Федеральное государственное автономное образовательное учреждение высшего образования **«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа **№2**

**«Численное решение нелинейных уравнений и систем»**

по дисциплине «Вычислительная математика**»**

Вариант: **14**

**Студент:**

Федоров Евгений Константинович

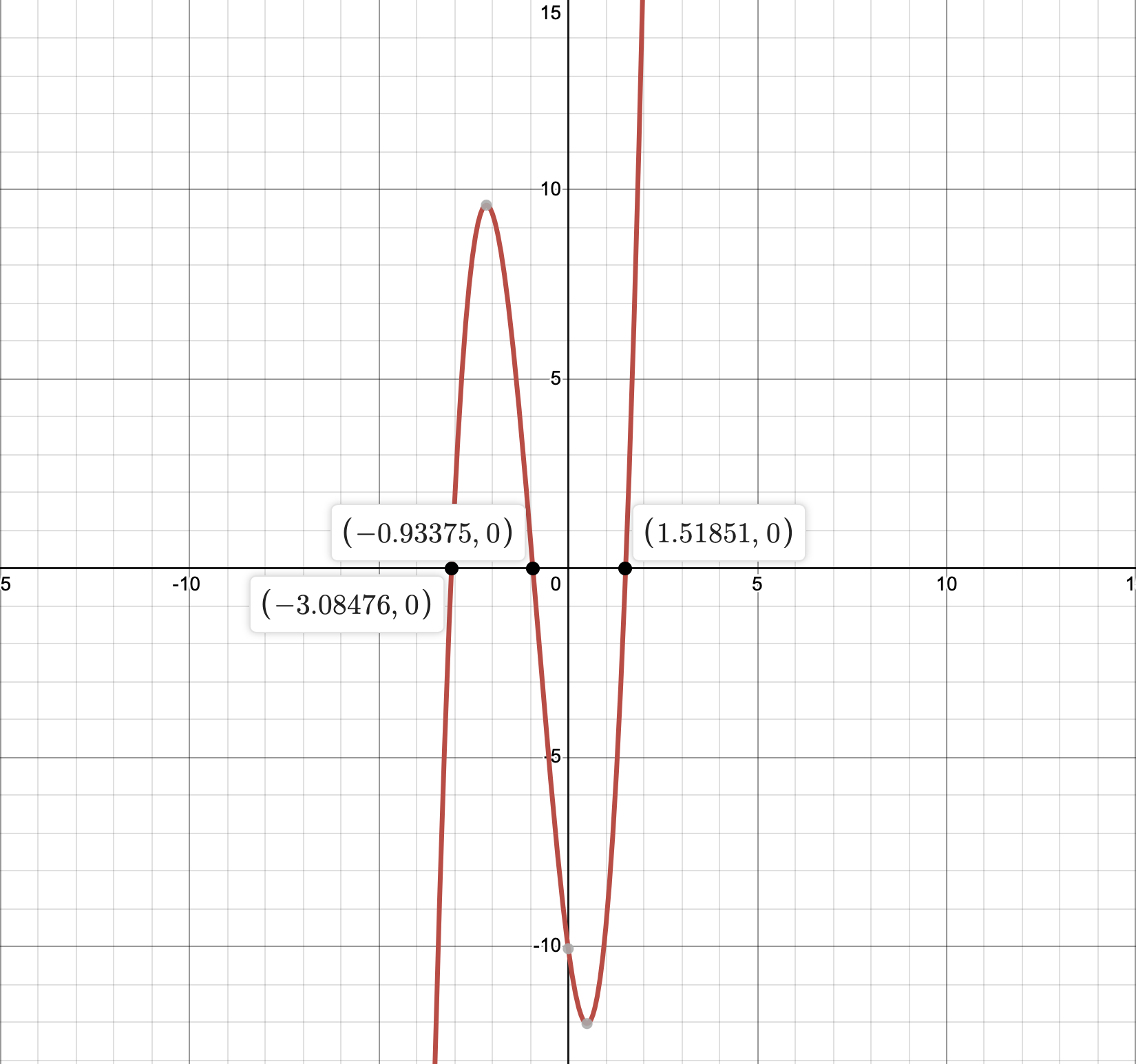
**Преподаватель:**   
Наумова Надежда Алексеевна

Санкт-Петербург, 2025 г.

Цель работы: изучить численные методы решения нелинейных уравнений и их систем, найти корни заданного нелинейного уравнения/системы нелинейных уравнений, выполнить программную реализацию методов.

# 1. Вычислительная реализация задачи

# 1. Решение нелинейного уравнения



Для определения интервалов изоляции корней данного уравнения, можно воспользоваться методом интервалов знакопеременности. Для этого нужно найти значения функции на различных интервалах и определить знак функции на каждом из них.  
  
Получим приближенные значения корней:  
x ≈ -3.1, x ≈ -1, x ≈ 1.6

Теперь нужно разбить ось x на 4 интервала: (-∞, -3.1), (-3.1, -1), (-1, 1.6) и (1.6, +∞). На каждом из этих интервалов нужно определить знак функции.

Для этого можем вычислить значения функции в произвольной точке каждого интервала. Например, для интервала (-∞, -3.1) можно выбрать x = -4, для интервала (-3.1, -1) x = -2, для интервала (-1, 1.6) x = 0, и для интервала (1.6, +∞) x = 2.

Таким образом, получим следующие значения функции:

для x = -4: f(-4) = 2,3(-4)3 + 5,75(-4)2 - 7,41(-4) – 10,06 = -35,62

для x = -2: f(-2) = 2,3(-2)3 + 5,75(-2)2 - 7,41(-2) – 10,06 = 9,36

для x = 0: f(0) = 2,3(0)3 + 5,75(0)2 - 7,41(0) – 10,06 = -10,06

для x = 2: f(2) = 2,3(-4)3 + 5,75(-4)2 - 7,41(-4) – 10,06 = 16,52

Знаки функции на каждом интервале будут соответственно:

|  |  |  |  |
| --- | --- | --- | --- |
| (-∞, -3.1) | (-3.1, -1) | (-1, 1.6) | (1.6, +∞) |
| - | + | - | + |

Интервалы изоляции корней уравнения:

(-4, -2.2), (-2, 0) и (1, 2).

x1 ≈

x2 ≈

x3 ≈

Крайний правый корень – **Метод Ньютона**

Необходимое условие сходиомти f(a) \* f(b) < 0 – выполняется для интервала (-4, -2).

Найдем начальное приближение x0:

x0 =

значит x0 = 2

каждое приближение будем находить по формуле:

Точность возмем равной 0.01.

xi+1= xi -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № итерации | xk | f(xk) | f ’(xk) | xk+1 | | xk+1 - xk | |
| 1 | 2 | 16.52 | 43.19 | 1.62 | 0.38 |
| 2 | 1.62 | 2.80 | 29.33 | 1.52 | 0.1 |
| 3 | 1.52 | 0.04 | 26.01 | 1.52 | 0.01 |

Крайний левый корень – **Метод простых итераций**

Проверим, сходится ли метод на данном интервале:

*f(x)* = = 0

*f ’(x)* = 6,9x2 + 11,5x – 7,41

*f ‘ (a)* = 56,990 > 0

*f ‘ (b)* = 0,6860 > 0  
 = x + f(x)

необходимое условие | => 1 + f ‘ (x) > 1

=-

= x +

= 1 +

= 0

= 0.98

, где q = 0.98

Сходимость будет медленной, так как значение q примерно равно единице.

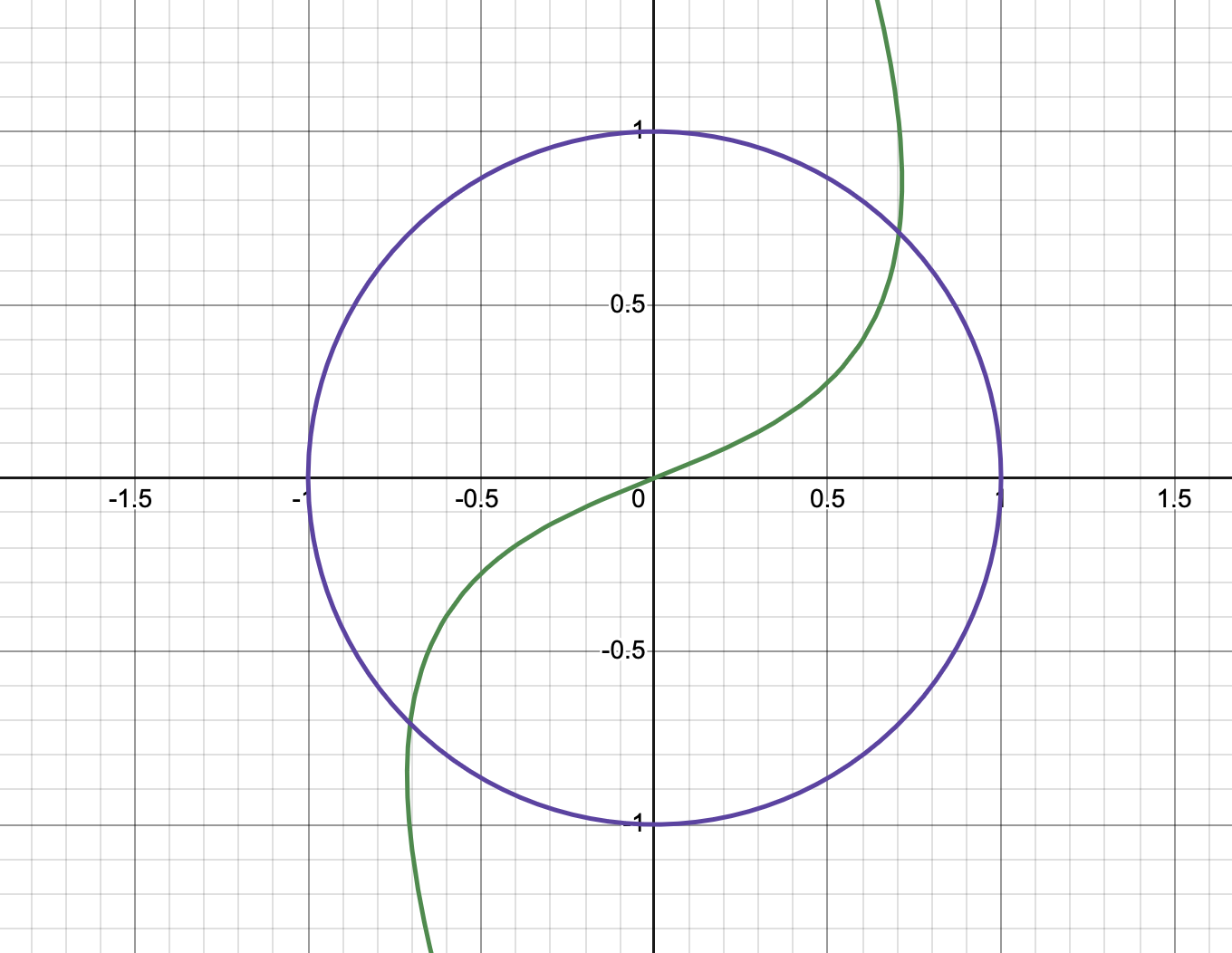
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № итерации | xk | xk+1 | f(xk) | | xk+1 - xk | |
| 1 | -4 | -3.37 | -7.986 | 3.97 |
| 2 | -3.37 | -3.23 | -3.72 | 0.35 |
| 3 | -3.23 | -3.16 | -1.956 | 1.27 |
| 4 | -3.16 | -3.12 | -1.02 | 2.13 |
| 5 | -3.12 | -3.10 | -0.48 | 0.014 |
| 6 | -3.10 | -3.09 | -0.21 | 0.006 |
| 6 | -3.09 | -3.08 | -0.07 | 0.002 |
| 7 | -3.08 | -3.082 | -0.06 | 0.002 |
| 8 | -3.082 | **-3.083** | -0.03 | 0.001 |

Центральный корень – **Метод половинного деления**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | a | b | x | f(a) | f(b) | f(x) | |a – b| |
| 1 | -2.00 | 0 | -1.00 | 9.36 | -10.06 | 0.80 | 2.00 |
| 2 | -1.00 | 0 | -0.50 | 0.80 | -10.06 | -5.21 | 1.002 |
| 3 | -1.00 | -0.50 | -0.75 | 0.80 | -5.205 | -2.24 | 0.501 |
| 4 | -1.00 | -0.75 | -0.875 | 0.80 | -2.24 | -0.71 | 0.253 |
| 5 | -1.00 | -0.88 | -0.9275 | 0.80 | -0.71 | 0.05 | 0.125 |
| 6 | -0.94 | -0.88 | -.0911 | -0.05 | -0.71 | -0.33 | 0.064 |
| 7 | -0.94 | -0.91 | 0.924 | 0.05 | -0.33 | -0.14 | 0.033 |
| 8 | -0.94 | -0.92 | -0.921 | 0.05 | -0.14 | -0.05 | 0.012 |
| 9 | -0.94 | -0.93 | **-0.933** | 0.05 | -0.05 | -0.001 | 0.007 |
| 10 | -0.94 | -0.93 | **-0.935** | 0.05 | -0.002 | 0.021 | 0.003 |
| 11 | -0.94 | -0.93 | **-0.934** | 0.02 | -0.001 | 0.001 | 0.001 |

# 2. Решение системы нелинейных уравнений

1. , Метод Ньютона



Отметим, что решение системы уравнений являются точки пересечения эллипса и

, следовательно, система имеет не более четырех различных решений.

Построим матрицу Якоби:

*, , ,*

**Корень 1:** Шаг 1: Выбираем

Шаг 2. Решаем полученную систему.

Шаг 3. Вычисляем очередные приближения:

+

,

+

,

-

,

# 2. Программная реализация задачи

**Исходный код:** [**https://github.com/2BuRy1/Computational-Maths-Lab2**](https://github.com/2BuRy1/Computational-Maths-Lab2)

**Результаты выполнения программы при различных исходных данных:**

|  |
| --- |
| Выберите метод ввода данных (1 - Файл, 2 - Консоль): 2  Введите команду (solve - решить, exit - выйти): solve  Выберите тип задачи (1 - Нелинейное уравнение, 2 - Система нелинейных уравнений): 1  Выберите уравнение:  1. sin(x) - x/2 = 0  2. 2 \* x\*\*3 + 5.75 \* x \*\* 2 - 7.41 \* x - 10.06 = 0  3. e^x - 3 \* x = 0  Введите номер уравнения: 1  Выберите метод решения:  1. Метод простых итераций  2. Метод Хорд  3. Метод Cекущих  Введите номер метода: 2  Введите интервал (например, -2;2): -2,5;-1  Введите точность (например, 0,001): ,001  Выберите способ вывода результата (1 - консоль, 2 - файл): 1  Результат:  Решение: -1.895494267033981, Значение функции: 0.0, Количество итераций: 25 |
|  |

# Блок схемы реализуемых методов

Метод Хорд:

# Изображение выглядит как текст, снимок экрана, графический дизайн, Шрифт Автоматически созданное описание

Метод Секущих:

Изображение выглядит как диаграмма, зарисовка, рисунок, План

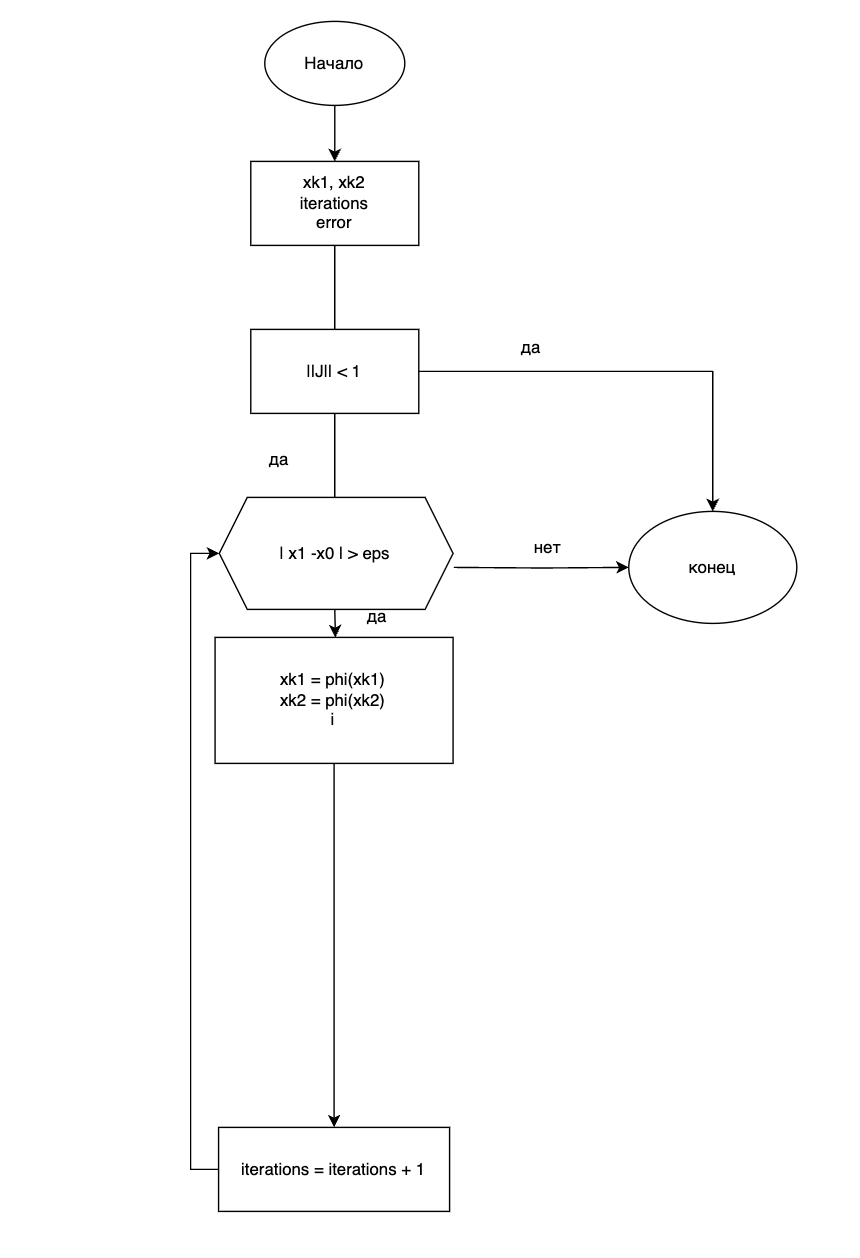
Автоматически созданное описание

Метод простых итераций:

Изображение выглядит как текст, снимок экрана, графический дизайн, Графика

Автоматически созданное описание

Метод простых итераций (система)



# Вывод

В ходе выполнения лабораторной работы были изучены численные методы решения нелинейных уравнений и систем нелинейных уравнений с использованием Python. В результате работы были найдены корни заданных уравнений и систем с использованием различных численных методов, а также были построены графики функций и блок схемы. Было написано приложение с использованием библиотеки TkInter для GUI, так же изучил работу с многопоточностью в данном языке.