**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»

Выполнила:

Студент группы ВТ-21

Сидорова Ангелина Сергеевна

Проверила: Бондаренко Т. В.

Белгород 2018

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;

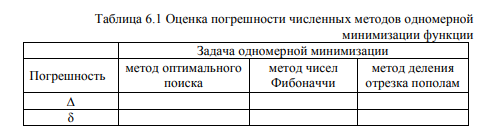
− метод деления отрезка пополам; с точностью ε =0,01.

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

Безымянный.png

Задание 1-3

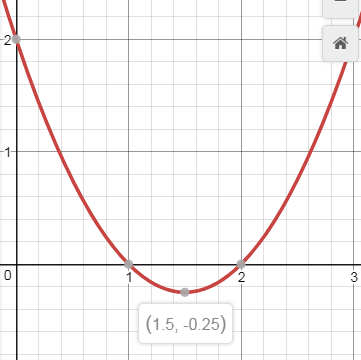
y = (x-1)(x-2)

y = x2- 2x – x + 2

y = x2 – 3x + 2

x ϵ [ -2 ; 2 ]

Обл. опр. D(y) = [ (-1/4) ; +∞)



y` = 2x – 3

y`` = 2

Задание 4

y` = 2x -3 = 0

2x – 3 = 0

2x = 3

x =3/2

xT=3/2

min(f(xT)) = (3/2)2 – 3\*3/2 + 2 = 9/4 – 9/2 + 2 = -1/4

Задание 5

Метод оптимального поиска

Безымянный.png

Безымянный.png

Безымянный.png

Безымянный.png

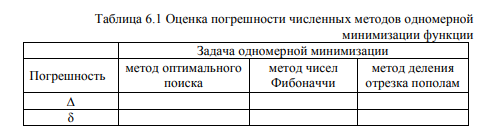
Метод, основанный на использовании чисел Фибоначи.

Безымянный.png

Метод деления отрезка пополам

Безымянный.png

Задание 6



Задание 7

#include <stdio.h>

#include <stdlib.h>

#include "main.h"

#include <windows.h>

float f(float x){

return(a\*a – 3\*a+ 2);

}

int main(){

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

float min = min\_optimal\_poisk(f,-0.5,1,5,0.01);

printf("оптимальный поиск \t\t\t\t\tх = %f\n метод, основанный на использовании чисел Фибоначчи \tх = %f\n метод деления отрезка пополам \t\t\t\t х = %f\n",min);

return 0;}

main.h

#ifndef MAIN\_H\_INCLUDED

#define MAIN\_H\_INCLUDED

float min\_optimal\_poisk(float (\*f)(float x), float a, float b, float e);

float min\_dif\_2(float (\*f)(float x), float a, float b, float e);

float min\_fib\_chisla(float (\*f)(float x), float a, float b, float e);

#endif // MAIN\_H\_INCLUDED

min.c

#include <stdio.h>

#include "main.h"

void getmem(float \*\*\*a, int n){

(\*a) = calloc(n + 1, sizeof(float\*));

for (int i = 0; i <= n; i++)

(\*a)[i] = calloc(2, sizeof(float));

}

float min\_optimal\_poisk(float (\*f)(float x), float a, float b, float e){

float h = e, x = a, \*\*y;

int n = (int)((b - a)/e + 0.5);

getmem(&y,n);

y[0][0] = a; y[0][1] = f(a);

y[n][0] = b; y[n][1] = f(b);

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

y[i][0] = y[i-1][0] + h;

y[i][1] = f(y[i][0]);

}

int min = 0;

for (int i = 1; i <= n; i++)

if (y[i][1] < y[min][1])

min = i;

return y[min][0];

}

float min\_dif\_2(float (\*f)(float x), float a, float b, float e){

float q = e/2, c, a1, b1;

while (fabs(b - a - e) > e/10){

c = (b + a)/2;

a1 = c - q;

b1 = c + q;

if (f(a1) <= f(b1))

b = b1;

else

a = a1;

}

return ((a+b)/2);

}

float min\_fib\_chisla(float (\*f)(float x), float a, float b, float e){

int \*F = malloc(2\*sizeof(int));

F[0] = 1; F[1] = 1;

int i = 2, n = 1;

while (F[i-2] + F[i-1] < (b-a)/e){

realloc(\*F,(i+1)\*sizeof(int));

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(a1) <= f(b1))

b = b1;

else

a = a1;

}

return (a);}