СОДЕРЖАНИЕ

[ВСТУПЛЕНИЕ 5](#_Toc516755746)

[2. СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ 7](#_Toc516755748)

[3. МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ 9](#_Toc516755749)

[4. МЕТОД ЖОРДАНА‑ГАУССА 10](#_Toc516755750)

[5. ОБЩАЯ СХЕМА АЛГОРИТМА РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ 13](#_Toc516755751)

[6. ОСОБЕННОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ С ТОЧКИ ЗРЕНИЯ ПРОГРАММИСТА 14](#_Toc516755752)

[7. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 16](#_Toc516755753)

[ЗАКЛЮЧЕНИЕ 23](#_Toc516755754)

[СПИСОК ЛИТЕРАТУРЫ 24](#_Toc516755755)

[ПРИЛОЖЕНИЕ А 25](#_Toc516755756)

# Вступление

Решение систем линейных уравнений одна из важнейших задач линейной алгебры.

В моей курсовой работе раскрывается вопрос решения систем линейных уравнений методом Жордана – Гаусса, составление программы для решения систем линейных уравнений и способы получения результата.

Для написания программы я выбрала язык программирования Visual Studio C#, так как этот язык программирования, предназначенный для разработки самых разнообразных приложений, предназначенных для выполнения в среде .NET Framework. Библиотека классов .NET Framework предоставляет доступ ко многим службам операционной системы и другим полезным классам, что существенно ускоряет цикл разработки.

Работа состоит из двух частей - теоретической и практической.

В теоретической части приведены определения таких понятий, как система линейных уравнений, общее и частное решение, совместность и несовместность систем, однородные и неоднородные системы, а также краткие сведения о методах решения систем линейных уравнений, более детально рассмотрен метод Жордана – Гаусса.

В практической части составлена программа для решения систем линейных уравнений.

Системы линейных алгебраических уравнений используются в различных экономических задачах, в задачах механики, в математической физике при численном решении дифференциальных и интегральных уравнений, являются важным атрибутом при расчете сложных электрических цепей различными методами: по законам Кирхгофа, контурными токами, узловыми потенциалами. Так же при конструировании инженерных сооружений, обработке результатов измерений, решении задач планирования производственного процесса и ряда других задач техники, научного эксперимента приходится решать системы линейных уравнений.

Актуальность моей курсовой работы заключается в том, что системы линейных уравнений – это математический аппарат, который имеет широкое применение в решении многих задач практического приложения математики.

Объект исследования: системы линейных алгебраических уравнений.

Предмет исследования: решение системы линейных уравнений методом Жордана‑Гаусса.  
Постановка задачи **-** составить программу для решения квадратной системы линейных уравнений методом Жордана-Гаусса.



# Системы линейных уравнений

Рассмотрим общие сведения о системах линейных уравнений, следуя [1].

Системой линейных уравнений с неизвестными называется система вида

 (1)

где  корни СЛАУ, коэффициенты системы, свободные члены.

В обозначении коэффициентов  первый индекс  обозначает номер уравнения, а второй  – номер неизвестного, при котором стоит этот коэффициент.  
Коэффициенты при неизвестных будем записывать в виде матрицы, которую назовём *матрицей системы*:

(2)

Числа, стоящие в правых частях уравнений, ,…, называются *свободными членами.*

В матричном виде квадратная СЛАУ выглядит следующим образом:

 (3)

или , где *А* - матрица коэффициентов системы, *х* - столбец неизвестных, *b* - столбец свободных членов.

Если все свободные члены системы линейных уравнений равны 0, то система называется *однородной*, в противном случае — *неоднородной*.

*Частным решением* (или просто *решением*) системы (1) называется упорядоченный набор чисел такой, что при подстановке в любое уравнение системы (1) вместо, вместо , . . . , вместо получается верное равенство.

Система линейных уравнений, имеющая хотя бы одно решение, называется *совместной*. В противном случае, т.е. если система не имеет решений, то она называется *несовместной*.

Множество всех решений системы линейных уравнений называется *общим решением* этой системы.

Заметим, что:

* общее решение есть у любой системы. В частности, у несовместной системы общим решением является пустое множество.
* общее решение совместной системы линейных уравнений равно сумме её частного решения и общего решения соответствующей однородной системы.

Наша задача будет заключаться в нахождении решений системы. При этом могут возникнуть три ситуации:

1. Система может иметь единственное решение.
2. Система может иметь бесконечное множество решений. Например, http://www.toehelp.ru/theory/math/lecture14/l14image006.gif. Решением этой системы является любая пара чисел, отличающихся знаком.
3. И третий случай, когда система вообще не имеет решения. Например, http://www.toehelp.ru/theory/math/lecture14/l14image008.gif, если бы решение существовало, то *x1 + x2* равнялось бы одновременно нулю и единице.

# Методы решения систем линейных уравнений

В работе [2] расмотрены методы решения СЛАУ, которые делятся на две группы:

* прямые (точные) методы;
* итерационные (приближенные) методы.

К ***прямым*** методам относятся такие методы, которые, в предположении, что вычисления ведутся без округлений, позволяют получить точные значения неизвестных. Они просты, универсальны и используются для широкого класса систем. К ним можно отнести: *правило Крамера*, методы *обратных матриц*, *Гаусса*, *прогонки*, *квадратного корня* и др.

К ***приближенным*** относятся методы, которые даже в предположении, что вычисления ведутся без округлений, позволяют получить решение системы лишь с заданной точностью. К ним относятся методы *простой итерации*, *Зейделя*.

Сравним прямые и итерационные методы по временным затратам. При реализации на языках высокого уровня алгоритмов прямых методов решения систем линейных уравнений размерностью N  необходимо учитывать, что число арифметических операций умножения будет составлять N3. Для систем с размерностью эта кубическая зависимость приводит к большим затратам количества времени при решении даже на самых современных ЭВМ N>1000.

Итерационные методы решения СЛАУ намного экономнее по затратам машинного времени. Так, если итерационный метод является быстро сходящимся с числом итераций m << N, то время решения пропорционально квадрату размера матрицы ~ m-N2. Эти затраты меньше, по сравнению с точными методами, в N/m раз для вещественной, и 2N/m раз для комплексной СЛАУ [3].

# 

# Метод Жордана‑Гаусса

*Метод Гаусса – Жордана (метод полного исключения неизвестных)* представляет собой удобный вычислительный алгоритм, построенный на последовательном применении эквивалентных преобразований системы линейных уравнений, которые приводят к равносильной системе. Метод является модификацией [метода Гаусса](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%93%D0%B0%D1%83%D1%81%D1%81%D0%B0) [2].

На преобразованиях Жордана - Гаусса основан и такой универсальный метод решения линейных задач оптимизации, как симплекс-метод.

В учебном пособии [4] подробно описан алгоритм решения системы линейных уравнений методом Жордана-Гаусса.

Пусть дано систему из m линейных уравнений с n переменными , :

**** (4)

или .

Шагом исключения Жордана-Гаусса для системы уравнений вида (4) является разрешающим элементом , где , , , называют переход к эквивалентной системе уравнений с новыми коэффициентами, переменная которой входит с единичным коэффициентом в r-ое уравнение системы и с нулевыми коэффициентами в остальные уравнения системы. Другими словами, метод исключения Жордана-Гаусса заключается в преобразовании матрицы коэффициентов к единичному виду.

Новая система уравнений имеет вид

.

Для реализации такого перехода можно с каждого -го уравнения () вычесть r-ое уравнение системы, умноженное на соответствующий коэффициент. Обозначим его через k.

Исходная система (1) имеет вид



После преобразований получим новую систему:

, . (5)

Новые коэффициенты r-ого ряда получим делением -ого ряда на разрешающий элемент , т. е.

. (6)

Для того чтоб получить коэффициенты произвольного i-го ряда, от системы вычитаем -й ряд, умноженный на некоторый коэффициент .

Определим этот коэффициент:

. (7)

Следовательно, произвольный коэффициент новой системы

. (8)

Соответствующие свободные члены

 (9)

Пусть система (1) имеет ранг , . Тогда с помощью шагов исключения Жордана-Гаусса можно получить новую систему, эквивалентную исходной, переменных которой входят с единичным коэффициентом только в одно с уравнений. Обозначим множества таких переменных через δ:

. Индексы переменных, которые остались, образуют множество

Новую систему можно получить в виде

 (10)

Эта система имеет название системы уравнений в каноническом виде. Переменные называют базисными, а , j – небазисными или свободными.

Из выражения (6) легко можно получить общее решение системы уравнений, для этого все небазисные переменные переносим в правую часть уравнения:

 (11)

Для любого конкретного набора значений небазисных переменных с выражения (7) базисные переменные могут приобрести конкретные значения.

Любой такой набор значений – частное решение системы (1), а частное решение, при котором значение всех небазисных переменных равняется 0, - базисное решение системы (1) за базисом δ:

 (12)

# Общая схема алгоритма решения систем линейных уравнений

Шаг метода состоит в следующем:

* выбрать в очередном уравнении неизвестное с коэффициентом, отличным от нуля (разрешающим элементом);
* разделить выбранное уравнение на разрешающий элемент;
* с помощью выбранного уравнения исключить неизвестное при разрешающем элементе из всех остальных уравнений;
* на следующем шаге аналогично исключается другое неизвестное из всех уравнений, кроме одного;
* процесс продолжается, пока не будут использованы все уравнения.

Выбран разрешающий элемент , тогда  - разрешающая строка,  ‑ разрешающий столбец.

Элементы вне разрешающих строки и столбца вычисляются по формулам (6) – (9).

Возможны следующие случаи:

1. Система имеет единственное решения.

2. Система несовместна или линейно – зависима.

# Особенности программной реализации с точки зрения программиста

Программная реализация выполнена на языке C# в среде Microsoft Visual Studio 2013.

В программе используется класс «Система\_линейных\_уравнений». Его полями являются:

int dim; // размерность системы

double [,]a; //коэффициенты системы

double[] b; // матрица свободных членов

В классе «Система\_линейных\_уравнений» использованы четыре метода:

1. Void Чтение\_из\_файла(StreamReader sr).

В этом методе происходит чтение исходных данных для задачи из текстового файла, формат которого определен в следующем разделе.

*Исходные данные метода* – имя файловой переменной sr, которая до этого уже должна быть связанной с физическим файлом и открыта для чтения.

*Результатом работы метода* будут заполненные поля объекта – , и .

1. Public bool Решение(out double[] x).

В этом методе происходит решение системы линейных уравнений.

*Исходные данные метода* – поля объекта – , и , содержащие описание исследуемой системы.

*Результатом работы метода* являются значения параметра x, содержащие решение задачи. В качестве возвращаемого методом решения будет *истина (true)*, если существует единственное решение системы. Если система несовместна или линейно – зависима, возвращаемым результатом будет *ложь (false)*. В этом случае значения параметра x неопределенны.

В результате получим значения, удовлетворяющие нашу исходную систему.

1. Public void запись\_в\_файл(double[] x)

В этом методе происходит запись в файл «out.txt» значений, удовлетворяющие нашу исходную систему.

*Исходные данные метода* – массив .

*Результатом работы метода* является записанное решение нашей системы.

1. Public void запись\_в\_файл(double[,] a, double [] b)

*Исходные данные метода* – двумернный массив и одномернный массив.

*Результатом работы метода* является запись в файл «out.txt» преобразований системы линейных уравнений методом Жордана‑Гаусса.

Также использованы классы:

Form1.cs – класс, реализующий интерфейс основного окна и вызов функций из остальных основных классов для решения систем линейных уравнений.

Program.cs – класс, который включает визуальные стили для приложения, запускает стандартный цикл обработки сообщений приложения в текущем потоке и делает Form1 – видимой.

Class1.cs – класс, реализующий чтение из файла, решение системы линейных уравнений и запись в файл.

Form1 (форма) — это визуальная поверхность, на которой выводится информация для пльзователя.

ToolStripMenuItem - представляет отдельные пункты, отображаемые на MenuStrip1, в данном случае – выбор «прочитать с файла».

Button1 – процедура, в которой находим решение систем линейных уранений.

Button2 – процедура для вывода решения на Form1.

[DataGridView](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.datagridview(v=vs.90).aspx) - таблица для отображения результат с файла out.txt

# Руководство пользователя

В данной программе происходит решение систем линейных уравнений, у которой есть единственное решение.

Исходные данные нашей системы должны быть записаны в текстовом файле формата «txt». Где первая строчка – это размерность матрицы, остальные значения – расширенная матрица системы. Разделяющим знаком между цифрами является пробел.

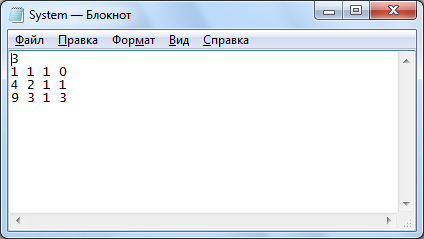


Рис. 1 Исходные данные

Вначале в главном окне программы (см. Рис.2) следует выбрать «Файл» (см. рис. 2), далее «Прочитать» (см. Рис. 3).

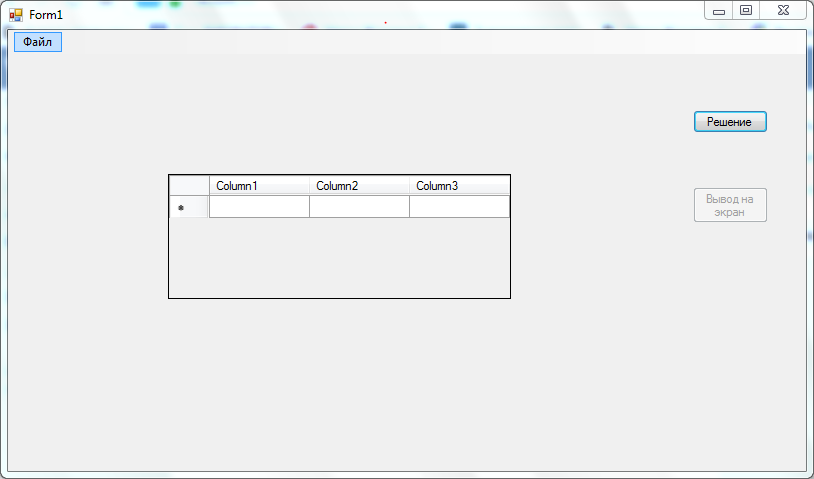


Рис. 2 Главное окно программы

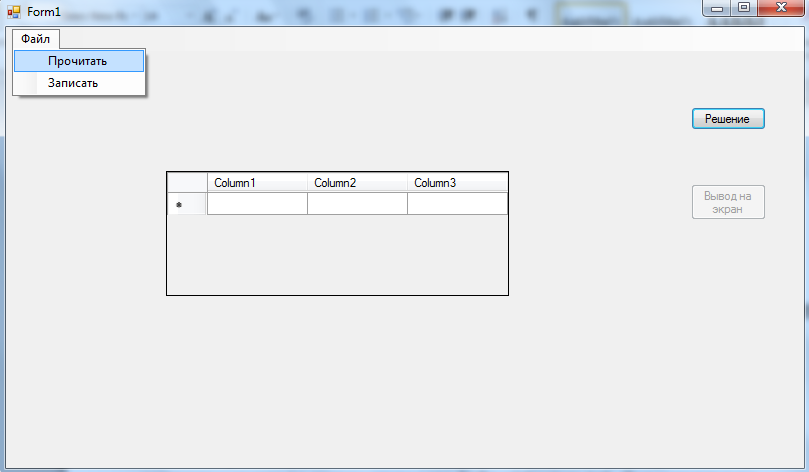


Рис. 3

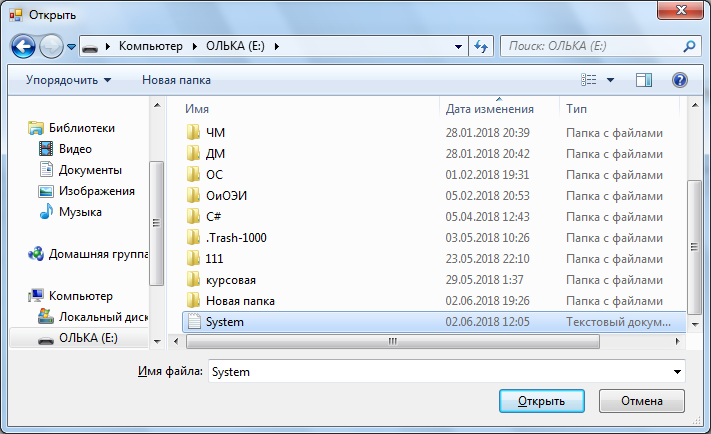


Рис. 4 Выбор файла

После этого появится окно, где следует выбрать файл в котором записаны данные (в нашем случае это файл e:\\System.txt – см. Рис. 4). Далее нажимаем кнопку «Решение» (см. Рис. 5). При нажатии этой кнопки выполнится решение задачи и его результат запишется в файл «out.txt ». Если для рассматриваемой системы существует единственное решение, то в файл запишутся координаты вектора решения. Иначе – сообщение, что решения нет или система линейно зависима.

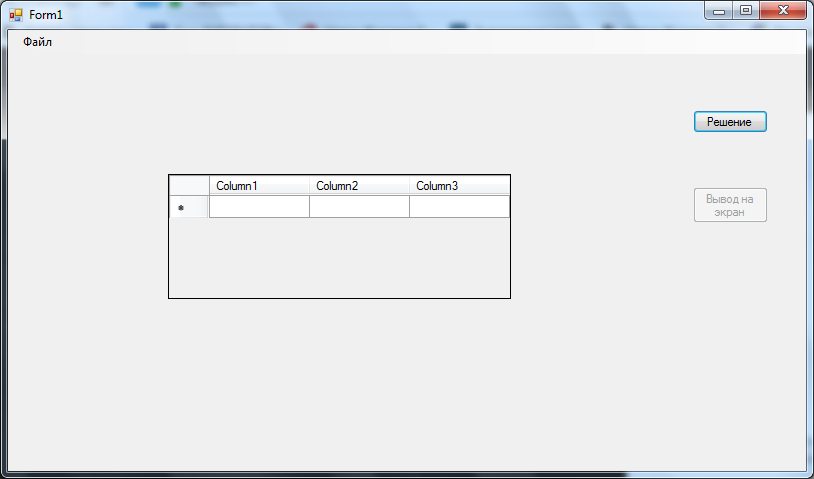


Рис. 5

Замечание. Кнопка «Вывод на экран» будет активна только при условии нажатия на кнопку «Решение» (см. Рис. 6).

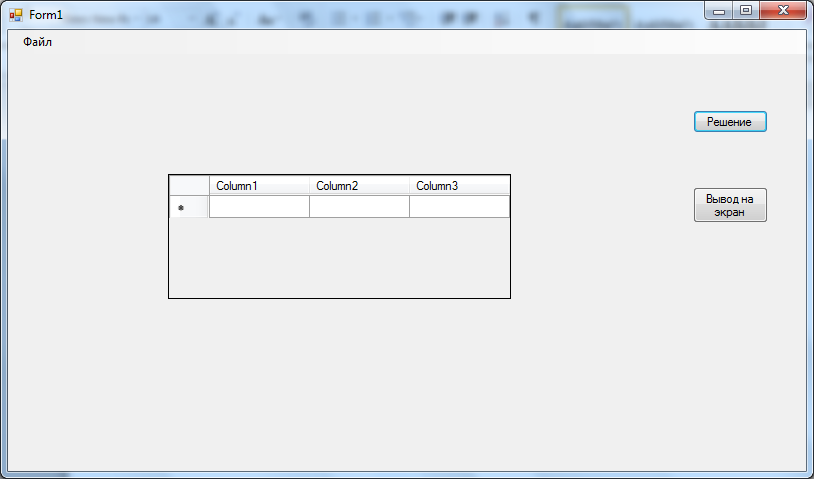


Рис. 6

Затем нажимаем кнопку «Вывод на экран» в результате в таблице будут выведены найденные неизвестные для исходной системы линейных уравнений (см. Рис. 7).

Также в папке с адресом E:\курсовая\курсач\Проект\WindowsFormsApplication1\bin\Debug находится решение и ответ записанные в текстовом виде (см. Рис. 8).

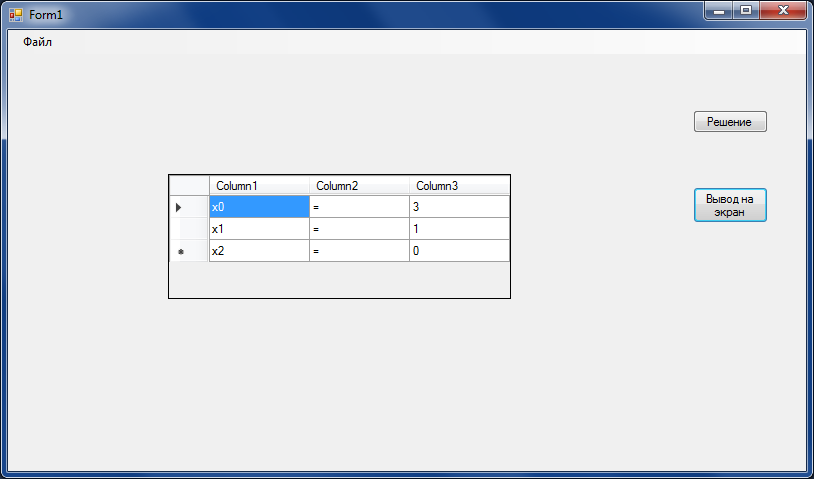


Рис. 7 Вывод ответа на Form1 при условии существования единственного решения

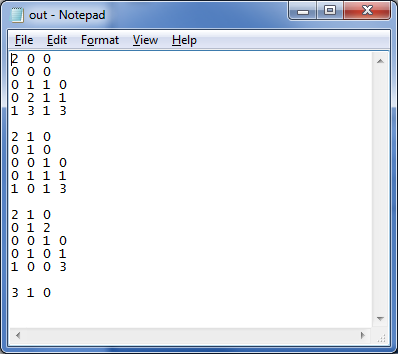


Рис. 8 Решение и ответ

Рассмотрим случай, когда решений нет или бесконечное множество.

Для этого выберем новый текстовой файл с исходными данными (см. рис. 9):

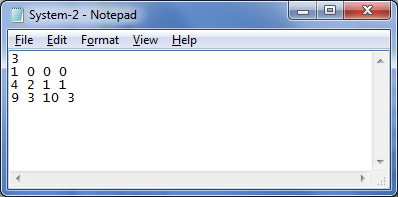


Рис. 9 Исходные данные

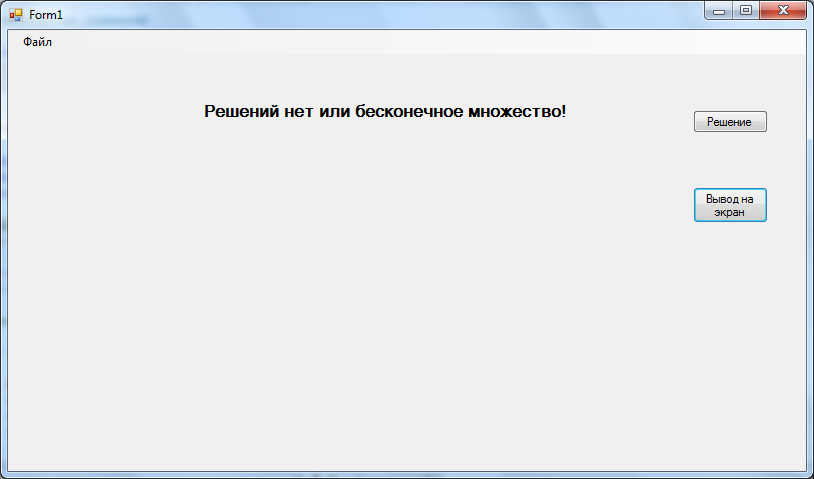


Рис. 10 Вывод ответа на экран программы

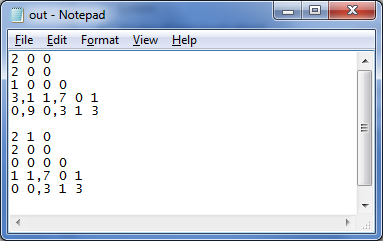


Рис. 11 Ответ в текстовом файле для системы, у которой нет решения

Как видим, на Рис. 11 на втором шаге у первого уравнения нет решения, следовательно, наша исходная система не имеет решений.

# **Заключение**

Метод Гаусса – Жордана представляет собой удобный вычислительный алгоритм, построенный на последовательном применении эквивалентных преобразований системы линейных уравнений, которые приводят к эквивалентной системе. Данный метод используется в симплекс методе для решения линейных задач оптимизации.

Решение систем линейных уравнений используется практически во всех науках, таких как экономика, математическая физика, дифференциальные уравнения, электротехника и другие.

При работе над проектом осуществлялась работа со считыванием данных с текстовых файлов и запись в текстовой файл.

В результате выполнения курсовой работы была разработана программа «Решение систем линейных уравнений метод Жордана ‑ Гаусса». В пояснительной записке содержатся все этапы разработки проекта, интерфейс рабочего приложения, исходный код программы, а также общие сведения систем линейных уравнений.

# Список литературы

|  |  |
| --- | --- |
| [1] | Воеводин В.В., Кузнецов Ю.А., Матрицы и вычисления, Москва: Наука, 1984. |
| [2] | Баландин М. Ю., Методы решения СЛАУ, Новосибирск: НГТУ, 2000, 70 стр. |
| [3] | Куксенко С.П., Газизов Т.Р., Использование методов решения СЛАУ: Учебное методическое пособие., Томск, 2012. |
| [4] | Карташов А. В., Бабкина А. В., Емцева Н. Ю., Пудло Р. А., Методы оптимизации, Харьков: НАУ им. Н. И. Жуковского "ХАИ", 2009, 112 стр. |

# Приложение А

***Program.cs***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

***Form1.cs***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

public partial class Form1 : Form

{

System.IO.StreamReader sr;

double[] x;

Система\_линейных\_уравнений s;

bool Решение\_существует;

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

button2.Enabled = true;

s = new Система\_линейных\_уравнений();

s.Чтение\_из\_файла(sr);//("e:\\System.txt");

Решение\_существует=s.Решение(out x);

if (Решение\_существует) s.запись\_в\_файл(x);

}

private void tableLayoutPanel1\_Paint(object sender, PaintEventArgs e)

{

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

if (Решение\_существует)

{

int n = s.dim;

DataGridViewTextBoxCell txtCell;

for (int i = 0; i < n - 1; ++i)

dataGridView1.Rows.Add();

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

{

txtCell =

(DataGridViewTextBoxCell)dataGridView1.Rows[i].Cells[0];

txtCell.Value = "x" + i.ToString();

txtCell =

(DataGridViewTextBoxCell)dataGridView1.Rows[i].Cells[1];

txtCell.Value = "=";

txtCell =

(DataGridViewTextBoxCell)dataGridView1.Rows[i].Cells[2];

txtCell.Value = x[i].ToString();

}

}

else

{

dataGridView1.Visible = false;

label2.Visible = true;

}

}

private void прочитатьToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

sr = new

System.IO.StreamReader(openFileDialog1.FileName);

}

}

}

}

***Class1.cs***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace WindowsFormsApplication1

{

class Система\_линейных\_уравнений

{

public int dim; // размерность системы

double [,]a; //коэффициенты системы

double[] b; // матрица свободных членов

public

void Чтение\_из\_файла(StreamReader sr)

{

string StrLine;//строка для чтения очередной строки из файла

string[] StrArr;//элементы стороки, разделенные пробелом

char[] differ = new char[] { ' ' };//разделитель между двумя элементами массива

StrLine = sr.ReadLine();//чтение размерностей массива

dim = int.Parse(StrLine);

a = new double[dim, dim];//инициализация массива

b = new double[dim];

//Построчное считывание данных

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

{

StrLine = sr.ReadLine();//чтение очередной строки

StrArr = StrLine.Split(differ);//раздление на элементы

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

{

a[i,j] = int.Parse(StrArr[j]);//преобразование к числовому виду

}

b[i] = int.Parse(StrArr[dim]);

}

sr.Close();

}

public bool Решение(out double[] x)

{

double eps = 1e-5;

bool solution\_exist=true;

x = new double[dim];

int [] row = new int [dim]; //номера строк разрешающих элементов

int [] col = new int[dim]; //номера столбцов разрешающих элементов

for (int k=0;k<dim && solution\_exist;++k) //k-номер итерации

{

//1.выбор разрещающего элемента

int r = 0;//номер строки разрещшающего элемента

int s = 0;//номер столбца разрешающего элемента

//row[k] = 0;

//col[k] = 0;

double max = Double.MinValue;

for (int i = 0; i < dim; ++i) // номера строк матрицы А

{ // индекс не должен быть в массиве row

int w = 0;

while (w < k && row[w] != i) ++w;

if (w==k)

for (int j = 0; j < dim; ++j) // номера столбцов матрицы А

{

w = 0;

while (w < k && col[w] != j) ++w;

if (w == k)

if (Math.Abs(a[i, j]) > max)

{

max=Math.Abs(a[i, j]);

row[k] = i;

col[k] = j;

}

}

}

r = row[k];

s = col[k];

if (Math.Abs(a[r, s]) > eps)

{

for (int i = 0; i < dim; ++i) // номера строк матрицы А

{

if (i == r)

{

for (int j = 0; j < dim; ++j) // номера столбцов матрицы А

a[i, j] /= a[r, s];

b[i] /= a[r, s];

}

else

{

for (int j = 0; j < dim; ++j) // номера столбцов матрицы А

a[i, j] -= a[i, s] / a[r, s] \* a[r, j];

b[i] -= a[i, s] / a[r, s] \* b[r];

}

}

запись\_в\_файл(row);

запись\_в\_файл(col);

запись\_в\_файл(a, b);

}

else solution\_exist = false;

}

if (solution\_exist)

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

x[row[j]] = b[j];

return solution\_exist;

}

public void запись\_в\_файл(double[] x)

{

StreamWriter sw = new StreamWriter("out.txt", true);

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

sw.Write("{0} ", x[i]);

sw.WriteLine();

sw.Close();

}

public void запись\_в\_файл(double[,] a, double [] b)

{

StreamWriter sw = new StreamWriter("out.txt", true);

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

{

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

{

sw.Write("{0} ", a[i, j]);

}

sw.Write("{0} ", b[i]);

sw.WriteLine();

}

sw.WriteLine();

sw.Close();

}

public void запись\_в\_файл(int[] x)

{

StreamWriter sw = new StreamWriter("out.txt", true);

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

sw.Write("{0} ", x[i]);

sw.WriteLine();

sw.Close();

}

}

}