МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики Кафедра прикладной математики**

**ОТЧЕТ**

**по дисциплине**

**«Методы оптимизации»**

Работу выполнил М. А. Пузырёв

Работу принял преподаватель Е. С. Троценко

Краснодар

2024

1. **Постановка задачи**

Требуется найти безусловный минимум функции двух переменных  методом наискорейшего градиентного спуска при начальных значениях , т.е. такие точки , что .

1. **Стратегия поиска**

Стратегия решения задачи состоит в построении последовательности точек {}, k = , таких, что , k = . Точки последовательности {} вычисляются по правилу

(1)

где точка задается пользователем; величина шага определяется для каждого значения k из условия

(2)

Решение задачи (2) может осуществляться с использованием необходимого условия минимума с последующей проверкой достаточного условия минимума . Такой путь может быть использован либо при достаточно простой минимизируемой функции , либо при предварительной аппроксимации достаточно сложной функции полиномом (как правило, второй или третьей степени), и тогда условие замещается условием , а условие условием .

Другой путь решения задачи (2) связан с использованием численных методов, когда ищется . Границы интервала задаются пользователем. При этом степень близости найденного значения к оптимальному значению , удовлетворяющему условиям зависит от задания интервала и точности методов одномерной минимизации.

Построение последовательности {}, k = , заканчивается в точке , для которой , где - заданное число, или, если , - предельное число итераций, или при двукратном одновременном выполнении неравенств , где - малое положительное число. Вопрос о том, может ли точка рассматриваться как найденное приближение искомой точки локального минимума , решается путем дополнительного исследования.

1. **Алгоритм**

Шаг l. Задать предельное число итераций . Найти градиент функции в произвольной точке

Шаг 2. Положить

Шаг 3. Вычислить.

Шаг 4. Проверить выполнение критерия окончания :

а) если критерий выполнен, то ;

б) если критерий не выполнен, то перейти к шагу 5.

Шаг 5. Проверить выполнение неравенства :

а) если неравенство выполнено, то ;

б) если нет, то перейти к шагу 6.

Шаг 6. Вычислить величину шага из условия

Шаг 7. Вычислить .

Шаг 8. Проверить выполнение условий

а) если оба условия выполнены при текущем значении и k = , то расчет окончен, ;

б) если хотя бы одно из условий не выполнено, то положить и перейти к шагу 3.

**4 Код программы**

Реализация кода представлена на языке C++:

#include <iostream>

#include <cmath>

#include <math.h>

double F(double x, double y) {

return pow(x, 2) + 4 \* pow(y, 2) + x \* y + x;

}

double Fx(double x, double y) {

return 2 \* x + y + 1;

}

double Fy(double x, double y) {

return x + 8 \* y;

}

double L(double x, double y) {

double l = pow(x, 2) + pow(y, 2);

return sqrt(l);

}

double G(double x, double y, double t) { // функция f(t)

return F(x - t \* Fx(x, y), y - t \* Fy(x, y));

}

double Zol(double x, double y) { // метод нахождения минимума функции f(t)

double a = -1;

double b = 1;

double k = 0.38196;

double l = 0.01;

double t1 = a + k \* (b - a);

double t2 = a + b - t1;

do {

if (G(x, y, t1) <= G(x, y, t2)) {

b = t2;

t2 = t1;

t1 = a + b - t2;

}

else {

a = t1;

t1 = t2;

t2 = a + b - t1;

}

} while (fabs(b - a) > l);

return (b + a) / 2;

}

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int m = 50, it = 0;

double eps1 = 0.1, eps2 = 0.15;

double x, y, x0, y0, t;

x0 = 3, y0 = 1;

for (int i = 0; i < m; i++) {

if (L(Fx(x0,y0), Fy(x0,y0)) < eps1)

break;

t = Zol(x0, y0);

x = x0 - t \* Fx(x0,y0);

y = y0 - t \* Fy(x0,y0);

if (L(x - x0, y - y0) < eps2 && fabs(F(x, y) - F(x0, y0)) < eps2)

break;

x0 = x, y0 = y;

it++;

}

cout << "Алгоритм завершился за " << it << " итераций.\n";

cout << "Точка минимума X = (" << x0 << "; " << y0 << ")\n";

cout << "Значение функции в минимуме: F = " << F(x0, y0) << endl;

return 0;

}

**5 Сходимость**

Оценки скорости сходимости получены только для сильно выпуклых функций, когда последовательность {} сходится к точке минимума функции со скоростью геометрической прогрессии (линейная сходимость):

где и - оценки наибольшего и наименьшего собственных значений матрицы функции .

**6 Вывод**

Для решения задачи по поиску безусловного минимума функции методом наискорейшего градиентного спуска была написана программа на языке C++. Результатом работы программы является вычисленное минимальное значение функции, равное -0,255038 в точке минимума X = (-0,456716; 0,0962219). Сходимость алгоритма оказалась в пределах отрезка [0,000133622; 0,4] (рис.1). При изменении параметра и , точка минимума, значение функции в ней и сходимость меняются. При уменьшении и число итераций, необходимых для нахождения безусловного минимума увеличивается, так, к примеру, для = 0,005 и = 0,01 число итераций увеличилось с 6 до 8, при этом значение функции стало равно -0,266321, в точке X = (-0,523471; 0,0734064). Сходимость алгоритма оказалась в пределах отрезка [2,76e-08; 0,4ы] (рис.2).

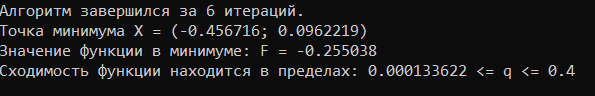


Рисунок 1 – Результат работы программы

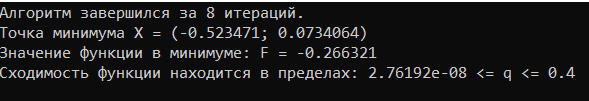


Рисунок 2 – Результат работы программы