|  |  |
| --- | --- |
|  | **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 | |  |  | Губин Е.В. | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Проверил: | |  |  | Семененко М.Г. | |
|  | | (Подпись) |  | (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: ; a = 4; b = -0.25.
3. Сравнить результат с результатом использования встроенной функции.

**Блок схема метода дихотомии:**

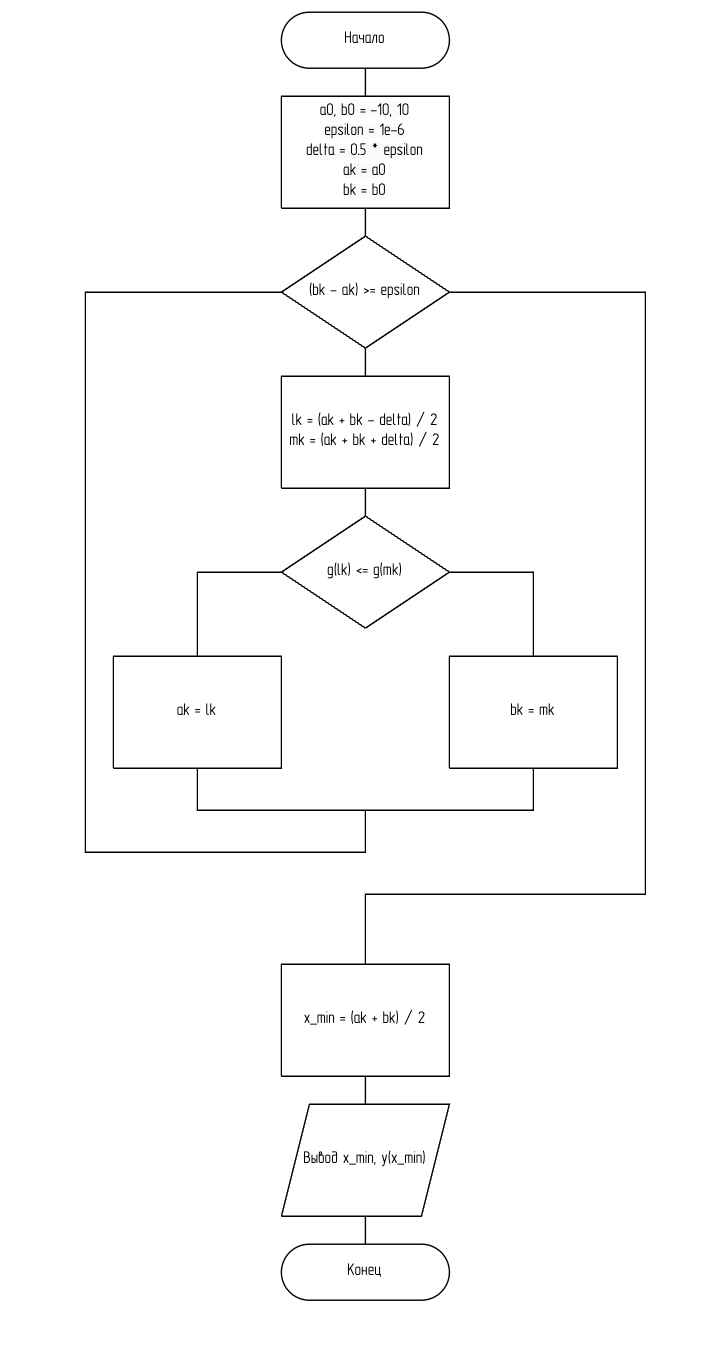


Рисунок Блок схема метода дихотомии

**Результаты выполнения программы:**

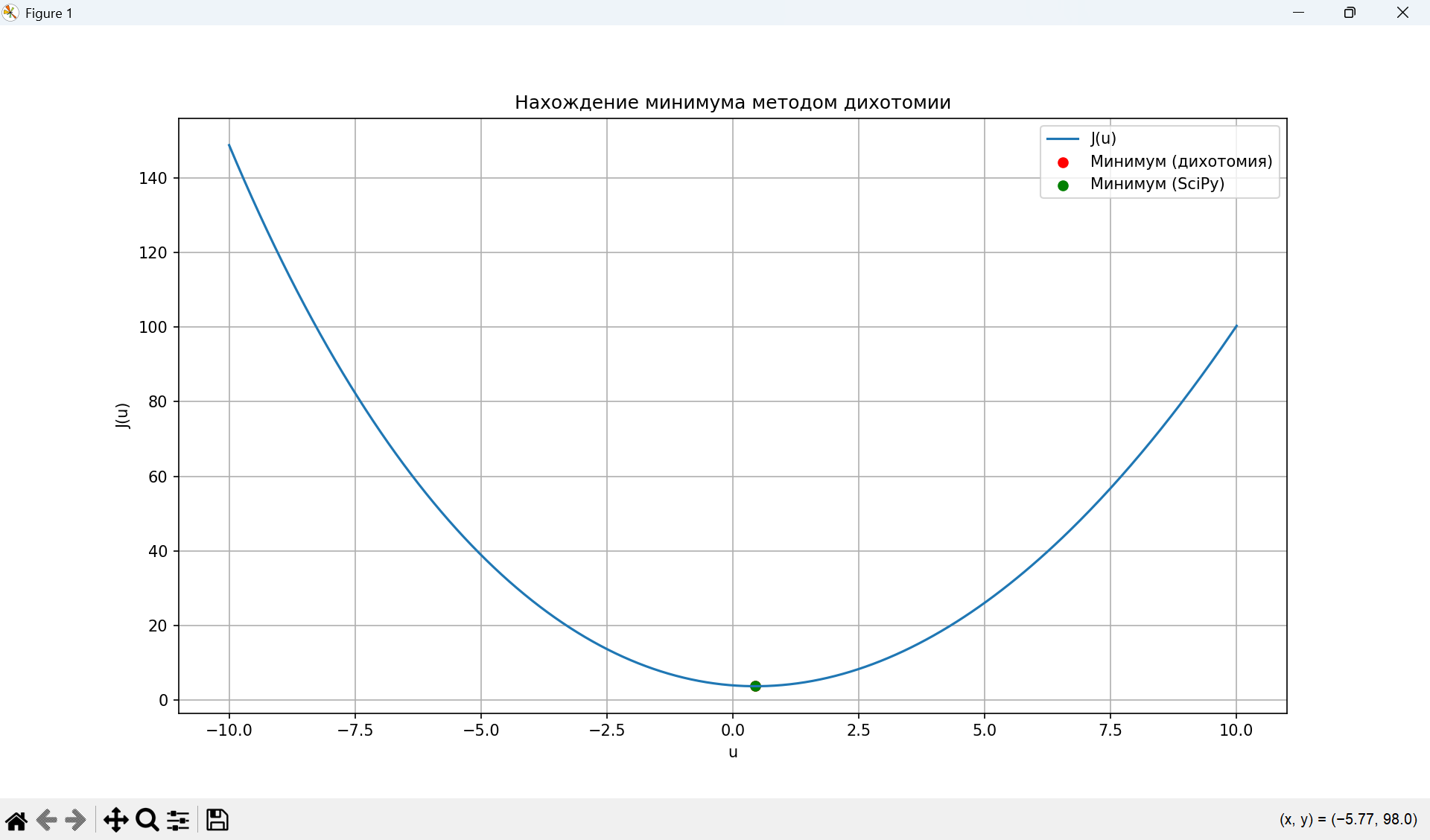


Рисунок Нахождение минимума методом дихотомии

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.

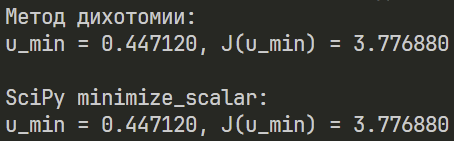


Рисунок Сравнение результата нгахождения минимума по методу дихотомии с встроенной функцией поиска минимума

**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.