СПб НИУ ИТМО

кафедра ИПМ

Вычислительная математика

Лабораторная работа №1

Решение систем линейных алгебраических уравнений

методом Гаусса с выбором главного элемента

Работу выполнил:

Студент II курса

Группы № 2120

Журавлев Виталий

Преподаватель:

Шипилов П.А.

Санкт-Петербург

2013 г.

**Задача:**

Написать программу, выполняющую решение заданной системы линейных уравнений методом Гаусса с выбором главного элемента. Программа должна содержать соответствующий класс, реализующий работу с расширенной матрицей коэффициентов; конструкторы для возможности по-разному задавать исходные данные; методы, содержащие алгоритмы решения системы уравнений и методы вывода данных на консоль.

**Описание:**

*Метод Гаусса с выбором главного элемента:*

Среди элементов матрицы image190 выберем элемент, наибольший по модулю, называемый *главным элементом*.

Например, пусть им будет элемент image191. Строка с номером *m*, содержащая главный элемент, называется *главной строкой*.

Далее вычисляем множители image192 для всех image193 .

Затем матрица image194 преобразуется так: к каждой image132-й, неглавной строке, прибавим почленно главную строку, умножив её на image195. В результате получим матрицу, у которой все элементы image196-го столбца, за исключением image191, равны 0. Отбрасывая этот столбец и главную строку, получаем новую матрицу image197 с меньшим на единицу числом строк и столбцов.

Над матрицей image197 повторяем те же операции, после чего получаем матрицу image198 и т.д. Эти преобразования продолжаются до тех пор, пока не получится матрица, содержащая одну строку из двух элементов, которая тоже считается главной. Затем объединяем все главные строки, начиная с последней. После некоторой перестановки они образуют треугольную матрицу, эквивалентную исходной. На этом заканчивается прямой ход метода Гаусса с выбором главного элемента.

Далее находим image199, решая систему с треугольной матрицей. Это обратный ход.

Далее находится степень отклонения полученного решения от точного - *невязка*, равная разности между правой и левой частями уравнений при подстановке в них решения.

**Алгоритм метода:**

****

**Схема программы:**

Программа содержит 2 класса: GaussMethod, содержащий 4 конструктора для возможности различного ввода данных и 4 метода для работы с ними, а так же класс Program, необходимый для работы с экземплярами класса Gauss.

**Код программы:**

class GaussMethod

{

int NumEq; // кол-во уравнений

double[,] Matrix; // матрица коэфициентов

double[,] WorkMatrix; //рабочая копия матрицы коэффициентов

public GaussMethod(int num) //случайные значения

{

this.NumEq = num;

Matrix = new double[NumEq, NumEq + 1];

Random a = new Random();

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

{

for (int j = 0; j < NumEq + 1; j++)

{

this.Matrix[i, j] = a.Next(-50, 50);

}

}

}

public GaussMethod(int kol, int id) // ввод с клавиатуры

{

this.NumEq = kol;

if (kol < 1 || kol > 11) throw new ArgumentException();

this.Matrix = new double[NumEq, NumEq + 1];

Console.WriteLine("Введите коэффициенты уравнения:");

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

{

for (int j = 0; j < NumEq + 1; j++)

{

Console.Write("K({0},{1}) = ", i + 1, j + 1);

this.Matrix[i, j] = Convert.ToDouble(Console.ReadLine());

}

}

}

public GaussMethod(string filename) // Ввод из файла

{

using (StreamReader Reader = new StreamReader(filename))

{

string numb = Reader.ReadLine();

this.NumEq = int.Parse(numb);

Matrix = new double[NumEq, NumEq + 1];

string s = Reader.ReadToEnd();

char[] separators = new char[] { ' ', '\n', '\t', '\r' };

string[] mas = s.Split(separators, StringSplitOptions.RemoveEmptyEntries);

int y = 0;

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

{

for (int j = 0; j < NumEq + 1; j++)

{

if (mas[y] != "") this.Matrix[i, j] = double.Parse(mas[y]);

y++;

}

}

}

}

public static void ShowEquation(GaussMethod gauss) // вывод уравнений на консоль

{

for (int i = 0; i < gauss.NumEq; i++)

{

for (int j = 0; j < gauss.NumEq; j++)

{

Console.Write("({0:F1})\*X{1} ", gauss.Matrix[i, j], j + 1);

if (j != gauss.NumEq - 1) Console.Write("+ ");

}

Console.Write("= {0:F1}\n", gauss.Matrix[i, gauss.NumEq]);

}

Console.WriteLine();

}

public static double[] Solution(GaussMethod gauss) // нахождение корней

{

gauss.WorkMatrix = new double[gauss.NumEq, gauss.NumEq + 1];

for (int i = 0; i < gauss.NumEq; i++)

{

for (int j = 0; j <= gauss.NumEq; j++)

{

gauss.WorkMatrix[i, j] = gauss.Matrix[i, j];

}

}

double[] solution = new double[gauss.NumEq]; // матрица решений системы уравнений

double determinant = GaussMethod.Determinant(gauss); // нахождение определителя

Console.WriteLine("Определитель системы равен: {0:F2}", determinant);

if (determinant != 0)

{

for (int i = gauss.NumEq - 1; i >= 0; i--)

{

double summ = 0;

for (int j = gauss.NumEq - 1; j > i; j--)

{

summ += solution[j] \* gauss.WorkMatrix[i, j];

}

solution[i] = (gauss.WorkMatrix[i, gauss.NumEq] - summ) / gauss.WorkMatrix[i, i];

}

for (int i = 0; i < gauss.NumEq; i++)

{

Console.WriteLine("X{0} = {1:F2}", i + 1, solution[i]);

}

}

else Console.WriteLine("Система не имеет единственного решения!");

GaussMethod.Nevyazka(gauss, solution);

return solution;

}

public static double Determinant(GaussMethod gauss) // нахождение определителя по методу Гаусса

{

double opred = 1;

double per1 = 0, per2 = 0;

int MaxInd = 0;

for (int j = 0; j < gauss.NumEq - 1; j++)

{

per2 = gauss.WorkMatrix[j, j];

MaxInd = j;

for (int i = j + 1; i < gauss.NumEq; i++)

{

per1 = gauss.WorkMatrix[i, j]; // присвоение переменной 1 значения i-j элемента массива коэффициентов

if (Math.Abs(per1) > Math.Abs(per2))

{

MaxInd = i;

}

}

if (MaxInd != j)

{

GaussMethod.Change(gauss, j, MaxInd); // когда в массиве находится ненулевой элемент, стоящий ниже нужной строки, меняем их с помощью метода Change

opred \*= -1; // при замене строк знак определителя меняется на противоположный

}

per2 = gauss.WorkMatrix[j, j];

if (per2 != 0)

{

for (int x1 = j + 1; x1 < gauss.NumEq; x1++) // обнуление элементов матрицы для сведения к треугольному виду

{

if (gauss.WorkMatrix[x1, j] != 0)

{

per1 = gauss.WorkMatrix[x1, j];

for (int x2 = j; x2 < gauss.NumEq + 1; x2++)

{

gauss.WorkMatrix[x1, x2] = gauss.WorkMatrix[j, x2] \* per1 / per2 - gauss.WorkMatrix[x1, x2];

}

}

}

}

else

{

opred = 0; break;

}

}

for (int i = 0; i < gauss.NumEq; i++) // Вычисление самого определителя

{

opred \*= gauss.WorkMatrix[i, i];

}

return opred;

}

private static void Change(GaussMethod gauss, int StrNum1, int StrNum2) // обмен местами двух строк

{

double[] per = new double[gauss.NumEq + 1]; // Строка, исп. при перемещении строк матрицы коэф.

for (int j = 0; j < gauss.NumEq + 1; j++)

{

per[j] = gauss.WorkMatrix[StrNum1, j];

gauss.WorkMatrix[StrNum1, j] = gauss.WorkMatrix[StrNum2, j];

gauss.WorkMatrix[StrNum2, j] = per[j];

}

}

public static void Nevyazka(GaussMethod gauss, double[] solution) //определение невязки

{

double nevyazka;

double[] podstanovka = new double[gauss.NumEq];

for (int i = 0; i < gauss.NumEq; i++)

{

for (int j = 0; j < gauss.NumEq; j++)

{

podstanovka[i] += gauss.Matrix[i, j] \* solution[j];

}

}

nevyazka = Math.Abs(podstanovka[0] - gauss.Matrix[0, gauss.NumEq]) \* Math.Pow(10, 15);

Console.WriteLine("\nНевязка равна: {0} \* 10^-15", nevyazka);

}

}

class Program

{

static void Main(string[] args)

{

GaussMethod Gauss;

Console.Write("\nВыберите способ ввода иходной матрицы:\n\n\t1 - Ввод случайным образом\n\t2 - Ввод с клавиатуры\n\t3 - Ввод из файла\n\tДругой символ - выход\n\nСимвол: ");

string input = Console.ReadLine();

switch (input)

{

case "1":

{

Console.Write("Введите количество неизвестных (по умолчанию 4): ");

int Num;

string ans = Console.ReadLine();

if (!Int32.TryParse(ans, out Num)) Num = 4;

Gauss = new GaussMethod(Num); break;

}

case "2":

{

Console.Write("Введите количество неизвестных (по умолчанию 4): ");

int NumVar;

string s = Console.ReadLine();

if (!Int32.TryParse(s, out NumVar)) NumVar = 4;

Gauss = new GaussMethod(NumVar, 1);

break;

}

case "3":

{

Console.WriteLine(@"Папка: E:\C#\Gauss\Gauss\bin\Debug");

Console.Write("Введите имя файла: ");

string name = Console.ReadLine() + ".txt";

Gauss = new GaussMethod(name);

break;

}

default: Gauss = new GaussMethod(4); Environment.Exit(0); break;

}

GaussMethod.ShowEquation(Gauss);

GaussMethod.Solution(Gauss);

Console.ReadLine();

}

}

**Результаты и выводы:**

Для проверки работоспособности кода были проверены следующих наборов коэффициентов:

* определитель матрицы равняется нулю
* на выходе должны получиться целые числа
* на выходе должны получиться дробные числа
* работа программы со случайными значениями
* различные виды исключений.

Метод Гаусса с выбором главного члена – наиболее мощный и универсальный инструмент для нахождения решения любой системы линейных уравнений. Этот метод позволяет весьма точно находить корни системы уравнений.