

Санкт-Петербургский государственный университет
математико-механический факультет

Программа для решения
системы линейных
алгебраических уравнений
методом Крамера

Бода Александр Павлович

Механика и математическое моделирование 23.Б12-ММ

Рецензент: ассистент кафедры теоретической механики, Нестерчук Г. А.

Санкт-Петербург 2024

Содержание

1	Введение	2
2	Выбор инструментов	2
2.1	Среда разработки	2
2.2	Библиотеки	2
3	Документация по коду	2
3.1	Заголовочный файл <code>cramer.h</code>	2
3.2	Файл реализации <code>cramer.cpp</code>	3
3.2.1	Класс <code>CVector</code>	4
3.2.2	Класс <code>CMatrix</code>	4
3.2.3	Функция <code>determinant</code>	6
3.2.4	Функция <code>outputSystemLaTeX</code>	7
3.2.5	Функция <code>outputSolutionStepsLaTeX</code>	8
3.2.6	Функция <code>solveSystem</code>	8
3.2.7	функция <code>latexDet</code>	9
3.3	<code>main.cpp</code>	9
4	Примеры решения задач	12
5	Заключение	15

1 Введение

Задача решения системы линейных алгебраических уравнений (СЛАУ) является одной из наиболее распространенных задач в математике и её приложениях. Различные способы решения СЛАУ изучаются студентами Механико-Математического факультета СПбГУ на первом и втором семестре в курсе высшей алгебры. Изучение линейной алгебры побудило меня на создание универсальной программы, которая решала бы систему линейных уравнений любого количества переменных. В качестве одного из наиболее частоиспользуемых способ решения был выбран метод Крамера. Программа должна была быть написана на языке C++ и выводить все итерации решения в отдельный файл в формате LaTeX, который должен был автоматически компилироваться.

2 Выбор инструментов

2.1 Среда разработки

Для реализации проекта была выбрана среда разработки Visual Studio 2022.

2.2 Библиотеки

- `include <fstream>` - для вывода данных в файл.
- `include <iostream>` - для работы с потоком и стандартными функциями.
- `include <cmath>` - для работы с математическими функциями.
- `include <cstdlib>` - для работы с функцией `system()`.

3 Документация по коду

3.1 Заголовочный файл `cramer.h`

В этом файле названы классы `CMatrix` и `CVector`, а так же их методы, кроме этого названы еще 5 функций.

```
// cramer.h
#ifndef CRAMER_H
#define CRAMER_H

#include <fstream>

class CVector {
public:
    CVector(int size);
```

```

CVector(const CVector& other);
~CVector();
double& operator[](int index);
const double& operator[](int index) const;
int size() const;
void initializeVector();

private:
    double* elements;
    int n;
};

class CMatrix {
public:
    CMatrix(int size);
    CMatrix(const CMatrix& other);
    ~CMatrix();
    double* operator[](int index);
    const double* operator[](int index) const;
    int size() const;
    double det() const;
    void initializeMatrix();
    void replaceColumn(CVector b, int colIndex);

private:
    double** elements;
    int n;
};

void latexDet(std::ofstream& outfile, CMatrix A);
double determinant(double** matrix, int n);
void outputSystemLaTeX(std::ofstream& outfile, CMatrix A, CVector b);
void outputSolutionStepsLaTeX(std::ofstream& outfile, CMatrix A, CVector
    b, double* solutions);
void solveSystem(CMatrix A, CVector b, double* solutions, int n);

#endif

```

3.2 Файл реализации cramer.cpp

В этом файле реализованы классы и их методы, а так же функции.

• 3.2.1 Класс CVector

Данный класс создан для работы с вектором свободных членов, возникающем при решении СЛАУ. Класс имеет два параметра, первый из них - `double* elements` - это динамический массив, в котором хранятся координаты вектора, второй - `int n` - это размер вектора. Реализованы конструктор по умолчанию, конструктор копирования и деструктор. Имеется оператор `[]`, благодаря которому возвращается координата вектора нужного номера. Реализованы два метода: `int size()` - для возвращения размера вектора, `void initializeVector()` - для инициализации вектора.

```
CVector::CVector(int size) : n(size) {
    elements = new double[n];
}

CVector::CVector(const CVector& other) : n(other.n) {
    elements = new double[n];
    for (int i = 0; i < n; ++i) {
        elements[i] = other.elements[i];
    }
}

CVector::~CVector() {
    delete[] elements;
}

double& CVector::operator[](int index) {
    return elements[index];
}

const double& CVector::operator[](int index) const {
    return elements[index];
}

int CVector::size() const {
    return n;
}
```

3.2.2 Класс CMatrix

Класс был реализован для работы с квадратной матрицей коэффициентов СЛАУ. Класс имеет два параметра: `double** elements` - динамический массив из динамических массивов, в котором хранятся сами коэффициенты, и `int n` - размер матрицы. Класс имеет конструктор по умолчанию, конструктор копирования и деструктор. Реализовано несколько методов для работы с матрицей, одним из которых являет-

ся void replaceColumn(CVector b, int i), который заменяет в матрицей i-ый столбец на вектор b, а также double det(), который возвращает детерминант матрицы. Есть оператор [] для получения элемента матрицы по индексу.

```

CMatrix::CMatrix(int size) : n(size) {
    elements = new double* [n];
    for (int i = 0; i < n; ++i) {
        elements[i] = new double[n];
    }
}

CMatrix::CMatrix(const CMatrix& other) : n(other.n) {
    elements = new double* [n];
    for (int i = 0; i < n; ++i) {
        elements[i] = new double[n];
        for (int j = 0; j < n; ++j) {
            elements[i][j] = other.elements[i][j];
        }
    }
}

CMatrix::~CMatrix() {
    for (int i = 0; i < n; ++i) {
        delete[] elements[i];
    }
    delete[] elements;
}

double* CMatrix::operator[](int index) {
    return elements[index];
}

const double* CMatrix::operator[](int index) const {
    return elements[index];
}

int CMatrix::size() const {
    return n;
}

double CMatrix::det() const {
    return determinant(elements, n);
}

void CMatrix::initializeMatrix()
{
    for (int i = 0; i < n; ++i) {

```

```

        for (int j = 0; j < n; ++j) {
            std::cout << "Enter coefficient for x_" << j + 1 << "
                in equation " << i + 1 << ": ";
            std::cin >> elements[i][j];
        }
    }
    std::cout << std::endl;
}

void CVector::initializeVector()
{
    for (int i = 0; i < n; ++i) {
        std::cout << "Enter constant for equation " << i + 1 << ":
            ";
        std::cin >> elements[i];
    }
    std::cout << std::endl;
}

void CMatrix::replaceColumn(CVector b, int colIndex) {

    for (int i = 0; i < n; ++i) {
        elements[i][colIndex] = b[i];
    }

}

```

• 3.2.3 Функция determinant

Функция типа double, которая считает определитель квадратной матрицы с помощью рекурентного разложения по первой строке, отдельно рассмотрен случай матрицы, состоящей из одного элемента. В качестве аргумента функция использует динамическую матрицу (динамический массив из динамических массивов) и целочисленный размер этой матрицы. Функция возвращает определитель данной матрицы, тип возвращаемых данных - double.

```

double determinant(double** matrix, int n)
{
    if (n == 1) {
        return matrix[0][0];
    }
    double det = 0;
    double** submatrix = new double* [n - 1];
    for (int i = 0; i < n - 1; ++i) {
        submatrix[i] = new double[n - 1];
    }
}

```

```

for (int x = 0; x < n; ++x) {
    int subi = 0;
    for (int i = 1; i < n; ++i) {
        int subj = 0;
        for (int j = 0; j < n; ++j) {
            if (j == x) continue;
            submatrix[subi][subj] = matrix[i][j];
            ++subj;
        }
        ++subi;
    }
    det += (pow(-1, x) * matrix[0][x] * determinant(submatrix,
        n - 1));
}
for (int i = 0; i < n - 1; ++i) {
    delete[] submatrix[i];
}
delete[] submatrix;
return det;
}

```

• 3.2.4 Функция outputSystemLaTeX

Функция типа void для вывода СЛАУ в отдельный файл в формате LaTeX. В качестве аргумента использует файл, матрицу класса CMatrix, вектор свободных членов класса CVector. С помощью цикла выводит СЛАУ в файл.

```

void outputSystemLaTeX(std::ofstream& outfile, CMatrix A, CVector
    b) {
    int n = A.size();
    outfile << "\\begin{equation}\\n\\left\\{\\begin{array}{l}" <<
        std::string(n, 'r') << "\\n";
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            if (j > 0 && A[i][j] >= 0) outfile << "+";
            outfile << A[i][j] << "x_" << j + 1 << "} ";
        }
        outfile << " = " << b[i] << " \\n";
    }
    outfile << "\\end{array}\\right.\\n\\end{equation}\\n";
}

```

3.2.5 Функция outputSolutionStepsLaTeX

Функция выводит в отдельный файл все итерации, которые требуются для решения СЛАУ методом Крамера. Сначала вычисляется детерминант матрицы коэффициентов. После чего вычисляются значения переменных. На каждом этапе отдельно вычисляется детерминант матрицы, которая полностью совпадает с матрицей коэффициентов, за исключением одного столбца, который заменён на вектор свободных членов СЛАУ.

```
void outputSolutionStepsLaTeX(std::ofstream& outfile, CMatrix A,
    CVector b, double* solutions) {
    int n = A.size();
    double detA = A.det();
    outfile << "\\begin{equation}\\n\\text{Determinant of } A: \\\n\\Delta = ";
    latexDet(outfile, A);
    outfile << " = " << detA << "\\n\\end{equation}\\n";

    for (int i = 0; i < n; ++i) {
        CMatrix Ai(n);
        for (int j = 0; j < n; ++j) {
            for (int k = 0; k < n; ++k) {
                Ai[j][k] = A[j][k];
            }
        }
        Ai.replaceColumn(b, i);
        double detAi = Ai.det();
        outfile << "\\begin{equation}\\n\\Delta_{" << i + 1 << "} = ";
        latexDet(outfile, Ai);
        outfile << " = " << detAi << "\\n\\end{equation}\\n";
        outfile << "\\begin{equation}\\nx_{" << i + 1 << "} = ";
        outfile << "\\frac{\\Delta_{" << i + 1 << "}}{\\Delta} = \\frac{" << detAi << "}{" << detA << "} = " << solutions[i] << "\\n\\end{equation}\\n";
    }
}
```

• 3.2.6 Функция solveSystem

Функция типа void, которая решает СЛАУ, используя некоторые вышеперечисленные функции, а именно: метод det из класса CMatrix,

replaceColumn. Программа с помощью цикла находит значения каждый из неизвестных и записывает их в динамический массив, который состоит из решений.

```
void solveSystem(CMatrix A, CVector b, double* solutions) {
    int n = A.size();
    double detA = A.det();
    for (int i = 0; i < n; ++i) {
        CMatrix Ai = A;
        Ai.replaceColumn(b, i);
        solutions[i] = Ai.det() / detA;
    }
}
```

3.2.7 функция latexDet

Данная функция типа void принимает в качестве аргументов файл и матрица класса CMatrix. Функция выводит определитель матрицы в явном виде в необходимый файл.

```
void latexDet(std::ofstream& outfile, CMatrix A)
{
    int n = A.size();
    outfile << "det\\begin{vmatrix}\\n";
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n - 1; j++)
        {
            outfile << A[i][j] << " & ";
        }
        outfile << A[i][n - 1] << "\\\" << "\\\" << std::endl;
    }
    outfile << "\\end{vmatrix}";
}
```

3.3 main.cpp

В этом файле реализовано сама программа для решения СЛАУ методом Крамера. Используются функции из файла cramer.h. Так же несколько строк программы отведено на стандартные для каждой СЛАУ элементы файла, в который выводится решение. Программа выводит все итерации в формате LaTeX в файл под названием solution.tex. Если этот файл не существует, то он создаётся автоматически. Так же этот код из этого файла автоматически компилируется и создается файл формата .pdf с итоговым решением.

При запуске программы пользователем ему предлагается ввести количество переменных. После чего пользователь может последовательно ввести коэффициенты для уравнений из СЛАУ. Последним этапом является инициализация вектора свободных членов СЛАУ. Далее программа запускает решение, выводит ответ в консоль и в файл `soulution.tex`. Если СЛАУ не может быть решена, об этом сообщается пользователю.

```
// main.cpp
#include <iostream>
#include <fstream>
#include "cramer.h"
#include <cstdlib>

int main(int argc, char* argv[]) {
    int n;
    std::cout << "Enter the number of variables: ";
    std::cin >> n;
    if (n <= 0)
    {
        std::cerr << "The number of variables must be more than zero!";
    }

    CMatrix A(n);
    CVector b(n);
    double* solutions = new double[n];

    std::cout << "Enter the coefficients of matrix A:" << std::endl;
    A.initializeMatrix();

    std::cout << "Enter the constants vector b:" << std::endl;
    b.initializeVector();

    std::ofstream outfile("solution.tex");
    outfile <<
        "\\documentclass{article}\\n\\usepackage{amsmath}\\n\\begin{document}\\n";
    outfile << "\\section*{Solution of the System of Linear";
        Equations:}\\n";
    outputSystemLaTeX(outfile, A, b);

    double detA = 0;
    detA = A.det();

    if (detA == 0) {
```

```

        outfile << "\\text{The determinant of matrix of coefficients
            equals to zero. The system is unsolvable}\\n";
        outfile << "\\end{document}\\n";
        outfile.close();
        std::cerr << "The system does not have a unique solution." <<
            std::endl;
    }

    else if (detA != 0)
    {
        solveSystem(A, b, solutions);

        outfile << "\\section*{Solution using Cramer's Method:}\\n";
        outputSolutionStepsLaTeX(outfile, A, b, solutions);
        outfile << "\\section*{Answer:}\\n";
        outfile << "\\begin{equation}\\n\\left\\{\\begin{array}{r}\\n";
        for (int i = 0; i < n; ++i) {
            outfile << "x_" << i + 1 << "} = " << solutions[i] <<
                "\\\\n";
        }
        outfile << "\\end{array}\\right.\\n\\end{equation}\\n";
        outfile << "\\end{document}\\n";
        outfile.close();

        std::cout << "Solution written to solution.tex" << std::endl<<
            std::endl;

        std::cout << "Solution:" << std::endl;
        for (int i = 0; i < n; ++i) {
            std::cout << "x_" << i + 1 << " = " << solutions[i] <<
                std::endl;
        }
    }

    std::string fName = "solution.tex";

    std::string command = "pdflatex " + fName;

    int result = std::system(command.c_str());
    if (result == 0) {
        std::cout << "\\n\\nPDF file with compiled LaTeX code was
            generated and saved successfully.\\n";
    }
    else {
        std::cout << "\\n\\nPDF was not generated due to the error.\\n";
    }
}

```

```

delete[] solutions;

return EXIT_SUCCESS;
}

```

4 Примеры решения задач

Рассмотрим пример решения СЛАУ с помощью созданной программы.

```

Enter the number of variables: 3
Enter the coefficients of matrix A:
Enter coefficient for x_1 in equation 1: 8.584
Enter coefficient for x_2 in equation 1: 3.98
Enter coefficient for x_3 in equation 1: -0.583
Enter coefficient for x_1 in equation 2: 31.48
Enter coefficient for x_2 in equation 2: -7
Enter coefficient for x_3 in equation 2: 8.574
Enter coefficient for x_1 in equation 3: 12.9
Enter coefficient for x_2 in equation 3: 31.44
Enter coefficient for x_3 in equation 3: -15

Enter the constants vector b:
Enter constant for equation 1: 28
Enter constant for equation 2: 38
Enter constant for equation 3: -47.92

Solution written to solution.tex

Solution:
x_1 = -16.1415
x_2 = 58.1314
x_3 = 111.156
This is pdfTeX, Version 3.141592653-2.6-1.40.26 (MiKTeX 24.4) (preloaded format=pdflatex.fmt)
 restricted \write18 enabled.
entering extended mode
(solution.tex
LaTeX2e <2023-11-01> patch level 1
L3 programming layer <2024-04-11>
(C:\Users\1290723\AppData\Local\Programs\MiKTeX\tex\latex\base\article.cls
Document Class: article 2023/05/17 v1.4n Standard LaTeX document class
(C:\Users\1290723\AppData\Local\Programs\MiKTeX\tex\latex\base\size10.clo))
(C:\Users\1290723\AppData\Local\Programs\MiKTeX\tex\latex\amsmath\amsmath.sty
For additional information on amsmath, use the '?' option.
(C:\Users\1290723\AppData\Local\Programs\MiKTeX\tex\latex\amsmath\amstext.sty
(C:\Users\1290723\AppData\Local\Programs\MiKTeX\tex\latex\amsmath\amsgen.sty))
(C:\Users\1290723\AppData\Local\Programs\MiKTeX\tex\latex\amsmath\amsbsy.sty)
(C:\Users\1290723\AppData\Local\Programs\MiKTeX\tex\latex\amsmath\amsopn.sty))
(C:\Users\1290723\AppData\Local\Programs\MiKTeX\tex\latex\l3backend\l3backend-p
dftex.def) (solution.aux) [1{C:/Users/1290723/AppData/Local/MiKTeX/fonts/map/pd
ftex/pdftex.map}] (solution.aux) >C:/Users/1290723/AppData/Local/Programs/MiK
TeX/fonts/type1/public/amsfonts/cm/cmbx12.pfb><C:/Users/1290723/AppData/Local/Pr
ograms/MiKTeX/fonts/type1/public/amsfonts/cm/cmex10.pfb><C:/Users/1290723/AppDa

```

Рис. 1: Ввод данных в консоль

Далее представлен код в формате LaTeX, который был выведен в файл solution.tex при решении данной СЛАУ.

```

\documentclass{article}
\usepackage{amsmath}
\begin{document}

```

```

\section*{Solution of the System of Linear Equations:}
\begin{equation}
\left\{\begin{array}{rrr}
8.584x_{\{1\}} + 3.98x_{\{2\}} - 0.583x_{\{3\}} = 28 \\
31.48x_{\{1\}} - 7x_{\{2\}} + 8.574x_{\{3\}} = 38 \\
12.9x_{\{1\}} + 31.44x_{\{2\}} - 15x_{\{3\}} = -47.92
\end{array}\right.
\end{equation}
\section*{Solution using Cramer s Method:}
\begin{equation}
\text{Determinant of } A: \Delta = \det\begin{vmatrix}
8.584 & 3.98 & -0.583 \\
31.48 & -7 & 8.574 \\
12.9 & 31.44 & -15
\end{vmatrix} = 277.265
\end{equation}
\begin{equation}
\Delta_{\{1\}} = \det\begin{vmatrix}
28 & 3.98 & -0.583 \\
38 & -7 & 8.574 \\
-47.92 & 31.44 & -15
\end{vmatrix} = -4475.47
\end{equation}
\begin{equation}
x_{\{1\}} = \frac{\Delta_{\{1\}}}{\Delta} = \frac{-4475.47}{277.265} = -16.1415
\end{equation}
\begin{equation}
\Delta_{\{2\}} = \det\begin{vmatrix}
8.584 & 28 & -0.583 \\
31.48 & 38 & 8.574 \\
12.9 & -47.92 & -15
\end{vmatrix} = 16117.8
\end{equation}
\begin{equation}
x_{\{2\}} = \frac{\Delta_{\{2\}}}{\Delta} = \frac{16117.8}{277.265} = 58.1314
\end{equation}
\begin{equation}
\Delta_{\{3\}} = \det\begin{vmatrix}
8.584 & 3.98 & 28 \\
31.48 & -7 & 38 \\
12.9 & 31.44 & -47.92
\end{vmatrix} = 30819.7
\end{equation}
\begin{equation}
x_{\{3\}} = \frac{\Delta_{\{3\}}}{\Delta} = \frac{30819.7}{277.265} = 111.156
\end{equation}
\section*{Answer:}
\begin{equation}
\left\{\begin{array}{r}
x_{\{1\}} = -16.1415 \\
\end{array}\right.

```

```

x_{2} = 58.1314\\
x_{3} = 111.156\\
\end{array}\right.
\end{equation}
\end{document}

```

Solution of the System of Linear Equations:

$$\begin{cases} 8.584x_1 + 3.98x_2 - 0.583x_3 = 28 \\ 31.48x_1 - 7x_2 + 8.574x_3 = 38 \\ 12.9x_1 + 31.44x_2 - 15x_3 = -47.92 \end{cases} \quad (1)$$

Solution using Cramer's Method:

$$\text{Determinant of } A : \Delta = \det \begin{vmatrix} 8.584 & 3.98 & -0.583 \\ 31.48 & -7 & 8.574 \\ 12.9 & 31.44 & -15 \end{vmatrix} = 277.265 \quad (2)$$

$$\Delta_1 = \det \begin{vmatrix} 28 & 3.98 & -0.583 \\ 38 & -7 & 8.574 \\ -47.92 & 31.44 & -15 \end{vmatrix} = -4475.47 \quad (3)$$

$$x_1 = \frac{\Delta_1}{\Delta} = \frac{-4475.47}{277.265} = -16.1415 \quad (4)$$

$$\Delta_2 = \det \begin{vmatrix} 8.584 & 28 & -0.583 \\ 31.48 & 38 & 8.574 \\ 12.9 & -47.92 & -15 \end{vmatrix} = 16117.8 \quad (5)$$

$$x_2 = \frac{\Delta_2}{\Delta} = \frac{16117.8}{277.265} = 58.1314 \quad (6)$$

$$\Delta_3 = \det \begin{vmatrix} 8.584 & 3.98 & 28 \\ 31.48 & -7 & 38 \\ 12.9 & 31.44 & -47.92 \end{vmatrix} = 30819.7 \quad (7)$$

$$x_3 = \frac{\Delta_3}{\Delta} = \frac{30819.7}{277.265} = 111.156 \quad (8)$$

Answer:

$$\begin{cases} x_1 = -16.1415 \\ x_2 = 58.1314 \\ x_3 = 111.156 \end{cases} \quad (9)$$

Рис. 2: Скомпилированный файл solution.tex

5 Заключение

В заключении можно сказать, что проект, целью которого было написание программы для решение СЛАУ методом Крамера, успешно реализован. Созданная программа не имеет видимых замечаний и может быть использована без каких-либо доработок.