Министерство науки и высшего образования РФ ФГАОУ ВПО

Национальный исследовательский технологический университет «МИСИС»

Институт Информационных технологий и компьютерных наук (ИТКН)

Кафедра автоматизированных систем управления (АСУ)

Отчет по лабораторной работе №2

по дисциплине «Методы оптимизаций»

Вариант 5

Выполнил:

Студент: Безыкорнов Н.Б.

         Группа: БИВТ-20-4

Проверил:

Лычев А.В.

Москва, 2023

**Цель работы:**

Приобретение практических навыков для решения задач одномерной минимизации численными методами.

**Постановка задачи:**

Вариант 5

Требуется найти безусловный минимум функции одной переменной y = f(x) на отрезке [a, b], где функция является выпуклой. То есть найти такую точку x ∗ ∈ [a, b], что f(x ∗ ) = min x∈[a,b] f(x). Методы, рассмотренные в лабораторной работе 1, используются при минимальных требованиях к целевой функции y = f(x) — она должна быть унимодальной. В данной работе предполагается, что целевая функция y = f(x) является выпуклой и дифференцируемой (один раз или дважды). Причём, производные могут быть вычислены в произвольно выбранных точках.

По условию использовались методы под номерами 1,2.

a = 0.5

b = 1

погрешность = 0.0001

Выполнение работы:

Рис. 1 – график функции

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название метода | Число итераций | Количество вычислений функций | Найденное решение | Значение функции |
| Метод средней точки | 13 | 18 | 0.703 | 3.442 |
| Метод хорд | 6 | 29 | 0.703 | 3.442 |

1. Метод средней точки

import math  
  
  
def func(x):  
 return math.exp(x) + 1 / x  
  
  
def prime(x):  
 global q  
 q += 1  
 return math.exp(x) - 1 / (x \*\* 2)  
  
  
def midpoint\_method(a, b, eps):  
 global k  
 mid = (b + a) / 2  
 while abs(b - a) > eps:  
 k += 1  
 if prime(mid) > 0:  
 b = mid  
 mid = (b + a) / 2  
 elif prime(mid) < 0:  
 a = mid  
 mid = (b + a) / 2  
 return (b + a) / 2  
  
  
a = 0.5  
b = 1.0  
eps = 0.0001  
k, q = 0, 0  
  
min\_x = midpoint\_method(a, b, eps)  
min\_y = func(min\_x)  
  
print(min\_x)  
print(min\_y)

Листинг 1 – Метод равномерного поиска

Изображение выглядит как текст

Автоматически созданное описание

Рис. 2 – результат работы программы

1. Метод хорд

import math  
  
  
def f(x):  
 global q  
 q += 1  
 return math.exp(x) + 1 / x  
  
  
def df(x):  
 global q  
 q += 1  
 return math.exp(x) - 1 / x \*\* 2  
  
  
def secant\_method(a, b, eps):  
 global k  
 a1, a2, = a, b  
 a3 = a1 - df(a1) \* (a1 - a2) / (df(a1) - df(a2))  
 while abs(df(a3)) > eps or abs(a1 - a2) > eps:  
 k += 1  
 a1 = a2  
 a2 = a3  
 a3 = a1 - df(a1) \* (a1 - a2) / (df(a1) - df(a2))  
  
 return a3  
  
  
q, k = 0, 0  
a, b = 0.5, 1.0  
x\_min = secant\_method(a, b, 0.0001)  
print("Кол-во итераций {}, кол-во вычислений функции {}".format(k, q))  
print("Минимум функции f(x) на отрезке [{}, {}]: x = {}, f(x) = {}".format(a, b, x\_min, f(x\_min)))

Листинг 2 – Метод хорд

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3 – результат работы программы

Сравнительная характеристика:

Оба алгоритма позволяют найти минимум функции за достаточно короткое время.  
Вывод:

Я ознакомился с некоторыми методами нахождения минимумов функций.