***Образец***

**Лабораторная работа 4**

**Сравнение решений системы линейных уравнений методом Гаусса и первым методом ортогонализации.**

**Студент гр Б22-544 Иванов П.Р.**

**1. Задание**

Для случайных систем размерности 10, 20 найти это решения методами Гаусса с процедурой выбора ведущего элемента первым методом ортогонализации. Сравнить различие этих решений (набрать статистику, проведя 5 или 10 прогонов в обоих случаях).

**2. Теория**

Алгоритм метода Гаусса уже обсуждался в лабораторной работе 2, поэтому здесь он опущен. Приведем формулы и алгоритмы только для метода ортогонализации.

Представим систему  в следующем виде

 (2.33)

где .

Если ввести -мерные вектора  и , то эти уравнения можно заменить системой скалярных произведений

 (2.34)

Соотношения (2.34) позволяют дать другую формулировку для исходной задачи: в -мерном пространства  найти -мерный вектор  с компонентом , перпендикулярный -мерной гиперплоскости, «натянутой» на вектора . Если удастся построить такой вектор, то первые  компонент этого вектора будут являться компонентами решения исходной системы уравнений.

Предлагается следующая схема решения этой задачи:

1. К исходной расширенной матрице добавляем снизу строку с компонентами (0,0,…,1). Если матрица исходной системы невырожденная, то строки этой новой матрицы в пространстве  образуют полную линейно-независимую систему векторов. Это следует из того обстоятельства, что ее определитель равен определителю исходной матрицы, который для невырожденной системы отличен от нуля.

2. Применяя к указанной полной линейно-независимой системе векторов процедуру ортогонализации Грама-Шмидта (2.35), строим ортонормированный базис для пространства  (см. рис. 2.17).



Рис. 2.17. Графическое пояснение операции процедуры Грамма-Шмидта

 (2.35)

Последний орт этой системы (последняя строка расширенной матрицы) будет перпендикулярен  - мерной гиперплоскости, «натянутой» на вектора .

Детализируем отдельные операции алгоритма процедуры ортогонализации:

- расчет скалярного произведения 



- нормировка вектора 



Нулевое значение подкоренного выражения означает линейную зависимость системы исходных векторов, т.е. вырожденность матрицы системы .

- корректировка вектора 



3. Поделив последнюю строку матрицы на ее последнюю компоненту, получим решение исходной системы

 (2.36)

*Замечание.* Алгоритм (2.36) необходимо поставить вместо последнего оператора в схеме (2.35), которая является общей записью алгоритма ортогонализации.

**3. Программа**

/\* Лабораторная работа 4 \*/

/\* первый метод ортогонализации решения Ax=b \*/

kill(all);

/\* находим скслярное произведение 2-х векторов \*/

sk(i,k):=block(

[j],

s:0,

for j thru n+1 do(

s:s+A[i,j]\*A[k,j]

),

return(s)

);

/\* процедура ортогонализации Грамма-Щмидта \*/

prog\_GSH(A):=block(

[k,i,j,norm,sk\_ik],

for k thru n do(

/\* нормировка к-ой строки \*/

norm:sk(k,k),

if norm<1.e-10 then return(1) else norm:sqrt(norm),

for j thru n+1 do A[k,j]:A[k,j]/norm,

/\* корректировка i-ых строк \*/

for i:k+1 thru n+1 do(

sk\_ik:sk(i,k),

for j thru n+1 do A[i,j]:A[i,j]-sk\_ik\*A[k,j]

),

/\* нормировка последней строки на A[n+1,n+1] \*/

for j thru n+1 do A[n+1,j]:A[n+1,j]/A[n+1,n+1]

),

return(0)

);

/\* главная программа \*/

numer:true;

fpprintprec:5;

n:5;

/\* задаем систему Ax=b случайным образом \*/

A:zeromatrix(n,n); b:zeromatrix(n,1); b1:zeromatrix(n,1);

for i thru n do(

b[i,1]:(0.5-random(1.0)), b1[i,1]:-b[i,1],

for j thru n do A[i,j]:(0.5-random(1.0))

);

print("Правая часть системы b и -b",b,b1);

/\* сохраняем исходную матрицу \*/

B:copy(A);

/\* Формируем дополнительную нижнюю строку матрицы \*/

c:zeromatrix(1,n+1); c[1,n+1]:1;

/\* формируем расширенную матрицу \*/

A:addcol(A,b1); A:addrow(A,c);

print("Сформированная матрица ",A);

code:prog\_GSH(A);

if code=1 then (

print("Матрица вырожденная !"),return(1)

);

print("Преобразованная матрица ",A);

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

for j thru n do x[j,1]:A[n+1,j];

print("Решение системы ",x);

/\* расчитывам невязку \*/

for i thru n do(

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

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

);

print("Невязка dx=b-Ax ",dx);

Ax:B.x;

dx1:zeromatrix(n,1);

for i thru n do dx1[i,1]:b[i,1]-Ax[i,1];

dx1;

/\* !!! добавляем метод Гаусса !!! \*/

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

/\* переставляем строки и вычисляем определитель \*/

maxElem(A,k):=block

(

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

n:length(A),

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

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+1 do

(

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

),

det:-det, m:m+1 /\* подсчитываем число перестановок \*/

),

det:det\*A[k,k]

);

/\* приводим матрицу к треугольному 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-1 do

(

maxElem(A,k),

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

rang:rang+1,

for i:k+1 thru n do

(

buf:A[i,k]/A[k,k],

for j:k thru n+1 do A[i,j]:A[i,j]-buf\*A[k,j]

)

),

if abs(A[n,n])<1.e-10 then det:0 else (det:det\*A[n,n],rang:rang+1)

);

/\* находим решение \*/

fine\_X(A,x):=block

(

[i,j,n],

n:length(A),

x[n,1]:A[n,n+1]/A[n,n],

for i:n-1 thru 1 step -1 do

(

x[i,1]:A[i,n+1],

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

x[i,1]:x[i,1]/A[i,i]

)

);

x1:zeromatrix(n,1);

D:addcol(B,b); /\* строим расширенную матрицу \*/

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

conv\_Matr(D);

print("Определитель матрицы = ",det);

print("Ранг матрицы = ",rang);

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

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

fine\_X(D,x1);

print("Решение системы = ",x1);

/\* Разность результатов решений системы двумя методами \*/

ddx:zeromatrix(n,1);

for i thru n do ddx[i,1]:x[i,1]-x1[i,1];

print("Разность двух решений ",ddx);

normKub:0;

for i thru n do(

buf:abs(ddx[i,1]),

if buf>normKub then normKub:buf

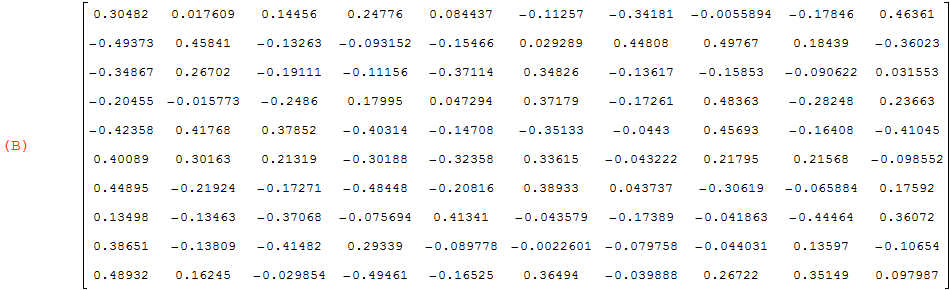
);

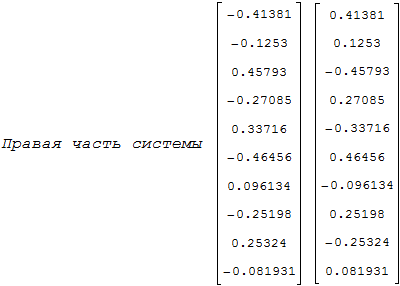
print("Кубическая норма вектора разности решений ",normKub);

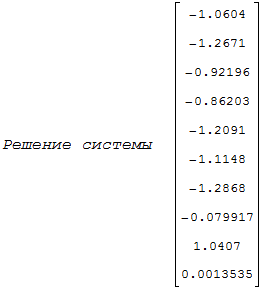
**4. Результаты**

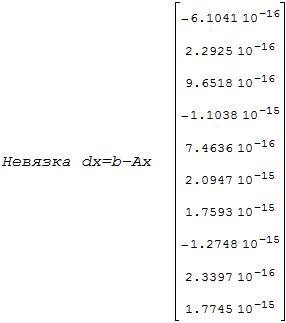
**Для n=10**

Исходная матрица системы

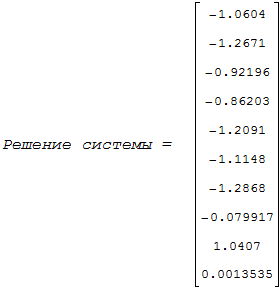


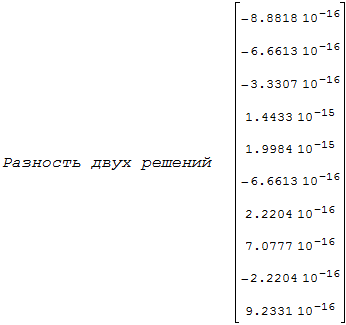






Метод Гаусса







**Для n=20**

Привести только результаты.

**5. Выводы**

**Литература**

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