

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Поволжский государственный университет телекоммуникаций и
информатики»

Факультет информатики и вычислительной техники

Кафедра При

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №6
Дисциплина: Численные методы

Выполнил: студент
При-21 Морзюков
М.А.
Проверил(а):
Осанов В.А.

Самара 2024

Вариант №11

Цель работы: изучить решение проблемы собственных значений и собственных векторов: QR-алгоритм.

11	.70954E - 03	-.35012E + 01	.23236E + 02	-.16032E + 00
	-.99360E - 04	.22264E + 01	.14775E + 02	.22450E - 01
	.37446E - 03	.25177E + 01	.16709E + 02	-.84609E - 01
	-.35194E - 04	-.78859E + 00	-.52335E + 01	.79521E - 02
0.070954	-0.00009936	0.0037446	-0.000035194	
-35.012	22.264	25.177	-0.78859	
232.36	147.75	167.09	-52.335	
-0.016032	-0.002245	-0.084609	0.0079521	

Метод QR-разложения — это метод, который преобразует матрицу A в произведение ортогональной матрицы Q и верхней треугольной матрицы R . Этот метод сходится для многих матриц и является надёжным способом нахождения собственных значений, особенно для симметричных матриц.

```
public class QRDecomposition {
    private static final double EPSILON = 0.0001;

    public void findEigenvaluesAndEigenvectors(Matrix matrix) {
        int n = matrix.getRows();
        Matrix A = matrix;
        Matrix QTotal = Matrix.identity(n);

        for (int iteration = 0; iteration < 100; iteration++) {
            QRResult qrResult = decomposeQR(A);
            Matrix Q = qrResult.getQ();
            Matrix R = qrResult.getR();

            A = R.multiply(Q);
            QTotal = QTotal.multiply(Q);

            if (hasConverged(A)) {
                System.out.println("Собственные значения:");
                for (int i = 0; i < n; i++) {
                    System.out.printf("%d = %.5f\n", i + 1, A.getData()[i][i]);
                }
                System.out.println("=====");

                System.out.println("Матрица Q");
                Q.printMatrix();
                System.out.println("=====");

                System.out.println("Матрица R");
                R.printMatrix();
                System.out.println("=====");

                System.out.println("Собственные значения:");
                QTotal.printMatrix();
                System.out.println("=====");

                System.out.println("Собственные векторы:");
            }
        }
    }
}
```

```

        QTotal.printMatrix();
        return;
    }

    System.out.println("Не удалось найти собственные значения за заданное количество итераций.");
}

private QRResult decomposeQR(Matrix A) {
    int n = A.getRows();
    double[][] qData = new double[n][n];
    double[][] rData = new double[n][n];

    for (int j = 0; j < n; j++) {
        double[] v = new double[n];
        for (int i = 0; i < n; i++) {
            v[i] = A.getData()[i][j];
        }

        for (int i = 0; i < j; i++) {
            double r_ij = 0;
            for (int k = 0; k < n; k++) {
                r_ij += qData[k][i] * v[k];
            }
            rData[i][j] = r_ij;
            for (int k = 0; k < n; k++) {
                v[k] -= r_ij * qData[k][i];
            }
        }

        double r_jj = 0;
        for (int k = 0; k < n; k++) {
            r_jj += v[k] * v[k];
        }
        r_jj = Math.sqrt(r_jj);
        rData[j][j] = r_jj;
    }

```

```

        for (int k = 0; k < n; k++) {
            qData[k][j] = v[k] / r_jj;
        }
    }

    Matrix Q = new Matrix(qData);
    Matrix R = new Matrix(rData);

    return new QRResult(Q, R);
}

private boolean hasConverged(Matrix A) {
    for (int i = 0; i < A.getRows() - 1; i++) {
        if (Math.abs(A.getData()[i + 1][i]) > EPSILON) {
            return false;
        }
    }
    return true;
}

private static class QRResult {
    private Matrix Q;
    private Matrix R;

    public QRResult(Matrix Q, Matrix R) {
        this.Q = Q;
        this.R = R;
    }

    public Matrix getQ() {
        return Q;
    }

    public Matrix getR() {
        return R;
    }
}

```

Результат выполнения программы:

```

Решение методом QR-разложения:
Матрица Q
-0.00030    0.00098    0.04411    0.99908
0.14898    -0.98891    0.00032    0.00097
-0.98875    -0.14902    -0.00003    -0.00015
0.00007    -0.00027    -0.99911    0.04413

=====

Матрица R
-233.00296 -142.77956 -161.45353 51.64310
0.00000    -44.03012 -49.79213 8.58211
0.00000    0.00000    0.08751 -0.00627
0.00000    0.00000    0.00000    0.00735

=====

Собственные значения:
λ1 = 189.786323232
λ2 = -0.8209902153
λ3 = 0.8155907458
λ4 = 0.0507859234

=====

Собственные векторы:
-0.000019 -0.148982 -0.988842 0.000437
-0.002868 -0.736016 0.673338 0.069811
-0.003302 0.758492 -0.648012 0.068361
0.073881 0.564157 -0.372076 0.733357

```

Проверка:

