Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Методы численного анализа

ОТЧЁТ

к лабораторной работе
на тему

Решение систем линейных алгебраических уравнений (СЛАУ)
методом Гаусса и с помощью его модификаций

Выполнил: студент группы 153501
Тимофеев Кирилл Андреевич

Проверил: Анисимов Владимир Яковлевич

Минск 2022

Вариант 8

Цель выполнения задания:

- изучить метод Гаусса и его модификации, составить алгоритм метода и программу его реализации, получить численное решение заданной СЛАУ;
- составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ;
- составить программу решения СЛАУ по разработанному алгоритму;
- выполнить тестовые примеры и проверить правильность работы программы.

Краткие теоретические сведения:

СЛАУ обычно записывается в виде

$$\sum_{j=1}^{n} a_{ij} x_j = b_i; i \le 1 \le n \quad \text{, или коротко} \quad \boldsymbol{Ax} = \boldsymbol{b}, \qquad (1.1)$$

где

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}; \quad \boldsymbol{a} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}; \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}.$$

Здесь $\boldsymbol{A}$ и $\boldsymbol{b}$ заданы, требуется найти $\boldsymbol{x}$.

Методы решения СЛАУ делятся на прямые и итерационные.

Метод Гаусса прекрасно подходит для решения систем линейных алгебраических уравнений (СЛАУ). Он обладает рядом преимуществ по сравнению с другими методами:

- во-первых, нет необходимости предварительно исследовать систему уравнений на совместность;
- во-вторых, методом Гаусса можно решать не только СЛАУ, в которых число уравнений совпадает с количеством неизвестных переменных и основная матрица системы невырожденная, но и системы уравнений, в которых число уравнений не совпадает с количеством неизвестных переменных или определитель основной матрицы равен нулю;
- в-третьих, метод Гаусса приводит к результату при сравнительно небольшом количестве вычислительных операций.

Процесс последовательного исключения неизвестных называется прямым ходом метода Гаусса. После завершения прямого хода у нас появляется возможность вычислить неизвестную переменную, находящуюся в последнем уравнении. С ее помощью из предпоследнего уравнения находим следующую неизвестную переменную и так далее. Процесс последовательного нахождения неизвестных переменных при движении от последнего уравнения к первому называется обратным ходом метода Гаусса.

Рассмотрим сначала простейший вариант метода Гаусса, называемый *схемой единственного деления*.

Прямой ход состоит из $n-1$ шагов исключения.

1-й шаг. Целью этого шага является исключение неизвестного x_1 из уравнений с номерами $i = 2, 3, \ldots, n$. Предположим, что коэффициент $a_{11} \neq 0$. Будем называть его *главным элементом 1-го шага*.

Найдем величины

$$q_{i1} = a_{i1}/a_{11} \quad (i = 2, 3, \ldots, n),$$

называемые *множителями 1-го шага*. Вычтем последовательно из второго, третьего, …, n-го уравнений системы первое уравнение, умноженное соответственно на q_{21}, q_{31}, …, q_{n1}. Это позволит обратить в нуль коэффициенты при x_1 во всех уравнениях, кроме первого. В результате получим эквивалентную систему

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1,$$

$$a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n = b_2^{(1)},$$

$$a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + \ldots + a_{3n}^{(1)}x_n = b_3^{(1)},$$

$$. \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad .$$

$$a_{n2}^{(1)}x_2 + a_{n3}^{(1)}x_3 + \ldots + a_{nn}^{(1)}x_n = b_n^{(1)},$$

в которой $a_{ij}^{(1)}$ и $b_{ij}^{(1)}$ вычисляются по формулам

$$a_{ij}^{(1)} = a_{ij} - q_{i1}a_{1j} \quad , \quad b_i^{(1)} = b_i - q_{i1}b_1.$$

Аналогично проводятся остальные шаги. Опишем очередной k-й шаг.

<u>k-й шаг</u>. В предположении, что главный(ведущий) элемент k-ого шага отличен от нуля, вычислим множители k-ого шага

$$q_{ik} = a_{ik}^{(k-1)} / a_{kk}^{(k-1)} \quad (i = k+1, \ldots, n)$$

и вычтем последовательно из (k+1)-ого, …, n-го уравнений полученной на предыдущем шаге системы k-е уравнение, умноженное соответственно на $q_{k+1,k}, q_{k+2,k}, \ldots, q_{nk}$.

После (n-1)-го шага исключения получим систему уравнений

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \quad , \\
a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)} \quad , \\
a_{33}^{(2)}x_3 + \ldots + a_{3n}^{(2)}x_n &= b_3^{(2)} \quad , \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\
a_{nn}^{(n-1)}x_n &= b_n^{(n-1)} \; ,
\end{aligned}$$

матрица которой является верхней треугольной. На этом вычисления прямого хода заканчиваются.

Обратный ход. Из последнего уравнения системы находим x_n. Подставляя найденное значение x_n в предпоследнее уравнение, получим x_{n-1}. Осуществляя обратную подстановку, далее последовательно находим x_{n-2}, …, x_1. Вычисления неизвестных здесь проводятся по формулам

$$x_n = b_n^{(n-1)} / a_{nn}^{(n-1)},$$

$$x_k = (b_n^{(k-1)} - a_{k,k+1}^{(k-1)} x_{k+1} - \ldots - a_{kn}^{(k-1)} x_n) / a_{kk}^{(k-1)}, \ (k = n-1, \ldots, 1).$$

Необходимость отличия от 0 главных элементов. Заметим, что вычисления множителей, а также обратная подстановка требуют деления на главный элемент $a_{kk}^{(k-1)}$. Поэтому если один из главных элементов оказывается равным нулю, то схема не может быть реализована. Здравый смысл подсказывает, что в ситуации, когда все главные элементы отличны от нуля, но среди них есть близкие к нулю, возможен неконтролируемый рост погрешности.

Метод Гаусса с выбором главного элемента по столбцу(схема частичного выбора). На k-м шаге прямого хода коэффициенты уравнений системы с номерами $i = k + 1, \ldots, n$ преобразуются по формулам

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - q_{ik} a_{kj}, \ b_i^{(k)} = b_i^{(k-1)} - q_{ik} b_k^{(k-1)}, \ i = k+1, \ldots, n.$$

Интуитивно ясно, что во избежание сильного роста коэффициентов системы и связанных с этим ошибок нельзя допускать появления больших множителей q_{ik}. В методе Гаусса с выбором главного элемента по столбцу гарантируется, что $|q_{ik}| <= 1$ для всех $k = 1, 2, \ldots, n - 1$ и $i = k + 1, \ldots, n$. Отличие этого варианта метода Гаусса от схемы единственного деления заключается в том, что на k-м шаге исключения в качестве главного элемента выбирают максимальный по модулю коэффициент $a_{i_k k}$ при неизвестных x_k в уравнениях с номерами $i = k + 1, \ldots, n$. Затем соответствующее выбранному коэффициенту уравнение с номером i_k

меняют местами с k-м уравнением системы для того, чтобы главный элемент занял место коэффициента $a_{kk}^{(k-1)}$. После этой перестановки исключение неизвестного x_k производят, как в схеме единого деления.

Метод Гаусса с выбором главного элемента по всей матрице(схема полного выбора). В этой схеме допускается нарушение естественного порядка исключения неизвестных. На 1-м шаге метода среди элементов a_{ij} определяют максимальный по модулю элемент $a_{i_1 j_1}$. Первое уравнение системы и уравнение с номером i_1 меняют местами. Далее стандартным образом производят исключение неизвестного x_{i_1} из всех уравнений, кроме первого.

На k-ом шаге метода среди коэффициентов $a_{ij}^{(k-1)}$ при неизвестных в уравнениях системы с номерами i = k, ..., n выбирают максимальный по модулю коэффициент $a_{i_k j_k}^{(k-1)}$ при неизвестных в уравнениях системы с номерами i = k, ..., n выбирают максимальный по модулю коэффициент, меняют местами и исключают неизвестное x_{j_k} из уравнений с номерами i = k + 1, ..., n.

На этапе обратного хода неизвестные вычисляют в следующем порядке: $x_{j_n}, x_{j_{n-1}}, \ldots, x_{j_1}$.

ЗАДАНИЕ.

Методом Гаусса и методом выбора главного элемента найти с точностью 0,0001 численное решение системы $Ax=b$,

где $A = kC + D$, A – исходная матрица для расчёта, k – номер варианта (0-15), матрицы C, D и вектор свободных членов b задаются ниже.

Исходные данные:

Вектор $\mathbf{b} = (4{,}2;\ 4{,}2;\ 4{,}2\ ;\ 4{,}2;\ 4{,}2)^T$,

$$C = \begin{bmatrix} 0{,}2 & 0 & 0{,}2 & 0 & 0 \\ 0 & 0{,}2 & 0 & 0{,}2 & 0 \\ 0{,}2 & 0 & 0{,}2 & 0 & 0{,}2 \\ 0 & 0{,}2 & 0 & 0{,}2 & 0 \\ 0 & 0 & 0{,}2 & 0 & 0{,}2 \end{bmatrix}, \quad D = \begin{bmatrix} 2{,}33 & 0{,}81 & 0{,}67 & 0{,}92 & -0{,}53 \\ -0{,}53 & 2{,}33 & 0{,}81 & 0{,}67 & 0{,}92 \\ 0{,}92 & -0{,}53 & 2{,}33 & 0{,}81 & 0{,}67 \\ 0{,}67 & 0{,}92 & -0{,}53 & 2{,}33 & 0{,}81 \\ 0{,}81 & 0{,}67 & 0{,}92 & -0{,}53 & 2{,}33 \end{bmatrix}.$$

Програмная реализация:

```
using System;
using System.IO;


namespace MNA_

{

    class Program

    {

        String projectLocation = "D:\\Mocha\\MNA!\\";

        decimal k = 8;

        int rows = 5;

        int columns = 5;

        Matrix A, b, C, D, tmp;
```

```
public Program()
{
    double[] solution = new double[columns];
    A = new Matrix(rows, columns + 1);
    b = new Matrix(rows, 1);
    C = new Matrix(rows, columns);
    D = new Matrix(rows, columns);
}

string[] BPath = new string[2] { "BDefault.txt", "B.txt" };
string[] CPath = new string[2] { "CDefault.txt", "C.txt" };
string[] DPath = new string[2] { "DDefault.txt", "D.txt" };

void ReadData(int mode)
{
    ReadB(mode);
    ReadC(mode);
    ReadD(mode);
}

void ReadB(int mode)
{
    StreamReader sr = new StreamReader(projectLocation + BPath[mode]);
    string line = sr.ReadLine();
    for (int i = 0; i < rows; ++i)
    {
        int f = line.IndexOf(' ');
        //Console.WriteLine("f = " + f);
        if (f != -1) b[i,0] = Convert.ToDecimal(line.Substring(0, f));
        else b[i,0] = Convert.ToDecimal(line);
        line = line.Substring(f + 1);
    }
}
```

```
void ReadC(int mode)
{
    StreamReader sr = new StreamReader(projectLocation + CPath[mode]);
    for (int i = 0; i < rows; ++i)
    {
        string line = sr.ReadLine();

        for (int j = 0; j < columns; ++j)
        {
            int f = line.IndexOf(' ');
            //Console.WriteLine("f = " + f);
            if (f != -1) C[i, j] = Convert.ToDecimal(line.Substring(0, f));
            else C[i, j] = Convert.ToDecimal(line);
            line = line.Substring(f + 1);
        }
    }
}

void ReadD(int mode)
{
    StreamReader sr = new StreamReader(projectLocation + DPath[mode]);
    for (int i = 0; i < rows; ++i)
    {
        string line = sr.ReadLine();

        for (int j = 0; j < columns; ++j)
        {
            int f = line.IndexOf(' ');
            //Console.WriteLine("f = " + f);
            if (f != -1) D[i, j] = Convert.ToDecimal(line.Substring(0, f));
            else D[i, j] = Convert.ToDecimal(line);
            line = line.Substring(f + 1);
        }
    }
```

```
}

public void run()
{
    Console.WriteLine("1. Решение тестового уравнения");
    Console.WriteLine("2. Решение своего уравнения");
    int a = Convert.ToInt32(Console.ReadLine());
    ReadData(a - 1);

    tmp = k*C;
    tmp += D;

    for(int i = 0; i < tmp.Rows; ++i)
    {
        for(int j = 0; j < tmp.Columns; ++j)
        {
            A[i, j] = tmp[i, j];
        }
    }

    for(int i = 0; i < A.Rows; ++i)
    {
        A[i, columns] = b[i, 0];
    }



    Matrix A2 = new Matrix(A);
    Matrix A3 = new Matrix(A);

    //Console.WriteLine(A.ToString());
    A.ToUpTriangle();
    A2.ToUpTriangleModified();
    A3.ToUpTriangle3RetutrnOfTheAElenaDmitrievna();
    //Console.WriteLine(A.ToString());
```

```
    var x1 = backMove(A);

    printResults(A, x1);
    checkResults(x1);

    Console.WriteLine("\n\n\nВторой метод(с выбором по столбцу)\n");
    x1 = backMove(A2);

    printResults(A2, x1);
    checkResults(x1);

    Console.WriteLine("\n\n\nТретий метод(с выбором по матрице)\n");
    x1 = decrypt(A3);

    printResults(A3, x1);
    checkResults(x1);

}

void printResults(Matrix slau, Matrix x)
{
    printx(x);

    Console.WriteLine(" \nРезультаты подстановки");
    for (int i = 0; i < rows; ++i)
    {
        decimal sum = 0;
        for (int j = 0; j < columns; ++j)
        {
            sum += slau[i, j] * x[0, j];
        }
```

```
                Console.WriteLine(i + ".  " + Math.Round(sum, 4) + " = " + Math.Round(slau[i,
columns], 4));
            }
        }

        void checkResults(Matrix x)
        {
            Console.WriteLine(" \nРезультаты подстановки в исходное уравнение");
            for (int i = 0; i < rows; ++i)
            {
                decimal sum = 0;
                for (int j = 0; j < columns; ++j)
                {
                    sum += tmp[i, j] * x[0, j];
                }

                Console.WriteLine(i + ".  " + sum + " = " + b[i,0] + " ==>> Невязка = " + (b[i,0] -
sum) );
            }
        }

        Matrix decrypt(Matrix slau)
        {
            Matrix _slau = new Matrix(slau);
            var x = new Matrix(1, _slau.Columns - 1);
            int[] _checked = new int[0];
            for (int g = 1; g <= _slau.Rows; ++g)
            {
                int i = 0, j = 0;
                for (i = 0; i < _slau.Rows; ++i)
                {
                    int counter = 0;
                    for (j = 0; j < _slau.Columns - 1; ++j)
                    {
                        if (Math.Round(_slau[i, j], 4) != 0)
```

```
                {
                    ++counter;
                }
            }
            if (counter == g) break;
        }

        decimal sum = _slau[i, slau.Columns - 1];
        for(int k = 0; k < _checked.Length; ++k)
        {
            sum -= x[0, _checked[k]] * slau[i, _checked[k]];
            _slau[i, _checked[k]] = 0;
        }

        int c = 0;
        for (; c < _slau.Columns - 1; ++c) if (Math.Round(_slau[i, c], 4) != 0) break;
        x[0, c] = sum;
        Array.Resize(ref _checked, _checked.Length + 1);
        _checked[_checked.Length - 1] = c;

    }
    return x;
}

Matrix backMove(Matrix slau)
{
    var x = new Matrix(1, slau.Columns - 1);

    for (int i = slau.Rows - 1; i >= 0; --i)
    {
        int k = i + 1;
        decimal d = slau[i, columns];
        for (; k < x.Columns; ++k)
        {
```

```
                d -= slau[i, k] * x[0, k];
            }

            x[0, i] = d;
        }

        return x;
    }

    void printx(Matrix x)
    {
        for (int i = 0; i < columns; ++i)
        {
            Console.WriteLine(" x" + i + " = " + Math.Round(x[0, i], 4));
        }
    }
}

class MainClass
{
    static void Main()
    {
        Program program = new Program();
        program.run();
    }
}
}
```

Результаты расчетов программы:

```
1. Решение тестового уравнения
2. Решение своего уравнения
1
3,93 0,81 2,27 -0,53 -0,53 4,2
-0,53 3,93 0,81 2,27 0,92 4,2
2,52 -0,53 3,93 0,81 2,27 4,2
0,67 2,52 -0,53 3,93 0,81 4,2
0,81 0,67 2,52 -0,53 3,93 4,2

 x0 = 0,9181
 x1 = 0,8624
 x2 = 0,1645
 x3 = 0,2454
 x4 = 0,6601

Результаты подстановки
0.  1,0687 = 1,0687
1.  1,1800 = 1,1800
2.  0,9930 = 0,9930
3.  0,6040 = 0,6040
4.  0,6601 = 0,6601

Результаты подстановки в исходное уравнение
0.  4,1999999999999999999999999987 = 4,2 ==>> Невязка = 0,0000000000000000000000000013
1.  4,1999999999999999999999999992 = 4,2 ==>> Невязка = 0,0000000000000000000000000008
2.  4,1999999999999999999999999998 = 4,2 ==>> Невязка = 0,0000000000000000000000000002
3.  4,1999999999999999999999999996 = 4,2 ==>> Невязка = 0,0000000000000000000000000004
4.  4,1999999999999999999999999997 = 4,2 ==>> Невязка = 0,0000000000000000000000000003
```

```
Второй метод(с выбором по столбцу)

 x0 = 0,9181
 x1 = 0,8624
 x2 = 0,1645
 x3 = 0,2454
 x4 = 0,6601

Результаты подстановки
0.  1,0687 = 1,0687
1.  1,1800 = 1,1800
2.  0,9930 = 0,9930
3.  0,6040 = 0,6040
4.  0,6601 = 0,6601

Результаты подстановки в исходное уравнение
0.  4,1999999999999999999999999987 = 4,2 ==>> Невязка = 0,0000000000000000000000000013
1.  4,1999999999999999999999999992 = 4,2 ==>> Невязка = 0,0000000000000000000000000008
2.  4,1999999999999999999999999998 = 4,2 ==>> Невязка = 0,0000000000000000000000000002
3.  4,1999999999999999999999999996 = 4,2 ==>> Невязка = 0,0000000000000000000000000004
4.  4,1999999999999999999999999997 = 4,2 ==>> Невязка = 0,0000000000000000000000000003
```

```
Третий метод(с выбором по матрице)

 x0 = 0,9181
 x1 = 0,8624
 x2 = 0,1645
 x3 = 0,2454
 x4 = 0,6601

Результаты подстановки
0.  0,1645 = 0,1645
1.  0,1412 = 0,1412
2.  0,6968 = 0,6968
3.  1,0687 = 1,0687
4.  1,1800 = 1,1800

Результаты подстановки в исходное уравнение
0.  4,1999999999999999999999999989 = 4,2 ==>> Невязка = 0,0000000000000000000000000011
1.  4,1999999999999999999999999993 = 4,2 ==>> Невязка = 0,0000000000000000000000000007
2.  4,1999999999999999999999999995 = 4,2 ==>> Невязка = 0,0000000000000000000000000005
3.  4,1999999999999999999999999996 = 4,2 ==>> Невязка = 0,0000000000000000000000000004
4.  4,1999999999999999999999999986 = 4,2 ==>> Невязка = 0,0000000000000000000000000014
```

Оценка:

Подсчет абсолютной погрешности приближенного решения:

Учитывая

$$\|\mathbf{r}\| \leqslant \|A^{-1}\| \, \|\eta\|.$$

Имеем для метода Гаусса и метода Гаусса с выбором по столбцу

> $A := Matrix([[3.93, 0.81, 2.27, -0.53, -0.53],$
$[-0.53, 3.93, 0.81, 2.27, 0.92,],$
$[2.52, -0.53, 3.93, 0.81, 2.27],$
$[0.67, 2.52, -0.53, 3.93, 0.81],$
$[0.81, 0.67, 2.52, -0.53, 3.93]]);$

$$A := \begin{bmatrix} 3.93 & 0.81 & 2.27 & -0.53 & -0.53 \\ -0.53 & 3.93 & 0.81 & 2.27 & 0.92 \\ 2.52 & -0.53 & 3.93 & 0.81 & 2.27 \\ 0.67 & 2.52 & -0.53 & 3.93 & 0.81 \\ 0.81 & 0.67 & 2.52 & -0.53 & 3.93 \end{bmatrix}$$

$$> \ b := \begin{bmatrix} 0.0000000000000000000000000013 \\ 0.0000000000000000000000000008 \\ 0.0000000000000000000000000002 \\ 0.0000000000000000000000000004 \\ 0.0000000000000000000000000003 \end{bmatrix}$$

$$b := \begin{bmatrix} 1.3\ 10^{-27} \\ 8.\ 10^{-28} \\ 2.\ 10^{-28} \\ 4.\ 10^{-28} \\ 3.\ 10^{-28} \end{bmatrix}$$

> $r \leq norm(A1, 1) \cdot norm(b, 1);$

$$r \leq 3.0000000000000012\ 10^{-27}$$

Для метода Гаусса с выбором по матрице

$$> \ b := \begin{bmatrix} 0.0000000000000000000000000011 \\ 0.0000000000000000000000000007 \\ 0.0000000000000000000000000005 \\ 0.0000000000000000000000000004 \\ 0.00000000000000000000000000014 \end{bmatrix}$$

$$b := \begin{bmatrix} 1.1\ 10^{-27} \\ 7.\ 10^{-28} \\ 5.\ 10^{-28} \\ 4.\ 10^{-28} \\ 1.4\ 10^{-28} \end{bmatrix}$$

> $r \leq norm(A1, 1) \cdot norm(b, 1);$

$$r \leq 2.8399999999999992\ 10^{-27}$$

>

Относительная погрешность приближенного числа:

Учитывая формулу

$$\left|\frac{a-a^*}{a}\right| = \frac{\beta_{1+1} r^{-(1+1)} + \beta_{1+1} r^{-(1+2)} + \dots}{\beta_1 r^{-1} + \beta_2 r^{-2} + \dots} \le \frac{r^{-1}}{\beta_1 r^{-1}} \le r^{1-1}$$

где r-основание системы исчисления

И параметры используемого типа decimal

Ниже в таблице даны параметры стандартных форматов чисел с плавающей запятой. Здесь: **w** — ширина битового поля для представления порядка, **t** — ширина битового поля для представления мантиссы, **k** — полная ширина битовой строки.

Параметр	*Binary32*	*Binary64*	*Binary128*	*Decimal64*	*Decimal128*
Параметры формата					
b	2	2	2	10	10
p, цифры	24	53	113	16	34
emax	127	1023	16383	384	6144
Параметры кодирования					
BIAS	127	1023	16383	398	6176
w, биты	8	11	15	13	17
t, биты	23	52	112	50	110
k, биты	32	64	128	64	128

То относительная погрешность приближенного числа не превышает 2^(1 – 110) = 1,50463*10^(-36)

Вывод:

В ходе лабораторной работы был изучен метод Гаусса и его модификации(метод Гаусса с выбором главного элемента по столбцу (схема частичного выбора) и метод Гаусса с выбором главного элемента по всей матрице (схема полного выбора)) для решения системы линейных уравнений, составлены алгоритмы и написаны реализации соответствующих программ на языке C# для решения поставленной задачи. Проведена оценка погрешности. Благодаря точности типа decimal результаты более чем удовлетворительны.

СПИСОК ЛИТЕРАТУРЫ

1. Deverstudents / Метод Гаусса: описание алгоритма решения системы линейных уравнений, примеры, решения [Электронный ресурс]. - http://www.cleverstudents.ru/systems/solving_systems_Gauss_method.html
2. Лабораторный практикум(краткая теоретические сведения)
3. KURSAK / МЕТОД ГАУССА С ВЫБОРОМ ГЛАВНОГО ЭЛЕМЕНТА [Электронный ресурс]. – Режим доступа: http://kursak.net/metod-gaussa-s-vyborom-glavnogo-elementa/
4. ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ АЛГЕБРЫ: БАЗОВЫЕ ПОНЯТИЯ И АЛГОРИТМЫ / Б.В, Фалейчик, 2010