Министерство образования Российской Федерации

Тамбовский государственный технический университет

Кафедра

**Отчёт**

**по лабораторной работе №2**

по предмету:

**«Вычислительная математика»**

**Выполнил:**

студент группы

**Проверил:**

Тамбов 20 г.

## Лабораторная работа №2

**«Исследование методов решений систем линейных уравнений»**

**1.Постановка задачи.**

Написать программу решения СЛАУ методом Гаусса и методом релаксаций

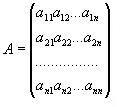
**2.Описание методов изучения СЛАУ и ее решения.**

**Метод Гаусса.**

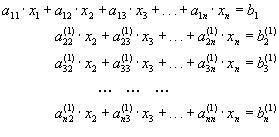
Пусть надо решить СЛАУ вида:

Vichsl17

или Ах=в

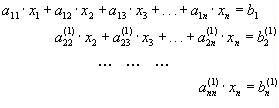
, Vichsl19, Vichsl20

Предположим, что матрица А - невырожденная, т.е. det А не равно 0. В этом случае решение системы существует и оно единственно, а рассматриваемая задача коректна. Вычисления с помощью метода Гаусса состоят из двух шагов, называемых прямым и обратным ходом. Прямой ход метода Гаусса заключается в последовательном исключении неизвестных из системы с верхней треугольной матрицей. Вычисления значений неизвестных производят на этапе обратного хода. Рассмотрим простейший вариант метода Гаусса, называемый схемой единственного деления.Прямой ход состоит из (n-1) шагов исключения. 1-й шаг. Целью этого шага является исключение неизвестного x1 из уравнений с номерами i = 2,3,...,n. Предположим, что коэффициент a11  0 (главный элемент первого шага) Вычислим величины i1=ai1 /a11(i=2,3,...,n), называемые множителями 1-го шага. Вычтем из второго, третьего и ... до n-го уравнений системы (1) первое уравнение, умноженное соответственно на  21,31,..., n1.Это позволит обратить в нуль коэффициенты при х1 во всех уравнениях , кроме первого. В результате получим эквивалентную систему.



в которой aij(1)=aij-i1 aij , bi(1)=bi-i1 b1.

2-й шаг. На этом шаге производится исключение х2 из уравнений с номерами i = 3, 4, . . . , n. Вычислим i2=ai2(1)/ a22(1) относительно главного элемента 2-го шага, после чего произведем те же действия по исключению элементов аi2 из 3-й n . . . строк. Аналогично проводятся остальные шаги. k-й шаг. Предположим, что главный элемент k-го шага akk(k-1)0, вычислим множители k-го шага ik= aik(k-1)/akk(k-1) , (i=k+1,...,n) и вычтем последовательно из (k + 1)-го , . . . , n-го уравнений полученной на предыдущем шаге системы k-е уравнение, умноженные соответственно на  k+1,k+2,..., nk. После (n -1) - го шага исключения получим систему уравнений

(2)

Получается матрица А(n-1), которая является верхней треугольной. На этом вычисления прямого хода заканчиваются. Обратный ход. Из последнего уравнения системы (2) вычислим хn. Подставляя полученное значение в предпоследнее уравнение, вычислим значение хn-1. Таким образом можно вычислить значения всех неизвестных. Вычисления здесь проводятся по формулам:

Vichsl30

Vichsl31

### Метод релаксации.

Имеем систему:

\left\{
\begin{matrix}
a_{11}x_1 + \ldots + a_{1n}x_n & = & b_1\\
a_{21}x_1 + \ldots + a_{2n}x_n & = & b_2\\
 & \ldots & \\
a_{n1}x_1 + \ldots + a_{nn}x_n & = & b_n\\
\end{matrix}
\right.

Преобразуем ее к виду:

\left\{
\begin{matrix}
b_{11}x_1 + b_{12}x_2 + \ldots + b_{1n}x_n + c_1 & = & 0\\
 & \ldots & \\
b_{n1}x_1 + b_{n2}x_2 + \ldots + b_{nn}x_n + c_n & = & 0\\
\end{matrix}
\right.

где b_{ij} = -\frac{a_{ij}}{a_{ii}}, c_i = \frac{b_i}{a_{ii}}

Зададим начальное приближение:

x(0)=(x1(0), x2(0),...,xn(0)) и подставим в полученную систему.

Получаем невязки (отклонения).

\left\{
\begin{matrix}
R_1^{(0)} & = & c_1 - x_1^{(0)} + \sum \limits_{j = 2}^n b_{1j}x_j^{(0)}\\
R_2^{(0)} & = & c_2 - x_2^{(0)} + \sum \limits_{j = 1, j \neq 2}^n b_{2j}x_j^{(0)}\\
 & \ldots & \\
R_n^{(0)} & = & c_n - x_n^{(0)} + \sum \limits_{j = 1}^{n - 1} b_{nj}x_j^{(0)}\\
\end{matrix}
\right.

Если одной из неизвестных xs(0) задать приращение  xs(0), то соответствующая невязка уменьшится на эту величину, а все остальные невязки Ri(0)(iS) увеличатся на величину bis xs(0).то есть, чтобы обратить очередную невязку Rs(1) в нуль, необходимо величине xs(0) дать приращение  xs(0)= Rs(0) и получим Rs(1)=0 и

Ri(1)= Ri(0) + bis xs(0).

Таким образом, идея метода состоит в том, чтобы на каждом шаге обращать в нуль максимальную по модулю невязку путем изменения значения соответствующей компоненты приближения. Процесс заканчивается, когда все невязки последней преобразованной системы будут равны нулю с заданной точностью.

Ответ находится по [формуле](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D1%83%D0%BB%D0%B0):

x_i \approx x_i^{(0)} + \sum_j \delta x_i^{(j)}

**3.Текст программы на языке С.**

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <math.h>

void gauss()

{

int n,i,j,k,flag=0,flag1=0;

float \*\*matr,\*mas,\*X,m,sum;

printf("Metod Gaussa\n");

printf("------------\n");

printf("Vvedite chislo uravneniy v sisteme: "); scanf("%d",&n);

matr=(float\*\*)malloc(n\*sizeof(float\*));

for(i=0;i<n;i++) matr[i]=(float\*)malloc(n\*sizeof(float));

mas=(float\*)malloc(n\*sizeof(float));

X=(float\*)malloc(n\*sizeof(float));

printf("\n");

printf("Vvedite koeffitsientui: \n") ;

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

{

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

{

printf("a[%d , %d] = ",j+1,i+1);

scanf("%f",&matr[i][j]);

}

printf("b[%d] = ",i+1);

scanf("%f",&mas[i]);

}

printf("\nIskhodnaya sistema:\n");

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

{

for (j=0;j<n;j++) printf("%+.3f\*x%d",matr[i][j],j+1);

printf("=%.3f\n",mas[i]);

}

//pryamoy hod

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

{

if (flag1==1) break;

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

{

if (matr[k-1][k-1]==0) {flag1=1; break;}

m=matr[i][k-1]/matr[k-1][k-1];

mas[i]=mas[i]-mas[k-1]\*m;

for(j=k-1;j<n;j++) matr[i][j]=matr[i][j]-m\*matr[k-1][j];

}

}

if (flag1==0)

{

printf("\nSistema s verkhney treugol'noy matritsey:\n");

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

{

for (j=0;j<n;j++) printf("%+.3f\*x%d",matr[i][j],j+1);

printf("=%.3f\n",mas[i]);

}

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

if ((matr[i][i]>-0.01)&&(matr[i][i]<0.01)) {flag=1; break;}

if (flag==0)

{

//obratny khod

for (i=0;i<n;i++) X[i]=0;

k=n;

for (i=n-1;i>=0;i--)

{

sum=0;

k--;

for (j=0;j<n;j++) if (j!=k) sum=sum-matr[i][j]\*X[j];

X[k]=(sum+mas[i])/matr[i][i];

}

printf("\nResheniya:\n");

for (i=0;i<n;i++) printf("x%d=%.3f\n",i+1,X[i]);

} else printf("\nSistema ne imeet edinstvennogo resheniya,na glavnoy diagonali\nvstretilsya 0!");

for(i=0;i<n;i++) free(matr[i]);

free(matr);

free(mas);

free(X);

} else printf("\nReshenie sistemy metodom Gaussa nevozmozhno,na glavnoy diagonali\nvstretilsya 0!");

return;

}

void relaksia()

{

int n,i,j,t,k;

float \*\*a,\*b,\*x1,\*r,\*r1,m,tg=0,t1;

clrscr();

printf("Metod relakcia\n");

printf("------------\n");

printf("Vvedite chislo uravneniy v sisteme: "); scanf("%d",&n);

a=(float\*\*)malloc(n\*sizeof(float\*));

for(i=0;i<n;i++) a[i]=(float\*)malloc(n\*sizeof(float));

b=(float\*)malloc(n\*sizeof(float));

x1=(float\*)malloc(n\*sizeof(float));

r=(float\*)malloc(n\*sizeof(float));

r1=(float\*)malloc(n\*sizeof(float));

printf("\n");

printf("Vvedite koeffitsientui: \n") ;

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

{

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

{

printf("a[%d , %d] = ",i+1,j+1);

scanf("%f",&a[i][j]);

}

printf("b[%d] = ",i+1);

scanf("%f",&b[i]);

}

printf("\nIskhodnaya sistema:\n");

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

{

for (j=0;j<n;j++) printf("%+.3f\*x%d",a[i][j],j+1);

printf("=%.3f\n",b[i]);

}

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

{

t1=a[i][i];

b[i]=b[i]/t1;

for(j=0;j<n;j++) a[i][j]=-a[i][j]/t1;

}

printf("\nsistema posle izmenhenia:\n");

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

{

for (j=0;j<n;j++) printf("%+.3f\*x%d",a[i][j],j+1);

printf(" %.3f\n",b[i]);

}

printf("\n\zadaite npervoe znachenhie:");

float tt=0;

for(i=0;i<n;i++) scanf("%f:",&r1[i]);

for(i=0;i<n;i++) r[i]=0;

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

{ for (j=0;j<n;j++)

{ if(i!=j) tt+=a[i][j]\*r1[j];}

r[i]=b[i]-r1[i]+tt;

tt=0;

}

printf("\n\nzadaite yclovie octanofki:"); scanf("%f:",&m);

printf("\n\n");

for(i=0;i<n;i++) x1[i]=0;

do

{

tg=0;

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

{ printf("r[%d]=%4.6f; ",i+1,r[i]);

if(r[i]>=tg) { tg=r[i];t=i;

}

}

printf("\n");

x1[t]=x1[t]+tg;

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

r[i]=r[i]+a[i][t]\*tg;

} while(tg>m);

printf("\n\npesenhie sictemui:");

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

printf("x1[%d]=%4.6f; ",i+1,x1[i]);

}

void main()

{

clrscr();

gauss();

getch();

clrscr();

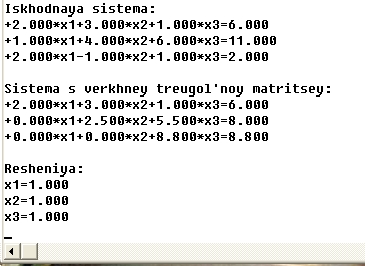
relaksia();

getch();

}

4.Результаты:

Метод гаусса:



Метод релаксаций:

