

Лабораторная работа  
по вычислительным методам алгебры на тему:

Определение наибольшего по модулю собственного значения  
матрицы

Выполнил:  
Архангельский И.А.

Проверил:  
Кондратюк А.П.

# Входные и выходные данные.

## Входные данные

Входной файл в первой строке содержит число  $n$  - размерность матрицы, следующие  $n$  строк содержат матрицу  $(A)$ , где  $A$  - квадратная матрица.

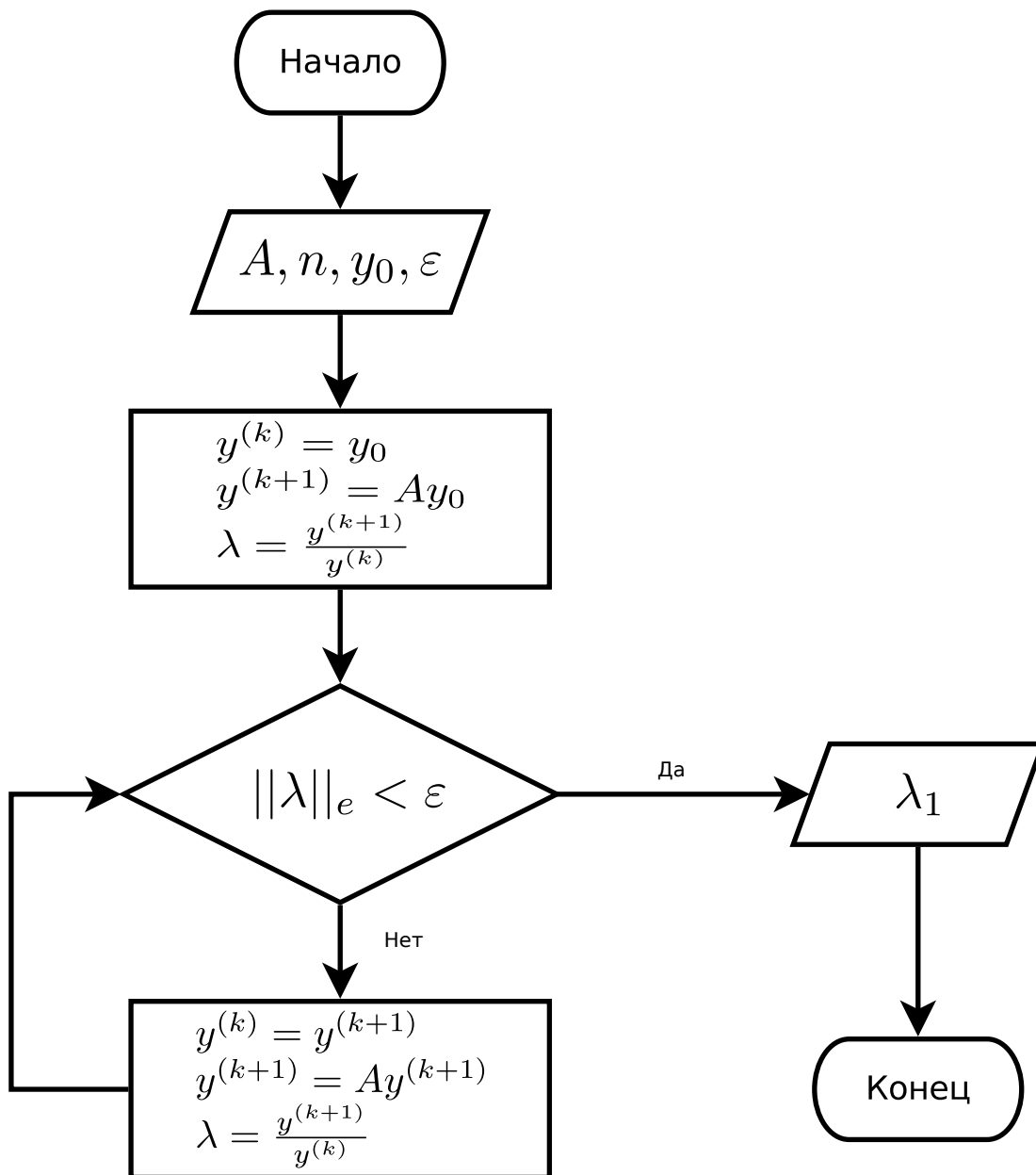
Строка  $n + 2$  содержит начальный вектор  $y_0$ .

Строка  $n + 3$  содержит величину  $\varepsilon$  - заданную точность.

## Выходные данные

На выход в stdout подается максимальное собственное значение матрицы  $A$

# Блок-схема



# Реализация

```
1 package maximeigenvalue;
2 import java.io.BufferedReader;
3 import java.io.FileReader;
4 import java.io.IOException;
5 import java.util.StringTokenizer;
6 public class MaximumEgien
7 {
8     double [][] matrix;
9     double [] y;
10    double epsilon;
11    int k;
12    public MaximumEgien(String filename) throws IOException
13    {
14        BufferedReader br = new BufferedReader(new FileReader(filename));
15        int dim = Integer.parseInt(br.readLine());
16        matrix = new double[dim][dim];
17        for (int i = 0; i < dim; i++)
18        {
19            StringTokenizer tmp = new StringTokenizer(br.readLine());
20            for (int j = 0; j < dim; j++)
21            {
22                matrix[i][j] = Double.parseDouble(tmp.nextToken());
23            }
24        }
25        StringTokenizer tmp = new StringTokenizer(br.readLine());
26        y = new double[dim];
27        for (int i = 0; i < dim; i++)
28        {
29            y[i] = Double.parseDouble(tmp.nextToken());
30        }
31        epsilon = Double.parseDouble(br.readLine());
32    }
33    public double getMaxEigen()
34    {
35        double [] ythis = mulMatrix(y);
36        double [] yprev = y;
37        double [] lambdas = new double[yprev.length];
38
39        for (int i = 0; i < lambdas.length; i++) {lambdas[i] = ythis[i] / yprev[i];}
40        while (eps(lambdas) > epsilon || k < 10)
41        {
42            yprev = ythis;
43            ythis = mulMatrix(ythis);
44            for (int i = 0; i < lambdas.length; i++) {lambdas[i] = ythis[i] / yprev[i];}
45            k++;
46        }
47        return lambdas[0];
48    }
49    private double [] mulMatrix(double [] x)
50    {
51        double [] res = new double[matrix.length];
52        for (int i = 0; i < matrix.length; i++)
53        {
54            res[i] = 0;
55            for (int j = 0; j < matrix.length; j++) {res[i] += x[j] * matrix[i][j];}
56        }
57        return res;
58    }
59    public int getK() {return k;}
60
61    private double eps(double [] x)
62    {
63        double e = 0;
64        for (int i = 1; i < x.length; i++) {e += Math.pow(Math.abs(x[0] - x[i]), 2);}
65        return Math.sqrt(e);
66    }
67 }
```

# Тестовые данные

test01.in				test01.out			
3				10.673392446138907			
1	2	3					
3	2	3					
3	5	7					
1	1	1					
0.0000001							

test02.in				test02.out			
3				90.6120501191381			
1	2	3					
3	2	4					
7	8	90					
1	2	3					
0.0000003							

test03.in				test03.out			
4				96.61095118459914			
1	2	3	10				
3	2	4	100				
7	8	90	31				
34	32	4	15				
1	2	3	5				
0.0000003							