МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«Южно-Уральский государственный университет

(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

**ОТЧЕТ**

о выполнении практического задания № 3

по дисциплине

«Вычислительные методы»

|  |  |
| --- | --- |
|  | Выполнил:  студент группы КЭ-202  Доблер А.М.  Проверил: Старший преподаватель кафедры МОИТ  Гаврилова Т.П. |

Челябинск – 2025

ОГЛАВЛЕНИЕ

[1. Теоретическая часть: 2](#_Toc197410382)

[1.1. Словесная постановка задачи: 2](#_Toc197410383)

[1.2. Алгоритмы 2](#_Toc197410384)

[1.3. Проверка условия производной для метода простых итераций 3](#_Toc197410385)

[2. Практическая часть 4](#_Toc197410386)

[2.1. Листинг №1: 4](#_Toc197410387)

[2.2. Листинг № 2: 4](#_Toc197410388)

[2.3. Листинг № 3: 4](#_Toc197410389)

[2.4. Листинг № 4: 5](#_Toc197410390)

[Вывод: 6](#_Toc197410391)

[Список литературы 7](#_Toc197410392)

1. Теоретическая часть:
   1. Словесная постановка задачи:

Найти решение уравнения на заданном отрезке [a;b] с точностью e.

1) методом простой итерации

2) методом Ньютона

Вариант 4



Рисунок 1 Вариант № 4

* 1. Алгоритмы

Метод простых итераций

* + 1. Общий вид уравнения

Дано уравнения вида f(x) = 0. Чтобы применить метод простой итерации, его необходимо преобразовать к эквивалентному виду:

где — некоторая функция, называемая итерационной.

* + 1. Условия сходимости

Для сходимости метода необходимо, чтобы выполнялось условие:

Если это условие не выполняется, уравнение нужно преобразовать иначе.

* + 1. Алгоритм метода
* Вычисляем следующее приближение по формуле:
* Процесс повторяем до тех пор, пока не будет достигнута заданная точность:

Где — допустимая погрешность.

Метод Ньютона (метод касательных)

1. Общий вид уравнения

Дано уравнение f(x)=0. Метод Ньютона использует не только значение функции, но и её производную.

2. Условие сходимости:

* функция f(x) должна быть дифференцируемой;
* начальное приближение x0 должно быть достаточно близко к корню;
* производная f′(x) не должна обращаться в ноль вблизи корня.

3. Алгоритм метода:

1. Выбираем начальное приближение x0.
2. Вычисляем следующее приближение по формуле:
3. Итерации продолжаем до достижения заданной точности:
   * 1. Геометрическая интерпретация - на каждой итерации метод Ньютона проводит касательную к графику функции f(x) в точке и находит точку её пересечения с осью x, которая становится новым приближением .
   1. Проверка условия производной для метода простых итераций

Выражаю x чтобы

При вычислениях данного уравнения в методе простых итераций будем использовать выражение .

1. Практическая часть

Программа реализована на языке python без использования внешних библиотек.

* 1. Листинг №1:

Функции для обозначения функции x и её производной

# начальная функция

def f(x):

return 0.25 \* x \*\* 3 + x - 1.2502

# производная

def df(x):

return 0.75 \* x \*\* 2 + 1

* 1. Листинг № 2:

Функция реализующая метод простой итерации.

def simple\_iteration(phi, x0, stnd\_eps, max\_iter=500):

x\_prev = x0

for i in range(max\_iter):

x\_next = phi(x\_prev)

if abs(x\_next - x\_prev) < stnd\_eps:

return x\_next, i + 1

x\_prev = x\_next

raise ValueError("Метод не сошелся за максимальное число итераций")

* 1. Листинг № 3:

Функция реализующая метод Ньютона.

def newton(f, df, x0, eps, max\_iter=500):

x\_prev = x0

for i in range(max\_iter):

f\_val = f(x\_prev)

df\_val = df(x\_prev)

x\_next = x\_prev - f\_val / df\_val

if abs(x\_next - x\_prev) < eps:

return x\_next, i + 1

x\_prev = x\_next

raise ValueError("Метод не сошелся за максимальное число итераций")

* 1. Листинг № 4:

Основной блок программы.

if \_\_name\_\_ == "\_\_main\_\_":

a = 0

b = 2

phi = lambda x: pow((1.2502 - x), (1 / 3))

x0 = (a+b) / 2 # Начальное приближение

eps = 0.0001

# Метод простых итераций

try:

root\_simple, iter\_simple = simple\_iteration(phi, x0, eps)

print(f"Метод простой итерации: x = {root\_simple:.6f}, итераций: {iter\_simple}")

except Exception as e:

print(f"\nОшибка в методе простой итерации: {e}")

# Метод Ньютона

try:

root\_newton, iter\_newton = newton(f, df, x0, eps)

print(f"Метод Ньютона: x = {root\_newton:.6f}, итераций: {iter\_newton}")

except Exception as e:

print(f"\nОшибка в методе Ньютона-Лейбнице: {e}")

Вывод:

В ходе выполнения работы были изучены и применены на практике два численных метода решения нелинейных уравнений: метод простой итерации и метод Ньютона. Исследование проводилось на примере уравнения 0.25x³ + x - 1.2502 = 0 на отрезке [0; 2] с точностью ε = 0.0001.

Основные результаты работы:

Сравнение эффективности методов показало:

* Метод простой итерации сошелся за 15 шагов
* Методу Ньютона потребовалось всего 2 итерации

Анализ характеристик методов выявил:

* Метод простой итерации проще в реализации и более устойчив
* Метод Ньютона демонстрирует более высокую скорость сходимости
* Метод Ньютона требует вычисления производной и чувствителен к начальному приближению

Список литературы

* Амосов, А. А. Вычислительные методы / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. — 5-е изд., стер. — Санкт-Петербург : Лань, 2023. — 672 с. — ISBN 978-5-507-47808-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/327497 (дата обращения: 08.04.2025). — Режим доступа: для авториз. пользователей.