|  |  |
| --- | --- |
|  | **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 и точка ;

**Задание:**

* Найти минимум функции двух переменных с точностью , начиная итерации из точки ;
* Найти минимум аналитичности;
* Сравнить полученные результаты.

**Индивидуальный вариант:**

, .

1. **Основные теоретические сведения**

Метод наискорейшего спуска является итерационным.  
Пусть для заданной функции на -том шаге имеется некоторое приближение к минимуму .

Рассмотрим функцию одной переменной :

где вектор – градиент функции в точке .

Функция представляет собой ограничение функции на прямую градиентного спуска, проходящую через точку -го приближения .

Для следующего приближения к точке минимума полагаем

где точка – это минимум функции .

Процесс поиска минимума продолжается до тех пор, пока не станет меньше допустимой погрешности ε.

В двумерном случае итерация имеет следующий вид:

где ;

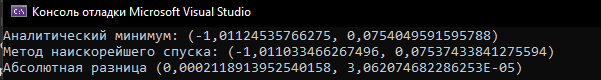
где все производные берутся в точке .

1. **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);  }  } |

1. **Результаты**

****

1. **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.