Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Факультет программной инженерии и компьютерной техники

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ 2

ЧИСЛЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ И СИСТЕМ

ВАРИАНТ 12

­

Студент: Пышкин Никита Сергеевич, P3213

Преподаватель: Машина Екатерина Алексеевна

Санкт Петербург 2025

Содержание

[**Цель лабораторной работы** 3](#_Toc193301534)

[**Порядок выполнения лабораторной работы** 3](#_Toc193301535)

[**Рабочие формулы используемых методов** 4](#_Toc193301536)

[**Графики функций на исследуемом интервале** 5](#_Toc193301537)

[**Заполненные таблицы вычислительной части 1 лабораторной работы** 5](#_Toc193301538)

[**Подробное решение системы нелинейных уравнений** 8](#_Toc193301539)

[**Листинг программы** 9](#_Toc193301540)

[**Результаты выполнения программы при различных исходных данных** 9](#_Toc193301541)

[**Заключение** 12](#_Toc193301542)

# **Цель лабораторной работы**

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

# **Порядок выполнения лабораторной работы**

**1 Вычислительная реализация задачи:**

Состоит из двух частей и отражается ТОЛЬКО в отчете.

**1 часть. Решение нелинейного уравнения**

Задание:

1. Отделить корни заданного нелинейного уравнения графически (вид уравнения представлен в табл. 6)

2. Определить интервалы изоляции корней.

3. Уточнить корни нелинейного уравнения (см. табл. 6) с точностью ε=10-2.

4. Используемые методы для уточнения каждого из 3-х корней многочлена представлены в таблице 7.

5. Вычисления оформить в виде таблиц (1-5), в зависимости от заданного метода. Для всех значений в таблице удержать 3 знака после запятой.

5.1 Для метода половинного деления заполнить таблицу 1.

5.2 Для метода хорд заполнить таблицу 2.

5.3 Для метода Ньютона заполнить таблицу 3.

5.4 Для метода секущих заполнить таблицу 4.

5.5 Для метода простой итерации заполнить таблицу 5. Проверить условие сходимости метода на выбранном интервале.

6. Заполненные таблицы отобразить в отчете.

**2 часть. Решение системы нелинейных уравнений**

Задание:

1. Отделить корни заданной системы нелинейных уравнений графически (вид системы представлен в табл. 8).

2. Используя указанный метод, решить систему нелинейных уравнений с точностью до 0,01.

3. Для метода простой итерации проверить условие сходимости метода.

4. Подробные вычисления привести в отчете.

**2 Программная реализация задачи:**

**Для нелинейных уравнений:**

1. Все численные методы (см. табл. 9) должны быть реализованы в виде отдельных подпрограмм/методов/классов.

2. Пользователь выбирает уравнение, корень/корни которого требуется вычислить (3-5 функций, в том числе и трансцендентные), из тех, которые предлагает программа.

3. Предусмотреть ввод исходных данных (границы интервала/начальное приближение к корню и погрешность вычисления) из файла или с клавиатуры по выбору конечного пользователя.

4. Выполнить верификацию исходных данных. Необходимо анализировать наличие корня на введенном интервале. Если на интервале несколько корней или они отсутствуют – выдавать соответствующее сообщение. Программа должна реагировать на некорректные введенные данные.

5. Для методов, требующих начальное приближение к корню (методы Ньютона, секущих, хорд с фиксированным концом, простой итерации), выбор начального приближения x0 (а или b) вычислять в программе.

6. Для метода простой итерации проверять достаточное условие сходимости метода на введенном интервале.

7. Предусмотреть вывод результатов (найденный корень уравнения, значение функции в корне, число итераций) в файл или на экран по выбору конечного пользователя.

8. Организовать вывод графика функции, график должен полностью отображать весь исследуемый интервал.

**Для систем нелинейных уравнений:**

1. Пользователь выбирает предлагаемые программой системы двух нелинейных

уравнений (2-3 системы).

2. Организовать вывод графика функций.

3. Начальные приближения ввести с клавиатуры.

4. Для метода простой итерации проверить достаточное условие сходимости.

5. Организовать вывод вектора неизвестных: x1,x2

6. Организовать вывод количества итераций, за которое было найдено решение.

7. Организовать вывод вектора погрешностей:

8. Проверить правильность решения системы нелинейных уравнений.

# **Рабочие формулы используемых методов**

**Рабочие формулы метода секущих:**

Рабочая формула:

Критерий окончания итерационного процесса:

Приближенное значение корня:

**Рабочие формулы метода простых итераций:**

Уравнение приведем к виду

Рабочая формула:

Достаточное условие сходимости:

При – скорость сходимости высокая

При – скорость сходимости низкая

При – нет сходимости

Критерий окончания итерационного процесса:

**Рабочие формулы метода хорд:**

Рабочая формула при фиксированном левом конце хорд:

Рабочая формула при фиксированном правом конце хорд:

Фиксирование конца происходит по следующему принципу:

Критерий окончания итерационного процесса:

# **Графики функций на исследуемом интервале**

# **Заполненные таблицы вычислительной части 1 лабораторной работы**

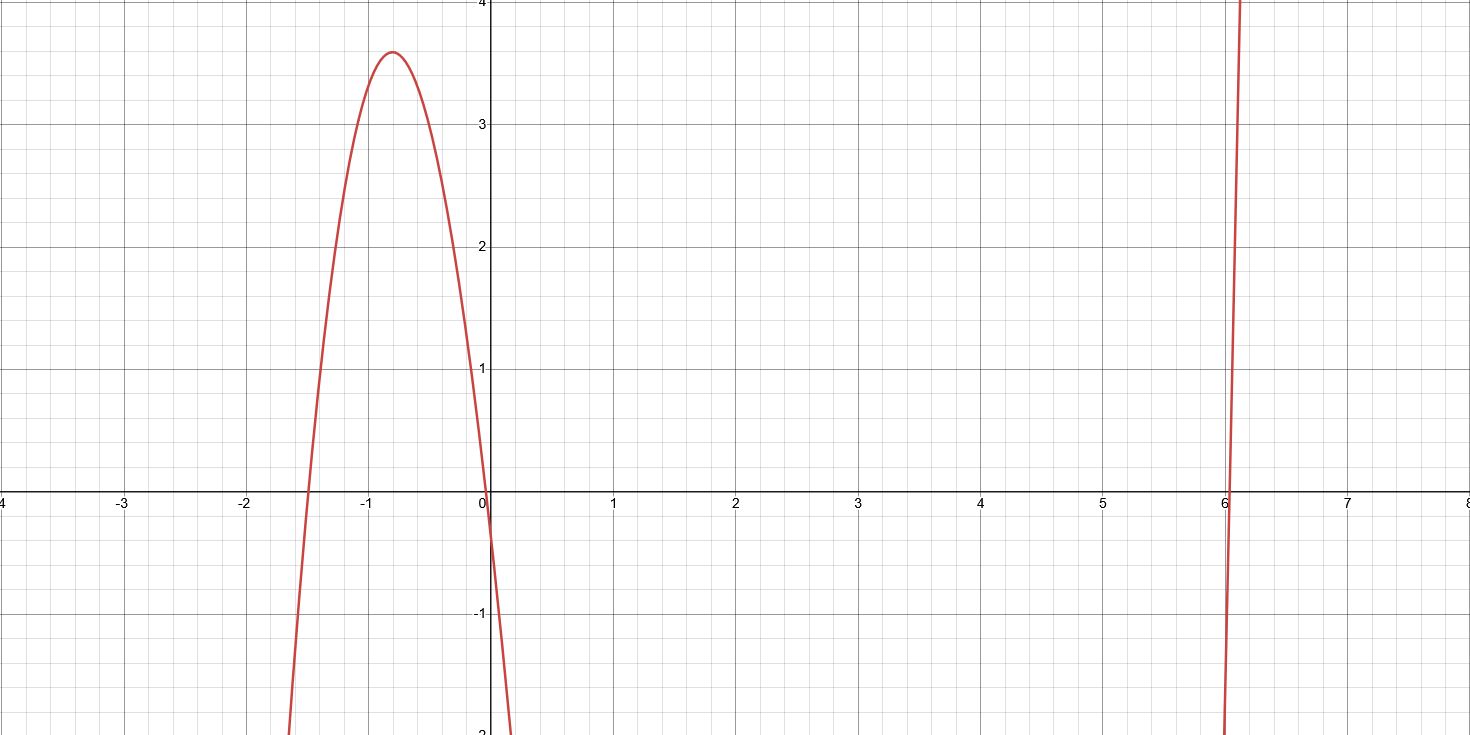
**1) Исследуемая функция**

**2) Определим интервалы изоляции корней**

Наша функция выглядит следующим образом:



Или если рассматривать поближе точки пересечения с осью абсцисс:



Исходя из этого графика аналитическим способом мы можем выделить 3 интервала изоляции корней:

Крайний левый корень: (-2, -1)

Центральный корень: (-0.5; 0.5)

Крайний правый корень: (5.5; 6.5)

**3) Уточним корни нелинейного уравнения с точностью**

Крайний левый корень

Центральный корень:

Крайний правый корень

**4) Вычислим корни используя представленные методы**

Крайний правый корень: уточнение корня уравнения методом секущих

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № итерации |  |  |  |  |  |
| 1 | 5.500 | 6.500 | 5.962 | -3.326 | 0.538 |
| 2 | 6.500 | 5.962 | 6.027 | -0.424 | 0.065 |
| 3 | 5.962 | 6.027 | 6.036 | -0.013 | 0.009 |

Крайний левый корень: уточнение корня уравнения методом простых итераций

Уравнение приведем к виду , для этого

1) Преобразуем уравнение к равносильному (при )

2) Прибавим x в обеих частях

3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № итерации |  |  |  |  |
| 1 | -2.000 | -1.621 | -1.537 | 0.379 |
| 2 | -1.621 | -1.548 | -0.619 | 0.073 |
| 3 | -1.548 | -1.519 | -0.281 | 0.029 |
| 4 | -1.519 | -1.506 | -0.135 | 0.013 |
| 5 | -1.506 | -1.500 | -0.068 | 0.006 |

Центральный корень: уточнение корня уравнения хорд

Наш интервал (-0.5; 0.5)

– фиксируем конец a

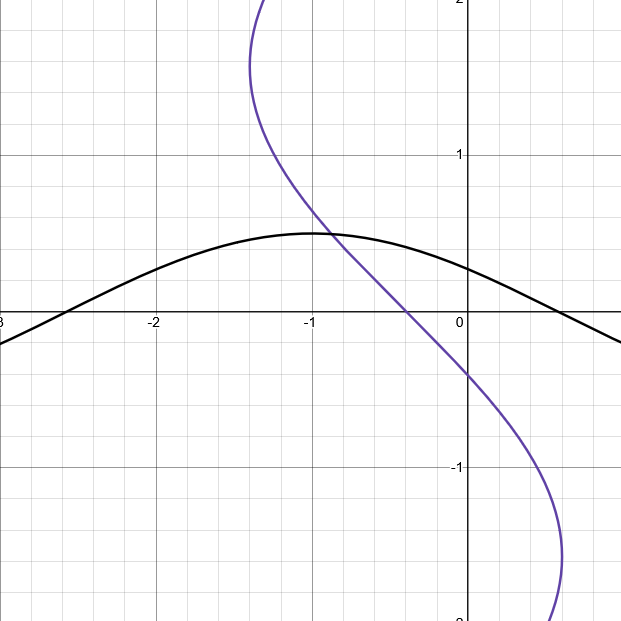
Используем формулу

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № шага |  |  |  |  |  |  |  |
| 1 | -0.500 | 0.500 | 0.400 | 2.972 | -5.988 | -4.723 | 0.100 |
| 2 | -0.500 | 0.400 | -0.152 | 2.972 | -4.723 | 0.909 | 0.552 |
| 3 | -0.500 | -0.152 | 0.001 | 2.972 | 0.909 | -0.392 | 0.153 |
| 4 | -0.500 | 0.001 | -0.057 | 2.972 | -0.392 | 0.127 | 0.058 |
| 5 | -0.500 | -0.057 | -0.037 | 2.972 | 0.127 | -0.048 | 0.020 |
| 6 | -0.500 | -0.037 | -0.044 | 2.972 | -0.048 | 0.013 | 0.007 |

# **Подробное решение системы нелинейных уравнений**

**1) Уравнение для решения методом простых итераций**

**2) Графически отделим корни заданной системы**



Решение системы уравнений находится в квадрате: -1 < x < 0, 0 < y < 1

**3) Проверим условие сходимости**

Процесс сходится

**4) Вычисляем корни**

В качестве начального приближения возьмем (0, 0)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № итерации |  |  |  |  |  |  |
| 1 | 0.0 | -0.4 | 0.0 | 0.27 | 0.4 | 0.27 |
| 2 | -0.4 | -0.667 | 0.27 | 0.413 | 0.267 | 0.143 |
| 3 | -0.667 | -0.801 | 0.413 | 0.473 | 0.134 | 0.06 |
| 4 | -0.801 | -0.855 | 0.473 | 0.49 | 0.054 | 0.018 |
| 5 | -0.855 | -0.871 | 0.49 | 0.495 | 0.016 | 0.005 |
| 6 | -0.871 | -0.875 | 0.495 | 0.496 | 0.004 | 0.001 |

# **Листинг программы**

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

**Пример 1:**

Введите номер типа решаемой задачи:

1. Линейное уравнение

2. Система нелинейных уравнений

Введите номер: 1

Выберите метод:

1. Метод половинного деления

2. Метод Ньютона

3. Метод простой итерации

Введите номер метода: 1

Выберите функцию:

1. x\*\*2 - 0.5

2. x\*\*3 - 4\*x + 1

3. x\*\*20 - 10\*x\*\*19 - 3\*x\*\*14 + 0.124\*x\*\*10 + x\*\*7 - x\*\*3 - 1

4. 2\*\*x - 4

Введите номер функции: 1

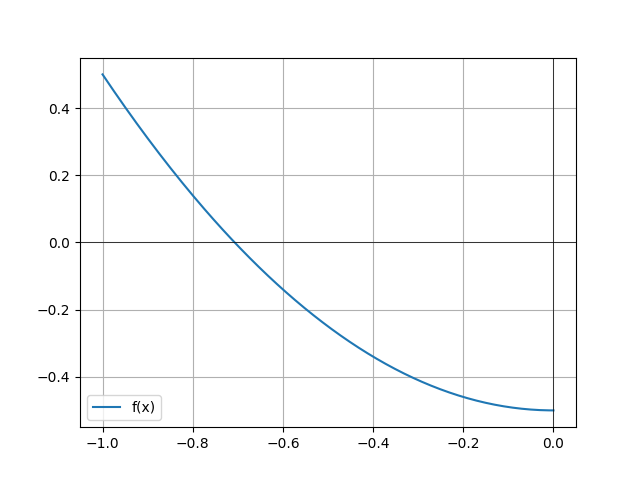
Введите границы интервала a и b через пробел (либо введите путь до файла с числами): -1 0

Получен ручной ввод: -1 0

Введите точность вычисления (либо введите путь до файла с этим числом): 0.0001

Получен ручной ввод: 0.0001

Найденный корень: -0.7071533203125, f(корень) = 6.58184289932251e-05, итераций: 13



**Пример 2:**

Введите номер типа решаемой задачи:

1. Линейное уравнение

2. Система нелинейных уравнений

Введите номер: 2

Выберите систему:

1)

x + sin(y) + 0.4 = 0

y - cos(x + 1)/2 = 0

2)

x\*\*2 + y\*\*2 - 4 = 0

-3\*x\*\*2 + y = 0

3)

-y + cos(x) = 0

x\*\*2 - y = 0

Введите номер системы: 1

Система будет решена методом Ньютона.

Введите начальное приближение x0 y0 (либо введите путь до файла с числами): 0 0

Получен ручной ввод: 0 0

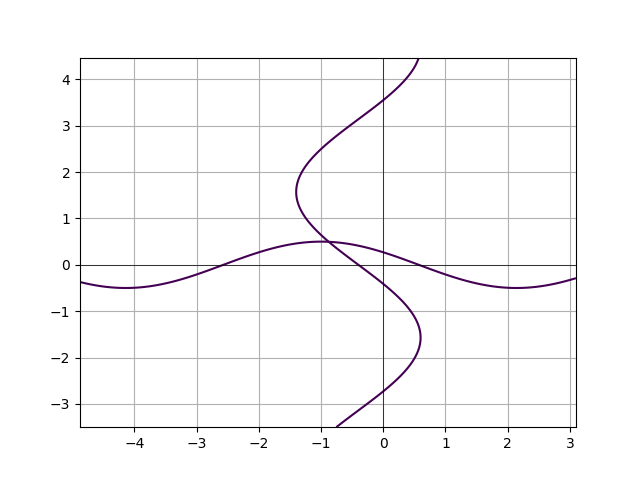
Введите точность вычисления (либо введите путь до файла с этим числом): 0.00001

Получен ручной ввод: 0.00001

Вектор неизвестных: [-0.876055948823948, 0.49616438208948005]

Вектор погрешностей: [4.6043069246159973e-08, -2.9135045098716006e-08]

Число итераций: 4



# **Заключение**

В результаты выполнения данной лабораторной работы были изучены методы решения нелинейных уравнений и систем нелинейных уравнений с использованием Python