Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Методы численного анализа»

**ОТЧЕТ**

к лабораторной работе №5

на тему:

**«Вычисление собственных значений и векторов»**

БГУИР 1-40 04 01

|  |
| --- |
| Выполнил студент группы 253505  Форинов Егор Вячеславович |
|  |
| (дата, подпись студента) |
| Проверил доцент кафедры информатики  АНИСИМОВ Владимир Яковлевич |
|  |
| (дата, подпись преподавателя) |

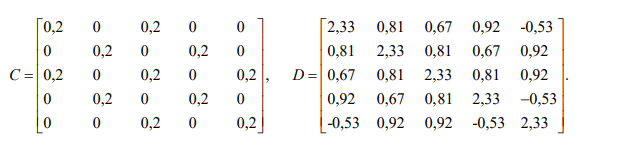
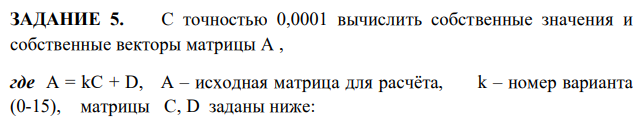
Минск 2023

**Содержание**

1. Цель работы
2. Задание
3. Программная реализация
4. Полученные результаты
5. Оценка полученных результатов
6. Вывод

**Цель работы**

* изучить метод Якоби (вращений) для поиска собственных значений и векторов матрицы;
* составить программу поиска собственных значений и векторов матрицы указанными методами, применимую для организации вычислений на ЭВМ;
* выполнить тестовые примеры и проверить правильность работы программы

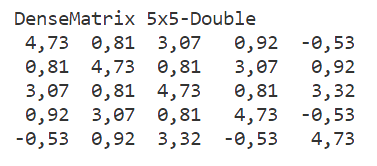


**Вариант 12**

**Программная реализация**

*Исходные данные*

Матрица, полученная в результате подстановки A = 12 \* C + D:



Код метода Якоби(вращений):

Tuple<Vector<double>, Matrix<double>> Jacobi(Matrix<double> matrix)

{

    if (!matrix.IsSymmetric())

        throw new ArgumentException("Source matrix is not symmetric");

    var n = matrix.RowCount;

    const double maxIterations = 1000;

    var eigenvectors = Matrix<double>.Build.DenseDiagonal(n, n, 1);

    for (var iteration = 0; iteration < maxIterations; iteration++)

    {

        double maxOfDiagonal = 0.0;

        int p = 0, q = 0;

        for (var i = 0; i < n - 1; i++)

        {

            for (var j = i + 1; j < n; j++)

            {

                if (Math.Abs(matrix[i, j]) > maxOfDiagonal)

                {

                    maxOfDiagonal = Math.Abs(matrix[i, j]);

                    p = i;

                    q = j;

                }

            }

        }

        if (maxOfDiagonal < 1e-8)

            break;

        double angle = 0.5 \* Math.Atan(2 \* matrix[p, q] / (matrix[p, p] - matrix[q, q]));

        double cos = Math.Cos(angle);

        double sin = Math.Sin(angle);

        var rotationMatrix = Matrix<double>.Build.Dense(n, n);

        for (var i  = 0; i < n; i++)

        {

            for (var j = 0; j < n; j++)

            {

                if (i == j && i != p && i != q)

                    rotationMatrix[i, j] = 1.0;

                else if (i == p && j == p)

                    rotationMatrix[i, j] = cos;

                else if (i == p && j == q)

                    rotationMatrix[i, j] = -sin;

                else if (i == q && j == p)

                    rotationMatrix[i, j] = sin;

                else if (i == q && j == q)

                    rotationMatrix[i, j] = cos;

                else

                    rotationMatrix[i, j] = 0.0;

            }

        }

        matrix = rotationMatrix.Transpose().Multiply(matrix).Multiply(rotationMatrix);

        eigenvectors = eigenvectors.Multiply(rotationMatrix);

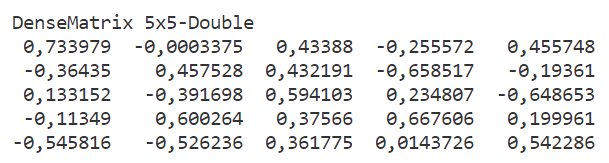
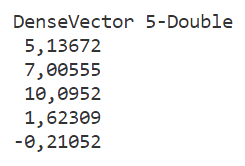
    }

    var eigenValues = matrix.Diagonal();

    return new (eigenValues, eigenvectors);

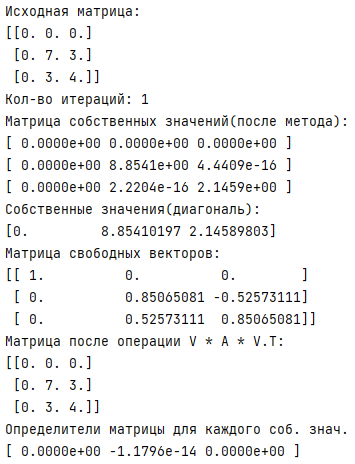
}

**Полученные результаты**

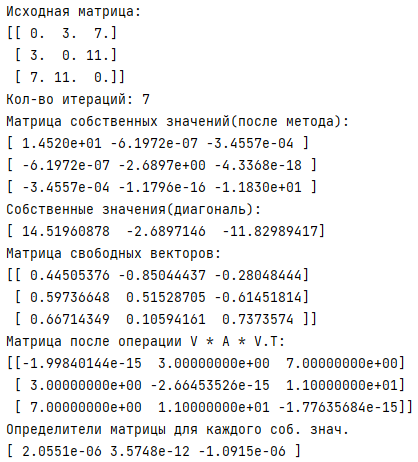


По значениям определителей матрицы для каждого собственного значения можно сделать вывод, что точность 0,0001 была достигнута.

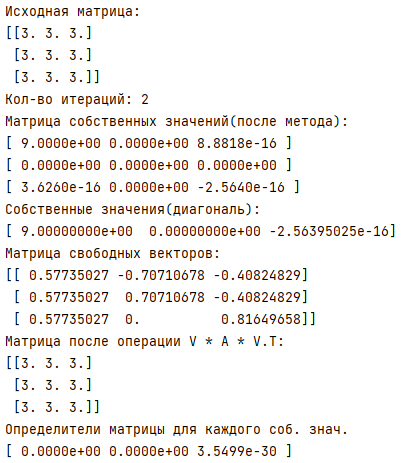
*Тестовый пример 1. Нулевые строка и столбец*



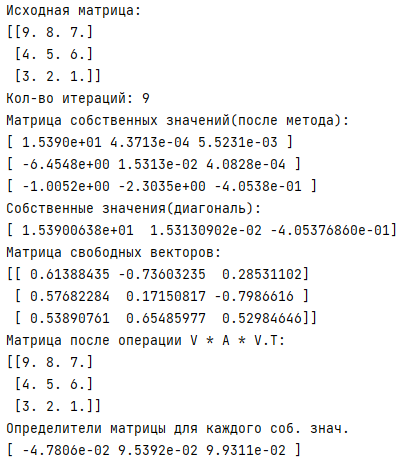
*Тестовый пример 2. Нулевая диагональ*



*Тестовый пример 3. Нулевая матрица*



*Тестовый пример 3. Несимметричная матрица*



**Вывод**

В ходе выполнения лабораторной работы я изучил метод Якоби(вращений), написал программу его реализации на языке C#, правильность работы программы проверил на тестовых примерах.

На основании тестов можно сделать следующие выводы:

* Программа позволяет получить решения системы с заданной точностью (заданная точность в условиях лабораторной работы 10^-4);
* Основное достоинство метода Якоби заключается в том, что при выполнении каждого плоского вращения уменьшается сумма квадратов недиагональных элементов; сходимость этой суммы к нулю по мере увеличения числа шагов гарантирует сходимость процесса диагонализации.