МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №2

КУРСОВАЯ РАБОТА  
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  |  |  | Галанина В.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ |
| Аппроксимация функции методом наименьших квадратов |
| по дисциплине: ИНФОРМАТИКА |
|  |
|  |

РАБОТУ ВЫПОЛНИЛА

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТКА ГР. № | 2746 |  |  |  | Келлер А.Г. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2018

Оглавление

[Аппроксимация функции методом наименьших квадратов 1](#_Toc514676807)

[1.Цель работы 4](#_Toc514676808)

[2.Постановка задачи 4](#_Toc514676809)

[2.2.Описание метода выбора аппроксимирующей функции. 6](#_Toc514676810)

[2.3.Описание метода простой итерации. 9](#_Toc514676811)

2.4. Метод последовательных приближений……………………………………………………..11

[2.5. Оценка погрешности аппроксимации ………………………………………………………12](#_Toc514676812)

[3.Ручной счёт. 13](#_Toc514676813)

[4.Схемы алгоритмов и их описание. 17](#_Toc514676815)

[5.Программа и результаты расчётов параметров на компьютере. 22](#_Toc514676819)

[6.Заключение. 26](#_Toc514676820)

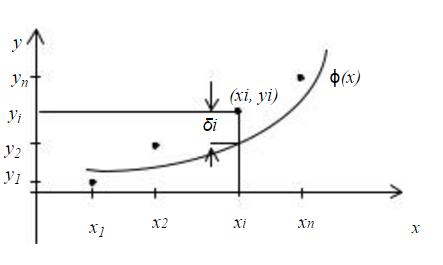
# 1.Цель работы

1. Закрепление навыков алгоритмизации и программирования.
2. Практическое освоение типовых вычислительных методов прикладной математики;
3. Совершенствование навыков разработки алгоритмов и построение программ на языке высокого уровня;
4. Освоение принципов модульного программирования и техники использования библиотек на DevC++.

# 2.Постановка задачи

Тематика курсовой работы связана с решением задачи аппроксимации зависимости между величинами Х и Y, полученной экспериментальным путем, с помощью известных функций или их комбинаций, подобранных надлежащим образом. Эта зависимость может быть задана значениями в отдельных точках (xi, yi), либо некоторой функцией y=f(x), заданной на интервале [a, b].

Исходными данными являются функциональная зависимость между X и Y, заданная таблично в n точках, и набор базисных функций. Требуется, используя МНК, найти аппроксимирующую функцию ϕ(x) из заданного класса функций и оценить степень ее отклонения от исходной функции.

Пусть из теоретических или иных соображений (например, по графику) выбран вид аппроксимирующей функции *y*=φ(*x*). Эта функция в качестве параметров помимо *x*содержит еще ряд числовых параметров (C1, …, Cm).

Один из распространённых подходов опирается на использование метода наименьших квадратов (МНК), в соответствии с которым наилучшей считается такая аппроксимирующая функция , для которой достигается наименьшее значение суммы квадратов отклонений от значений аппроксимирующей функции во всех точках *х*, принимаемых во внимание.

Запишем требование МНК аналитически:



В настоящей курсовой работе исходные данные заданы в виде табличной зависимости . Уточним условия МНК для этой задачи.

**2.1**. **Методика выбора аппроксимирующей функции**

Аппроксимирующую функцию выбирают из некоторого семейства функций, для которого задан вид функции, но остаются неопределенными (и подлежат определению) её параметры , т.е.

(2)

Определение аппроксимирующей функции подразделяется на два основных этапа:

- подбор подходящего вида функций ;

- нахождение параметров функции в соответствии с критерием МНК.

Подбор вида функции представляет собой сложную задачу, решаемую методом проб и последовательных приближений. Исходные данные, представленные в графической форме, сопоставляются с семействами графиков ряда типовых функций, используемых обычно для аппроксимации.

После того как выбран вид аппроксимирующей функции и, следовательно, определена функциональная зависимость, необходимо найти в соответствии с требованием МНК значения параметров С1, С2,…, Сm.

Параметры должны быть определены таким образом, чтобы значения критерия было наименьшим.

Для решения задачи подставим выражение (2) в выражение (1) и проведем необходимые операции суммирования. В результате величина, критерий аппроксимации, представится функцией искомых параметров

(3)

# 2.2.Описание метода выбора аппроксимирующей функции.

Аппроксимирующую функцию выбирают из некоторого семейства функций, для которого задан вид функции, но остаются неопределенными (и подлежат определению) её параметры , т.е.

(2)

Для решения задачи подставим выражение (2) в выражение (1) и проведем необходимые операции суммирования. В результате величина , критерий аппроксимации, представится функцией искомых параметров

(3)

Последующие действия сводятся к отыскиванию минимума этой функции переменных . Определение значений, соответствующих этому минимуму , и является целью решаемой задачи. Поскольку величина неотрицательна (как сумма квадратов) и нижняя её граница есть 0 (), т.о., если существующее решение системы единственно, оно отвечает именно минимуму

Уравнения, используемые в МНК, называются нормальными, поэтому описываемый способ решения задачи условимся называть методом нормальных уравнений.

Структура этих уравнений получается более простой в том случае, когда аппроксимирующая функция выбирается линейной функцией искомых параметров и выражение (2) имеет вид:

где определяемые параметры; – система некоторых линейно-независимых функций, называемых в курсовой работе базисными функциями.

*Замечание*. Функции называются линейно-независимыми, если при любых *х* равенство

справедливо только тогда, когда все =0.

Примеры функций, которые можно использовать в качестве базисных

|  |  |
| --- | --- |
| Вид функции | Название функций |
|  | линейная |
| *x2* | параболическая |
|  | обратно-пропорциональная |
| logax | логарифмическая |
| sin(x) | тригонометрические |

В этом случае, подставляя (4) в выражение (1) и выполняя дифференцирование, получим систему уравнений относительно искомых .

Подставим выражение аппроксимирующей функции для m базисных функций

в формулу критерия аппроксимации.

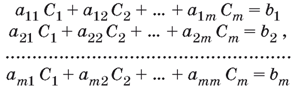
Получим

*2*

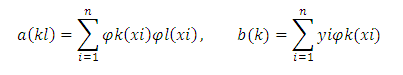
Применим операцию дифференцирования к параметру С1:

и, выполняя необходимые алгебраические преобразования, получим уравнение

Аналогичные уравнения можно получить, применяя описанные выше действия по отношению к переменным С2, …, Сm. Эти уравнения образую систему нормальных уравнений



где коэффициенты akl и величины bk(k, l=1, 2, …, m) определяются выражениями



Систему m линейных уравнений можно записать посредством матричных обозначений в следующем виде:



Квадратная матрица А называется матрицей системы, вектор С – вектором-столбцом неизвестных системы, а вектор В – вектором-столбцов свободных членов.

Решение системы линейных уравнений сводится к отысканию значений элементов вектора-столбца С, называемых корнями системы. Для получения единственного решения системы, входящие в нее m уравнений должны быть линейно независимыми. Необходимым и достаточным условием этого является неравенство нулю определителя данной системы, то есть det A0.

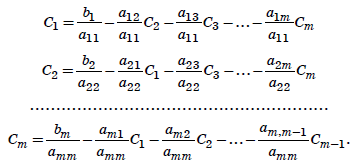
Алгоритмы решения систем линейных уравнений подразделяются на прямые и итерационные. Теоретически для получения точного решения итерационные методы требуют бесконечного числа в арифметических операций. Практически это число приходится брать конечным, поэтому решение имеет некоторую ошибку. Что же касается прямых методов, то они даже при конечном числе операций могут в принципе дать точное решение, если пренебречь ошибками округлений при вычислениях на современных ПК.

# 2.3.Описание метода простой итерации.

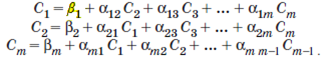
Итерационные методы позволяют получить значения корней системы с заданной точностью в виде предела последовательности некоторых векторов C(0),C(1), …,C(k). Процесс получения элементов такой последовательности носит итерационный (повторяющийся) характер.

Эффективность применения таких методов зависит от удачного выбора начального вектора C(0) и быстроты сходимости процесса.

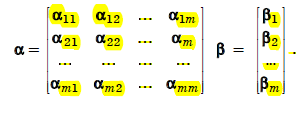
Полагая, что в системе уравнений все диагональные элементы отличны от нуля, то есть *aii*≠ 0 (*i*= 1, 2, …,*m*), выразим *C*1 через первое уравнение системы,*C*2 через второе уравнение и т.д. В результате получим новую систему.



Обозначим bi / aij =i- aij. где i,j = 1, 2,...,m. В этом случае новая система принимает вид



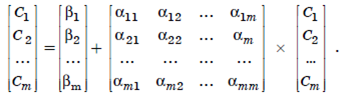
Введём обозначения



Отметим, что в данном случае aij=0(i=1, 2, …, m). Тогда система может быть записана в матричной форме

                                            С=β+ a c

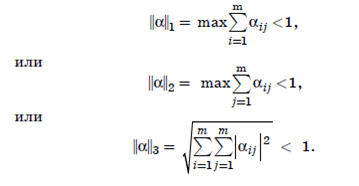
или



Сходимость процесса зависит от величины элементов матрицы a следующим образом: если сумма модулей элементов строк или сумма модулей элементов столбцов меньше единицы, то итерационный процесс отыскания приближений к истинным значениям корней системы сходится к единственному решению независимо от выбора начальных (в частности, нулевых) приближений,то есть условия сходимости можно записать следующим образом:



Сходимость итерационного процесса можно оценить и посредством норм матрицы а. А именно, процесс сходится, если выполняется одно из следующих условий:



Эти условия как достаточные, предъявляют завышенные требования к матрице, поэтому в некоторых случаях сходимость обеспечивается, если даже |a|≥1.

Условия сходимости также выполняются, если в матрице А диагональные элементы преобладают, то есть



Другими словами, модули диагональных коэффициентов в каждом уравнении системы больше суммы модулей недиагональных коэффициентов (свободные члены не рассматриваются).

Пусть С- вектор точных значенийй неизвестных (корней) системы уравнений, а – вектор k-х приближений к этимточным значениям. Тогда для оценки погрешности метода последовательных приближений, где eps – относительная точность вычислений.

||C-C(k)||≤ eps

На практике при заданном значении eps часто в качестве критерия окончания итерационного процесса вычисления значенийвектора используется неравенство

**2.4.Схема алгоритма решения системы линейных уравнений методом последовательных приближений**

Любое (k + 1) - е приближение вычисляется по формуле

= β + α ×.

Алгоритм начинается с определения условия сходимости итерационного процесса. В случае, если условия сходимости не выполняется, происходит аварийное завершение программы и вычисления прекращаются.

## 2.5. Оценка погрешности аппроксимации

Результатом этапа решения системы нормальных уравнений является получение значений параметров аппроксимирующей функции

**

для заданного набора базисных аппроксимирующих функций

, , …,

Решение системы нормальных уравнений определяет значение параметров, при которых критерий качества аппроксимации J принимает минимально возможное значение Jmin=J(C1\*, C2\*, …, Cm\*). При всех других допустимых значениях параметров величина критерия будет больше. Тем самым полученное значение Jmin может быть принято за характеристику эффективности аппроксимации заданной функциональной зависимости функциями выбранного класса. При изменении класса аппроксимирующей функции, а также при изменении набора базисных функций значение Jmin может меняться. Сравнение различных классов функций по их эффективности аппроксимации может осуществляться на основе сравнений соответствующих значений Jmin.

Для количественной оценки погрешности аппроксимации может использоваться также величина максимального отклонения исходной функциональной зависимости от найденной аппроксимирующей. Для этого определяется отклонение во всех заданных точках и определяется максимальное из этих отклонений:

# 3.Ручной счёт.

# Представление исходных данных (табличное)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер варианта | n | Значения | | | | | | Базисные функции | | | Метод решения СЛАУ |
|  |  |  |
| 7 | 5 | Х | -1 | -0.6 | -0.1 | 0.2 | 0.7 | x | 1 |  | Простой итерации |
| У | 0.4 | 0.6 | 1 | 1.3 | 1.8 |

Выражение для аппроксимации функции будет иметь следующий вид:

F(x)=

F(x)=(3)

Выражение для критерия аппроксимации:

В соответствии с условиями локального минимума функции J() найдём частные производные ,, и приравняем их к нулю:

Получаем систему уравнений:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | x | y |  |  |  |  |  |  |
| 1 | -1 | 0,4 | 1 | -2 | 2 | 4 | -0,4 | 0,8 |
| 2 | -0,6 | 0,6 | 0,36 | -0,048 | 0,08 | 0,0064 | -0,36 | 0,048 |
| 3 | -0,1 | 1 | 0,01 | 0,097 | -0,97 | 0,9409 | -0,1 | -0,97 |
| 4 | 0,2 | 1,3 | 0,04 | -0,176 | -0,88 | 0,7744 | 0,26 | -1,144 |
| 5 | 0,7 | 1,8 | 0,49 | 0,329 | 0,47 | 0,2209 | 1,26 | 0,846 |
| 𝜮 | -0,8 | 5,1 | 1,9 | -1,798 | 0,7 | 5,9426 | 0,66 | -0,42 |

Используя значение из таблицы, запишем систему уравнений в окончательном виде:

1,9=0,66

+5+0,7=

-1,798+0,7+5,9426=-0,42

=

AC=B

Решаем систему уравнений методом простой итерации

Оценка сходимости процесса:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 2,097284211 | 1,1844 | -0,471732912- |
| 0,399654855 | 1,421608081 | 0,424365936 |
| 1,347524 | 1,024534 | -0,11721 |
| 0,667831051 | 1,252013641 | 0,216348979 |
| 1,079266514 | 1,096564111 | -0,016095534 |
| 0,793848168 | 1,194936017 | 0,126699814 |
| 0,97039741 | 1,129277733 | 0,028755729 |
| 0,8500658 | 1,1712378 | 0,0899068 |
| 0,925601393 | 1,143423574 | 0,048556495 |
| 0,87476 | 1,161298 | 0,074687 |
| 0,907014 | 1,149505 | 0,057199 |
| 0,8855 | 1,1571 | 0,0683 |

Вычисляем погрешность:

0,8855

1,1571

0,0683

(x)=0,8855X+1,1571+0,0683 ()

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | -1 | -0,6 | -0,1 | 0,2 | 0,7 |
|  | 0,4 | 0,6 | 1 | 1,3 | 1,8 |
| () | 0,408 | 0,631 | 1 | 1,274 | 1,809 |
|  | 0,0083 | 0,031 | 0,002 | 0,026 | 0,009 |

J()==0,001

=0,031 при x=-0,6

[](https://vk.com/photo270903376_456242915)

# 4.Схемы алгоритмов и их описание.

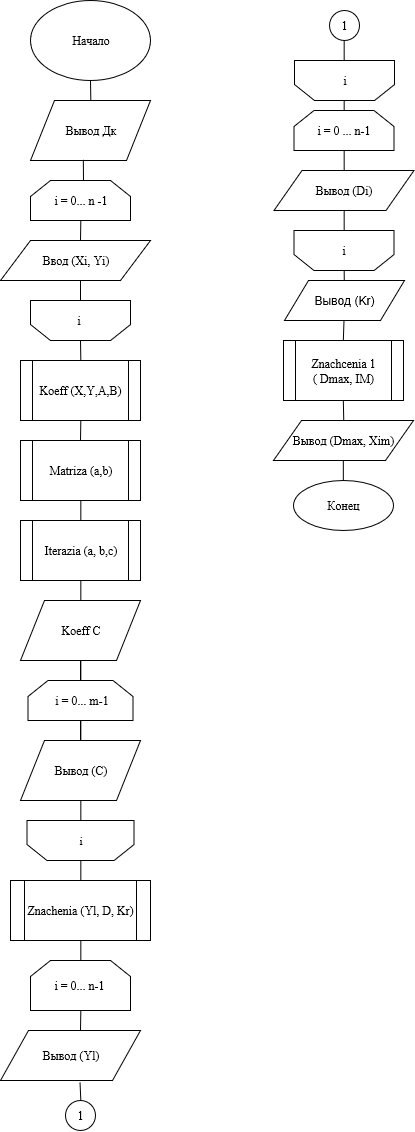
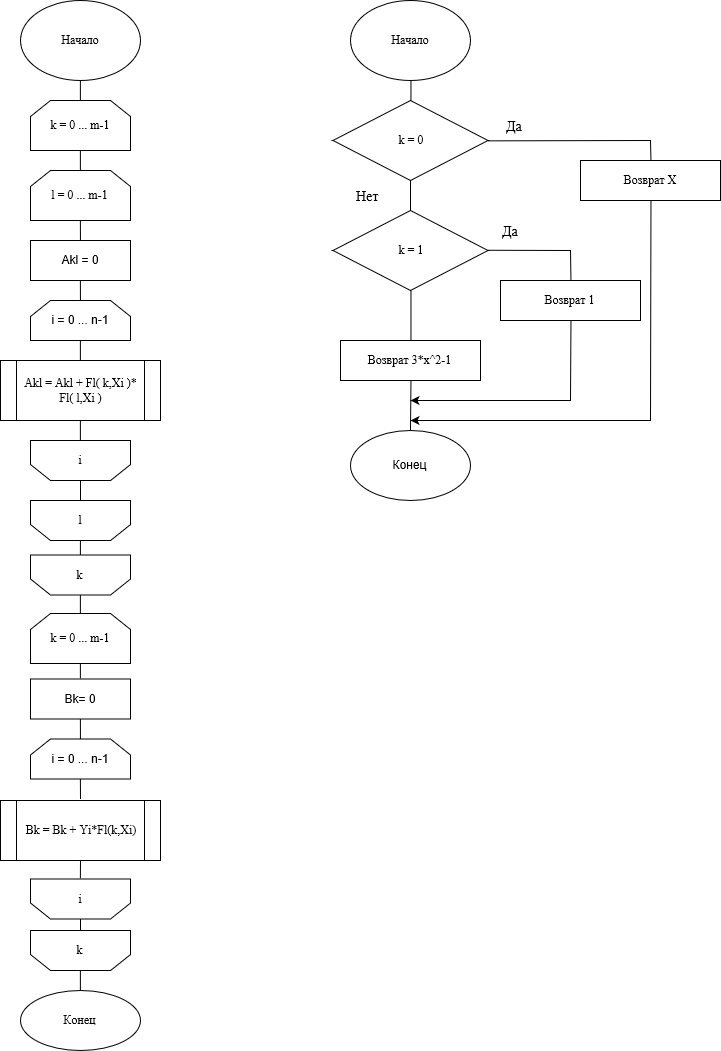
[](https://vk.com/photo78137189_456241022)

Схема алгоритма основной программы int main ()

Схема алгоритма koeff Fl (k,x[i])

[](https://vk.com/photo78137189_456241023)

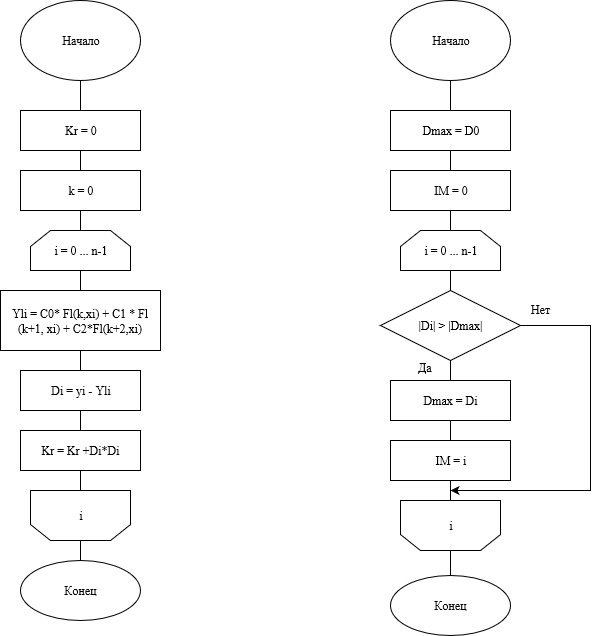
# Iterazia

# [https://sun9-9.userapi.com/c834303/v834303772/110432/ls4OLMzyIa4.jpg](https://vk.com/photo78137189_456241031)

Opred shodimosti

# [https://sun9-4.userapi.com/c831208/v831208772/cb1a3/1v9he8rtvSQ.jpg](https://vk.com/photo78137189_456241029)

Znachenia (Yl, D, Kr) Znachenia 1 (Dmax, IM)

[](https://vk.com/photo78137189_456241032)

# 5.Программа и результаты расчётов параметров на компьютере.

#include <conio.h>

#include <math.h>

#include <stdio.h>

#include <string.h>

#include <iostream>

#define n 5

#define m 3

using namespace std;

void vvod(float E[n])

{

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

{

cin>>E[i];

}

cout<<("\n");

}

float Fl(int k, float x)

{

if (k==0)

return x;

if (k==1)

return 1;

if (k==2)

return 3\*pow(x,2)-1;

else return 0;

}

void koeffA(float A[(m)][(m)],float X[n])

{

int i,l,k;

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

{

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

{

A[k][l]=0;

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

{

A[k][l]=A[k][l]+Fl(k,X[i])\*Fl(l,X[i]);

}

}

}

}

void koeffB(float X[n],float Y[n],float B[(m)])

{

int k,i;

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

{

B[k]=0;

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

{

B[k]=B[k]+Y[i]\*Fl(k,X[i]);

}

}

}

int schod(float A[m][m])

{

int Flag,i,j,Key;

float Ad,Sd;

Flag=0;

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

{

Ad=fabs(A[i][i]);

Sd=0;

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

if(j!=i)

Sd+=fabs(A[i][j]);

if(Ad>Sd)

Flag=1;

}

if (Flag==1)

Key=0;

else

Key=1;

return Key;

}

void iterazia(float A[m][m],float B[m],float C[m])

{ int Key;

float eps,test,sum,p[m];

cout<<"Vvedide eps \n";

cin>>eps;

Key=schod(A);

if(Key==1)

{

cout<<("Pr rashoditsa\n");

abort();

}

else

{

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

{

C[i]=B[i];

}

do

{

test=0;

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

{

sum=0;

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

if(j!=i)

sum+=A[i][j]\*C[j];

p[i]=(B[i]-sum)/A[i][i];

test+=fabs(p[i]-C[i]);

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

C[i]=p[i];

}

}

while (test>= eps);

}

}

float approcs (float C[m],float Yl[n],float D[n],float X[n],float Y[n])

{

float Kr;

int k, i;

Kr=0;

k=0;

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

{

Yl[i]=C[0]\*Fl(k,X[i])+C[1]\*Fl((k+1),X[i])+C[2]\*Fl((k+2),X[i]);

D[i]=fabs(Y[i]-Yl[i]);

Kr=Kr+pow(D[i],2);

}

return Kr;

}

void krappr(float Y[m],float Yl[n],float D[n],float \*Dmax,int \*IM)

{

int i;

\*Dmax=D[0];

\*IM=0;

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

{

if(fabs(D[i])>fabs(\*Dmax))

{

\*Dmax=(D[i]);

\*IM=i;

}

}

}

void vyvod(float E[n])

{

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

{

printf("%.3f ",E[i]);

}

printf("\n");

}

int main()

{

float X[n],Y[n], A[m][m],B[m],C[m],Yl[n],D[n],Dmax,Kr;

int IM, i, j, k;

cout<<("vvedite X\n");

vvod(X);

cout<<("vvedite Y\n");

vvod(Y);

koeffA(A,X);

koeffB(X,Y,B);

cout<<" matrica  A\n";

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

{

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

cout<<" "<<A[i][j];

cout<<"\n";

}

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

{

B[i]=0;

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

B[i]= B[i]+Fl(i,X[k])\*Y[k];

}

cout<<"massiv  B\n";

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

cout<<" "<<B[i];

cout<<"\n";

iterazia(A,B,C);

cout<<"Znachenie C: \n";

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

{

cout<<" "<<C[i];

cout<<"\n";

}

Kr=approcs(C,Yl,D,X,Y);

krappr(Y,Yl,D,&Dmax,&IM);

cout<<"Znachenie appr f(x):\n";

vyvod(Yl);

cout<<"Znacheniya otkloneniy: \n";

vyvod(D);

cout<<"Max pri = "<<Dmax<<"\n";

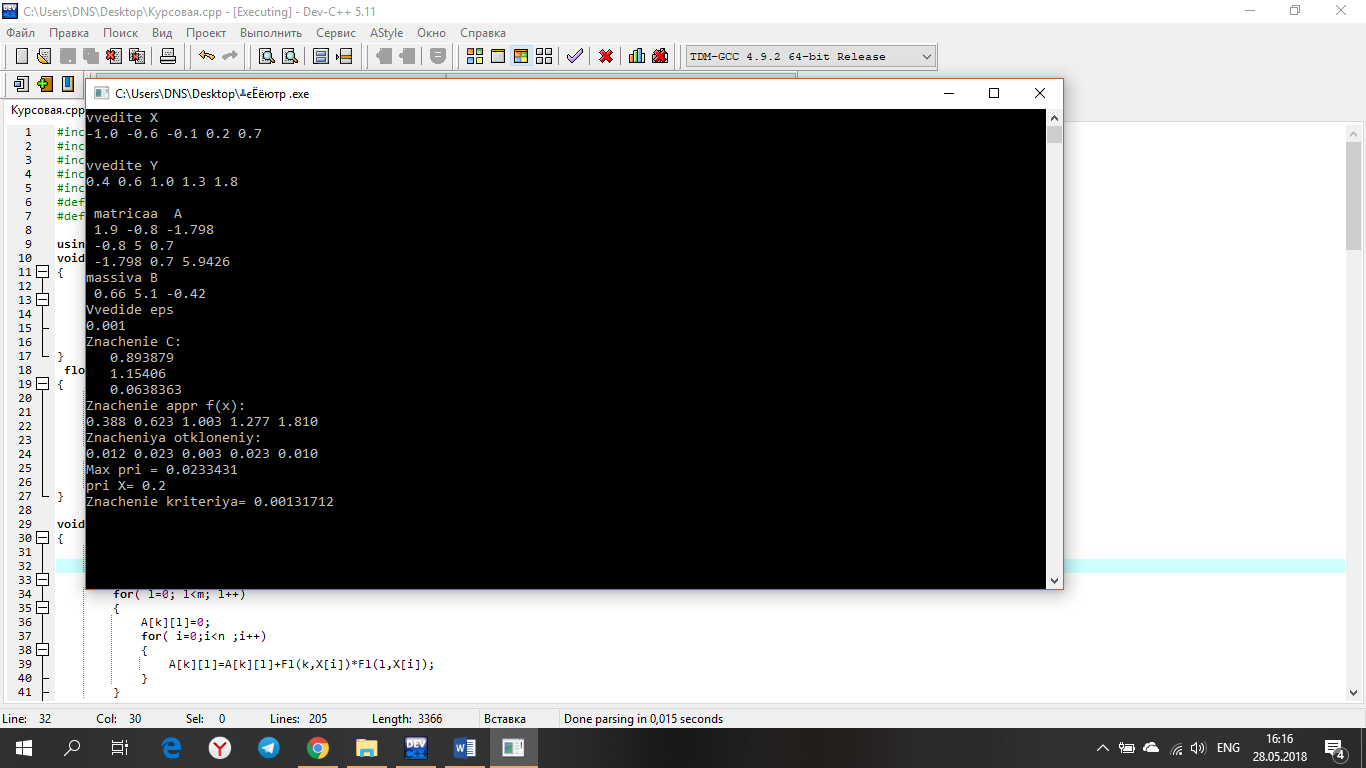
cout<<"pri X= "<<X[IM]<<"\n";

cout<<"Znachenie kriteriya= "<<Kr<<"\n";

getch ();

return 0;

}



# 6.Заключение.

В результате проделанной работы был описан критерий аппроксимации, способ его минимизации, составлена система нормальных уравнений, параметры которой вычислили при помощи метода простой итерации, рассчитаны отклонения аппроксимирующей функции, а также максимальное по модулю отклонение. Расчёты составлены двумя способами:

* Подсчитанные вручную;
* Подсчитанные на ЭВМ, при помощи составленной программы.