|  |  |
| --- | --- |
|  | **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 ИУ9-62Б

Марченко Андрей

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 ;

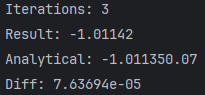
где все производные берутся в точке .

1. **Реализация**

Листинг 1. Метод наискорейшего спуска поиска минимума функции многих переменных

#include <iostream>  
#include <cmath>  
  
  
double f(double x, double y) {  
 return x \* x + 2 \* y \* y + 2 \* x + 0.3 \* std::atanh(x \* y);  
}  
  
double df\_dx(double x, double y) {  
 return 2 \* x + 2 + (3 \* y) / (10 + 10 \* x \* x \* y \* y);  
}  
  
double df\_dy(double x, double y) {  
 return 4 \* y + (3 \* x) / (10 + 10 \* x \* x \* y \* y);  
}  
  
double df\_dxdx(double x, double y) {  
 return 2 - (60 \* x \* y \* y \* y) / ((10 + 10 \* x \* x \* y \* y) \* (10 + 10 \* x \* x \* y \* y));  
}  
  
double df\_dxdy(double x, double y) {  
 return (30 - 30 \* x \* x \* y \* y) / ((10 + 10 \* x \* x \* y \* y) \* (10 + 10 \* x \* x \* y \* y));  
}  
  
double df\_dydy(double x, double y) {  
 return 4 - (60 \* x \* x \* x \* y) / ((10 + 10 \* x \* x \* y \* y) \* (10 + 10 \* x \* x \* y \* y));  
}  
  
double fi\_first(double x, double y) {  
 return - std::pow(df\_dx(x, y), 2) - std::pow(df\_dy(x,y), 2);  
}  
  
double fi\_second(double x, double y) {  
 return df\_dxdx(x, y) \* std::pow(df\_dx(x, y), 2) +  
 2 \* df\_dxdy(x, y) \* df\_dx(x, y) \* df\_dy(x, y) +  
 df\_dydy(x, y) \* std::pow(df\_dy(x, y), 2);  
}  
  
int main() {  
 double eps = 0.001 ;  
 double x\_k = -1, y\_k = 0; // начальные значения  
 double x\_check = -1.011, y\_check = 0.07;  
 double fi1, fi2, t, error = 100;  
 int n = 0;  
 while (error > eps) {  
 n++;  
 fi1 = fi\_first(x\_k, y\_k);  
 fi2 = fi\_second(x\_k, y\_k);  
 t = - fi1 / fi2;  
 x\_k -= t \* df\_dx(x\_k, y\_k);  
 y\_k -= t \* df\_dy(x\_k, y\_k);  
 error = std::max(std::abs(df\_dx(x\_k, y\_k)), std::abs(df\_dy(x\_k, y\_k)));  
 std::cout << x\_k << " " << y\_k << " " << error << std::endl;  
 }  
 std::cout << "Iterations: " << n << std::endl;  
 std::cout << "Result: " << f(x\_k, y\_k) << std::endl;  
 std::cout << "Analytical: " << f(x\_check, y\_check) << y\_check << std::endl;  
 std::cout << "Diff: " << std::abs(f(x\_k, y\_k) - f(x\_check, y\_check));  
 return 0;  
}

1. **Результаты**

****

1. **Вывод**

В ходе выполнения лабораторной работы был изучен и реализован метод наискорейшего спуска, получено приближенное значение минимума функции двух переменных и был найден ее минимум аналитичности. В результате тестирования для приведенной функции после 3 итераций вычислительная погрешность составила около 0.00007.