МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРОНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине

«Методы оптимизации»

Вариант № 13

Выполнил работу:

Студент группы P3218

Рамеев Тимур Ильгизович

Преподаватель:

Кудашов Вячеслав Николаевич

Санкт-Петербург 2024

Содержание

Задание

Найти минимум функции на заданном интервале. Решить задачу методом половинного деления, методом золотого сечения и методом Ньютона; написать программу на языке Python. Параметры ε и δ принять равными 10^{-10}. Программа должна выполнить 25 итераций или остановиться при достижении критерия остановки итерационного процесса. Осуществлять вывод вычисленных значений на каждой итерации с точностью до 5 знаков после запятой.

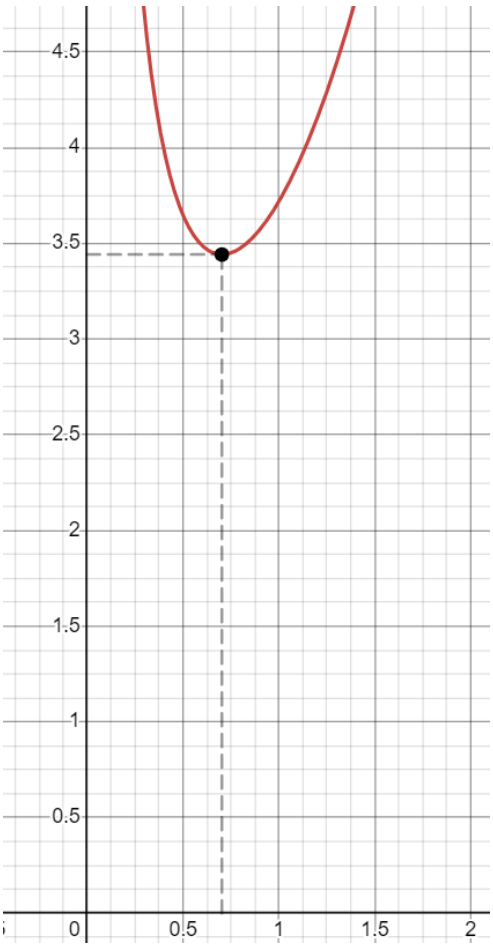
Исходная функция

$$f(x) = \frac{1}{x} + e^x$$

Интервал

$$[a, b] = [0.5, 1.5]$$

График



Исходный код реализованных методов

Метод половинного деления

```
def dividing_segment_in_half(rounding):
    print("Метод половинного деления:")
    current_a = a
    current_b = b
    counter = 0
    while current_b - current_a > 2 * eps and counter < 25:
        counter += 1
        x1 = (current_a + current_b - eps) / 2
        x2 = (current_a + current_b + eps) / 2
        y1 = function_value(x1)
        y2 = function_value(x2)
        if y1 > y2:
            current_a = x1
        else:
            current_b = x2
            curr_val = (current_a + current_b) / 2
        print(f'{counter}: x={round(curr_val, rounding)}\tf(x)={round(function_value(curr_val),
rounding)}')
    ret_val = (current_a + current_b) / 2
    print_result(ret_val, function_value(ret_val))
```

Метод Ньютона

```
def newton_method(rounding):
    print("Метод Ньютона:")
    current_approximation = (a + b) / 2
    counter = 0
    while abs(first_derivative_value(current_approximation)) > eps and counter < 25:
        counter += 1
        current_approximation = current_approximation -
first_derivative_value(current_approximation) / second_derivative_value(current_approximation)
        print(f'{counter}: x={round(current_approximation,
rounding)}\tf(x)={round(function_value(current_approximation), rounding)}')
    print_result(current_approximation, function_value(current_approximation))
```

Метод золотого сечения

```
def golden_ratio(rounding):
    print("Метод золотого сечения:")
    golden = (1 + 5 ** 0.5) / 2
    current_a = a
    current_b = b
    y1 = function_value(a + (b - a) * (1 - 1 / golden))
    y2 = function_value(a + (b - a) * (1 / golden))
    counter = 0
    while  current_b - current_a > 2 * eps and counter < 25:
        counter += 1
        if y1 < y2:
            current_b = current_a + (current_b - current_a) * (1 / golden)
            y2 = y1
            y1 = function_value(current_a + (current_b - current_a) * (1 - 1 / golden))
        else:
            current_a = current_a + (current_b - current_a) * (1 - 1 / golden)
            y1 = y2
            y2 = function_value(current_a + (current_b - current_a) * (1 / golden))
        curr_val = (current_a + current_b) / 2
        print(f'{counter}: x={round(curr_val, rounding)}\tf(x)={round(function_value(curr_val),
rounding)}')
    ret_val = (current_a + current_b) / 2
    print_result(ret_val, function_value(ret_val))
```

Вывод программы

	Метод половинного деления		Метод золотого сечения		Метод Ньютона	
№ итерации	a	b	a	b	x	f'(x)
1	0.5	1.5	0.5	1.5	0.538	-1.74E+00
2	0.5	1.118	0.5	1.0	0.752	3.54E-01
3	0.5	0.882	0.5	0.75	0.673	-2.48E-01
4	0.646	0.882	0.625	0.75	0.72	1.28E-01
5	0.646	0.792	0.687	0.75	0.693	-7.99E-02
6	0.646	0.736	0.687	0.719	0.709	4.50E-02
7	0.68	0.736	0.703	0.719	0.7	-2.70E-02
8	0.68	0.715	0.703	0.711	0.705	1.56E-02
9	0.693	0.715	0.703	0.707	0.702	-9.26E-03
10	0.693	0.707	0.703	0.705	0.704	5.41E-03
11	0.699	0.707	0.703	0.704	0.703	-3.18E-03
12	0.702	0.707	0.703	0.704	0.704	1.87E-03
13	0.702	0.705	0.703	0.704	0.703	-1.10E-03
14	0.703	0.705	0.703	0.703	0.704	6.43E-04
15	0.703	0.704	0.703	0.703	0.703	-3.78E-04
16	0.703	0.704	0.703	0.703	0.703	2.22E-04
17	0.703	0.704	0.703	0.703	0.703	-1.30E-04
18	0.703	0.704	0.703	0.703	0.703	7.64E-05
19	0.703	0.704	0.703	0.703	0.703	-4.49E-05
20	0.703	0.704	0.703	0.703	0.703	2.63E-05
21	0.703	0.704	0.703	0.703	0.703	-1.55E-05
22	0.703	0.703	0.703	0.703	0.703	9.08E-06
23	0.703	0.703	0.703	0.703	0.703	-5.33E-06
24	0.703	0.703	0.703	0.703	0.703	3.13E-06
25	0.703	0.703	0.703	0.703	0.703	-1.84E-06
$x_к$	0.7034664005337656		0.7034667702091327		0.7034671860632785	
$F(x_к)$	3.4422772944990294		3.4422772944990294		3.4422772944951916	