

**Федеральное государственное автономное образовательное
учреждение высшего образования «Национальный
исследовательский университет ИТМО»**

**Факультет программной инженерии и компьютерной
техники**

Вычислительная математика

Лабораторная работа №1

Вариант 24

Студент: Крикунов Олег Евгеньевич

Р3267

Преподаватель: Машина Екатерина Алексеевна

Оценка: _____

Подпись преподавателя: _____

1. Цели работы

Написать программу, которая бы решала систему алгебраических уравнений указанным математическим методом.

2. Описание метода, расчётные формулы

Метод Гаусса с выбором главного элемента по столбцам — это алгоритм численного метода решения систем линейных уравнений путем последовательного исключения переменных. Он основан на приведении системы к треугольному виду и последующем обратном ходе.

Основные шаги метода Гаусса:

1. Прямой ход (Приведение к треугольному виду):

- Проходим по каждой строке системы уравнений.
- Для каждой строки выбираем ведущий элемент (наибольший по модулю) и используем его для обнуления всех элементов под ним в столбце.
- После этого матрица принимает треугольный вид.

2. Обратный ход:

- Начиная с последнего уравнения, находим значение последней неизвестной.
- Подставляем это значение в предыдущие уравнения и находим значения предыдущих неизвестных.
- Продолжаем этот процесс до тех пор, пока не найдем значения всех неизвестных.

3. Листинг программы

Код написан на языке C++ в редакторе Clion. Код программы:

```
#include <iostream>
#include <fstream>
#include <vector>
#include <cmath>
using namespace std;

const double E = 0.0001;

double CalculateDeterminant(vector<vector<double>> a, int n) {
```

```

double det = 1.0;

if (n == 2) {
    det = a[0][0] * a[1][1] - a[0][1] * a[1][0];
    return det;
}
if (n == 3) {
    det = a[0][0] * a[1][1] * a[2][2] + a[0][1] * a[1][2] * a[2][0] +
a[0][2] * a[1][0] * a[2][1] -
        a[0][2] * a[1][1] * a[2][0] - a[0][0] * a[1][2] * a[2][1] -
a[0][1] * a[1][0] * a[2][2];
    return det;
}

for (int i = 0; i < n; ++i) {
    int pivot = i;
    for (int j = i + 1; j < n; ++j) {
        if (abs(a[j][i]) > abs(a[pivot][i])) {
            pivot = j;
        }
    }
    if (pivot != i) {
        det *= -1.0;
        swap(a[i], a[pivot]);
    }
    det *= a[i][i];
    if (abs(det) < E) return 0.0;
    for (int j = i + 1; j < n; ++j) {
        double coefficient = a[j][i] / a[i][i];
        for (int k = i; k < n; ++k) {
            a[j][k] -= coefficient * a[i][k];
        }
    }
}
return det;
}

```

```

vector<double> GaussSolve(vector<vector<double>> a, vector<double> y, int
n) {
    vector<double> answers(n);
    int k, index;
    for (k = 0; k < n; k++) {
        double maxVal = abs(a[k][k]);
        index = k;
        for (int i = k + 1; i < n; i++) {
            if (abs(a[i][k]) > maxVal) {
                maxVal = abs(a[i][k]);
                index = i;
            }
        }

        swap(a[k], a[index]);
        swap(y[k], y[index]);

        for (int i = k; i < n; i++) {
            double temp = a[i][k];
            if (abs(temp) < E) continue;

```

```

        for (int j = k; j < n; j++)
            a[i][j] = a[i][j] / temp;
        y[i] = y[i] / temp;
        if (i == k) continue;
        for (int j = 0; j < n; j++)
            a[i][j] = a[i][j] - a[k][j];
        y[i] = y[i] - y[k];
    }
}

for (k = n - 1; k >= 0; k--) {
    answers[k] = y[k];
    for (int i = 0; i < k; i++)
        y[i] = y[i] - a[i][k] * answers[k];
}
return answers;
}

void PrintMatrix(vector<vector<double>> a, const vector<double>&b, int n) {
    cout << "Треугольная матрица:" << endl;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            cout << a[i][j] << "\t";
        }
        cout << "| " << b[i] << endl;
    }
}

void PrintResults(const vector<double>&x, const vector<double>&residuals) {
    cout << "Вектор неизвестных:" << endl;
    for (int i = 0; i < x.size(); ++i) {
        cout << "x[" << i << "] = " << x[i] << endl;
    }
    cout << "Вектор невязок:" << endl;
    for (int i = 0; i < residuals.size(); ++i) {
        cout << "r[" << i << "] = " << residuals[i] << endl;
    }
}

int main() {
    int n;
    cout << "Введите количество уравнений: ";
    cin >> n;
    if (n > 20) {
        cout << "Количество уравнений превышает допустимое";
        return 0;
    }

    vector<vector<double>> a(n, vector<double>(n));
    vector<double> y(n);

    cout << "Каким способом заполнить матрицу?" << "\n" << "1 - Через файл"
    << "\n" << "2 - Вручную" << "\n";
    int choice;
    cin >> choice;
    if (choice == 1) {
        string filename;

```

```

cout << "Введите имя файла: ";
cin >> filename;
ifstream file(filename);
if (!file.is_open()) {
    cout << "Ошибка открытия файла" << endl;
    return 1;
}
// Считываем элементы матрицы
for (int i = 0; i < n; ++i)
    for (int j = 0; j < n; ++j)
        if (!(file >> a[i][j])) {
            cout << "Ошибка чтения из файла" << endl;
            return 1;
        }
// Считываем элементы вектора
for (int i = 0; i < n; ++i)
    if (!(file >> y[i])) {
        cout << "Ошибка чтения из файла" << endl;
        return 1;
    }
file.close();
}
else {
    for (int i = 0; i < n; i++) {
        cout << "Введите коэффициенты " << i + 1 << "-го уравнения
через пробел: ";
        for (int j = 0; j < n; j++)
            cin >> a[i][j];
    }
    cout << "Введите правые части уравнений через пробел: ";
    for (int i = 0; i < n; i++)
        cin >> y[i];
}

PrintMatrix(a, y, n);
double det = CalculateDeterminant(a, n);
cout << "Определитель матрицы: " << det << endl;
if (det != 0.0) {
    vector<double> x = GaussSolve(a, y, n);
    vector<double> residuals(n);
    for (int i = 0; i < n; ++i) {
        double sum = 0.0;
        for (int j = 0; j < n; ++j) {
            sum += a[i][j] * x[j];
        }
        residuals[i] = y[i] - sum;
    }
    PrintResults(x, residuals);
}
else {
    cout << "Определитель матрицы равен 0 => система либо имеет
бесконечное множество решений, либо не имеет решений, т. е. несовместна.";
}

return 0;
}

```

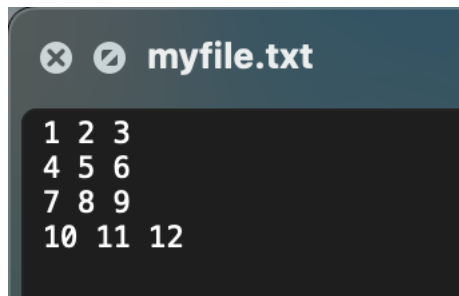
4. Примеры и результаты работы программы

Возьмем два примера и продемонстрируем работоспособность:

1. Ввод вручную

```
Введите количество уравнений: 3
Каким способом заполнить матрицу?
1 - Через файл
2 - Вручную
2
Введите коэффициенты 1-го уравнения через пробел: 3 7 9
Введите коэффициенты 2-го уравнения через пробел: 5 12 6
Введите коэффициенты 3-го уравнения через пробел: 47 16 90
Введите правые части уравнений через пробел: 23 45 99
Треугольная матрица:
3    7    9    | 23
5    12   6    | 45
47   16   90   | 99
Определитель матрицы: -2580
Вектор неизвестных:
x[0] = 2.23721
x[1] = 3.13023
x[2] = -0.624806
Вектор невязок:
r[0] = 0
r[1] = 0
r[2] = -1.42109e-14
```

2. Ввод из файла



```
Введите количество уравнений: 3
Каким способом заполнить матрицу?
1 - Через файл
2 - Вручную
2
Введите коэффициенты 1-го уравнения через пробел: 1 2 3
Введите коэффициенты 2-го уравнения через пробел: 4 5 6
Введите коэффициенты 3-го уравнения через пробел: 7 8 9
Введите правые части уравнений через пробел: 10 11 12
Треугольная матрица:
1  2  3  | 10
4  5  6  | 11
7  8  9  | 12
Определитель матрицы: 0
Определитель матрицы равен 0 => система либо имеет бесконечное множество решений, либо не имеет решений, т. е. несовместна.
```

3. Случайным образом

```

Введите количество уравнений: 20
Каким способом заполнить матрицу?
1 - Через файл
2 - Вручную
3 - Случайным образом
3
Треугольная матрица:
74 76 46 -23 77 -26 26 -11 35 -74 88 -1 90 -40 64 -69 61 62 -83 81 | -4
46 7 36 -43 -26 13 38 -25 93 35 8 -65 37 -91 5 13 -78 64 64 70 | 78
18 -61 40 56 -95 -64 90 43 -28 -39 -71 50 -14 96 -52 -97 69 93 -23 -40 | 32
-39 1 66 -54 -54 39 -60 90 -12 -16 -9 -45 -99 -70 -31 -66 14 -76 7 50 | 32
86 33 -10 97 -92 25 53 -62 98 66 36 79 0 -44 71 2 61 -28 -30 -54 | -98
6 -51 36 15 -74 64 -12 22 16 74 57 -71 49 -19 -29 69 -40 -22 -47 -53 | -31
41 57 -27 61 4 78 -30 -92 -99 -29 -58 -60 4 83 79 -10 88 88 92 -62 | 42
5 -45 28 -62 18 2 56 0 65 40 60 90 93 72 -26 -57 -57 -56 53 10 | -86
1 -11 -55 -84 24 79 -52 -24 30 67 30 45 71 -95 -69 -61 64 -19 -55 10 | 95
68 -41 39 -8 49 -28 -68 -9 41 48 -2 52 -62 0 46 36 -59 36 -5 -80 | -22
-17 54 4 27 91 29 95 -17 -89 -32 20 9 -84 -33 -49 78 -23 -86 23 54 | 29
-75 -74 62 -14 -48 -37 -62 -67 -19 90 -32 -19 58 96 -5 66 -28 -28 64 -78 | -99
46 -62 97 89 -99 55 66 3 -62 -19 -30 69 -17 -68 -91 -49 91 -19 12 -12 | -56
32 -2 -67 -21 -10 -75 59 89 -31 44 -23 11 -85 -41 -74 88 78 -44 3 -59 | -41
-3 -66 -20 27 -25 69 74 30 48 76 -46 -40 82 56 -72 -34 23 -16 85 -53 | 83
-50 97 97 -17 -60 0 -92 40 -38 57 -69 -1 19 -20 21 35 -74 -96 86 -69 | -93
-76 20 -43 83 -74 65 -56 -27 91 18 -1 7 60 39 63 61 -91 -97 93 -2 | 59
-27 34 66 87 -33 60 81 87 -38 -56 27 61 -13 -2 37 35 -16 67 60 23 | 78
-81 -15 40 -96 -35 92 -98 -84 42 42 -29 -42 21 13 -14 -85 -40 59 56 84 | 97
-26 -13 87 -1 64 17 0 -69 -94 -75 72 12 -100 -1 -24 -62 -92 -58 33 -44 | 86
Определитель матрицы: 7.20153e+42
Вектор неизвестных:
x[0] = -8.34161
x[1] = -4.81634
x[2] = 8.22572

```

5. Вывод

В ходе работы мы повторно изучили метод Гаусса с выбором главного элемента по столбцам. Написали программу, которая способная решать СЛАУ размерностью до 20. Данные могут подаваться с клавиатуры, из .txt файла и случайным образом.