namespace HelloWorld

{

class Program

{

static void SolveSystem(double[,] coefs)

{

int n = coefs.GetLength(0);

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

{

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

{

double ratio = coefs[j, i] / coefs[i, i];

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

{

coefs[j, k] -= ratio \* coefs[i, k];

}

}

}

double[] solutions = new double[n];

for (int i = n - 1; i >= 0; i--)

{

double sum = 0.0;

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

{

sum += coefs[i, j] \* solutions[j];

}

solutions[i] = (coefs[i, n] - sum) / coefs[i, i];

}

Console.WriteLine("Решение:");

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

{

Console.WriteLine($"x{i + 1} = {solutions[i]}");

}

}

static void Main(string[] args)

{

double[,] coefs =

{

{3.2, -11.5, 3.8, 2.8},

{0.8, 1.3, 6.4, -6.5},

{2.4, 7.2, -1.2, 4.5 }

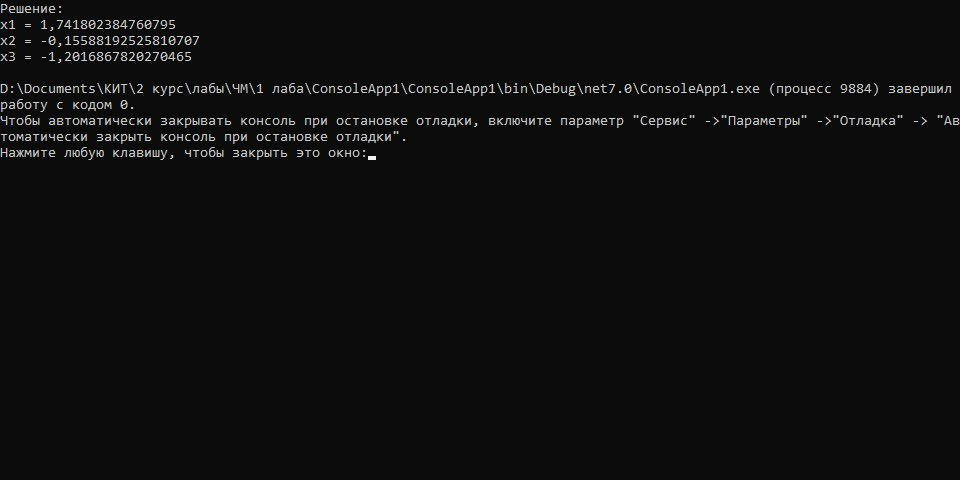
};

SolveSystem(coefs);

}

}

}



namespace HelloWorld

{

class Program

{

static double MaxArray(double[] array)

{

return array.Max();

}

static void SolveSystem(double[,] A, double[] vectorB, double[] initialGuess, double epsilon)

{

int n = A.GetLength(0);

double[] x = new double[n];

double[] previousX = new double[n];

double[] error = new double[n];

Array.Copy(initialGuess, x, n);

int iteration = 0;

do

{

iteration++;

Console.Write($"Iteration {iteration}: ");

Array.Copy(x, previousX, n);

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

{

double sigma = 0;

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

{

if (j != i)

{

sigma += A[i, j] \* previousX[j];

}

}

x[i] = (vectorB[i] - sigma) / A[i, i];

Console.WriteLine(x[i]);

error[i] = Math.Abs(x[i] - previousX[i]);

}

} while (MaxArray(error) > epsilon);

}

static void Main(string[] args)

{

double[,] matrixA = {

{1, -1, 1},

{2, 3, -2},

{3, -2, 5 } };

double[] vectorB = {4, -5, 8};

double[] initialGuess = { 0, 0, 0 };

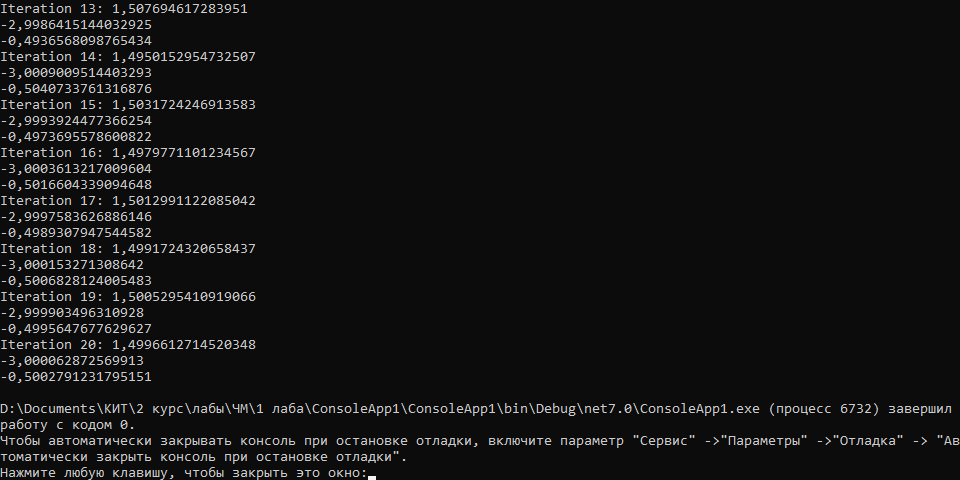
double epsilon = 0.001;

SolveSystem(matrixA, vectorB, initialGuess, epsilon);

}

}

}



using System.ComponentModel.DataAnnotations;

namespace HelloWorld

{

class Program

{

static void PrintVector(double[] vector)

{

for (int i = 0; i < vector.Length; i++)

{

Console.WriteLine(vector[i]);

}

}

static double CalculateError(double[] x, double[] prevX)

{

double max = 0;

for (int i = 0; i < 3; i++)

{

if (max == 0)

{

max = Math.Abs(prevX[i] - x[i]);

} else if (max < Math.Abs(prevX[i] - x[i]))

{

max = Math.Abs(prevX[i] - x[i]);

}

}

return max;

}

static void SolveSeidel(double[,] coefficients, double[] constants, int maxIterations, double tolerance)

{

int n = constants.Length;

double[] x = new double[n];

double[] prevX = new double[n];

int iteration = 0;

double error = 0;

do

{

Array.Copy(x, prevX, n);

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

{

double sigma = 0;

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

{

if (j != 1)

{

sigma += coefficients[i, j] \* x[j];

}

}

x[i] = (constants[i] - sigma) / coefficients[i, i];

}

iteration++;

error = CalculateError(x, prevX);

Console.Write("Iteration {0}: ", iteration);

PrintVector(x);

Console.WriteLine("Error {0}", error);

} while (iteration < maxIterations && error > tolerance);

}

static void Main(string[] args)

{

double[,] coefficients =

{

{5, -1, 3 },

{1, -4, 2 },

{2, -1, 5 }

};

double[] constants = { 5, 20, 10 };

int maxIterations = 100;

double tolerance = 1e-6;

SolveSeidel(coefficients, constants, maxIterations, tolerance);

Console.ReadLine();

}

}

}

