

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА



Институт радиоэлектроники и информационных технологий
Кафедра Информатики и систем управления

Лабораторная работа №2 «Разреженные матрицы.»

ОТЧЕТ по лабораторной работе № 2

по дисциплине
Технологии программирования

РУКОВОДИТЕЛЬ:

(подпись)

Капранов С.Н.
(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Куликова Е.А.
(фамилия, и.,о.)

18-ИСТ-4
(шифр группы)

Работа защищена «__» _____

С оценкой _____

Нижний Новгород

2020

Содержание

Введение.....	2
1. Цель работы	3
2. Задачи	3
3. Описание алгоритма	3
4. Код программы.....	4
5. Реализация программы	6
Заключение	8
Используемая литература.....	9

					ЛР2 – НГТУ – 18-ИСТ-4 – 908 – 10		
Изм.	Лист	№ Докум.	Подпись	Дата	Лабораторная работа №2		
Разраб.	Куликова Е.А.						
Проверил	Капранов С.Н.						
Н. контр.							
Утв.							
					Лит.	Лист	Листов
						1	9
					Каф. ИСУ 18-ИСТ-4		

Введение

Разрежённая матрица – это матрица с преимущественно нулевыми элементами. В противном случае, если большая часть элементов матрицы ненулевые, матрица считается плотной.

Среди специалистов нет единства в определении того, какое именно количество ненулевых элементов делает матрицу разрежённой. Разные авторы предлагают различные варианты.

Определитель – в линейной алгебре определитель (или детерминант) – это скалярная величина, которая может быть вычислена и поставлена в однозначное соответствие любой квадратной матрице.

В рамках второй лабораторной работы (вариант 10) необходимо создать программу, которая должна находить определитель на вход пришедшей матрицы.

					ЛР4 – НГТУ – 18-ИСТ-4 – 908 – 10	Лист
						2
Изм.	Лист	№ докум.	Подпись	Дата		

1. Цель работы

Создать программу, соответствующую правилам второй лабораторной работы, на вход приходит разреженная структурно симметричная матрица и необходимо найти её определитель.

2. Задачи

Поставленные задачи:

1. Разработать алгоритм, по которому будет выполняться программа.
2. Написать код, реализующий задание.
3. Протестировать, чтобы убедиться в правильности решения.

3. Описание алгоритма

Вход: Размер матрицы и значения матрицы;

Выход: Представление матрицы в виде CRS и определитель матрицы;

Начало

Цикл по строкам матрицы:

Сохраняем количество ненулевых элементов в строке;

Цикл по столбцам матрицы:

Если элемент находится на диагонали матрицы, то сохраняем элемент в массив диагональных элементов;

Иначе:

Если элемент ненулевой, то сохраняем элемент в массив остальных значений и сохраняем столбец элемента в массив номеров столбцов элементов;

Вывод матрицы в виде CRS структуры;

Рекурсивно вычисляем детерминант матрицы:

Формируем минор матрицы;

Если размер матрицы равен 1, то детерминант равен значению матрицы;

					ЛР4 – НГТУ – 18-ИСТ-4 – 908 – 10	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

Если размер матрицы равен 2, то детерминант равен значению
вычисленному по формуле $mas[0][0] * mas[1][1] - (mas[1][0] * mas[0][1])$;

Если размер матрицы больше 2, то:

Цикл по строкам матрицы:

Получаем минор матрицы без i -ой строки и j -ого столбца;

Увеличиваем значение детерминанта на -1 в степени i умноженное на
текущий элемент столбца умноженный на определитель минора матрицы без
 i -ой строки и j -ого столбца;

Конец.

4. Код программы

Main.cpp

```
#include <iostream>
#include "CRS.h"

int main()
{
    setlocale(LC_ALL, "Russian");
    unsigned int n; // Размер матрицы
    unsigned int nz = 0; // Количество ненулевых элементов
    double ** a; // Матрица вводимая пользователем
    std::cout << "Введите размер матрицы - ";
    std::cin >> n;
    std::cout << "Введите матрицу" << std::endl;
    a = new double*[n];
    for (unsigned i = 0; i < n; i++)
    {
        a[i] = new double[n];
        for (unsigned j = 0; j < n; j++)
        {
            std::cin >> a[i][j];
            if (j <= i && a[i][j] != 0)
                nz++;
        }
    }
    std::cout << std::endl;
    CRS A(a, n, nz); // Создание CRS представления матрицы
    A.Print(); // Печать представления CRS
    std::cout << std::endl << "Определитель - " << A.Determinant(a, n);
    return 0;
}
```

CRS.h

```
#pragma once
#include <iostream>

class CRS
{

```

					ЛР4 – НГТУ – 18-ИСТ-4 – 908 – 10	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

```

double * adiag; // Массив значений по диагонали
double * altr; // Массив остальных значений
unsigned int * jptr; // Массив номеров столбцов элементов altr
unsigned int * row_index; // Массив индексов строк
unsigned int n; // Размер матрицы
unsigned int nz; // Количество ненулевых элементов

public:
CRS(double ** a, unsigned int _n, unsigned int _nz)
{
    nz = _nz;
    n = _n;
    adiag = new double[n];
    altr = new double[nz - n];
    jptr = new unsigned int[nz - n];
    row_index = new unsigned int[n];
    unsigned int cur1 = 0;
    unsigned int cur2 = 0;
    for (unsigned i = 0; i < n; i++)
    {
        row_index[i] = cur2;
        for (unsigned j = 0; j < i + 1; j++)
        {
            if (i == j)
                adiag[cur1++] = a[i][j];
            else
            {
                if (a[i][j] != 0)
                {
                    altr[cur2] = a[i][j];
                    jptr[cur2++] = j;
                }
            }
        }
    }
}

~CRS()
{
    delete[] adiag;
    delete[] altr;
    delete[] jptr;
    delete[] row_index;
}

void Print()
{
    // Представление разреженной симметричной матрицы в виде CRS
    std::cout << "Количество ненулевых элементов - " << nz << std::endl;
    std::cout << "Массив значений по диагонали:" << std::endl;
    for (unsigned i = 0; i < n; i++)
        std::cout << adiag[i] << " ";
    std::cout << std::endl << "Массив остальных значений:" << std::endl;
    for (unsigned i = 0; i < nz - n; i++)
        std::cout << altr[i] << " ";
    std::cout << std::endl << "Массив номеров столбцов недиагональных элементов:" << std::endl;
    for (unsigned i = 0; i < nz - n; i++)
        std::cout << jptr[i] << " ";
    std::cout << std::endl << "Массив индексов строк:" << std::endl;
    for (unsigned i = 0; i < n; i++)
        std::cout << row_index[i] << " ";
}

void GetMatr(double ** mas, double ** p, int i, int j, int m)
{
    // Получение матрицы без i-й строки и j-го столбца
    int ki, kj, di, dj;
    di = 0;
    for (ki = 0; ki < m - 1; ki++)

```

```

        { // проверка индекса строки
            if (ki == i) di = 1;
            dj = 0;
            for (kj = 0; kj < m - 1; kj++)
            { // проверка индекса столбца
                if (kj == j) dj = 1;
                p[ki][kj] = mas[ki + di][kj + dj];
            }
        }
    }
}

double Determinant(double ** mas, int m)
{ // Детерминант
    double ** p;
    p = new double*[m];
    for (int i = 0; i < m; i++)
        p[i] = new double[m];
    int k = 1;
    double d = 0;
    if (m == 1)
        return mas[0][0];
    if (m == 2)
        return mas[0][0] * mas[1][1] - (mas[1][0] * mas[0][1]);
    if (m > 2)
    {
        for (int i = 0; i < m; i++)
        {
            GetMatr(mas, p, i, 0, m);
            d = d + k * mas[i][0] * Determinant(p, m - 1);
            k = -k;
        }
    }
    for (int i = 0; i < m; i++)
        delete[] p[i];
    delete[] p;
    return d;
}
};

```

Листинг 1 – Код программы

5. Реализация программы

Приходящая на вход матрица, используемая как пример для проверки кода, выглядит следующим образом.

```

2 0 0 1 0 0 0
0 3 0 0 0 0 1
0 0 4 0 0 0 0
1 0 0 5 3 0 0
0 0 0 3 7 0 0
0 0 0 0 0 8 2
0 1 0 0 0 2 6

```

Рисунок 1 – Пример разреженной симметричной матрицы

Консольный вид реализации выглядит следующим образом.

```

Введите размер матрицы - 7
Введите матрицу
2 0 0 1 0 0 0
0 3 0 0 0 0 1
0 0 4 0 0 0 0
1 0 0 5 3 0 0
0 0 0 3 7 0 0
0 0 0 0 0 8 2
0 1 0 0 0 2 6

Количество ненулевых элементов - 11
Массив значений по диагонали:
2 3 4 5 7 8 6
Массив остальных значений:
1 3 1 2
Массив номеров столбцов недиагональных элементов:
0 3 1 5
Массив индексов строк:
0 0 0 0 1 2 2
Определитель - 22320

```

Рисунок 2 – Результат работы программы

Заключение

В ходе второй лабораторной работы была создана программа, соответствующая правилам лабораторной работы, на вход приходит разреженная структурно симметричная матрица и происходят вычисления её определителя. Также программа была протестирована необходимое количество раз для проверки на корректность.

					ЛР4 – НГТУ – 18-ИСТ-4 – 908 – 10	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

Используемая литература

1. Определитель – <https://ru.wikipedia.org/wiki/Определитель>
2. Разреженная матрица – https://ru.wikipedia.org/wiki/Разреженная_матрица
3. Симметричная матрица – https://ru.wikipedia.org/wiki/Симметричная_матрица

					ЛР4 – НГТУ – 18-ИСТ-4 – 908 – 10	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		