**МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

**Кафедра программного обеспечения вычислительной техники и автоматизированных систем**

Лабораторная работа №7

Дисциплина: Вычислительная математика

по теме «Минимизация функции многих переменных методом градиента с дроблением шага»

Выполнил: ст. группы ВТ-22  
Макаров Даниил Сергеевич

Проверила: Бондаренко Т.В.

**Белгород 2018**

**Лабораторная работа № 7**

**Минимизация функции многих переменных методом градиента с дроблением шага**

Цель работы: изучить метод градиента с дроблением шага для решения задачи минимизации функции многих переменных и получить практические навыки его применения.

Задания к работе

1. Найти точные значения координат точки минимума и минимальное значение функции у = f (x1, х2) для функции соответствующего варианта задания, используя необходимые и достаточные условия локального минимума.

2. Выполнить вручную вычисление приближенного значения точки минимума целевой функции методом градиента с дроблением шага, начиная с произвольно выбранного начального приближения М0 (x10, x20). Точность решения ε = 0,01. Вручную подробно достаточно выполнить первый шаг метода градиента с дроблением шага. Параметры метода выбрать самостоятельно.

3. Реализовать логическую функцию для нахождения приближенного значения точки локального минимума и минимального значения целевой функции у = f(x1, х2) методом градиента с дроблением шага. Входными данными для логической функции являются:

* целевая функция у = f (x1, х2); 54
* градиент целевой функции grad (f (x1, х2));
* начальное приближение М0(x10, x20) к точке локального минимума;
* точность решения ɛ;
* ограничение на максимальное число итераций n;
* параметры метода градиента с дроблением шага α, β, γ.

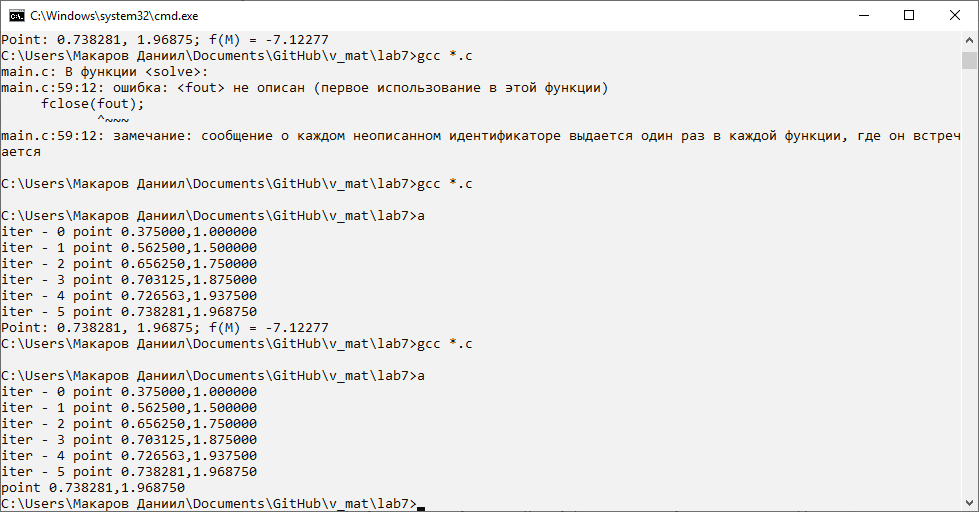
Функция возвращает значение «истина», если приближенное решение с заданной точностью получено за число итераций, не превышающее n, и «ложь» в противном случае.

Результат работы программы: приближенное значение точки локального минимума и минимальное значение целевой функции у = f (x1, х2) с заданной точностью, количество выполненных итераций. Предусмотреть возможность сохранения пошаговых результатов реализации метода градиента с дроблением шага в файл.

**Задание варианта 30**

***Решим систему уравнений.***

– начальное приближение



#include <stdio.h>

#include <math.h>

typedef double (\*func)(double, double);

typedef struct {

func der\_x1;

func der\_x2;

}grad;

typedef struct {

double x1;

double x2;

}point;

double f(double x1, double x2){

return 2\*x1\*x1-3\*x1+2\*x2\*x2-8\*x2+2;}

double der\_x1(double x1, double x2){

return 4\*x1-3;}

double der\_x2(double x1, double x2){

return 4\*x2-8;}

int solve(func f, grad g, point M0, double epsilon, int max\_iterations,

double alpha, double beta, double gamma, point \*minimum, double \*min\_val){

point M1;

double f\_M0, f\_M1;

int k = 0;

double x\_1, x\_2, r, t;

while (k<=max\_iterations){

f\_M0 = f(M0.x1, M0.x2);

int i = 0;

do{

x\_1 = M0.x1-alpha\*pow(gamma, i)\*g.der\_x1(M0.x1, M0.x2);

x\_2 = M0.x2-alpha\*pow(gamma, i)\*g.der\_x2(M0.x1, M0.x2);

r = f(x\_1, x\_2) - f\_M0;

x\_1 = g.der\_x1(M0.x1, M0.x2);

x\_2 = g.der\_x2(M0.x1, M0.x2);

t = pow(x\_1, 2) + pow(x\_2, 2);

t = -beta\*alpha\*pow(gamma, i)\*t;

i++;

}while (!(r<t));

i = i-1;

M1.x1 = M0.x1-alpha\*pow(gamma, i)\*g.der\_x1(M0.x1, M0.x2);

M1.x2 = M0.x2-alpha\*pow(gamma, i)\*g.der\_x2(M0.x1, M0.x2);

f\_M1 = f(M1.x1, M1.x2);

printf("iter - %d point %f,%f\n",k, M1.x1, M1.x2);

if (fabs(f\_M0-f\_M1)<epsilon){

minimum->x1 = M1.x1;

minimum->x2 = M1.x2;

\*min\_val = f\_M1;

return 1; }

M0.x1 = M1.x1;

M0.x2 = M1.x2;

k++; }

return 0;

}

int main(){

const grad g = {der\_x1, der\_x2};

const point M0 = {0,0};

const double epsilon = 0.01;

const int n = 500;

const double alpha = 1, beta = 0.5, gamma = 0.5;

point p\_min;

double f\_min;

if (solve(f, g, M0, epsilon, n, alpha, beta, gamma, &p\_min, &f\_min)){

printf("point %f,%f", p\_min.x1, p\_min.x2);

}else{

printf("can't resolve");}

return 0;

}