***Образец***

**Лабораторная работа 3**

**Нахождение решения системы линейных уравнений через обратную матрицу.**

**Студент гр Б22-544 Иванов П.Р.**

**1. Задание**

Для случайных систем размерности 3, 7 найти обратную матрицу, добавив к ней единичную той же размерности и выполнив над расширенной матрицей преобразования метода Гаусса-Жордано. Протестировать обратную матрицу, которая сформируется на месте добавленной единичной матрицы (). Затем найти решение системы через эту обратную матрицу (). Вектор сформировать дополнительно случайным образом. Далее протестировать полученное решение, ,найдя для него невязку ().

**2. Теория**

Полное преобразование матрицы методом оптимального исключения может быть записано в матричном виде

,

из которого следует матричная форма нахождения обратной матрицы (т.к. по определению )

 (1)

где - матрица перестановок строк при  - ом преобразовании, а  имеет вид



где .

Согласно выражению (1), если исходную матрицу  дополнить справа единичной матрицей  размерности , то в качестве алгоритма нахождения обратной матрицы можно использовать алгоритм Гаусса-Жордано, заменив в нем правые пределы циклов со значения  на . При этом ту же замену необходимо выполнить и в процедуре выбора ведущего элемента при перестановке строк. После  преобразований на месте единичной матрицы будет располагаться обратная матрица.

**Алгоритм метода Гаусса-Жордано для расширенной матрицы**



Алгоритм процедуры выбора ведущего элемента



**3. Программа**

/\* находит обратную матрицу на месте

дополнительной единичной матрицы методом Гаусса-Жордано \*/

kill(all);

/\* находим максимальный элемент в подстолбце матрицы, \*/

/\* переставляем строки и вычисляем определитель \*/

maxElem(A,k):=block

(

[buf,Amax,kmax,i,j,n],

n:length(A),

Amax:abs(A[k,k]),kmax:k,

if k<n then

(

for i:k+1 thru n do

(

buf:abs(A[i,k]),

if buf>Amax then (Amax:buf,kmax:i)

)

),

if Amax<1.e-10 then return(det:0),

if kmax#k then

(

for j:k thru n+n do

(

buf:A[k,j],A[k,j]:A[kmax,j],A[kmax,j]:buf

),

det:-det, m:m+1

),

det:det\*A[k,k]

);

/\* приводим матрицу к единичному виду \*/

conv\_Matr(A):=block

(

[i,j,n,buf],

n:length(A),

rang:0,det:1,m:0, /\* ранг, определитель, число перестановок \*/

for k thru n do

(

maxElem(A,k),

if abs(det)=0 then return(det),

rang:rang+1,

for j:n+n thru k step -1 do

(

A[k,j]:A[k,j]/A[k,k]

),

for i thru n do

(

if i#k then

(

for j:n+n thru k step -1 do

(

A[i,j]:A[i,j]-A[i,k]\*A[k,j]

)

)

)

)

);

/\* главная программа \*/

numer:true;

fpprintprec:3;

n:3;

/\* задаем расширенную систему случайным образом \*/

AE:zeromatrix(n,n+n); A:zeromatrix(n,n); Aobr:zeromatrix(n,n);

for i thru n do for j thru n do(

AE[i,j]:(0.5-random(1.0)),

A[i,j]:AE[i,j]

);

/\* формируем дополнительную единичную подматрицу \*/

for i thru n do AE[i,n+i]:1;

print("Исходная матрица = ",A);

print("Исходная расширенная матрица = ",AE);

conv\_Matr(AE);

print("Определитель матрицы = ",det);

print("Ранг матрицы = ",rang);

print("Число перестановок строк = ",m);

print("Приведенная расширенная матрица = ",AE);

for i thru n do for j thru n do Aobr[i,j]:AE[i,n+j];

print("Обратная матрица = ",Aobr);

print("Тест для обратной матрицы ",Aobr.A);

print("Обратная матрица через команду invert ",invert(A));

/\* использование обратной матрицы для поиска решения СЛУ \*/

b:zeromatrix(n,1); x:zeromatrix(n,1); dx:zeromatrix(n,1);

/\* задаем правую часть СЛУ \*/

for i thru n do b[i,1]:(0.5-random(1.0));

print("Правая часть системы ",b);

/\* находим решение через ранее полученную обратную матрицу \*/

for i thru n do for j thru n do x[i,1]:x[i,1]+Aobr[i,j]\*b[j,1];

print("Решение ",x);

for i thru n do(

dx[i,1]:b[i,1],

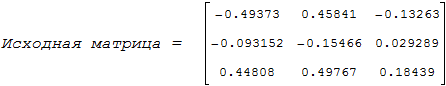
for j thru n do dx[i,1]:dx[i,1]-A[i,j]\*x[j,1]

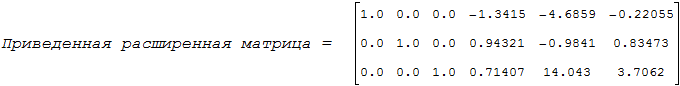
);

print("Невязка dx=b-Ax ",dx);

**4. Результаты**

**Для n=3**

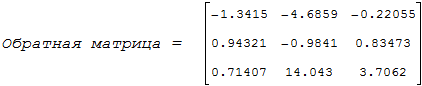
****

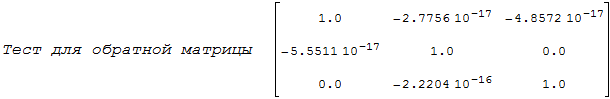
****

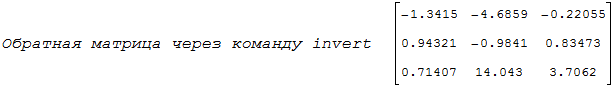
****

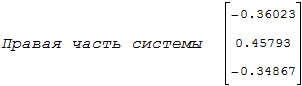
****

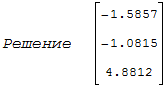
****

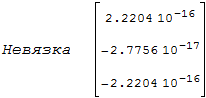
****

****

****

****

****

****  dx=b-Ax

**Для n=7**

Аналогично

**5. Выводы**

Что изменилось?..........

**Литература**

1. Козин Р.Г. Алгоритмы численных методов линейной алгебры и их программная реализация. М.: НИЯУ МИФИ, 2019. – 252 с.

2. …….