**Приложение 1.**

Матвеева Антонина Гавриловна №241-922-342

Учитель информатики МОУ СОШ №17 с углубленным изученим математики г. Тверь

**Разработка программы на языке программирования Паскаль «Решения системы линейных уравнений» разными методами.**

Содержание

[1. Описание математических методов решения систем линейных уравнений](#_Toc345711491)

[1.1 Метод Гаусса](#_Toc345711492)

[1.2 Матричный метод](#_Toc345711493)

[1.3 Вычисление определителей второго и третьего порядка](#_Toc345711494)

[1.4 Решение системы линейных уравнений с тремя неизвестными методом Крамера](#_Toc345711495)

[2. Описание программы](#_Toc345711496)

[2.1 Работа программы](#_Toc345711497)

[2.2 Блок-схема программы](#_Toc345711498)

## 1. Описание математических методов решения систем линейных уравнений

## 1.1 Метод Гаусса

Идея метода Гаусса состоит в последовательном исключении неизвестных. Алгоритм решения системы уравнений этим методом проследим на примере.

Пример 1. 

Выбирается ведущее уравнение с коэффициентом при х1, равным 1. В нашем примере ведущим уравнением будет второе. Систему лучше переписать, поставив это уравнение на первое место:



Умножаем первое уравнение на 6 и вычитаем из полученного второе, чтобы исключить из второго неизвестное х1. Первое уравнение записываем, а на место второго - результат вычитания.

Затем первое уравнение умножим на 3 и складываем с третьим уравнением. Тогда получаем систему

 Или

первое уравнение переписываем без изменения, а второе умножаем на 7 и вычитаем из него третье уравнение, умноженное на 15, чтобы избавиться от х2 в третьем уравнении. При этом второе записываем без изменения, на месте третьего - результат вычитания. Тогда



Из третьего следует х3 =-3, подставим его во второе, получим х2 = - 2. Далее подставим найденные х2 и х3 в первое уравнение, получим х1 = 1.

Решение системы: х1 = 1, х2 = - 2, х3= - 3.

Примечание: если система уравнений не содержит уравнения с коэффициентом 1 при х1, тогда исключение х1 из второго и третьего достигается умножением сначала первого на коэффициент второго, а второго на коэффициент первого. Затем умножаем первое на коэффициент третьего, а третье на коэффициент первого. Таким образом при вычитании исключаем х1.

## 1.2 Матричный метод

Запишем систему линейных 3 уравнений с 3 неизвестными



Составим матрицу из коэффициентов при неизвестных

А = 

Введем в рассмотрение матрицы - столбцы для неизвестных и свободных членов:

Х = ; В = .

Тогда систему (2) можно переписать в матричной форме

*АХ*=*В*

Умножив это уравнение на  слева, получим , откуда =или 

Следовательно, матрица - решение *Х* находится как произведение на *В*.

Пример 2. Решить систему уравнений матричным методом



Решение: определитель матрицы

А=

∆=-1, значит, существует обратная матрица .

Матрица - столбец при неизвестных:

*Х* = 

Матрица - столбец из свободных членов:

*В* = 

Тогда решение запишется в виде

==

Откуда следует, х1 = 1; х2 = 0; х3 = 2.

## 1.3 Вычисление определителей второго и третьего порядка

Число (*а*11 *а*22 - *а*12 *а*21) называется определителем второго порядка и обозначается символом



Определитель второго порядка содержит две строки и два столбца. Числа *а*11, *а*12, *а*21, *а*22 называются элементами определителя. Диагональ определителя, на которой расположены числа *а*11, *а*22 - главная, а элементы *а*12, *а*21 составляют побочную диагональ.

Определитель 3-го порядка содержит три строки и три столбца:



Для вычисления определителя третьего порядка существует несколько способов.

Рассмотрим метод вычисления определителя разложением по элементам первой строки.

Введем понятие минора и алгебраического дополнения.

Минором некоторого элемента определителя называется определитель, полученный из данного вычеркиванием той строки и того столбца в которых этот элемент расположен. Обозначается Мij (i - номер строки, j - номер столбца).

Например, минором элемента а12 является определитель



Алгебраическим дополнением данного элемента определителя называется его минор, умноженный на (-1) i+j. Алгебраические дополнения обозначаются буквами Аij*,* и тогда Аy= (-1) i+j My.

Определитель вычисляется так:

=.

Так же можно разложить определитель по любой строке или столбцу.

Изложенный метод применим к вычислению определителей 4-го и т.д. порядков.

Пример3. Вычислить определитель разложением по элементам первой строки



Решение: Элементы первой строки

*а*11 = 1, *а*12 = 2, *а*13 = - 2.

А11 = (-1) 1+1. М11==4+1=5.

М11 получили, вычеркнув первую строку и первый столбец.

А12 = (-1) 1+2. М12= -  = - (8+3) = - 11.

М12 получили, вычеркнув первую строку и второй столбец.

А13 = (-1) 1+3. М13 =  = 2-3 = - 1.

М13 получили, вычеркнув первую строку и третий столбец.

Окончательно

 = 1.5+2. (-11) - 2. (-1) = - 15

## 1.4 Решение системы линейных уравнений с тремя неизвестными методом Крамера

Система линейных уравнений:

Определители:

Решение:

.

Пример:

Определители:


## 2. Описание программы

## 2.1 Работа программы

Для решения систем линейных уравнений методом Гаусса и матричным методом создана программа на языке Паскаль. Программа запрашивает исходные данные (рис.1):

матрицу коэффициентов при неизвестных х;

столбец свободных членов

способ решения системы линейных уравнений - вариант 1 или 2.

Рисунок 3.1 Ввод исходных данных

В зависимости от выбранного вариант в программе происходит решение системы уравнений методом Гаусса (рис.2) или матричным методом (рис.3) с выдачей на экран результатов:

Рисунок 3.2 Результаты расчетов системы линейных уравнений методом Гаусса.

Рисунок 3.3 Результаты расчетов системы линейных уравнений матричным методом.

Программа состоит из 7 подпрограмм - 6 процедур и одной функции:

процедура Gauss обеспечивает решение системы линейных уравнений по методу Гаусса;

процедура matrica обеспечивает решение системы линейных уравнений матричным методом;

процедура PrintMatr2 предназначена для выдачи на экран исходной и обратной матрицы;

процедура MultString предназначена для умножения строк матрицы на число r;

процедура AddStrings прибавляет к i1-ой строке матрицы i2-ю, умноженную на число r;

процедура MultMatr предназначена для умножения матриц.

Функция Sign используется для изменения знака на противоположный при вычислении обратной матрицы.

Программа настроена на решение системы 3-х линейных уравнений с тремя неизвестными. Чтобы решить систему из 2-х уравнений с 2-мя неизвестными необходимо в программе изменить значение константы N с N=3 на N =2 (рис.4).

Рисунок 3.4. Фрагмент программы с описанием констант и переменных.

## 2.2 Блок-схема программы

начало

I=1 до N (1)

J=1 до N (1)

Ввод а[i,j]

I=1 до N (1)

Ввод b[i]

Var=1

Процедура GAUSS

конец

N=3

Ввод варианта

Var=2

Процедура MATRICA

неверный вариант

##

Вход (a, b, x, n)

k=1 до N-1 (1)

i=k+1 до N (1)

m=a[I,k]/a[k,k]

a[k,k]=0

вывод х[i]

j=k+1 до N (1)

a[I,j]=a[I,j]-m\*a[k,j]

b[i]=b[i]-m\*b[k]

x[n]=b[i]/a[n,n]

i=n-1 до 1 (1)

s=0

s=s-a[I,j]\*x[j]

j=i+1 до n (1)

x[i]=(b[i]+s)/a[i,i]

Выход из процедуры

Процедура GAUSS

**Текст программы**

"Решение систем линейных уравнений матричным способом и методом Гаусса"

Program Lin\_yravneniya;

uses crt;

const N=3;

eps=0.00001; { all numbers less than eps are equal 0 }

type matr=array [1. n,1. n] of real;

mas=array [1. n] of real;

var

i,j: integer;

b,x: mas;

variant: byte;

a,c: matr;

dt: real;

imx,np: integer;

{\*\*\* печать исходной и обратной матрицы\*\*\* }

procedure PrintMatr2 (m,m1: matr; n,nz,nd: integer);

var i,j: integer;

begin

for i: =1 to n do

begin

if (i=1) then write (np: 2,': ')

else write (' ');

for j: =1 to n do

write (m [i,j]: nz: nd); write (' ');

for j: =1 to n do

write (m1 [i,j]: nz: nd);

writeln;

end;

inc (np);

end;

procedure MultString (var a,b: matr; i1: integer; r: real);

var j: integer;

begin

for j: =1 to n do

begin

a [i1,j]: =a [i1,j] \*r;

b [i1,j]: =b [i1,j] \*r;

end;

end;

procedure AddStrings (var а,b: matr; i1, i2: integer; r: real);

{ процедура прибавляет к i1 строке матрицы а i2-ю умноженную на r}

var j: integer;

begin

for j: =1 to n do

begin

a [i1,j]: =a [i1,j] +r\*a [i2,j] ;

b [i1,j]: =b [i1,j] +r\*b [i2,j] ;

end;

end;

procedure MultMatr (a,b: matr; var c: matr);

var i,j,k: byte;

s: real;

begin

for i: =1 to n do

for j: =1 to n do

begin

s: =0;

for k: =1 to n do

s: =s+a [i,k] \*b [k,j] ;

c [i,j]: =s;

end;

end;

function sign (r: real): shortint;

begin

if (r>=0) then sign: =1 else sign: =-1;

end;

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

{\*\* вычеркивание из матрицы строки и столбца \*\*}

procedure GetMatr (a: matr; var b: matr; m, i,j: integer);

var ki,kj,di,dj: integer;

begin

di: =0;

for ki: =1 to m-1 do

begin

if (ki=i) then di: =1;

dj: =0;

for kj: =1 to m-1 do

begin

if (kj=j) then dj: =1;

b [ki,kj]: =a [ki+di,kj+dj] ;

end;

end;

end;

**{\*\*\* метод Гаусса \*\*\*\*\*\*\*}**

procedure gauss (a: matr; b: mas; var x: mas; n: integer);

Var k: byte;

m, s: real;

begin

{ приведение к треугольному виду}

For k: =1 to N-1 do

For i: =k+1 to n do

begin

m: =a [i,k] /a [k,k] ;

a [i,k]: =0;

For j: =k+1 to N do

a [i,j]: =a [i,j] -m\*a [k,j] ;

b [i]: =b [i] -m\*b [k] ;

end;

{расчет неизвестных х в обратном порядке}

x [n]: =b [n] /a [n,n] ;

writeln;

writeln ('Вывод результатов решения системы уравнений методом Гаусса');

writeln ('x [',n,'] =',x [n]: 6: 2);

for i: = (n-1) downto 1 do

begin s: =0;

For j: =i+1 to n do

s: =s-a [i,j] \*x [j] ;

x [i]: = (b [i] +s) /a [i, i] ;

writeln ('x [', i,'] =',x [i]: 6: 2);

end;

end;

{\*\*\* матричный способ \*\*\*}

procedure matrica (a: matr; y: mas; n: integer);

var z,a0: matr;

imx,np: integer;

s: mas;

begin

for i: =1 to n do

begin

for j: =1 to n do z [i,j]: =0;

z [i, i]: =1;

end;

for i: =1 to n do

for j: =1 to n do

a0 [i,j]: =a [i,j] ;

for i: =1 to n do

begin

{ к i-ой строке прибавляем (или вычитаем) j-ую строку

взятую со знаком i-того элемента j-ой строки. Таким образом,

на месте элементова a [i, i] возникает сумма модулей элементов i-того

столбца (ниже i-ой строки) взятая со знаком бывшего элемента a [i, i],

равенство нулю которой говорит о несуществовании обратной матрицы }

for j: =i+1 to n do

AddStrings (a,z, i,j,sign (a [i, i]) \*sign (a [j, i]));

{ PrintMatr (a,b,n,6,1); }

{ прямой ход }

if (abs (a [i, i]) >eps) then

begin

MultString (a,z, i,1/a [i, i]);

for j: =i+1 to n do

AddStrings (a,z,j, i,-a [j, i]);

{ PrintMatr (a,b,n,6,1); }

end

else

begin

writeln ('Обратной матрицы не существует. ');

halt;

end

end;

{обратный ход: '); }

if (a [n,n] >eps) then

begin

for i: =n downto 1 do

for j: =1 to i-1 do

begin

AddStrings (a,z,j, i,-a [j, i]);

end;

{ PrintMatr (a,b,n,8,4); }

end

else writeln ('Обратной матрицы не существует. ');

MultMatr (a0,z,a);

writeln ('Начальная матрица, обратная к ней матрица: ');

PrintMatr2 (a0,z,n,7,3);

{\*\* умножение обратной матрицы на столбец свободных членов \*\*}

for i: =1 to n do s [i]: =0;

for i: =1 to n do

for j: =1 to n do

s [i]: =s [i] +z [i,j] \*y [j] ;

writeln ('Вывод результатов решения системы уравненй матричным способом');

for i: =1 to n do write (' ', s [i]: 5: 2);

end;

begin {\*\*\*\*\* тело программы \*\*\*\*\*\*}

clrscr;

writeln ('ввод матрицы коэффициентов при неизвестных х');

for i: =1 to N do

for j: =1 to N do

begin

write (' введите a [', i,',',j,'] => ');

read (a [i,j]);

end;

writeln ('ввод столбца свободных членов');

for i: =1 to N do

begin

write (' введите b [', i,'] => ');

read (b [i]);

end;

writeln ('введите вариант ');

writeln (' 1 - решение системы линейных уравнений методом Гаусса ');

write (' 2 - решение системы линейных уравнений матричным методом => ');

readln (variant);

case variant of

1: gauss (a,b,x,n);

2: matrica (a,b,n);

else writeln ('неверно указан вариант');

end;

end.