Laboratorium metod numerycznych

Temat 1. Podstawy technik obliczeniowych w języku Python cz 1

UWAGA: Kody programów uruchom w Jupyter Notebook (iPython notebook).

Zadanie 1. Uruchom poniższy kod. Kod demonstruje użycie kilku funkcji numerycznych pochodzących z biblioteki. Podstawowa biblioteka funkcji numerycznych to NumPy. W rezultacie działania programu obliczono i wydrukowano wartości 4 funkcji trygonometrycznych.

```
#import biblioteki NumPy do przestrzeni nazw np;
#funkcje dostepne sa z poziomu przestrzeni nazw 'np'
import numpy as np
x=np.pi/3#tworz zmienna x
#oblicz wartosci f trygonometrycznych
print("sin(%f)=%f" % (x,np.sin(x)))
print("cos(%f)=%f" % (x,np.cos(x)))
print("tan(%f)=%f" % (x,np.tan(x)))
print("ctg(%f)=%f" % (x,1.0/np.sin(x)))
```

Rezultat uruchomienia

```
sin(1.047198)=0.866025
cos(1.047198)=0.500000
tan(1.047198)=1.732051
ctg(1.047198)=1.154701
```

Zadanie 2. Napisz program drukujący tablicę wartości 4 funkcji trygonometrycznych dla podanego zakresu [xp, xk]. Wartości xp i xk mogą być hardkodowane w programie.

Wskazówka:

```
xp=0
xk=2*np.pi
krok=0.1
liczba_punktow=np.abs((xk-xp)/krok)#wyznacz l. punktow
liczba_punktow_int=int(np.ceil(liczba_punktow))#zaadbaj o zgodnosc typow
print("liczba punktow wartosci x: %d" % liczba_punktow)
x=xp
for i in range(0,liczba_punktow_int):
    x=x+krok
    print("sin(%f)=%f" % (x,np.sin(x)))
```

Pętla 'for' iteruje zawsze po składnikach listy/tablicy. Tablicę wartości dla

'i' generujemy funkcją **range(poczatek, koniec, krok)**, co pozwala na uzyskanie szeregu liczb w zakresie [**poczatek**, **koniec**-1] z krokiem **krok**. (konstrukcja pętli i warunków w Python: https://docs.python.org/3/tutorial/controlflow.html)

Przykładowy rezultat:

```
liczba punktow wartosci x: 62
0 , sin: 0.099833, cos: 0.995004, tan: 0.100335, ctg: 9.966644
1 , sin: 0.198669, cos: 0.980067, tan: 0.202710, ctg: 4.933155
2 , sin: 0.295520, cos: 0.955336, tan: 0.309336, ctg: 3.232728
3 , sin: 0.389418, cos: 0.921061, tan: 0.422793, ctg: 2.365222
4 , sin: 0.479426, cos: 0.877583, tan: 0.546302, ctg: 1.830488
5 , sin: 0.564642, cos: 0.825336, tan: 0.684137, ctg: 1.461696
6 , sin: 0.644218, cos: 0.764842, tan: 0.842288, ctg: 1.187242
7 , sin: 0.717356, cos: 0.696707, tan: 1.029639, ctg: 0.971215
8 , sin: 0.783327, cos: 0.621610, tan: 1.260158, ctg: 0.793551
9 , sin: 0.841471, cos: 0.540302, tan: 1.557408, ctg: 0.642093
10 , sin: 0.891207, cos: 0.453596, tan: 1.964760, ctg: 0.508968
11 , sin: 0.932039, cos: 0.362358, tan: 2.572152, ctg: 0.388780
12 , sin: 0.963558, cos: 0.267499, tan: 3.602102, ctg: 0.277616
13 , sin: 0.985450, cos: 0.169967, tan: 5.797884, ctg: 0.172477
14 , sin: 0.997495, cos: 0.070737, tan: 14.101420, ctg: 0.070915
15 , sin: 0.999574, cos: -0.029200, tan: -34.232533, ctg: -0.029212
16 , sin: 0.991665, cos: -0.128844, tan: -7.696602, ctg: -0.129927
17 , sin: 0.973848, cos: -0.227202, tan: -4.286262, ctg: -0.233304
18 , sin: 0.946300, cos: -0.323290, tan: -2.927098, ctg: -0.341635
19 , sin: 0.909297, cos: -0.416147, tan: -2.185040, ctg: -0.457658
20 , sin: 0.863209, cos: -0.504846, tan: -1.709847, ctg: -0.584848
21 , sin: 0.808496, cos: -0.588501, tan: -1.373823, ctg: -0.727896
22 , sin: 0.745705, cos: -0.666276, tan: -1.119214, ctg: -0.893484
```

Zadanie 3. Napisz program drukujący N kolejnych liczb parzystych. Wykorzystaj pętlę 'for'.

Wskazówka: realizacja na pętli 'while'

```
x=0#indeks kolejnej liczby pierwszej
i=0#wartosc kolejnej liczby pierszej
N=10#ilosc liczb pierwszysch zadana z gory
while x<N:
    if i%2==0:
        print("%d : i=%d" % (x,i))
        x=x+1
    i=i+1
```

Zadanie 4. Napisz program obliczający i drukujący rozwiązanie równania kwadratowego postaci $a \cdot x^2 + b \cdot x + c = 0$. Wykonaj test programu dla uzyskania rozwiązań rzeczywistych i zespolonych (https://pl.wikipedia.org/wiki/R%C3%B3wnanie_kwadratowe). Porównaj działanie programu z www.wolframalpha.com.

Zadanie 5.Napisz program obliczający n! (silnia) dla z góry ustalonego n.

Zadanie 6. Uruchom poniższy kod. Wyjaśnij (opisz) jaki algorytm jest przez niego realizowany?

```
x = 3.0
xn = 0.1
blad=0.000001
x_old=0
while abs(x_old-xn)>blad:
    x_old=xn
    xn=(xn+x/xn)/2

print ("x=%lf" % xn)
```