РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА №2

СТУДЕНТ: Копорушкин Данил
ГРУППА: МТ-301

ВАРИАНТ:12

Тема: численное решение нелинейных уравнений.

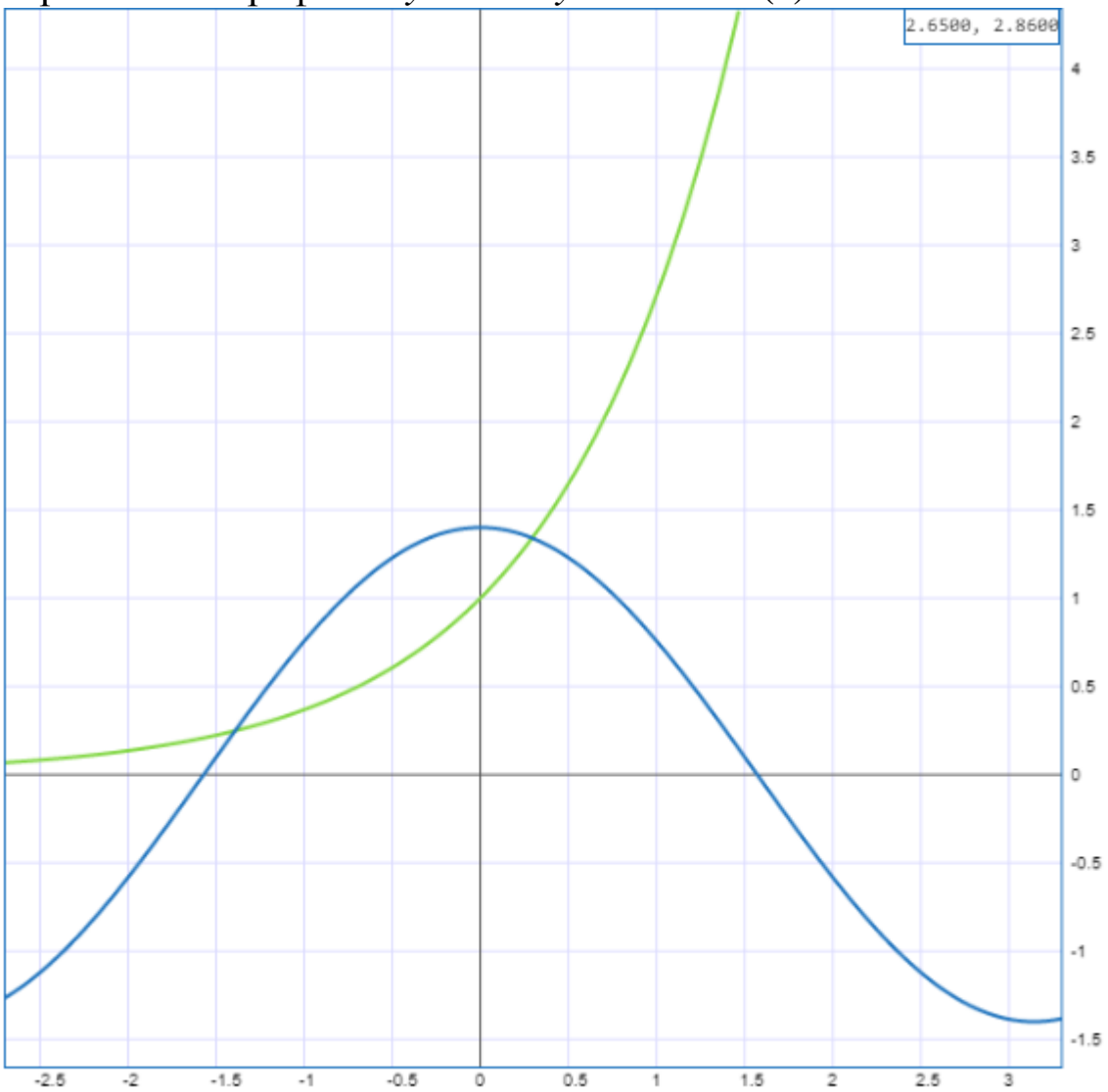
Постановка задачи.

Дана функция $f(x) = 1.4 * \cos(x) - \exp(x)$

требуется вычислить один корень уравнения с заданной точность $\varepsilon = 0.5*10^{-4}$,применяя: метод половинного деления, мод. метод Ньютона, метод простой итерации.

Локализация корня

Необходимо найти отрезок [a,b], на котором существует единственный корень данного уравнения. Для этого преобразуем уравнение к равносильному виду $e^x = 1.4*\cos(x)$; и найдем точки пересечения графиков $y = e^x$ *и* $y = 1.4*\cos(x)$



Очевидно, корень уравнения $\wp \in [0; 1]$.

Метод половинного деления

1. Отрезок, на котором существует единственный корень найден.

2.Найдем середину текущего отрезка [a,b], c=$\frac{a+b}{2}$т.е. c=$\frac{2.5+3}{2}$

3.Если f(a)*f(c)<0, то положить b=c, а если f(c)*f(b)<0, то a=c. В результате находится текущий отрезок локализации корня [a, b], в 2 раза меньше предыдущего

4. Если |b-a|< ε=0.5*10^{-4}, то процесс завершить и положить приближенным значением корня уравнения середину последнего найденного отрезка, в противном случае переходим к пункту 2.

5)количество итераций считается по формуле: $\varepsilon = \frac{b-a}{2^n} \Rightarrow n = \frac{\ln \frac{b-a}{\varepsilon}}{\ln 2}$ ~17.6

Приложение

```
import matplotlib.pyplot as plt
import math
import numpy as np
from scipy.misc import *

a = 0
b = 1
step = 0
eps = 0.5*10**-4
c = (a+b)/2
i = 3

def f(x):
    return 1.4 * math.cos(x) - math.exp(x)

def derivF(x):
    return derivative(f, x)

def deriv2F(x):
    return derivative(derivF,x)

def mpd(a, b, c, eps, step):
    while (abs(a-b) >= eps and f(c)!=0):
        step += 1
        if f(a) * f(c) < 0:
            b = c
        elif f(b) * f(c) < 0:
            a = c
        c = (a + b) / 2
    print('Метод Дифотомии: за '+str(step)+' шагов', 'был получен корень ' + f'{c:.{i}f}' + '

mpd(a, b, c, eps, step)
```

```
C:\Users\darl0\Anaconda3\python.exe "C:/Users/darl0/Desktop/Лабы/Лаба 2.py"
Метод Дифотомии: за 15 шагов был получен корень 0.293 с точностью до 3 знака
Модиф.метод Ньютона: за 12 шагов был получен корень 0.293 с точностью до 3 з
МПИ: за 8 шагов был получен корень 0.293 с точностью до 3 знака после запято

Process finished with exit code 0
```

Модифицированный метод Ньютона

Выберем начальное приближение к корню x_0из условия:

$f(x_0) * f''(x_0) < 0$

$f'(x) = -e^x - 1.4*\cos(x) < 0$

$f'(a) = -2.37$

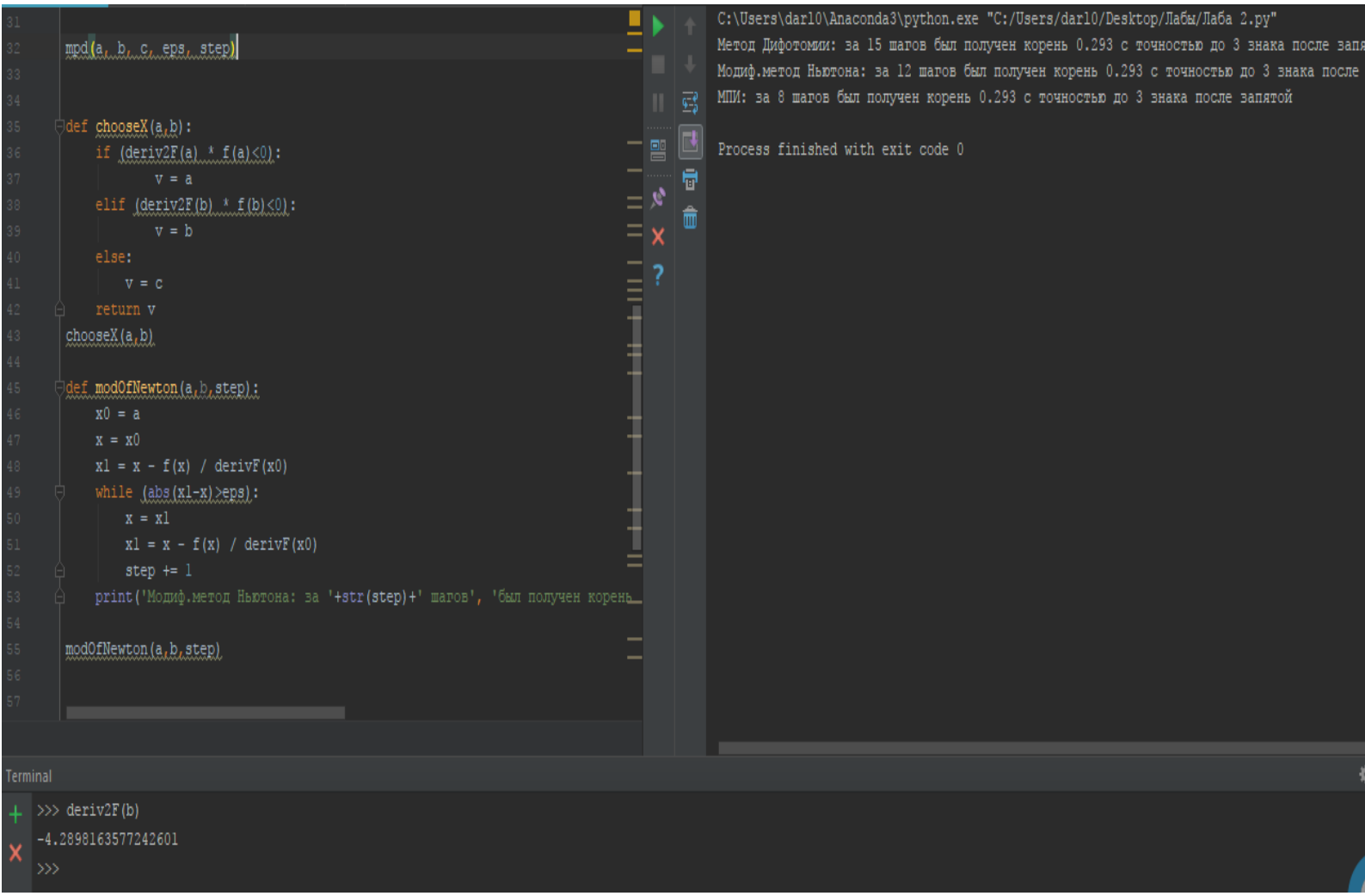
$f'(b) = -4.29$

$f(a) = 0.4 > 0; f(b) = -1.96 < 0$ Отсюда следует, что x_0=a

Каждое следующее приближение можно вычислить по формуле:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)} = x_n - \frac{1.4 * \cos(x) - \exp(x)}{-e^x - 1.4*\sin(x)}$$

Приложение



Метод простой итерации

Заменим исходное уравнение f(x)=0 на эквивалентное x=c(x) на отрезке [0;1].

Пусть, x=x+ a*f(x)=x+ a(1.4 * cos(x) - exp(x))

Найдем a = $\frac{2}{M+м}$, где M=max(f'(x)) m = min(f'(x))

Получим a~-0.408, m~-3,9, M~-1

Чтобы МПИ сходился нужно выполнение условия:

$|c(x)'| = |(1 - a*f'(x))| \le q < 1$ (1)

При x=0,293 и a=0,4 (2) это условие выполняется => Метод простой итерации сходится.

Так же аналитически подсчитаем количество итераций:

q≤0,24 из (1) при (2)

$0{,}5*10^{-4} < 1*0{,}24^{n}$ => $n > \frac{\ln 0{,}5*10^{-4}}{\ln 0{,}24} \sim 7$

Приложение

```
    while (abs(x1-x)>eps):
        x = x1
        x1 = x - f(x) / derivF(x0)
        step += 1
    print('Модиф.метод Ньютона: за '+str(step)+' шагов', 'был получен корень

modOfNewton(a,b,step)


def mpi(a,b,step):
    x0 = chooseX(a,b)
    x = x0
    p = 2/(-1-3.9)
    x1 = x - p*f(x)
    while (abs(x1-x)>eps):
        x = x1
        x1 = x - p * f(x)
        step += 1
    print('МПИ: за '+str(step)+' шагов', 'был получен корень ' + f'{x:.{i}f}')

mpi(a,b,step)
```

```
C:\Users\darl0\Anaconda3\python.exe "C:/Users/darl0/Desktop/Лабы/Лаба 2.py"
Метод Дифотомии: за 15 шагов был получен корень 0.293 с точностью до 3 знака после запят
Модиф.метод Ньютона: за 12 шагов был получен корень 0.293 с точностью до 3 знака после з
МПИ: за 8 шагов был получен корень 0.293 с точностью до 3 знака после запятой

Process finished with exit code 0
```

```
Terminal
>>> deriv2F(b)
-4.2898163577242601
>>>
```

Сравнение скоростей сходимости: корень уравнения $f(x) = 1.4 * \cos(x) - \exp(x) = 0.2929 \sim 0{,}293$ на отрезке [0; 1] вычисляется за: 15 шагов методом половинного деления, за 12 шагов Модифицированным методом Ньютона и за 8 шагов методом простой итерации.

Вывод:

Метод простой итерации сходятся быстрее других методов.

Метод	***Корень***	***Число итераций(аналит.)***	***Число итераций на пк***
Дихотомии	0,293	≤17.6	15
Мод метод Ньютона	0,293	----------	12
МПИ	0,293	>7	8