Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«Вычисление арифметических выражений»**

**Выполнил**:

студент группы 382003-1

Боев Р.А.

**Проверил**:

ассистент каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2019

**Содержание**

[Постановка задачи 3](#_Toc26962562)

[Метод решения 4](#_Toc26962563)

[Руководство пользователя 5](#_Toc26962564)

[Описание программной реализации 6](#_Toc26962565)

[Подтверждение корректности 7](#_Toc26962566)

[Результаты экспериментов 8](#_Toc26962567)

[Заключение 9](#_Toc26962568)

[Приложение 10](#_Toc26962569)

# Постановка задачи

Реализовать метод Гаусса для квадратной матрицы с элементами типа double с выбором ведущего элемента.

# Метод решения

Реализовать класс шаблонного вектора (как шаблонный массив определённой длины), также перегрузить необходимые операции и добавить некоторые полезные функции для удобства (функция swap для перемены строк местами, функции позволяющие получить доступ к защищённым полям и др.)

Реализовать класс шаблонной матрицы как наследник шаблонного вектора с элементами – шаблонными векторами. Добавить функции, позволяющие задать матрицу и напечатать её, также функцию collmax для нахождения максимального (по мод.) элемента в определённом столбце начиная с i0-й строки.

Реализовать класс СЛАУ как наследник шаблонной матрицы. Матрица квадратная (высота равна ширине) с элементами типа double. Также добавить новое поле – вектор значений (VOV) и функцию, позволяющую задать этот вектор (setVOV). Реализовать функцию gaussSolve, которая решает СЛАУ следующим методом:

- последовательно в каждом i-м столбце из элементов с номером >=i находится номер indmax максимального элемента

- меняются строки с номерами indmax и i

- если максимальный элемент не 0, то при помощи строки с максимальным элементом (которую перед этим поделили на максимальный элемент) происходит зануление всех элементов в i-м столбце (посредством элементарных преобразований со строками). Решением в таком случае

- в приведённой матрице проверяются нулевые строки: если некая строка (а точнее некий элемент на диагонали) зануляется, а соотвествующее ей значений вектора значений не 0, то решений нет, если 0 то решений больше чем одно

# Руководство пользователя

- После запуска программа потребует ввести размер матрицы - введите

- Программа потребует ввести элементы матрицы – введите их построчно, после этого матрица будет выведена на экран (для удобства)

- Программа потребует ввести вектор значений – введите

- После этого программа попросит богов решить эту матрицу, и решение будет выведено на экран (слава богам). Либо появится надпись, что решений нет или решений больше чем одно.

# Описание программной реализации

Код программы находится в одном файле. Также содержатся прочие файлы, созданные Visual Studio.

# Подтверждение корректности

Как мы видим, программа считает правильно с точностью до 10^(-8), правильно определяет когда решений нет и когда больше чем одно.

# Результаты экспериментов

По данным экспериментов видно, что всё считается с нужной точностью.

# Заключение

Метод Гаусса для квадратной матрицы с элементами типа double с выбором ведущего элемента реализован. Всё считается с нужной точностью.

# Приложение

Основной код это задание СЛАУ с функцией самого метода Гаусса

class SLAU : public matrix<double>

{

private:

vector<double> VOV;

public:

SLAU(size\_t height) : matrix(height, height)

{

VOV = vector<double>(height);

}

void setVOV()

{

VOV.setdata();

}

vector<double> gaussSolve()

{

size\_t indmax;

for (size\_t i = 0; i < width; i++)

{

indmax = collmax(i, i);

double max = this->data[indmax][i];

if (indmax > i)

{

this->swap(indmax, i);

VOV.swap(indmax, i);

}

indmax = i;

if (abs(max) != 0)

{

this->data[indmax] /= max;

VOV[indmax] /= max;

for (size\_t j = 0; j < this->size; j++)

{

if (j != indmax)

{

VOV[j] -= (VOV[indmax] \* this->data[j][i]);

this->data[j] -= (this->data[indmax] \* this->data[j][i]);

}

}

}

}

for (size\_t i = 0; i < this->size; i++)

{

if (this->data[i][i] == 0)

{

if (VOV[i] == 0)

cout << "More than one solutions" << endl;

else cout << "No solution" << endl;

exit(1);

}

}

for (size\_t i = 0; i < VOV.getsize(); i++)

{

double a = VOV[i];

if (abs(a) < 0.00000001)

{

VOV[i] = 0;

}

}

return VOV;

}

vector<double> gaussSolve2(vector<double> vec) /\* нужно для того чтобы функция имела нужный вид

{

VOV = vec;

return gaussSolve();

}

};

Также функция, которая вызывает метод Гаусса (для SLAU M = SLAU(height)) :

vector<double> Sol = M.gaussSolve2(X); (X это vector<double>(height))