Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф. Уткина»

Факультет вычислительной техники

Кафедра вычислительной и прикладной математики

Отчет по лабораторной работе №3

Решение систем линейных алгебраических уравнений

по дисциплине

«Вычислительная математика»

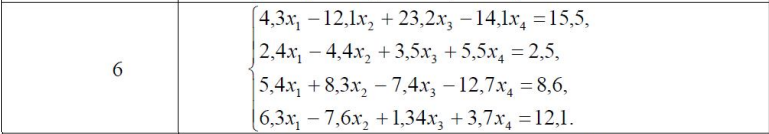
Выполнил: студент группы 044

Лаптев Никита Александрович

Проверил: доцент

Проказникова Елена Николаевна

Рязань 2022

**Задание:** Для заданного варианта решить методом итераций систему уравнений A ⋅ X = B . Для остановки процесса последовательных приближений использовать условие: сумма модулей приращений элементов вектора X на последнем шаге итераций меньше ε = 0.001. Используя метод Гаусса, вычислить определитель и число обусловленности матрицы A.

**Блок – схемы:**

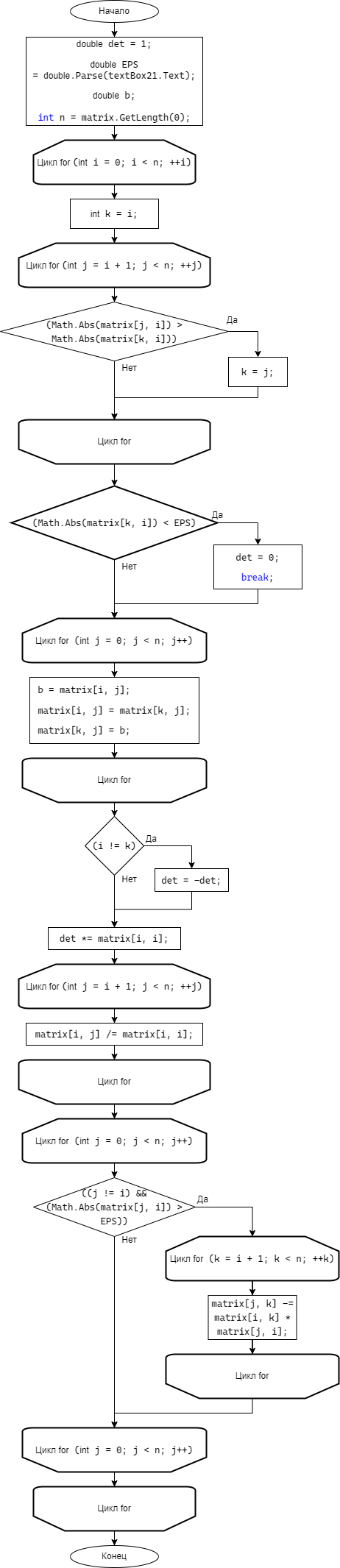


Рис. 1 – Нахождение определителя

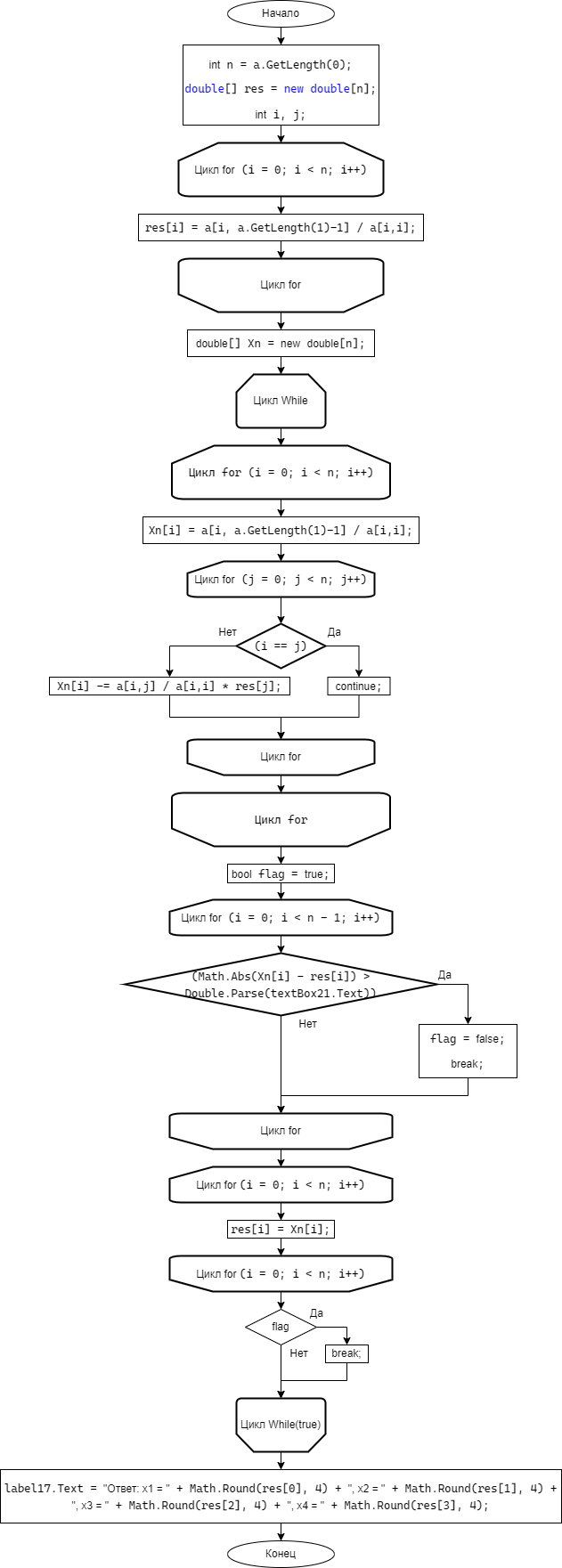


Рис. 2 – Метод итераций

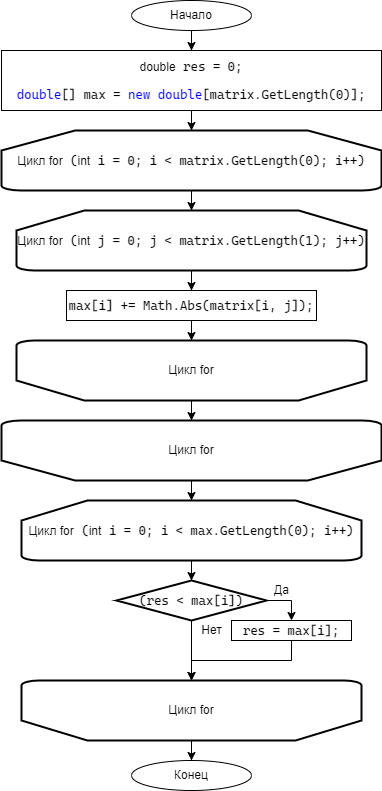


Рис. 3 – Нахождение нормы матрицы

**Код программы:**

private double findDet(double[,] matrix)

{

double det = 1;

double EPS = double.Parse(textBox21.Text);

double b;

int n = matrix.GetLength(0);

for (int i = 0; i < n; ++i)

{

int k = i;

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

{

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

{

k = j;

}

}

if (Math.Abs(matrix[k, i]) < EPS)

{

det = 0;

break;

}

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

{

b = matrix[i, j];

matrix[i, j] = matrix[k, j];

matrix[k, j] = b;

}

if (i != k)

{

det = -det;

}

det \*= matrix[i, i];

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

{

matrix[i, j] /= matrix[i, i];

}

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

{

if ((j != i) && (Math.Abs(matrix[j, i]) > EPS))

{

for (k = i + 1; k < n; ++k)

{

}

}

}

}

return det;

}

private double[] iter(double[,] a)

{

int n = a.GetLength(0);

double[] res = new double[n];

int i, j;

for (i = 0; i < n; i++)

{

res[i] = a[i, a.GetLength(1)-1] / a[i,i];

}

double[] Xn = new double[n];

do

{

for (i = 0; i < n; i++)

{

Xn[i] = a[i, a.GetLength(1)-1] / a[i,i];

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

{

if (i == j)

continue;

else

{

Xn[i] -= a[i,j] / a[i,i] \* res[j];

}

}

}

bool flag = true;

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

{

if (Math.Abs(Xn[i] - res[i]) > Double.Parse(textBox21.Text))

{

flag = false;

break;

}

}

for (i = 0; i < n; i++)

{

res[i] = Xn[i];

}

if (flag)

break;

} while (true);

return res;

}

private double findNormMatrix(double[,] matrix)

{

double res = 0;

double[] max = new double[matrix.GetLength(0)];

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

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

}

}

for (int i = 0; i < max.GetLength(0); i++)

{

if (res < max[i])

{

}

}

return res;

}