**Решение систем уравнений методом Гаусса**

1. Написать программу для реализации решения систем линейных уравнений вида AX=B методом Гаусса последовательного исключения для матрицы коэффициентов системы линейных уравнений.

**Решение:**

Программа написана на языке программирования C++, посредством Visual Studio 2017

**Код программы:**

#include "stdafx.h"

#include <iostream>

using namespace std;

// Вывод системы уравнений

void sysout(double \*\*a, double \*y, int n)

{

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

{

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

{

cout << a[i][j] << "\*x" << j;

if (j < n - 1)

cout << " + ";

}

cout << " = " << y[i] << endl;

}

return;

}

double \* gauss(double \*\*a, double \*y, int n)

{

double \*x, max;

int k, index;

const double eps = 0.00001; // точность

x = new double[n];

k = 0;

while (k < n)

{

// Поиск строки с максимальным a[i][k]

max = abs(a[k][k]);

index = k;

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

{

if (abs(a[i][k]) > max)

{

max = abs(a[i][k]);

index = i;

}

}

// Перестановка строк

if (max < eps)

{

// нет ненулевых диагональных элементов

cout << "Solution can not be obtained due to the null column ";

cout << index << " матрицы A" << endl;

return 0;

}

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

{

double temp = a[k][j];

a[k][j] = a[index][j];

a[index][j] = temp;

}

double temp = y[k];

y[k] = y[index];

y[index] = temp;

// Нормализация уравнений

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

{

double temp = a[i][k];

if (abs(temp) < eps) continue; // для нулевого коэффициента пропустить

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

a[i][j] = a[i][j] / temp;

y[i] = y[i] / temp;

if (i == k) continue; // уравнение не вычитать само из себя

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

a[i][j] = a[i][j] - a[k][j];

y[i] = y[i] - y[k];

}

k++;

}

// обратная подстановка

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

{

x[k] = y[k];

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

y[i] = y[i] - a[i][k] \* x[k];

}

return x;

}

int main()

{

double \*\*a, \*y, \*x;

int n;

system("chcp 1251");

system("cls");

cout << "Enter the number of equations: ";

cin >> n;

a = new double\*[n];

y = new double[n];

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

{

a[i] = new double[n];

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

{

cout << "a[" << i << "][" << j << "]= ";

cin >> a[i][j];

}

}

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

{

cout << "y[" << i << "]= ";

cin >> y[i];

}

sysout(a, y, n);

x = gauss(a, y, n);

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

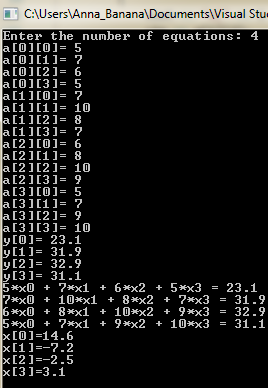
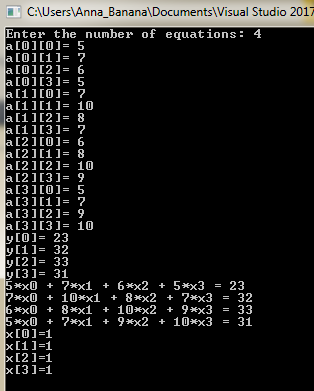
cout << "x[" << i << "]=" << x[i] << endl;

cin.get(); cin.get();

return 0;

}

**Результат выполнения программы:**



1. Написать программу для реализации решения систем линейных уравнений вида AX=B методом Гаусса оптимального исключения для матрицы коэффициентов системы линейных уравнений.

**Решение:**

Программа написана на языке программирования Pascal посредством Lazarus

**Код программы:**

program gauss\_opt\_is;

uses crt;

var

i,j,k,l,n : integer;

m,z : real;

a:array[1..99,1..100]of real;

procedure Out;

begin

for i:=1 to n do

begin

for j:=1 to n+1 do write(a[i,j]:4:4,' ');

writeln;

end;

end;

begin

clrscr;

write('Razmer matrix N = ');readln(n);

writeln('Vvodite po strokam');

for i:=1 to n do

begin

writeln(i,'-ia stroka');

for j:=1 to n+1 do readln(a[i,j]);

end;

clrscr;

writeln('Nachal6naia matrix : ');out;

for i:=1 to n do

begin

z:=a[i,i];

for j:=i to n+1 do a[i,j]:=a[i,j]/z;

for l:=i+1 to n do

begin

z:=a[l,i];

for j:=i+1 to n+1 do a[l,j]:=a[l,j]-a[i,j]\*z;

a[l,i]:=0;

end;

for l:=i-1 downto 1 do

begin

z:=a[l,i];

for j:=i+1 to n+1 do a[l,j]:=a[l,j]-a[i,j]\*z;

a[l,i]:=0;

end;

end;

writeln;

writeln('Konechnaia matrix : ');out;

writeln;

writeln('Korni : ');

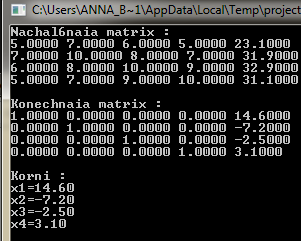
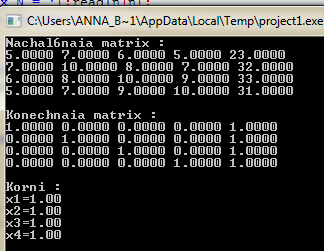
for i:=1 to n do

writeln ('x', i, '=', a[i,n+1]:2:2);

readkey;

end.

**Результат выполнения работы:**



1. Написать Написать программу для реализации решения систем линейных

уравнений вида AX=B методом Гаусса с выбором главного элемента для матрицы коэффициентов системы линейных уравнений

**Решение:**

Программа написана на языке программирования Pascal посредством Lazarus

**Код программы:**

Program Metod\_Gauss;

Uses CRT;

Const maxn = 10;

Type

Data = Real;

Vector = Array[1..maxn] of Data;

Matrix = Array[1..maxn, 1..maxn] of Data;

{ Процедура ввода расширенной матрицы }

Procedure ReadSystem(n: Integer; var a: Matrix; var b: Vector);

Var r, i, j: Integer;

Begin

r := WhereY;

GotoXY(2, r);

Write('A');

For i := 1 to n do begin

GotoXY(i\*6+2, r);

Write(i);

GotoXY(1, r+i+1);

Write(i:2);

end;

GotoXY((n+1)\*6+2, r);

Write('b');

For i := 1 to n do begin

For j := 1 to n do begin

GotoXY(j \* 6 + 2, r + i + 1);

Read(a[i, j]);

end;

GotoXY((n + 1) \* 6 + 2, r + i + 1);

Read(b[i]);

end;

End;

{ Процедура вывода результатов }

Procedure WriteX(n :Integer; x: Vector);

Var i: Integer;

Begin

For i := 1 to n do

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

End;

{ Функция, реализующая метод Гаусса }

Function Gauss(n: Integer; a: Matrix; b: Vector; var x:Vector): Boolean;

Var

k, l, i, j: Integer;

m, q, t: Data;

Begin

For k := 1 to n - 1 do begin

{ Ищем строку l с максимальным элементом в k-ом столбце}

m := 0;

l := 0;

For i := k to n do

If Abs(a[i, k]) > m then begin

l := i;

m := Abs(a[i, k]);

end;

{ Если у всех строк от k до n элемент в k-м столбце нулевой,

то система не имеет однозначного решения }

If l = 0 then begin

Gauss := false;

Exit;

end;

{ Меняем местами l-ую и k-ую строки }

If l <> k then begin

For j := 1 to n do begin

t := a[k, j];

a[k, j] := a[l, j];

a[l, j] := t;

end;

t := b[k];

b[k] := b[l];

b[l] := t;

end;

{ Преобразуем матрицу }

For i := k + 1 to n do begin

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

For j := 1 to n do

If j = k then

a[i, j] := 0

else

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

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

end;

end;

{ Вычисляем решение }

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

For i := n - 1 downto 1 do begin

t := 0;

For j := 1 to n-i do

t := t + a[i, i + j] \* x[i + j];

x[i] := (1 / a[i, i]) \* (b[i] - t);

end;

Gauss := true;

End;

Var n, i: Integer;

a: Matrix ;

x, b: Vector;

Begin

ClrScr;

Writeln('Программа решения СЛАУ по методу Гаусса');

Writeln;

Writeln('Введите порядок матрицы системы (макс. 10)');

Repeat

Write('>');

Read(n);

Until (n > 0) and (n <= maxn);

Writeln;

Writeln('Введите расширенную матрицу системы');

ReadSystem(n, a, b);

Writeln;

If Gauss(n, a, b, x) then begin

Writeln('Результат вычислений по методу Гаусса');

WriteX(n, x);

readln;

end

else

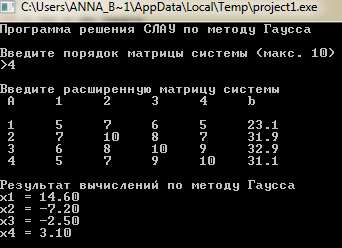
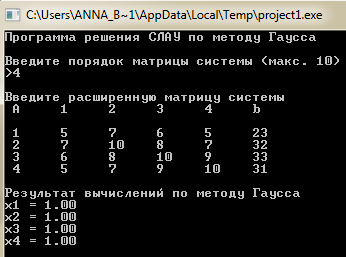
Writeln('Данную систему нельзя решить по методу Гаусса');

Writeln;

readln;

End.

**Результат выполнения программы:**



1. Написать программу для реализации решения систем линейных уравнений вида AX=B методом Гаусса - Жордана для матрицы коэффициентов системы линейных уравнений.

**Решение:**

Программа написана на языке программирования C++, посредством Visual Studio 2017

**Код программы:**

#include "stdafx.h"

#include<iostream>

#include<cmath>

#include<cstdlib>

#include <stdio.h>

#include <conio.h>

int main()

{

int i, j, k, n;

float A[20][20], c, x[10];

printf("\nEnter the size of matrix: ");

scanf("%d", &n);

printf("\nEnter the elements of augmented matrix row-wise:\n");

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

{

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

{

printf(" A[%d][%d]:", i, j);

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

}

}

/\* Now finding the elements of diagonal matrix \*/

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

{

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

{

if (i != j)

{

c = A[i][j] / A[j][j];

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

{

A[i][k] = A[i][k] - c\*A[j][k];

}

}

}

}

printf("\nThe solution is:\n");

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

{

x[i] = A[i][n + 1] / A[i][i];

printf("\n x%d=%f\n", i, x[i]);

}

getch();

return(0);

}

**Результат выполнения работы:**

