Федеральное государственное автономное образовательное учреждение высшего образования Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Вычислительная математика

Лабораторная работа №2

Вариант № 1

Выполнил: студент группы P3208,
Васильев Н. А.

Преподаватель: Машина Е.А.

Санкт-Петербург 2025

Текст задания

Численное решение нелинейных уравнений и систем.

Цель работы

Изучить численные методы решения нелинейных уравнений и их систем, найти корни заданного нелинейного уравнения/системы нелинейных уравнений, выполнить программную реализацию методов

Описание метода, расчётные формулы

Метод половинного деления:

$$x_i = \frac{a_i + b_i}{2}$$

Метод секущих:

$$x_{i+1} = x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} f(x_i)$$

Метод хорд:

$$x_i = a_i - \frac{b_i - a_i}{f(b_i) - f(a_i)} f(a_i)$$

Метод Ньютона:

$$x_i = x_{i-1} - \frac{f(x_{i-1})}{f'(x_{i-1})}$$

Метод секущих:

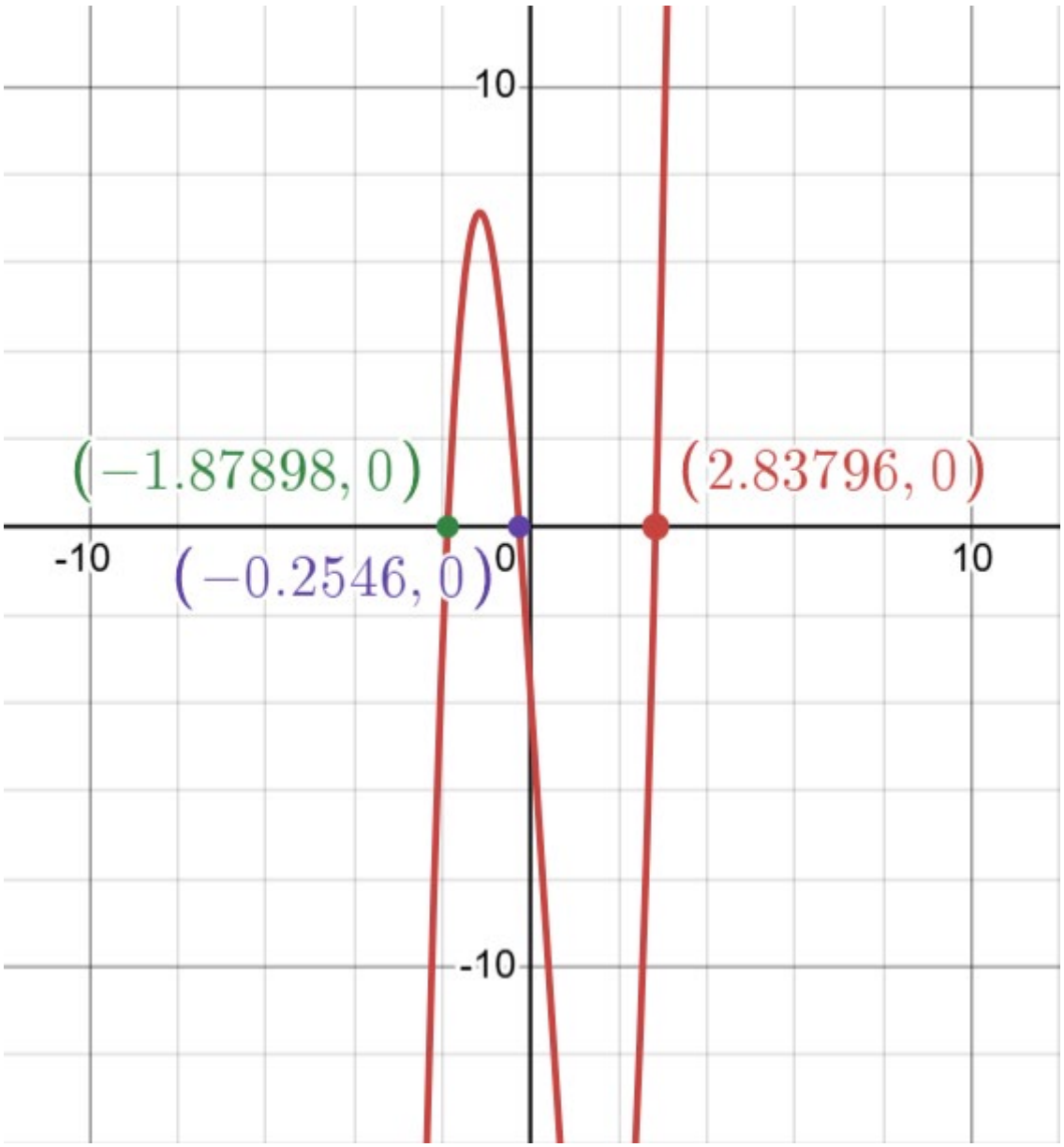
$$x_{i+1} = x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} f(x_i)$$

Вычислительная реализация задачи:

1 часть. Решение нелинейного уравнения

1. Отделить корни заданного нелинейного уравнения графически.

$$2{,}74x^3 - 1{,}93x^2 - 15{,}28x - 3{,}72$$



2. Определить интервалы изоляции корней.

Определение интервалов изоляции корней нелинейного уравнения — это процесс нахождения отрезков, на которых содержится ровно один корень уравнения.

Найдем точки пересечения графика с осью x: $x_1 \approx -1{,}88$; $x_2 \approx -0{,}25$; $x_3 \approx 2{,}84$.

Найдем интервалы изоляции корней:

x	$f(x)$
$-2{,}50$	$-20{,}40$
$-2{,}25$	$-10{,}32$
$\mathbf{-2{,}00}$	$\mathbf{-2{,}80}$
$\mathbf{-1{,}75}$	$\mathbf{2{,}43}$
$-1{,}50$	$5{,}61$
$-1{,}25$	$7{,}01$
$-1{,}00$	$6{,}89$
$-0{,}75$	$5{,}50$
$\mathbf{-0{,}50}$	$\mathbf{3{,}10}$
$\mathbf{-0{,}25}$	$\mathbf{-0{,}06}$

0,00	$-3,72$
0,25	$-7,62$
0,50	$-11,50$
0,75	$-15,11$
1,00	$-18,19$
1,25	$-20,48$
1,50	$-21,73$
1,75	$-21,69$
2,00	$-20,08$
2,25	$-16,66$
2,50	$-11,17$
$\mathbf{2,75}$	$\mathbf{-3,35}$
$\mathbf{3,00}$	$\mathbf{7,05}$
3,25	20,29
3,50	36,64

Интервалы изоляции корней: $(-2,00;\ -1,75)$, $(-0,50;\ -0,25)$ и $(2,75;\ 3,00)$.

3. Уточнить корни нелинейного уравнения с точностью $\varepsilon = 10^{-2}$.

$x_1 \approx -1,01$

$x_2 \approx 1,99$

$x_3 \approx 2,841$

4. Используемые методы для уточнения каждого из 3-х корней многочлена.

Метод половинного деления – крайний правый корень.

$$x_i = \frac{a_i + b_i}{2}$$

№ шага	a	b	x	$f(a)$	$f(b)$	$f(x)$	$\lvert a-b\rvert$
1	2,750	3,000	2,875	-3,350	7,050	1,510	0,250
2	2,750	2,875	2,813	-3,350	1,510	-0,985	0,125
3	2,813	2,875	2,844	-0.985	1,510	0,242	0,062
4	2,813	2,844	2,829	-0.985	0,242	-0,376	0,031
5	2,829	2,844	2,837	-0,376	0,242	-0,058	0,016
6	2,837	2,844	**2,841**	-0.058	0,242	0,101	0,008

Метод секущих – крайний левый корень.

$$x_{i+1} = x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} f(x_i)$$

№ шага	x_{k-1}	x_k	x_{k+1}	$f(x_{k+1})$	$\lvert x_{k+1} - x_k\rvert$
1	2,750	3,000	2,831	-0,295	0,169
2	3,000	2,831	**2,837**	-0,024	0,007

Метод простой итерации – центральный корень.

Найдем производную функции $f(x) = 2,74x^3 - 1,93x^2 - 15,28x - 3,72$

$$f'(x) = 8{,}22x^2 - 3{,}86x - 15{,}28$$

На границах интервала производная равна:

$$f'(-0{,}50) = -11{,}295$$

$$f'(-0{,}25) = -13{,}801$$

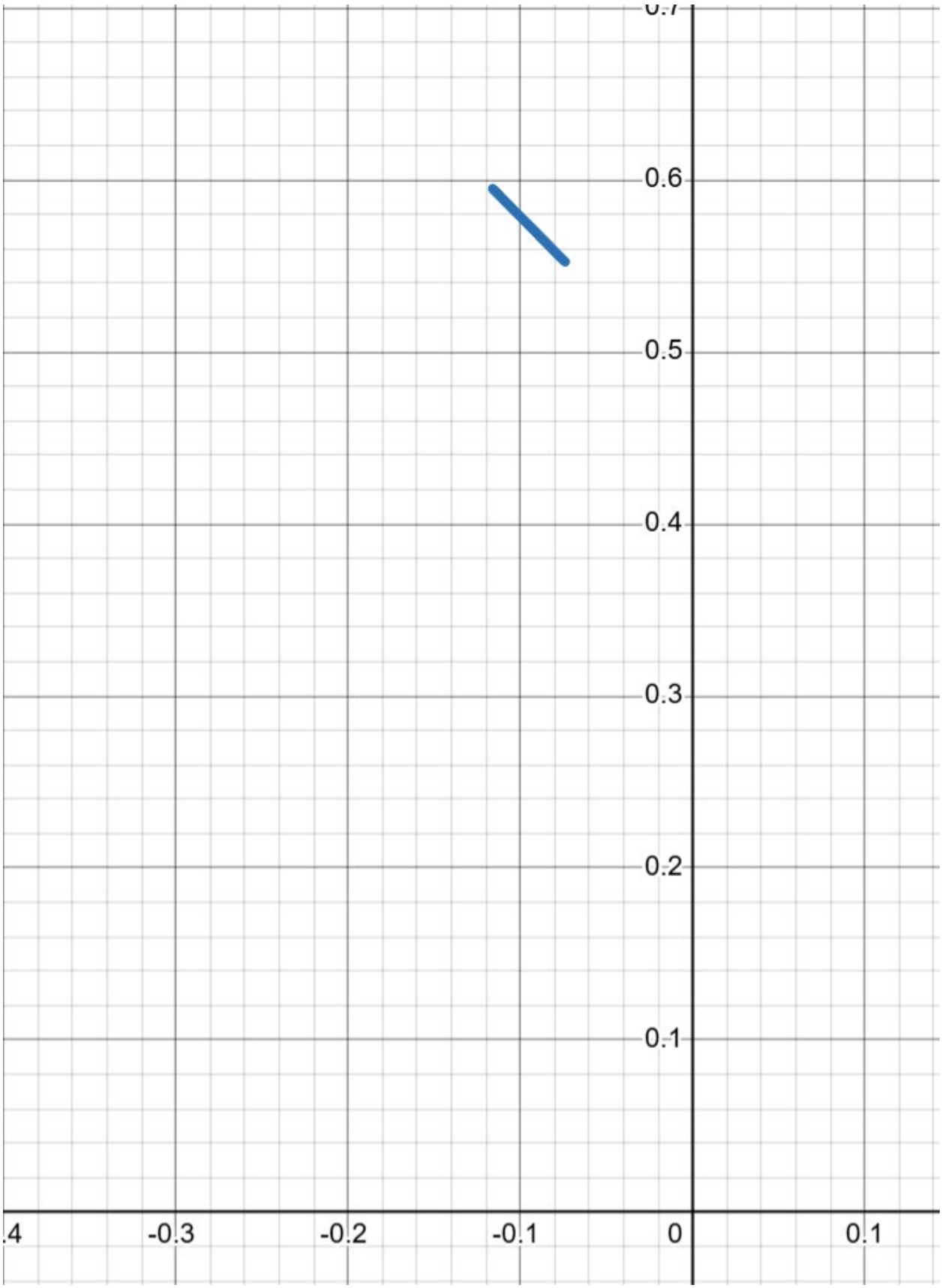
$$\lambda = \frac{1}{\max\left(|f'(-0{,}50)|,\ |f'(-0{,}25)|\right)} = \frac{1}{13{,}801}$$

$$\varphi(x) = x + \lambda f(x) = x + \frac{2{,}74x^3 - 1{,}93x^2 - 15{,}28x - 3{,}72}{13{,}801}$$

$$\varphi'(x) = 1 + \lambda f'(x) = 1 + \frac{8{,}22x^2 - 3{,}86x - 15{,}28}{13{,}801}$$

$$\varphi'(-0{,}50) = -0{,}116$$

$$\varphi'(-0{,}25) = -0{,}074$$



$$x_{i+1} = \varphi(x_i)$$

№ шага	x_k	x_{k+1}	$f(x_{k+1})$	$\lvert x_{k+1} - x_k \rvert$
1	-0,500	-0,276	0,293	0,224
2	-0,276	-0,256	0,019	0,020

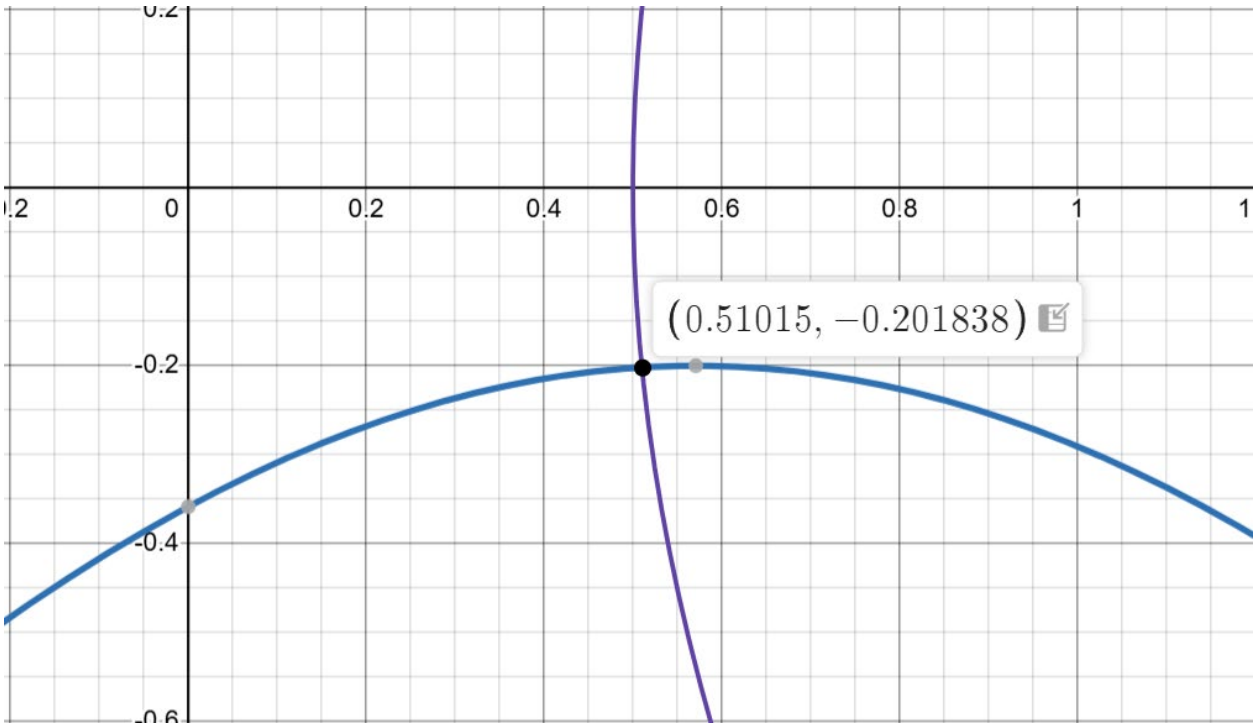
3	-0,256	**-0,255**	0,0001	0,001

2 часть. Решение системы нелинейных уравнений

$$\begin{cases} \sin(x+1) - y = 1{,}2 \\ 2x + \cos y = 2 \end{cases}$$

В привычном виде получаем:

$$\begin{cases} y = 1{,}2 - \sin(x+1) \\ x = \dfrac{\cos y - 2}{2} \end{cases}$$



Области изоляции корней:

$$G \begin{bmatrix} 0 < x < 1 \\ -1 < y < 0 \end{bmatrix}$$

Проверим выполнение условия сходимости:

$$\max_{G} \max_{i} \sum_{j=1}^{n} \left| \frac{\partial \varphi_i(x)}{\partial x_j} \right| \le q < 1$$

$$\left| \frac{\partial \varphi_1}{\partial x} \right| = 0, \left| \frac{\partial \varphi_2}{\partial x} \right| = |\cos(x+1)|$$

$$\left| \frac{\partial \varphi_1}{\partial y} \right| = \left| \frac{1}{2} \cos y \right|, \left| \frac{\partial \varphi_2}{\partial y} \right| = 0$$

$$\left| \frac{\partial \varphi_1}{\partial x} \right| + \left| \frac{\partial \varphi_2}{\partial x} \right| = |\cos(x+1)| \le 0{,}540 < 1$$

$$\left| \frac{\partial \varphi_1}{\partial y} \right| + \left| \frac{\partial \varphi_2}{\partial y} \right| = \left| \frac{1}{2} \cos y \right| \le 0{,}421 < 1$$

Следовательно, условие сходимости на области G выполнено, и метод простых итераций будет сходиться.

Возьмем начальное приближение в центре G: $x = 0{,}500; y = -0{,}500$

$$\begin{cases} x^{(k+1)} = \varphi_1(x, y) \\ y^{(k+1)} = \varphi_2(x, y) \end{cases}$$

№ итерации	x_k	x_{k+1}	$\lvert x_{k+1} - x_k \rvert$	y_k	y_{k+1}	$\lvert y_{k+1} - y_k \rvert$
1	0,500	0,561	0,061	-0,500	-0,202	0,297
2	0,561	0,510	0,051	-0,202	-0,200	0,002
3	0,510	**0,510**	0,0001	-0,200	**-0,202**	0,002

Листинг программы

```
def method_of_chords(f, a, b, epsilon, max_iter=1000):
    if not verificator.check_root(f, a, b):
        return None

    x0, x1 = a, b
    for _ in range(max_iter):
        f0, f1 = f(x0), f(x1)
        if abs(f1 - f0) < 1e-12:
            break
        x2 = x1 - f1 * (x1 - x0) / (f1 - f0)
        if abs(x2 - x1) < epsilon:
            plots.plot_function(f, a, b)
            return x2, f(x2), _ + 1
        x0, x1 = x1, x2
    plots.plot_function(f, a, b)
    return max_iter


def newtons_method(f, df, x0, epsilon, max_iter=1000):
    for _ in range(max_iter):
        f0, df0 = f(x0), df(f, x0)
        if abs(df0) < 1e-12:
            break
        x1 = x0 - f0 / df0
        if abs(x1 - x0) < epsilon:
            plots.plot_function_without_points(f, x0)
            return x1, f(x1), _ + 1
        x0 = x1
    plots.plot_function_without_points(f, x0)
    return max_iter


def simple_iterations(f, phi, x0, epsilon, max_iter=1000):
    x = x0
    for _ in range(max_iter):
        x_next = phi(f, x)
        if abs(x_next - x) < epsilon:
            plots.plot_function_without_points(f, x0)
            return x_next, f(x_next), _ + 1
```

```
            x = x_next
        plots.plot_function_without_points(f, x0)
        return max_iter


def newton_system_method(f, J, x0, epsilon, max_iter=1000):
    x = np.array(x0)
    errors = []
    for _ in range(max_iter):
        F = np.array([fi(x) for fi in f])
        J_matrix = J(f, x)
        try:
            J_inv = np.linalg.inv(J_matrix)
        except np.linalg.LinAlgError:
            return None
        delta = np.dot(J_inv, F)
        x_next = x - delta
        error = np.linalg.norm(x_next - x)
        errors.append(error)
        if error < epsilon:
            plots.plot_functions_system(f[0], f[1], x0)
            return x_next, _, errors
        x = x_next
    return max_iter
```

Примеры и результаты работы программы

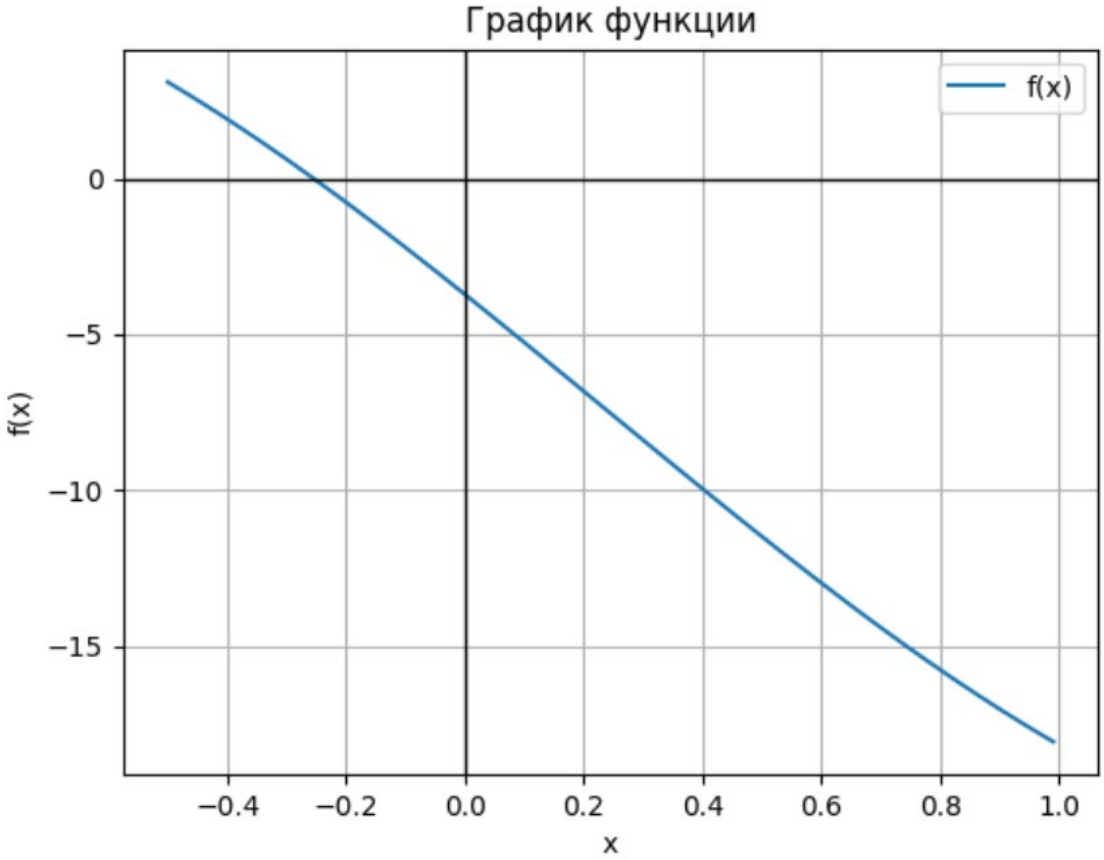
Пример №1. Решение уравнения 2.74 * x**3 - 1.93 * x**2 - 15.28 * x - 3.72 методом хорд с точностью 0,01, границы [-0.5; 1]:

Вывод:

Корень: -0.2545898431759913

Значение функции: -0.00017614244687402802

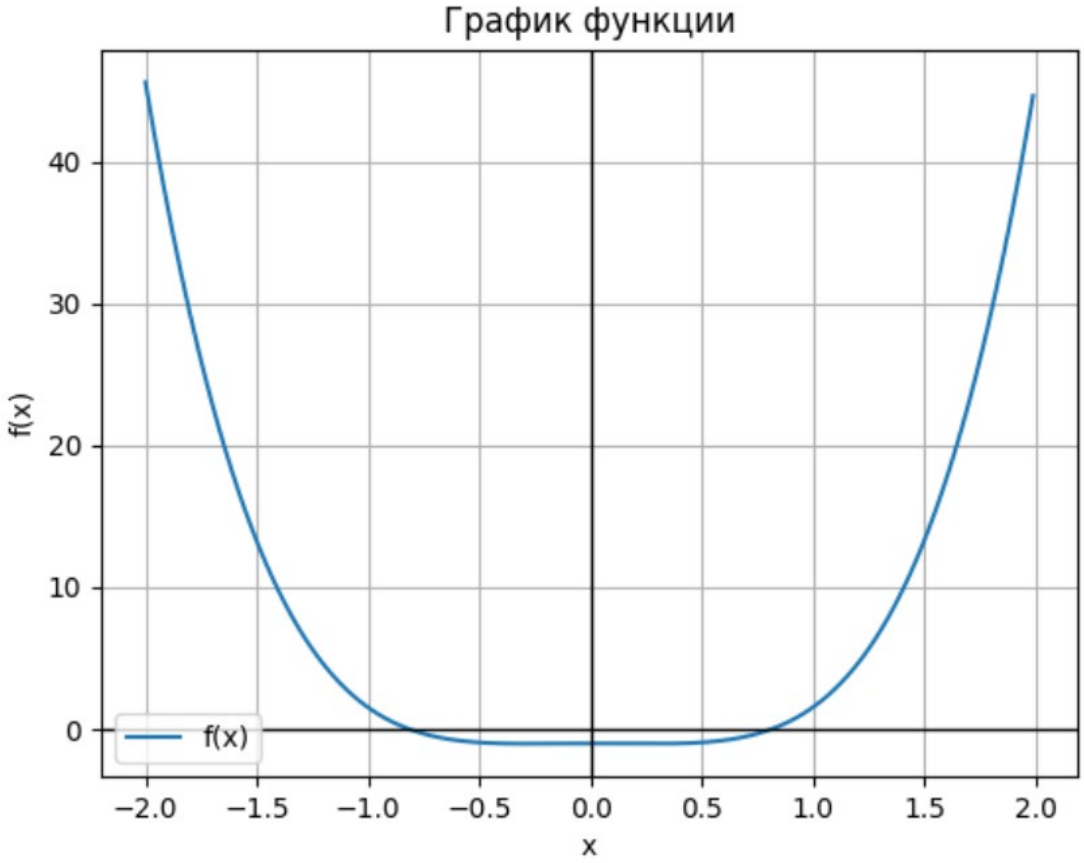
Количество итераций: 3



Пример №2. Решение уравнения cos(x) + 3 * x^4 - 2 методом Ньютона с точностью 0,01, начальное приближение 0:

Вывод:

Не удалось найти корень за 1000 итераций.



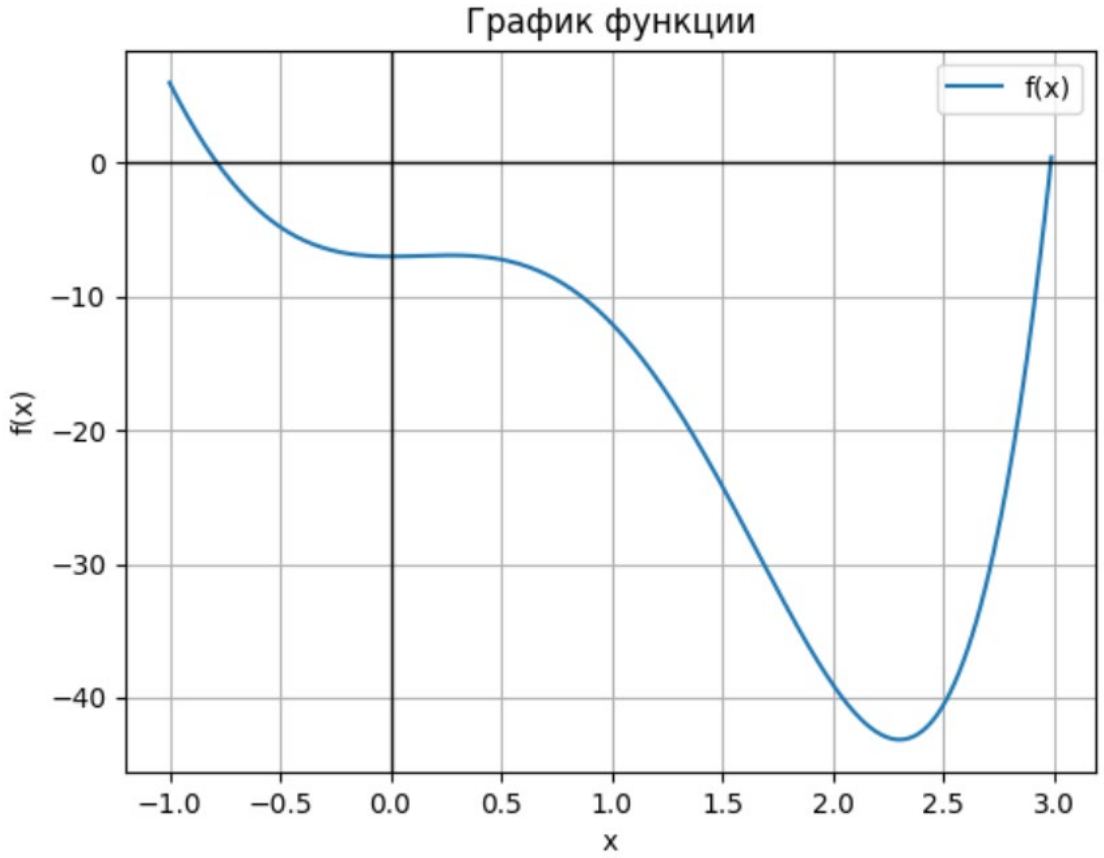
Пример №3. Решение уравнения x^5 - 10 * x^3 + 4 * x^2 - 7 методом простых итераций с точностью 0,1, начальное приближение 1:

Вывод:

Корень: -3.3274839281065893

Значение функции: -2.211954634076349

Количество итераций: 15



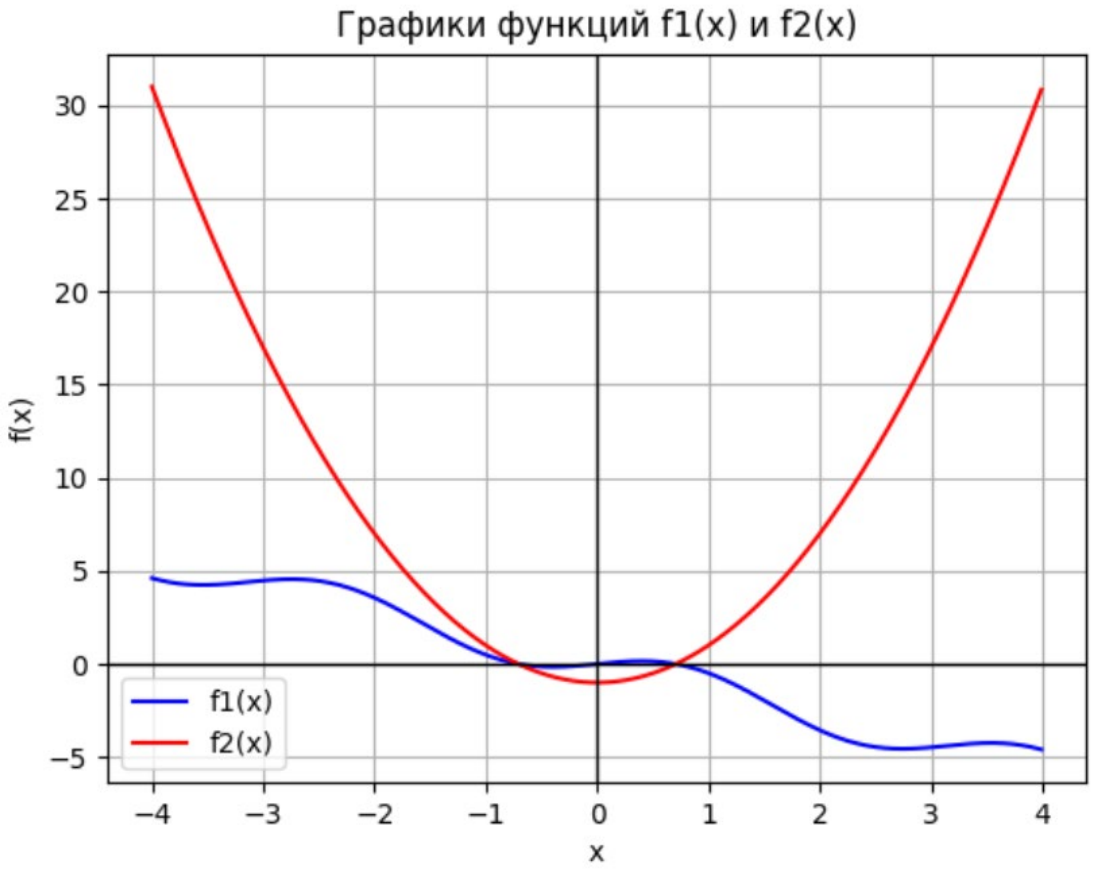
Пример №4. Решение системы уравнений f1(x, y) = sin(x + y) - 1.4x, f2(x, y) = x^2 + y ^2 - 1 методом ньютона с точностью 0,01, начальное приближение x=0, y=1:

Вывод:

Решение системы: [0.70555676 0.70866245]

Количество итераций: 3

Погрешности: [0.9787986992925334, 0.3503284549711896, 0.08470979029734955, 0.0035781018495097726]



Вывод

В процессе выполнения лабораторной работы я вычислил корни нелинейных уравнений и систем нелинейных уравнений, а также программно реализовал несколько методов для нахождения корней нелинейных уравнений и систем нелинейных уравнений. В частности, были рассмотрены следующие методы: метод хорд, метод Ньютона, метод простых итераций, метод половинного деления и метод секущих.

Я создал программу на языке Python, которая решает задачи нахождения корней нелинейных уравнений и систем с использованием популярных методов. В процессе решения задачи была добавлена возможность визуализации результатов с помощью matplotlib, а также функциональность для записи результатов в файл или вывода их в консоль.