РЕШЕНИЕ СИСТЕМЫ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ МЕТОДОМ КРАМЕРА

**Содержание**

Введение

1. Создание С#

2. Постановка задачи

3. Метод Крамера

4. Программная реализации алгоритма метода Крамера

Заключение

Список использованных источников

**Введение**

На практике в большинстве случаев найти точной решение возникшей математической задачи не удается. Это происходит главным образом не потому, что мы не умеем этого сделать, а поскольку искомое решение обычно не выражается в привычных для нас элементарных или других известных функциях. Поэтому большое значение приобрели численные методы, особенно в связи с возрастанием роли математических методов в различных областях науки и техники и с появлением высокопроизводительных ЭВМ.

В настоящей курсовой работе рассмотрена важная, с точки зрения прикладных задач: метод Крамера для решение линейных алгебраических уравнений.

**1. СОЗДЕНИЕ С #**

Зачастую слишком многого требований от инструментов, с которыми работаем, особенно, когда это касается языков программирования. Хотя таких языков существует великое множество, но только некоторые из них по-настоящему сильны. Эффективность языка заключается в его мощности и одновременно — в гибкости. Синтаксис языка должен быть лаконичным, но ясным. Он должен способствовать созданию корректного кода и предоставлять реальные возможности, а не ультрамодные (и, как правило, тупиковые) решения. Наконец, мощный язык должен иметь одно нематериальное качество: вызывать ощущение гармонии. Как раз таким языком программирования и является С#. Созданный компанией Microsoft для поддержки среды .NET Framework, язык С# опирается на богатое наследие в области программирования. Его главным архитектором был ведущий специалист в этой области — Андерс Хейлсберг (Anders Hejlsberg).

С# -— прямой потомок двух самых успешных в мире компьютерных языков: С и C++. От С он унаследовал синтаксис, ключевые слова и операторы. Он позволяет построить и усовершенствовать объектную модель, определенную в C++. Кроме того, С# близко связан с другим очень успешным языком: Java. Имея общее происхождение, но различаясь во многих важных аспектах, С# и Java — это скорее "двоюродные братья". Например, они оба поддерживают программирование распределенных систем и оба используют промежуточный код для достижения переносимости, но различаются при этом в деталях реализации. Опираясь на мощный фундамент, который составляют унаследованные характеристики, С# содержит ряд важных новшеств, поднимающих искусство программирования на новую ступень. Например, в состав элементов языка С# включены такие понятия, как делегаты (представители), свойства, индексаторы и события. Добавлен также синтаксис, который поддерживает атрибуты; упрощено создание компонентов за счет исключения проблем, связанных с COM (Component Object Model — модель компонентных объектов Microsoft — стандартный механизм, включающий интерфейсы, с помощью которых объекты предоставляют свои службы другим объектам).

И еще. Подобно Java язык С# предлагает средства динамического обнаружения ошибок, обеспечения безопасности и управляемого выполнения программ. Но, в отличие от Java, C# дает программистам доступ к указателям. Таким образом, С# сочетает первозданную мощь C++ с типовой безопасностью Java, которая обеспечивается наличием механизма контроля типов (type checking) и корректным использованием шаблонных классов (template class). Более того, язык С# отличается тем, что компромисс между мощью и надежностью тщательно сбалансирован и практически прозрачен (не заметен для пользователя или программы).

На протяжении всей истории развития вычислительной техники эволюция языков программирования означала изменение вычислительной среды, способа мышления программистов и самого подхода к программированию. Язык С# не является исключением. В непрекращающемся процессе усовершенствования, адаптации и внедрения нововведений С# в настоящее время находится на переднем крае. Это — язык, игнорировать существование которого не может ни один профессиональный программист.

**2. Постановка задачи**

К решению систем линейных уравнений сводятся многочисленные практические задачи. Можно с полным основанием утверждать, что решение линейных систем является одной из самых распространенных и важных задач вычислительной математики [1,2].

 (1)

Совокупность коэффициентов этой системы запишем в виде таблицы:



Запишем систему *n* линейных алгебраических уравнений с *n* неизвестными.

Данная таблица *n*2 элементов, состоящая из *n* строк и *n* столбцов, называется квадратной матрицей порядка *n*. Если подобная таблица содержит *nm* элементов, расположенных в *n* строках и *m* столбцах, то она называется прямоугольной матрицей.

Используя понятие матрицы *А*, систему уравнений (3) можно записать в векторно-матричном виде:



,

или, в более компактной записи,



где *х* и *b* — вектор-столбец неизвестных и вектор-столбец правых частей соответственно.

**3. Метод Крамера**

Алгоритм Крамера, согласно [1,2], выражается формулами



где 

…,

При этом необходимым и достаточным условием существование единственного решения, является не равенство нулю главного определителя системы

.

Блок-схема алгоритма представлена на рисунке.

Конец

Корней нет

Начало





Вычислить 

Вычислить:



Вычислить:





Да

Нет

**4. Программная реализации алгоритма МЕТОДА КРАМЕРА**

Основным методом класса Programm, является метод Main. С него начинается выполнение программы. В нашем случае, он содержит простейший пользовательский интерфейс, по средством которого пользователь вводит размерность системы, элементы матрицы системы *А* и вектора правых частей *b* (1, глава 1), а после необходимых вычислений на экране появляется результат – элементы вектора *x*.

В работе, алгоритм Крамера для большей читабельности, разбит на отдельные функции – методы:

static double det(int n, double [,]B) – метод вычисляющий определитель матрицы. Параметрами этого метода являются – количество уравнений (*n*), а так же матрица, в нашем случае *B*. Определитель матрицы вычисляется непосдедственно, т.е. разложением по первой строке [3];

static void equal(int n, double [,]A, double [,]B) – метод присваивающий матрицы (), где *n*-размерность матриц;

static int SLAU\_kramer(int n, double[,] A, double[] b, double[] x) – метод реализующий метод Крамера, согласно блок схеме главы 2.

В качестве языка программирования мы использовали объектно – ориентированный язык С#. Наш выбор обусловлен его гибкостью в разработке и создании программых продуктов [4-7].

Текст программы приведет ниже.

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApplication\_Kramer

{

class Program

{

static void Main(string[] args)

{

int n; /\* количество уравнений \*/

double [,] A = new double [3,3]; /\* матрица системы \*/

double [] b = new double [3]; /\* вектор правых частей \*/

double [] x = new double [3]; /\* вектор решения \*/

char qq;

Console.Write("Введите количество уравнений(<=3) n -> ");

n = Convert.ToInt32(Console.ReadLine());

if (n > 3 || n <= 1)

{

Console.WriteLine("Ошибка в размерности системы (n=2,3)");

Convert.ToInt32(Console.ReadLine());

return;

}

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

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

{

Console.Write("A{0}{1} -> ",i,j);

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

}

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

{

Console.Write("b{0} -> ", i);

b[i] = Convert.ToDouble(Console.ReadLine());

}

if(SLAU\_kramer(n,A, b, x)==1)

{

Console.WriteLine("Система не имеет решение");

Convert.ToInt32(Console.ReadLine());

return;

}

else

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

Console.WriteLine("x"+i+" = "+x[i]);

Console.ReadLine();

}

private

static double det(int n, double [,]B)

{

if (n == 2)

return B[0,0] \* B[1,1] - B[0,1] \* B[1,0];

return B[0,0] \* (B[1,1] \* B[2,2] - B[1,2] \* B[2,1]) - B[0,1] \* (B[1,0] \* B[2,2] - B[1,2] \* B[2,0])+

B[0,2]\*(B[1,0] \* B[2,1] - B[1,1] \* B[2,0]);

}

static void equal(int n, double [,]A, double [,]B)

{

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

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

A[i,j]=B[i,j];

}

static void change(int n, int N, double[,] A, double[] b)

{

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

A[i,N]=b[i];

}

public

static int SLAU\_kramer(int n, double[,] A, double[] b, double[] x)

{

double [,]An = new double [3,3];

double det1 = det(n, A);

if (det1 == 0) return 1;

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

{

equal(n, An, A);

change(n, i, An, b);

x[i] = det(n, An) / det1;

}

return 0;

}

}

}

Программа, реализующая метод Крамера, была протестирована на следующий тестовых примерах.

Решить систему второго порядка



решением систены является вектор

.

Результат выполнения программы представлен на рис. 1.

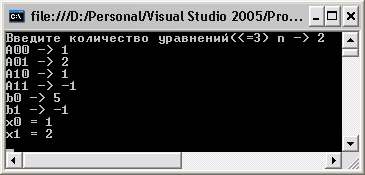


Рис. 1. Результат выполнения программы для системы второго порядка.

Решить систему третьего порядка



решением систены является вектор

.

Результат выполнения программы представлен на рис. 2.

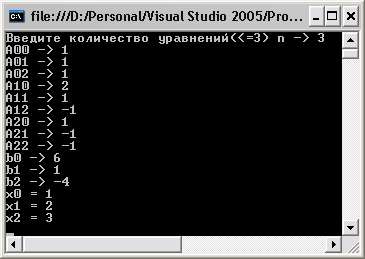


Рис. 2. Результат выполнения программы для системы третьего порядка.

В ходе тестированя, так же были рассмотрены случаи неправильного ввода размерности, результат выполнения на рис.3 и случай несовместности системы уравнений рис.4.

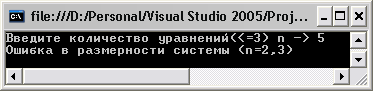


Рис. 3. Результат выполнения программы в слуае ошибочного ввода размерности системы.

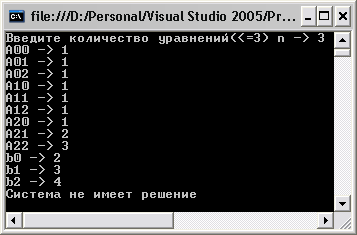


Рис. 4. Результат выполнения программы в случае несовместности системы.

Следует заметить, что в программе не ошуществляется проверка правильность ввода элементов матрицы и вектора правых частей.

**ЗАКЛЮЧЕНИЕ**

В работе, нами был программно реализован метод Крамера для решения системы линейных алгебраических уравнений. Мы использовали необходимое условие существования решения, т.е. не равенство нулю главного определителя системы.

Отличительная черта этого метода заключается в неоднократном вычислении определителя матрицы. С вычислительной точки зрения это трудоемкая операция с ростом количества элементов. В работе была рассмотрена система 3-го порядка, а определители вычислялись непосредственно.

Однако это не снижает ценность работы, так как переход к решению СЛАУ с неизвестным количеством уравнений осуществляется изменением метода, вычисляющий определитель матрицы.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Турчак Л.И. Основы численных методов / Л.И. Турчак, П.В. Плотников. - М.: ФИЗМАТЛИТ, 2002. - 304 с.

2. Демидович Б.П.Численные методы анализа / Б.П. Демидович, И.А. Марон,

Э.З. Шувалова. - М.: Наука, 1967.- 368 с.

3. Высшая математика для экономистов:Уч. Для вузов/Н.Ш. Кремер, Б.А. Путко, И.М. Тришин, М.Н. Фридман.-М.:Банки и биржи, 1998.-471 с.

4. Мохика Х. Язык С#: разработка Web-приложений на ASP.NET / Х. Мохика; пер. с англ. А.А. Слинкина. – М.: НТ Пресс, 2006. – 464 с. – (Quick Start).

5. Либерти Дж. Программирование на C#: пер. с англ. / Дж. Либерти. – 2-е изд. – СПб.: Символ, 2003. – 688 с.: ил.

6. С#: пер. с англ. / К. Ватсон, М. Беллиназо, О.Корнс и др. – СПб.: Питер, 2006. – 861 с.

7. Галисеев Г.В. Программирование на языке С#: самоучитель / Г.В. Галисеев. – М.: Вильямс, 2006. – 368 с.: ил.