**МИНОБРНАУКИ РОССИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕНЫЙ 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:

Студент группы ВТ-22

Воскобойников Илья Сергеевич

Проверила: Бондаренко Т. В.

Белгород 2019

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

Задания к работе

1. Найти область определения заданной функции у = f(x) и построить её график, используя равномерную сетку значений хi (шаг сетки выбрать самостоятельно).

2. Найти промежутки унимодальности функции у = f(x), используя построенный график.

3. Найти первую y´=f´(x) и вторую y´´= f´´ (x) производные заданной функции у = f(x).

4. Найти точное решение задачи одномерной минимизации ― минимум функции у = f(x), точку х Т , и минимальное значение функции 𝑚𝑖𝑛(𝑓(𝑥 Т )).

5. Найти приближенное решение задачи одномерной минимизации, точку х̃ такую, что 𝑥 Т ≈ х̃ вручную, используя численные методы одномерной минимизации:

− метод оптимального поиска;

− метод, основанный на использовании чисел Фибоначчи;

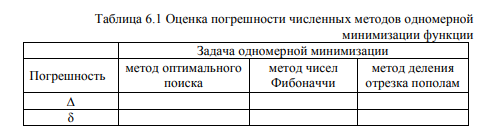
− метод деления отрезка пополам; с точностью ε =0,01.

Необходимые параметры методов выбрать самостоятельно. Подробно «вручную» достаточно выполнить только первый шаг численного метода решения. Окончательный результат вычислений может быть получен с помощью приложения MS Excel.

6. Определить абсолютную Δ и относительную δ погрешность решения задачи одномерной минимизации для каждого из используемых численных методов. Представить полученные результаты в виде таблицы (см. табл. 6.1).

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.

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



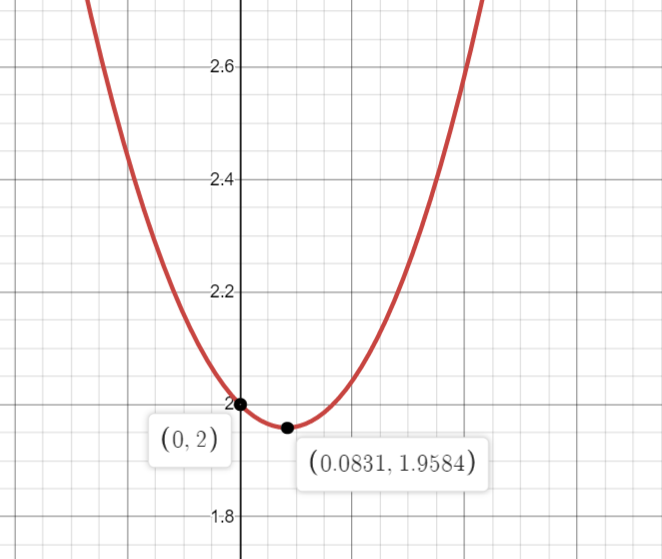
Вариант 4



Задание 1-3

y = x^4+6x^2-x+2

Обл. опр. D(y) = [ -∞ ; +∞)



функция убывает (-∞ ;0.0831)

функция возрастает (0.0831; +∞)

y` = 4x^3+12x-1

y`` = 12x^2+12

Задание 4

y` = 0

4x^3+12x-1 = 0

x1 = 0.08314

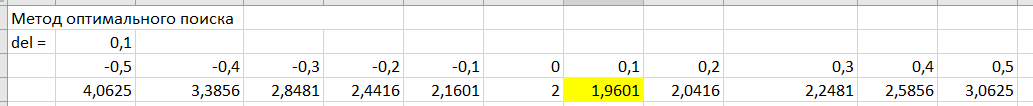
x2,x3 - комплексные корни

xT=0.08314

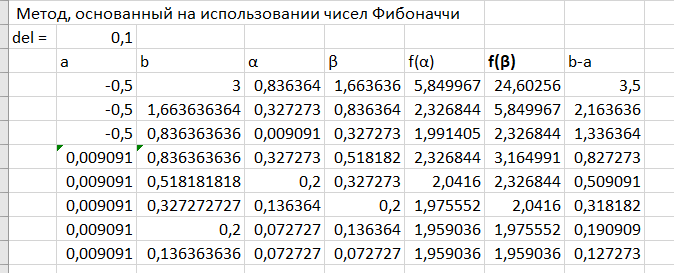
min(f(xT)) = 0.08314^4+6\*0.08314^2-0.08314+2= 1,95838

Задание 5

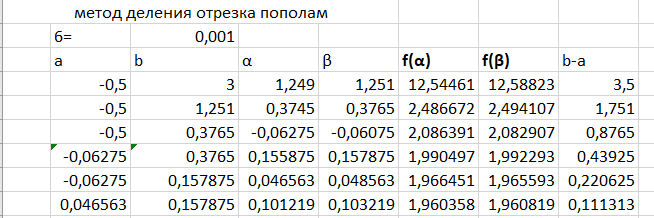
Метод оптимального поиска



Метод, основанный на использовании чисел Фибоначи.



Метод деления отрезка пополам



-1.9603

Задание 7

Заголовочный файл модуля

#ifndef **VYCH\_6\_MIN\_F\_H**#define **VYCH\_6\_MIN\_F\_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*#endif *//VYCH\_6\_MIN\_F\_H*

Модуль

#include **"min\_f.h"**#include **"stdio.h"**#include **"math.h"  
double** f\_x(**double** x) {  
 **return** (powl(x,4)+6\*powl(x,2)-x+2);  
}  
**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) {  
 **float** q = exp/2, c, a1, b1;  
 **while** (fabs(b - a - exp) > exp){  
 c = (b + a)/2;  
 a1 = c - q;  
 b1 = c + q;  
 **if** (f\_x(a1) <= f\_x(b1))  
 b = b1;  
 **else** a = a1;  
 }  
 **return** f\_x((a+b)/2);  
  
}  
**double** metod\_fibanachi(**double** exp, **double** a, **double** b) {  
 **int** F[100];  
 F[0] = 1; F[1] = 1;  
 **int** i = 2, n = 1;  
 **while** (F[i-2] + F[i-1] < (b-a)/exp){  
 F[i] = F[i-2] + F[i-1];  
 i++;  
 n++;  
 }  
 **float** a1,b1;  
 **for** (**int** j = 1; j < n; j++){  
 a1 = a + (**float**)F[n - j - 1]/F[n - j + 1]\*(b - a);  
 b1 = a + (**float**)F[n - j]/F[n - j + 1]\*(b - a);  
 **if** (f\_x(a1) <= f\_x(b1))  
 b = b1;  
 **else** a = a1;  
 }  
 **return** f\_x(a);

}

Основная программа

#include **<stdio.h>**#include **"min\_f.h"  
  
int** main() {  
 **double** exp;  
 **double** a, b;  
 **double** znach;  
  
 printf(**"Введите требуюмую точность решения E: "**);  
 scanf(**"%lf"**, &exp);  
  
 printf(**"\nВведите левую границу промежутка унимодальности : "**);  
 scanf(**"%lf"**, &a);  
 printf(**"\nВведите левую 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);  
  
  
 **return** 0;  
}

Задание 6

