6_j.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/lec

File Edit Format Run Options Window Help

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 14
plt.rcParams['lines.linewidth'] = 3

x = np.linspace(0,10,9)
y = np.ones(9)

plt.plot(x, y, color='k')
plt.plot(x, y*2, color=(0.5,0.5,0.9)) # (R, G, B)
plt.plot(x, y*3, color='DeepSkyBlue')
plt.plot(x, y*4, color='OrangeRed')
plt.plot(x, y*5, color='#00FF00')


plt.axis([-1, 11, 0, 6])

plt.show()
```

b=blue, g=green, r=red, c=cyan, m=magenta, y=yellow, k=black, w=white

Można stosować wszystkie kolory używane w HTML, zobacz:
https://www.w3schools.com/colors/colors_names.asp

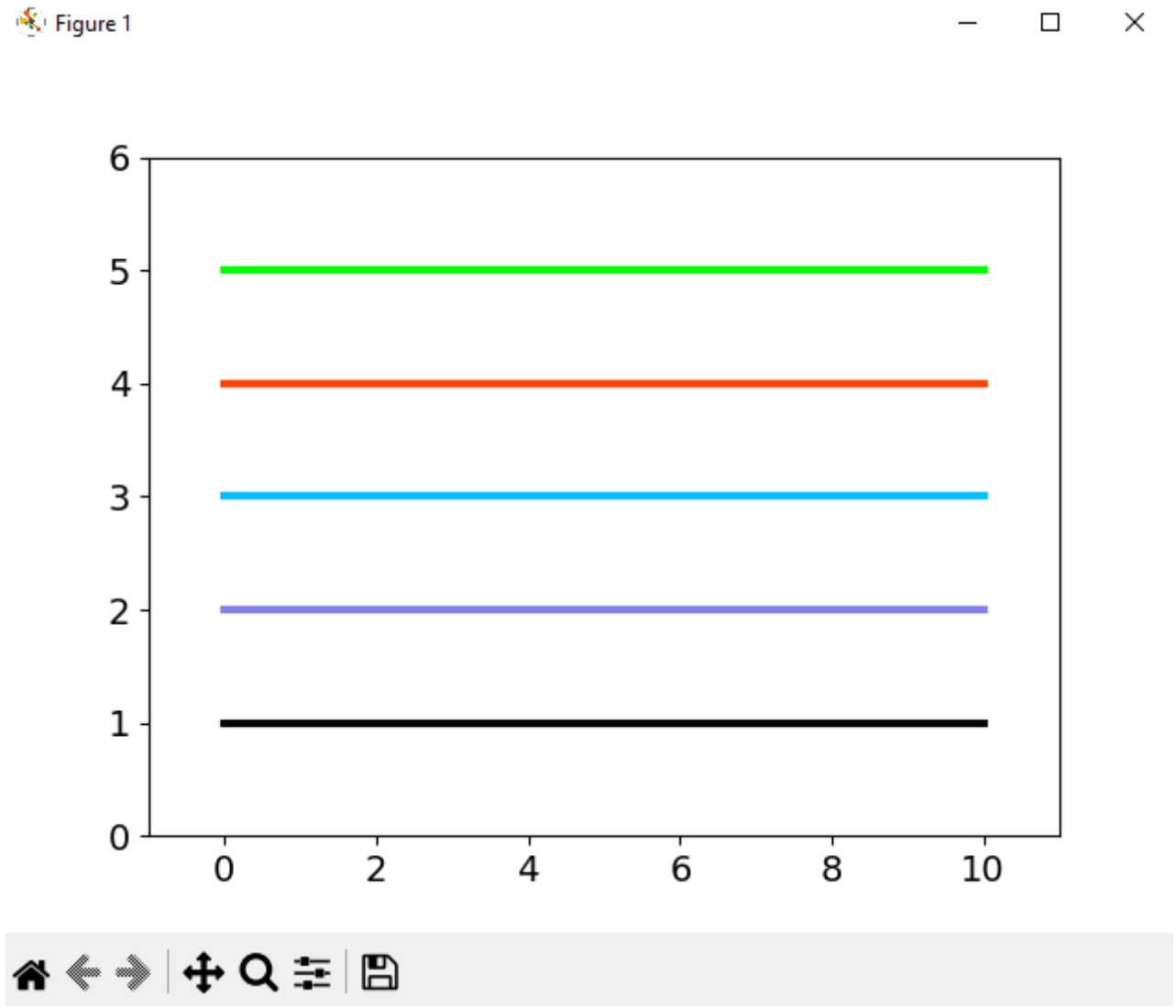


Figure 1
6
5
4
3
2
1
0
0
2
4
6
8
10

Znaczniki i linie

lec6_k.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/lec6_k.py (3.10.7)

File Edit Format Run Options Window Help

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 14
plt.rcParams['lines.linewidth'] = 3

x = np.linspace(0,10,9)
y = np.ones(9)

plt.plot(x, y, 'r-')            # czerwona ciągła linia
plt.plot(x, y*2, 'o-', ms=10)   # kółko i ciągła linia
plt.plot(x, y*3, 's--', ms=10) # prostokąt i przerywana linia

plt.plot(x, y*4, 'D-', color='k',
         ms=10, mfc='w', mec='b', mew=2)

plt.axis([-1, 11, 0, 5])
plt.show()
```

ms = marker size,
mfc = marker face color
mec = marker edge color
mew = marker edge width

sprawdź: mfc='None'

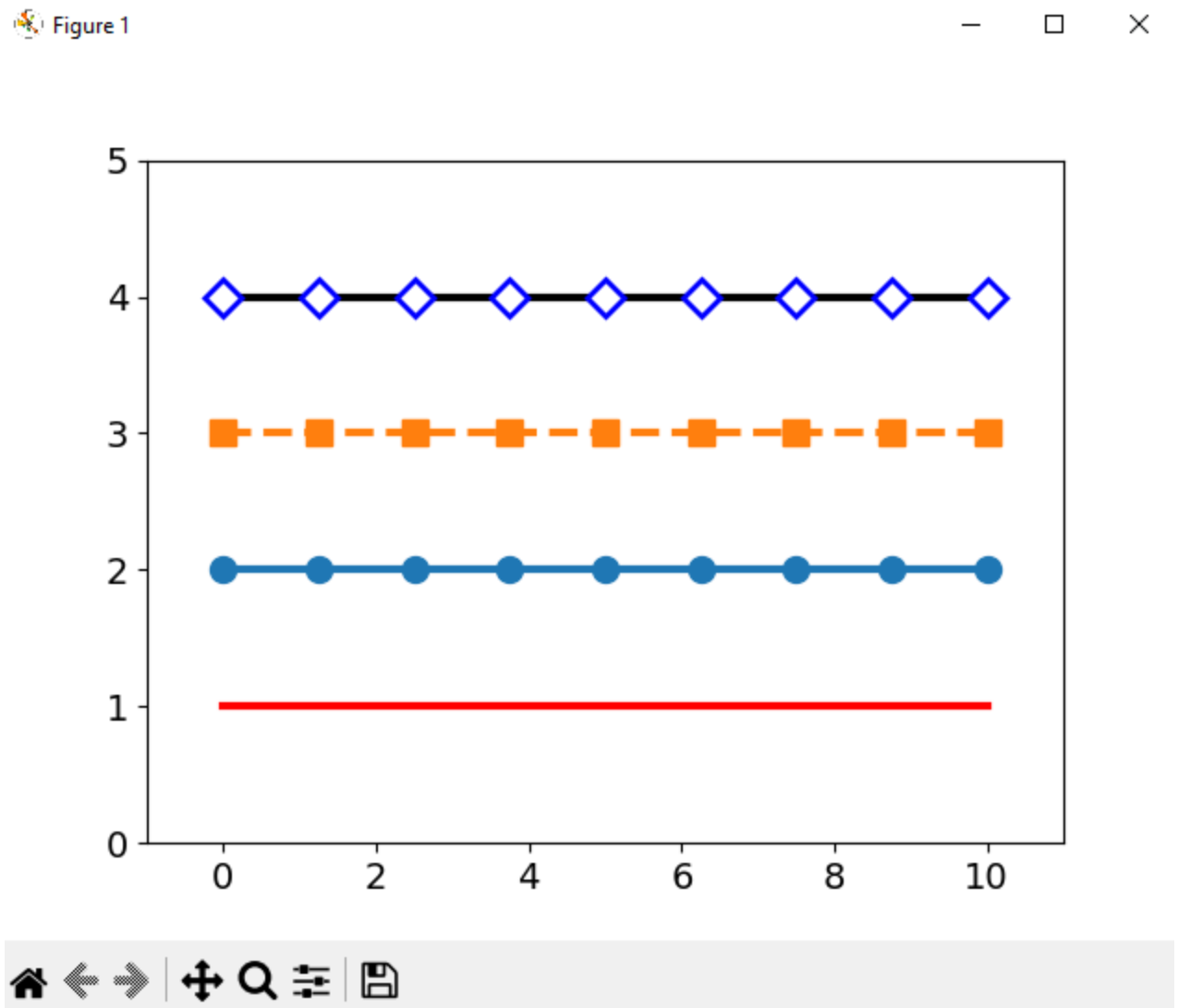


Figure 1
5
4
3
2
1
0
0
2
4
6
8
10

Legenda

lec6_l.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/lec6_l.py (3.10.7)

File Edit Format Run Options Window Help

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 14
plt.rcParams['legend.fontsize'] = 18  # rozmiar czcionki w legendzie

x = np.linspace(0,10,100)
y1 = np.sin(x)
y2 = np.exp(-x/5.)

plt.plot(x, y1, 'r-', lw=1.8, label='sin(x)')
plt.plot(x, y2, 'b-', lw=5, dashes=[8,3], label='exp(-x/5)')

plt.axis([-2, 12, -1.5, 2.0])
plt.legend(loc='upper right')    # położenie legendy
plt.show()
```

położenie = best, upper right, upper left, lower right, lower left, center left, center right, lower center, upper center, center

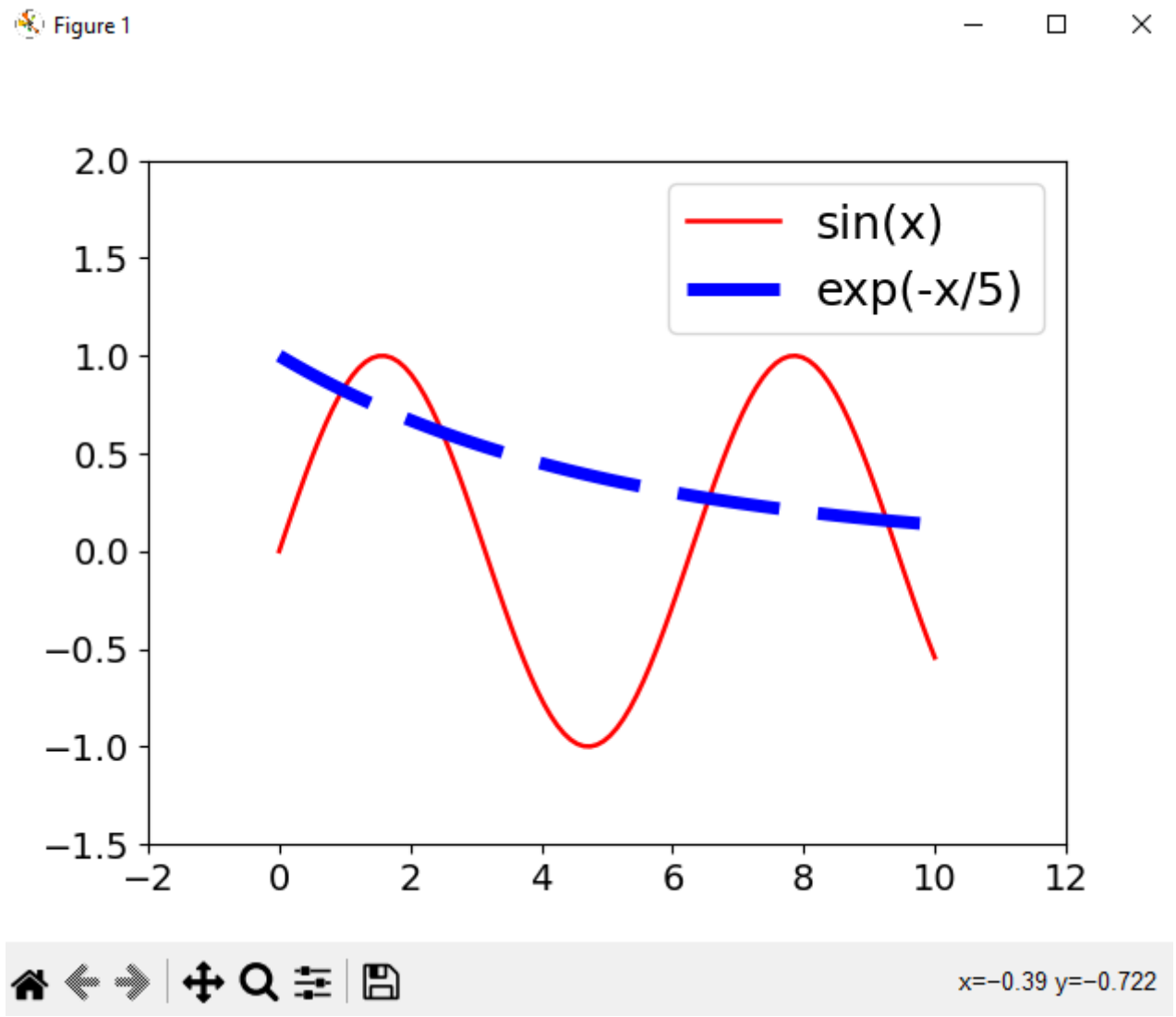


Figure 1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2
0
2
4
6
8
10
12
sin(x)
exp(-x/5)
x=−0.39 y=−0.722

Rozmiar rysunków i zapisywanie pliku

lec6_m.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/lec6_m.py (3.10.7)

File Edit Format Run Options Window Help

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'Times New Roman'    # krój czcionki
plt.rcParams['font.size'] = 26

plt.figure(figsize=(8, 6.9))  # rozmiar rysunku

x = np.linspace(0,10,30)
y = np.sin(x)

plt.plot(x, y, 'o-', color='k', lw=2, ms=10)

plt.axis([-0.5, 10.5, -1.7, 1.5])

plt.savefig('myfig.pdf',format='pdf', pad_inches=0.05)
plt.show()
```

lw = line width

ms = marker size

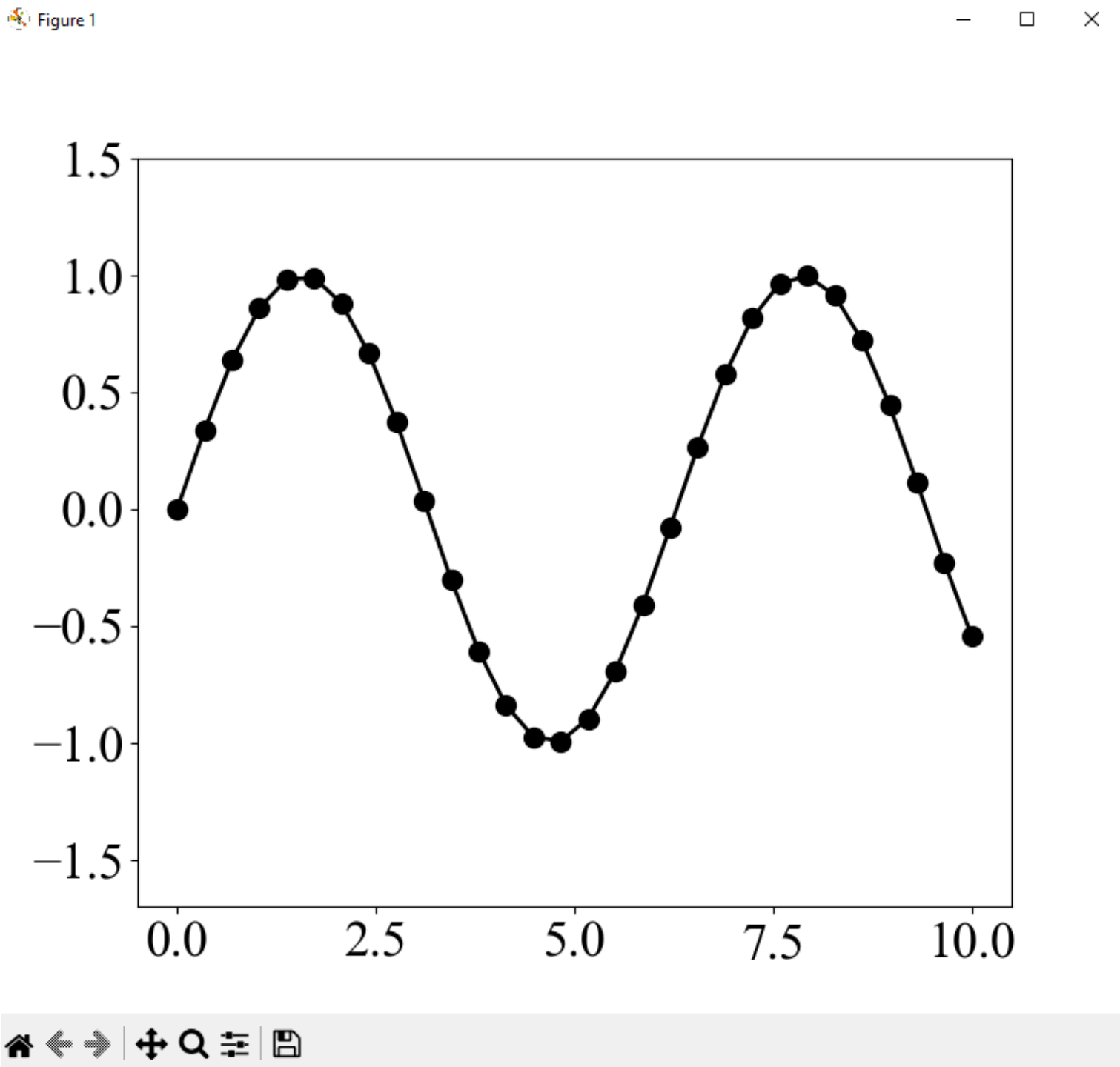
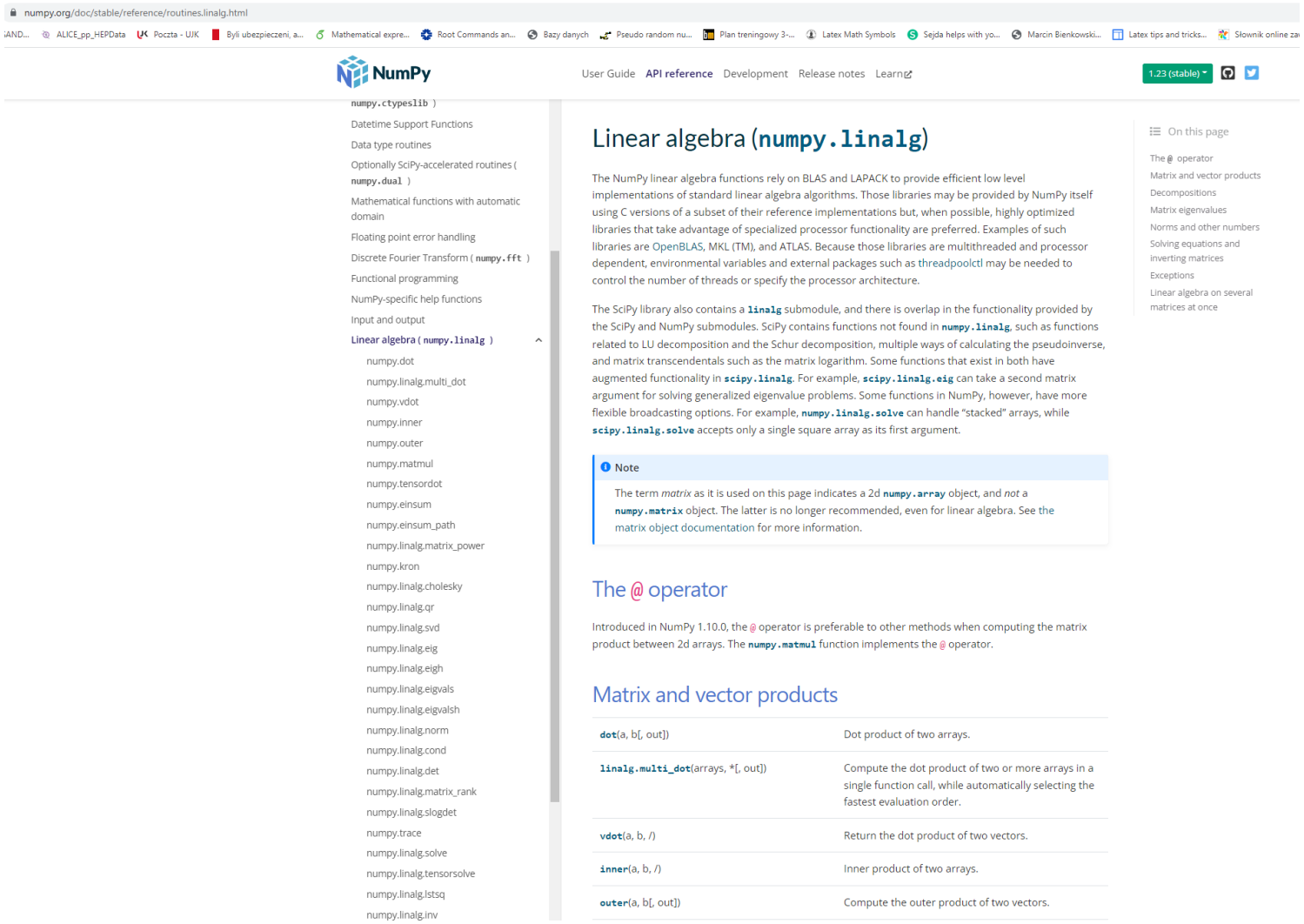


Figure 1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
2.5
5.0
7.5
10.0

More Stuff
(if you are interested)

Algebra w numpy

https://numpy.org/doc/stable/reference/routines.linalg.html



Rozwiązywanie równań, wartości własne, wektory własne, etc.

na przykład:

lec6_n.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/

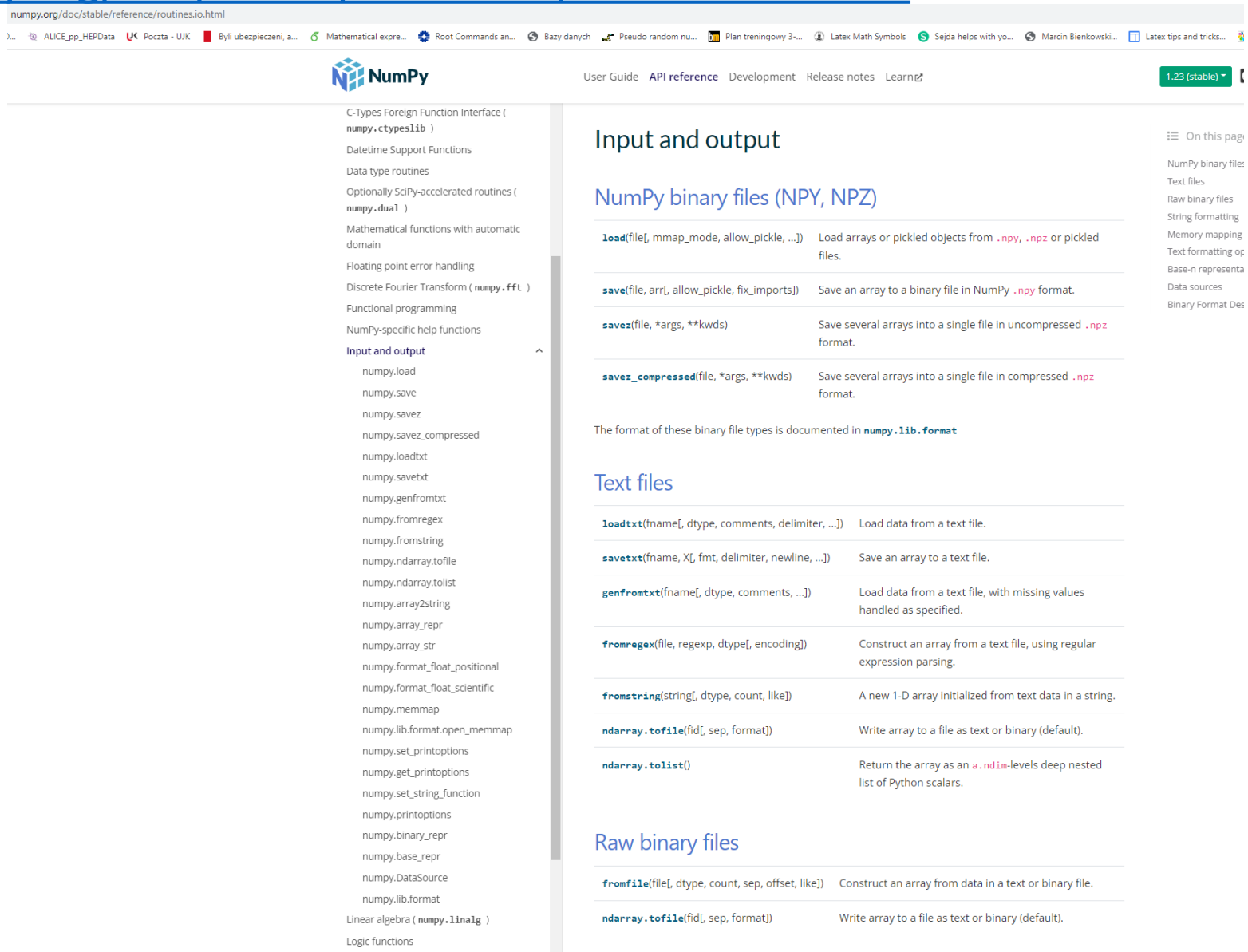
File Edit Format Run Options Window Help

```
import numpy as np

M = np.identity(3)*5
print(M)
print()

print(np.linalg.inv(M))   # macierz odwrotna
print()
print(np.linalg.det(M))   # wyznacznik macierzy
```

```
= RESTART: D:/CERNBox/zajecia,
[[5. 0. 0.]
 [0. 5. 0.]
 [0. 0. 5.]]

[[0.2 0.  0. ]
 [0.  0.2 0. ]
 [0.  0.  0.2]]

124.99999999999994
```

Zapisywanie i odczytywanie w numpy

https://numpy.org/doc/stable/reference/routines.io.html



na przykład:

```
import numpy as np

L = np.random.uniform(0,100,(5,2))
np.savetxt('data.txt', L)

G = np.loadtxt('data.txt')
print(G)
```

```
==== RESTART: D:/CERNBox/zajecia,
[[29.95346542 89.62627773]
 [73.02800627 89.69336579]
 [43.88002532 16.60657207]
 [36.5991981  10.90497709]
 [ 5.65016692 76.68034952]]
```

Lister - [d:\CERNBox\zajecia\Inzynieria_danych\Python\moje\wyklad_6\data.txt]

Plik Edytuj Opcje Kodowanie Pomoc

```
2.995346542297998482e+01 8.962627772664065162e+01
7.302800626784514293e+01 8.969336578880766808e+01
4.388002532349967311e+01 1.660657206785697326e+01
3.659919810230033477e+01 1.090497708988283954e+01
5.650166919055510384e+00 7.668034951565616097e+01
```

piksele

lec6_p.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_6/lec6_p.py (3.10.7)

File Edit Format Run Options Window Help

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 14

x = np.random.uniform(-10, 10, 100000)
y = np.random.uniform(-10, 10, 100000)

plt.plot(x, y, 'r,')  # przecinek oznacza kolorowanie pojedynczych pikseli

plt.axis([-11,11,-11,11])
plt.show()
```

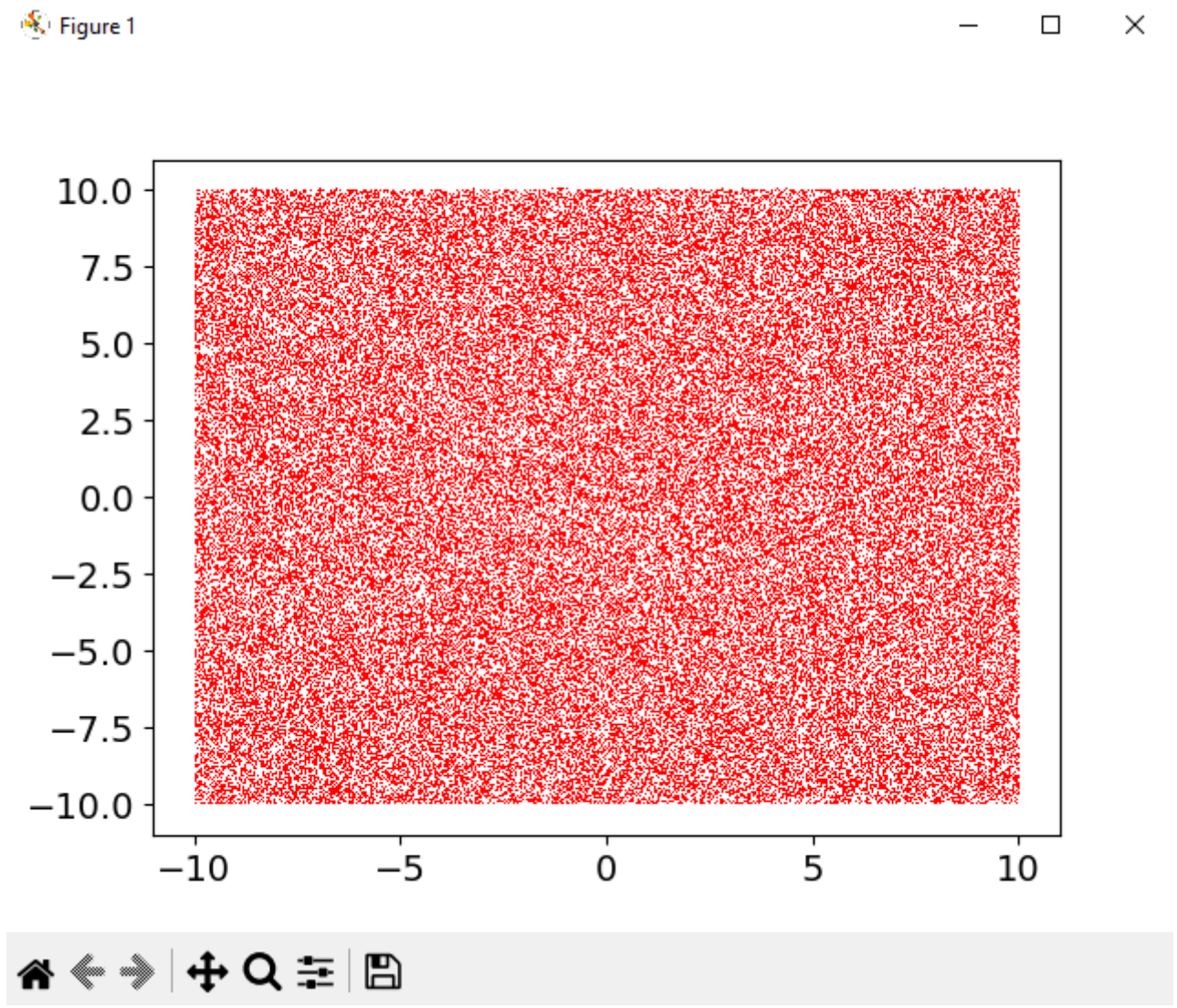


Figure 1
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10
−5
0
5
10