Przegląd alternatyw dla MATLAB

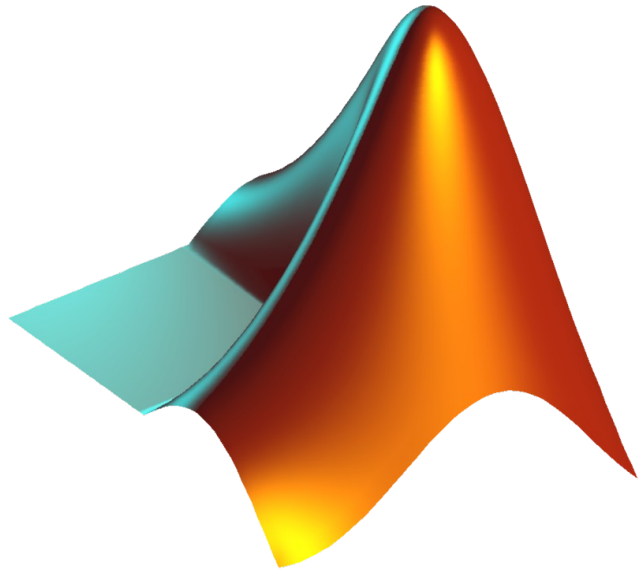
Bartosz Krakowiak

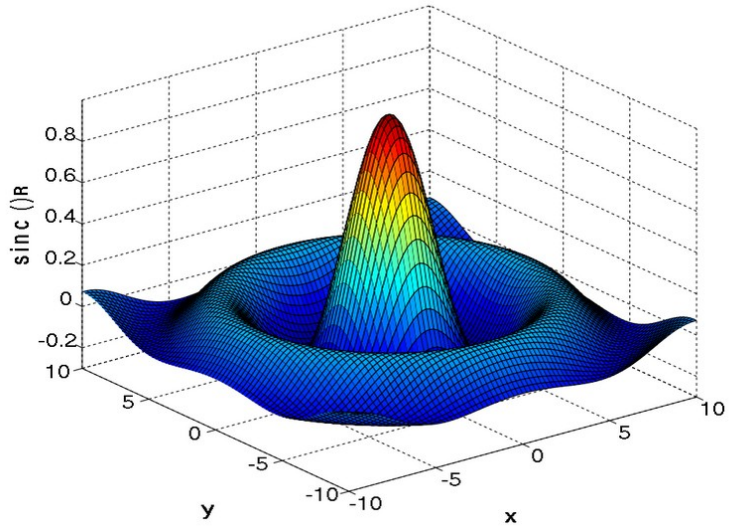
Czym jest MATLAB

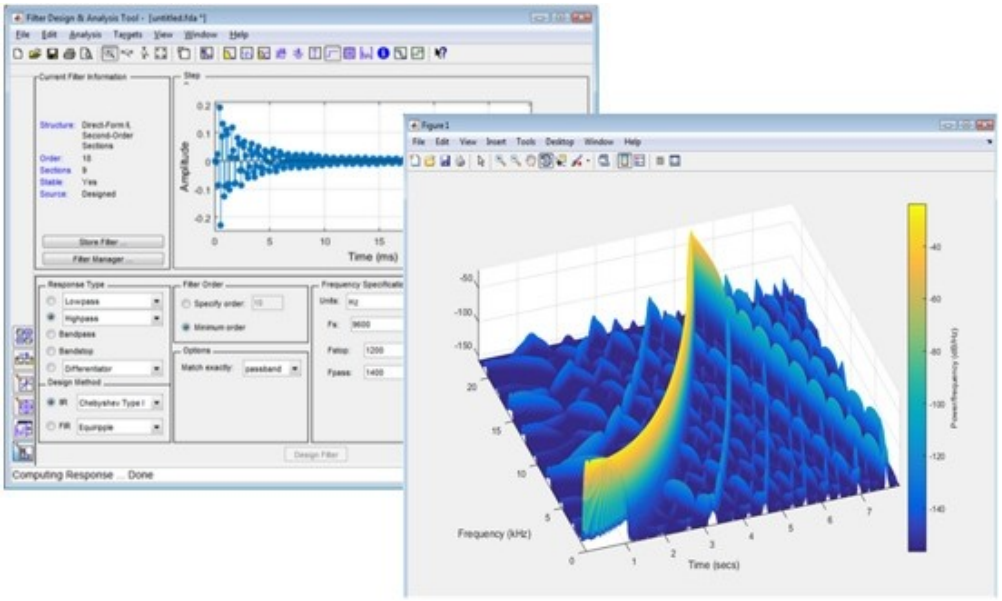
MATLAB – interaktywne środowisko do obliczeń i symulacji

- Wielu użytkowników

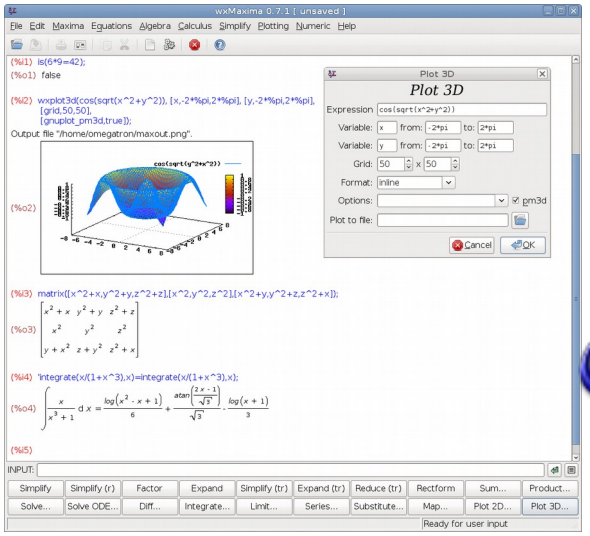
- Bardzo dużo zaimplementowanych funkcji



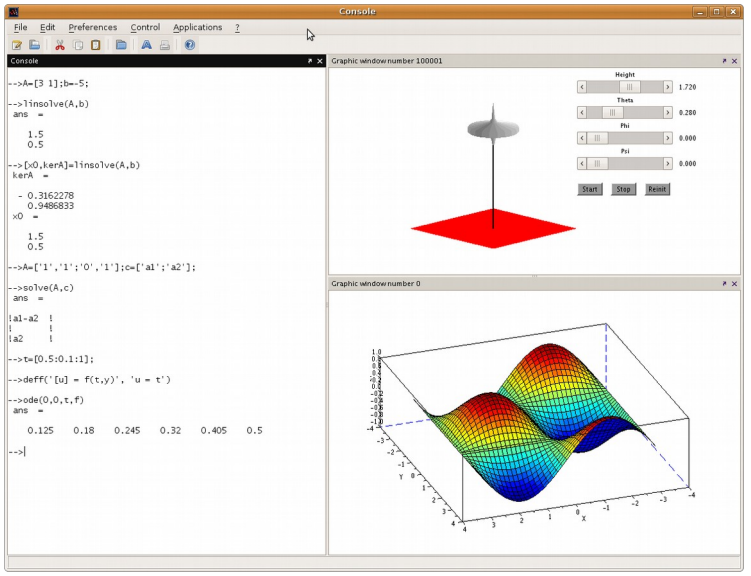


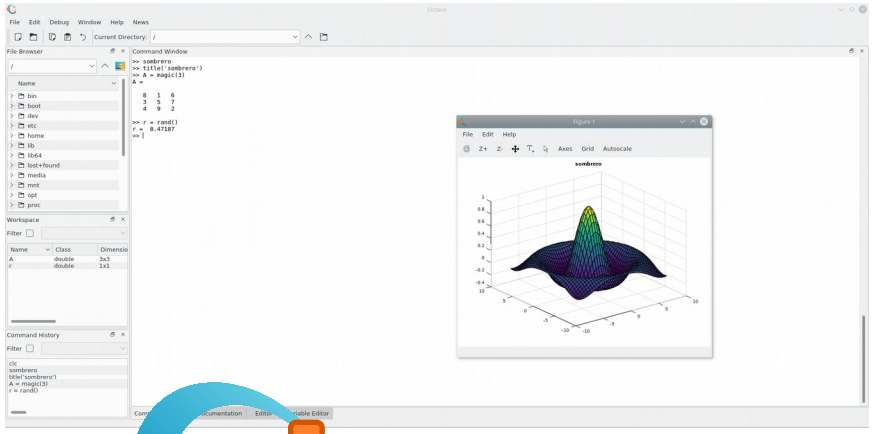


Wolne odpowiedniki



Maxima



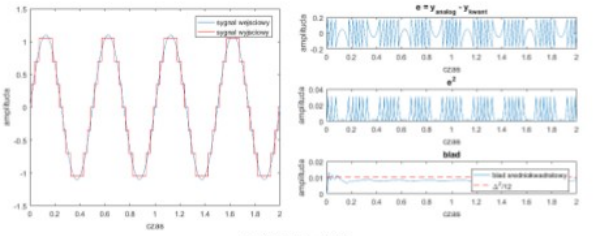


GNU Octave

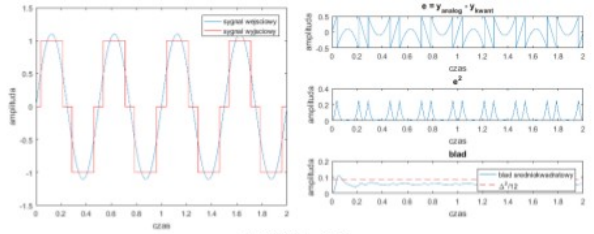
Scilab

Czego potrzebują studenci WEiT?

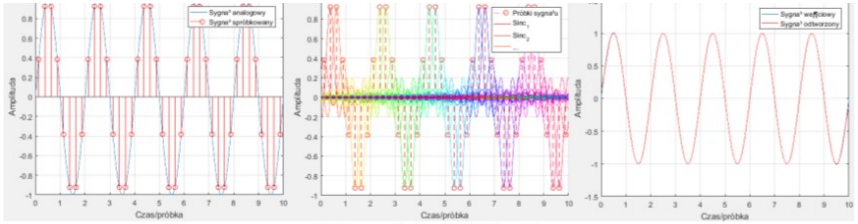
- **Wykresiki**
- **Operacje na macierzach**
- **DFT, FFT**
- **sin, cos, tan**



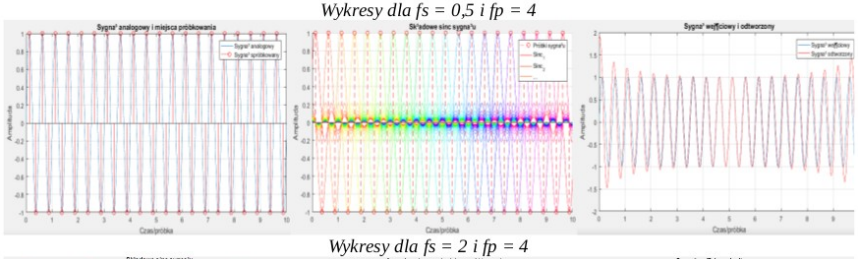
Δ=0,35 A=1,1



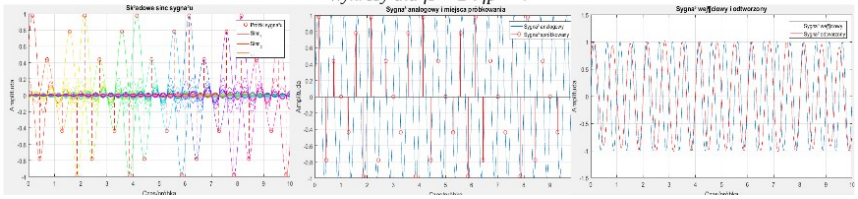
Δ=0,50 A=1,1



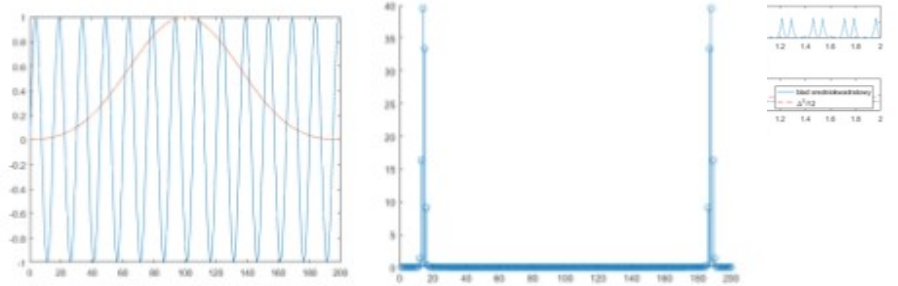
Wykresy dla fs = 0,5 i fp = 4



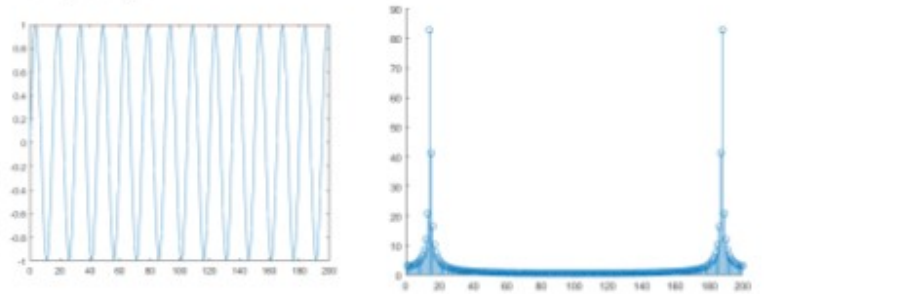
Wykresy dla fs = 2 i fp = 4



Analogicznie okno Blackmana:

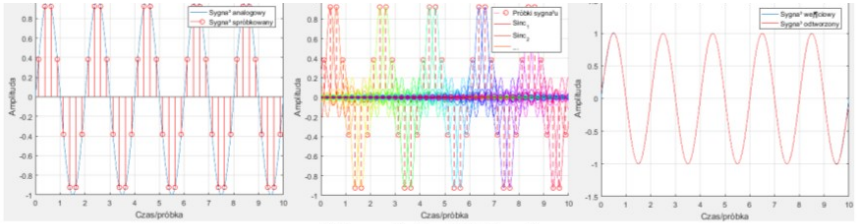


Okno prostokątne:

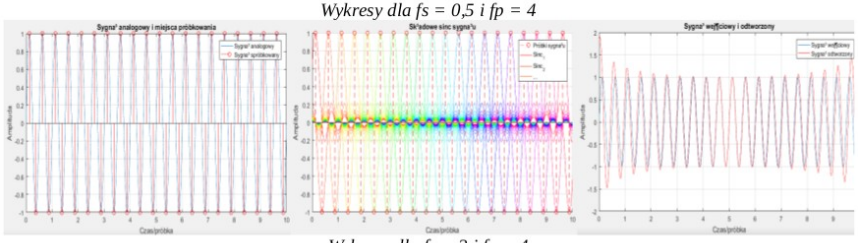


Czego potrzebują studenci WEiT?

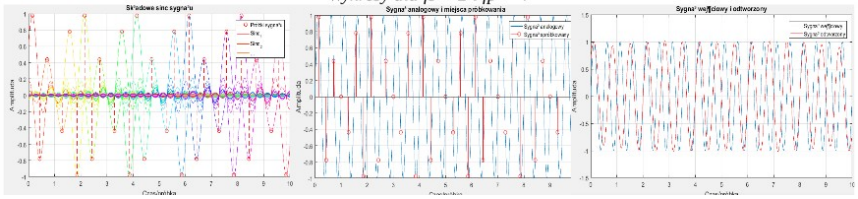
- Wykresiki
- Operacje na macierzach
- DFT, FFT
- sin, cos, tan
- ***Wykresiki***

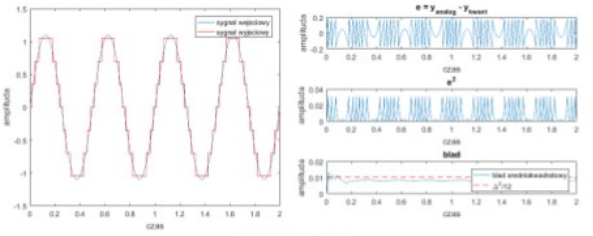


Wykresy dla fs = 0,5 i fp = 4

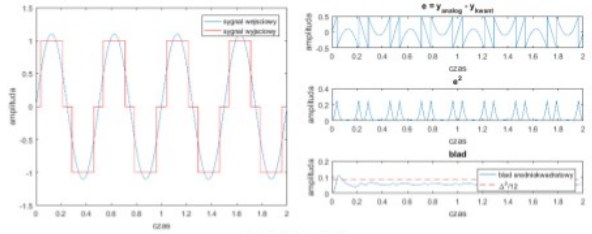


Wykresy dla fs = 2 i fp = 4



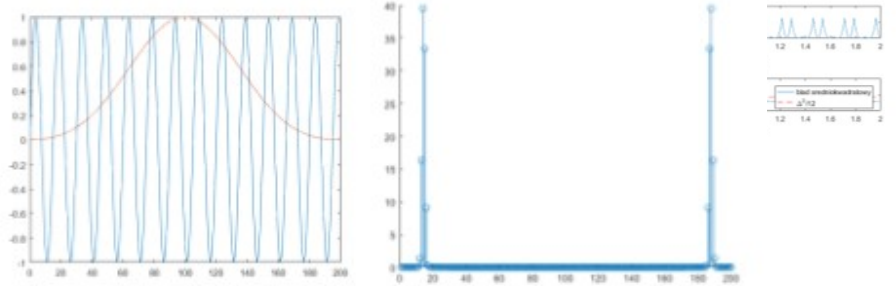


Δ=0,35 A=1,1

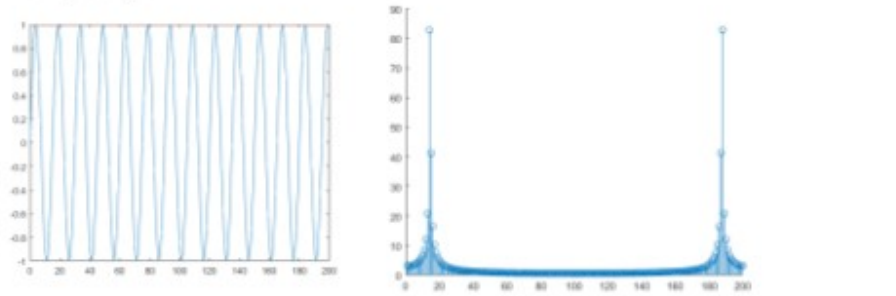


Δ=0,50 A=1,1

Analogicznie okno Blackmana:



Okno prostokątne:



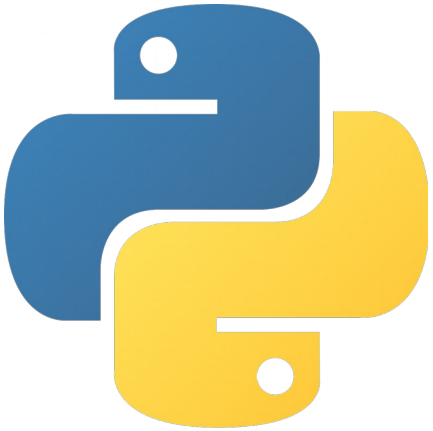
?!

Dlaczego Python?

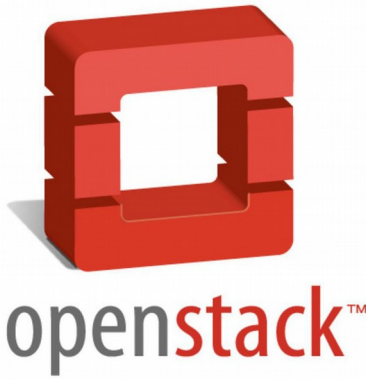
TensorFlow
Keras
PyTorch

MicroPython

Matplotlib
SciPy
NumPy
Pandas



Django
Flask



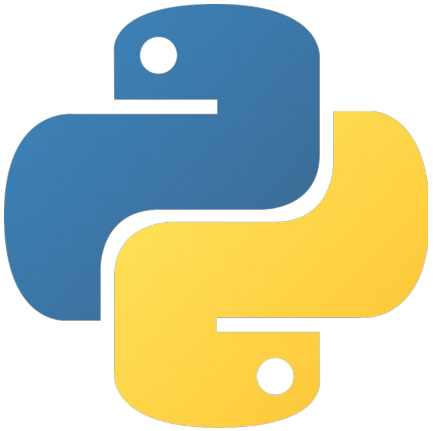
To co nas interesuje

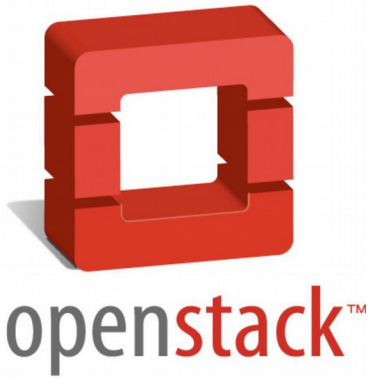
TensorFlow
Keras
PyTorch

MicroPython

Django
Flask

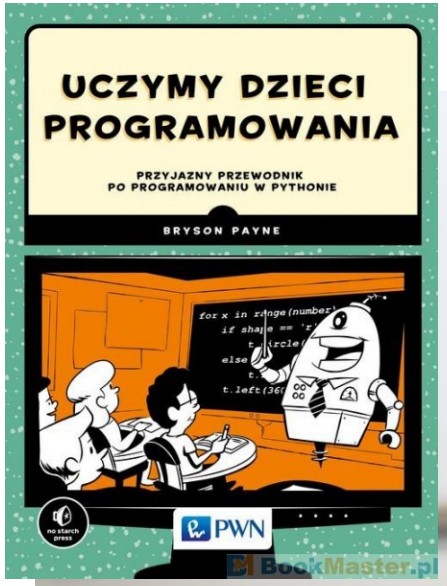
Matplotlib
SciPy
NumPy
Pandas





...

Nawet student WEiT da radę



Szybki start

```python
import sys

imie = input("Mam na imie: ")
wiek = input("Mam lat: ")
print("Jestem "+imie+", mam "+wiek+" lat i też programuje w pythonie!")
print("Jestem ", imie, ", mam ", wiek, " lat i też programuje w pythonie!")
print("Jestem {}, mam {} lat i też programuje w pythonie!".format(imie, wiek))

moja_lista = [1234, 234, 334]
moj_slownik = {"dwa": 2, "cztery": 4}
moja_krotka = ("a","b","c")

print(moja_lista, moj_slownik, moja_krotka)

moja_lista.append(9999)

# komentarz nad pętlą
for rzecz in moja_lista:
    print(rzecz)

for dynks in range(5):
    print("*"+"*"*dynks)

moj_slownik["nowe_slowo"] = "dużo"

for klucz, wartosc in moj_slownik.items():
    print("| {}: {}".format(klucz, wartosc))

wjednejlinii = [a*a for a in range(1,5)]
print(wjednejlinii)

pliczek = open("szczotkapasta.txt", 'r')
for linia in pliczek:
    #slowka = linia.split()
    for slowo in linia.split():
        sys.stdout.write(slowo + ' ')
    sys.stdout.write('\n')

def moja_funkcja(a, b):
    return a + b

class moja_klasa():

    def __init__(self, costam):
        self.moje = costam

    def pokaz(self):
        print("to moje: {}".format(self.moje))

wynik = moja_funkcja(100000000.000, 1)
print("tak policzyła moja funkcja: {}".format(wynik))

moja_instancja = moja_klasa("kluski z cebulą")
moja_instancja.pokaz()
```

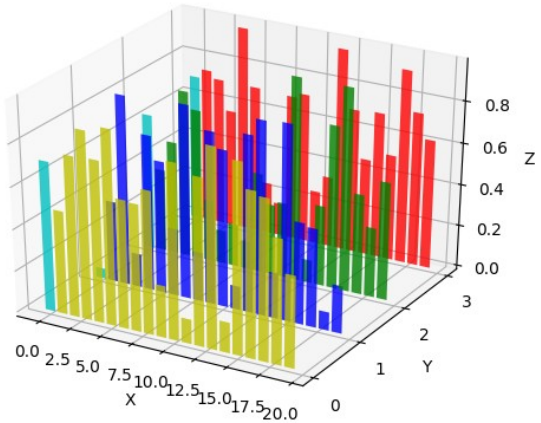
```
$ python3 kodzik.py
```

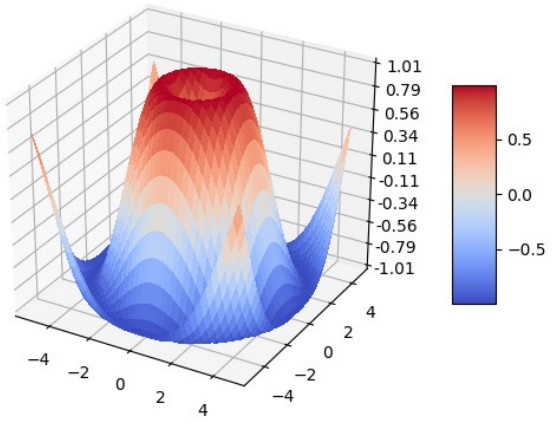
Matplotlib czyli wykresiki

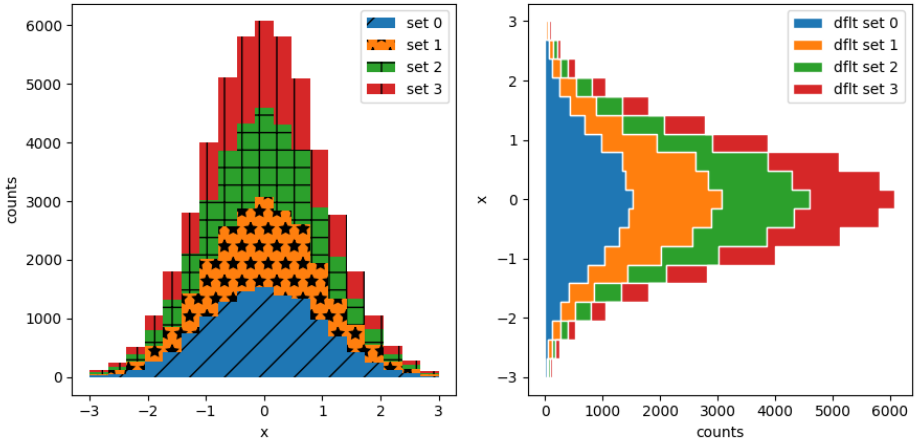
```python
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

t = np.arange(0.0, 4.0, 0.01)
s = 1 + np.sin(2 * np.pi * t)
# tworzymy 1 oś i 1 rycine
fig, ax = plt.subplots()

ax.plot(t,s)
ax.set(xlabel='czas (s)', ylabel='napięcie (mV)',
       title='Pierwszy wykres')
ax.grid()

fig.savefig("wykres.png")

# ----------------------------------
plt.figure()

plt.plot(s,t)
plt.title('Drugi wykres')
plt.grid()

# ----------------------------------
plt.figure()

x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
x3 = {"czerwoni": 500, "zieloni": 100, "niebiescy": 1000}

y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)

plt.subplot(2, 2, 1)
plt.plot(x1, y1)
plt.title('3 w jednym')

plt.subplot(2, 2, 2)
plt.plot(x2, y2, '.-')

plt.subplot(2, 1, 2)
plt.bar(x3.keys(), x3.values(), color='rgb')

# ----------------------------------
# axes - lista osi
fig2, axes = plt.subplots(2, 2)
axes[0][0].plot(x1, y1)
axes[0,1].pie(x3.values(), labels=x3.keys())
axes[1,0].bar(x3.keys(), x3.values(), color='rgb')
axes[1,1].scatter(x1, y1, c=y2, cmap='coolwarm')

plt.show()
```

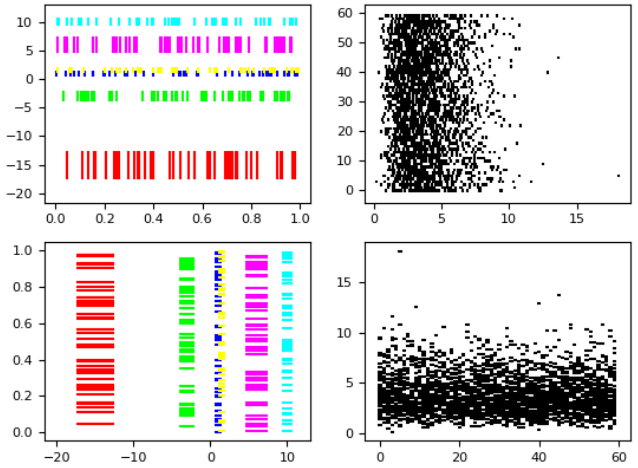
```
$ virtualenv --python=python3 venv
$ source venv/bin/activate
(venv)$ pip install matplotlib
(venv)$ python plots.py
```

Przykłady


0.8
0.6
0.4
0.2
0.0
Z
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0
X
0 1 2 3
Y

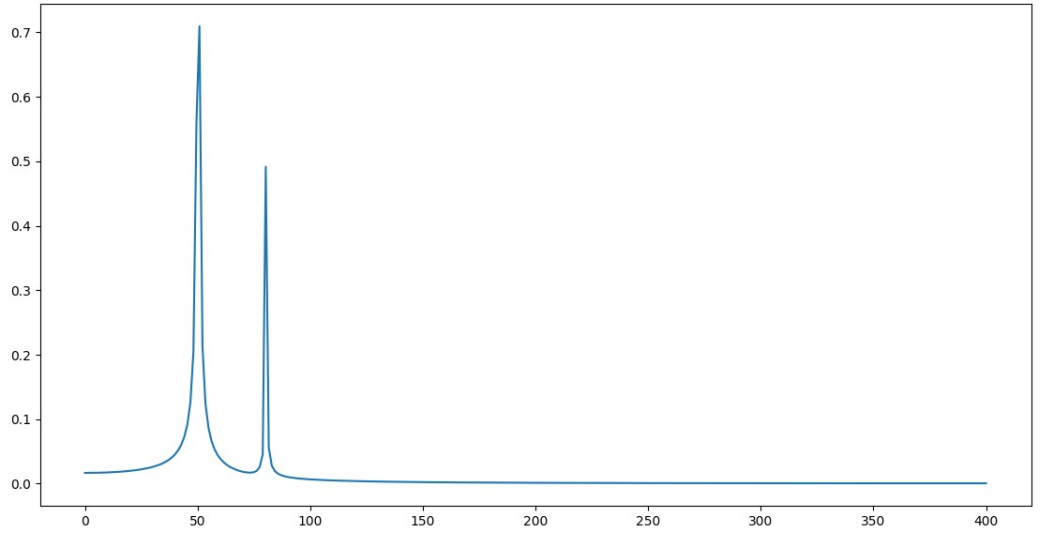

1.01
0.79
0.56
0.34
0.11
-0.11
-0.34
-0.56
-0.79
-1.01
0.5
0.0
−0.5
−4 −2 0 2 4
−4 −2 0 2 4


set 0
set 1
set 2
set 3
counts
x
dflt set 0
dflt set 1
dflt set 2
dflt set 3
x
counts

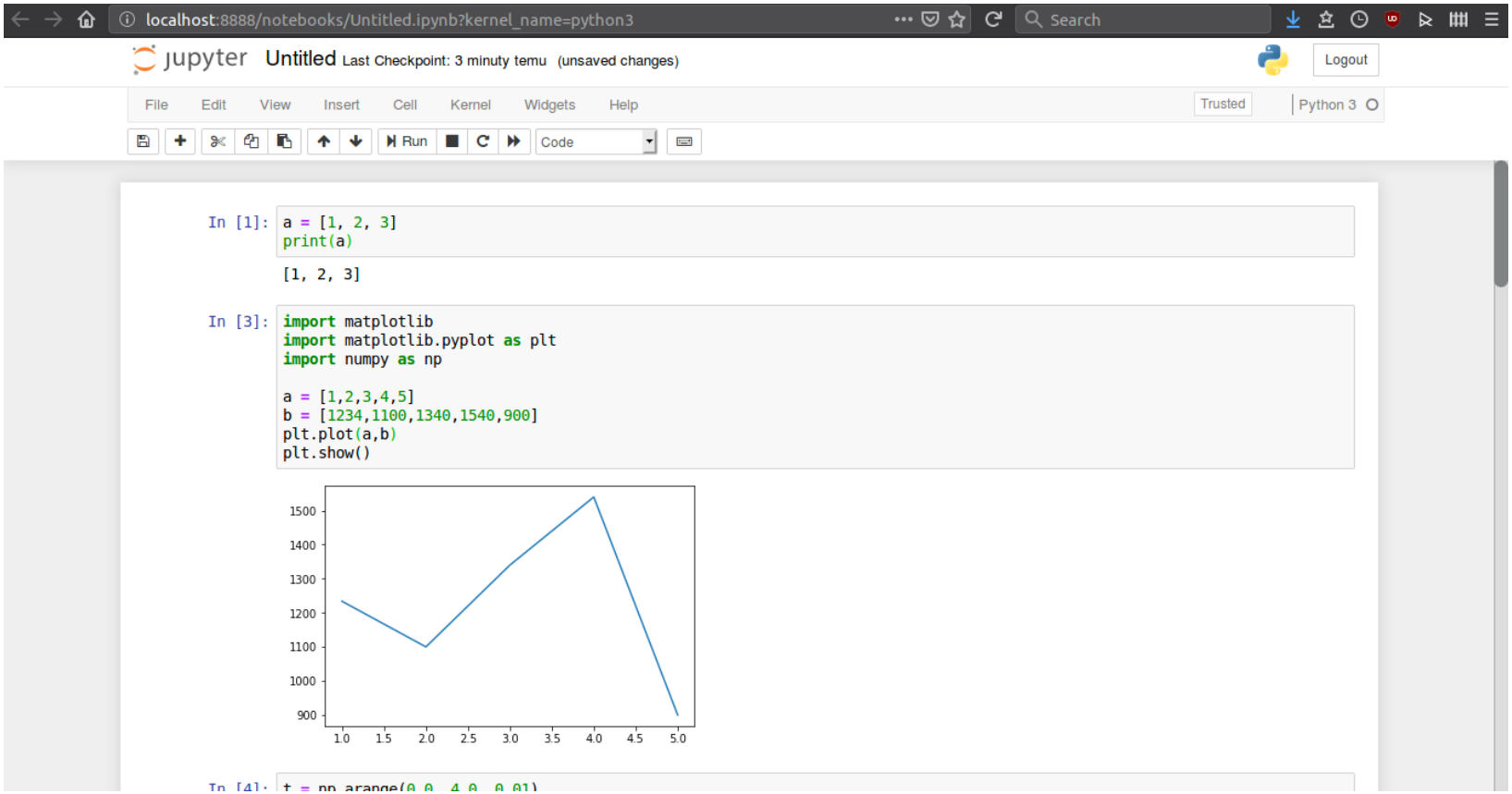


Numpy i SciPy

```
 1 import numpy as np
 2 import matplotlib.pyplot as plt
 3 import scipy.fftpack
 4
 5 N = 600
 6 T = 1.0 / 800.0
 7 x = np.linspace(0.0, N*T, N)
 8 y = np.sin(50.0 * 2.0*np.pi*x) + 0.5*np.sin(80.0 * 2.0*np.pi*x)
 9 yf = scipy.fftpack.fft(y)
10 xf = np.linspace(0.0, 1.0/(2.0*T), N/2)
11
12 fig, ax = plt.subplots()
13 ax.plot(xf, 2.0/N * np.abs(yf[:N//2]))
14 plt.show()
```

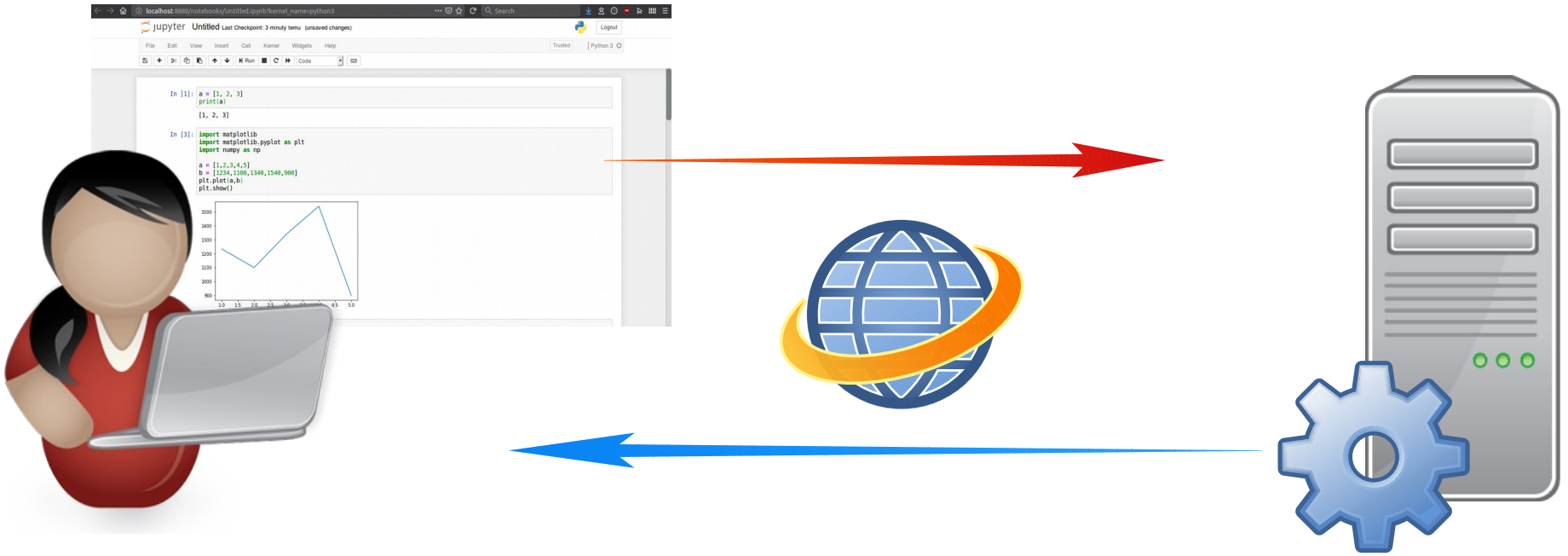


Jupyter Notebook




```
(venv)$ pip install jupyter
(venv)$ jupyter notebook
```

Jupyter Notebook



julialang.org

Dziękuję za uwagę

"You can't just copy-pase pseudocode into a program and expect it to work"

Odnośniki

http://maxima.sourceforge.net/

https://www.gnu.org/software/octave/

https://www.scilab.org/

https://docs.python.org/3/

https://matplotlib.org/

https://scipy.org/

http://www.numpy.org/

https://jupyter.org/

https://scipy-lectures.org

https://julialang.org/