Міністерство освіти і науки України

Західноукраїнський національний університет

Факультет комп’ютерних інформаційних технологій

Кафедра комп’ютерних наук

**Лабораторна робота №3**

з дисципліни «Інтервальні обчислення»

на тему:

«Програма система для обчислення інтервальних систем лінійних алгебричних рівнянь»

Виконав:

студент групи ІПЗм – 12

Тимчишин Богдан

Перевірила:

Порплиця Н.П.

Тернопіль – 2020

**Завдання**

Розробити програмну систему для обчислення інтервальних систем лінійних алгебричних рівнянь за методом Гауса та LU-розкладу. У програмній системі повинно бути передбачено:

* зручний графічний інтерфейс;
* перевірка на коректність введених даних;
* ввід розмірності інтервальних систем (системи 2-х, 3-х та 4-х рівнянь);
* відповідно до введеної розмірності матриці, користувачеві повинна надаватись можливість проставити інтервальні коефіцієнти (матрицю А) та інтервальні виходи (матрицю В) системи:

а) можна проставляти коефіцієнти одразу інтервалами;

б) можна проставляти коефіцієнти числом і окремо зазначити відсоток відхилення.

- можливість обрання методу обчислення;

- вивід спрощуваних під час обчислень матриць А та В (метод Гауса) та матриць L, U та у (метод LU-розкладу), а такожрезультуючого вектора х.



 на параметри на вихід

Лістинг програми:

using Lr\_1\_kv.Functions;

using Lr\_1\_kv.Models;

usingSystem;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows;

usingSystem.Windows.Controls;

usingSystem.Windows.Data;

usingSystem.Windows.Documents;

usingSystem.Windows.Input;

usingSystem.Windows.Media;

usingSystem.Windows.Media.Imaging;

usingSystem.Windows.Shapes;

namespace Lr\_1\_kv

{

/// <summary>

/// Логикавзаимодействия для LU.xaml

/// </summary>

publicpartialclass LU : Window

{

intrang = 3;

TextBox[,] tb;

publicPromizok[,] inpArr;

public LU()

{

InitializeComponent();

inpArr = newPromizok[,] {

{ newPromizok(0.985,1.015),newPromizok(-1.015,-0.985),newPromizok(-2.03,-1.97),newPromizok(-1.08,-0.92) },

{ newPromizok(2.955,3.045),newPromizok(-2.03,-1.97),newPromizok(3.94,4.06),newPromizok(9.2, 10.8) },

{ newPromizok(-1.015,-0.985),newPromizok(2.955,3.045),newPromizok(-2.03,-1.97),newPromizok(-5.4,-4.6) }

};

drowMatrix();

matrixToTB();

//Gaus();

}

publicvoidluRorp()

{

LtextBlock.Text = "";

UtextBlock.Text = "";

YtextBlock.Text = "";

XtextBlock.Text = "";

Promizok[,] L = zeros(rang, 'l');

Promizok[,] U = zeros(rang, 'u');

for (int i = 1; i <rang; i++)

{

for (int j = i; j <rang; j++)

{

Promizokcount = newPromizok(0, 0);

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

{

count = Function.Summ(count, Function.Dobutok(L[j, k], U[k, i]));

}

L[j, i] = Function.Minnus(inpArr[j, i], count);

}

if (i + 1 <rang)

{

for (int j = i + 1; j <rang; j++)

{

Promizokcount = newPromizok(0, 0);

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

{

count = Function.Summ(count, Function.Dobutok(L[i, k], U[k, j]));

}

count = Function.Minnus(inpArr[i, j], count);

U[i, j] = Function.Chastka(count, L[i, i]);

}

}

}

Promizok[] y = newPromizok[rang];

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

{

Promizokcount = newPromizok(0, 0);

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

{

count = Function.Summ(count, Function.Dobutok(L[n, k], y[k]));

}

count = Function.Minnus(inpArr[n, rang], count);

y[n] = Function.Chastka(count, L[n, n]);

}

Promizok[] x = newPromizok[rang];

for (int i = rang - 1; i >= 0; i--)

{

Promizokcount = newPromizok(0, 0);

for (int m = i + 1; m <rang; m++)

{

count = Function.Summ(count, Function.Dobutok(U[i, m], x[m]));

}

x[i] = Function.Minnus(y[i], count);

}

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

{

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

{

LtextBlock.Text += StringParser.promizokToString(L[i, j]) + " ";

UtextBlock.Text += StringParser.promizokToString(U[i, j]) + " ";

}

LtextBlock.Text += "\n";

UtextBlock.Text += "\n";

XtextBlock.Text += "x(" + i + ")= " + StringParser.promizokToString(x[i]) + "\n";

YtextBlock.Text += "y(" + i + ")= " + StringParser.promizokToString(y[i]) + "\n";

}

}

publicvoidGaus()

{

Promizok[,] A = newPromizok[rang, rang];

Promizok[] B = newPromizok[rang];

AGaus.Text = "";

BGaus.Text = "";

AGaus\_Copy.Text = "";

BGaus\_Copy.Text = "";

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

{

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

{

if (i == 0 || j == 0)

{

A[i, j] = newPromizok(0, 0);

}

else

{

A[i, j] = Function.Minnus(inpArr[i, j], Function.Dobutok(inpArr[0, j], Function.Chastka(inpArr[i, 0], inpArr[0, 0])));

}

AGaus.Text += StringParser.promizokToString(A[i, j]) + " ";

}

AGaus.Text += "\n";

}

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

{

if (i > 0)

{

B[i] = Function.Minnus(inpArr[i, rang], Function.Dobutok(inpArr[0, rang], Function.Chastka(inpArr[i, 0], inpArr[0, 0])));

}

else

{

B[i] = inpArr[i, rang];

}

BGaus.Text += "B(" + i + ")= " + StringParser.promizokToString(B[i]) + "\n";

}

Promizok[,] Astr = newPromizok[rang, rang];

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

{

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

{

if (i == 0)

{

Astr[i, j] = inpArr[i,j];

}

else

if (i > j)

{

Astr[i, j] = newPromizok(0, 0);

}

else

if(i == 1)

{

Astr[i, j] = A[i, j];

}

else

{

Astr[i, j] = Function.Minnus(A[i, j], Function.Dobutok(A[i - 1, j], Function.Chastka(A[i, j - 1], A[i - 1, j - 1])));

}

AGaus\_Copy.Text += StringParser.promizokToString(Astr[i, j]) + " ";

}

AGaus\_Copy.Text += "\n";

}

Promizok[] X = newPromizok[rang];

for (int i = rang - 1; i >= 0; i--)

{

Promizokcount = newPromizok(0, 0);

for (int m = i + 1; m <rang; m++)

{

count = Function.Summ(count, Function.Dobutok(Astr[i, m], X[m]));

}

count = Function.Minnus(B[i], count);

X[i] = Function.Chastka(count, Astr[i, i]);

BGaus\_Copy.Text += "X(" + i + ")= " + StringParser.promizokToString(X[i]) + "\n";

}

}

privatevoidbutton\_Click(objectsender, RoutedEventArgs e)

{

tbToMatrix();

Gaus();

luRorp();

}

publicPromizok[,] zeros(int x, chartype)

{

Promizok[,] ResMatr = newPromizok[x, x];

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

{

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

{

if (type == 'l')

{

if (j > i)

{

ResMatr[i, j] = newPromizok(0, 0);

}else

if (j == 0)

{

ResMatr[i, j] = inpArr[i, j];

}

} else

if (type == 'u')

{

if (j < i)

{

ResMatr[i, j] = newPromizok(0, 0);

}

elseif (j == i)

{

ResMatr[i, j] = newPromizok(1, 1);

}

elseif (i == 0)

{

ResMatr[i, j] = Function.Chastka(inpArr[0,j], inpArr[0,0]);

}

}

}

}

returnResMatr;

}

privatevoidradioButton\_Checked(objectsender, RoutedEventArgs e)

{

if (radioButton.IsChecked == true)

{

rang = 2;

} else

if (radioButton\_Copy.IsChecked == true)

{

rang = 3;

}

else

if (radioButton\_Copy1.IsChecked == true)

{

rang = 4;

}

matrix.Children.Clear();

drowMatrix();

}

publicvoiddrowMatrix()

{

tb = newTextBox[rang , rang + 1 ];

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

{

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

{

tb[i, j] = newTextBox()

{

Name = "tb" + i.ToString(),

Width = 75,

Height = 20,

HorizontalAlignment = HorizontalAlignment.Left,

VerticalAlignment = VerticalAlignment.Top,

Margin = newThickness(j \* 78, i \* 30, 0, 0)

};

matrix.Children.Add(tb[i, j]);

}

}

}

publicvoidmatrixToTB()

{

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

{

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

{

tb[i, j].Text = StringParser.promizokToString(inpArr[i,j]);

}

}

}

publicvoidtbToMatrix()

{

inpArr = newPromizok[rang, rang + 1];

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

{

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

{

inpArr[i, j] = StringParser.stringToPromizok(tb[i, j].Text);

}

}

}

}

}

На рисунку 1 зображено результат виконання програми:

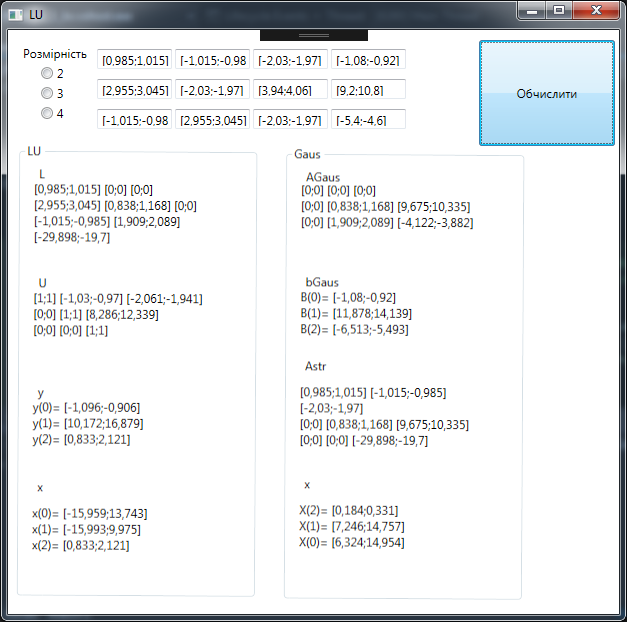


Рисунок 1 – Результат виконання програми

**Висновок:** під час виконання лабораторної роботи було розроблено програмну систему для обчислення інтервальних систем лінійних алгебричних рівнянь за методом Гауса та LU-розкладу.