**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«Сибирский государственный университет науки и технологий   
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Вычислительная математика

|  |
| --- |
| Вычисление действительных корней уравнения с одним неизвестным |

Руководитель Сенашов А. В.

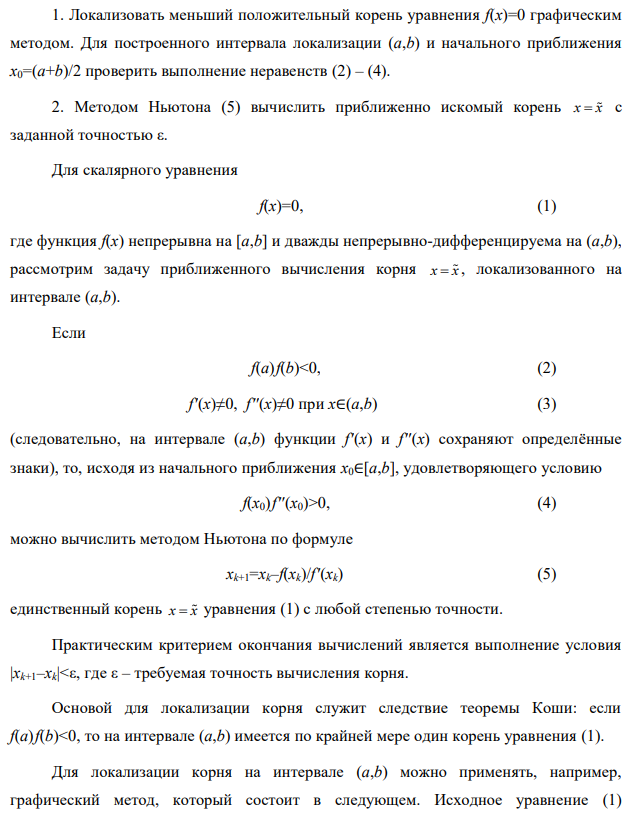
подпись, дата инициалы, фамилия

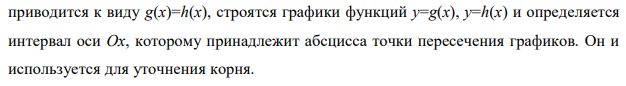
Обучающийся БПИ22-02, 221219040 К.В. Трифонов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2023 г.

# постановка задачи





*Вариант* 24:



# ХОД РАБОТЫ

import math

import matplotlib.pyplot as plt

# Определение функции

def f(x):

    return 3-0.5\*x\*\*0.5-math.exp(-0.5\*x\*\*2)

# Создание списка значений x

x\_values = [i/100 for i in range(0, 501)]  # Генерация значений от 0 до 5 с шагом 0.01

# Вычисление значений функции

y\_values = [f(x) for x in x\_values]

# Построение графика

plt.figure(figsize=(8, 6))

plt.plot(x\_values, y\_values, label='$f(x) = 3-0,5\sqrt{x}-e^{-0,5x^2}$')

plt.xlabel('x')

plt.ylabel('f(x)')

plt.grid(True)

plt.legend()

plt.title('График функции f(x)')

plt.show()

# Определение производной функции f(x)

def f\_prime(x):

    return (math.exp(-(x\*\*2)/2)\*(4\*x\*\*(3/2)-math.exp((x\*\*2)/2)))/(4\*(x\*\*0.5))

# Определение второй производной функции f(x)

def f\_double\_prime(x):

    return (math.exp(-(x\*\*2)/2)\*((x\*\*0.5)\*math.exp((x\*\*2)/2)-8\*x\*\*4+8\*x\*\*2))/(8\*x\*\*2)

# Определение метода Ньютона для вычисления корня

def newton\_method(f, f\_prime, x\_0, epsilon, max\_iterations=1000):

    x\_prev = x\_0

    for \_ in range(max\_iterations):

        x\_next = x\_prev - f(x\_prev) / f\_prime(x\_prev)

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

            return x\_next

        x\_prev = x\_next

    return None

# Локализация интервала и начального приближения

a = 1

b = 2

x\_0 = (a + b) / 2

# Проверка условий

if f(a) \* f(b) < 0 and f\_double\_prime(x\_0) != 0 and f\_prime(x\_0) != 0 and f(x\_0) \* f\_double\_prime(x\_0) > 0:

# Вычисление корня методом Ньютона

    root = newton\_method(f, f\_prime, x\_0, epsilon=0.01)

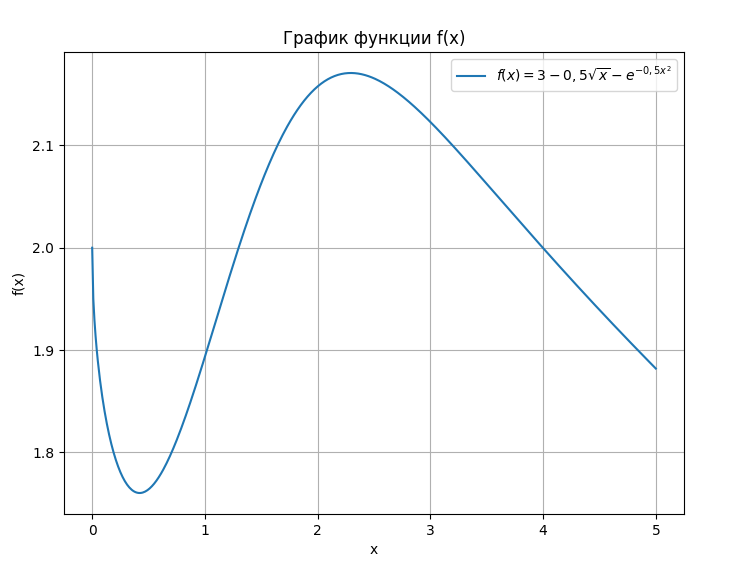
    print("корень приблизительно равен", root)

    # Проверка

    print(f(root))

else:

   print("Условия для применения метода Ньютона не выполняются.")



Условия для применения метода Ньютона не выполняются (нет пересечений с f(x) = 0).