## Федеральное агентство по образованию

Государственное образовательное учреждение высшего профессионального образования

## Тульский государственный университет

Кафедра “Автоматика и телемеханика”

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

по дисциплине: «Программирование на языке высокого уровня»

на тему: «Вычисление определителя матрицы прямым методом»

Выполнил: студент группы 260661

Ю.В. Красов

Проверил: ассистент кафедры АТМ

А.С. Карцева

Тула 2008

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ

1. ВЫБОР И ОБОСНОВАНИЕ ЧИСЛЕННОГО МЕТОДА РЕШЕНИЯ ЗАДАЧИ
	1. Определение матрицы
	2. Определение детерминанта
	3. Метод исключения Гаусса. Вычисление определителя методом исключения
2. АЛГОРИТМ РАБОТЫ ПРОГРАММЫ
	1. Структура алгоритма и данных
	2. Схема алгоритма
3. ТЕКСТ ПРОГРАММЫ
	1. Описание переменных и структур данных
	2. Текст программы на языке Pascal
4. ТЕСТОВАЯ ЗАДАЧА
	1. Математическое решение задачи
	2. Решение, полученное с использованием разработанного программного обеспечения
5. ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЮ

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

**ВВЕДЕНИЕ**

Современная математика ориентирована на использование компьютеров для прикладных расчетов. Любые математические приложения начинаются с построения модели явления (изделия, действия, ситуации или другого объекта), к которому относится изучаемый вопрос. Классическими примерами математических моделей могут служить определенный интеграл, уравнение колебаний маятника, уравнение теплообмена, уравнения упругости, уравнения электромагнитных волн и другие уравнения математической физики и даже модель формальных рассуждений – алгебру Буля.

Основополагающими средствами изучения математических моделей являются аналитические методы: получение точных решений в частных случаях (например, табличные интегралы), разложения в ряды. Определенную роль издавна играли приближенные вычисления. Например, для вычисления определенного интеграла использовались квадратурные формулы.

Появления в начале XX века электронных вычислительных машин (компьютеров) радикально расширило возможности приложения математических методов в традиционных областях (механике, физике, технике) и вызвало бурное проникновение математических методов в нетрадиционные области (управление, экономику, химию, биологию, психологию, лингвистику, экологию и т.п.).

Компьютер дает возможность запоминать большие (но конечные) массивы чисел и производить над ними арифметические операции и сравнения с большой (но конечной) скоростью по заданной вычислителем программе. Поэтому на компьютере можно изучать только те математические модели, которые описываются конечными наборами чисел, и использовать конечные последовательности арифметических действий, а также сравнений чисел по величине (для автоматического управления дальнейшими вычислениями).

С использованием компьютера стал возможен вычислительный эксперимент, т. e. расчет в целях проверки гипотез, а также в целях наблюдения за поведением модели, когда заранее не известно, что именно заинтересует исследователя. В процессе численного эксперимента происходит по существу уточнение исходной математической постановки задачи. В процессе расчетов на компьютере происходит накопление информации, что дает возможность в конечном счете произвести отбор наиболее интересных ситуаций. На этом пути сделано много наблюдений и открытий, стимулирующих развитие теории и имеющих важные практические применения.

С помощью компьютера возможно применение математических методов и в нетрадиционных областях, где не удается построить компактные математические модели вроде дифференциальных уравнений, но удается построить модели, доступные запоминанию и изучению на компьютере. Модели для компьютеров в этих случаях представляют собой цифровое кодирование схемы, изучаемого объекта (например, языка) и отношений между его элементами (словами, фразами). Сама возможность изучения таких моделей на компьютере стимулирует появление этих моделей, а для создания обозримой модели необходимо выявление законов, действующих в исходных объектах. С другой стороны, получаемые на компьютере результаты (например, машинный перевод упрощенных текстов с одного языка на другой) вносят критерий практики в оценку теорий (например, лингвистических теорий), положенных в основу математической модели.

Благодаря компьютерам стало возможным рассматривать вероятностные модели, требующие большого числа пробных расчетов, имитационные модели, которые отражают моделируемые свойства объекта без упрощений (например, функциональные свойства телефонной сети).

Разнообразие задач, где могут быть использованы компьютеры, очень велико. Для решения каждой задачи нужно знать многое, связанное именно с этой задачей.

Численные методы решения систем линейных алгебраических уравнений в линейной алгебре называют первой основной задачей. К ней примыкают задачи вычисления определителей и элементов обратной матрицы, которые иногда называют второй и третьей основными задачами линейной алгебры. В данной работе описаны методы вычисления определителя матрицы и разработана программа для его вычисления с использованием компьютера, основанная на применении метода Гаусса с выбором главного элемента.

**1. ВЫБОР И ОБОСНОВАНИЕ ЧИСЛЕННОГО МЕТОДА РЕШЕНИЯ ЗАДАЧИ**

* 1. **Определение матрицы**

Матрицей называют совокупность чисел, расположенных в прямоугольной таблице

,

состоящей из m строк и n столбцов.

Числа называют элементами матрицы. Первый индекс в обозначении элемента ( i ) указывает на номер строки, а второй индекс ( j )- на номер столбца, в которых расположен этот элемент.

В нашем случае () матрица называется прямоугольной размера . Если число строк в матрице равно числу столбцов (m=n), то матрицу называют квадратной порядка m.

* 1. **Определение детерминанта**

Для квадратной матрицы может быть введено понятие детерминанта (определителя). Детерминант матрицы [A] обозначают

 или .

*6*

Детерминантом матрицы порядка n>1 называют число

 , (1)

где - детерминант матрицы порядка n-1, полученной из матрицы [A] вычеркиванием первой строки и k -ого столбца.

Матрица порядка 1 состоит из одного числа, и ее детерминант по определению считают равным этому числу:

 (2)

Детерминант матрицы второго порядка в соответствии с (1) и (2) можно вычислить по следующей формуле:

.

Для матрицы третьего порядка

В соответствии с определением детерминант матрицы четвертого порядка может быть выражен через определитель третьего порядка, тот в свою очередь через определители второго порядка и т.д.

Число называют дополнительным минором элемента . Для произвольного элемента матрицы также можно ввести понятия дополнительного минора: - это определитель матрицы, получаемой из исходной вычеркиванием i -ой строки и j-ого столбца. Например, для матрицы [A] третьего порядка дополнительным минором элемента будет определитель

Одним из важных свойств определителей является то, что при перестановке местами двух строк или двух столбцов определителя, он должен быть умножен на -1:

.

При непосредственном вычислении определителей вышеприведенным способом, для отыскания решения системы линейных уравнений по правилу Крамера требуется приблизительно арифметических операций типа умножения. Использование метода исключения Гаусса позволяет уменьшить время, необходимое для решения задачи, до величины менее одной секунды.

* 1. **Метод исключения Гаусса. Вычисление определителя методом исключения**

Пусть дана матрица

Метод Гаусса можно интерпретировать как метод, в котором матрица приводится к верхней треугольной форме (прямой ход).

Приведем матрицу к верхней треугольной. Вычтем из второй строки первую, умноженную на такое число, при котором первый элемент второй строки обратится в нуль. То же проделаем со всеми остальными строками. В результате все элементы первого столбца, лежащие ниже главной диагонали, обратятся в нуль. Затем, используя вторую строку, обратим в нуль соответствующие элементы второго столбца. Последовательно продолжая этот процесс, приведем матрицу систему к верхней треугольной форме.

Запишем общие формулы метода Гаусса. Пусть проведено исключение к элементов из (k-1)-го столбца. Тогда останутся строки с ненулевыми элементами ниже главной диагонали.

Умножим k-ю строку на число

 m > k (3)

и вычтем из m-й строки. Первый ненулевой элемент этой строки обратится в нуль, а остальные изменятся по формулам

(4)

 k < m(5)

Проведя вычисления по этим формулам при всех указанных индексах, обратим в нуль элементы k-го столбца, лежащие ниже главной диагонали. Аналогичная процедура приводит матрицу системы к верхней треугольной форме, при этом весь процесс приведения называется прямым ходом метода Гаусса.

Определитель треугольной матрицы равен произведению диагональных элементов. В результате выполнения прямого хода метода исключения система линейных уравнений приводится к верхней треугольной матрице. Следовательно, определитель матрицы системы может быть вычислен как произведение диагональных элементов:

(6)

где k- количество перестановок строк при использовании метода исключения с выбором главного элемента.

На некотором шаге прямого хода может оказаться, что коэффициент но мал по сравнению с остальными элементами матрицы системы и, в частности, мал по сравнению с элементами первого столбца. Деление коэффициентов системы на малую величину может привести к значительным ошибкам округления.

Для уменьшения ошибок округления поступают следующим образом. Среди элементов первого столбца каждой промежуточной матрицы выбирают наибольший по модулю (главной) элемент и путем перестановки i-го строки со строкой, содержащей главный элемент, добиваются того, что главный элемент становится ведущим. Такая модификация метода исключения Гаусса называется методом Гаусса с выбором главного элемента. Случай появления нулевых элементов обходится при этом сам собой.

Важным достоинством данного метода, является то, что вычисление определителя требует примерно (2/3)n³ операций, что несравнимо меньше с операциями, при вычислении определителя по правилу Крамера, поэтому метод Гаусса с выбором главного элемента наиболее применим при обработке данных на компьютере.

**2. АЛГОРИТМ РАБОТЫ ПРОГРАММЫ**

**2.1 Структура алгоритма и данных**

Задача вычисления определителя матрицы разбивается на несколько подзадач:

1. Заполнение массива начальными данными;
2. Изменение порядка матрицы по желанию пользователя и заполнение нового массива;
3. Ввод и фильтрация вводимых пользователем данных в массив;
4. Вычисление детерминанта;

4.1) Ввод исходных данных в массив

4.2) Выбор главного элемента;

4.3) Замена строк местами;

4.4) Заполнение нового массива;

4.5) Приведение матрицы к верхнему треугольному виду;

4.6) Вычисление определителя матрицы.

Согласно вышеприведенной структуре, программа будет состоять из четырех подпрограмм:

1. Подпрограмма создания формы и ввода начальных данных в массив.

В данной подпрограмме задается начальное число столбцов и строк матрицы (ее порядок), вводятся заголовки матрицы, строк и столбцов в соответствии с заданным размером.

1. Подпрограмма изменения порядка матрицы;

В данной подпрограмме формируется новая матрица, исходя из данных, введенных пользователем, вводятся новые заголовки матрицы, строк и столбцов в соответствии с заданным размером.

1. Подпрограмма фильтрации вводимых пользователем данных, при нажатии на кнопки клавиатуры;

Данная подпрограмма разрешает пользователю вводить в матрицу только цифры, разделитель дробной и целой части и знак «-». Ввод других символов запрещается. Также в этой подпрограмме производится замена неверного разделителя на верный.

1. Подпрограмма вычисления определителя.

В данной подпрограмме происходит заполнение массива данных, поочередно для каждого столбца производится выбор главного элемента (наибольшего по модулю), затем строки меняются местами и производится приведение матрицы к верхней треугольной форме, т.е. когда ниже главной диагонали содержатся только нулевые элементы. Согласно вышеприведенным формулам производится вычисление значения детерминанта и полученный результат выводится на экран.

**2.2 Схема алгоритма**

На рисунке 1 представлен алгоритм работы программы при возникновении события OnCreate. Процедура TForm1.FormCreate(Sender: TObject).

Рис. 1. Алгоритм работы программы при возникновении события OnCreate

На рисунке 2 представлен алгоритм работы программы при нажатии на кнопку «Изменить размерность массива». Процедура TForm1.Button2Click(Sender: TObject).

Рис. 2. Алгоритм работы программы при нажатии на кнопку «Изменить размерность массива»

На рисунке 3 представлен алгоритм работы программы при вводе данных с клавиатуры (событие OnKeyPress). Процедура TForm1.StringGrid1KeyPress(Sender: TObject; var Key: Char).

Рис. 3. Алгоритм работы программы при при вводе данных с клавиатуры (событие OnKeyPress). Процедура TForm1.StringGrid1KeyPress(Sender: TObject; var Key: Char)

На рисунке 4 представлен алгоритм работы программы при нажатии на кнопку «Расчет». Процедура TForm1.Button1Click(Sender: TObject).

алгоритм программа pascal матрица определитель

Рис. 4. Алгоритм работы программы при нажатии на кнопку «Расчет»

**3. ТЕКСТ ПРОГРАММЫ**

**3.1 Описание переменных и структур данных**

При выполнении программы используются следующие переменные: N – максимальное число строк (столбцов) массива; r, c, max, j, z, p, s, zam – номера строк и столбцов и количество производимых замен строк – все они являются переменными типа integer (целое), переменные detA, k, buf – детерминант, коэффициент и буфер, используемый при замене строк – переменные типа extended (действительное число), а также переменная А – массив, тип массива – двумерный (Massiv = array[1..Nmax,1..Nmax] of extended).

При запуске программы возникает событие «создание формы» (OnCreate), процедура TForm1.FormCreate(Sender: TObject). При этом задается количество строк и столбцов двумерного массива (по умолчанию 4 и 4) StringGrid1.RowCount := N+1; StringGrid1.ColCount := N+1; но ячейки первой строки и первого столбца не редактируемые, они используются для вывода надписей над строками и столбцами, для чего используются функции StringGrid1.Cells [0,r] := ' r = ' + IntToStr(r) и StringGrid1.Cells [c,0] := ' c = ' + IntToStr(c). Вывод данных поочередно в каждую из этих ячеек производится посредством стандартной инструкции for … to … do begin … end.

Нажатие на кнопку влечет за собой возникновение события OnClick процедура TForm1.Button2Click(Sender: TObject). Данные о количестве строк и столбцов массива считываются из поля Edit1. Так как численное значение переменной N имеет целочисленный тип для преобразования строковой записи числа, находящегося в переменной Edit1.Text в целое, используется стандартная функция N:=StrToInt(Edit1.Text).

При вводе пользователем данных в поле StringGrid1 происходит событие OnKeyPress, процедура TForm1.StringGrid1KeyPress(Sender: TObject; var Key: Char). Посредством стандартной инструкции case Key of, которая позволяет реализовать множественный выбор, происходит фильтрация вводимых пользователем данных. Разрешается ввод только цифр, разделителя (DecimalSeparator), знака «минус» и нажатие клавиши Backspace. Посредством инструкции if Key <> DecimalSeparator then производится выбор одиного из двух возможных вариантов развития программы: разделитель заменяется на правильный, если он введен неверно (Key := DecimalSeparator), либо разделитель оставляется без изменений. Ввод остальных символов запрещается (else key := Chr(0)).

*20*

При нажатии на кнопку возникает событие (OnClick), процедура TForm1.Button1Click(Sender: TObject). Задаются начальные значения переменных max:= 1, detA := 1, zam:=0. Производится заполнение массива (A[c,r]:=StrToFloat(StringGrid1.Cells[c,r])), свойство StringGrid1.Cells[c,r] определяет содержимое ячейки с табличными координатами (c,r), строковые значения переменных, находящихся в ячейках (c,r) преобразуются в вещественный тип посредством стандартной инструкции StrToFloat. Стандартная инструкция for … to … do begin … end позволяет выполнять несколько раз действия, заключенные в этой инструкции. Функция abs(A[c,j]) возвращает модуль аргумента.

Инструкция if (stringgrid1.cells [c,r] > stringgrid1.cells [c-1,r]) and (stringgrid1.cells [c,r] < stringgrid1.cells [c+1,r]) then begin k := k+1; end; используется для выбора одного из вариантов развития программы, т.е. в случае выполнения данного условия число «особых» элементов увеличивается на 1 (k := k+1), если нет, то цикл повторяется до предпоследнего элемента матрицы. Реализует этот выбор стандартная инструкция for … to … do begin … end. Так как переменная detA действительное число, то для ее преобразования в строковый вид используется инструкция FloatToStrF(detA,fffixed,6,3). Функция zam mod 2 – проверка на четность количества замен строк, если число нечетное, то определитель умножается на -1.

Выход из программы осуществляется нажатием кнопки .

**3.2 Текст программы на языке Pascal**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Grids, Menus, Buttons;

type

TForm1 = class(TForm)

StringGrid1: TStringGrid;

Button1: TButton;

Edit1: TEdit;

Label1: TLabel;

Label3: TLabel;

Button2: TButton;

BitBtn1: TBitBtn;

BitBtn2: TBitBtn;

procedure FormCreate(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure StringGrid1KeyPress(Sender: TObject; var Key: Char);

private

{ Private declarations }

public

{ Public declarations }

end;

const

Nmax=10;

Type

Massiv1 = array[1..Nmax,1..Nmax] of extended;

var

Form1: TForm1;

A : Massiv1;

N, r, c: integer;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

N := 4;

Edit1.Text := FloatToStr(N);

StringGrid1.RowCount := N+1;

StringGrid1.ColCount := N+1;

Label3.Caption := ' для вычисления определителя матрицы нажмите расчет';

StringGrid1.Cells [0,0] := 'Матрица А';

for r := 1 to N do begin

StringGrid1.Cells [0,r] := ' строка ' + IntToStr(r);

end;

for c := 1 to N do begin

StringGrid1.Cells [c,0] := ' столбец ' + IntToStr(c);

end;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

N:=StrToInt(Edit1.Text);

StringGrid1.RowCount:=N+1;

StringGrid1.ColCount:=N+1;

for r := 1 to N do begin

StringGrid1.Cells [0,r] := ' строка ' + IntToStr(r);

end;

for c := 1 to N do begin

StringGrid1.Cells [c,0] := ' столбец ' + IntToStr(c);

end;

end;

procedure TForm1.Button1Click(Sender: TObject);

var

detA, k, buf: extended;

max, j, z, p, s, zam:integer;

begin

max:= 1;

detA := 1;

zam:=0;

for c := 1 to N do

for r := 1 to N do

A[c,r]:=StrToFloat(StringGrid1.Cells[c,r]);

for c := 1 to N-1 do begin

for z := c to N-1 do begin

max:=z;

for j := z+1 to N do begin

if abs(A[c,j]) > abs(A[c,max]) then

max:=j;

end;

 for p := 1 to N do begin

buf:=A[p,z]; A[p,z]:=a[p,max]; A[p,max]:=buf;

end;

end;

for r := c+1 to N do begin

k := A[c,r]/A[c,c];

for s := 1 to N do begin

A[s,r]:= A[s,r]-A[s,c]\*k;

end;

end;

if c<>max then begin

zam := zam+1;

end;

end;

for c := 1 to N do

detA := detA\*A[c,c];

if zam mod 2 <> 0 then

detA := (-1)\*detA;

label3.Caption := 'Детерминант матрицы равен: ' + FloatToStrF(detA,fffixed,6,3);

end;

procedure TForm1.BitBtn1Click(Sender: TObject);

begin

MessageDlg(Программа вычисляет детерминант (определитель) матрицы методом Гаусса с выбором главного элемента. Внимание!!! Матрица должна быть квадратной!',mtInformation,[mbOK],0);

end;

procedure TForm1.StringGrid1KeyPress(Sender: TObject; var Key: Char);

begin

case Key of

#8,'0'..'9', '-' : ;

'.',',':

begin

if Key <> DecimalSeparator then

Key := DecimalSeparator;

end;

else

key := Chr(0)

end;

end;

end.

**4. ТЕСТОВАЯ ЗАДАЧА**

**4.1 Математическое решение задачи**

В матрице вида

Определить детерминант.

Решение:

Вычисление определителя данной матрицы вручную целесообразно производить с помощью разложения элементов по 1-й строке по формуле (1). В итоге получится:

(7)

Воспользовавшись правилом Саррюса (правилом треугольников), вычисляются определители третьего порядка входящие в состав выражения (7):

detA = 4(10∙7∙2+(-20) ∙3∙5+5∙7∙10-5∙7∙10-10∙7∙3-5∙(-20) ∙2)-7∙(7,5∙7∙2+(-20) ∙3∙2+7∙3∙10-10∙7∙2-(-20)∙3∙2-7,5∙7∙3)+5∙(7,5∙5∙2+10∙3∙2+3∙5∙10-10∙5∙2-10∙3∙2-3∙5∙7,5)-6∙(7,5∙5∙7+3∙5∙(-20)+10∙7∙2-(-20) ∙5∙2-10∙3∙7-7,5∙7∙5)

detA = 4∙(140-300+350-350-210+200)-7∙(105-120+210-140+120-157,5)+5∙(75+60+150-100-60-112,5)-6∙(262,5-300+140+200-210-262,5)

detA = 4∙(-170)-7∙17,5+5∙12,5-6∙(-170)

detA = 280

Ответ: определитель матрицы равен 280

**4.2 Решение, полученное с использованием разработанного программного обеспечения**

Введя исходные данные в программу получим следующий результат: «Определитель матрицы равен: 280». Результат, полученный с использованием разработанной программы соответствует результату, вычисленному математически.

Вывод: разработанное программное обеспечение верно вычисляет определитель произвольной матрицы.

**5. ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЮ**

Для запуска программы необходимо запустить файл Determinant.exe, дважды щелкнув по нему мышью. В появившемся окне при необходимости изменить порядок матрицы, введя значение в поле напротив надписи «Порядок матрицы» и нажав на кнопку «Изменить порядок матрицы». В ячейках таблицы ввести значения элементов матрицы. Вводимые данные должны являться действительными числами, содержать только цифры, знак « - » и разделитель целой и дробной части. После заполнения ВСЕХ элементов матрицы нажать кнопку «Расчет». Ответ будет написан под таблицей в формате: «Детерминант матрицы равен: -280,000»

Выход из программы осуществляется с помощью кнопки .

Внешний вид окна программы представлен на рисунке 5.

Рис. 5. Внешний вид окна программы

**ЗАКЛЮЧЕНИЕ**

В данной работе были изучены численные методы нахождения определителя матрицы и выбран наиболее оптимальный, с точки зрения реализации его на компьютере – метод исключения с выбором главного элемента. Написана программа с использованием массивов. Данная программа позволяет определить детерминант матрицы размером N×N, размер матрицы задается пользователем, вводимые данные – действительные числа. Вычисление определителя матрицы является второй главной задачей линейной алгебры, и применяется при решении сложных систем линейных уравнений с несколькими неизвестными.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

*28*

1. Методические указания по выполнению курсовых работ по дисциплине «программирование» для студентов дневной формы обучения, обучающихся по программе направлений подготовки бакалавров 550200, 553000, 552800. Разработал О.С. Середин, к.ф.-м.н., доцент каф. АТМ. Тула 2003 г.
2. Бахвалов Н.С., Жидков Н.П. Кобельков Г.М. Численные методы. – М.: Лаборатория Базовых Знаний, 2000, – 624с.
3. Волосевич А.А. Язык Object Pascal и система программирования Delphi. Учебное пособие. Минск: Белорусский государственный университет информатики и радиоэлектроники, 2003, - 61с.
4. Курс лекций по дисциплине "вычислительная математика. Тула 2007, - 162с.

*29*

*29*