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 253505
Снежко Максим Андреевич

Проверил: Анисимов Владимир Яковлевич

Минск 2023

ЦЕЛИ

- Изучить метод Гаусса и его модификации, составить алгоритм метода и программу его реализации, получить численное решение заданной СЛАУ
- Написать 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

Введение

Для эффективного решения систем линейных алгебраических уравнений (СЛАУ) в практических задачах часто требуются быстрые методы. Один из классических методов для этой цели - метод Гаусса и его вариации. Основная идея метода Гаусса заключается в поэтапном исключении переменных из системы, чтобы преобразовать ее в эквивалентную систему с верхней треугольной матрицей. Модификации метода Гаусса, включая выбор главного элемента, используются для уменьшения значений коэффициентов матрицы СЛАУ, при этом незначительно уступая в скорости вычислений. Это позволяет снизить погрешности в результате вычислений.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Система линейных алгебраических уравнений (СЛАУ) - это набор уравнений, в которых все переменные входят в первой степени и коэффициенты при них являются константами. Общий вид СЛАУ можно записать следующим образом:

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

где

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

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

Методы решения систем линейных алгебраических уравнений (СЛАУ) можно разделить на две основные категории: прямые (или методы прямого подстановочного типа) и итерационные методы.

1 Прямые методы: Прямые методы применяются для точного решения СЛАУ и обеспечивают точное решение системы. Они основаны на последовательных математических операциях, таких как умножение, вычитание и деление, чтобы найти значения неизвестных.

2 Итерационные методы: Итерационные методы используются для приближенного решения СЛАУ. Они начинают с начального приближения и последовательно уточняют решение на каждой итерации.

Прямые методы обычно требуют более высокой вычислительной мощности и памяти, но они гарантируют точное решение. Наибольшее распространение среди прямых методов получили метод Гаусса и его модификации.

Метод Гаусса обладает рядом преимуществ по сравнению с другими методами:

1 Точное решение: Метод Гаусса гарантирует точное решение СЛАУ при условии, что система уравнений имеет решение и матрица коэффициентов невырожденная.

2 Универсальность: Метод Гаусса применим к разнообразным видам СЛАУ с различными коэффициентами и правыми частями. Он не зависит от

специфических характеристик системы и применим к системам с произвольным числом переменных и уравнений.

3 Простота реализации: Алгоритм метода Гаусса относительно прост в понимании и реализации. Он состоит из двух основных этапов: прямого и обратного хода, что делает его доступным для программирования и вычислений вручную.

4 Стабильность: В большинстве случаев метод Гаусса стабилен и надежен. Он не подвержен численным неустойчивостям, которые могут возникнуть при использовании некоторых итерационных методов.

Метод Гаусса, используемый для решения систем линейных алгебраических уравнений (СЛАУ), включает два основных этапа: прямой и обратный ходы.

Прямой ход метода Гаусса заключается в преобразовании расширенной матрицы системы так, чтобы получить верхнюю треугольную матрицу с нулями под главной диагональю. Это достигается путем последовательного исключения переменных из уравнений.

Обратный ход, который иногда называется методом Гаусса-Жордана, представляет собой этап, в котором матрица приводится к диагональному виду, и значения переменных вычисляются от последней к первой. Это также включает в себя последовательное исключение переменных, но на этот раз вверх от главной диагонали.

Таким образом, метод Гаусса и метод Гаусса-Жордана отличаются только последовательностью шагов исключения переменных. Оба метода являются прямыми методами решения СЛАУ, и оба обеспечивают точное решение системы, при условии, что система имеет единственное решение и матрица коэффициентов невырожденная.

Метод Гаусса идеально подходит для решения систем содержащих больше трех линейных уравнений, для решения систем уравнений, которые не являются квадратными (чего не скажешь про метод Крамера и матричный метод). То есть метод Гаусса - наиболее универсальный метод для нахождения решения любой системы линейных уравнений, он работает в случае, когда система имеет бесконечно много решений или несовместна.

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

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

Найдем величины

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

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

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

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

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

$$\dots\dots\dots\dots\dots\dots\dots$$

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

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

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

2-й шаг. Целью этого шага является исключение неизвестного x_2 из уравнений с номерами $i = 3, 4, \dots, n$. Пусть $a_{22}^{(1)} \neq 0$, где $a_{22}^{(1)}$ – коэффициент, называемый *главным* (или *ведущим*) *элементом* 2-*го шага*. Вычислим множители 2-го шага

$$q_{i2} = a_{i2}^{(1)} / a_{22}^{(1)} \quad (i = 3, 4, \dots, n)$$

и вычтем последовательно из третьего, четвертого, …, n-го уравнений системы второе уравнение, умноженное соответственно на $q_{32}, q_{42}, \dots, q_{n2}$. В результате получим систему

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1, \\
a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\
a_{33}^{(2)}x_3 + \ldots + a_{3n}^{(2)}x_n &= b_3^{(2)}, \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{n3}^{(2)}x_3 + \ldots + a_{nn}^{(2)}x_n &= b_n^{(2)}.
\end{aligned}
$$

Здесь коэффициенты $a_{ij}^{(2)}$ и $b_j^{(2)}$ вычисляются по формулам

$$a_{ij}^{(2)} = a_{ij}^{(1)} - q_{i2}a_{2j}^{(1)}, \quad b_i^{(2)} = b_i^{(1)} - q_{i2}b_2^{(1)}.$$

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

k-й шаг. В предположении, что *главный* (*ведущий*) *элемент* k-го шага $a_{kk}^{(k-1)}$ отличен от нуля, вычислим *множители k-го шага*

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

и вычтем последовательно из $(k + 1)$-го, …, 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

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

матрица $\boldsymbol{A}^{(n-1)}$ 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 вычисление множителей, а также обратная подстановка требуют деления на главные элементы $a_{kk}^{(k-1)}$. Поэтому если один из главных элементов оказывается равным нулю, то схема не может быть реализована. Здравый смысл подсказывает, что и 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 $|q_{ik}| \le 1$ 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 неизвестной 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 единственного 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-м шаге метода среди коэффициентов $a_{ij}^{(k-1)}$ при неизвестных в уравнениях системы с номерами $i = k, \ldots, n$ выбирают максимальный по модулю коэффициент $a_{i_k j_k}^{(k-1)}$. Затем k-е уравнение и уравнение, содержащее найденный коэффициент, меняют местами и исключают неизвестное x_{j_k} из уравнений с номерами $i = k + 1, \ldots, n$.

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

Вариант 27

Исходные данные:

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

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

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

```
#include <iostream>
#include <iomanip>

using namespace std;

const int n = 5;

double* gauss1(double A[n][n], double* b, int n)  // схема единстенного деления
{
	double* x = new double[n];
	for (int i = 0; i < n; i++)  // прямой ход, приведение к верхетреугольному виду
	{
		if (A[i][i] != 0)
		{
			for (int j = i + 1; j < n; j++)
			{
				double q = A[j][i] / A[i][i];  // получаем q
				for (int k = i; k < n; k++)  // вычитаем i-ую строку умноженную на q из последующих строк
				{
					if (k == i)
					{
						A[j][k] = 0;  // исключение погрешности для нулевых элементов
					}
					else
					{
						A[j][k] -= q * A[i][k];
					}
				}
				b[j] -= q * b[i];
			}
		}
		else
		{
			cout << "Нельзя решить систему данным методом: ведущий элемент стал равен 0\n";
			return x;
		}
	}
	for (int i = n - 1; i >= 0; i--)  // обратный ход, получение решений уравнений
	{
		for (int j = n - 1; j > i; j--)
		{
			b[i] -= x[j] * A[i][j];
		}
		x[i] = b[i] / A[i][i];
	}
	return x;
}
double* gauss2(double A[n][n], double* b, int n)
{
```

```
	double* x = new double[n];
	for (int i = 0; i < n; i++)  //прямой ход, приведение к верхетреугольному виду
	{
		int max = abs(A[i][i]);
		int maxIndex = i;
		for (int j = i + 1; j < n; j++)  //поиск наибольшего главного элемента в столбце
		{
			if (abs(A[j][i]) > max)
			{
				maxIndex = j;
				max = abs(A[j][i]);
			}
		}
		for (int j = 0; j < n; j++)  //перестановка уравнения с наибольшим главным элементом
		{
			swap(A[i][j], A[maxIndex][j]);
		}
		swap(b[i], b[maxIndex]);
		if (A[i][i] != 0)		// аналогично первой схеме решения
		{
			for (int j = i + 1; j < n; j++)
			{
				double q = A[j][i] / A[i][i];
				for (int k = i; k < n; k++)
				{
					if (k == i)
					{
						A[j][k] = 0;
					}
					else
					{
						A[j][k] -= q * A[i][k];
					}
				}
				b[j] -= q * b[i];
			}
		}
		else
		{
			cout << "Нельзя решить систему данным методом: ведущий элемент стал равен 0\n";
			return x;
		}
	}
	for (int i = n - 1; i >= 0; i--)
	{
		for (int j = n - 1; j > i; j--)
		{
			b[i] -= x[j] * A[i][j];
		}
		x[i] = b[i] / A[i][i];
	}
	return x;
}
```

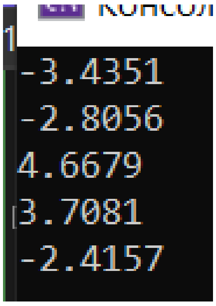
```
double* gauss3(double A[n][n], double* b, int n)
{
        double* x = new double[n];

        for (int i = 0; i < n; i++)  //прямой ход
        {
                double max = abs(A[i][0]);
                int maxIndexX = i;
                int maxIndexY = 0;
                for (int j = i; j < n; j++)  //поиск наибольшего элемента в
матрице под текущей строкой
                {
                        for (int k = 0; k < n; k++)
                        {
                                if (abs(A[j][k]) > max)
                                {
                                        maxIndexX = j;
                                        maxIndexY = k;
                                        max = abs(A[j][k]);
                                }
                        }
                }
                for (int j = 0; j < n; j++)  //перестановка уравнения с
наибольшим главным элементом
                {
                        swap(A[i][j], A[maxIndexX][j]);
                }
                swap(b[i], b[maxIndexX]);
                if (A[i][maxIndexY] != 0)
                {
                        for (int j = i + 1; j < n; j++)
                        {
                                double q = A[j][maxIndexY] / A[i][maxIndexY];
//получаем q
                                for (int k = 0; k < n; k++)  //вычитаем i-ую
строку умноженную на q из последующих строк
                                {
                                        if (k == maxIndexY)
                                        {
                                                A[j][k] = 0;  //исключение
погрешности для нулевых элементов
                                        }
                                        else
                                        {
                                                A[j][k] -= q * A[i][k];
                                        }
                                }
                                b[j] -= q * b[i];
                        }
                }
                else
                {
                        cout << "Нельзя решить систему данным методом: элементы
в оставшихся строках стали нулевыми (система не имеет единственное
решение)\n";
                        return x;
                }
        }
```

```
	bool* xFound = new bool[n];
	for (int i = 0; i < n; i++)
	{
		xFound[i] = 0;
	}
	for (int i = n - 1; i >= 0; i--)  //обратный ход, получение решений 
уравнений
	{
		int curX;
		for (int j = 0; j < n; j++)
		{
			if (A[i][j] != 0 && xFound[j] == 0)
			{
				curX = j;
			}
			if (A[i][j] != 0 && xFound[j] == 1)
			{
				b[i] -= x[j] * A[i][j];
			}
		}
		x[curX] = b[i] / A[i][curX];
		xFound[curX] = 1;
	}
	return x;
}


int main()
{
	setlocale(LC_ALL, "ru");
	std::cout.precision(4);		// Устанавливает точность
	std::cout.setf(std::ios::fixed, std::ios::floatfield);
	double b[n] = { 4.2, 4.2, 4.2, 4.2, 4.2 };

	double C[n][n] = { {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} };

	double D[n][n] = { {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} };

	double A[n][n]{};
	for (int i = 0; i < n; i++)
	{
		for (int j = 0; j < n; j++)
		{
			A[i][j] = 27 * C[i][j] + D[i][j];
		}
	}
	for (int i = 0; i < n; i++)
	{
		for (int j = 0; j < n; j++)
		{
```

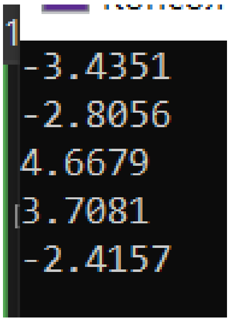
```
				cout << A[i][j] << " ";
			}
			cout << '\n';
		}
		cout << "\n\n";
		int choise;
		while (true)
		{
			cout << "1. Метод Гаусса (схема единственного деления)\n";
			cout << "2. Метод Гаусса (схема частичного выбора)\n";
			cout << "3. Метод Гаусса (схема полного выбора)\n";
			cout << "4. Выход\n";
			cin >> choise;
			system("cls"); // Очистка экрана консоли
			if (choise == 1)
			{
				double* x = gauss1(A, b, n);
				for (int i = 0; i < n; i++)
				{
					cout << x[i] << "\n";
				}
				return 0;
			}
			else if (choise == 2)
			{
				double* x = gauss2(A, b, n);
				for (int i = 0; i < n; i++)
				{
					cout << x[i] << "\n";
				}
				return 0;
			}
			else if (choise == 3)
			{
				double* x = gauss3(A, b, n);
				for (int i = 0; i < n; i++)
				{
					cout << x[i] << "\n";
				}
				return 0;
			}
			else if (choise == 4)
			{
				return 0;
			}
		}
	}
```

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

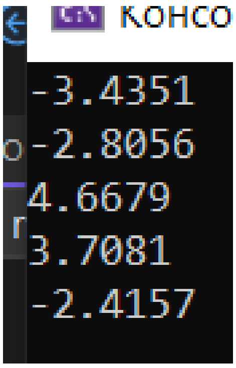
Метод Гаусса (схема единственного деления):



Метод Гаусса (схема частичного выбора):



Метод Гаусса (схема полного выбора):



ТЕСТОВЫЙ ПРИМЕР

```
double b[n] = { 3.2, 1.2, 7.2, 3.2, 2.2 };

double C[n][n] = { {0.4000,0.0300,0.2000,0.0000,0.0000},
                   {0.0000,0.2000,0.0000,0.2000,0.0000},
                   {0.2000,0.0000,0.2000,0.0000,0.2000},
                   {0.8000,0.2000,0.0000,0.2000,0.0000},
                   {0.0000,0.9000,0.2000,0.0000,0.2000} };

double D[n][n] = { {3.3300,1.8100,0.6700,0.9200,-0.5300},
                   {-0.5300,2.3300,0.8100,0.6700,0.9200},
                   {0.9200,-1.5300,2.3300,0.8100,0.6700},
                   {0.6700,0.9200,-0.5300,2.3300,0.8100},
                   {0.8100,2.6700,0.9200,-0.5300,2.3300} };
```

Должны были получится следующие корни: 0.0392, -0.0780, 0.1915, 0.1511, 0.2926

Согласно результатам программы:

Метод Гаусса (схема единственного деления):

```
0.0392
-0.0780
0.1915
0.1511
0.2926
```

Метод Гаусса (схема частичного выбора):

```
0.0392
-0.0780
0.1915
0.1511
0.2926
```

Метод Гаусса (схема полного выбора):

```
0.0392
-0.0780
0.1915
0.1511
0.2926
```

Что соответствует ожидаемым корням.

Должны были получится следующие корни: 0.4795, -1.9364, 0.2506, 0.5368, 1.0762

Согласно результатам программы (метод Гаусса (схема единственного деления), метод Гаусса (схема полного выбора), метод Гаусса (схема полного выбора)):

```
0.4795
-1.9364
0.2506
0.5368
1.0762
```

Что соответствует ожидаемым корням.

Должны были получится следующие корни: 1.7630, 2.2925, 0.6282, 0.2520, 1.3607

Согласно результатам программы (метод Гаусса (схема единственного деления), метод Гаусса (схема полного выбора), метод Гаусса (схема полного выбора)):

```
1.7630
2.2925
0.6282
0.2520
1.3607
```

Что соответствует ожидаемым корням.

-

По результату решений не должно быть.

Согласно результатам программы (метод Гаусса (схема единственного деления), метод Гаусса (схема полного выбора), метод Гаусса (схема полного выбора)):

```
Нельзя решить систему данным методом: ведущий элемент стал равен 0
```

Что соответствует ожидаемому результату.

Должны были получится следующие корни: 0.1677, -0.3740, 0.1014, -0.2280, 0.0147.

Согласно результатам программы (метод Гаусса (схема единственного деления), метод Гаусса (схема полного выбора), метод Гаусса (схема полного выбора)):

```
Выбрать Ко
0.1677
-0.3740
0.1014
-0.2280
0.0147
```

Что соответствует ожидаемому результату.

ОЦЕНКА ПОГРЕШНОСТИ

$\Delta(x_1) = |x - x_1| = |-3.4351 + 3.43512707808| = 0.00002707808$

$\Delta(x_2) = |x - x_2| = |-2.8056 + 2.80562722508| = 0.00002722508$

$\Delta(x_3) = |x - x_3| = |4.6679 - 4.66791846713| = 0.00001846713$

$\Delta(x_4) = |x - x_4| = |3.7081 - 3.70813063443| = 0.00003063443$

$\Delta(x_5) = |x - x_5| = |-2.4157 + 2.41574544657| = 0.00004544657$

$$\delta(x_1)\frac{\Delta(x_1)}{|x_1|} = \frac{0.00002707808}{0.4351} = 0.00006223415307$$

$$\delta(x_2)\frac{\Delta(x_2)}{|x_2|} = \frac{0.00002722508}{0.8056} = 0.00003379478649$$

$$\delta(x_3)\frac{\Delta(x_3)}{|x_3|} = \frac{0.00001846713}{0.6679} = 0.00002764954334$$

$$\delta(x_4)\frac{\Delta(x_4)}{|x_4|} = \frac{0.00003063443}{0.7081} = 0.00004326285835$$

$$\delta(x_5)\frac{\Delta(x_5)}{|x_5|} = \frac{0.00004544657}{0.4157} = 0.0001040723078$$

ВЫВОД

В ходе выполнения данной лабораторной работы был применён метод Гаусса по схеме единственного деления, 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, также были составлены алгоритмы и созданы реализации соответствующих программ на языке C++ для решения поставленной задачи. Была произведена оценка точности результатов.