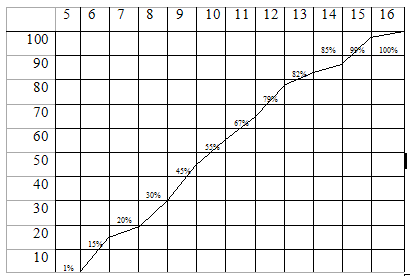
ФГБОУ ВПО

«Уфимский государственный авиационный технический университет»

**Кафедра Информатики**



|  |
| --- |
| Тема КР |
| Решение систем |
| линейных уравнений |

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

|  |  |
| --- | --- |
| **к курсовой работе по** | **информатике** |

|  |
| --- |
| 1308.300202.000ПЗ |
| (обозначение документа) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Группа |  |  | Фамилия, и., о. | Подпись | Дата | Оценка |
| ИВТ-109 |  |
|  |  |
| Студент | | | Вафин Р. Р. |  |  |  |
| Консультант | | | Смирнова Е.А. |  |  |  |
| Принял | | |  |  |  |  |

**Уфа 2015 г.**

**Содержание**

[1.1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА 1](#_Toc418421062)

[1 Введение 3](#_Toc418421063)

[1.1 Класс задач 3](#_Toc418421064)

[2 Описание метода 6](#_Toc418421065)

[2.1 Метод Гаусса 6](#_Toc418421066)

[2.2 Блок-схема 8](#_Toc418421067)

[2.4 Код программы 9](#_Toc418421068)

[3 Описание программы 11](#_Toc418421069)

[3.1 Общая структура программы 11](#_Toc418421070)

[3.2 Описание работы программы 12](#_Toc418421071)

[3.3 Form1 14](#_Toc418421072)

[3.4 VisForm 16](#_Toc418421073)

[3.5 Description\_form 19](#_Toc418421074)

[3.6 Flowchart\_form 19](#_Toc418421075)

[3.7 Ref\_form 20](#_Toc418421076)

[3.8 About\_form 21](#_Toc418421077)

[4 Вывод 22](#_Toc418421078)

[5 Используемая литература 24](#_Toc418421079)

[Приложение 25](#_Toc418421080)

# Введение

## Класс задач

Системы линейных алгебраических уравнений (СЛАУ) применяются в самых разных областях науки. Это может быть электротехника, механика, термодинамика, информатика и тд.

В моей практике СЛАУ встречается в физике (Правила Кирхгофа), в теоретических основах электротехники, ну и главным образом в математическом анализе, где СЛАУ одна из основных тем 1-го семестра.

Задачи на правила Кирхгофа – это задача на нахождение соотношения тока и напряжения на участках любой электрической цепи. С этими задачами сталкиваются на курсах общей физики, на теоретических основах электротехники, а так же те, кто самостоятельно разрабатывает электрические устройства. Применение правил Кирхгофа к линейной электрической цепи позволяет получить систему линейных уравнений относительно токов или напряжений, и соответственно, найти значение токов на всех ветвях цепи и все межузловые напряжения.

Так же довольно обширной областью применения СЛАУ является экономика. К таким типам задач можно отнести: прогноз и оценки экономики предприятий, организаций; прогноз курса валют, ценных бумаг, акций, и тд.

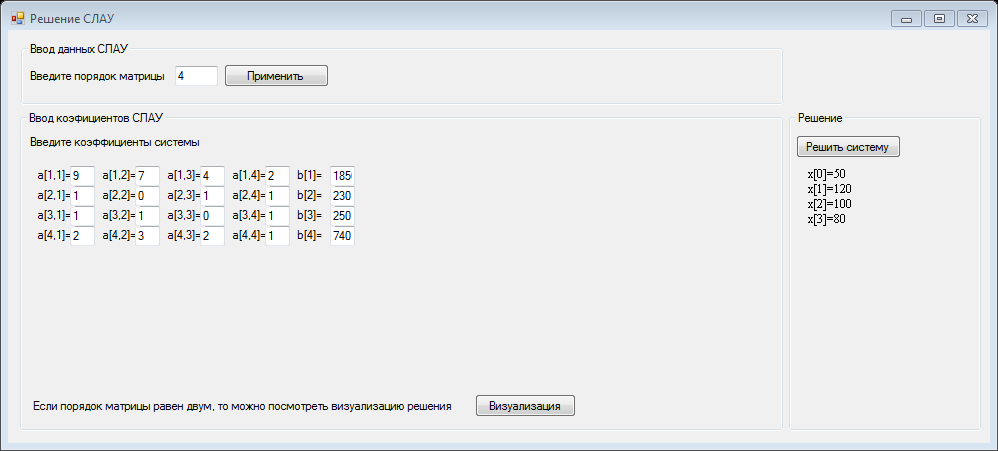
**Задача, связанная с экономикой.**

На предприятие с работниками четырех категорий привезли заработную плату в купюрах следующего достоинства: по 100 рублей – 1850 купюр, по 10 рублей – 250 купюр, 1 рублю – 740 купюр. Заработная плата работника 1-й категории составляет 962 руб., 2-й категории –713 руб., 3-й категории – 452 руб., 4-й категории – 261 руб. Определить, сколько сотрудников каждой категории работает на предприятии, если каждому сотруднику выдали заработную плату минимальным числом купюр.

Пусть *x*, *y*, *z*, *w* – количество работников категорий соответственно с первой по четвертую. Тогда решением данной задачи будет система уравнений:



И ее решение методом Гаусса: x=50, y=120, z=100, w=80



Прогноз погоды – так же является одной из целью применения СЛАУ. Обширные объемы данных (температура, давление, влажность и тд.) требуют большой вычислительной мощности, и для решения этих систем применяются наиболее эффективные алгоритмы решения СЛАУ, а так же большие вычислительные мощности.

Весь этот спектр задач предопределяет методы, необходимые для их решения. Так, к примеру, простые задачи на курсах физики или математического анализа требуют легкого, простого для понимания метода решения, (например метод Гаусса, метод Крамера), а задачи связанные с прогнозом погоды требуют большей эффективности, и вследствие сложных, но быстрых алгоритмов решения СЛАУ. (метод Халецкого, метод Якоби)

В данной работе я буду рассматривать метод Гаусса, как самый простой для использования и понимания, и быстрый для СЛАУ небольшого порядка. Но при порядке систем более 30-40 решения может занимать довольно продолжительное время, и вследствие этого, моя программа предназначения для решения СЛАУ только до 10-го порядка.

# Описание метода

## Общее описание СЛАУ

Система линейных уравнений обычно записывается в виде:

 (2.1)

В матричном виде система линейных уравнений записывается так:



где

## Метод Гаусса

Метод Гаусса (метод исключения) для решения систем линейных уравнений относится к точным методам. Идея метода Гаусса состоит в том, что система (2.1) путем последовательного исключения неизвестных приводится к системе с треугольной матрицей, из которой и определяются значения неизвестных.

Процесс исключения неизвестных состоит в следующем:

Пусть a11≠0. Разделим первое уравнение на a11. Затем вычтем из каждого i-го (i≥2) уравнения, полученного после деления, первое, умноженное на ai1. В результате, после преобразований x1 окажется исключенным из всех уравнений кроме первого. По той же схеме исключается x2 (разделив второе уравнение на a22≠0), затем x3 и т.д.

В результате получается треугольная матрица с единичной главной диагональю.

 (2.2)

Особенность этой системы – в строках с номером i все коэффициенты aij при j<i равны нулю. Эту систему уравнений треугольного вида решить уже просто. Из последнего уравнения определяется xn, далее, подставляя его в предпоследнее уравнение, получаем xn-1 и т.д.

Общая формула определения неизвестных имеет вид

 (2.3)

Приведение системы (2.1) к треугольному виду (2.2) называется *прямым ходом метода Гаусса.* Процесс исключения k-го неизвестного называется *k-м шагом прямого хода*. Элементы называются *ведущими*.

Общие формулы пересчета коэффициентов системы на k-м шаге имеют вид:







Определение неизвестных по формулам (2.3) называется обратным ходом метода Гаусса.

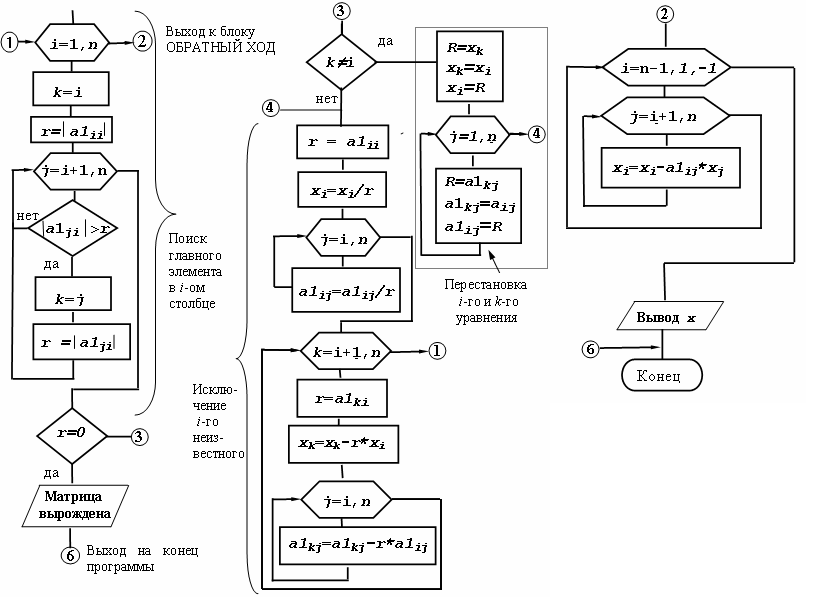
В методе Гаусса происходит деление строк на соответствующие ведущие элементы, поэтому, если на каком-то *k*-м шаге на главной диагонали окажется нулевой элемент  то среди элементов(i=k+1,…, n)следует найти ненулевой и перестановкой строк переместить его на главную диагональ, а затем продолжить вычисления.

Для этого следует воспользоваться, например, методом Гаусса выбора главного элемента в столбце, суть которого состоит в определении максимального элемента в столбце текущей строки и перестановке строки с максимальным элементом в столбце с текущей строкой, если таковой найден.

Если такого ненулевого элемента не найдется, то определитель системы равен нулю и система либо не имеет решений, либо решений бесконечно много.

На рис. 2.1 представлена блок-схема прямого хода – исключение *i*-го неизвестного по методу Гаусса. На рис. 2.2 представлена блок-схема обратного хода – определение неизвестных по методу Гаусса

## Блок-схема



## Код программы

public double[] solve(double[,] A, double[] X, int N)

{

double r = 0, R = 0;

int i, j, k;

//

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

{

k = i;

r = A[i, i];

for (j = i + 1; j < N; j++)

{

if (A[j, i] > r)

{

k = j;

r = A[j, i];

}

}

if (r == 0)

{

break;

}

if (k != i)

{

R = X[k];

X[k] = X[i];

X[i] = R;

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

{

R = A[k, j];

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

A[i, j] = R;

}

}

r = A[i, i];

X[i] = X[i] / r;

for (j = i; j < N; j++)

A[i, j] = A[i, j] / r;

for (k = i + 1; k < N; k++)

{

r = A[k, i];

X[k] = X[k] - r \* X[i];

for (j = i; j < N; j++)

A[k, j] = A[k, j] - r \* A[i, j];

}

}

if (r != 0)

{

for (i = N - 2; i >= 0; i--)

for (j = i + 1; j < N; j++)

X[i] = X[i] - A[i, j] \* X[j];

return X;

}

else

{

throw new System.Exception();

}

}

# Описание программы

## Общая структура программы

Программа для решения СЛАУ написана на языке программирования C# в интегрированной среде разработки Microsoft Visual Studio 2013.

Программа состоит из 7 форм:

1. Main\_form – Титульный лист. Появляется при запуске программы. (Рисунок3.2)
2. Menu\_form– Основное меню. Из него можно вызвать все остальные окна. (Рисунок 3.3)
3. Form1 – Окно решения СЛАУ (Рисунок 3.4)
4. VisForm – Окно визуализации.
5. About\_form – Окно «О программе».
6. Ref\_form – Справка.
7. Description\_form – Описание метода Гаусса.
8. Flowchart\_form – Блок-схема метода Гаусса.

Визуально структуру программы можно предоставить в виде графа, который показывает взаимодействие форм между собой.

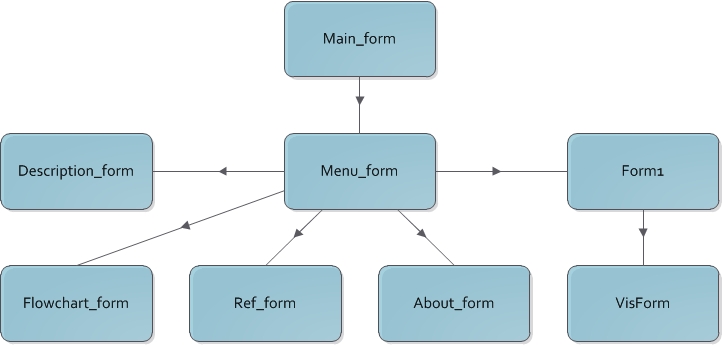


Рисунок3.1 Структура программы

## Описание работы программы

Для работы программы необходима операционная система Windows 7 и выше, а также .net Framework 4.5 и выше.

Для запуска программы необходимо запустить файл kursovaya.exe, который находиться на диске с курсовой работой. При запуске программы откроется титульный лист.

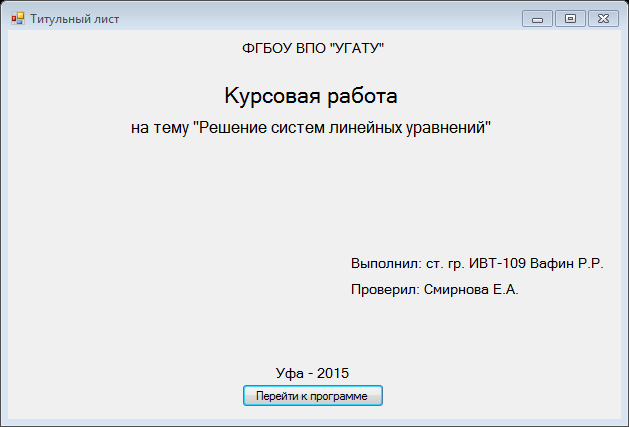


Рисунок 3.2 Титульный лист.

При нажатии на кнопку "Перейти к программе" (Main\_form.cs | privatevoid button1\_Click(object sender, EventArgs e)) окно "Титульный лист" закроется, и откроется основное меню программы (Рисунок3.1).

Данное окно содержит меню, которое открывает доступ к описанию метода, его блок-схеме, справке программы, и главной форме, где и происходит решение СЛАУ.

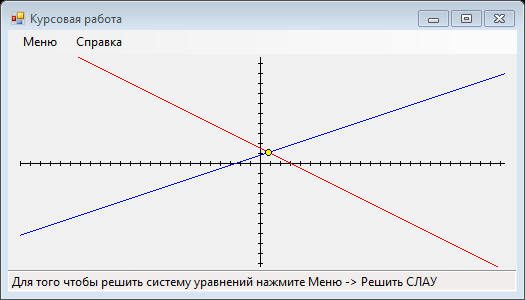
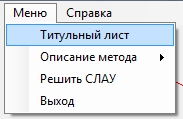
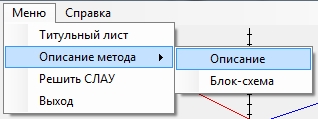
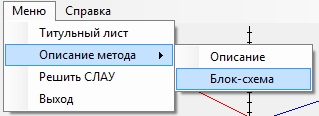
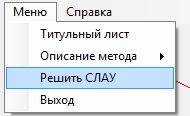
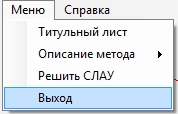
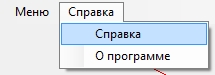
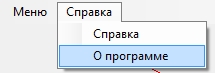


Рисунок 3.3 Основное меню.

**Пункты меню:**

1. Меню
   1. Титульный лист (Menu\_form.cs | privatevoidTL\_menuItem\_Click(object sender, EventArgs e)) – открывает Титульный лист.  
      
   2. Описание метода
      1. Описание (Menu\_form.cs | privatevoidDesc\_menuItem\_Click(object sender, EventArgs e)) – открывает описание метода.  
         
      2. Блок-схема (Menu\_form.cs | privatevoidFC\_menuItem\_Click(object sender, EventArgs e)) – открывает блок-схему метода.  
         
   3. Решить СЛАУ (Menu\_form.cs | privatevoidSolve\_menuItem\_Click(objectsender, EventArgse)) – открывает основную форму для решения СЛАУ.  
      
   4. Выход (Menu\_form.cs | privatevoidMenu\_form\_FormClosed(object sender, FormClosedEventArgs e)) – закрывает программу.  
      
2. Справка
   1. Справка (Menu\_form.cs | privatevoidRef\_menuItem\_Click(object sender, EventArgs e)) – открывает справку программы.  
      
   2. О программе (privatevoidAbout\_nemuItem\_Click(object sender, EventArgs e)) – открывает окно с названием программы, и ее автором.  
      

Каждый из пунктов меню имеет свой обработчик события нажатия, который указаны выше. В каждом из них создается новая форма, и делается видимой.

## Решение СЛАУ.

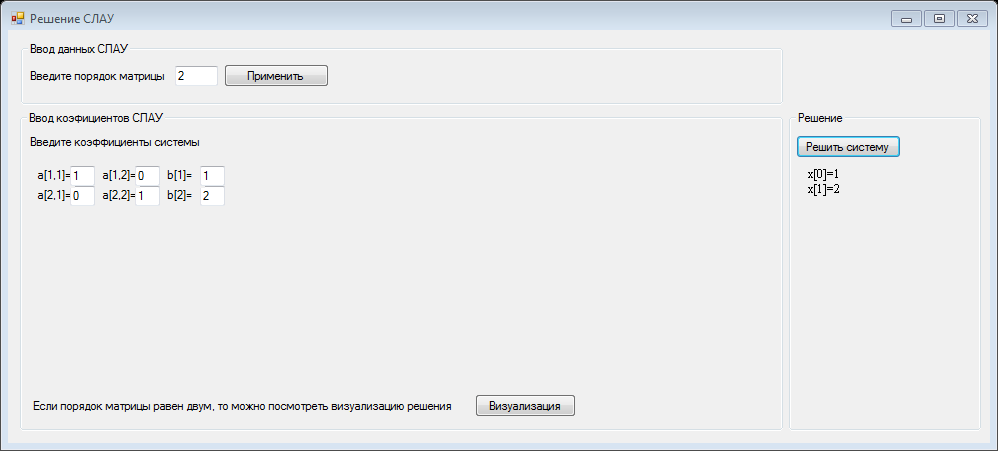


Рисунок 3.4 Форма решения СЛАУ.

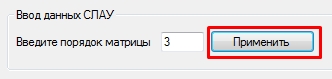
Основная форма для решения СЛАУ.

Для того, чтобы решить СЛАУ, необходимо:

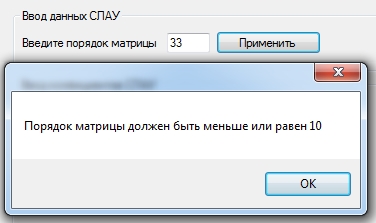
1. Ввести порядок матрицы.



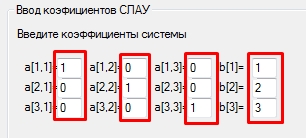
1. Нажать кнопку «Применить» (Form1.cs | privatevoidInit\_Input\_Field(object sender, EventArgs e) ).



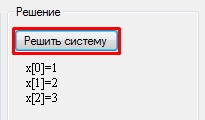
Если порядок больше 10, то выдаст сообщение о недопустимом порядке матрицы. (на след. странице)



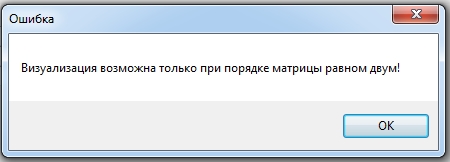
1. Ввести коэффициенты системы в поле ввода.



1. Нажать кнопку «Решить систему» (Form1.cs | privatevoidSolve\_SLAU(object sender, EventArgs e) ). Под кнопкой появится решение системы.



1. Если порядок введённой системы равен двум, то чтобы посмотреть визуализацию решения системы нужно нажать кнопку «Визуализация»  
     
   (Form1.cs | privatevoidbtn\_vis(objectsender, EventArgse)), иначе выдаст сообщение о недопустимом порядке матрицы.



**Визуализация.**

Визуализация представляет собой графическое решение системы второго порядка.

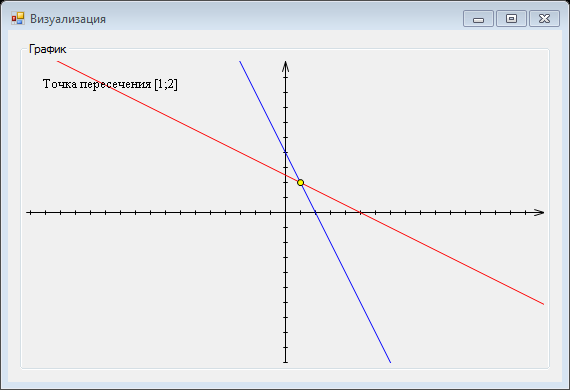


Рисунок 3.5 Визуализация решения.

На рисунке 3.5 показана визуализация решения системы 2-го порядка. Это 2 прямые, координаты точки пересечения который и есть решение данной системы. Если же система не имеет решения, то возможны 2 варианта:

1. Графики накладываются друг на друга, и тогда система имеет бесконечное количество решений. Рисунок 3.6
2. Графики параллельны, и тогда решения нет вообще. Рисунок 3.7

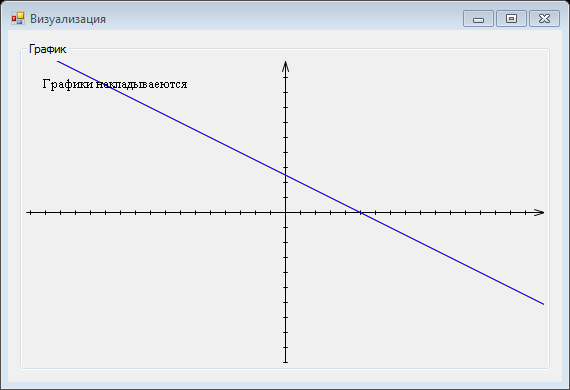


Рисунок 3.6 Графики накладываются.

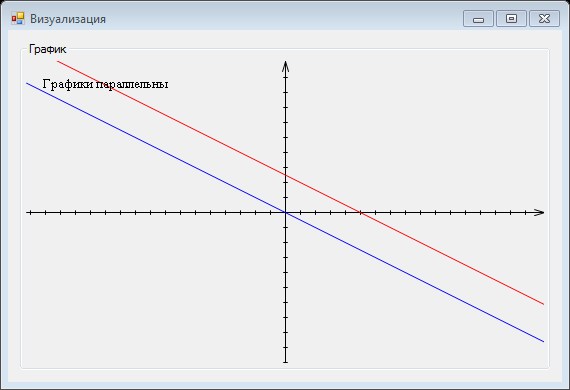


Рисунок 3.7 Графики параллельны.

Ниже показаны возможности программы для решения систем десятого порядка. Рисунок 3.8

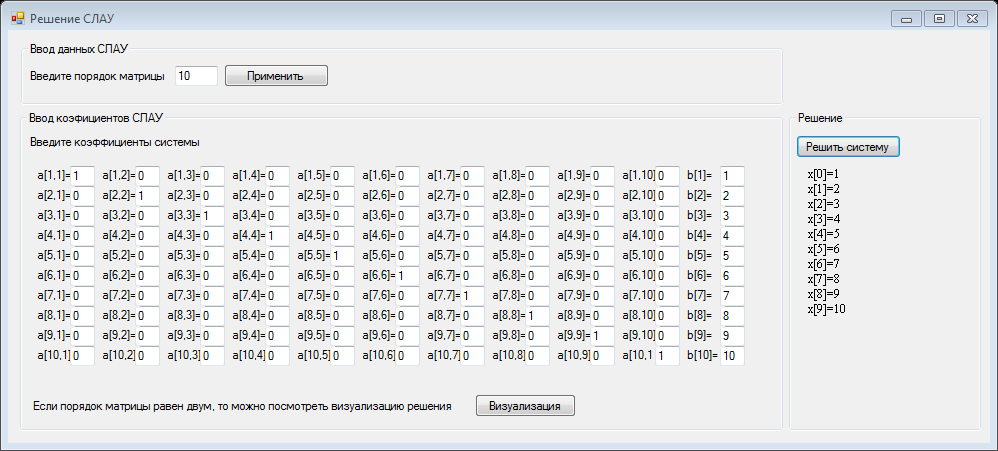


Рисунок 3.8 Решение системы 10-го порядка.

## Справка программы.

Форма описания метода Гаусса. В ней описан метод гаусса.

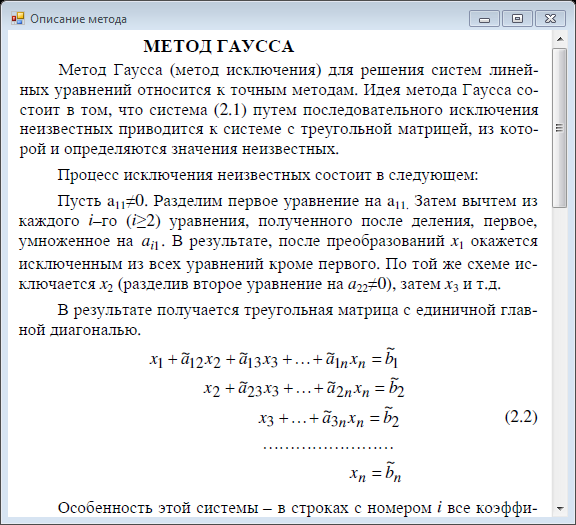


Рисунок 3.9 Форма описание метода.

Форма блок-схемы метода гаусса.

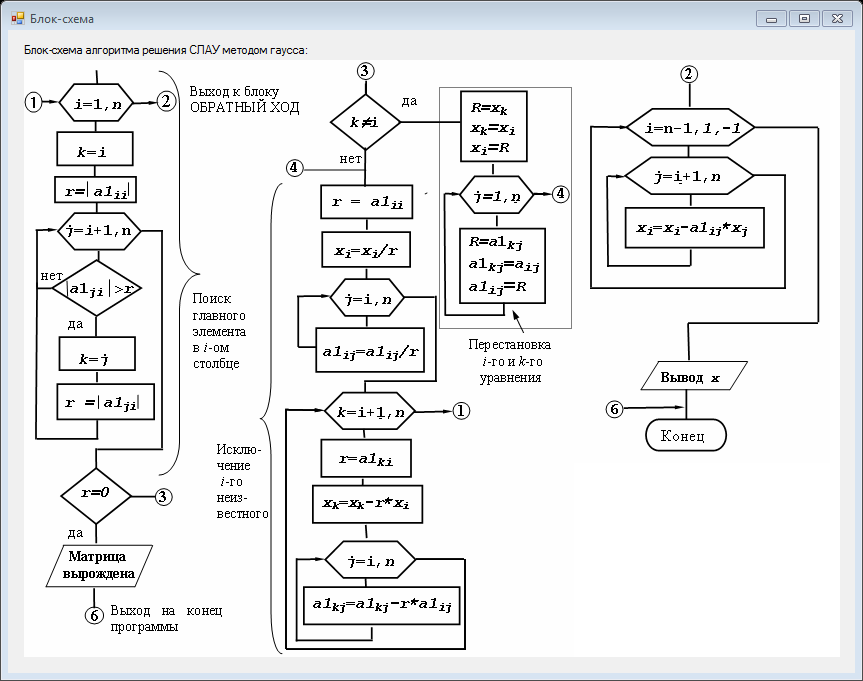


Рисунок 3.10 Форма блок-схемы метода Гаусса.

Форма справки. Там описано, как работать с программой. Описание всех элементов управления.

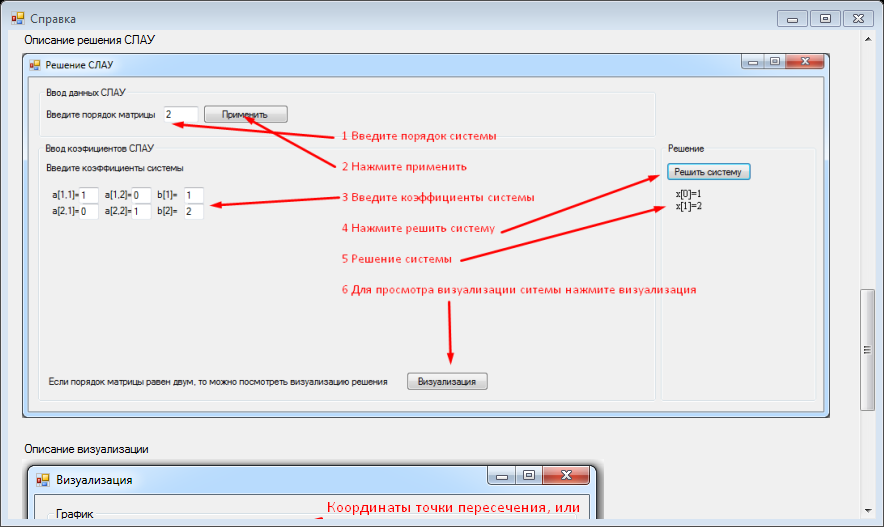


Рисунок 3.11 Справка

Форма «О программе». Указаны название программы, логотип программы, а так же автор программы.

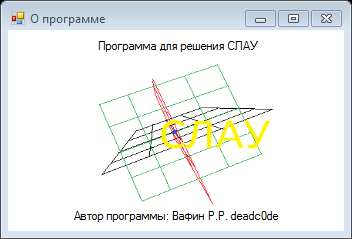


Рисунок 3.12 О программе.

# Вывод

В ходе выполнения курсовой работы были изучены базовые возможности интегрированной среды разработки (IDE) Microsoft Visual Studio 2013 для разработки на языке высокого уровня C#, изучены основы языке С# для разработки многооконных приложений, создания ветвящихся, циклических, и рекурсивных алгоритмов, изучены основы ООП (Объектно-Ориентированного Программирования).

Самым большим препятствие в работе над курсовой работой было недостаточное знание языка программирования C#. Поэтому в начале работы часть логики работы программы были написаны с использованием, так называемых, «костылей» - участков кода, которые хоть и решают поставленную задачу, но делают это нестандартным образом. К примеру, можно отнести реализацию закрытия программы из неосновной формы. Так как в то время мне не было известно как правильно закрывать приложение, я из другой формы, удалял основную, и тогда программа закрывалась. В ходе работы большая часть «костылей» была убрана. Язык C# изучался по «Методические указания к выполнению курсовых работ по дисциплине «Информатика»», «Герберт Шилдт - C# 4.0. Полное руководство» а так же по «Каталог API (Microsoft) и справочных материалов»

Данная программа может применяться студентами младших курсов для упрощения работы с математическим анализом, так как СЛАУ – это одна из тем первого семестра, с физикой и другими предметными областями, в которых, для решения задач, необходимо решать системы уравнений. Возможностей программы хватит для решения практически любой задачи на СЛАУ из первого курса математического анализа, так как там системы больше десятого порядка не рассматриваются.

Подводя итоги можно сказать, что в ходе работы над курсовой работой, я получил большой опыт в создании алгоритмов для сложных многооконных приложений, и создал программу, которая была бы очень полезной для студентов младших курсов.

# Используемая литература

1. Методические указания к выполнению курсовых работ по дисциплине «Информатика»А.Т. Бикмеев, М.П. Карчевская, Е. А. Кузьмина,О.Л. Рамбургер
2. Герберт Шилдт - C# 4.0. Полное руководство – 2011г

1. <https://msdn.microsoft.com/ru-ru/library/>Каталог API (Microsoft) и справочных материалов
2. <http://studopedia.ru/3_93698_primenenie-sistem-lineynih-uravneniy-v-ekonomike.html>

# Приложение

Form1.cs

using System;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespaceKursovaya

{

publicpartialclassForm1 : Form

{

publicint N = 2;

publicSystem.Windows.Forms.Control[,] In\_matrix, In\_matrix\_L;

privateGraphics g;

privateVisFormvs;

public Form1()

{

InitializeComponent();

Init\_Input\_Field(this,null);

}

publicdouble[] solve(double[,] A, double[] X, int N)

{

double r = 0, R = 0;

inti, j, k;

//

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

{

k = i;

r = A[i, i];

for (j = i + 1; j < N; j++)

{

if (A[j, i] > r)

{

k = j;

r = A[j, i];

}

}

if (r == 0)

{

break;

}

if (k != i)

{

R = X[k];

X[k] = X[i];

X[i] = R;

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

{

R = A[k, j];

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

A[i, j] = R;

}

}

r = A[i, i];

X[i] = X[i] / r;

for (j = i; j < N; j++)

A[i, j] = A[i, j] / r;

for (k = i + 1; k < N; k++)

{

r = A[k, i];

X[k] = X[k] - r \* X[i];

for (j = i; j < N; j++)

A[k, j] = A[k, j] - r \* A[i, j];

}

}

if (r != 0)

{

for (i = N - 2; i>= 0; i--)

for (j = i + 1; j < N; j++)

X[i] = X[i] - A[i, j] \* X[j];

return X;

}

else

{

thrownewSystem.Exception();

}

}

privatevoidprint\_false()

{

Out\_pic.Image = newBitmap(Out\_pic.Width, Out\_pic.Height);

Brush b = newSolidBrush(Color.Black);

Font f = newFont("Times New Roman", 10);

g = Graphics.FromImage(Out\_pic.Image);

g.TextRenderingHint = System.Drawing.Text.TextRenderingHint.AntiAliasGridFit;

g.DrawString("Матрицавырождена", f, b, newPointF(15,15));

}

privatevoidprint\_X(double[] X)

{

Out\_pic.Image = newBitmap(Out\_pic.Width, Out\_pic.Height);

g = Graphics.FromImage(Out\_pic.Image);

g.TextRenderingHint = System.Drawing.Text.TextRenderingHint.AntiAliasGridFit;

Brush b = newSolidBrush(Color.Black);

Font f = newFont("Times New Roman", 10);

int H = Out\_pic.Height, W = Out\_pic.Width;

inti;

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

{

g.DrawString("x[" + Convert.ToString(i) + "]=" + Convert.ToString(X[i]), f, b, newPointF(10, i \* 15));

}

}

privatevoidInit\_Input\_Field(object sender, EventArgs e)

{

inti, j;

if (In\_matrix!=null)

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

for (j = 0; j < N + 1; j++)

{

In\_matrix[i, j].Dispose();

In\_matrix\_L[i, j].Dispose();

}

N=Convert.ToInt16(Rang.Text);

if (N > 10)

{

N = 10;

MessageBox.Show("Порядок матрицы должен быть меньше или равен 10");

}

In\_matrix = newSystem.Windows.Forms.Control[N, N + 1];

In\_matrix\_L = newSystem.Windows.Forms.Control[N, N + 1];

int offset = 40;

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

for (j = 0; j < N + 1; j++)

{

In\_matrix[i, j] = newSystem.Windows.Forms.TextBox();

In\_area.Controls.Add(In\_matrix[i, j]);

In\_matrix[i, j].Location = newPoint(j \* 65 + 50, i \* 20 + 15 + offset);

In\_matrix[i, j].Size = newSize(25, 20);

In\_matrix[i, j].TabIndex = 4;

if (i == j) In\_matrix[i, j].Text = "1";

elseif (j == N) In\_matrix[i, j].Text = Convert.ToString(i+1);

elseIn\_matrix[i, j].Text = "0";

In\_matrix\_L[i, j] = newSystem.Windows.Forms.Label();

In\_area.Controls.Add(In\_matrix\_L[i, j]);

In\_matrix\_L[i, j].Location = newPoint(j \* 65 + 15, i \* 20 + 17 + offset);

In\_matrix\_L[i, j].Size = newSize(40, 20);

if (j != N) In\_matrix\_L[i, j].Text = "a[" + Convert.ToString(i + 1) + "," + Convert.ToString(j + 1) + "]=";

elseIn\_matrix\_L[i, j].Text = "b[" + Convert.ToString(i + 1) + "]=";

}

}

privatevoidSolve\_SLAU(object sender, EventArgs e)

{

double[,] A = newdouble[N, N];

double[] B = newdouble[N];

double[] X;

inti, j;

try

{

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

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

A[j, i] = Convert.ToDouble(In\_matrix[j, i].Text);

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

B[i] = Convert.ToDouble(In\_matrix[i, N].Text);

X = solve(A, B, N);

print\_X(X);

}

catch

{

print\_false();

}

}

privatevoidbtn\_vis(object sender, EventArgs e)

{

if (N == 2)

{

inti, j;

double[,] A = newdouble[N, N];

double[] B = newdouble[N];

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

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

A[j, i] = Convert.ToDouble(In\_matrix[j, i].Text);

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

B[i] = Convert.ToDouble(In\_matrix[i, N].Text);

vs = newVisForm();

vs.InitMatrix(A, B);

vs.Show();

}

else

MessageBox.Show("Визуализация возможна только при порядке матрицы равном двум!", "Ошибка");

}

}

}

VisForm.cs

using System;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespaceKursovaya

{

publicpartialclassVisForm : Form

{

double[,] AB,X;

double A1, A2, B1, B2, C1, C2;

double k, b;

publicVisForm()

{

InitializeComponent();

}

publicvoidInitMatrix(double[,] A, double[] B)

{

inti,j;

Out\_vis.Image = newBitmap(Out\_vis.Width, Out\_vis.Height);

Graphics g = Graphics.FromImage(Out\_vis.Image);

g.TextRenderingHint = System.Drawing.Text.TextRenderingHint.AntiAliasGridFit;

g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.HighQuality;

Pen p = Pens.Black;

Brushbr = newSolidBrush(Color.Black);

Font f = newFont("Times New Roman", 10);

float H = Out\_vis.Height, W = Out\_vis.Width;

double Scale = 15;

A1 = A[0, 0]; A2 = A[1, 0];

B1 = A[0, 1]; B2 = A[1, 1];

C1 = B[0]; C2 = B[1];

//grid

intarrow\_a = 3, arrow\_b = 10;

g.DrawLine(p, 0, H / 2, W, H / 2); //x line

g.DrawLine(p, W / 2, 0, W / 2, H); //y line

g.DrawLine(p, W - arrow\_b, H / 2 + arrow\_a, W, H / 2); //x arrow

g.DrawLine(p, W - arrow\_b, H / 2 - arrow\_a, W, H / 2);

g.DrawLine(p, W / 2 - arrow\_a, arrow\_b, W / 2, 0); //y arrow

g.DrawLine(p, W / 2 + arrow\_a, arrow\_b, W / 2, 0);

for (i = 0; i<= W - arrow\_b; i++) //x

{

if ((W / 2 - i) % Scale == 0)

g.DrawLine(p,i,H/2+2,i,H/2-2);

}

for (i = arrow\_b; i<= H; i++) //y

{

if ((H / 2 - i) % Scale == 0)

g.DrawLine(p, W / 2 + 2, i, W / 2 - 2, i);

}

//line1

if (B1 != 0)

{

k = -A1 / B1;

b = C1 / B1;

g.DrawLine(Pens.Red, 0, Convert.ToSingle(-(-W / 2 \* k + b \* Scale) + H / 2),

Convert.ToSingle(W), Convert.ToSingle(-(W / 2 \* k + b \* Scale) + H / 2));

}

else

{

g.DrawLine(Pens.Red, Convert.ToSingle(C1 / A1 \* Scale + W / 2), 0, Convert.ToSingle(C1 / A1 \* Scale + W / 2), H);

}

//line2

if (B2 != 0)

{

k = -A2 / B2;

b = C2 / B2;

g.DrawLine(Pens.Blue, 0, Convert.ToSingle(-(-W / 2 \* k + b \* Scale) + H / 2),

Convert.ToSingle(W), Convert.ToSingle(-(W / 2 \* k + b \* Scale) + H / 2));

}

else

{

g.DrawLine(Pens.Blue, Convert.ToSingle(C2 / A2 \* Scale + W / 2), 0, Convert.ToSingle(C2 / A2 \* Scale + W / 2), H);

}

if ((A1==0 && A2==0 && C1/B1==C2/B2) || (B1==0 && B2==0 && C1/A1==C2/A2) || (B1!=0 && B2!=0 && A1/B1==A2/B2 && C1/B1==C2/B2))

g.DrawString("Графики накладываеются", f, br, 15, 15);

elseif ((A1==0 && A2==0 && C1/B1!=C2/B2) || (B1==0 && B2==0 && C1/A1!=C2/A2) || (B1!=0 && B2!=0 && A1/B1==A2/B2 && C1/B1!=C2/B2))

g.DrawString("Графики параллельны", f, br, 15, 15);

else

{

double x, y;

if (A1==0 && B2==0)

{

x = C2 / A2;

y = C1 / B1;

}

elseif (A2 == 0 && B1 == 0)

{

x = C1 / A1;

y = C2 / B2;

}

else

{

y = (C1 / A1 - C2 / A2) / (B1 / A1 - B2 / A2);

x = C1 / A1 - B1 \* y / A1;

}

g.DrawString("Точкапересечения [" + Convert.ToString(x) + ";" + Convert.ToString(y) + "]", f, br, 15, 15);

g.FillEllipse(Brushes.Yellow,

Convert.ToSingle(W / 2 + x \* Scale - 3), Convert.ToSingle(H / 2 - y \* Scale - 3), 6, 6);

g.DrawEllipse(Pens.Black,

Convert.ToSingle(W / 2 + x \* Scale - 3), Convert.ToSingle(H / 2 - y \* Scale - 3), 6, 6);

}

}

}

}

Menu\_form.cs

using System;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespaceKursovaya

{

publicpartialclassMenu\_form : Form

{

objecttl;

publicMenu\_form(object sender)

{

InitializeComponent();

tl = sender;

}

privatevoidTL\_menuItem\_Click(object sender, EventArgs e)

{

(tlasMain\_form).Visible = true;

this.Dispose();

}

privatevoidSolve\_menuItem\_Click(object sender, EventArgs e)

{

Form1fr = newForm1();

fr.Show();

}

privatevoidDesc\_menuItem\_Click(object sender, EventArgs e)

{

Description\_formfr = newDescription\_form();

fr.Show();

}

privatevoidFC\_menuItem\_Click(object sender, EventArgs e)

{

Flowchart\_formfr = newFlowchart\_form();

fr.Show();

}

privatevoidExit\_menuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

privatevoidRef\_menuItem\_Click(object sender, EventArgs e)

{

ref\_formfr = newref\_form();

fr.Show();

}

privatevoidAbout\_nemuItem\_Click(object sender, EventArgs e)

{

About\_formfr = newAbout\_form();

fr.Show();

}

privatevoidMenu\_form\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

}

}

Program.cs

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespaceKursovaya

{

staticclassProgram

{

///<summary>

/// Главная точка входа для приложения.

///</summary>

[STAThread]

staticvoid Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(newMain\_form());

}

}

}

Main\_form.cs

using System;

usingSystem.Collections.Generic;

usingSystem.ComponentModel;

usingSystem.Data;

usingSystem.Drawing;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Windows.Forms;

namespaceKursovaya

{

publicpartialclassMain\_form : Form

{

publicMain\_form()

{

InitializeComponent();

}

privatevoid button1\_Click(object sender, EventArgs e)

{

Menu\_form mf = newMenu\_form(this);

mf.Show();

this.Visible = false;

}

}

}