

Laboratorium **Programowanie w języku Python 2**
Wydział Elektrotechniki Automatyki I Informatyki
Politechnika Świętokrzyska

Studia: Stacjonarne I stopnia	Kierunek: Informatyka
Data wykonania: 20.05.2021	Grupa: 3ID16B
Imię i nazwisko: Arkadiusz Więclaw	Temat ćwiczenia: Obliczenia

Zad 1:

Przykład 1

```
import numpy as np

a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print("Przykład 1 =")
print(a.ndim)
print(b.ndim)
print(c.ndim)
print(d.ndim)
```

Wynik:

Przykład 1 =

0

1

2

3

Przykład 2

```
import numpy as np
"""
Przykład tworzy tablice wypełniono danymi i pustą listę. Następnie
wrzyna elementy do listy
jeśli element jest całkowicie podzielny przez 2, ustawia wartość na
True, w przeciwnym razie False.
"""
arr = np.array([1, 2, 3, 4, 5, 6, 7])

# Create an empty list
filter_arr = []

# go through each element in arr
```

```

for element in arr:
    # if the element is completely divisble by 2, set the value to True,
    otherwise False
    if element % 2 == 0:
        filter_arr.append(True)
    else:
        filter_arr.append(False)

newarr = arr[filter_arr]
print("Przykład 2 =")
print(filter_arr)
print(newarr)

```

Wynik:

Przykład 2 =

[False, True, False, True, False, True, False]

[2 4 6]

Przykład 3

"""

Przykład tworzy macierz 1x4 i wyświetla macierz z oblicza wykładnicza dla każdego elementu tej macierzy.

"""

```

print("Przykład 3 =")
B = np.arange(6)
print("\nMacierz A\n",np.exp(B))
A = np.arange(9)
print("\nMacierz B\n",np.exp(A))

```

Wynik:

Przykład 3 =

Macierz A

[1. 2.71828183 7.3890561 20.08553692 54.59815003
148.4131591]

Macierz B

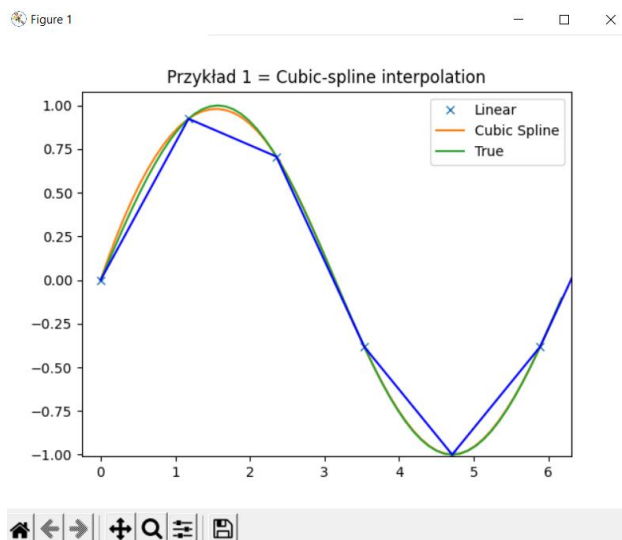
```
[1.00000000e+00 2.71828183e+00 7.38905610e+00 2.00855369e+01  
5.45981500e+01 1.48413159e+02 4.03428793e+02 1.09663316e+03  
2.98095799e+03]
```

Zad 2:

Przykład 1

```
from scipy.fft import fft, ifft  
from scipy.fft import ifftn  
import matplotlib.pyplot as plt  
import matplotlib.cm as cm  
import scipy.integrate as integrate  
import scipy.special as special  
import matplotlib.pyplot as plt  
from scipy.interpolate import interp1d  
from numpy import sqrt, sin, cos, pi  
from scipy.integrate import quad  
import numpy as np  
from scipy import interpolate  
from scipy.integrate import quad, dblquad  
from scipy.integrate import simps  
from scipy.fft import fft, fftfreq  
  
"""  
Przykład wyświetla wykres dla interpolacji spline.  
"""  
  
x = np.arange(0, 3*np.pi+np.pi/4, 3*np.pi/8)  
y = np.sin(x)  
tck = interpolate.splrep(x, y, s=0)  
xnew = np.arange(0, 2*np.pi, np.pi/30)  
ynew = interpolate.splev(xnew, tck, der=0)  
plt.figure()  
plt.plot(x, y, 'x', xnew, ynew, xnew, np.sin(xnew), x, y, 'b')  
plt.legend(['Linear', 'Cubic Spline', 'True'])  
plt.axis([-0.25, 6.31, -1.01, 1.08])  
plt.title('Przykład 1 = Cubic-spline interpolation')  
plt.show()
```

Wynik:



Przykład 2

"""

Przykład pokazuje użycie podwójnej integracji do obliczenia kilku wartości.

"""

```
def I(n):
    return dblquad(lambda t, x: np.exp(-x*t)/t**n, 0, np.inf, lambda x:
1, lambda x: np.inf)
print("Przykład 2 =")
print(I(15))
print(I(20))
print(I(17))
print(I(5))
print(I(81))
print(I(10))
```

Wynik:

Przykład 2 =

```
(0.066666666666664971, 1.3893194210507051e-08)
(0.0499999999999984876, 2.2954740049285213e-09)
(0.05882352941174849, 1.2219001246148928e-08)
(0.2000000000189363, 1.3682975855986131e-08)
(0.012345679012355594, 9.809252342492316e-09)
(0.09999999999998103, 1.247933706369846e-09)
```

Przykład 3

```
"""
Przykład całkowanie za pomocą metody simpsona.
"""
def f1(x):
    return x**2
def f2(x):
    return x**3
x = np.array([14,29,18,53])
y1 = f1(x)
I1 = simpson(y1, x)
print("Przykład 3 =")
print(I1)
```

Wynik:

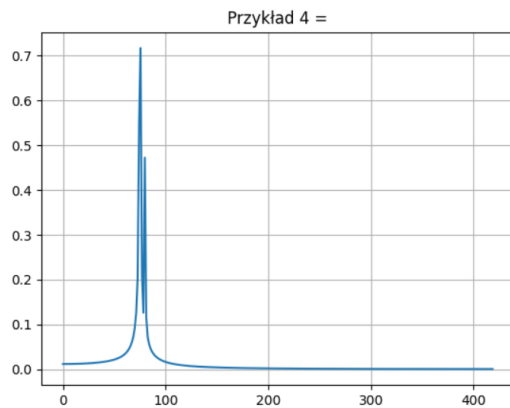
Przykład 3 =
52565.16666666667

Przykład 4

```
"""
Przykład przedstawia wykres transformaty fourierasumy dwóch sinusów.
"""
N = 600
T = 1.0 / 840.0
x = np.linspace(0.0, N*T, N, endpoint=False)
y = np.sin(75.0 * 2.0*np.pi*x) + 0.5*np.sin(80.0 * 2.0*np.pi*x)
yf = fft(y)
xf = fftfreq(N, T)[:N//2]
plt.plot(xf, 2.0/N * np.abs(yf[0:N//2]))
plt.title('Przykład 4 =')
plt.grid()
plt.show()
```

Wynik:

Figure 1



Zad 3:

Przykład 1

```
from sympy import *
x, y, z = symbols('x y z')
#Przykład liczy granice funkcji.
print("Przykład 1 =")
print(limit(sin(x)/x, x, 0))
print(limit(cos(x)/x, x, 0))
```

Wynik:

Przykład 1 =

1

0o

Przykład 2

```
"""
```

Przykład przedstawia liczenie pochodnych z wykorzystaniem biblioteki sympy.

```
"""
```

```
x, y, z = symbols('x y z')
init_printing(use_unicode=True)
print("Przykład 2 =")
print("Pochodna z cos = ", diff(cos(x), x))
print("Pochodna z tan = ", diff(tan(x), x))
print("Pochodnia z wykladnika x**3 = ", diff(exp(x**3), x))
```

Wynik:

Przykład 2 =

Pochodna z $\cos = -\sin(x)$

Pochodna z $\tan = \tan(x)^2 + 1$

Pochodnia z wykladnika $x^3 = 3x^2 \cdot \exp(x^3)$

Zad 4:

```
import pandas as pd
import matplotlib.pyplot as plt
"""
Przyklad wykorzystania bilioiteki pandas i matplotlib.
"""

data = pd.read_csv("rpi_describe.csv")
print("piec pierwszych rekordow ")
print(data.head(5))
print("\npiec ostatnich rekordow ")
print(data.tail(5))
print("\npostawowe statystyki ramki danych ")
print(data.describe())
data['rounded'] = data['weight'].round()
print("\npo dodaniu kolumny")
print(data)
print("\n po dodaniu kolumny ")
data['diff'] = (data['weight'] - data['rounded']).round()
print(data)
count = data['rounded'].count()
mean = data['rounded'].mean()
median = data['rounded'].median()
std = data['rounded'].std()
rng = data['rounded'].min() , data['rounded'].max()
```

Wynik:

piec pierwszych rekordow

	weight
0	20.440
1	20.244
2	20.549
3	20.755
4	20.740

piec ostatnich rekordow

	weight
9995	20.419
9996	20.554
9997	20.768
9998	20.829
9999	20.667

postawowe statystyki ramki danych

	weight
count	10000 000000
mean	20.499212
std	0.199874
min	19.752000
25%	20.365000
50%	20.500000
75%	20.635000
max	21.171000

po dodaniu kolumny

	weight	rounded
0	20.440	20.0
1	20.244	20.0
2	20.549	21.0
3	20.755	21.0
4	20.740	21.0
...
9995	20.419	20.0
9996	20.554	21.0
9997	20.768	21.0
9998	20.829	21.0
9999	20.667	21.0

[10000 rows x 2 columns]

po dodaniu kolumny

	weight	rounded	diff
0	20.440	20.0	0.0
1	20.244	20.0	0.0
2	20.549	21.0	-0.0
3	20.755	21.0	-0.0
4	20.740	21.0	-0.0
...
9995	20.419	20.0	0.0
9996	20.554	21.0	-0.0
9997	20.768	21.0	-0.0
9998	20.829	21.0	-0.0
9999	20.667	21.0	-0.0

[10000 rows x 3 columns]