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 №6

Тема: «Одномерная минимизация функции»

Выполнил: Ст. группы ПВ-21

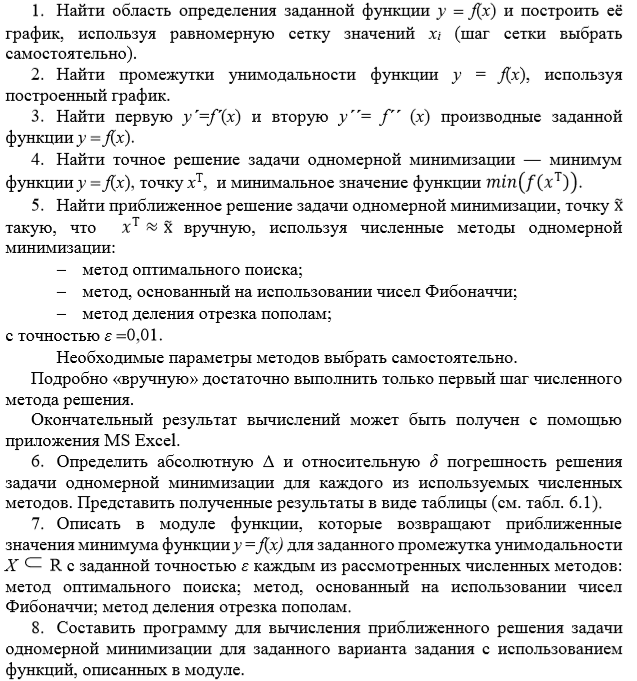
Донцов Александр Алексеевич

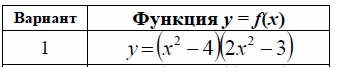
Проверил: Бондаренко Т.В.

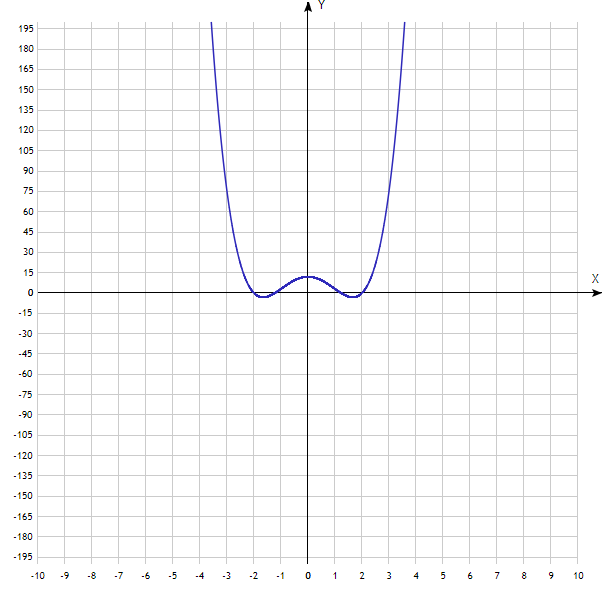
Белгород 2018 г.

**Цель работы:** изучить методы нахождения приближенного решения задачи одномерной минимизации функции одной переменной, и получить практические навыки их применения.

**Задания к работе**







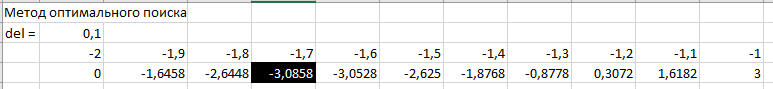
3.

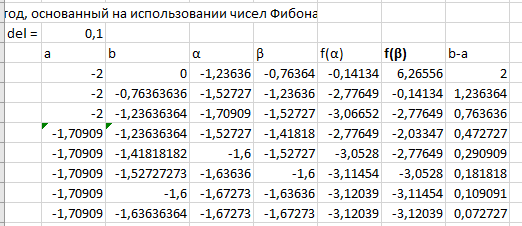
4.

x=-1,658, x=1,658

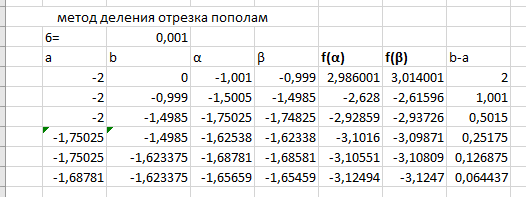
y = -3,125

5.





-3,12



-3,125

7. Описать в модуле функции, которые возвращают приближенные значения минимума функции у = f(x) для заданного промежутка унимодальности X ⊂ R с заданной точностью ε каждым из рассмотренных численных методов: метод оптимального поиска; метод, основанный на использовании чисел Фибоначчи; метод деления отрезка пополам.

**Заголовочный файл minimal\_function\_mod.h**

#pragma once

#ifndef MINIMAL\_FUNCTION\_MOD\_H\_

#define MINIMAL\_FUNCTION\_MOD\_H\_

double f\_x(double); // Вычисление значения функции в точке x

double optimal\_find(double, double, double); // Оптимальный поиск

double cat\_in\_half(double, double, double); // Поиск путем деления пополам

double metod\_fibanachi(double, double, double); // Поиск с использованием чисел Фибаначи

**Файл с модулем minimal\_function\_mod.cpp**

#include "minimal\_function\_mod.h";

#include "stdio.h"

#include "math.h"

double f\_x(double x) {

return ((x\*x - 4)\*(2\*x\*x - 3));

}

double optimal\_find(double exp, double a, double b) {

double min = f\_x(a);

double znach;

for (double count = a + exp; count <= b; count = count + exp) {

znach = f\_x(count);

if (znach < min)

min = znach;

}

return min;

}

double cat\_in\_half(double exp, double a, double b) {

double alf, bet;

while ((b - a) < exp) {

alf = ((a + b) / 2) - 0.001;

bet = ((a + b) / 2) + 0.001;

if (f\_x(alf) >= f\_x(bet))

a = alf;

else

b = bet;

}

return f\_x(((a + b) / 2));

}

double metod\_fibanachi(double exp, double a, double b) {

int fib[12] = { 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 };

double alf, bet;

int i = 0;

while (((b - a) < exp) && (i <= 11)) {

alf = a + (fib[(11 - i - 1)] \* (b - a) / fib[(11 - i + 1)]);

bet = a + (fib[(11 - i)] \* (b - a) / fib[(11 - i + 1)]);

if (f\_x(alf) >= f\_x(bet))

a = alf;

else

b = bet;

i++;

}

return f\_x(((a + b) / 2));

}

8. Составить программу для вычисления приближенного решения задачи одномерной минимизации для заданного варианта задания с использованием функций, описанных в модуле.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include "windows.h"

#include "od\_minimization.h"

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double exp;

double a, b;

double znach;

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

scanf("%lf", &exp);

printf("Введите левую границу промежутка унимодальности : ");

scanf("%lf", &a);

printf("Введите правую границу промежутка унимодальности : ");

scanf("%lf", &b);

printf("\nМинимальное значение функции на данном отрезке с заданной точностью\n");

znach = optimal\_find(exp, a, b);

printf("\nМетодом оптимального поиска : %lf\n", znach);

znach = cat\_in\_half(exp, a, b);

printf("\nМетодом оптимального поиска : %lf\n", znach);

znach = metod\_fibanachi(exp, a, b);

printf("\nМетодом оптимального поиска : %lf\n", znach);

getchar();

getchar();

getchar();

return 0;

}

