МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧЕРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г. Шухова»

(БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Дисциплина: вычислительная математика

Лабораторная работа №7

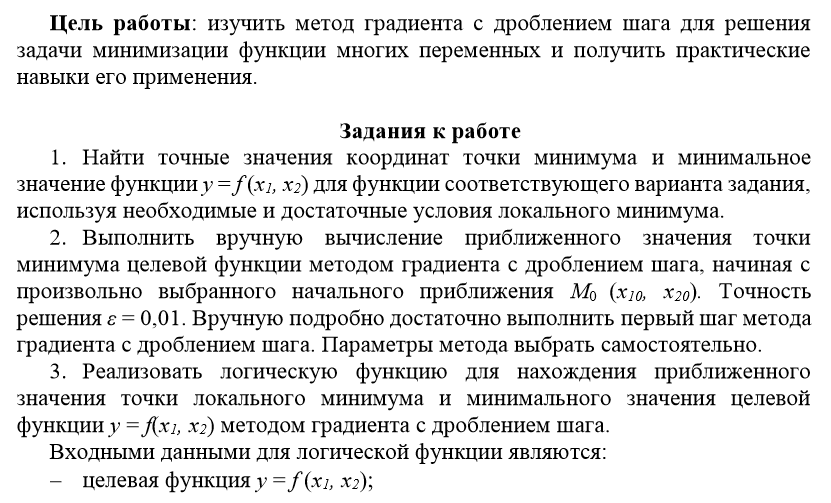
Тема: «Минимизация функции многих переменных методом градиента с дроблением шага»

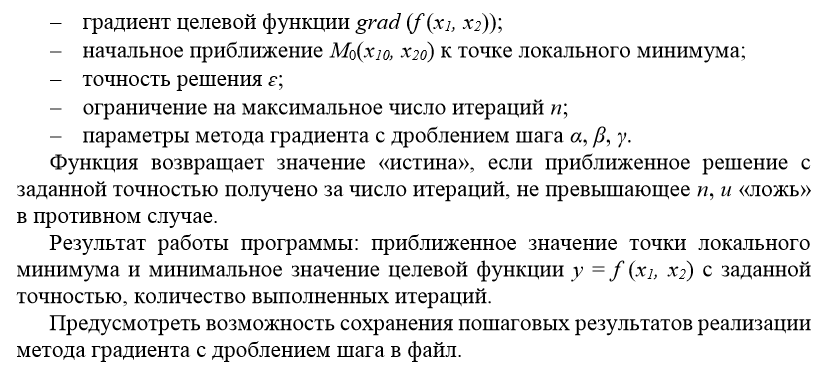
Выполнил: Ст. группы ПВ-21

Донцов Александр Алексеевич

Проверил: Бондаренко Т.В.

Белгород 2018 г.





**Вариант 1**



**Заголовочный файл модуля GRAD\_METOD\_H\_** **с функцией для решения задачи методом градиента с дроблением шага.**

#pragma once

#ifndef GRAD\_METOD\_H\_

#define GRAD\_METOD\_H\_

//целевая функция у = f (x1, х2)

double f\_x1x2(double, double);

//градиент целевой функции grad (f (x1, х2))

void find\_grad(double, double, double[]);

// Метод градиент с дроблением шага

int metod\_grad\_whith\_drob(double \*x1, double \*x2, double exp, int n, double alf, double bet, double gam);

#endif

**Исходный файл модуля GRAD\_METOD\_H\_**

#include "grad\_metod.h";#include "pch.h"

#include "stdio.h"

#include "math.h"

#define N 2//целевая функция у = f (x1, х2)

double f\_x1x2(double x1, double x2) {

double znach = 2 \* x1 \* x1 \* x1 \* x1 + 4 \* x2 \* x2 \* x2 \* x2 - 3 \* x1 \* x2 + 2;

return znach;

}

//градиент целевой функции grad (f (x1, х2)

void find\_grad(double x1, double x2, double a[]) {

a[0] = 8 \* x1 \* x1 \* x1 - 3 \* x2;

a[1] = 16 \* x2 \* x2 \* x2 - 3 \* x1;

}

// Метод градиента с дроблением

int metod\_grad\_whith\_drob(double \*x1, double \*x2, double exp, int n, double alf, double bet, double gam) {

double x1\_0 = \*x1, x2\_0 = \*x2;

double grad[N];

double left, right;

double f\_0;

int i;

for (int j = 1; j <= n; j++) {

i = 0;

f\_0 = f\_x1x2(x1\_0, x2\_0);

find\_grad(x1\_0, x2\_0, grad);

left = f\_x1x2((x1\_0 - alf \* pow(gam, i) \* grad[0]), (x2\_0 - alf \* pow(gam, i) \* grad[1])) - f\_0;

right = (-1) \* bet \* alf \* pow(gam, i) \* (pow(grad[0], 2) + pow(grad[1], 2));

while (left >= right) {

i++;

left = f\_x1x2((x1\_0 - alf \* pow(gam, i) \* grad[0]), (x2\_0 - alf \* pow(gam, i) \* grad[1])) - f\_0;

right = (-bet) \* alf \* pow(gam, i) \* (pow(grad[0], 2) + pow(grad[1], 2));

}

x1\_0 = x1\_0 - alf \* pow(gam, i) \* grad[0];

x2\_0 = x2\_0 - alf \* pow(gam, i) \* grad[1];

if ((fabs(f\_0 - f\_x1x2(x1\_0, x2\_0))) < exp) {

\*x1 = x1\_0;

\*x2 = x2\_0;

return 1;

}

}

return 0;

}

**Код программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include "windows.h"

#include "grad\_metod.h"

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n;

double exp;

double x1, x2;

double znach;

double alf, bet, gam;

printf("Введите требуюмую точность решения E: ");

scanf("%lf", &exp);

printf("Введите начальное приближение M0(x1\_0,x2\_0) к точке локального минимума: ");

scanf("%lf %lf", &x1, &x2);

printf("\nВведите параметры метода градиента с дроблением шага alf, bet, gam: ");

scanf("%lf %lf %lf", &alf, &bet, &gam);

printf("\nВведите максимальное колличество итераций для проверки: ");

scanf("%d", &n);

if (metod\_grad\_whith\_drob(&x1, &x2, exp, n, alf, bet, gam)) {

znach = f\_x1x2(x1, x2);

printf("\nВычисление приближенного значения точки локального минимума и минимального значения целевой функции методом градиента с дроблением шага: Мт (%lf ; %lf) со значением в ней: y(Mт) = %lf\n\n", x1, x2, znach);

}

else

printf("\nЗа данное количество операций задача решение с точностью E не найдено!");

return 0;

}

