



Министерство науки и высшего образования Российской Федерации  
Калужский филиал федерального государственного автономного  
образовательного учреждения высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИУК Информатика и управление

КАФЕДРА ИУК4 Программное обеспечение ЭВМ, информационные технологии

## **ПРАКТИЧЕСКАЯ РАБОТА**

### **«Методы минимизации»**

**по дисциплине: «Методы машинного обучения»**

Выполнил: студент группы ИУК4-72Б

\_\_\_\_\_  
(Подпись)

Губин Е.В.

\_\_\_\_\_  
(И.О. Фамилия)

Проверил:

\_\_\_\_\_  
(Подпись)

Семененко М.Г.

\_\_\_\_\_  
(И.О. Фамилия)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

**Цель:** изучить и реализовать метод половинного деления для нахождения минимума функции одной переменной и сравнить его эффективность с встроенными методами оптимизации.

**Задачи:**

- Реализовать алгоритм метода половинного деления для поиска минимума функции.
- Применить метод к заданной функции
- Построить график функции с отмеченной точкой минимума и сравнить результат с встроенной функцией Python.

#### Вариант №4

**Формулировка задания:**

1. Написать блок-схему алгоритма нахождения минимума функции методом половинного деления.
2. Методом половинного деления найти минимум функции  $J(u)$  на отрезке  $[-10; 10]$  (по вариантам). Построить график функции и показать на нем точку минимума.  
Функция:  $u^2 + ae^{bu}$ ;  $a = 4$ ;  $b = -0.25$ .
3. Сравнить результат с результатом использования встроенной функции.

**Блок схема метода дихотомии:**

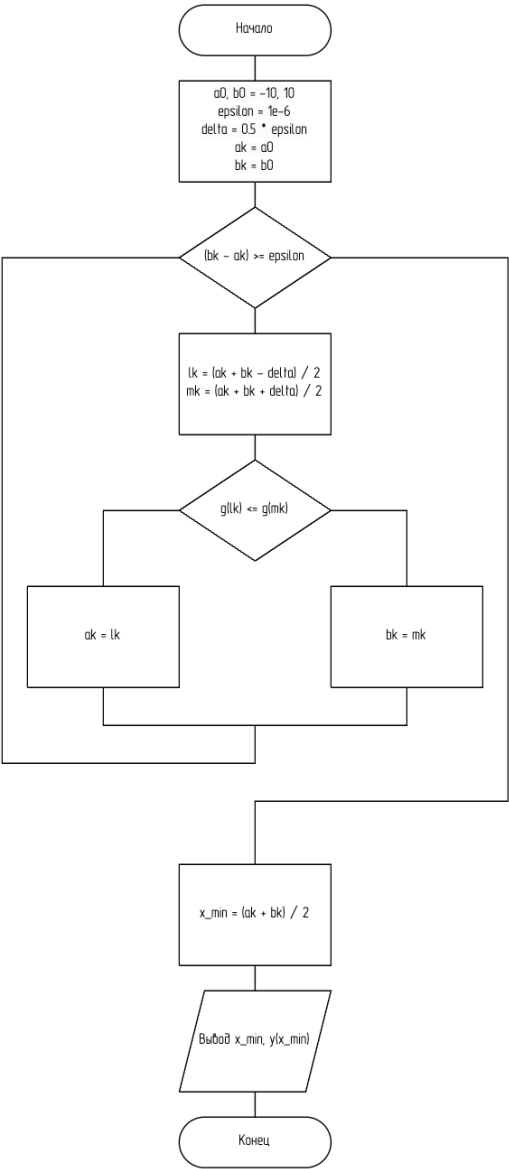


Рисунок 1 Блок схема метода дихотомии

**Результаты выполнения программы:**

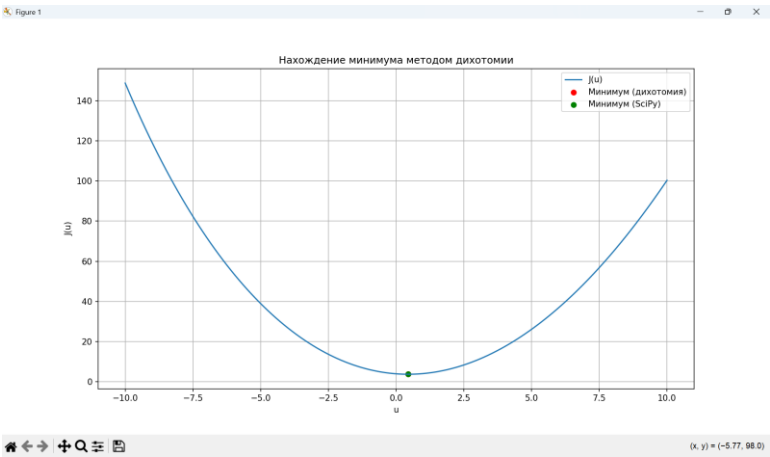
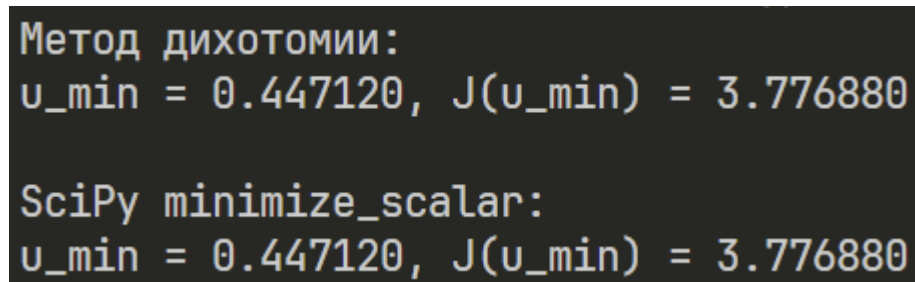


Рисунок 2 Нахождение минимума методом дихотомии

На Рисунках 2 и 3 видно, что минимум на отрезке, найденный по методу дихотомии приблизительно равен минимуму, найденному с помощью встроенной функции Python `scipy.optimize.minimize_scalar`.



```
Метод дихотомии:
u_min = 0.447120, J(u_min) = 3.776880

SciPy minimize_scalar:
u_min = 0.447120, J(u_min) = 3.776880
```

Рисунок 3 Сравнение результата нахождения минимума по методу дихотомии с встроенной функцией поиска минимума

### Листинг программы:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize_scalar

a = 4
b = -0.25

def J(u):
    return u**2 + a * np.exp(b * u)

def dichotomy_method(func, left, right, eps=1e-5, delta=1e-4,
max_iter=10000):
    iteration = 0
    while abs(right - left) > eps and iteration < max_iter:
        x1 = (left + right - delta) / 2
        x2 = (left + right + delta) / 2
        f1, f2 = func(x1), func(x2)

        if f1 < f2:
            right = x2
        else:
            left = x1

        iteration += 1

    return (left + right) / 2

u_min = dichotomy_method(J, -10, 10)
J_min = J(u_min)

res = minimize_scalar(J, bounds=(-10, 10), method='bounded')

print("Метод дихотомии:")
print(f"u_min = {u_min:.6f}, J(u_min) = {J_min:.6f}")
print("\nSciPy minimize_scalar:")
print(f"u_min = {res.x:.6f}, J(u_min) = {res.fun:.6f}")

u = np.linspace(-10, 10, 400)
plt.plot(u, J(u), label='J(u)')
plt.scatter(u_min, J_min, color='red', label='Минимум (дихотомия)')
plt.scatter(res.x, res.fun, color='green', label='Минимум (SciPy)')
plt.title("Нахождение минимума методом дихотомии")
```

```
plt.xlabel("u")  
plt.ylabel("J(u)")  
plt.legend()  
plt.grid(True)  
plt.show()
```

**Вывод:** в ходе лабораторной работы был реализован алгоритм поиска минимума функции методом дихотомии (с заданным максимальным количеством итераций) и произведено сравнение с результатом поиска минимума встроенной библиотечной функцией.