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 №6

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 многомерной минимизации различными численными методами нулевого порядка.

**Постановка задачи:**

Требуется найти минимум функции многих переменных y = f(x1, . . . , xn), то есть такую точку x ∗ ∈ U, что f(x ∗ ) = min x∈U f(x), где множество точек U определяется ограничениями вида

Вариант 5.

В лабораторной работе использовались методы № 3

Функция:

-8 \* x \*\* 2 + 4 \* x - y \*\* 2 + 12 \* y – 7

Ограничение:

2x + 3y = 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название метода | Число итераций | Количество вычислений функций | Найденное решение | Значение функции |
| Метод проекции градиента | 21 | 43 | [0.25, 1.83] | 12.1389 |

Ход работы:

1. Метод проекции градиента

import numpy as np  
  
  
def f(x, y):  
 global calcs  
 calcs += 1  
 return -8 \* x \*\* 2 + 4 \* x - y \*\* 2 + 12 \* y - 7  
  
  
def constraint(x):  
 return (6 - 2 \* x) / 3  
  
  
def gradient\_with\_projekcia(start\_point, step\_size, eps):  
 global iters  
 x = start\_point[0]  
 y = start\_point[1]  
 x\_all = [x]  
 y\_all = [y]  
  
 while True:  
 iters += 1  
 grad\_x = -16 \* x + 4  
 grad\_y = -2 \* y + 12  
 x\_new = x + step\_size \* grad\_x  
 y\_new = y + step\_size \* grad\_y  
 if constraint(x\_new) < y\_new:  
 y\_new = constraint(x\_new)  
 if abs(f(x\_new, y\_new) - f(x, y)) < eps:  
 break  
  
 x = x\_new  
 y = y\_new  
 x\_all.append(x)  
 y\_all.append(y)  
 return (x, y, x\_all, y\_all)  
  
  
iters, calcs = 0, 0  
start\_point = np.array([0, 0])  
step\_size = 0.1  
precision = 0.0001  
x\_max, y\_max, x\_history, y\_history = gradient\_with\_projekcia(start\_point, step\_size, precision)  
  
print(f'Максимум в x = {x\_max:.2f}, y = {y\_max:.2f}. F = {f(x\_max, y\_max)}')  
print(f"Итерации: {iters} \nВычисления функции: {calcs}")

листинг 1 – реализация алгоритма

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис 1 – результат работы программы c начальной точкой 0 0

Изображение выглядит как круг, снимок экрана, линия, искусство

Автоматически созданное описание

Рис 2 – графический результат работы программы c начальной точкой 0 0

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рис 3 – результат работы программы c начальной точкой 7 7

Изображение выглядит как линия, Красочность, диаграмма, снимок экрана

Автоматически созданное описание

Рис 4 – графический результат работы программы c начальной точкой 7 7

Вывод:

Метод проекции градиента является простым и эффективным методом оптимизации, который быстрее методов градиентного спуска благодаря включению проекции на допустимое множество, что ускоряет сходимость и уменьшает количество вычислений функций. В данной задаче метод проекции градиента достиг решения после 21 итерации с общим количеством 43 вычислений функции. Найденное решение удовлетворяет заданной точности и имеет координаты x=0.25, y=1.83, а значение функции f(x,y) в этой точке равно 12.1389.