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

Дисциплина: Вычислительная математика

Лабораторная работа № 10

Тема: «Метод наименьших квадратов»

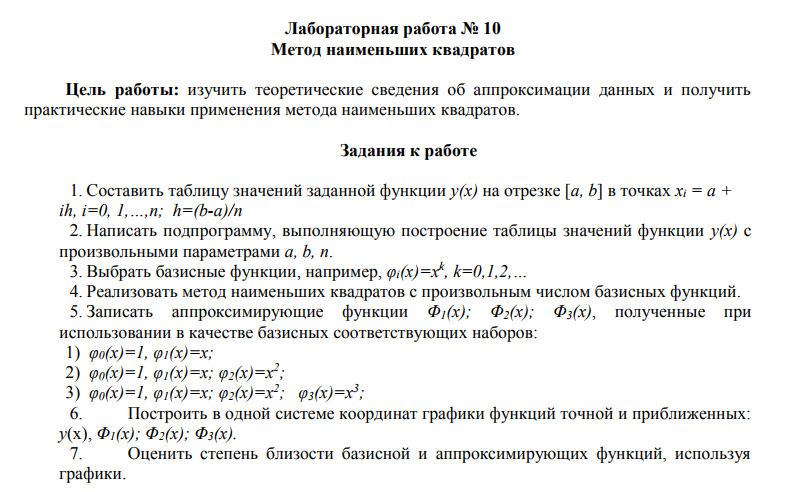
Выполнил:

студент группы ПВ–21

Черкасова Анна Родионовна

Проверил: Бондаренко Т. В.

Белгород 2017



**Вариант: y = 3х cos(x – 6)**

**Задание 1**



**Задание 2**

float\*\* CreateTable (float a, float b, size\_t n, func\_f f)

{

int i, k=n+1;

float h = (b-a)/n;

float\*\* matr = (float\*\*)calloc(n, sizeof(float\*));

for (i=0; i<k; i++)

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

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

for (i=1;i<k;i++)

{

matr[i][0]=matr[0][0]+i\*h;

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

}

OutputMatr(matr, k, 2);

return matr;

}

**Задание 3**

В качестве базисных функций возьмём *, k=0, 1, 2, ...*

**Задание 4**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include "gauss.h"

#include <locale.h>

typedef float (\*func\_f)(float);

typedef float (\*func\_2)(float, int);

float FuncMy (float x);

void OutputMatr (float\*\* a, size\_t n, size\_t m);

float\*\* GetMem (size\_t n, size\_t m);

void FreeMem (float\*\* matr, size\_t n);

float\*\* CreateTable (float a, float b, size\_t n, func\_f f);

float BasFu(float x, int k);

int MinKvadr (float a, float b, size\_t n, size\_t m, func\_f myf, func\_2 aproc, float \*rez);

int main()

{

setlocale(LC\_ALL, "Rus");

float a=0, b=1,

int n=15, m=1;

int i;

float \*rez = (float\*)calloc(m+1, sizeof(float));

if (MinKvadr(a, b, n, 1, FuncMy, BasFu, rez))

{

for (i=0; i<m+1; i++)

printf("%f ",rez[i]);

}

free(rez);

return 0;

}

float FuncMy (float x)

{

return pow(3, x)\*cos(x-6);

}

void OutputMatr (float\*\* a, size\_t n, size\_t m)

{

int i, j;

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

{

for (j=0; j<m; j++)

printf("%.2f ", a[i][j]);

printf("\n");

}

}

float\*\* GetMem (size\_t n, size\_t m)

{

int i;

float\*\* matr = (float\*\*)calloc(n, sizeof(float\*));

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

matr[i] = (float\*)calloc(m, sizeof(float));

return matr;

}

void FreeMem (float\*\* matr, size\_t n)

{

int i;

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

free(matr[i]);

free(matr);

}

float\*\* CreateTable (float a, float b, size\_t n, func\_f f)

{

int i;

int k=n+1;

float h = (b-a)/n;

float\*\* matr = GetMem(k, 2);

matr[0][0]=a;

matr[0][1]=f(matr[0][0]);

for (i=1; i<k; i++)

{

matr[i][0] = matr[0][0]+i\*h;

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

}

printf("Таблица значений: \n");

OutputMatr(matr, k, 2);

printf("\n");

printf("\n");

return matr;

}

float BasFu(float x, int k)

{

return powf(x,k);

}

int MinKvadr (float a, float b, size\_t n, size\_t m, func\_f myf, func\_2 aproc, float \*rez)

//отрезок [a, b], колво точек n, кол-во аппрокс функций m+1

//функция варианта myf, аппроксимирующая функция aproc

// \*rez - результат = y

//возвращает 1, если найдено решение и 0 в противном случае

{

float \*\*Amatr = CreateTable(a, b, n, myf);

float \*\*Fmatr = GetMem(m+1, m+2);

int i, j, p, flag=0;

for (i=0; i<m+1; i++)

{

for (j=0; j<m+1; j++)

{

Fmatr[i][j]=0;

for (p=0; p<n+1; p++)

{

Fmatr[i][j] = Fmatr[i][j]+aproc(Amatr[p][0], j)\*aproc(Amatr[p][0], i);

}

}

Fmatr[i][m+1]=0;

for (p=0; p<n+1; p++)

{

Fmatr[i][m+1]=Fmatr[i][m+1]+Amatr[p][1]\*aproc(Amatr[p][0],i);

}

}

if (DecisSLAUGaus(Fmatr, m+1, m+2))

{

for (i=0;i<m+1;i++)

rez[i]=Fmatr[i][m+1];

flag=1;

}

else

printf("Нет решений.\n");

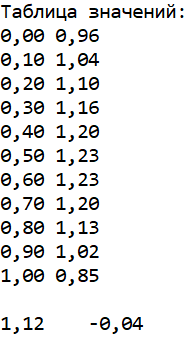
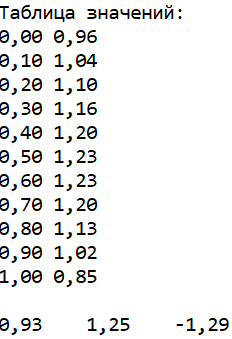
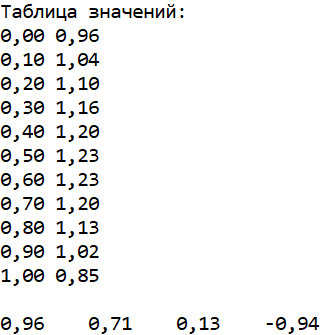
FreeMem(Amatr, n+1);

FreeMem(Fmatr, m+1);

return flag;

}

**a = 0; b = 1; m = 1, 2, 3; n = 15;**

****  

**Задание 5**

1)

2)

3)

**Задание 6**

****