Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

Лабораторная работа № 5

«Метод наискорейшего спуска поиска минимума функции многих переменных»

по курсу «Численные методы»

Выполнил:
студент группы ИУ9-61Б
Митрошкин Алексей
Проверила:
Домрачева А. Б.

Москва, 2024

1. Цель

Целью данной работы является изучение метода наискорейшего спуска для поиска минимума функции многих переменных и сравнение полученного результата со значением минимума функции, найденным аналитически.

Постановка задачи

Дано: функция многих переменных $y = f(x_1, x_2, \dots, x_n)$ и точка X^0;

Задание:

- Найти минимум функции двух переменных с точностью $\varepsilon = 0.001$, начиная итерации из точки X^0;
- Найти минимум аналитичности;
- Сравнить полученные результаты.

Индивидуальный вариант:

$f(x) = x^2 + y^2 + x + 0.3arctg(xy),\ X^0 = (0{,}0).$

2. Основные теоретические сведения

Метод наискорейшего спуска является итерационным. Пусть для заданной функции $f(x_1, x_2, \dots, x_n)$ на -том шаге имеется некоторое приближение к минимуму $X^k = \left(x_1^k, x_2^k, \dots, x_n^k\right)$.

Рассмотрим функцию одной переменной $\varphi_k(t)$:

$$\varphi_k(t) = f\left(X^k - t * grad\, f(X^k)\right),$$

где вектор $grad\, f(X^k) = \left(\frac{\partial f}{\partial x_1}(X^k), \dots, \frac{\partial f}{\partial x_n}(X^k)\right)$ – градиент функции f в точке X^k.

Функция $\varphi_k(t)$ представляет собой ограничение функции f на прямую градиентного спуска, проходящую через точку -го приближения X^k.

Для следующего приближения к точке минимума полагаем

$$X^{k+1} = X^k - t^* * grad\, f(X^k),$$

где точка t^*– это минимум функции $\varphi_k(t)$.

Процесс поиска минимума продолжается до тех пор, пока $\|grad\ f(X^k)\| = \max\limits_{1 \le i \le n} \left| \frac{\partial f}{\partial x_i}(X^k) \right|$ не станет меньше допустимой погрешности ε.

В двумерном случае итерация имеет следующий вид:

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

где $t^* = -\frac{\varphi_k'(0)}{\varphi_k''(0)}$;

$$\varphi_k'(0) = -\left(\frac{\partial f}{\partial x}\right)^2 - \left(\frac{\partial f}{\partial y}\right)^2,$$

$$\varphi_k''(0) = \frac{\partial f^2}{\partial x^2}\left(\frac{\partial f}{\partial x}\right)^2 + 2\frac{\partial f^2}{\partial x \partial y}\frac{\partial f}{\partial x}\frac{\partial f}{\partial y} + \frac{\partial f^2}{\partial y^2}\left(\frac{\partial f}{\partial y}\right)^2,$$

где все производные берутся в точке (x_k, y_k).

3. Реализация

Листинг 1. Метод наискорейшего спуска поиска минимума функции многих переменных

```
using System;

class Program
{
    static void Main()
    {
        double eps = 0.001;
        Tuple<double, double> analyticalMin = AnalyticalMin();
        Console.WriteLine("Аналитический минимум: ({0}, {1})",
analyticalMin.Item1, analyticalMin.Item2);

        int k = 0;
        double xk = 0.0, yk = 0.0;

        while (Math.Max(DerivativeX(xk, yk), DerivativeY(xk, yk)) >= eps)
        {
            double phi1 = -Math.Pow(DerivativeX(xk, yk), 2) -
Math.Pow(DerivativeY(xk, yk), 2);
            double phi2 = Derivative2X(xk, yk) * Math.Pow(DerivativeX(xk, yk),
2) +
                                   2 * DerivativeXY(xk, yk) * DerivativeX(xk, yk) *
DerivativeY(xk, yk) +
                                   Derivative2Y(xk, yk) * Math.Pow(DerivativeY(xk, yk),
2);
            double tStar = -phi1 / phi2;

            xk = xk - tStar * DerivativeX(xk, yk);
            yk = yk - tStar * DerivativeY(xk, yk);
            k++;
```

```
        }

        Console.WriteLine($"Метод наискорейшего спуска: ({xk}, {yk})");
        Console.WriteLine($"Абсолютная разница ({Math.Abs(analyticalMin.Item1 -
xk)}, {Math.Abs(analyticalMin.Item2 - yk)})");
    }

    static double Function(double x, double y)
    {
        return (x*x)+2*(y*y)+2*x+0.3*Math.Atan(x*y);
    }

    static double DerivativeX(double x, double y)
    {
        double h = 1e-5;
        return (Function(x + h, y) - Function(x, y)) / h;
    }

    static double DerivativeY(double x, double y)
    {
        double h = 1e-5;
        return (Function(x, y + h) - Function(x, y)) / h;
    }

    static double Derivative2X(double x, double y)
    {
        double h = 1e-5;
        return (DerivativeX(x + h, y) - DerivativeX(x, y)) / h;
    }

    static double Derivative2Y(double x, double y)
    {
        double h = 1e-5;
        return (DerivativeY(x, y + h) - DerivativeY(x, y)) / h;
    }

    static double DerivativeXY(double x, double y)
    {
        double h = 1e-5;
        return (DerivativeX(x, y + h) - DerivativeX(x, y)) / h;
    }

    static Tuple<double, double> AnalyticalMin()
    {
        return Tuple.Create(-1.01124535766275, 0.0754049591595788);
    }
}
```

4. Результаты

```
Консоль отладки Microsoft Visual Studio
Аналитический минимум: (-1,01124535766275, 0,0754049591595788)
Метод наискорейшего спуска: (-1,011033466267496, 0,07537433841275594)
Абсолютная разница (0,0002118913952540158, 3,062074682286253E-05)
```

5. Вывод

В процессе выполнения лабораторной работы был изучен и использован метод наискорейшего спуска для нахождения приближенного значения минимума функции с двумя переменными, а также аналитически определён её минимум. В результате тестирования данного метода на конкретной функции выяснилось, что уже после 4 итераций вычислительная погрешность составила около 0.0001.