

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

ОТЧЕТ

по лабораторной работе №2

дисциплина: Вычислительные методы

Студент: Яссин Мохамад Аламин
Студенческий билет: 1032205004
Группа: НКНБД-01-20

МОСКВА

2022г.

Содержание

Справка – 3 стр.

Код на Python – 4 стр.

Численные расчеты – 6 стр.

Справка

В лабораторной номер два разбирается задача интерполяции, интерполяционный полином Ньютона и погрешность интерполяции.

Полином ньютона

$$Q_i(x) = P_i(x) - P_{i-1}(x)$$

Погрешность интерполяции

$$\delta(\bar{x}_j) = \left| P_N(\bar{x}_j) - f(\bar{x}_j) \right|$$

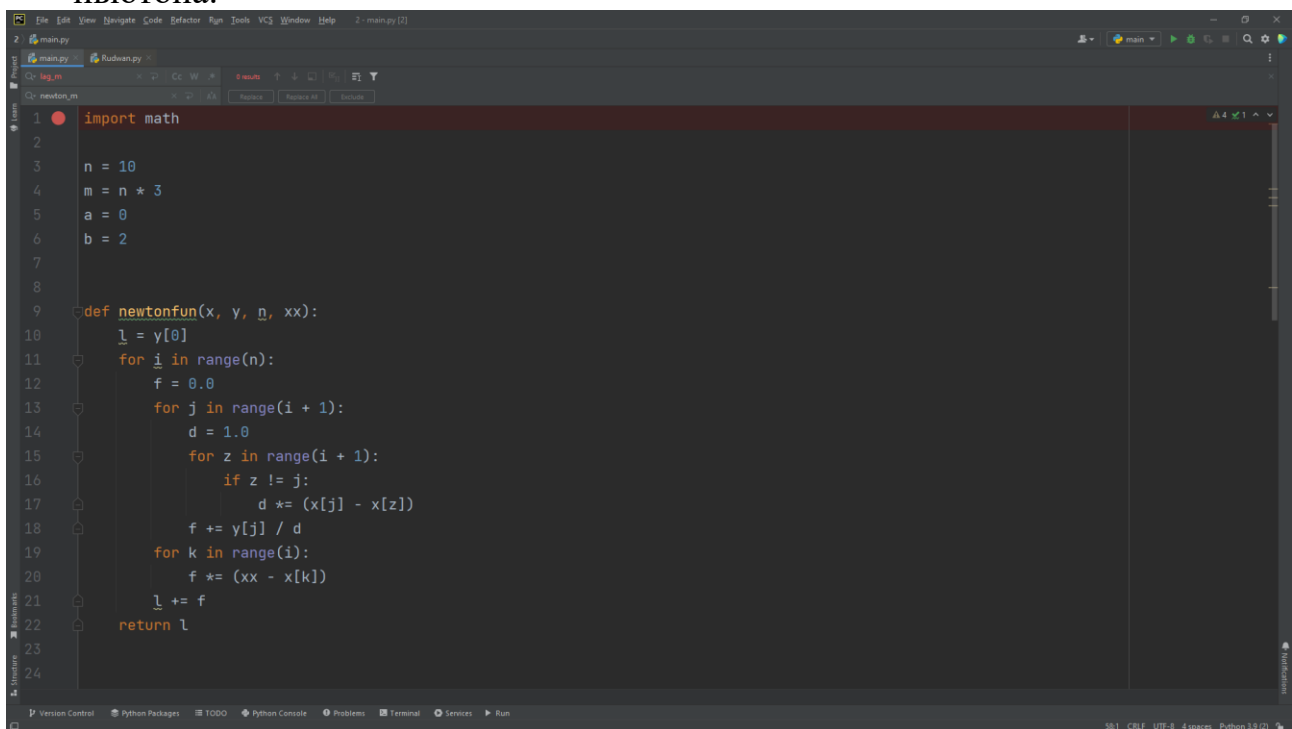
Ход работы

Мой вариант:

18.	$\sqrt{x} - x$	$x \in [0, 2]$
-----	----------------	----------------

Код на питоне:

- 1- первый раздел кода состоит из 2 частей, переменных частей и функции НЬЮТОНА.



```
1 import math
2
3 n = 10
4 m = n * 3
5 a = 0
6 b = 2
7
8
9 def newtonfun(x, y, n, xx):
10     l = y[0]
11     for i in range(n):
12         f = 0.0
13         for j in range(i + 1):
14             d = 1.0
15             for z in range(i + 1):
16                 if z != j:
17                     d *= (x[j] - x[z])
18             f += y[j] / d
19         for k in range(i):
20             f *= (xx - x[k])
21         l += f
22     return l
23
24
```

- 2- В то время как во втором разделе мы заполняем массивы x и y элементами на основе «n» в первой и второй задаче, а в четвертом на основе переменной «m», в то же время мы заполняем массив «newton» с ответами от функции newtonfun.

```
24
25 # first & second task
26 x_values = []
27 for i in range(n + 1):
28     x_values.append(round((0.2 * i), 1))
29
30 y_values = []
31 for i in range(n + 1):
32     y_values.append(round(math.sqrt(x_values[i]) - x_values[i], 5))
33
34 # third task
35 newton = []
36 for i in range(n + 1):
37     newton.append(newtonfun(x_values, y_values, n + 1, x_values[i]))
38
39 # fourth task
40 xm_values = []
41 for i in range(m + 1):
42     xm_values.append(round(((2 / m) * i), 5))
43
```

3- Здесь, в последнем разделе, мы видим применение последних двух задач, вычисление диапазона задержки и дельты в пятой и печать в правильной форме, а в шестой мы проверяем, какие поля значений «n» будут подходит для наших условий.

```
43
44 # fifth task
45 ym_values = []
46 newton_m = []
47 delta = []
48 for i in range(m + 1):
49     ym_values.append(round(math.sqrt(xm_values[i]) - xm_values[i], 5))
50     newton_m.append(newtonfun(x_values, y_values, n, xm_values[i]))
51     delta.append(abs(newton_m[i] - ym_values[i]))
52
53     print(format(xm_values[i], '.5f'), format(ym_values[i], '.5f'), format(newton_m[i], '.5f'), format(delta[i], '.5f'))
54
55 print("\n\n\n\n")
56
57
58 # sixth task
59 for i in range(n + 1):
60     if delta[i] <= 0.1:
61         print(x_values[i])
62         print(format(xm_values[i], '.5f'), format(ym_values[i], '.5f'), format(newton_m[i], '.5f'),
63               format(delta[i], '.5f'))
64
```

Вывод.

1- Здесь мы можем увидеть результаты для «m»

```
"K:\test_py\Computing modules\2\venv
0.00000 0.00000 0.00000 0.00000
0.06667 0.19154 0.15090 0.04064
0.13333 0.23181 0.22069 0.01112
0.20000 0.24721 0.24721 0.00000
0.26667 0.24973 0.25164 0.00191
0.33333 0.24402 0.24501 0.00099
0.40000 0.23246 0.23246 0.00000
0.46667 0.21646 0.21612 0.00034
0.53333 0.19696 0.19674 0.00022
0.60000 0.17460 0.17460 0.00000
0.66667 0.14983 0.14996 0.00013
0.73333 0.12302 0.12312 0.00010
0.80000 0.09443 0.09443 0.00000
0.86667 0.06428 0.06421 0.00007
0.93333 0.03276 0.03269 0.00007
1.00000 0.00000 -0.00000 0.00000
1.06667 -0.03387 -0.03380 0.00007
1.13333 -0.06875 -0.06867 0.00008
1.20000 -0.10455 -0.10455 0.00000
1.26667 -0.14121 -0.14130 0.00009
1.33333 -0.17863 -0.17875 0.00012
1.40000 -0.21678 -0.21678 0.00000
1.46667 -0.25561 -0.25537 0.00024
1.53333 -0.29505 -0.29469 0.00036
1.60000 -0.33509 -0.33509 0.00000
1.66667 -0.37547 -0.37674 0.00107
1.73333 -0.41677 -0.41887 0.00210
1.80000 -0.45836 -0.45836 0.00000
1.86667 -0.50041 -0.48712 0.01329
1.93333 -0.54289 -0.48814 0.05475
2.00000 -0.58579 -0.42955 0.15624
```

2- и здесь мы видим, что условие будет выполнено, где $\Delta \leq 0.1$

```
0.00000 0.00000 0.00000 0.00000
0.06667 0.19154 0.15090 0.04064
0.13333 0.23181 0.22069 0.01112
0.20000 0.24721 0.24721 0.00000
0.26667 0.24973 0.25164 0.00191
0.33333 0.24402 0.24501 0.00099
0.40000 0.23246 0.23246 0.00000
0.46667 0.21646 0.21612 0.00034
0.53333 0.19696 0.19674 0.00022
0.60000 0.17460 0.17460 0.00000
0.66667 0.14983 0.14996 0.00013

Process finished with exit code 0
```