Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
Московский государственный технический университет имени Н.Э. Баумана

Факультет: Информатика и системы управления
Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №6
по курсу «Численные методы»
«Метод наискорейшего спуска поиска минимума функции многих переменных»

Выполнила:
студентка группы ИУ9-62Б
Самохвалова П. С.

Проверила:
Домрачева А. Б.

Москва, 2023

Цель:

Анализ метода наискорейшего спуска поиска минимума функции многих переменных.

Постановка задачи:

Дано: Функция нескольких переменных $f(x_1, x_2, ..., x_n)$ и некоторое начальное приближение $x^0 = (x_1^0, ..., x_n^0)$.

Найти: Минимум функции нескольких переменных с заданной точностью.

Тестовый пример:

Вариант 21

$$f(x) = (1 + 2x_1^2 + x_2^2)^{\frac{1}{2}} + e^{x_1^2 + 2x_2^2} - x_1 x_2, \quad x^0 = (0, 0).$$

$$\varepsilon = 0.001$$

Описание метода:

Пусть для функции $f(x_1, x_2, ..., x_n)$ на k-м шаге имеем некоторое приближение к минимуму $x^k = (x_1^k, ..., x_n^k)$. Рассмотрим функцию одной переменной

$$\varphi_k(t) = f(x_1^k - t\frac{\partial f}{\partial x_1}(x^k), ..., x_n^k - t\frac{\partial f}{\partial x_n}(x^k)) = f(x^k - t grad f(x^k)),$$

где вектор $grad f(x^k) = (\frac{\partial f}{\partial x_1}(x^k), ..., \frac{\partial f}{\partial x_n}(x^k))$ - градиент функции f в точке x^k.

Обозначим точку минимума функции $\varphi_k(t)$ через t^*. Теперь для следующего приближения к точке экстремума полагаем

$$x^{k+1} = x^k - t^* grad f(x^k).$$

Процесс поиска минимума продолжаем до тех пор, пока $||grad f(x^k)|| = \max_{1 \le i \le n} |\frac{\partial f}{\partial x_i}(x^k)|$ не станет меньше допустимой погрешности ε.

В большинстве случаев точно искать минимум функции $\varphi_k(t)$ не нужно и достаточно ограничиться лишь одним приближением. Тогда особенно простым будет вид итерации в двумерном случае

$$(x_{k+1}, y_{k+1}) = (x_k - t^k\frac{\partial f}{\partial x}, y_k - t^k\frac{\partial f}{\partial y}),$$

где $t^k = \frac{\phi_k'(0)}{\phi_k''(0)}$, $\quad \phi_k'(0) = -(\frac{\partial f}{\partial x})^2 - (\frac{\partial f}{\partial y})^2$, $\quad \phi_k''(0) = \frac{\partial f^2}{\partial x^2}(\frac{\partial f}{\partial x})^2 + 2\frac{\partial f^2}{\partial x \partial y}\frac{\partial f}{\partial x}\frac{\partial f}{\partial y} + \frac{\partial f^2}{\partial y^2}(\frac{\partial f}{\partial y})^2$, все производные беруться в точке (x_k, y_k).

Листинг 1. Метод наискорейшего спуска поиска минимума функций двух переменных

```
import math


def f(x1, x2):
    return (1 + 2 * x1**2 + x2**2)**0.5 + math.exp(x1**2 + 2 * x2**2) - \
```

```python
        x1 * x2


def df_dx1(x1, x2):
    return 2 * x1 * math.exp(x1**2 + 2 * x2**2) + 2 * x1 / \
        ((2 * x1**2 + x2**2 + 1)**0.5) - x2


def df_dx2(x1, x2):
    return 4 * x2 * math.exp(x1**2 + 2 * x2**2) + x2 / \
        ((2 * x1**2 + x2**2 + 1)**0.5) - x1


def d2f_dx1x1(x1, x2):
    return 4 * x1**2 * math.exp(x1**2 + 2 * x2**2) - 4 * x1**2 / \
        ((2 * x1**2 + x2**2 + 1)**1.5) + 2 * math.exp(x1**2 + 2 * x2**2) + \
        2 / ((2 * x1**2 + x2**2 + 1)**0.5)


def d2f_dx2x2(x1, x2):
    return 16 * x2**2 * math.exp(x1**2 + 2 * x2**2) - x2**2 / \
        ((2 * x1**2 + x2**2 + 1)**1.5) + 4 * math.exp(x1**2 + 2 * x2**2) + \
        1 / ((2 * x1**2 + x2**2 + 1)**0.5)


def d2f_dx1x2(x1, x2):
    return 8 * x1 * x2 * math.exp(x1**2 + 2 * x2**2) - 2 * x1 * x2 / \
        ((2 * x1**2 + x2**2 + 1)**1.5) - 1


def mns(xk, yk, eps):
    k = 0
    df_dx = df_dx1(xk, yk)
    df_dy = df_dx2(xk, yk)
    max_df_dxi = max(abs(df_dx), abs(df_dy))

    while max_df_dxi >= eps:
        df_dx = df_dx1(xk, yk)
        df_dy = df_dx2(xk, yk)
```

```python
        d2f_dxdx = d2f_dx1x1(xk, yk)
        d2f_dydy = d2f_dx2x2(xk, yk)
        d2f_dxdy = d2f_dx1x2(xk, yk)
        phi1 = -df_dx ** 2 - df_dy ** 2
        phi2 = d2f_dxdx * df_dx ** 2 + 2 * d2f_dxdy * df_dx * df_dy + \
               d2f_dydy * df_dy ** 2
        tk = - phi1 / phi2
        xk = xk - tk * df_dx
        yk = yk - tk * df_dy
        max_df_dxi = max(abs(df_dx), abs(df_dy))
        k += 1

    return k, xk, yk


eps = 0.001

xk = 0
yk = 0

n, x, y = mns(xk, yk, eps)
print(n)
print(x, y)

print()

xk = 1
yk = 0

n, x, y = mns(xk, yk, eps)
print(n)
print(x, y)
```

Результаты работы:

Если начать итерации из точки (0, 0), заданная точность достигается сразу же, в точке (0, 0).

Если начать итерации из точки (1, 0), заданная точность достигается за 7 итераций, в точке (1.2646227829257956e-05, 1.623834843454447e-05).

Выводы:

В результате выполнения лабораторной работы был изучен метод наискорейшего спуска поиска минимума функции двух переменных, была написала реализация данного метода на языке программирования Python. На тестовом примере было получено, что, начиная итерации из точки (0, 0), заданная точность достигается сразу же, в точке (0, 0). Начиная итерации из точки (1, 0), заданная точность достигается за 7 итераций, в точке (1.2646227829257956e-05, 1.623834843454447e-05).