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 №4

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 градиентными методами и методами второго порядка.

**Постановка задачи:**

Вариант 5

Требуется найти безусловный минимум функции многих переменных y = f(x1, . . . , xn), то есть такую точку x ∗ ∈ R n , что f(x ∗ ) = min x∈Rn f(x).

По условию использовались методы под номерами 2, 4.

**Ход работы:**

1. метод Флетчера-Ривза

import numpy as np  
  
  
def f(x):  
 global q  
 q += 1  
 return ((x[0] - 5) \*\* 2) \* ((x[1] - 4) \*\* 2) + (x[0] - 5) \*\* 2 + (x[1] - 4) \*\* 2 + 1  
  
  
def grad\_f(x):  
 global q  
 q += 1  
 return np.array([4 \* (x[0] - 5) \* ((x[1] - 4) \*\* 2) + 2 \* (x[0] - 5),  
 4 \* (x[1] - 4) \* ((x[0] - 5) \*\* 2) + 2 \* (x[1] - 4)])  
  
  
def fletcher\_reeves(x0, eps1, eps2, M):  
 global j  
 k = 0  
 x = x0  
 grad = grad\_f(x)  
 while np.linalg.norm(grad) > eps1 and k < M:  
 j += 1  
 if k == 0:  
 beta = 0  
 else:  
 beta = np.dot(grad, grad) / np.dot(grad\_prev, grad\_prev)  
 d = -grad + beta \* d\_prev if k > 0 else -grad  
 tk = 1.0  
 while f(x + tk \* d) > f(x) + eps2 \* tk \* np.dot(grad, d):  
 tk \*= 0.5  
 x\_next = x + tk \* d  
 grad\_prev = grad  
 grad = grad\_f(x\_next)  
 if (np.linalg.norm(x\_next - x) <= eps2 and  
 abs(f(x\_next) - f(x)) <= eps2):  
 x = x\_next  
 break  
 k += 1  
 x = x\_next  
 d\_prev = d  
 return x  
  
  
j, q = 0, 0  
answer = fletcher\_reeves([0, 1], 0.001, 0.001, 100)  
print("Итерации: {}, Вычисления: {}".format(j, q))  
print("Минимум функции находится в [{};{}]. Min = {}".format(round(answer[0], 4), round(answer[1], 4),  
 round(f(answer), 4)))

Листинг 1 – Метод Флетчера-Ривза

Изображение выглядит как текст

Автоматически созданное описание

Рис. 1 – результат работы программы с начальной точной [0; 1]

Изображение выглядит как текст

Автоматически созданное описание

Рис. 2 – результат работы программы с начальной точной [1; 1]

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3 – результат работы программы с начальной точной [0; 0]

1. метод Давидона-Флетчера-Пауэлла

import numpy as np  
  
  
def f(x):  
 global q  
 q += 1  
 return ((x[0] - 5) \*\* 2) \* ((x[1] - 4) \*\* 2) + (x[0] - 5) \*\* 2 + (x[1] - 4) \*\* 2 + 1  
  
  
def grad\_f(x):  
 global q  
 q += 1  
 return np.array(  
 [2 \* (x[0] - 5) \* ((x[1] - 4) \*\* 2) + 2 \* (x[0] - 5), 2 \* (x[1] - 4) \* ((x[0] - 5) \*\* 2) + 2 \* (x[1] - 4)])  
  
  
def DFP\_method(x0, eps, max\_iter):  
 global j  
 k = 0  
 H = np.identity(2)  
 x = x0  
 grad = grad\_f(x)  
 while np.linalg.norm(grad) >= eps and k < max\_iter:  
 j += 1  
 d = -np.dot(H, grad)  
 gamma = 0.1  
 while f(x + gamma \* d) > f(x) + 0.1 \* gamma \* np.dot(grad, d):  
 gamma \*= 0.5  
 x\_new = x + gamma \* d  
 grad\_new = grad\_f(x\_new)  
 Delta\_x = x\_new - x  
 Delta\_y = grad\_new - grad  
 H = H + np.outer(Delta\_x, Delta\_x) / np.dot(Delta\_x, Delta\_y) - np.dot(np.dot(H, np.outer(Delta\_y, Delta\_y)),  
 H) / np.dot(np.dot(Delta\_y, H), Delta\_y)  
 x = x\_new  
 grad = grad\_new  
 k += 1  
 return x, f(x)  
  
  
x0 = np.array([0, 0])  
eps = 1e-6  
max\_iter = 100  
j, q = 0, 0  
x\_min, f\_min = DFP\_method(x0, eps, max\_iter)  
  
print("Кол-во итераций:", j, "Кол-во вычислений функции:", q)  
print("Минимум функции: x = {}, f(x) = {}".format(x\_min, f\_min))

Листинг 2 – Метод Давидона-Флетчера-Пауэлла

Изображение выглядит как текст

Автоматически созданное описание

Рис. 4 – результат работы программы с начальной точной [0; 0]

Изображение выглядит как текст

Автоматически созданное описание

Рис. 5 – результат работы программы с начальной точной [1; 0]

Изображение выглядит как текст

Автоматически созданное описание

Рис. 6 – результат работы программы с начальной точной [1; 1]

**Вывод:**

В лабораторной работе были реализованы методы оптимизации: Давидона-Флетчера-Пауэлла и Флетчера-Ривза. Я ознакомился с принципами работы этих методов и их применением для решения задач оптимизации функций.