ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»

(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК

КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ С РАЦИОНАЛЬНЫМИ КОЭФФИЦИЕНТАМИ МОДИФИЦИРОВАННЫМ МЕТОДОМ ГАУССА

Магистерская диссертация

обучающегося по направлению подготовки 02.04.01 «Математика и компьютерные науки»

очной формы обучения, группы 07001531

Сухомлинова Алексея Дмитриевича

Научный руководитель
доц., к.т.н. Чашин Ю.Г.

Рецензент
доц., к.п.н Ерина Т.А.

БЕЛГОРОД 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Данная тема является актуальной, т.к. решение систем линейных алгебраических уравнений - одна из основных задач вычислительной линейной алгебры. Хотя задача решения именно системы линейных уравнений сравнительно редко представляет самостоятельный интерес для прикладных задач, но от умения эффективно решать данные системы часто зависит сама возможность математического моделирования самых разнообразных процессов с применением ЭВМ. Значительная часть численных методов решения различных (в особенности - нелинейных) задач включает в себя решение систем линейных уравнений как элементарный шаг соответствующего алгоритма.

Научная новизна данной работы состоит в том, чтобы получить точное решение систем линейных алгебраических уравнений модифицированным методом Гаусса без деления на разрешающий элемент, без потери точности в случае рациональных коэффициентов.

Целью магистерской диссертации является модифицирование метода Гаусса для решения системы линейных алгебраических уравнений (СЛАУ) с рациональными коэффициентами, с целью уменьшения возможной погрешности вычислений.

Основными задачами работы являются:

1) Разработка метода приведения СЛАУ с рациональными коэффициентами к виду с целочисленными коэффициентами;
2) Разработка алгоритма модифицированного метода Гаусса без деления на разрешающий элемент;
3) Компьютерная реализация модифицированного метода Гаусса;
4) Тестирование метода и инструментальных средств поддержки.

Объект исследования — системы линейных алгебраических уравнений, их современное состояние и проблемы их решения.

Предмет исследования — решение СЛАУ модифицированным методом Гаусса.

Методы исследования — в данной работе будут рассмотрены основные методы решения СЛАУ, их особенности и сложности реализации.

Достоверность полученных результатов обеспечивается использованием общепризнанных математических методов, сравнением с результатами других авторов, совпадением аналогичных результатов, сравнением теоретических результатов с экспериментальными.

Данная магистерская диссертация состоит из введения, трех разделов, заключения и приложения. Она изложена на 71 страницах машинописного текста, включающего 17 рисунков и список литературных источников из 45 наименований.

Во введении обосновывается актуальность темы магистерской диссертации, научная новизна и формулируется её цель, дается общий обзор содержания работы.

В первом разделе магистерской диссертации «Системы уравнений: понятие, история» анализируются: определение системы линейных алгебраических уравнений, проблемы их решения, их применение в различных научных сферах деятельности, основные методы решения и выбор метода Гаусса в качестве предмета научного исследования.

Во втором разделе «Создание и описание метода» ведется разработка алгоритма модифицированного метода Гаусса, ориентированного на компьютерные вычисления без потери точности в случае рациональных коэффициентов, а также описание блок-схемы алгоритма, языка программирования и среды программной разработки, в которой будет реализован данный метод.

В третьем разделе «Программная реализация» описано выполнение созданного приложения, показаны результаты работы модифицированного алгоритма Гаусса, а также наглядное сравнение модифицированного метода Гаусса с традиционным.

В заключении обсуждается научная и практическая значимость полученных результатов.

В приложении представлен модифицированный автором алгоритм для решения СЛАУ методом Гаусса в случае рациональных коэффициентов, реализованный в среде Visual Studio. Приведен листинг формы приложения и программы, реализующей данный модифицированный метод, а также даны комментарии к ним.

1 СИСТЕМЫ УРАВНЕНИЙ: ИСТОРИЯ, ПОНЯТИЯ

Система уравнений - это условие, состоящее в одновременном выполнении нескольких уравнений относительно нескольких (или одной) переменных [1].

Решением системы уравнений называется упорядоченный набор чисел - значений неизвестных, при подстановке которых каждое уравнение системы обращается в верное равенство.

Системой линейных алгебраических уравнений (СЛАУ) называется система вида:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}$$

Упорядоченный набор значений $\{x_1^0, x_2^0, \ldots, x_n^0\}$ называется решением системы, если при подстановке в уравнения все уравнения превращаются в тождество.

Если система уравнений имеет хотя бы одно решение, то она называется *совместной*. Если система уравнений решений не имеет, то она называется *несовместной*. Если СЛАУ имеет единственное решение, то ее называют *определенной*; если решений больше одного, то – *неопределенной* [2].

Если свободные члены всех уравнений системы равны нулю $b_1=b_2=\ldots=b_n=0$, то система называется *однородной*, в противном случае – *неоднородной*.

Решение практических задач с помощью СЛАУ известно еще со времен древности. Зачатки алгебраических операций просматриваются в вавилонских табличках, египетских папирусах, в «Арифметике» Диофанта.

Задачи, соответствующие современным задачам на составление и решение систем уравнений с несколькими неизвестными, встречаются еще в вавилонских и египетских рукописях II века до н.э., а также в трудах древнегреческих, индийских и китайских мудрецов. В китайском трактате "Математика в девяти книгах" [3] словесно изложены правила решения систем уравнений, были замечены некоторые закономерности при решении.

Система может состоять из алгебраических уравнений, линейных алгебраических уравнений, нелинейных уравнений, дифференциальных уравнений.

Методы решения системы уравнений зависят от типа системы. Например, решения систем линейных алгебраических уравнений хорошо известны (метод Крамера, метод Гаусса, матричный метод, метод итераций и т.д.). Для нелинейных же систем общего аналитического решения не найдено, они решаются разного рода численными методами. Аналогично дело обстоит и с системами дифференциальных уравнений.

Системы линейных уравнений широко используются в задачах экономики, физики, химии и других науках.

1.1 Общая характеристика проблем решения систем линейных алгебраических уравнений (СЛАУ)

К решению систем линейных алгебраических уравнений сводится подавляющее большинство задач вычислительной математики. В настоящее время предложено огромное количество алгоритмов решения таких систем [4].

Все методы решения линейных алгебраических уравнений можно разделить на две большие группы: прямые и итерационные. В прямых (или точных) методах решение системы находится за конечное число арифметических действий. Итерационные методы позволяют найти за конечное число итераций приближенное решение системы с любой наперед заданной точностью ε.

Примером прямого метода решения СЛАУ служит метод Крамера [5], в соответствии с которым:

$$x_i = \frac{\Delta_i}{\Delta}, \; i = \overline{1,n}. \tag{1.1}$$

Однако на практике этот метод не используется, так как он требует выполнения очень большого количества арифметических операций. Большая часть существующих прямых методов укладывается в следующую схему. Пусть задана система

$$Ax = b \tag{1.2}$$

линейных алгебраических уравнений. Умножим обе части равенства (1.2) слева на такие матрицы $L_1, L_2, \ldots, L_k$, при которых новая система

$$L_k L_{k-1} \ldots L_2 L_1 A x = L_k L_{k-1} \ldots L_2 L_1 b \tag{1.3}$$

равносильна исходной и легко решается. Для этого достаточно, чтобы матрица $L_k L_{k-1} \ldots L_2 L_1 A$ была треугольной или диагональной. Методы, основанные на подобных преобразованиях, составляют в настоящее время самую значительную группу среди численных методов задач алгебры.

Одним из старейших является метод Гаусса, в основе которого лежит идея последовательного исключения неизвестных [6,7]. Он использует левые треугольные матрицы Li и позволяет свести исходную систему уравнений к системе с правой треугольной матрицей. Этот метод легко реализуется на компьютере, его схема с выбором главного элемента позволяет решать системы с произвольной невырожденной матрицей, а компактная схема – получить результаты с повышенной точностью. Среди всех прямых методов метод Гаусса требует минимального объема вычислений.

Непосредственно к методу Гаусса примыкают метод Жордана и метод оптимального исключения. Эти методы используют треугольные матрицы Li (как левые, так и правые) и позволяют привести исходную систему к системе с диагональной матрицей. Метод оптимального исключения позволяет при заданном объеме оперативной памяти решать системы более высокого порядка, чем метод Жордана.

Перечисленные методы входят в группу методов исключения. Это название объясняется тем, что при каждом умножении на матрицу Li в матрице системы исключается один или несколько элементов. Существуют методы решения систем, которые сочетают в себе как свойства прямых методов, так и итерационных. Как итерационные они построены на минимизации некоторого функционала, достигающего своего минимума на решении системы (1.2). Однако итерации обрываются не позднее, чем на n-ом шаге (n - порядок системы), давая точный ответ. К таким методам относится метод сопряженных градиентов [8].

1.2 Решение системы 2-х линейных уравнений с 2-мя неизвестными. Понятие определителя 2-го порядка

Изучение систем линейных уравнений было начато Лейбницем около 1678 г.; он использовал индексы в случае системы 3-х уравнений с 2-мя неизвестными. Лейбниц исключал обе неизвестные и в результате они *определялись* с помощью выражения специального вида, обращение в ноль которого было условием разрешимости системы. Такое выражение Лейбниц предложил называть *определителем* [9,10]. Позднее, в 1748 г. Маклорен получил явные формулы для решения систем в случаях m=n=2 и m=n=3. Последуем рассуждениям великих ученых.

Рассмотрим систему двух линейных уравнений с двумя неизвестными:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}.$$

Решением этой системы уравнений является пара чисел (x_1,x_2), обращающая оба уравнения системы в тождество.

Для нахождения решения применим метод алгебраического сложения с последующим исключением неизвестных. Для получения x_1 (исключения x_2) из данной системы уравнений умножим первое уравнение системы на a_{22}, а второе на a_{12}, получим:

$$\begin{cases} a_{11}a_{22}x_1 + a_{12}a_{22}x_2 = b_1a_{22} \\ a_{21}a_{12}x_1 + a_{22}a_{12}x_2 = b_2a_{12} \end{cases}.$$

Вычтем почленно из 1-го уравнения второе:

$(a_{11}a_{22} - a_{21}a_{22})x_1 + (a_{12}a_{22} - a_{22}a_{12})x_2 = b_1a_{22} - b_2a_{12}$,
т.е. $(a_{11}a_{22} - a_{21}a_{22})x_1 = b_1a_{22} - b_2a_{12}$.

Полагая выражение $a_{11}a_{22} - a_{21}a_{12} \neq 0$, получаем:

$$x_1 = \frac{b_1a_{22} - b_2a_{12}}{a_{11}a_{22} - a_{21}a_{12}}.$$

Аналогично, исключая из системы уравнений x_1 (умножая 1-е уравнение системы на a_{21}, а 2-е - на a_{11} и вычитая из второго первое), получим:

$$x_2 = \frac{b_2a_{11} - b_1a_{21}}{a_{11}a_{22} - a_{21}a_{12}}.$$

Непосредственной проверкой легко убедиться, что полученные значения x_1 и x_2 действительно удовлетворяют исходной системе, следовательно, являются решением.

Определение. Выражение $a_{11}a_{22} - a_{21}a_{12}$, являющееся знаменателем дробей, определяющих значения неизвестных x_1 и x_2, называется *определителем (детерминантом) 2-го порядка* [9,10], обозначается символом Δ или D (det) и символически записывается в виде квадратной таблицы из 4-х чисел, называемых *элементами* определителя:

$$\Delta = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}.$$

Если элементы $a_{11}, a_{12}, a_{21}, a_{22}$ являются коэффициентами при неизвестных в рассматриваемой системе, Δ называется *основным* определителем системы.

Определитель 2-го порядка имеет 2 строки и 2 столбца, которые называются *рядами*, следовательно, *порядок* определителя равен количеству его строк или столбцов. Говорят, элементы с одинаковыми индексами a_{11}, a_{22} лежат на *главной* диагонали определителя, а с разными a_{21}, a_{12} – на *побочной (вспомогательной).*

Таким образом, определитель 2-го порядка равен разности произведений элементов, стоящих на главной и побочной диагоналях. Это *практическое правило* вычисления определителя 2-го порядка [10].

Примеры вычисления определителей 2-го порядка:

$$\Delta_1 = \begin{vmatrix} 3 & 4 \\ 2 & 1 \end{vmatrix} = 3 \cdot 1 - 4 \cdot 2 = 3 - 8 = -5$$

$$\Delta_2 = \begin{vmatrix} \sin\alpha & \cos\alpha \\ \sin\beta & \cos\beta \end{vmatrix} = \sin\alpha\cos\beta - \cos\alpha\sin\beta = \sin(\alpha - \beta)$$

$$\Delta_3 = \begin{vmatrix} x-1 & y-1 \\ x^2 & y^2 \end{vmatrix} = (x-1)y^2 - x^2(y-1) = xy^2 - y^2 - x^2 y + x^2 = (y-x)(xy - y - x)$$

Приняв введенное нами понятие определителя 2-го порядка, замечаем, что числители в формулах для нахождения x_1 и x_2 также могут быть представлены в виде определителей 2-го порядка, которые мы назовем *вспомогательными* и обозначим Δ_1 и Δ_2:

$$\Delta_1 = \begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}, \ \Delta_2 = \begin{vmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{vmatrix}.$$

Легко заметить, что вспомогательный определитель Δ_1 получается из основного определителя путем замены столбца коэффициентов при неизвестной x_1 столбцом правых частей. Аналогично составляется Δ_2.

Тогда формулы для нахождения решения исходной системы линейных уравнений принимают следующий вид:

$$\begin{cases} x_1 = \dfrac{\Delta_1}{\Delta} \\ x_2 = \dfrac{\Delta_2}{\Delta} \end{cases}, \Delta \neq 0.$$

Впервые в таком виде они были записаны Крамером в 1754 г., поэтому носят название *формул Крамера* [11] для системы двух линейных уравнений с двумя неизвестными.

Замечание: часто в случае СЛАУ небольших размеров $m = n = 2 = 3$ неизвестные обозначают x, y, z.

При решении подобных систем могут быть следующие случаи:

1) $\Delta \neq 0$. Тогда система *совместна* и имеет *единственное решение*.

Например, $\begin{cases} 2x_1 - 3x_2 = 1 \\ 3x_1 + 5x_2 = 11 \end{cases}$. $\Delta = \begin{vmatrix} 2 & -3 \\ 3 & 5 \end{vmatrix} = 2 \cdot 5 - 3 \cdot (-3) = 19,$

$$\Delta_1 = \begin{vmatrix} 1 & -3 \\ 11 & 5 \end{vmatrix} = 1 \cdot 5 - 11 \cdot (-3) = 38, \ \Delta_2 = \begin{vmatrix} 2 & 1 \\ 3 & 11 \end{vmatrix} = 2 \cdot 11 - 1 \cdot 3 = 19,$$

$x_1 = \dfrac{\Delta_1}{\Delta} = \dfrac{38}{19} = 2,\ x_2 = \dfrac{\Delta_2}{\Delta} = \dfrac{19}{19} = 1.$Проверка: $\begin{cases} 2 \cdot 2 - 3 \cdot 1 = 1 \\ 3 \cdot 2 + 5 \cdot 1 = 11 \end{cases}, \begin{cases} 1 = 1 \\ 11 = 11 \end{cases}.$

2) $\Delta = 0, \Delta_1 = 0, \Delta_2 = 0.$ Система *совместна* и имеет *бесчисленное множество* решений. Фактически система сводится к одному уравнению (второе – следствие первого), из которого неизвестные однозначно не определяются.

Например, $\begin{cases} 2x+5y=3 \\ 4x+10y=6 \end{cases}$. Здесь $\Delta = \begin{vmatrix} 2 & 5 \\ 4 & 10 \end{vmatrix} = 20-20=0,$

$\Delta_x = \begin{vmatrix} 3 & 5 \\ 6 & 10 \end{vmatrix} = 30-30=0, \Delta_y = \begin{vmatrix} 2 & 3 \\ 4 & 6 \end{vmatrix} = 12-12=0.$

3) $\Delta=0,$ но $\Delta_x \neq 0,$ или $\Delta_y \neq 0$. Тогда система *несовместна*.

Например, $\begin{cases} 5x+3y=7 \\ 10x+6y=2 \end{cases}$. $\Delta = \begin{vmatrix} 5 & 3 \\ 10 & 6 \end{vmatrix} = 30-30=0,$

$\Delta_x = \begin{vmatrix} 7 & 3 \\ 2 & 6 \end{vmatrix} = 42-6=36 \neq 0, \Delta_y = \begin{vmatrix} 5 & 7 \\ 10 & 2 \end{vmatrix} = 10-70=-60 \neq 0.$

Система несовместна, т.е. не имеет решений.

Основным недостатком метода Крамера (если это можно назвать недостатком) является трудоемкость вычисления определителей, когда число уравнений системы больше трех.

1.3 Матричный способ решения СЛАУ

Введенные нами операции над матрицами позволяют:

1) предложить матричную форму записи СЛАУ.

Для этого матрицу, составленную из коэффициентов при неизвестных системы m линейных уравнений с n неизвестными

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

назовем *основной матрицей системы* [9,10].

Ее размерность $(m \times n)$.

Основная матрица системы, дополненная столбцом свободных членов, называется *расширенной матрицей системы* [12,13]. Она имеет вид:

$$A_p = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{pmatrix},$$

ее размерность $m \times (n+1)$.

Обозначим матрицу – столбец, элементы которой – неизвестные системы, через X (ее размерность $n \times 1$), а матрицу – столбец, элементы которой – свободные члены системы, B (ее размерность $m \times 1$):

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}.$$

Тогда систему уравнений можно записать в матричной форме:

$A \cdot X = B.$

2) получить еще один способ решения СЛАУ для случая m=n с невырожденной матрицей системы, который называется *матричным способом* [14]. В случае $\Delta = |A| \neq 0$ существует обратная матрица A^{-1}. Запишем систему в матричной форме: $A \cdot X = B.$

Умножим слева обе части равенства на A^{-1}. Получим:

$A^{-1} \cdot A \cdot X = A^{-1} \cdot B.$

На основании свойств произведения матриц и определения обратной матрицы преобразуем левую часть равенства:

$A^{-1} \cdot A \cdot X = (A^{-1} \cdot A) \cdot X = E \cdot X = X,$

где X – матрица–столбец из неизвестных системы. Тогда решение системы уравнений имеет вид:

$X = A^{-1} \cdot B.$

Пример. Решить систему уравнений матричным способом:

$$\begin{cases} 3x + y - z = 2 \\ 2x - y + 4z = 5. \\ x + 2y - 3z = 0 \end{cases}$$

Запишем для данной системы уравнений матрицы A, X, B:

$$A = \begin{pmatrix} 3 & 1 & -1 \\ 2 & -1 & 4 \\ 1 & 2 & -3 \end{pmatrix}; \quad X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}; \quad B = \begin{pmatrix} 2 \\ 5 \\ 0 \end{pmatrix}.$$

Для нахождения обратной матрицы A^{-1} найдем основной определитель системы Δ и алгебраические дополнения каждого элемента матрицы A:

$$\Delta = \begin{vmatrix} 3 & 1 & -1 \\ 2 & -1 & 4 \\ 1 & 2 & -3 \end{vmatrix} = 3\begin{vmatrix} -1 & 4 \\ 2 & -3 \end{vmatrix} - 1\begin{vmatrix} 2 & 4 \\ 1 & -3 \end{vmatrix} - 1\begin{vmatrix} 2 & -1 \\ 1 & 2 \end{vmatrix} = -10 \neq 0.$$

$$A_{11} = \begin{vmatrix} -1 & 4 \\ 2 & -3 \end{vmatrix} = -5; \quad A_{12} = -\begin{vmatrix} 2 & 4 \\ 1 & -3 \end{vmatrix} = 10; \quad A_{13} = \begin{vmatrix} 2 & -1 \\ 1 & 2 \end{vmatrix} = 5;$$

$$A_{21} = -\begin{vmatrix} 1 & -1 \\ 2 & -3 \end{vmatrix} = 1; \quad A_{22} = \begin{vmatrix} 3 & -1 \\ 1 & -3 \end{vmatrix} = -8; \quad A_{23} = -\begin{vmatrix} 3 & 1 \\ 1 & 2 \end{vmatrix} = -5;$$

$$A_{31} = \begin{vmatrix} 1 & -1 \\ -1 & 4 \end{vmatrix} = 3; \quad A_{32} = -\begin{vmatrix} 3 & -1 \\ 2 & 4 \end{vmatrix} = -14; \quad A_{33} = \begin{vmatrix} 3 & 1 \\ 2 & -1 \end{vmatrix} = -5.$$

Составим матрицу из алгебраических дополнений

$$A^* = \begin{pmatrix} -5 & 10 & 5 \\ 1 & -8 & -5 \\ 3 & -14 & -5 \end{pmatrix}.$$ Транспонируем ее $$(A^*)' = \begin{pmatrix} -5 & 1 & 3 \\ 10 & -8 & -14 \\ 5 & -5 & -5 \end{pmatrix}.$$

Найдем $A^{-1} = \dfrac{1}{\Delta} \cdot (A^*)'$.

$$A^{-1}=-\frac{1}{10}\begin{pmatrix}-5 & 1 & 3\\ 10 & -8 & -14\\ 5 & -5 & -5\end{pmatrix}=\begin{pmatrix}\frac{1}{2} & -\frac{1}{10} & -\frac{3}{10}\\ -1 & \frac{4}{5} & \frac{7}{5}\\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2}\end{pmatrix}.$$

$$X=A^{-1}\cdot B\Rightarrow\begin{pmatrix}x\\ y\\ z\end{pmatrix}=\begin{pmatrix}\frac{1}{2} & -\frac{1}{10} & -\frac{3}{10}\\ -1 & \frac{4}{5} & \frac{7}{5}\\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2}\end{pmatrix}\cdot\begin{pmatrix}2\\ 5\\ 0\end{pmatrix}=\begin{pmatrix}1-\frac{1}{2}\\ -2+4\\ -1+\frac{5}{2}\end{pmatrix}=\begin{pmatrix}\frac{1}{2}\\ 2\\ \frac{3}{2}\end{pmatrix}.$$

$$\begin{pmatrix}x\\ y\\ z\end{pmatrix}=\begin{pmatrix}\frac{1}{2}\\ 2\\ \frac{3}{2}\end{pmatrix}\Rightarrow\begin{cases}x=\frac{1}{2}\\ y=2.\\ z=\frac{3}{2}\end{cases}$$

Ответ: $\left(\frac{1}{2},2,\frac{3}{2}\right)$.

1.4 Метод Гаусса

Большим недостатком двух рассмотренных методов (формулы Крамера и матричный метод) является их большая трудоемкость, связанная с громоздкими вычислениями при большом значении n. Возможность применения этих методов ограничена также размерностью системы уравнений $m=n$ и требованием невырожденности матрицы системы.

Обойти эти сложности помогает универсальный метод решения СЛАУ - *метод Гаусса*, который еще называют *методом последовательного исключения неизвестных* [14,15,16].

Условно в методе Гаусса можно выделить 2 этапа: «прямой» и «обратный» ходы.

Прямой ход предполагает последовательное исключение неизвестных из уравнений системы, начиная с 1-го (движение сверху вниз).

Обратный ход – последовательное нахождение значений неизвестных, начиная с последнего уравнения системы (движение снизу вверх).

В основу метода Гаусса положены свойства *равносильных (эквивалентных)* систем (т.е. имеющих одни и те же решения) [16].

Примем без доказательства тот факт, что система переходит в ей равносильную в результате *элементарных преобразований*, к которым относятся:

- перестановка уравнений местами;

- умножение всех членов уравнения на одно и то же число, отличное от нуля;

- прибавление к обеим частям одного уравнения соответствующих частей другого уравнения, умноженных на одно и то же число.

Таким образом, суть метода Гаусса заключается в том, что с помощью элементарных преобразований СЛАУ приводится к равносильной системе ступенчатого вида, из которой затем последовательно находятся значения переменных, начиная с последнего уравнения.

Замечание. Преобразования по схеме Гаусса удобно проводить не с самими уравнениями, а с расширенной матрицей системы:

$$A_p = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{pmatrix}.$$

В этом случае элементарные преобразования производятся над строками [11]. Тогда, прямой ход будет состоять в получении нулей под главной диагональю. Затем по расширенной матрице восстанавливают систему и обратным ходом последовательно находят значения переменных.

Алгоритм метода Гаусса цикличен, т.е. одна и та же операция будет по очереди повторяться для каждой неизвестной [16,17]. Условно его можно выразить фразой: «диагональный элемент не ноль, под ним нули».

Рассмотрим СЛАУ (случай $m \neq n$):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}. \qquad (1.4)$$

Пусть $a_{11} \neq 0$ (если $a_{11} = 0$ то перестановкой уравнений местами добьемся, чтобы выполнялось условие $a_{11} \neq 0$. Это приведет к равносильной системе уравнений). Разделим 1-е уравнение системы на a_{11}. Оно примет вид:

$$x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + \ldots + a_{1n}^{(1)}x_n = b_1^{(1)},$$

где $a_{12}^{(1)} = \dfrac{a_{12}}{a_{11}}$; $a_{13}^{(1)} = \dfrac{a_{13}}{a_{11}} \ldots a_{1n}^{(1)} = \dfrac{a_{1n}}{a_{11}}$; $b_1^{(1)} = \dfrac{b_1}{a_{11}}$.

1. Умножая это уравнение последовательно на $-a_{21}, -a_{31} \ldots -a_{m1}$ и прибавляя полученные уравнения в том же порядке ко 2-му, 3-му … m-му уравнениям, исключим переменную x_1 из всех уравнений системы, начиная со второго.

Получим:

$$\begin{cases} x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + \ldots + a_{1n}^{(1)}x_n = b_1^{(1)} \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 \ldots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m2}^{(1)}x_2 + a_{m3}^{(1)}x_3 + \ldots + a_{mn}^{(1)}x_n = b_m^{(1)} \end{cases}. \qquad (1.5)$$

2. Пусть $a_{22}^{(1)} \neq 0$ (если это не так, то перестановкой уравнений или неизвестных с изменением их номеров добьемся, чтобы выполнялось условие $a_{22}^{(1)} \neq 0$).

Разделим второе уравнение системы на $a_{22}^{(1)}$. Это уравнение примет вид

$$x_2 + a_{23}^{(2)} x_3 + \ldots + a_{2n}^{(2)} x_n = b_2^{(2)},$$

где $\quad a_{23}^{(2)} = \dfrac{a_{23}^{(1)}}{a_{22}^{(1)}} \ldots a_{2n}^{(2)} = \dfrac{a_{2n}^{(1)}}{a_{22}^{(1)}};\ b_2^{(2)} = \dfrac{b_2^{(1)}}{a_{22}^{(1)}}.$

Умножая полученное уравнение последовательно на $-a_{32}^{(1)} \ldots -a_{m2}^{(1)}$ и прибавляя полученные уравнения в том же порядке к третьему … m-му уравнению системы, исключим переменную x_2 из всех уравнений системы, начиная с третьего.

Продолжая процесс последовательного исключения переменных $x_3, x_4 \ldots x_{r-1}$ после r-го шага получим систему вида

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + \ldots a_{1r}^{(1)} x_r + a_{1r+1}^{(1)} x_{r+1} + \ldots + a_{1n}^{(1)} x_n = b_1^{(1)} \\ x_2 + a_{23}^{(2)} x_3 + \ldots + a_{2r}^{(2)} x_r + a_{2r+1}^{(2)} x_{r+1} + \ldots + a_{2n}^{(2)} x_n = b_2^{(2)} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_r + a_{rr+1}^{(r)} x_{r+1} + \ldots + a_{rn}^{(r)} x_n = b_r^{(r)} \\ 0 = b_{r+1}^{(r)} \\ \ldots\ldots\ldots \\ 0 = b_m^{(r)} \end{cases}. \qquad (1.6)$$

Число нуль в последних $m-r$ уравнениях означает, что их левые части имеют вид $0 \cdot x_1 + 0 \cdot x_2 + \ldots + 0 \cdot x_n$. Если в полученной системе уравнений хотя бы одно из чисел $b_{r+1}^{(r-1)} \ldots b_m^{(r-1)}$ не равно нулю, то соответствующее равенство противоречиво, и система (1.6) несовместна.

Для любой совместной системы (1.4) числа $b_{r+1}^{(r)} \ldots b_m^{(r)}$ в системе (1.6) равны нулю. В этом случае последние $m-r$ уравнений являются тождествами и их можно исключить при решении системы (1.6). После исключения из системы (1.6) последних $m-r$ уравнений возможны два случая:

1) число уравнений, оставшихся в системе (1.6), равно числу неизвестных, т.е. $r = n$, тогда система (1.6) имеет треугольный вид и решение

системы (4) единственное;

2) $r < n$ (тогда система (1.6) имеет ступенчатый вид и система (1.4) неопределенная, имеет бесчисленное множество решений).

Пример 1. Решить систему уравнений методом Гаусса:

$$\begin{cases} 2x - 3y + 4z = 20 \\ 3x + 4y - 2z = -11. \\ 4x + 2y + 3z = 9 \end{cases} \quad a_{11} = 2 \neq 0.$$

Составим расширенную матрицу системы

$$A_p = \begin{pmatrix} 2 & -3 & 4 & 20 \\ 3 & 4 & -2 & -11 \\ 4 & 2 & 3 & 9 \end{pmatrix}.$$

Разделим первую строку матрицы на 2.

$$\begin{pmatrix} 1 & -\frac{3}{2} & 2 & 10 \\ 3 & 4 & -2 & -11 \\ 4 & 2 & 3 & 9 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & -\frac{3}{2} & 2 & 10 \\ 3 & 4 & -2 & -11 \\ 4 & 2 & 3 & 9 \end{pmatrix} \begin{array}{l} \cdot(-3) \quad \cdot(-4) \\ \qquad +\hookleftarrow \\ \qquad\quad +\hookleftarrow \end{array} \Rightarrow \begin{pmatrix} 1 & -\frac{3}{2} & 2 & 10 \\ 0 & \frac{17}{2} & -8 & -41 \\ 0 & 8 & -5 & -31 \end{pmatrix}.$$

Разделим вторую строку полученной матрицы на $\frac{17}{2}$:

$$\begin{pmatrix} 1 & -\frac{3}{2} & 2 & 10 \\ 0 & 1 & -\frac{16}{17} & -\frac{82}{17} \\ 0 & 8 & -5 & -31 \end{pmatrix} \begin{array}{l} \\ \cdot(-8) \\ +\hookleftarrow \end{array} \Rightarrow \begin{pmatrix} 1 & -\frac{3}{2} & 2 & 10 \\ 0 & 1 & -\frac{16}{17} & -\frac{82}{17} \\ 0 & 0 & \frac{43}{17} & \frac{129}{17} \end{pmatrix}.$$

Разделим третью строку на $\frac{43}{17}$:

$$\begin{pmatrix} 1 & -\frac{3}{2} & 2 & 10 \\ 0 & 1 & -\frac{16}{17} & -\frac{82}{17} \\ 0 & 0 & 1 & 3 \end{pmatrix}.$$

Прямой ход на этом окончен.

Последней матрице соответствует система уравнений:

$$\begin{cases} x_1 - \frac{3}{2}x_2 + 2x_3 = 10 \\ x_2 - \frac{16}{17}x_3 = -\frac{82}{17} \\ x_3 = 3 \end{cases}.$$

Обратный ход: из третьего уравнения системы $x_3 = 3$;

из второго уравнения системы

$$x_2 = \frac{16}{17}x_3 - \frac{82}{17} = \frac{16}{17}\cdot 3 - \frac{82}{17} = \frac{48-82}{17} = -\frac{34}{17} = -2;$$

из первого уравнения системы

$$x_1 = \frac{3}{2}x_2 - 2x_3 + 10 = \frac{3}{2}\cdot(-2) - 2\cdot 3 + 10 = -3 - 6 + 10 = 1.$$

Ответ: $(1;-2;3)$.

Пример 2. Методом Гаусса решить систему уравнений:

$$\begin{cases} x_1 + 2x_2 - x_3 = 7 \\ 2x_1 - 3x_2 + x_3 = 3 \\ 4x_1 + x_2 - x_3 = 16 \end{cases}.$$

Запишем и преобразуем расширенную матрицу системы

$$\begin{pmatrix} 1 & 2 & -1 & 7 \\ 2 & -3 & 1 & 3 \\ 4 & 1 & -1 & 16 \end{pmatrix} \begin{matrix} \cdot(-2) \quad \cdot(-4) \\ +\hookleftarrow \quad \\ \quad +\hookleftarrow \end{matrix} \Rightarrow \begin{pmatrix} 1 & 2 & -1 & 7 \\ 0 & -7 & 3 & -11 \\ 0 & -7 & 3 & -12 \end{pmatrix} \begin{matrix} \\ \cdot(-1) \\ +\hookleftarrow \end{matrix} \Rightarrow \begin{pmatrix} 1 & 2 & -1 & 7 \\ 0 & -7 & 3 & -11 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

Уравнение, соответствующее третьей строке полученной матрицы, представляет равенство $0 = -1$. Следовательно, система несовместна.

Ответ: решений нет.

Замечания:

1) Кроме решения СЛАУ, методом Гаусса можно вычислять определители. После выполнения прямого хода определитель равен произведению элементов, стоящих на главной диагонали [18,19].

2) Метод Гаусса помогает также находить обратную матрицу через присоединенную: к исходной матрице A приписывается рядом единичная E [20,21]. Затем проводятся элементарные преобразования, приводящие матрицу A к виду единичной, при этом единичная матрица приводится к обратной A^{-1}.

3) При решении СЛАУ методом Гаусса одновременно осуществляется исследование системы [22].

Методом Гаусса можно решать системы линейных алгебраических уравнений любого вида без предварительного их исследования на совместность [23,24]. Процесс последовательного исключения неизвестных переменных позволяет сделать вывод как о совместности, так и о несовместности СЛАУ, а в случае существования решения дает возможность отыскать его.

С точки зрения вычислительной работы метод Гаусса является предпочтительным.

2 СОЗДАНИЕ И ОПИСАНИЕ ПРОЕКТА

2.1 Разработка модифицированного алгоритма Гаусса

Как было ранее сказано, в методе Гаусса можно выделить 2 этапа: «прямой» и «обратный» ходы.

Прямой ход предполагает последовательное исключение неизвестных из уравнений системы, начиная с 1-го (движение сверху вниз).

Обратный ход – последовательное нахождение значений неизвестных [14,15,16], начиная с последнего уравнения системы (движение снизу вверх).

В основу метода Гаусса положены свойства *равносильных (эквивалентных)* систем (т.е. имеющих одни и те же решения) [16].

Суть метода Гаусса заключается в том, что с помощью элементарных преобразований СЛАУ приводится к равносильной системе ступенчатого вида, из которой затем последовательно находятся значения переменных, начиная с последнего уравнения.

На рис. 1 изображена блок-схема модифицированного метода Гаусса.

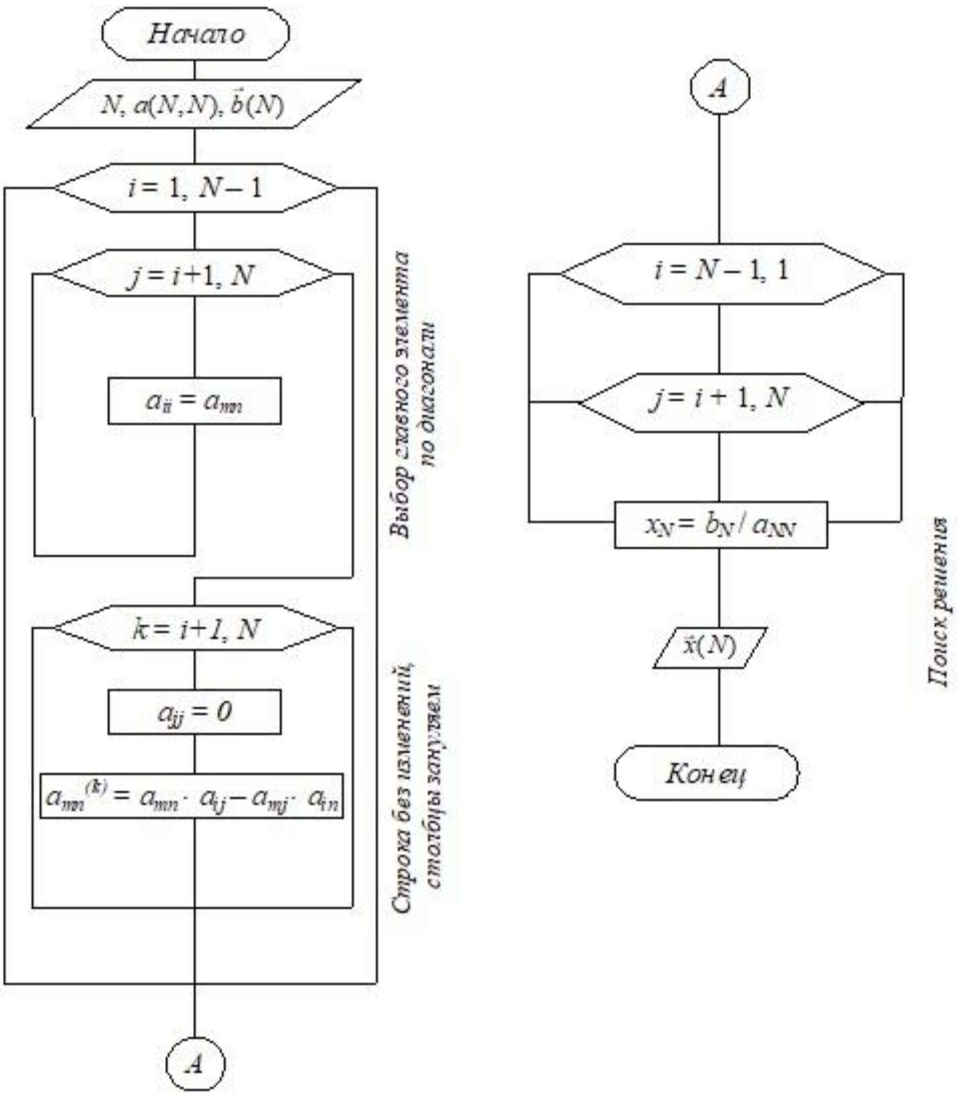

Рис. 2.1. Блок-схема модифицированного метода Гаусса

Рассмотрим матрицу следующего вида:

$$A_p = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{pmatrix}.$$

Остальные элементы находим по формуле

$$a_{mn}^{(k)} = a_{mn} \cdot a_{ij} - a_{mj} \cdot a_{in} \tag{2.1}$$

где $a_{mn}^{(k)}$ – элементы новой матрицы.

Модифицированный алгоритм Гаусса работает следующим образом:

1) Выбираем главный элемент матрицы по диагонали;

2) Переписываем разрешающую строку без изменений;

3) Разрешающий столбец заполняем нулями;

4) Вычисляем остальные элементы по формуле (2.1) без деления на разрешающий элемент.

Таким образом, модифицированный метод позволяет получить точное решение СЛАУ без потери точности и без ошибок.

Пример. Решить СЛАУ модифицированным методом Гаусса:

$$\begin{cases} -2x_1 + 5x_2 - x_3 = -2 \\ x_1 + 3x_2 + 2x_3 = 11 \\ 3x_1 - 2x_2 + 2x_3 = 10 \end{cases}.$$

Составим матрицу системы

$$A = \begin{pmatrix} -2 & 5 & -1 & -2 \\ 1 & 3 & 2 & 11 \\ 3 & -2 & 2 & 10 \end{pmatrix}.$$

Выбираем в качестве главного элемента -2, переписываем разрешающую строку без изменений, разрешающий столбец заполняем нулями, остальные элементы вычисляем по формуле (2.1):

$$\begin{pmatrix} -2 & 5 & -1 & -2 \\ 0 & -11 & -3 & -20 \\ 0 & -11 & -1 & -14 \end{pmatrix}.$$

Теперь в качестве разрешающего элемента берем следующий по диагонали, и проделываем те же шаги:

$$\begin{pmatrix} 22 & 0 & 26 & 122 \\ 0 & -11 & -3 & -20 \\ 0 & 0 & -22 & -66 \end{pmatrix}.$$

Выбираем последний разрешающий элемент, и проводим последнюю итерацию:

$$\begin{pmatrix} -484 & 0 & 0 & 968 \\ 0 & 242 & 0 & 242 \\ 0 & 0 & -22 & -66 \end{pmatrix}.$$

Таким образом, мы получаем следующее решение:

$x_1 = 2;$

$x_2 = 1;$

$x_3 = 3.$

2.2 Обоснование выбора языка программирования

Прогресс компьютерных технологий определил процесс появления новых разнообразных знаковых систем для записи алгоритмов - языков программирования. Смысл появления такого языка - оснащённый набор вычислительных формул дополнительной информации, превращает данный набор в алгоритм. Язык программирования служит двум связанным между собой целям: он даёт программисту аппарат для задания действий, которые должны быть выполнены, и формирует концепции, которыми пользуется программист, размышляя о том, что делать.

C# (говоря на русском, си Шарп) – это объектно-ориентированное программирование [25]. Он был разработан в 2001 году, инженерами под руководством Андерса Хейлсберга в компании Microsoft. На данное время существует 4 версии языка «си Шарп».

Название «Си Шарп» (от англ. sharp - диез) происходит от музыкальной нотации, где знак диез [26], прибавляемый к основному обозначению ноты, означает повышение соответствующего этой ноте звука на полутон. Это аналогично названию языка C++, где «++» обозначает, что переменная должна быть увеличена на 1.

К числу принципиально важных решений, которые реализованы корпорацией Microsoft в языке программирования C#, можно отнести следующие:

1) компонентно-ориентированный подход к программированию (который характерен и для идеологии Microsoft .NET в целом);

2) свойства как средство инкапсуляции данных (характерно также в целом для ООП);

3) обработка событий (имеются расширения, в том числе в части обработки исключений, в частности, оператор try);

4) унифицированная система типизации (соответствует идеологии Microsoft .NET в целом);

5) делегаты (delegate – развитие указателя на функцию в языках C и C++);

6) индексаторы (indexer – операторы индекса для обращения к элементам класса- контейнера);

7) перегруженные операторы (развитие ООП); - оператор foreach (обработка всех элементов классов-коллекций, аналог Visual Basic);

8) механизмы boxing и unboxing для преобразования типов; - атрибуты (средство оперирования метаданными в COM-модели);

9) прямоугольные массивы (набор элементов с доступом по номеру индекса и одинаковым количеством столбцов и строк).

Язык программирования C# имеет следующие достоинства:

1) C# создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности - как FCL, так и CLR;

2) C# является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;

3) C# является мощным объектным языком с возможностями наследования и универсализации;

4) C# является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования [27]. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход программистов от C++ к C#;

5) Сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства C++ как указатели, адресация, разыменование, адресная арифметика;

6) Благодаря каркасу Framework .Net, ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java. Повышается эффективность кода, поскольку исполнительная среда CLR представляет собой компилятор промежуточного языка, в то время как виртуальная Java-машина является интерпретатором байт-кода;

7) Мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;

8) Реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#.

Выделение и объединение лучших идей современных языков программирования делает язык C# не просто суммой их достоинств, а языком программирования нового поколения.

2.3 Обоснование выбора среды разработки

Для реализации поставленной задачи мною была выбрана среда разработки Microsoft Visual Studio, так как она наиболее подходит для создания данного проекта, а именно автоматизированной информационной системы, имеет понятный интерфейс и большой набор функций, инструментов [28].

Microsoft Visual Studio — линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Microsoft Silverlight [29].

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя:

1) редактор форм для упрощения создания графического интерфейса приложения;
2) веб-редактор;
3) дизайнер классов;
4) дизайнер схемы базы данных.

Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения.

Каждая новая версия программы состоит из новейших инструментов и технологий, позволяющих разрабатывать приложения с учетом особенностей и положительных моментов современных платформ [30]. Например, Visual Studio 2012 может поддерживать более ранние версии, в том числе Windows XP и Windows Server 2003. При этом разработчикам открыта дорога к созданию новых и модернизации уже существующих приложений, предназначенных для ранних версий ОС Windows. Стоит отметить, что в процессе использования поддерживаемых системой вариантов исходные файлы, проекты и решения в программе Visual Studio будут работоспособными, но исходный код может нуждаться в изменениях.

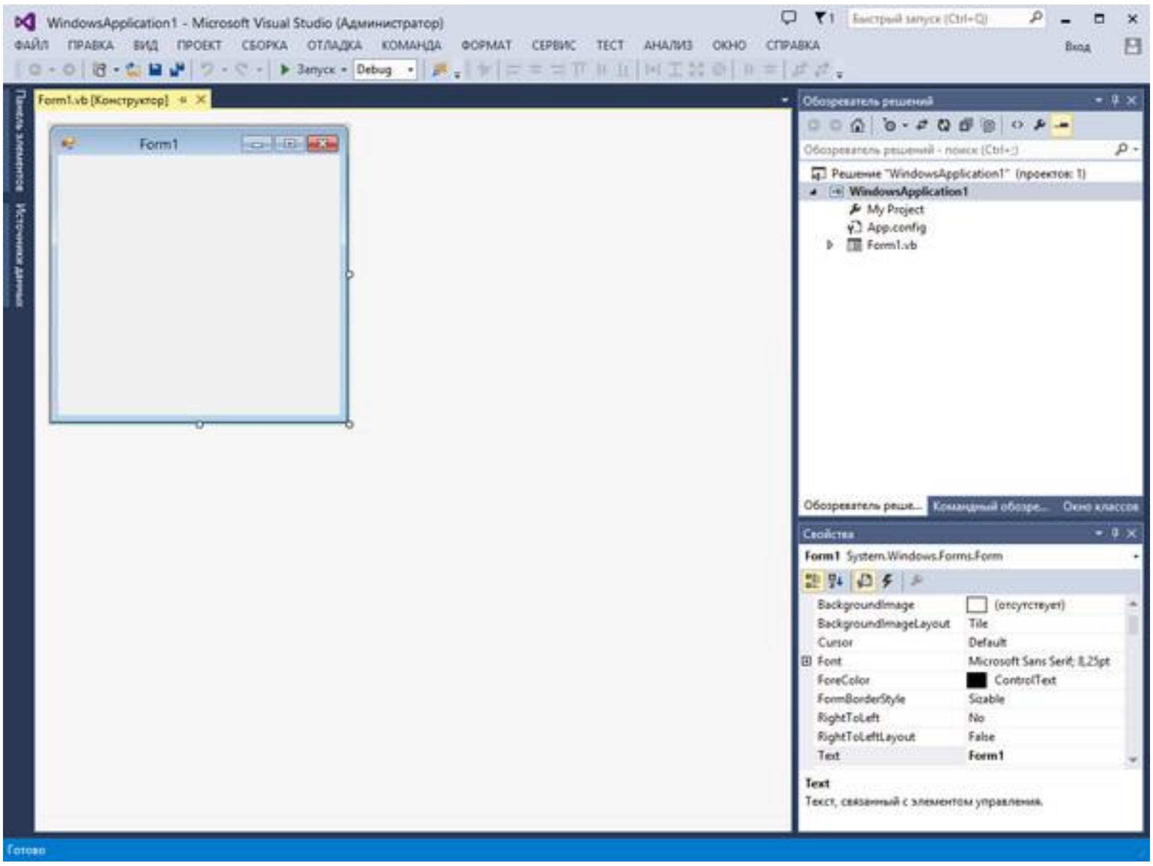

Рис. 2.2. Среда разработки MS Visual Studio

Средства, входящие в состав Visual Studio для Windows, можно использовать для создания привлекательных инновационных приложений для Магазина Windows в среде Windows 8.1. Эти средства включают полнофункциональный редактор кода, мощный отладчик, специальный профилировщик и широкие возможности языковой поддержки, которые позволяют выполнять сборку приложений, написанных на языках HTML5/JavaScript, C++, C# и Visual Basic. В состав Visual Studio для Windows также входит имитатор устройств, который можно использовать для тестирования приложений Магазина Windows на устройствах различных видов [31].

Была выбрана данная среда программирования, потому что она позволяет работать с данными различного типа. Имеет понятный интерфейс, широкий набор инструментов для разработки приложений который расширяется. Интеграция со многими языками программирования и программами, быстро компилирует программный код и выявляет ошибки.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

3.1 Создание приложения

Создаем форму приложения с созданными файлами

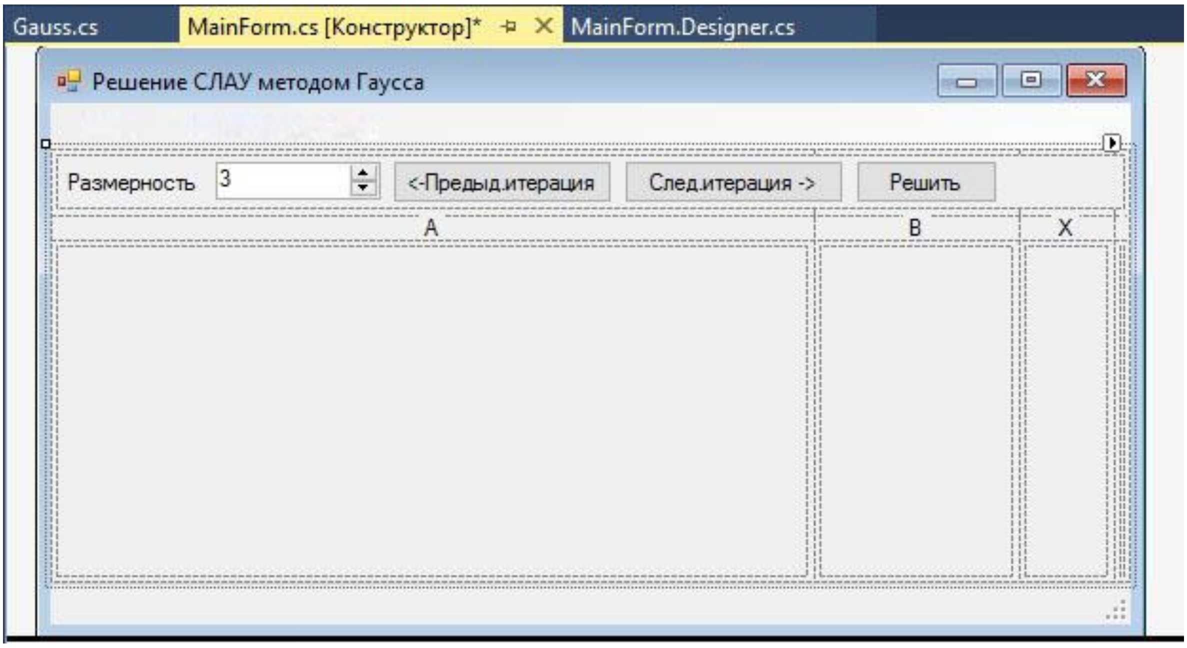

Рис. 3.1. Форма приложения с созданными файлами

В свойствах формы расписываем название, все параметры и т.д., которые показаны на рис.3.2.

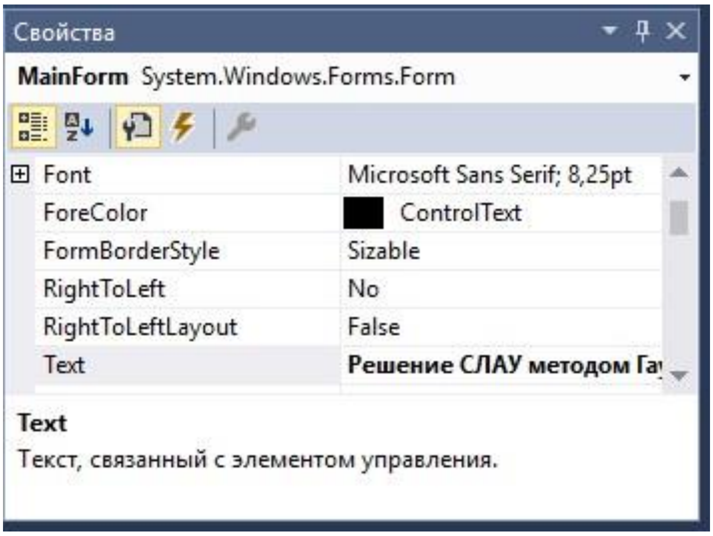

Рис. 3.2. Свойства формы приложения

Более подробно создание формы описано в приложении.

При выполнении программы появляется следующее окно:

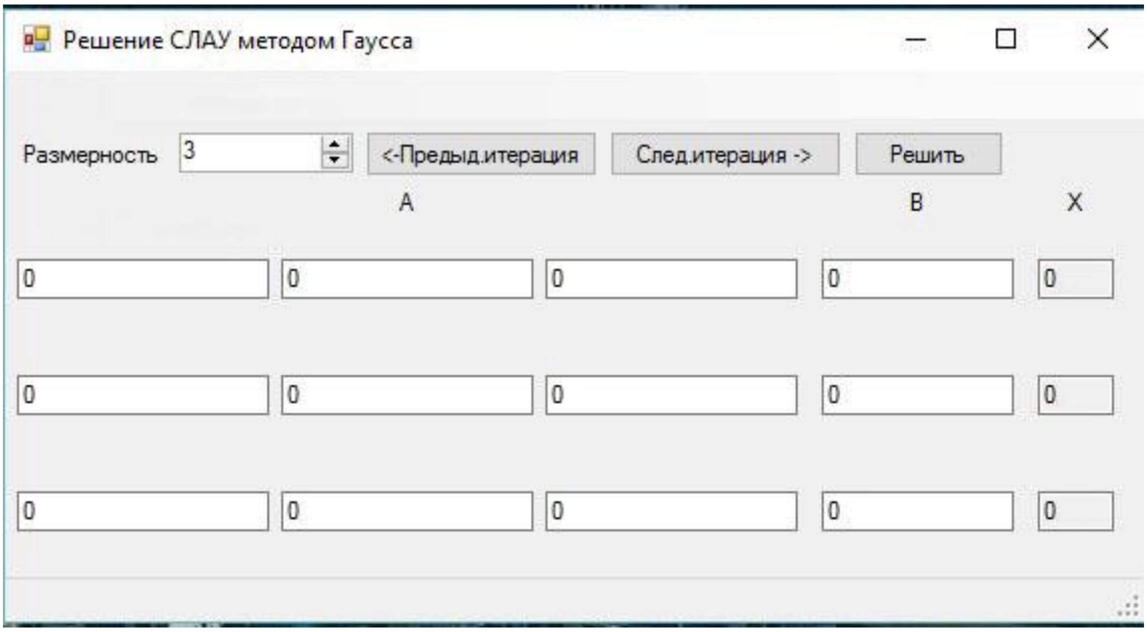

Рис. 3.3. Окно приложения

3.2 Тестирование приложения

Теперь подробнее опишем окно приложения, показанное на рис.3.3.

«Размерность» - количество элементов матрицы и соответственно число итераций, необходимых для получения решения. Его можно регулировать с помощью стрелок, вверх или вниз.

«Предыдущая итерация» - перед выполнением она недоступна, но станет таковой после выполнения первой итерации.

«Следующая итерация» - данная кнопка доступна с самого начала выполнения. Когда останется провести последнюю итерацию, она перестанет быть доступной.

«Решить» - становится доступной перед последней итерацией, когда необходимо найти окончательное решение.

A – матрица системы.

B – столбец свободных членов.

X – решение системы.

Сначала вводим необходимые нам данные.

Листинг 3.1. Реализация модифицированного метода Гаусса (ввод данных)

```
public class LinearSystem {
    private double[,] initial_a_matrix;
    private double[,] a_matrix;  // матрица A
    private double[] x_vector;   // вектор неизвестных x
    private double[] initial_b_vector;
    private double[] b_vector;   // вектор b
```

Затем выбираем разрешающий элемент, проводим необходимое количество итераций и находим решение.

Листинг 3.2. Реализация модифицированного метода Гаусса (решение)

```
                // 1) выбор главного элемента
                double r = FindR(i, index);

                // 2) преобразование матрицы A
                for (int j = 0; j < size; ++j)
                    a_matrix[k, l] =
=a_matrix[k,l]*a_matrix[I,j]-a_matrix[k,j]*a_matrix[i,l];


                for (int k = i + 1; k < size; ++k) {
                    for (int j = i; j < size; ++j)
                    a_matrix[k, index[i]] = 0.0;
                }
            }
        }
                // 3) поиск решения
                for (int j = i + 1; j < size; ++j)
                    x_i = b_matrix[k,l]/a_matrix[k,l];
                x_vector[index[i]] = x_i;
            }
        }
```

Пример 1. Решить СЛАУ размерностью 2х2:

$$\begin{cases} 4x_1 + x_2 = 15 \\ x_1 - 3x_2 = 7 \end{cases}$$

Вводим коэффициенты матрицы и правой части:

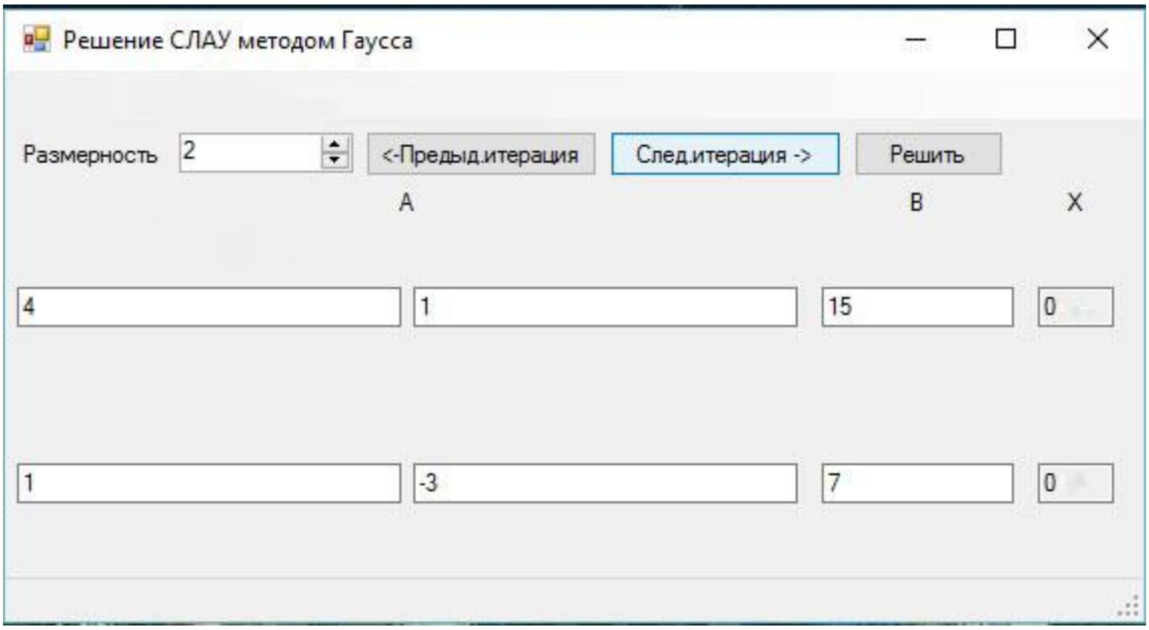

Рис. 3.4. Коэффициенты уравнений

Нажимаем на кнопку «Следующая итерация» и в результате получаем:

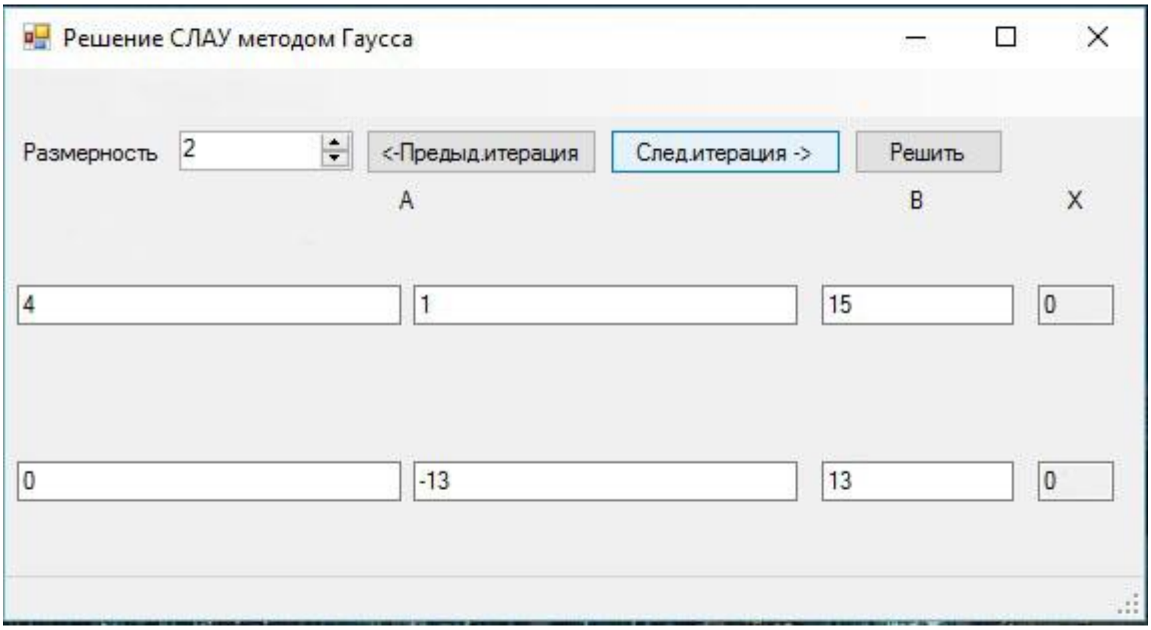

Рис. 3.5. Результаты первой итерации

Нажимаем на кнопку «Решить» и получаем ответ:

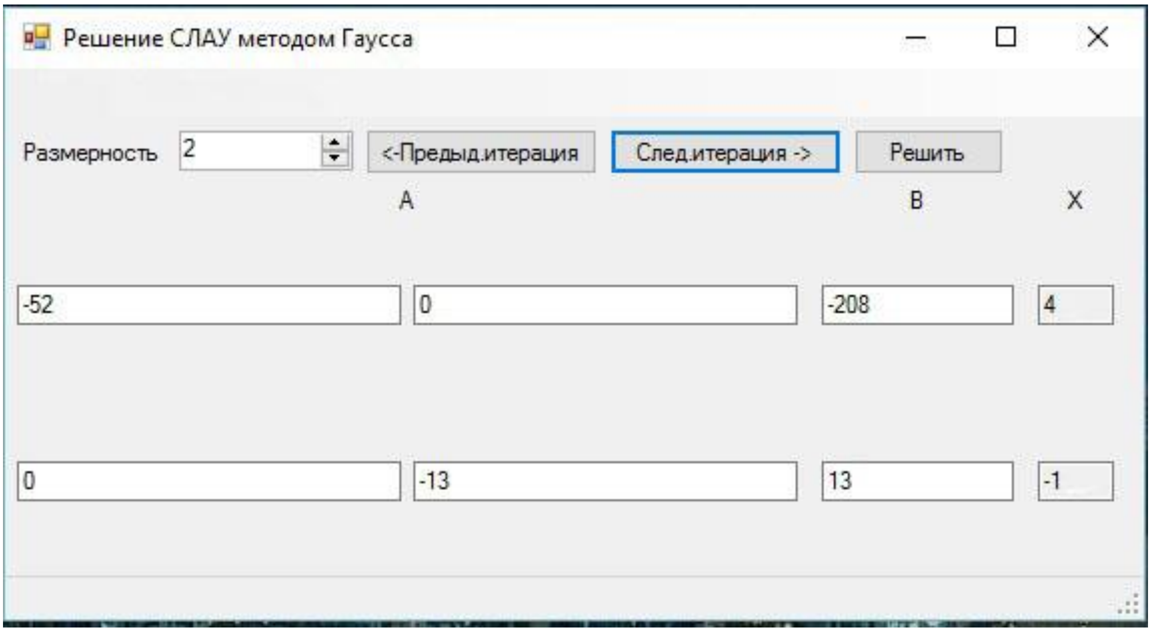

Рис. 3.6. Решение СЛАУ

Пример 2. Решить СЛАУ размерностью 3х3

$$\begin{cases} 2x_1 + 7x_2 - 3x_3 = -5 \\ 3x_1 - 4x_2 + 2x_3 = 20 \\ x_1 + 3x_2 + 6x_3 = 18 \end{cases}$$

Результаты выполнения программы показаны на рисунках 3.7-3.10.

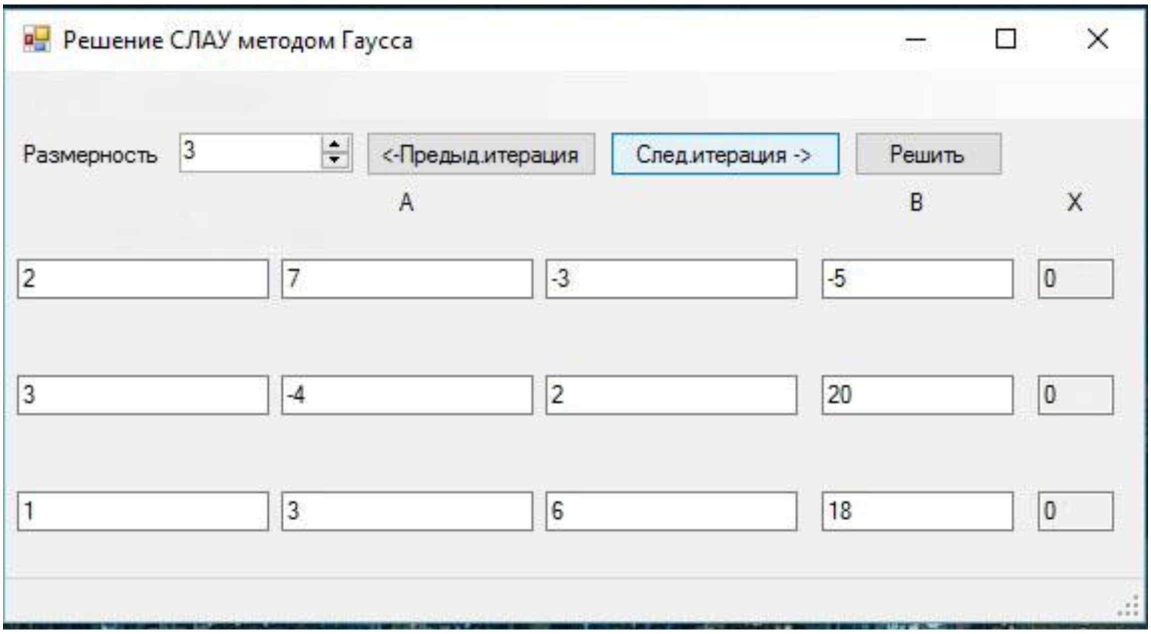

Рис. 3.7. Ввод коэффициентов

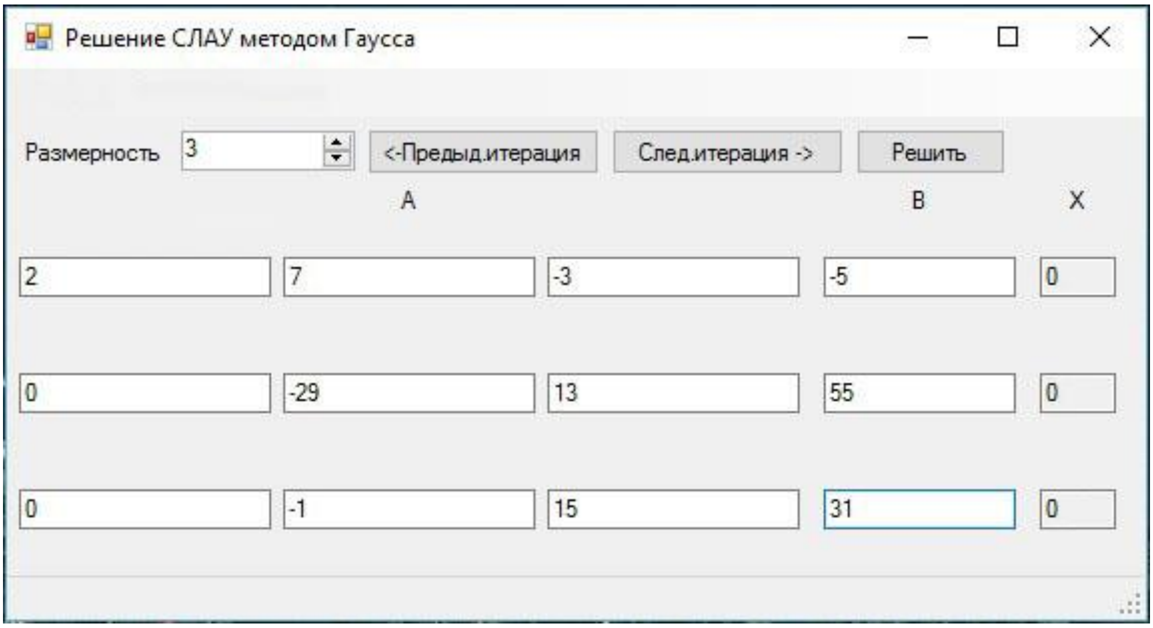

Рис. 3.8. Результаты первой итерации

Нажимаем на кнопку «Следующая итерация» и в результате получаем:

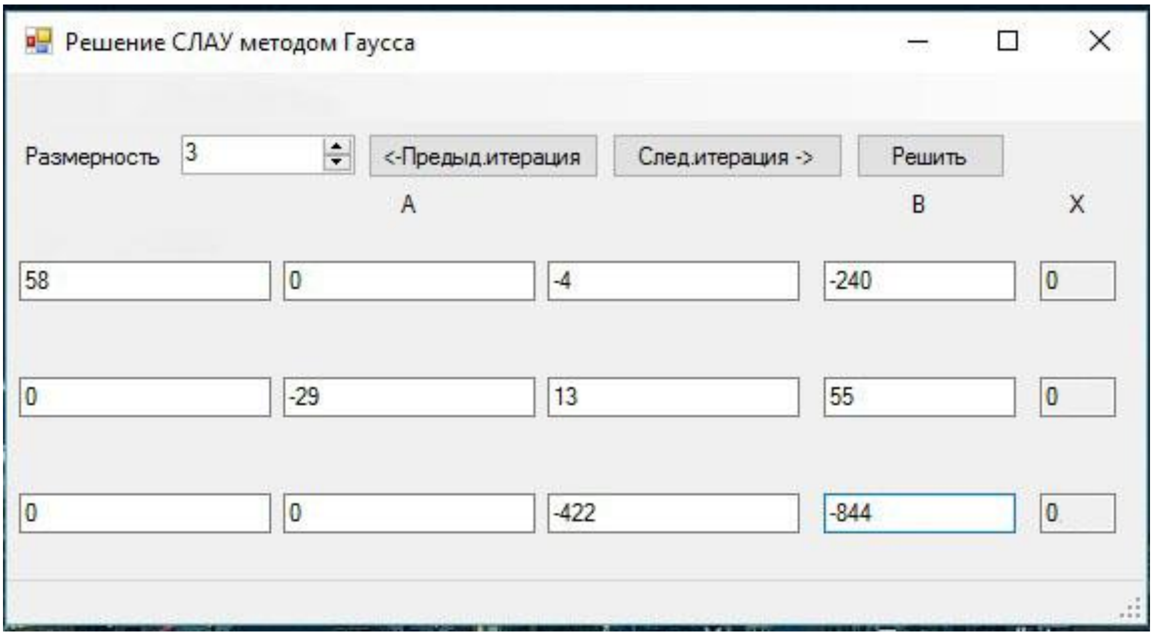

Рис. 3.9. Результаты второй итерации

Нажимаем на кнопку «Решить», в результате получаем следующий ответ:

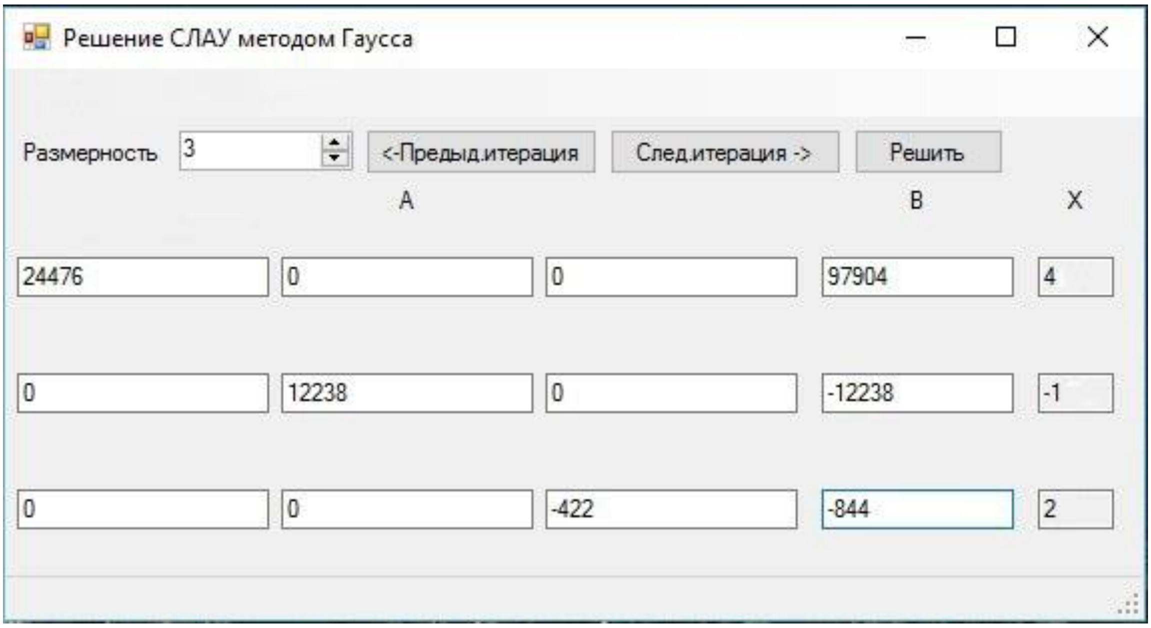

Рис. 3.10. Последняя итерация и получение решения

Пример 3. Решить СЛАУ размерностью 4x4

$$\begin{cases} 2x_1 + 5x_2 - x_3 - 3x_4 = 19 \\ -x_1 + 2x_2 + 5x_3 + 2x_4 = 1 \\ 3x_1 - x_2 - 2x_3 + 4x_4 = 21 \\ x_1 - 3x_2 + 3x_3 - 2x_4 = -21 \end{cases}$$

Результаты выполнения программы показаны на рисунках 3.11-3.15.

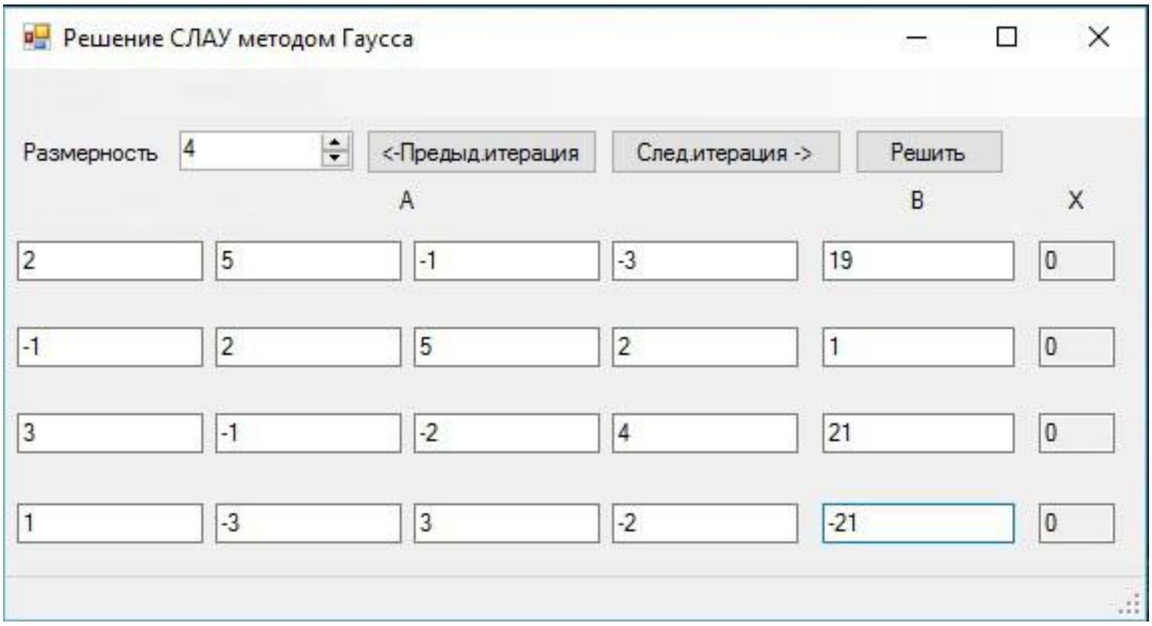

Рис. 3.11. Коэффициенты

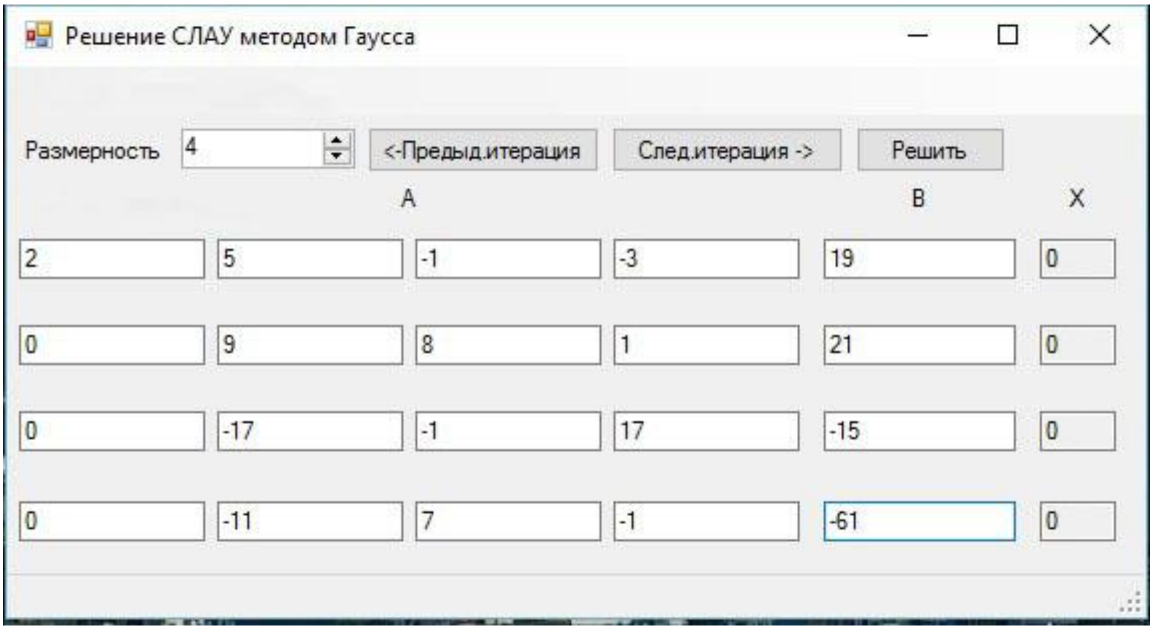

Рис. 3.12. Результаты первой итерации

Нажимаем на кнопку «Следующая итерация» и в результате получаем:

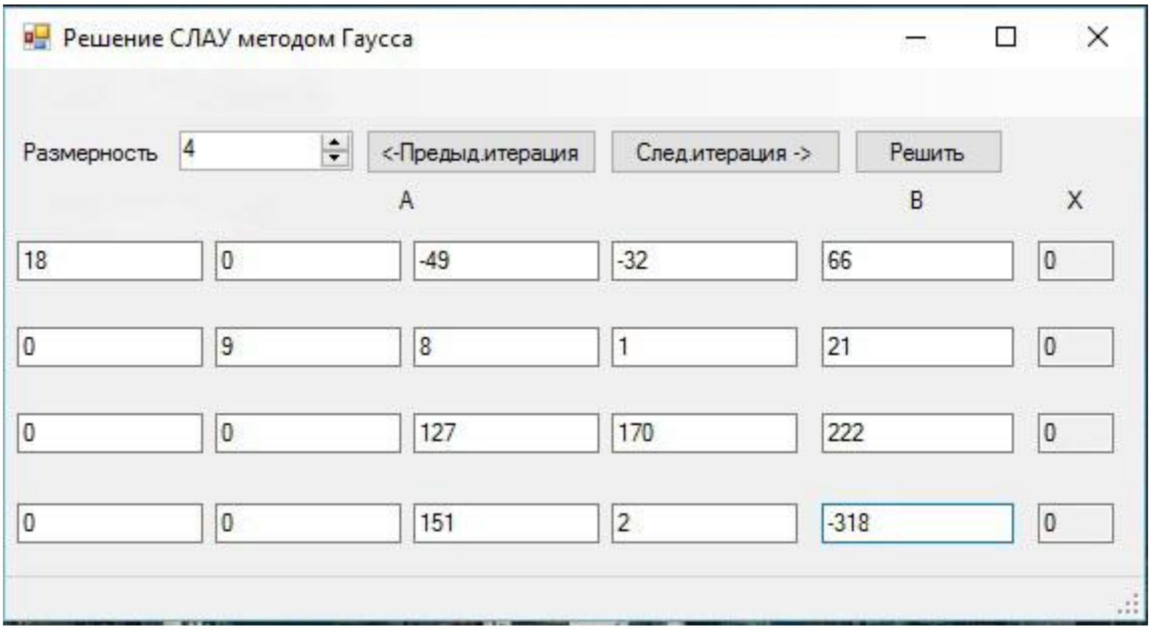

Рис. 3.13. Результаты второй итерации

Снова жмем на кнопку «Следующая итерация», получаем следующий результат:

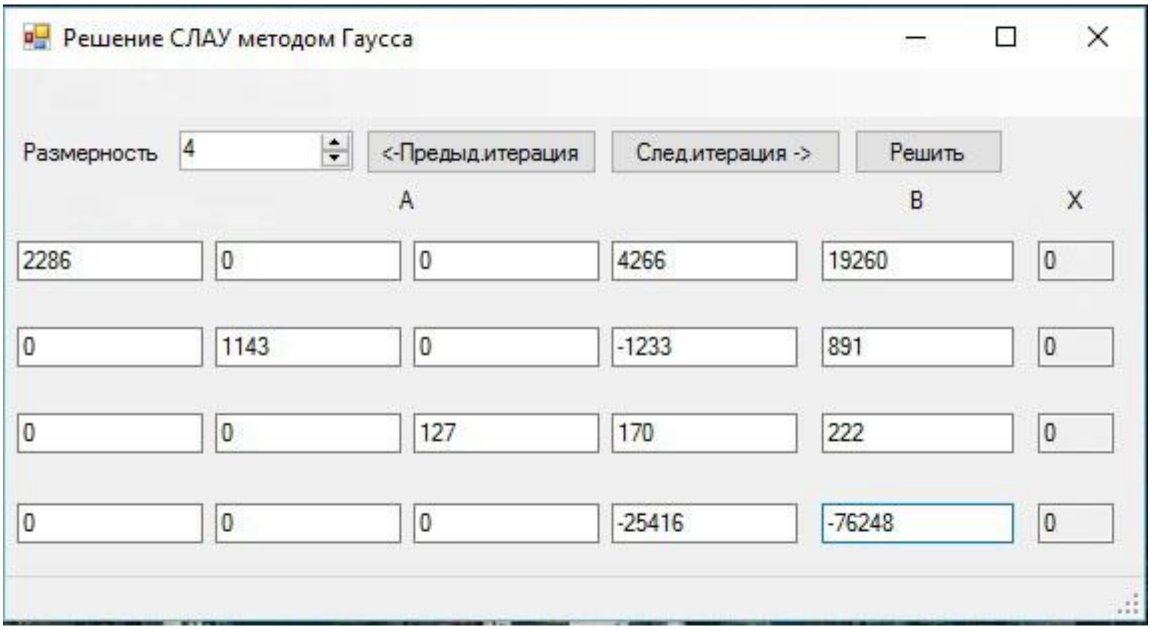

Рис. 3.14. Результаты третьей итерации

Осталась последняя итерация, поэтому нажимаем на кнопку «Решить», и получаем решение

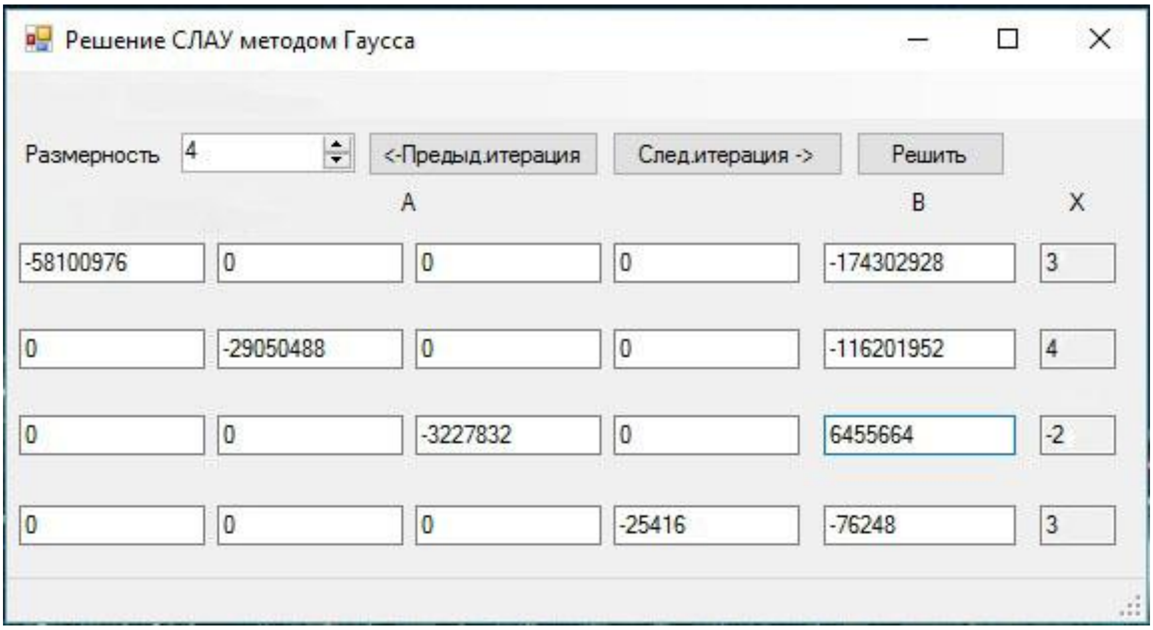

Рис. 3.15. Последняя итерация и получение решения

3.3 Сравнение модифицированного метода Гаусса с традиционным методом

На конкретном примере мы решим СЛАУ размерностью 3х3 традиционным методом и модифицированным, сравним их между собой и сделаем выводы.

Пример. Решить следующую СЛАУ

$$\begin{cases} 3x_1 - 2x_2 + 5x_3 = -5 \\ x_1 + 4x_2 + x_3 = 11 \\ 4x_1 + x_2 - 3x_3 = 14 \end{cases}$$

Традиционный метод Гаусса.

Наша матрица

X1	X2	X3	b
3	-2	5	-5
1	4	1	11
4	1	-3	14

Нашли единицу в 1-ом столбце и поменяли местами 2-ую и 1-ую строки

X1	X2	X3	b
1	4	1	11
3	-2	5	-5

4	1	-3	14

Умножили 1-ую строку на 3

X1	X2	X3	b
3	12	3	33
3	-2	5	-5
4	1	-3	14

Вычли 1-ую строку из 2-ой строки и восстановили ее

X1	X2	X3	b
1	4	1	11
0	-14	2	-38
4	1	-3	14

Умножили 1-ую строку на 4

X1	X2	X3	b
4	16	4	44
0	-14	2	-38
4	1	-3	14

Вычли 1-ую строку из 3-ей строки и восстановили ее

№	X1	X2	X3	b
1	1	4	1	11
2	0	-14	2	-38
3	0	-15	-7	-30

Получили единицу в 2-ом столбце, разделив 2-ую строку на -14

№	X1	X2	X3	b
1	1	4	1	11
2	0	1	-0.14285714285714285	2.7142857142857144
3	0	-15	-7	-30

Умножили 2-ую строку на 4

№	X1	X2	X3	b
1	1	4	1	11
2	0	4	-0.5714285714285714	10.857142857142858
3	0	-15	-7	-30

Вычли 2-ую строку из 1-ой строки и восстановили ее

X1	X2	X3	b
1	0	1.5714285714285714	0.14285714285714235
0	1	-0.14285714285714285	2.7142857142857144
0	-15	-7	-30

Умножили 2-ую строку на -15

X1	X2	X3	b
1	0	1.5714285714285714	0.14285714285714235
0	-15	2.142857142857143	-40.714285714285715
0	-15	-7	-30

Вычли 2-ую строку из 3-ей строки и восстановили ее

X1	X2	X3	b
1	0	1.5714285714285714	0.14285714285714235
0	1	-0.14285714285714285	2.7142857142857144
0	0	-9.142857142857142	10.714285714285715

Получили единицу в 3-ем столбце, разделив 3-ю строку на

-9.142857142857142

X1	X2	X3	b
1	0	1.5714285714285714	0.14285714285714235
0	1	-0.14285714285714285	2.7142857142857144
0	0	1	-1.1718750000000002

Умножили 3-ю строку на 1.5714285714285714

X1	X2	X3	b
1	0	1.5714285714285714	0.14285714285714235
0	1	-0.14285714285714285	2.7142857142857144
0	0	1.5714285714285714	-1.8415178571428574

Вычли 3-ю строку из 1-ой строки и восстановили ее

X1	X2	X3	b
1	0	0	1.9843749999999998
0	1	-0.14285714285714285	2.7142857142857144
0	0	1	-1.1718750000000002

Умножили 3-ю строку на -0.14285714285714285

X1	X2	X3	b
1	0	0	1.9843749999999998
0	1	-0.14285714285714285	2.7142857142857144
0	0	-0.14285714285714285	0.1674107142857143

Вычли 3-ю строку из 2-ой строки и восстановили ее

X1	X2	X3	b
1	0	0	1.9843749999999998
0	1	0	2.546875
0	0	1	-1.1718750000000002

Ответ:

x1 = 1.9843749999999998

x2 = 2.546875

x3 = -1.1718750000000002

Погрешность вычислений:

x_1: 2-1.9843749999999998=0.015625

x_2: 3-2.546875=0.453125

x_3: -1-(-1.1718750000000002)=0.171875

Модифицированный метод.

Составим матрицу системы

$$A=\begin{pmatrix}3 & -2 & 5 & -5\\ 1 & 4 & 1 & 11\\ 4 & 1 & -3 & 14\end{pmatrix}.$$

Выбираем в качестве главного элемента 3, переписываем разрешающую строку без изменений, разрешающий столбец заполняем нулями, остальные элементы вычисляем по формуле (2.1):

$$\begin{pmatrix}3 & -2 & 5 & -5\\ 0 & 14 & -2 & 38\\ 0 & 11 & -29 & 62\end{pmatrix}.$$

Теперь в качестве разрешающего элемента берем следующий по диагонали, и проделываем те же шаги:

$$\begin{pmatrix}42 & 0 & 66 & 6\\ 0 & 14 & -2 & 38\\ 0 & 0 & -384 & 384\end{pmatrix}.$$

Выбираем последний разрешающий элемент, и проводим последнюю итерацию:

$$\begin{pmatrix}-16128 & 0 & 0 & -32256\\ 0 & -5376 & 0 & -16128\\ 0 & 0 & -384 & 384\end{pmatrix}.$$

Таким образом, мы получаем следующее решение:

$x_1 = 2;$

$x_2 = 3;$

$x_3 = -1.$

Погрешность вычислений:

x_1: 2-2=0

x_2: 3-3=0

x_3: -1-(-1)=0

Таким образом, мы смогли получить точное решение без потери точности в рамках рациональных коэффициентов, без наличия ошибок и, самое главное, значительно быстрее, чем общеизвестным методом.

ЗАКЛЮЧЕНИЕ

При выполнении магистерской диссертации были выполнены следующие задачи:

1) было сформулировано определение системы линейных алгебраических уравнений, их применение в различных областях науки;

2) рассмотрены основные методы решения СЛАУ, проблемы их решения, а также обоснован выбор метода Гаусса в качестве предмета научно-исследовательской работы;

3) выбраны и описаны язык программирования и среда программной разработки;

4) разработан алгоритм модифицированного метода Гаусса, ориентированного на компьютерные вычисления без потери точности в случае рациональных коэффициентов.

В приложении приведены листинг формы приложения, а также листинг основной программы на языке C# в среде разработки Visual Studio.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Амосов А. А., Дубинский Ю. А., Копченова Н. П. Вычислительные методы для инженеров. — М.: Мир, 1998.

2. Бахвалов Н. С., Жидков Н. П., Кобельков Г. Г. Численные методы. — 8-е изд. — М.: Лаборатория Базовых Знаний, 2000.

3. Даан-Дальмедико А. Пути и лабиринты. Очерки по истории математики: Пер. с франц./ А. Даан-Дальмедико, Ж. Пейффер - М.: Мир,1986. - 432 с.

4. Беклемишев Д.В. Дополнительные главы линейной алгебры. М.: Наука, 2003. 335 с.

5. Волков Е. А. Численные методы. — М.: Физматлит, 2003.

6. Калиткин Н.Н. и др. Численные методы. М.: Наука, 2002

7. Калиткин Н.Н. Численные методы. М.: Наука, 2003. 512 с.

8. Беклемишев Д.В. Курс аналитической геометрии и линейной алгебры/ Д.В. Беклемишев – М,: Наука, 1974. – 320 с.

9. Боревич З.И. Определители и матрицы: Учеб. Пособие для вузов/ З.И. Боревич – М.: Наука. Гл. ред. физ.-мат. лит., 1986. – 184 с.

10. Райс Дж. Матричные вычисления и математическое обеспечение. М.: Мир, 2004. 264 с.

11. Воробьева Г.Н., Данилова А.Н. Практикум по численным методам. М.: Высш. шк., 2009. 184 с.

12. Годунов С.К. Решение систем линейных уравнений. Новосибирск : Наука,2000.

13. Данко П.Е. Высшая математика в упражнениях и задачах. В 2 –х ч. Ч. I: Учеб. пособие для втузов./П.Е. Данко, А.Г. Попов, Т.Я. Кожевникова – М.: Высш. шк., 1997.– 304с.

14. Е.А. Волков. Численные методы: Учеб. Пособие для вузов - М.: Наука. Гл. ред. физ-мат. лит., 2007. - 248 с.

15. Демидович Б.П., Марон И.А. Основы вычислительной математики. М.: Наука, 2000. 664 с.

16. Ильин В. А., Позняк Э. Г. Линейная алгебра: Учебник для вузов. — 6-е изд., стер. — М.: ФИЗМАТЛИТ, 2004. — 280 с.

17. Каплан И.А. Практические занятия по высшей математике. Ч. 1-3/ И.А. Каплан – Харьков: Изд-во ХГУ, 1967. – 946 с.

18. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. — М.: Наука, 1970. — С. 575-576.

19. Маркович Э.С. Курс высшей математики с элементами теории вероятностей и математической статистики: учеб. Пособие для вузов/ Э.С. Маркович – М.: Высш. школа, 1972. – 480 с.

20. Минорский В.П. Сборник задач по высшей математике. Учеб. пособие для вузов. / В.П.Минорский – М.: Наука, 1977. – 285 с.

21. Р.Ф. Хемминг "Численные методы (для научных работников и инженеров)". - Москва, 2002.

22. Тихонов А.Н., Костомаров Д.П. Вводные лекции по прикладной математике. М.: Наука, 2004. 190 с.

23. Турчак Л.И. Основы численных методов. М.: Наука, 2007

24. Ф.В. Формалев, Д.Л. Ревизников "Численные методы". - М.: ФИЗМАТЛИТ, 2004.

25. Фаронов В.В. Программирование в С#: учеб. пособие / В.В. Фаронов. - 7-е изд., перераб. - М.: Нолидж, 2006. – 412 с.

26. Павловская Т. А. С#. Программирование на языке высокого уровня: Учебник для вузов. - СПб.: БХВ-Петербург. 2007

27. Фаронов В. В., Создание приложений с помощью С#: Руководство программиста. - М.: Эксмо, 2008

28. Карли Уотсон, Кристиан Нейгел, Якоб Хаммер Педерсен, и др. Visual C# 2008: базовый курс. Visual Studio® 2008 = Beginning Visual C# 2008. — М.: «Диалектика», 2009. — С. 1216.

29. Бежанова, М.М. Практическое программирование: структуры данных и алгоритмы: учеб. для вузов / М.М. Бежанова, Л.А. Москвина, И.В. Поттосин. - М.: Логос, 2001. – 223 с.

30. Голицина О.Л., Попов И.И. Основы алгоритмизации и программирования: Учебное пособие. - М.: ФОРУМ: ИНФРА-М, 2005

31. Майо Д. Самоучитель Microsoft Visual Studio 2010 = Microsoft Visual Studio 2010: A Beginner's Guide (A Beginners Guide). — С.: «БХВ-Петербург», 2010. — С. 464.

32. Andrew Moore Visual Studio® 2010 All–in–One For Dummies®; Вимбо, Союз - Москва, 2010. - 912 с.

33. Andrew Parsons, Nick Randolph Professional Visual Studio 2005; Wrox - Москва, 2012. - 912 с.

34. Brain Johnson Working with Microsoft Visual Studio 2005; Н. Фену и Ко - Москва, 2010. - 304 с.

35. Istvan Novak Beginning Visual Studio 2010 LightSwitch Development; М.: Медгиз; Издание 5-е, перераб. и доп. - Москва, 2011. - 360 с.

36. James Avery Visual Studio Hacks; China Books & Periodicals - Москва, 2011. - 500 с.

37. Joydip Kanjilal Visual Studio 2010 and .NET 4 Six–in–One; Literatura Mondadori - Москва, 2010. - 816 с.

38. Keyvan Nayyeri Professional Visual Studio Extensibility; Wrox - Москва, 2013. - 552 с.

39. Lars Powers, Mike Snell Microsoft Visual Studio 2005 Unleashed; Sams - , 2012. - 888 с.

40. Sara Ford Microsoft Visual Studio Tips; М.: Всероссийское театральное общество - Москва, 2011. - 272 с.

41. Гарнаев А. Самоучитель Visual Studio .NET 2003; БХВ-Петербург - Москва, 2013. - 688 с.

42. Голощапов Алексей Microsoft Visual Studio 2010; БХВ-Петербург - Москва, 2011. - 544 с.

43. Левинсон Джефф Тестирование ПО с помощью Visual Studio 2010; ЭКОМ Паблишерз - Москва, 2012. - 314 с.

44. Понамарев Вячеслав Программирование на C++/C# в Visual Studio .NET 2003; БХВ-Петербург - Москва, 2013. - 352 с.

45. Рендольф Ник , Гарднер Дэвид , Минутилло Майкл , Андерсон Крис Visual Studio 2010 для профессионалов; Диалектика - Москва, 2011. - 692 с.

ПРИЛОЖЕНИЕ

Листинг формы приложения

```
namespace Gauss {
    partial class MainForm {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing) {
            if (disposing && (components != null)) {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
       # region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
```

```
private void InitializeComponent() {
    this.menuStrip1 = new System.Windows.Forms.MenuStrip();
    this.выходToolStripMenuItem = new
System.Windows.Forms.ToolStripMenuItem();
    this.statusStrip1 = new System.Windows.Forms.StatusStrip();
    this.tableLayoutPanel1 = new System.Windows.Forms.TableLayoutPanel();
    this.layoutMatrixA = new System.Windows.Forms.TableLayoutPanel();
    this.layoutVectorX = new System.Windows.Forms.TableLayoutPanel();
    this.layoutVectorB = new System.Windows.Forms.TableLayoutPanel();
    this.flowLayoutPanel1 = new System.Windows.Forms.FlowLayoutPanel();
    this.label1 = new System.Windows.Forms.Label();
    this.numericUpDown1 = new System.Windows.Forms.NumericUpDown();
    this.button1 = new System.Windows.Forms.Button();
    this.button2 = new System.Windows.Forms.Button();
    this.button3 = new System.Windows.Forms.Button();
    this.label2 = new System.Windows.Forms.Label();
    this.label3 = new System.Windows.Forms.Label();
    this.label4 = new System.Windows.Forms.Label();
    this.label5 = new System.Windows.Forms.Label();
    this.menuStrip1.SuspendLayout();
    this.tableLayoutPanel1.SuspendLayout();
    this.flowLayoutPanel1.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.numericUpDown1)).BeginInit()
;
```

```
this.SuspendLayout();
//
// menuStrip1
//
this.menuStrip1.Items.AddRange(new
System.Windows.Forms.ToolStripItem[] {
this.menuStrip1.Location = new System.Drawing.Point(0, 0);
this.menuStrip1.Name = "menuStrip1";
this.menuStrip1.Size = new System.Drawing.Size(569, 24);
this.menuStrip1.TabIndex = 0;
this.menuStrip1.Text = "menuStrip1";
//
// выходToolStripMenuItem
//
this.выходToolStripMenuItem.Name = "выходToolStripMenuItem";
this.выходToolStripMenuItem.Size = new System.Drawing.Size(149, 22);
this.выходToolStripMenuItem.Text = "Выход";
this.выходToolStripMenuItem.Click += new
System.EventHandler(this.выходToolStripMenuItem_Click);
//
// statusStrip1
//
this.statusStrip1.Location = new System.Drawing.Point(0, 253);
this.statusStrip1.Name = "statusStrip1";
this.statusStrip1.Size = new System.Drawing.Size(569, 22);
```

```
this.statusStrip1.TabIndex = 1;
this.statusStrip1.Text = "statusStrip1";
//
// tableLayoutPanel1
//
this.tableLayoutPanel1.ColumnCount = 4;
this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
67.30769F));
this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
9.615385F));
this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
9.615385F));
this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
13.46154F));
this.tableLayoutPanel1.Controls.Add(this.layoutMatrixA, 0, 2);
this.tableLayoutPanel1.Controls.Add(this.layoutVectorX, 2, 2);
this.tableLayoutPanel1.Controls.Add(this.layoutVectorB, 1, 2);
this.tableLayoutPanel1.Controls.Add(this.flowLayoutPanel1, 0, 0);
this.tableLayoutPanel1.Controls.Add(this.label2, 0, 1);
this.tableLayoutPanel1.Controls.Add(this.label3, 1, 1);
this.tableLayoutPanel1.Controls.Add(this.label4, 2, 1);
this.tableLayoutPanel1.Controls.Add(this.label5, 3, 1);
this.tableLayoutPanel1.Dock = System.Windows.Forms.DockStyle.Fill;
```

```
this.tableLayoutPanel1.Location = new System.Drawing.Point(0, 24);
this.tableLayoutPanel1.Name = "tableLayoutPanel1";
this.tableLayoutPanel1.RowCount = 3;
this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle());
this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle());
this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
100F));
this.tableLayoutPanel1.Size = new System.Drawing.Size(569, 229);
this.tableLayoutPanel1.TabIndex = 2;
//
// layoutMatrixA
//
this.layoutMatrixA.ColumnCount = 1;
this.layoutMatrixA.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
50F));
this.layoutMatrixA.Dock = System.Windows.Forms.DockStyle.Fill;
this.layoutMatrixA.Location = new System.Drawing.Point(3, 51);
this.layoutMatrixA.Name = "layoutMatrixA";
this.layoutMatrixA.RowCount = 1;
this.layoutMatrixA.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
50F));
this.layoutMatrixA.Size = new System.Drawing.Size(376, 175);
```

```
this.layoutMatrixA.TabIndex = 0;
//
// layoutVectorX
//
this.layoutVectorX.CausesValidation = false;
this.layoutVectorX.ColumnCount = 1;
this.layoutVectorX.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
50F));
this.layoutVectorX.Dock = System.Windows.Forms.DockStyle.Fill;
this.layoutVectorX.Location = new System.Drawing.Point(439, 51);
this.layoutVectorX.Name = "layoutVectorX";
this.layoutVectorX.RowCount = 1;
this.layoutVectorX.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
50F));
this.layoutVectorX.Size = new System.Drawing.Size(48, 175);
this.layoutVectorX.TabIndex = 1;
//
// layoutVectorB
//
this.layoutVectorB.ColumnCount = 1;
this.layoutVectorB.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent,
50F));
this.layoutVectorB.Dock = System.Windows.Forms.DockStyle.Fill;
```

```
this.layoutVectorB.Location = new System.Drawing.Point(385, 51);
this.layoutVectorB.Name = "layoutVectorB";
this.layoutVectorB.RowCount = 1;
this.layoutVectorB.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 50F
));
this.layoutVectorB.Size = new System.Drawing.Size(48, 175);
this.layoutVectorB.TabIndex = 2;
//
// flowLayoutPanel1
//
this.flowLayoutPanel1.AutoSize = true;
this.tableLayoutPanel1.SetColumnSpan(this.flowLayoutPanel1, 4);
this.flowLayoutPanel1.Controls.Add(this.label1);
this.flowLayoutPanel1.Controls.Add(this.numericUpDown1);
this.flowLayoutPanel1.Controls.Add(this.button1);
this.flowLayoutPanel1.Controls.Add(this.button2);
this.flowLayoutPanel1.Controls.Add(this.button3);
this.flowLayoutPanel1.Dock = System.Windows.Forms.DockStyle.Fill;
this.flowLayoutPanel1.Location = new System.Drawing.Point(3, 3);
this.flowLayoutPanel1.Name = "flowLayoutPanel1";
this.flowLayoutPanel1.Size = new System.Drawing.Size(563, 29);
this.flowLayoutPanel1.TabIndex = 4;
//
// label1
```

```
//
this.label1.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(3, 8);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(75, 13);
this.label1.TabIndex = 0;
this.label1.Text = "Размерность";
//
// numericUpDown1
//
this.numericUpDown1.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.numericUpDown1.Location = new System.Drawing.Point(84, 4);
this.numericUpDown1.Maximum = new decimal(new int[] {
15,
0,
0,
0});
this.numericUpDown1.Minimum = new decimal(new int[] {
1,
0,
0,
0});
this.numericUpDown1.Name = "numericUpDown1";
```

```
this.numericUpDown1.Size = new System.Drawing.Size(87, 20);
this.numericUpDown1.TabIndex = 1;
this.numericUpDown1.Value = new decimal(new int[] {
3,
0,
0,
0});
this.numericUpDown1.ValueChanged += new
System.EventHandler(this.numericUpDown1_ValueChanged);
//
// button2
//
this.button2.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.button2.Location = new System.Drawing.Point(177, 3);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(116, 23);
this.button2.TabIndex = 3;
this.button2.Text = "<-Предыд.итерация";
this.button2.UseVisualStyleBackColor = true;
//
// button3
//
this.button3.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.button3.Location = new System.Drawing.Point(299, 3);
this.button3.Name = "button3";
```

```
this.button3.Size = new System.Drawing.Size(116, 23);
this.button3.TabIndex = 4;
this.button3.Text = "След.итерация ->";
this.button3.UseVisualStyleBackColor = true;
//
// button1
//
this.button1.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.button1.Location = new System.Drawing.Point(421, 3);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(75, 23);
this.button1.TabIndex = 2;
this.button1.Text = "Решить";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// label2
//
this.label2.Anchor = System.Windows.Forms.AnchorStyles.Bottom;
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(184, 35);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(14, 13);
this.label2.TabIndex = 5;
```

```
this.label2.Text = "A";
//
// label3
//
this.label3.Anchor = System.Windows.Forms.AnchorStyles.Bottom;
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(402, 35);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(14, 13);
this.label3.TabIndex = 6;
this.label3.Text = "B";
this.label3.Click += new System.EventHandler(this.label3_Click);
//
// label4
//
this.label4.Anchor = System.Windows.Forms.AnchorStyles.Bottom;
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(456, 35);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(14, 13);
this.label4.TabIndex = 7;
this.label4.Text = "X";
//
// label5
```

```
//
this.label5.Anchor = System.Windows.Forms.AnchorStyles.Bottom;
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(522, 35);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(15, 13);
this.label5.TabIndex = 8;
this.label5.Text = "U";
//
// MainForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(569, 275);
this.Controls.Add(this.tableLayoutPanel1);
this.Controls.Add(this.statusStrip1);
this.Controls.Add(this.menuStrip1);
this.MainMenuStrip = this.menuStrip1;
this.Name = "MainForm";
this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Решение СЛАУ методом Гаусса";
this.Load += new System.EventHandler(this.MainForm_Load);
this.menuStrip1.ResumeLayout(false);
this.menuStrip1.PerformLayout();
```

```
            this.tableLayoutPanel1.ResumeLayout(false);
            this.tableLayoutPanel1.PerformLayout();
            this.flowLayoutPanel1.ResumeLayout(false);
            this.flowLayoutPanel1.PerformLayout();
((System.ComponentModel.ISupportInitialize)(this.numericUpDown1)).EndInit();
            this.ResumeLayout(false);
            this.PerformLayout();
        }
        #endregion
        private System.Windows.Forms.MenuStrip menuStrip1;
        private System.Windows.Forms.StatusStrip statusStrip1;
        private System.Windows.Forms.TableLayoutPanel tableLayoutPanel1;
        private System.Windows.Forms.TableLayoutPanel layoutMatrixA;
        private System.Windows.Forms.TableLayoutPanel layoutVectorX;
        private System.Windows.Forms.TableLayoutPanel layoutVectorB;
        private System.Windows.Forms.FlowLayoutPanel flowLayoutPanel1;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.NumericUpDown numericUpDown1;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label label4;

        private System.Windows.Forms.ToolStripMenuItem
выходToolStripMenuItem;
```

```
        private System.Windows.Forms.Button button2;

        private System.Windows.Forms.Button button3;

        private System.Windows.Forms.Label label5;

    }

}
```

Листинг программы, реализующей модифицированный метод Гаусса

```
using System;
using System.Collections;
using System.Data;

namespace Gauss {
    public class GaussSolutionNotFound : Exception {
        public GaussSolutionNotFound(string msg)
            : base("Решение не может быть найдено: \r\n" +
msg) {
        }
    }

    public class LinearSystem {
        private double[,] initial_a_matrix;
        private double[,] a_matrix;  // матрица A
        private double[] x_vector;   // вектор неизвестных x
        private double[] initial_b_vector;
        private double[] b_vector;   // вектор b

        public LinearSystem(double[,] a_matrix, double[]
b_vector)
            : this(a_matrix, b_vector, 0.1) {
        }
        public LinearSystem(double[,] a_matrix, double[]
b_vector) {
            if (a_matrix == null || b_vector == null)
                throw new ArgumentNullException("Один из
параметров равен null.");

            int b_length = b_vector.Length;
            int a_length = a_matrix.Length;
            if (a_length != b_length * b_length)
```

```
                throw new ArgumentException(@"Количество
строк и столбцов в матрице A должно совпадать с количеством
элементров в векторе B.");

            this.initial_a_matrix = a_matrix;  // запоминаем
исходную матрицу
            this.a_matrix = (double[,])a_matrix.Clone(); //
с её копией будем производить вычисления
            this.initial_b_vector = b_vector;  // запоминаем
исходный вектор
            this.b_vector = (double[])b_vector.Clone();  //
с его копией будем производить вычисления
            this.x_vector = new double[b_length];
            this.size = b_length;

            GaussSolve();
        }

        public double[] XVector {
            get {
                return x_vector;
            }
        }


        // инициализация массива индексов столбцов
        private int[] InitIndex() {
            int[] index = new int[size];
            for (int i = 0; i < index.Length; ++i)
                index[i] = i;
            return index;
        }

        // поиск главного элемента в матрице
        private double FindR(int row, int[] index) {
            int max_index = row;
            double max = a_matrix[row, index[max_index]];
            double max_abs = Math.Abs(max);
            //if(row < size - 1)
            for (int cur_index = row + 1; cur_index < size;
++cur_index) {
                double cur = a_matrix[row,
index[cur_index]];
                double cur_abs = Math.Abs(cur);
```

```
                if (cur_abs > max_abs) {
                    max_index = cur_index;
                    max = cur;
                    max_abs = cur_abs;
                }
            }

            if (max_abs < eps) {
                if (Math.Abs(b_vector[row]) > eps)
                    throw new GaussSolutionNotFound("Система
уравнений несовместна.");
                else
                    throw new GaussSolutionNotFound("Система
уравнений имеет множество решений.");
            }

            // меняем местами индексы столбцов
            int temp = index[row];
            index[row] = index[max_index];
            index[max_index] = temp;

            return max;
        }

        // Нахождение решения СЛАУ методом Гаусса
        private void GaussSolve() {
            int[] index = InitIndex();
            GaussForwardStroke(index);
        }

        private void GaussForwardStroke(int[] index) {
            // перемещаемся по диагонали
            for (int i = 0; i < size; ++i) {
                // 1) выбор главного элемента
                double r = FindR(i, index);

                // 2) преобразование матрицы А
                for (int j = 0; j < size; ++j)
                    a_matrix[k, l] =
a_matrix[k,l]*a_matrix[I,j]-a_matrix[k,j]*a_matrix[i,l];


                for (int k = i + 1; k < size; ++k) {
                    for (int j = i; j < size; ++j)
```

```
                    a_matrix[k, index[i]] = 0.0;
                }
            }
        }
                         // 3) поиск решения
                for (int j = i + 1; j < size; ++j)
                    x_i = b_matrix[k,l]/a_matrix[k,l];
                x_vector[index[i]] = x_i;
            }
        }
```

Магистерская диссертация выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы имеют ссылки на них.

01.06.2017 г.

_______________ Сухомлинов А.Д.

(подпись)