Домашняя работа №1
по дисциплине "Методы оптимизации"
Вариант IV

Винницкая Дина Сергеевна

Группа: Б9122-02.03.01сцт

Задание

1. Найти наименьшее и наибольшее значение функции на отрезке $[0, 1]$, используя аналитические методы.
2. Найти минимальное значение функции, используя методы дихотомии и золотого сечения.
3. Провести сравнение по числу вычислений функции для достижения точности ϵ.
4. Построить график зависимости количества вычислений целевой функции от логарифма задаваемой точности ϵ.

1 Найти наименьшее и наибольшее значение функции на отрезке $[0, 1]$

Аналитическая (почти) часть

Дана функция $f(x) = \frac{1}{2}x^2 - \sin(x)$, отрезок $[0, 1]$ и точность $\epsilon = 0.03$.

Для нахождения наименьшего значения найдём точку минимума:

$$f'(x) = x - \cos(x) = 0$$

Точка минимума задаётся уравнением $x = \cos(x)$, решение которого единственное и аналитически не получаемое. Это иррациональная константа, известная как "число Дотти оно примерно равно $x_{\min} \approx 0.739085133215$.

В этой точке функция принимает значение:

$$f(x_{\min}) \approx -0.400488612113$$

Это значение также не может быть выражено точно.

На границах отрезка получаем следующие значения:

$$f(0) = 0$$

$$f(1) = \frac{1}{2} - \sin(1) \approx -0.341470984808$$

Таким образом, видно, что наибольшее значение функции на данном отрезке выражается точно и равно 0, в то время как минимальное значение является иррациональной константой, полученной численно.

2 Найти минимальное значение функции, используя методы дихотомии и золотого сечения.

Для нахождения минимального значения функции $f(x) = \frac{x^2}{2} - \sin(x)$ на интервале $[0, 1]$ были реализованы два численных метода: метод дихотомии и метод золотого сечения. Эти методы позволяют эффективно находить минимум функции при заданной точности ϵ.

Реализация была выполнена на языке программирования Python, который был выбран за его простоту и удобство для работы с численными методами. Важным моментом является подсчет количества вызовов функции в каждом методе, что позволяет оценить их эффективность.

```python
from math import *

def f(x):
    return x ** 2 / 2 - sin(x)

n = 0
k = 0

def bisection_method(a_, b_, eps):
    global n
    a = a_
    b = b_

    while abs(b - a) > eps:
        n += 1
        c = (a + b) / 2
        fa = f(a)
        fc = f(c)

        if fa * fc < 0:
            b = c
        else:
            a = c

    return f((a + b) / 2)

phi = 1.6180339887

def golden_mean(a_, b_, eps):
    global k
    a = a_
    b = b_

    x1 = b - (b - a) / phi
    x2 = a + (b - a) / phi
    f1 = f(x1)
    f2 = f(x2)

    while abs(b - a) > eps:
        k += 1
        if f1 > f2:
            a = x2
            x2 = x1
            f2 = f1
            x1 = a + (b - a) / phi
            f1 = f(x1)
        else:
            b = x1
            x1 = x2
            f1 = f2
            x2 = b - (b - a) / phi
            f2 = f(x2)

    return f((a + b) / 2)

print(bisection_method(0, 1, 0.03), golden_mean(0, 1, 0.03))
```

1. **Метод дихотомии** последовательно делит отрезок пополам и выбирает половину, где функция имеет минимальное значение, пока длина отрезка не станет меньше заданной точности. 2. **Метод золо-**

того сечения использует пропорции золотого сечения для выбора точек на отрезке, позволяя быстрее находить минимум, чем при дихотомии.

Оба метода возвращают минимальное значение функции и количество вычислений, необходимых для достижения точности.

3 Провести сравнение по числу вычислений функции для достижения точности ϵ

Результаты

- **Метод дихотомии**: -0.34500621887850935
- **Метод золотого сечения**: -0.3998597257012174

Вывод

Из представленных данных следует, что метод дихотомии требует большего числа вычислений для достижения заданной точности и при этом демонстрирует менее точные результаты по сравнению с методом золотого сечения.

4 Построить график зависимости количества вычислений целевой функции от логарифма задаваемой точности ϵ

```python
import matplotlib.pyplot as plt
from math import log

epsilons = [0.1, 0.05, 0.03, 0.01, 0.005, 0.001, 0.0005, 0.0001]

dichotomy_calls = []
golden_calls = []

for eps in epsilons:
    n = 0
    k = 0

    bisection_method(0, 1, eps)
    dichotomy_calls.append(n)

    golden_mean(0, 1, eps)
    golden_calls.append(k)

plt.step([log(eps) for eps in epsilons], dichotomy_calls,
plt.step([log(eps) for eps in epsilons], golden_calls,
label="Golden Section Method", where='mid', color='red')

plt.xlabel(r'$\ln(\epsilon)$', fontsize=18, fontstyle='italic')
plt.ylabel(r'$n$', fontsize=18, fontstyle='italic')
plt.legend()
plt.grid(True)

plt.xticks(fontsize=12, fontstyle='italic')
plt.yticks(fontsize=12, fontstyle='italic')
plt.show()
```

В коде производится расчет минимального значения функции $f(x) = \frac{x^2}{2} - \sin(x)$ на интервале $[0, 1]$ с использованием методов дихотомии и золотого сечения для различных значений точности ϵ. Подсчитывается количество вызовов функции для каждого метода, а затем строится график зависимости количества вычислений от логарифма точности ϵ. График наглядно сравнивает эффективность методов по числу итераций при изменении точности.

График

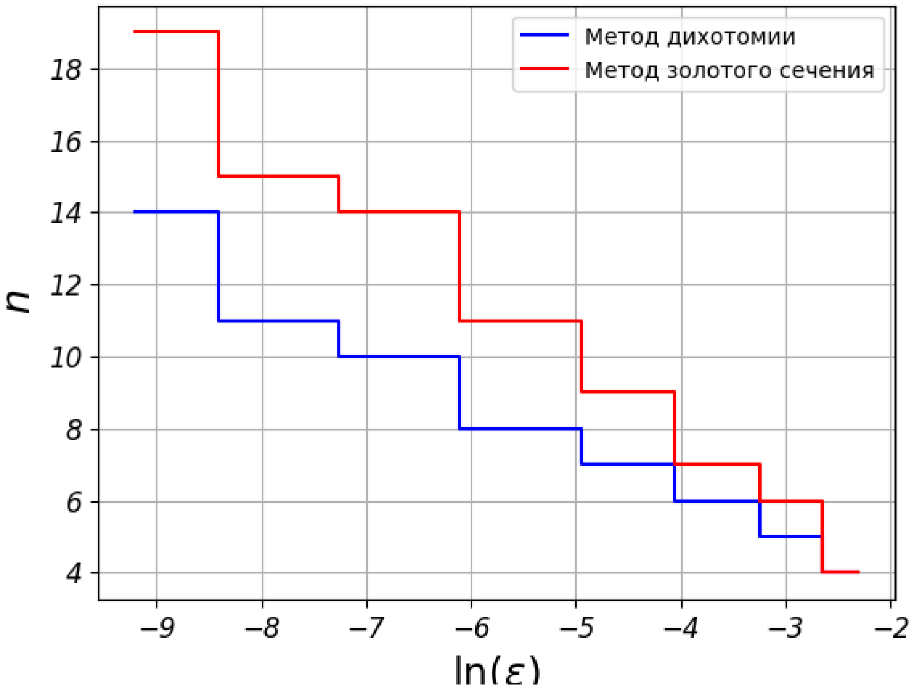

Рис. 1: График

5 Вывод

В ходе лабораторной работы были реализованы методы дихотомии и золотого сечения для нахождения минимального значения функции. Проведенное сравнение показало, что метод золотого сечения демонстрирует более высокую эффективность, требуя меньшее количество вычислений по сравнению с методом дихотомии при одинаковой точности ϵ. Таким образом, метод золотого сечения является предпочтительным для задач оптимизации на заданных интервалах при необходимости снижения вычислительных затрат.