```
###########

## f = open("C:/war&peace.txt")

arr = []

###for s in f:

t = re.split("[\s;:\-_*\".,?!()]", text) #s)

t = [a for a in t if a != '']

arr.extend(t)

print("----\n", arr)

#################

import sys

pattern = re.compile("(([\w]+[-'])*[\w']+'?)", re.U)

##????line = unicode(text, 'cp1251')

line=text

line = line.replace('--', ' -- ')

for token in line.split(' '):

   m = pattern.match(token)

   if m:

     print( m.group() )
```

Лабораторна робота №4 (ру-4)

Рішення нелінійного рівняння

Необхідно використовуючи пакет matplotlib і можливо numpy графічно дослідити рішення нелінійного рівняння і отримати все (якщо можливо) рішення. Зберегти малюнки графіків на окремий файл. Вставити цей файл у звіт роботи

Рішення уточнити шляхом половинного поділу з точністю eps=10^{-8}. Для цього скласти програму.

Варіанти

1) $\sqrt{\left(25 - x^2\right)} = arctg\ (2x)$

2) $\left(x^3 - 2x^2 - 3x + 4\right) = 2\cos(2x + 1)$

3) $arctg 2x - \frac{(x-1)^4}{(5)} + \sin^2(5x) = 0$

4) $\sin^2 x \cdot \sqrt{\left(81 - x^2\right)} = 5e^{-x^2}$

5) $\frac{\left(x^2 - 9\right)}{\left(x^2 + 4\right)} = \sqrt{x^2 + 1}e^{x\cos x}$

6) $\ln^2(x-1) = 3\cos(2x) + 1$

7) $\sin(x)\sqrt{(81-x^2)} = 5arctg(x)$

8) $\frac{10}{(1+x^2)} - 2\cos 2x + x$ $= 0$

9) $\frac{(10x-2)}{(3+x^2)} - 2\cos 2x \cdot \sqrt[4]{x}$ $=0$

10) $\frac{10}{(1-x^2)} = 2\sin(2x) + x$

11) $\sqrt{(25-x^2)} = arctg(2x)$

12) $(x^3-2x^2-3x+4) = 2\cos(2x+1)$

13) $arctg\,2x - \frac{(x-1)^4}{(5)} + \sin^2(5x) = 0$

14) $\sin^2 x \cdot \sqrt{(81 - x^2)} = 5e^{-x^2}$

15) $\frac{(x^2 - 9)}{(x^2 + 4)} = \sqrt{x^2 + 1}e^{x\cos x}$

16) $\ln^2(x - 1) = 3\cos(2x) + 1$

17) $\left|\sin x \cdot \sqrt{(81 - x^2)} - 5arctgx\right. =0$

18) $\frac{10}{(1 + x^2)} - 2\cos 2x + x =0$

19) $\frac{(10x - 2)}{(3 + x^2)} - 2\cos 2x \cdot \sqrt[4]{x} = 0$

20) $\frac{10}{(1 - x^2)} = 2\sin(2x) + x$

21) $\sqrt{(25-x^2)} = arctg(2x)$

22) $\sin^2 x \cdot \sqrt{(81-x^2)} = 5e^{-x^2}$

23) $(x^3 - 2x^2 - 3x + 4) = 2\cos(2x+1)$

24) $\cos(x)^2 \sqrt{(81-x^2)} = 5arctg(x)$

25) $4\cos(x)^2 = \sqrt{(25-x^2)}$

****

Приклад рішення

Графічно дослідити рішення нелінійного рівняння та отримати всі (якщо можливо) рішення.

Зберегти малюнки графіків на окремий файл.

Рішення уточнити шляхом половинного поділу з точністю eps=10-8. Для цього скласти програму.

$$4\cos(x)^2 = \sqrt{(25-x^2)}$$

Рішення

Для рівняння Ф1(x) = Ф2(x) значення x буде розв'язанням у точці перетину графіків функцій
y1(x)=Ф1(x) та y2(x)=Ф2(x)

Для нашого випадку Ф1(x) – це графік функцій

$$y1(x) = 4\cos(x)^2$$

Ф2(x) – це графік функцій

$$y2a(x) = +\sqrt{(25 - x^2)}$$

$$y2b(x) = -\sqrt{(25 - x^2)}$$

Т.к. підкорене вираз $\sqrt{(25 - x^2)}$ має бути більше нуля, то

$25 - x^2 \geq 0$ звідки слідує $x \in [-5, +5]$

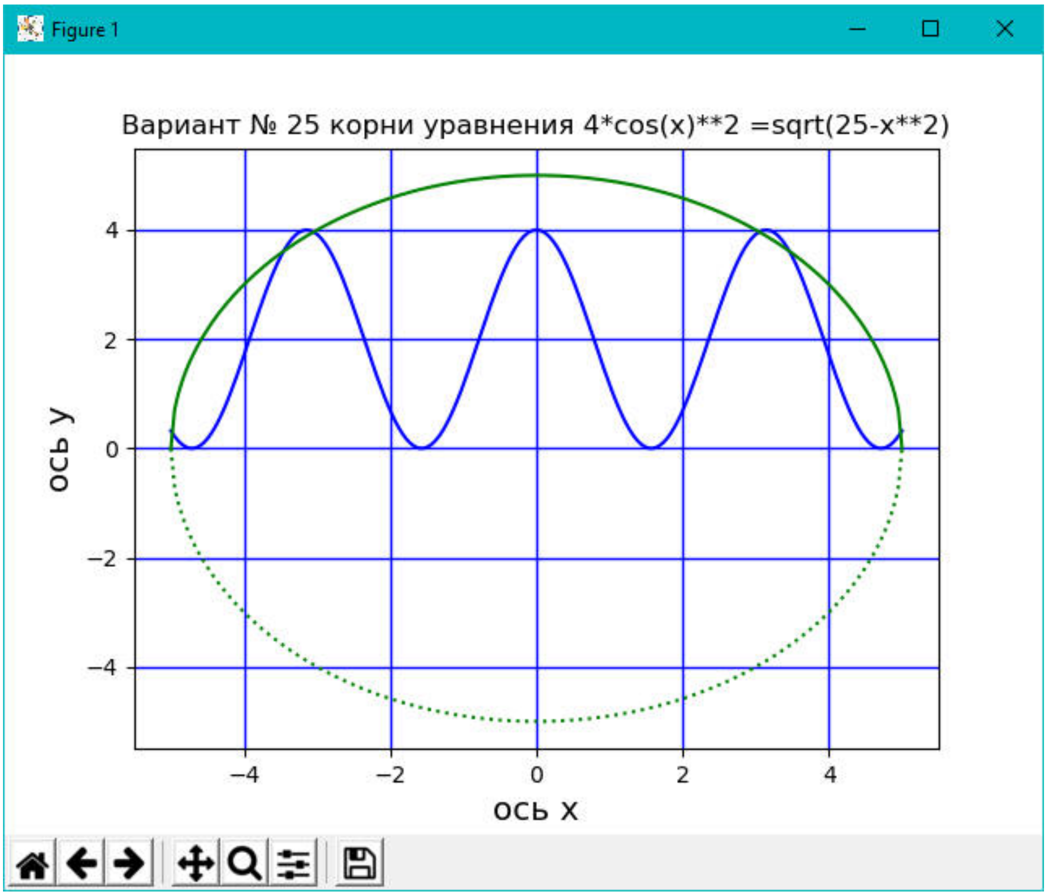
Запишемо програму

(виводити легенду не будемо)

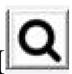
```
import matplotlib.pyplot as plt
import numpy as np
# независимая переменная
x = np.linspace(-5, +5, 200) # 200 точек
# название осей
xlab='ось x'
ylab='ось y'
# первый график
y1= 4*np.cos(x)**2
# легенда
leg1='cos(x)**2'
# второй график - a
y2a= +np.sqrt(25-x**2)
leg2a='+sqrt(25-x**2)'
# второй график - b
y2b= -np.sqrt(25-x**2)
leg2b='-sqrt(25-x**2)'
# Включаем сетку по оси X и оси Y.
# Задаем цвет толщину сетки
plt.grid(color = 'b',linewidth = 1)
# Задаем подписи к осям X и оси Y и размер шрифта
```

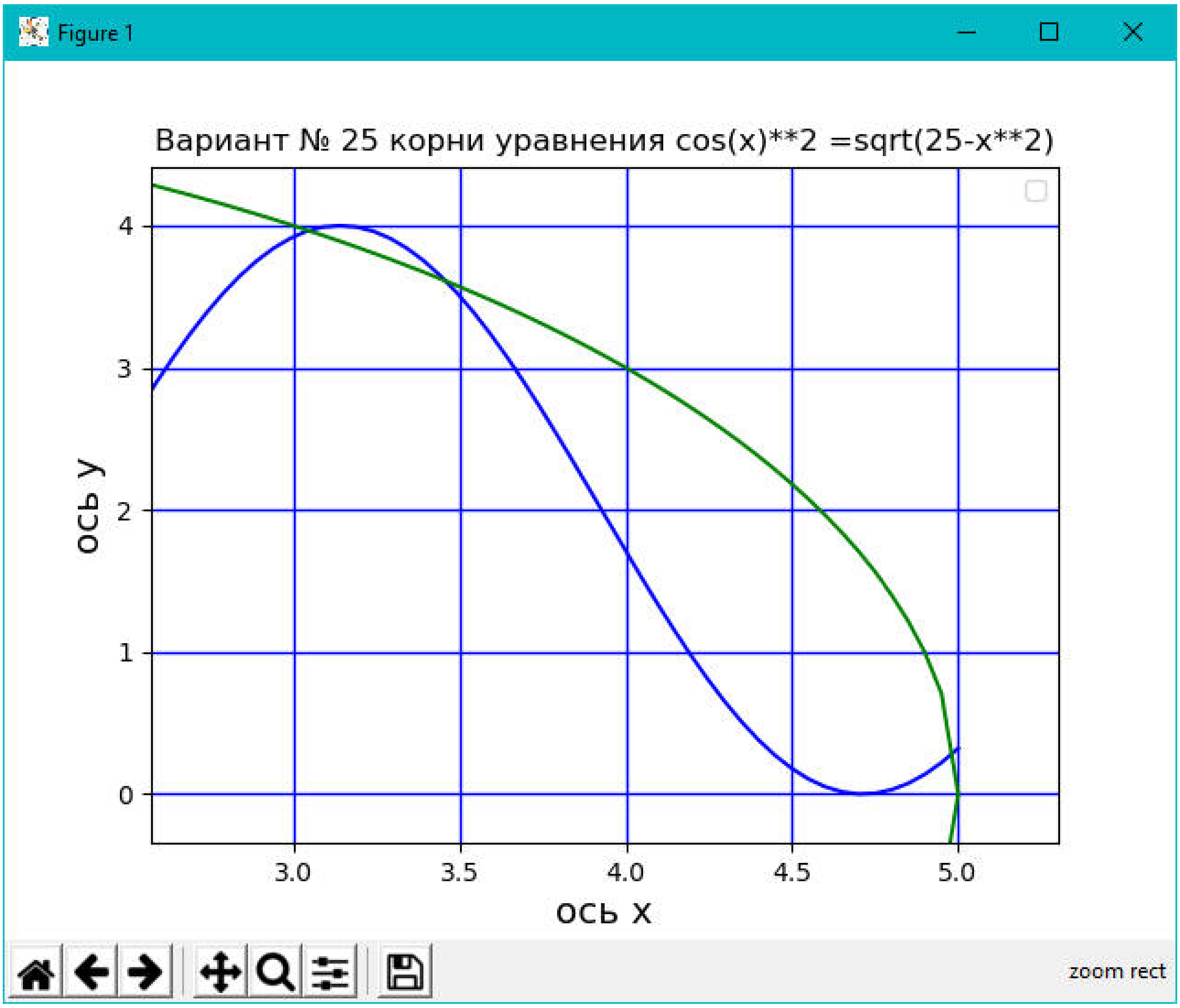
```
plt.xlabel(xlab, fontsize = 'x-large')
plt.ylabel(ylab, fontsize = 'x-large')
# Задаем заголовок диаграммы
plt.title('корни уравнения cos(x)**2 =sqrt(25-x**2)')
# строим графики
plt.plot(x,y1,'b-') # ,label=leg1)
plt.plot(x,y2a,'g-') #,label=leg2a)
plt.plot(x,y2b,'g-') #,label=leg2b)
# задаем вывод легенды и ее расположение
# plt.legend(loc='best')
# Включаем сетку
plt.grid(True)
# Сохраняем построенную диаграмму в файл
# Задаем имя файла и его тип
plt.savefig('var25.png', format = 'png')
# визуализируем графики
plt.show()
```

Результат роботи програми



Бачимо ШІСТЬ точок перетину графіків

використовуючи ZOOM



знаходимо

$x1 \in [3.0, \quad 3.25]$

$x2 \in [3.25, \quad 3.5]$

$x3 \in [4.75, \quad 5.0]$

Значення x4, x5 і x6 з симетрії дорівнюють -x1, -x2 і -x3

Складемо програму уточнення коренів методом половинного поділу

```
import math
def f(x):
   return 4*math.cos(x)**2 - math.sqrt(25-x**2)

############
def pd(a, b, eps=1e-8):
   if f(a)*f(b) >0:
      print("error f(a)*f(b) > 0")
```

```
        return -1
    if a >= b:
        print("error a < b")
        return -1
    while True:
        x=a+(b-a)/2
        fx=f(x)
        #print(a,b,x,f(a), fx, f(b)); input()
        if abs(fx) < eps:
            return x
        if f(a)*fx < 0:
            b=x
        else:
            a=x
        pass

##############
x1=pd(3.0, 3.25)
x2=pd(3.25, 3.5)
x3=pd(4.75, 5.0)
print('x1=', x1)
print('x2=', x2)
print('x3=', x3)
x4=-x1; x5=-x2; x6=-x3;
print('x4=', x4)
print('x5=', x5)
print('x6=', x6)
print(10*'-')
for x in [x1,x2,x3,x4,x5,x6]:
print('f(',x,')=',f(x) )



================================ RESTART:
D:/Test/Python/lab3_pd.py ================================
x1= 3.04699943959713
x2= 3.4589415714144707
x3= 4.990856625139713
x4= -3.04699943959713
x5= -3.4589415714144707
x6= -4.990856625139713
----------
f( 3.04699943959713 )= -4.768277772626561e-09
f( 3.4589415714144707 )= -5.146310044779057e-10
f( 4.990856625139713 )= 8.167858944752027e-09
f( -3.04699943959713 )= -4.768277772626561e-09
f( -3.4589415714144707 )= -5.146310044779057e-10
```