

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

Маслов Александр Евгеньевич

Проверил:

ассистент каф. МОСТ,

Волокитин В.Д.

Нижний Новгород
2021

Содержание

Постановка задачи	3
Метод решения	4
Руководство пользователя	6
Описание программной реализации	8
Подтверждение корректности	9
Результаты экспериментов	10
Заключение.....	11
Приложение	12

Постановка задачи

Необходимо реализовать метод Гаусса (Жордана-Гаусса) для квадратной матрицы произвольного размера.

Метод решения

Функция `MaxInCol()` принимает номер столбца и нижнюю границу сравнения. Изначально принимает за максимальный элемент, элемент на главной диагонали. А номер максимального элемента его номер в массиве соответственно. Далее циклом `for` максимальный элемент сравнивается со всеми элементами в столбце ниже него. Если такой элемент находится, то он становится максимальным, а номер максимального элемента становится номером этого элемента соответственно. Функция возвращает номер максимального элемента.

Функция `Swap()` принимает номера элементов двух векторов в разных строках и меняет их местами. Алгоритм смены местами проводится путем создания временной переменной `tmp` и записи в неё значения первого вектора.

Функция `CompareAndSwap()` принимает по ссылке исходную матрицу и номер столбца. Меняет местами (функция `swap()`) строку с элементом на главной диагонали со строкой с максимальным элементом ниже неё (функция `MaxInCol()`)

Функция `Action()` принимает по ссылке преобразованную функцией `CompareAndSwap()` матрицу, номер столбца и размерность матрицы. Исходная строка делится на элемент на главной диагонали для того, чтобы он стал равен 1. Создаётся временный вектор `tmp`, равный преобразованной исходной строке. Далее вектор `tmp` умножается на элемент ниже главной диагонали следующей строки. Из всей следующей строки вычитается вектор `tmp`. вектор `tmp` снова становится равным исходной строке. Далее, циклом `for` данный алгоритм проводится для всех элементов ниже главной диагонали. Таким образом, функция `Action()` зануляет все элементы, ниже главной диагонали в исходном столбце.

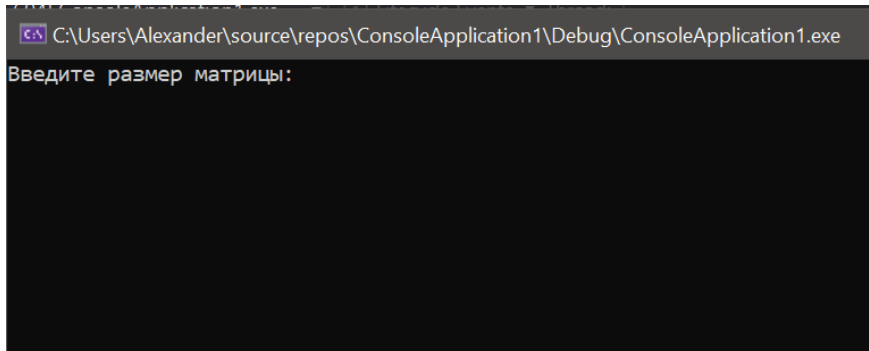
Функция `Action2()` является аналогом функции `Action2()`, работает по аналогичному алгоритму, но зануляет элементы, выше главной диагонали в исходном столбце.

Функция `GetAnswer()` принимает по ссылке преобразованную после функций `Magic` и `Magic2` матрицу, размерность матрицы. Сначала проводится аналог проверки матрицы на линейную зависимость. Если система линейно зависима, то в последнем столбце обязательно будут элементы со значениями `-nan(ind)`. Поэтому проводится проверка на то, являются ли элементы в последнем столбце действительными числами. Если это так то выполняется цикл вывода ответа, в противном случае, выводится соответствующее сообщение. Вывод ответа: выводит последний столбец матрицы.

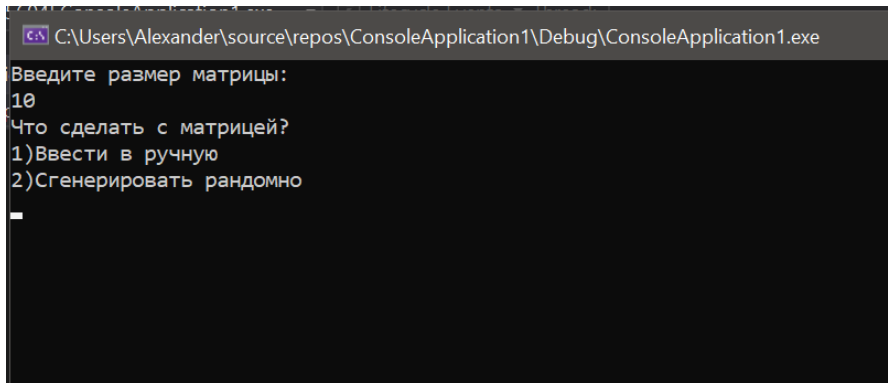
Функция `main()` (см. приложение). Запрашивает ввести размерность матрицы. Далее программа спрашивает, способ заполнения матрицы 1 - заполнение матрицы вручную, 2 - случайное заполнение матрицы. Независимо от выбора способа заполнения матрицы алгоритм преобразований выполняется одинаково. Цикл `for` выполняет функцию `CompareAndSwap()` и `Magic()`, тем самым зануляя все элементы ниже главной диагонали. Далее циклом `for` выполняется функция `Magic2`, тем самым, зануляя все элементы выше главной диагонали. Итогом всех преобразований является единичная матрица исходной размерности и столбец ответов правее неё. В конце функция `GetAnswer()` выводит этот последний столбец (если система линейно независима).

Руководство пользователя

После запуска, программа запрашивает ввести размер матрицы.



Далее пользователю необходимо выбрать способ заполнения матрицы.



В случае выбора ввода матрицы вручную, программа запросит ввести данные матрицы (коэффициенты при x и столбец ответов). В случае выбора рандомной генерации, матрица заполнится автоматически случайными числами.

Результатом программы служит список найденных неизвестных.

```
Microsoft Visual Studio Debug Console

Ваша матрица:
41 29 34 40 17 16 12 0 40 20 35
5 13 27 7 29 13 14 39 18 9 30
38 27 40 34 33 26 20 29 29 12 29
30 21 4 5 7 25 24 17 39 32 9
33 2 21 25 29 22 11 8 28 39 3
7 29 39 19 22 10 11 14 36 36 34
10 38 28 10 20 33 14 9 10 20 12
21 33 20 15 23 18 4 24 38 10 26
29 16 18 15 14 25 35 34 32 24 39
35 3 35 19 6 24 18 33 38 5 0

Матрица после преобразований:
1 0 0 0 0 0 0 0 0 0 -3.58806
0 1 0 0 0 0 0 0 0 0 -0.623461
0 0 1 0 0 0 0 0 0 0 -1.31187
0 0 0 1 0 0 0 0 0 0 2.42102
0 0 0 0 1 0 0 0 0 0 1.77088
0 0 0 0 0 1 0 0 0 0 1.01468
0 0 0 0 0 0 1 0 0 0 3.1382
0 0 0 0 0 0 0 1 0 0 -1.60878
0 0 0 0 0 0 0 0 1 0 2.60729
0 0 0 0 0 0 0 0 0 1 -2.01682

Ответ
x1 = -3.58806
x2 = -0.623461
x3 = -1.31187
x4 = 2.42102
x5 = 1.77088
x6 = 1.01468
x7 = 3.1382

Microsoft Visual Studio Debug Console

Ваша матрица:
41 29 34 40 17 16 12 0 40 20 35
5 13 27 7 29 13 14 39 18 9 30
38 27 40 34 33 26 20 29 29 12 29
30 21 4 5 7 25 24 17 39 32 9
33 2 21 25 29 22 11 8 28 39 3
7 29 39 19 22 10 11 14 36 36 34
10 38 28 10 20 33 14 9 10 20 12
21 33 20 15 23 18 4 24 38 10 26
29 16 18 15 14 25 35 34 32 24 39
35 3 35 19 6 24 18 33 38 5 0

Матрица после преобразований:
1 0 0 0 0 0 0 0 0 0 -3.58806
0 1 0 0 0 0 0 0 0 0 -0.623461
0 0 1 0 0 0 0 0 0 0 -1.31187
0 0 0 1 0 0 0 0 0 0 2.42102
0 0 0 0 1 0 0 0 0 0 1.77088
0 0 0 0 0 1 0 0 0 0 1.01468
0 0 0 0 0 0 1 0 0 0 3.1382
0 0 0 0 0 0 0 1 0 0 -1.60878
0 0 0 0 0 0 0 0 1 0 2.60729
0 0 0 0 0 0 0 0 0 1 -2.01682

Ответ
x1 = -3.58806
x2 = -0.623461
x3 = -1.31187
x4 = 2.42102
x5 = 1.77088
x6 = 1.01468
x7 = 3.1382
x8 = -1.60878
x9 = 2.60729
x10 = -2.01682
```

Описание программной реализации

Шаблонный класс вектор (Vector), представляющий из себя класс динамического массива с соответствующими методами и перегруженными операциями.

Шаблонный класс матрица (Matrix), как наследник класса вектор, представляет собой вектор векторов с соответствующими методами.

Функция CompareAndSwap - ставит на место главной диагонали максимальный элемент.

Функция Action - зануляет элементы ниже главной диагонали.

Функция Action2 - зануляет элементы выше главной диагонали.

Функция GetAnswer - выводит список решений.

Функция main (см. приложение) – запрашивает размерность матрицы, способ заполнения матрицы, в случае ввода вручную, значения матрицы. Функции Magic и Magic2 преобразуют матрицу до приведенного вида. Функция GetAnswer выводит решения.

Подтверждение корректности

Для проверки на линейную зависимость системы в функции GetAnswer проводится проверка того, является ли ответ набором действительных чисел. В противном случае, результату будет присвоено значение -nan(ind), и программа выдаст соответствующее сообщение.

```
Microsoft Visual Studio Debug Console
Введите размер матрицы:
2
Что сделать с матрицей?
1)Ввести в ручную
2)Сгенерировать рандомно
1
1 2 3
2 4 6

Ваша матрица:
1 2 3
2 4 6

Матрица после преобразований:
-nan(ind) -nan(ind) -nan(ind)
-nan(ind) -nan(ind) -nan(ind)

Ответ
Система линейна зависима. Ответ не верный.
```

```
Microsoft Visual Studio Debug Console
Введите размер матрицы:
3
Что сделать с матрицей?
1)Ввести в ручную
2)Сгенерировать рандомно
1
1 2 3 4
5 4 3 2
1 4 2 5

Ваша матрица:
1 2 3 4
5 4 3 2
1 4 2 5

Матрица после преобразований:
1 0 0 -1
0 1 0 1
0 0 1 1

Ответ
x1 = -1
x2 = 1
x3 = 1
```

Калькулятор Инструкция

Решить систему линейных уравнений методом Гаусса

Количество неизвестных величин в системе: 3

Изменить названия переменных в системе

Заполните систему линейных уравнений:

$$\begin{cases} 1x_1 + 2x_2 + 3x_3 = 4 \\ 5x_1 + 4x_2 + 3x_3 = 2 \\ 1x_1 + 4x_2 + 2x_3 = 5 \end{cases}$$

Решить систему уравнений

Для подтверждения корректности результата программы, его можно сравнить с любым онлайн калькулятором метода Гаусса в интернете.

Ответ:

$$\begin{cases} x_1 = -1 \\ x_2 = 1 \\ x_3 = 1 \end{cases}$$

Результаты экспериментов

По данным экспериментов видно, что программа работает корректно, в пределах допустимой точности.

```
Microsoft Visual Studio Debug Console
Введите размер матрицы:
2
Что сделать с матрицей?
1)Ввести в ручную
2)Сгенерировать рандомно
1
333333333333333333 33333333333333333333333333333333 1111111111111111
222222222222222222 566666666666666666666666666666 2222222222222222

Ваша матрица:
3.33333e+17  3.33333e+24  1.11111e+13
2.22222e+19  5.66667e+21  2.22222e+12

Матрица после преобразований:
1  0  9.91525e-08
0  1  3.32342e-12

Ответ
x1 = 9.91525e-08
x2 = 3.32342e-12
Action 2 (A - 5)

Microsoft Visual Studio Debug Console
Введите размер матрицы:
2
Что сделать с матрицей?
1)Ввести в ручную
2)Сгенерировать рандомно
1
333333333333333333 33333333333333333333333333333333 1111111111111111
222222222222222222 566666666666666666666666666666 2222222222222222

Ваша матрица:
3.33333e+17  3.33333e+24  1.11111e+13
2.22222e+19  5.66667e+21  2.22222e+12

Матрица после преобразований:
1  0  9.91525e-08
0  1  3.32342e-12

Ответ
x1 = 9.91525e-08
x2 = 3.32342e-12
```

Заключение

По результатам экспериментов видно, что все работает отлично. Но, во избежание потери точности можно установить большее количество знаков после запятой.

Приложение

```
int main()
{
    size_t a;
    int choice;

    Matrix<double> A;
    cout << "Enter the dimension of the matrix:" << endl;
    cin >> a;

    cout << "What to do with the matrix?" << endl;
    cout << "1)Enter the matrix manually" << endl << "2)Generate matrix randomly" << endl;
    cin >> choice;
    if (choice == 1) {
        Matrix<double> A(a, a + 1);
        cout << endl << "Your matrix:" << endl;
        A.PrintMatrix();
        cout << endl;
        for (size_t i = 0; i < a; i++) {
            CompareAndSwap(A, i);
            Magic(A, i, a);
        }
        for (size_t i = 1; i < a; i++)
            Magic2(A, i, a);
        cout << endl << "Matrix after transformations:" << endl;
        A.PrintMatrix();
        GetAnswer(A, a);
    }
    else if (choice == 2) {
        Matrix<double> A(a, a + 1, "ItIsRandomMatrix");
        cout << endl << "Your matrix:" << endl;
        A.PrintMatrix();
        cout << endl;
        for (size_t i = 0; i < a; i++) {
            CompareAndSwap(A, i);
            Magic(A, i, a);
        }
        for (size_t i = 1; i < a; i++)
            Magic2(A, i, a);
        cout << endl << "Matrix after transformations:" << endl;
        A.PrintMatrix();
        GetAnswer(A, a);
    }
    else {
        cout << "Wrong choice..." << endl;
        return 0;
    }
}
```

Онлайн калькулятор: <https://ru.onlimeschool.com/math/assistance/equation/haus/>