

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

Кафедра геометрии и компьютерных наук

Т.А. Фомина, О.Н. Казакова

СИСТЕМЫ АНАЛИТИЧЕСКИХ ВЫЧИСЛЕНИЙ

Часть 2

Методические указания

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по программам высшего образования по направлениям подготовки 01.03.01 Математика, 02.03.02 Фундаментальная информатика и информационные технологии

Оренбург
2017

УДК 512.5 (076.5)
ББК 22.14.я7
Ф 76

Рецензент – кандидат физико-математических наук, доцент О.А. Пихтилькова

Фомина, Т.А.

Ф 76 Системы аналитических вычислений: методические указания к лабораторным работам в 2 ч. Ч. 2/ Т.А. Фомина, О.Н. Казакова; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2017. – 49с.

Методические указания содержат материал, предназначенный для выполнения лабораторных работ по курсу «Системы аналитических вычислений».

Методические указания предназначены для обучающихся по направлениям подготовки 01.03.01 Математика, 02.03.02 Фундаментальная информатика и информационные технологии, и составлены в соответствии с утвержденными рабочими программами дисциплины «Системы аналитических вычислений». Материалы могут быть полезны преподавателям высших учебных заведений, ведущим соответствующую дисциплину и изучающим эту дисциплину студентам.

УДК 512.5 (076.5)
ББК 22.14.я7

© Фомина Т.А.,
Казакова О.Н, 2017
© ОГУ, 2017

Содержание

Введение.....	4
1 Программирование в среде Matlab.....	5
1.1 Лабораторная работа №6. Файл-функции и файл-программы.....	5
1.1.1 Теоретические сведения.....	5
1.1.2 Практическая часть.....	11
1.1.3 Контрольные вопросы.....	17
1.2 Лабораторная работа №7. Интерактивный ввод и вывод данных в m-сценариях.....	18
1.2.1 Теоретические сведения.....	18
1.2.2 Практическая часть.....	21
1.2.3 Контрольные вопросы.....	22
1.3 Лабораторная работа №8. Программирование.....	23
1.3.1 Теоретические сведения.....	23
1.3.2 Практическая часть.....	29
1.3.3 Контрольные вопросы.....	32
2 Символьные вычисления в среде Matlab.....	33
2.1 Лабораторная работа №9. Аналитические преобразования. Решение уравнений и систем уравнений.....	33
2.1.1 Теоретические сведения.....	33
2.1.2 Практическая часть.....	37
2.1.3 Контрольные вопросы.....	39
2.2 Лабораторная работа №10. Дифференцирование и интегрирование	40
2.2.1 Теоретические сведения.....	40
2.2.2 Практическая часть.....	42
2.2.3 Контрольные вопросы.....	45
2.3 Лабораторная работа №11. Исследование функции.....	45
2.3.1 Теоретические сведения.....	45
2.3.2 Практическая часть.....	46
2.3.3 Контрольные вопросы.....	47
Список использованных источников.....	49

Введение

Дисциплина «Системы аналитических вычислений» изучается обучающимися по направлениям подготовки 01.03.01 Математика, 02.03.02 Фундаментальная информатика и информационные технологии и направлена на изучение программирования с использованием математической системы Matlab.

Рабочей программой дисциплины предусмотрено выполнение студентами серии лабораторных работ, предназначенных для знакомства студентов с пакетом Matlab и получения навыков работы с ним.

Методические указания содержат материал, предназначенный для выполнения лабораторных работ и способствующий активной самостоятельной работе студентов.

Лабораторный цикл состоит из 11 работ. Каждая лабораторная работа содержит необходимый для выполнения индивидуальных заданий теоретический материал, практическую часть и список контрольных вопросов для самопроверки. Для проведения лабораторных работ необходим компьютер с установленным пакетом Matlab. От студентов требуются первичные навыки работы в среде Windows и знания основ программирования.

Первая часть методических указаний включает в себя пять лабораторных работ, позволяющих отработать основные навыки работы с данным математическим пакетом: основы работы с Matlab, операции с векторами и матрицами в Matlab и графические возможности среды Matlab.

Вторая часть включает в себя шесть лабораторных работ по двум разделам: программирование в пакете Matlab и символьные вычисления в среде Matlab.

Материалы могут быть полезны преподавателям высших учебных заведений, ведущим соответствующую дисциплину, и изучающим эту дисциплину студентам.

1 Программирование в среде Matlab

1.1 Лабораторная работа №6. Файл-функции и файл-программы

1.1.1 Теоретические сведения

Решение многих задач можно осуществить в командном режиме, т.е. без использования традиционного программирования. Однако нередко возникает необходимость сохранения определенной последовательности вычислений для последующего самостоятельного ее использования. Другими словами, возникает необходимость в создании программ.

Все команды, операторы и функции являются объектами входного языка системы Matlab. Наряду с ними язык программирования включает в себя управляющие структуры, системные операторы и многое другое. Но самое главное отличие языка программирования от входного языка заключается в способе фиксации создаваемых ими кодов.

Для языка программирования Matlab можно не различать понятия команды (выполняемой в командном режиме) и оператора (выполняемого из программы). Часто используется и понятие функции. Функция преобразует одни данные в другие.

Программы хранятся в виде М-файлов. Записанная программа (М-файл) становится частью системы и может вызываться как из командной строки, так и из другой программы.

Существует два типа М-файлов: файл-программы и файл-функции.

Файл-программы (скрипты)

Файл-программа представляет собой зафиксированную в виде файла последовательность операций, полностью аналогичную той, что используется в сессии. Файл-сценарий, именуемый также *script-файлом*, является просто записью серии команд без входных и выходных параметров. Он имеет следующую структуру:

%Основной комментарий

%Дополнительный комментарий

Тело файла с любыми выражениями

В Matlab имеется редактор М-файлов, для запуска которого следует нажать кнопку New M-file на панели инструментов рабочей среды, либо выбрать в меню File в пункте New подпункт М-file. На экране появляется окно редактора. Наберите в нем какие-либо команды, например для построения графика (листинг 1):

Листинг 1. Простейшая файл-программа

```
x = [-1:0.01:1];
```

```
y = exp(x);
```

```
plot(x, y)
```

```
grid on
```

```
title('Экспоненциальная функция')
```

Сохраните теперь файл с именем *myprog.m* в подкаталоге work основного каталога MatLab, выбрав пункт Save as меню File редактора. Для запуска на выполнение всех команд, содержащихся в файле, следует выбрать пункт Run в меню Debug, либо набрать в командной строке имя М-файла (без расширения) и нажать <Enter>, то есть выполнить, как команду Matlab. На экране появится графическое окно Figure No.1, содержащее график функции.

При таких способах запуска программы следует учесть важное обстоятельство – путь к каталогу с М-файлом должен быть известен Matlab. Сделайте каталог с файлом *myprog* текущим. Когда текущий каталог установлен, то все М-файлы, находящиеся в нем, могут быть запущены из командной строки, либо из редактора М-файлов.

Все переменные файл-программы после ее запуска доступны в рабочей среде, т.е. являются глобальными. Можно убедиться в этом, выполнив команду *whos*. Более того, файл-программа может использовать переменные рабочей среды.

Файл-функции

Файл-функции отличаются от файл-программ тем, что они могут иметь входные и выходные аргументы, а все переменные, определенные внутри файл-функции, являются локальными и не видны в рабочей среде.

Такой файл является типичным модулем с точки зрения структурного программирования. Структура этого модуля имеет вид:

```
function [var1, var2,...] = f_name(Список_параметров)
% Основной комментарий
%Дополнительный комментарий
Тело файла
var1 = выражение
var2 = выражение
.....
```

Если функция возвращает одно значение (var1), то квадратные скобки не ставятся.

Данная функция используется как отдельный элемент программы в виде:

```
[var1, var2,...] = f_name(Список_параметров)
```

Если же функция используется в программе в виде `f_name(Список_параметров)`, то возвращается только первый элемент множества выходных параметров – var1.

При желании внутри функции можно использовать и глобальные переменные. Для этого достаточно внутри функции сделать соответствующие объявления:

```
Global var1 var2 ...
```

Листинг 2 содержит пример простейшей файл-функции с двумя входными и одним выходным аргументами.

Листинг 2. Файл-функция mysum

```
function c = mysum(a,b)
c = a+b;
```

Наберите этот пример в новом файле в редакторе и сохраните его. Обратите внимание, что Matlab предлагает в качестве имени М-файла название файл-функции, т.е. *mysum.m*. Всегда сохраняйте файл-функцию в М-файле, имя которого совпадает с именем файл-функции! Убедитесь, что каталог с файлом *mysum.m* является текущим и вызовите файл-функцию *mysum* из командной строки:

```
>> s = mysum(2,3)
```

```
s =
```

```
5
```

При вызове файл-функции *mysum* произошли следующие события:

- входной аргумент *a* получил значение 2;
- входной аргумент *b* стал равен 3;
- сумма *a* и *b* записалась в выходной аргумент *c*;
- значение выходного аргумента *c* получила переменная *s* рабочей среды и

результат вывелся в командное окно.

Заметьте, что оператор $c = a + b$ в файл-функции *mysum* завершен точкой с запятой для подавления вывода локальной переменной *c* в командное окно. Для просмотра значений локальных переменных при отладке файл-функций, очевидно, не следует подавлять вывод на экран значений требуемых переменных.

Практически все функции Matlab являются файл-функциями и хранятся в одноименных М-файлах. Функция *sin* допускает два варианта вызова: $\sin(x)$ и $y = \sin(x)$, в первом случае результат записывается в *ans*, а во втором — в переменную *y*. Наша функция *mysum* ведет себя точно так же. Более того, входными аргументами *mysum* могут быть массивы одинаковых размеров или массив и число.

Разберем теперь, как создать *файл-функцию с несколькими выходными аргументами*.

В качестве примера на листинге 3 приведена файл-функция *quadeq*, которая по заданным коэффициентам квадратного уравнения находит его корни.

Листинг 3. Файл-функция для решения квадратного уравнения

```
function [x1,x2] = quadeq(a,b,c)
```

```
D = b^2-4*a*c;
```


$$x1 = (-b+\sqrt{D})/(2*a);$$

$$x2 = (-b-\sqrt{D})/(2*a);$$

При вызове *quadeq* из командной строки используйте квадратные скобки для указания переменных, в которые будут занесены значения корней:

```
>> [r1,r2] = quadeq(1,3,2)
```

```
r1 =
```

```
-1
```

```
r2 =
```

```
-2
```

Заметьте, что файл-функцию *quadeq* можно вызвать без выходных аргументов, или только с одним выходным аргументом. В этом случае вернется только первый корень.

Файл-функция может и не иметь входных или выходных аргументов, заголовки таких файл-функций приведены ниже:

```
function noout(a,b), function [v,u] = noin, function noarg()
```

Умение писать собственные файл-функции и файл-программы необходимо как при программировании в Matlab, так и при решении различных задач средствами Matlab (в частности, поиска корней уравнений, интегрирования, оптимизации). Разберем только один пример, связанный с построением графика функции $f(x) = e^{-x}(\sin x + 0.1 \sin(100\pi x))$ на отрезке $[0, 1]$. Запрограммируйте файл-функцию *myfun* для вычисления $f(x)$. Используйте поэлементные операции (листинг 4) для того, чтобы *myfun* можно было вызывать от вектора значений аргумента и получать вектор соответствующих значений функции.

Листинг 4. Файл-функция myfun

```
function y = myfun(x);
```

```
y = exp(-x).*(sin(x)+0.1*sin(100*pi*x));
```

График $f(x)$ можно получить двумя способами. Первый способ состоит в создании вектора значений аргумента, скажем с шагом 0.01, заполнении вектор значений функции и вызове *plot*:

```
>> x = [0:0.01:1];  
>> y = myfun(x);  
>> plot(x,y)
```

В результате получается график, приведенный на рисунке 1а, который, очевидно, неверен. Действительно, при вычислении значений функции на отрезке $[0, 1]$ с шагом 0.01 слагаемое $\sin(100\pi x)$ все время обращалось в ноль и *plot* построила график не $f(x)$, а другой функции. Непродуманный выбор шага часто приводит к потере существенной информации о поведении функции. В Matlab имеется встроенная функция *fplot* — некоторый аналог *plot*, но с автоматическим подбором шага при построении графика. Первым входным аргументом *fplot* является имя файл-функции, а вторым — вектор, элементы которого есть границы отрезков:

```
fplot('имя файл-функции', [a,b]).
```

Постройте теперь в новом окне график $f(x)$ при помощи *fplot*:

```
>> figure  
>> fplot('myfun',[0,1])
```

Получился график, точно отражающий поведение функции (рисунок 1б).

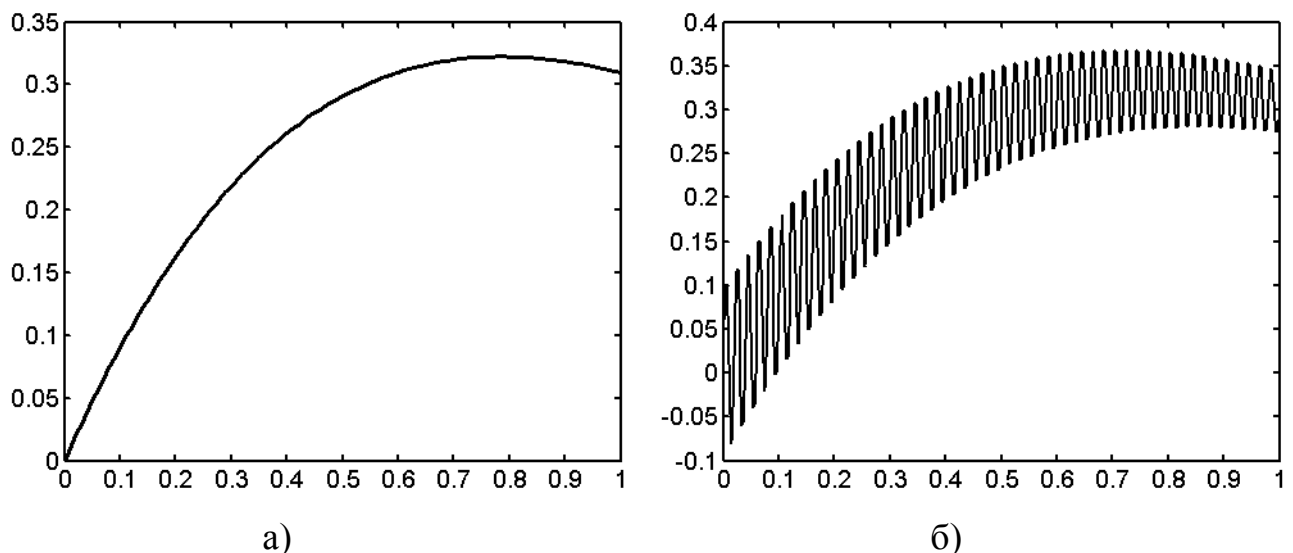


Рисунок 1 – График функции $f(x) = e^{-x}(\sin x + 0.1 \sin(100\pi x))$

1.1.2 Практическая часть

Контрольные задания

1) Написать файл-функции и построить графики на заданном отрезке при помощи *plot* (с шагом 0.05) и *fplot* для следующих функций:

Варианты:

$$1. f(x) = \sin \frac{1}{x}, x \in [0.05; 1]$$

$$2. f(x) = e^{3x \sin 5\pi x} + e^{3x \cos 5\pi x}, x \in [0; 1]$$

$$3. f(x) = \frac{10}{11 - 10 \sin 21\pi x}, x \in [0.05; 1]$$

$$4. f(x) = \sqrt{\frac{|\sin 21\pi x|}{2 + \sin 20\pi x}}, x \in [0; 1]$$

$$5. f(x) = \frac{1}{\arctg\left(\frac{1}{1.1 + \sin 5\pi x}\right) - \frac{3}{2}}, x \in [0; 1]$$

$$6. f(x) = \cos\left(\frac{1}{\frac{2\pi}{11} - \arctg x^x}\right), x \in [0; 1]$$

$$7. f(x) = \sin\left(6\pi\left|x - \frac{2}{3}x^3\right|\right), x \in [0; 1]$$

$$8. f(x) = \sin 2\pi \sqrt{\left|\sqrt{1 - x^3} - \frac{4}{7}\right|}, x \in [0; 1]$$

$$9. f(x) = |\sin 20\pi x|, x \in [0.05; 1]$$

$$10. f(x) = \frac{1}{\sin(e^{2x} - e^{-2x}) + \cos(e^{2x} - e^{-2x}) - \frac{3}{2}}, x \in [-1; 1].$$

$$11. f(x) = x^2 \operatorname{tg} \sqrt{\arcsin x}, x \in \left[0; \frac{1}{3}\right]$$

$$12. f(x) = |\cos 20\pi x|, x \in [0.05; 1]$$

$$13. f(x) = x^2 \operatorname{tg} \sqrt{\arcsin x}, x \in [-1; 1]$$

$$14. f(x) = \cos \frac{1}{x}, x \in [0.05; 1]$$

$$15. f(x) = x \sin x + x^3 \frac{e^x}{x+1}, x \in [0; 1]$$

2) Написать файл-функцию, которая для произвольной квадратной матрицы $n \times n$ находит значение выражения.

Варианты:

$$1. p = \prod_{i=1}^n \sum_{j=1}^n (a_{ij})^{a_{ij}}$$

$$9. s = \sum_{i=1}^n \sum_{j=1}^n |a_{ij}|$$

$$2. s = \sum_{i=1}^n \max_{j=1..n} (a_{ij} + a_{ji})$$

$$10. m = \max_{j=1..n} \left\{ \sum_{i=1}^n a_{ij}^2 \right\}$$

$$3. m = \max_{i=1..n} \min_{j=1..n} a_{ij}$$

$$11. s = 2 \sum_{i=1}^6 a_{ii} - \sum_{i=2}^6 a_{ii-1}$$

$$4. s = \sum_{i=1}^n \sum_{j=1}^n \sin \left(\frac{\pi}{6} a_{ij}^2 \right)$$

$$12. m = \min_{j=1, \dots, 6} \left\{ \sum_{i=1}^6 a_{ij}^3 \right\}$$

$$5. s = \sum_{i=1}^n a_{ii} + \sum_{i=1}^{n-1} a_{ii+1}$$

$$13. s = \sum_{i=2}^6 a_{ii-1}$$

$$6. s = \sum_{i=1}^{n-1} a_{ii+1}$$

$$14. s = \sum_{i=1}^6 \sum_{j=1}^6 \cos \left(\frac{\pi}{3} a_{ij}^3 \right)$$

$$7. s = \sum_{k=1}^n a_{kk}^3$$

$$15. s = \sum_{i=1}^6 \min_{j=1, \dots, 6} (a_{ij} + a_{ji})$$

$$8. m = \min_{i, j=1..n} a_{ij}^3$$

3) Написать m-сценарии для заполнения следующих матриц.

Варианты:

$$1. \quad a) \begin{bmatrix} 1 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 0 & 0 & 0 & 5 & 0 & 2 \\ 1 & 0 & 0 & 5 & 0 & 5 & 3 \\ 1 & 0 & 5 & 0 & 5 & 0 & 4 \\ 1 & 5 & 0 & 5 & 0 & 0 & 5 \\ 1 & 0 & 5 & 0 & 0 & 0 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 & 1 \end{bmatrix}$$

$$б) A = \begin{bmatrix} 1 & 0 & 0 & 1 & -3 & -3 \\ 0 & 1 & 1 & 0 & -3 & -3 \\ 0 & 1 & 1 & 0 & -3 & -3 \\ 1 & 0 & 0 & 1 & -3 & -3 \\ -2 & -2 & -2 & -2 & 4 & 4 \\ -2 & -2 & -2 & -2 & 4 & 4 \end{bmatrix}$$

$$2. \quad a) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 3 & 0 & 5 & 0 & 1 \\ 1 & 0 & 0 & 3 & 0 & 0 & 1 \\ 1 & 0 & 5 & 0 & 3 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 3 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$б) A = \begin{bmatrix} 1 & 0 & 0 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 4 & 0 \\ 0 & 0 & 1 & 0 & 0 & 4 \\ 2 & 2 & 2 & 6 & 6 & 6 \\ 2 & 2 & 2 & 6 & 6 & 6 \\ 2 & 2 & 2 & 6 & 6 & 6 \end{bmatrix}$$

$$3. \quad a) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$$б) A = \begin{bmatrix} 0 & 0 & 0 & 0 & 3 & 4 \\ 0 & 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 3 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 & 0 & 4 \\ 3 & 0 & 0 & 0 & 0 & 4 \\ 1 & 1 & 1 & 1 & 1 & -2 \end{bmatrix}$$

$$4. \quad a) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$б) A = \begin{bmatrix} -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 4 & 4 \end{bmatrix}$$

$$5. \quad a) \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$b) \quad A = \begin{bmatrix} -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 3 & 3 & 3 \\ 2 & 2 & 2 & 3 & 3 & 3 \\ 2 & 2 & 2 & 3 & 3 & 3 \end{bmatrix}$$

$$6. \quad a) \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 2 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 2 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 2 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$b) \quad A = \begin{bmatrix} 1 & 1 & -3 & -3 & -3 & -3 \\ 1 & 1 & -3 & -3 & -3 & -3 \\ -3 & -3 & 2 & 0 & 0 & 0 \\ -3 & -3 & 0 & 2 & 0 & 0 \\ -3 & -3 & 0 & 0 & 2 & 0 \\ -3 & -3 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$7. \quad a) \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 1 \\ 2 & 0 & 0 & 0 & 7 & 0 & 1 \\ 3 & 0 & 0 & 7 & 0 & 7 & 1 \\ 4 & 0 & 7 & 0 & 7 & 0 & 1 \\ 5 & 7 & 0 & 7 & 0 & 0 & 1 \\ 6 & 0 & 7 & 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 1 \end{bmatrix}$$

$$b) \quad A = \begin{bmatrix} 1 & 1 & -3 & -3 & -3 & -3 \\ 1 & 1 & -3 & -3 & -3 & -3 \\ -3 & -3 & 6 & 0 & 0 & 0 \\ -3 & -3 & 0 & 6 & 0 & 0 \\ -3 & -3 & 0 & 0 & 6 & 0 \\ -3 & -3 & 0 & 0 & 0 & 6 \end{bmatrix}$$

$$8. \quad a) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$b) \quad A = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 2 & 0 & 4 \\ 0 & 0 & 2 & 0 & 0 & 4 \\ 0 & 2 & 0 & 0 & 0 & 4 \\ 2 & 0 & 0 & 0 & 0 & 4 \\ 1 & 1 & 1 & 1 & 1 & -3 \end{bmatrix}$$

$$9. \quad a) \begin{bmatrix} 4 & 3 & 3 & 3 & 3 & 3 & 9 \\ 3 & 4 & 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 4 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 4 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 4 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 & 4 & 3 \\ 9 & 3 & 3 & 3 & 3 & 3 & 4 \end{bmatrix}$$

$$b) A = \begin{bmatrix} 2 & 2 & -1 & -1 & 2 & 2 \\ 2 & 2 & -1 & -1 & 2 & 2 \\ 2 & 2 & -1 & -1 & 2 & 2 \\ 2 & 2 & -1 & -1 & 2 & 2 \\ 2 & 2 & -1 & -1 & 2 & 2 \\ 2 & 2 & -1 & -1 & 2 & 2 \end{bmatrix}$$

$$10. \quad a) \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$b) A = \begin{bmatrix} 1 & 0 & 0 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 4 & 0 \\ 0 & 0 & 1 & 0 & 0 & 4 \\ 2 & 2 & 2 & 3 & 3 & 3 \\ 2 & 2 & 2 & 3 & 3 & 3 \\ 2 & 2 & 2 & 3 & 3 & 3 \end{bmatrix}$$

$$11. \quad a) \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$b) A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 3 & -1 & 0 & 0 & 0 & 0 \\ 0 & 3 & -1 & 0 & 0 & 0 \\ 0 & 0 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 4 & 4 \\ 0 & 0 & 0 & 0 & 4 & 4 \end{bmatrix}$$

$$12. \quad a) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$b) A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ -1 & -2 & -3 & -4 & -5 & -6 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

$$13. \quad \text{a)} \quad \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 5 \\ -1 & 3 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 5 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 6 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 7 & 0 \\ 5 & 0 & 0 & 0 & 0 & -1 & 8 \end{bmatrix}$$

$$\text{б)} \quad A = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 \\ -1 & -1 & -1 & 4 & 0 & 0 \\ -1 & -1 & -1 & 0 & 4 & 0 \\ -1 & -1 & -1 & 0 & 0 & 4 \end{bmatrix}$$

$$14. \quad \text{a)} \quad \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{б)} \quad A = \begin{bmatrix} 1 & 0 & 0 & 0 & -5 & -5 \\ 0 & 1 & 0 & 0 & -5 & -5 \\ 0 & 0 & 1 & 0 & -5 & -5 \\ 0 & 0 & 0 & 1 & -5 & -5 \\ -3 & -3 & -3 & -3 & 4 & 4 \\ -3 & -3 & -3 & -3 & 4 & 4 \end{bmatrix}$$

$$15. \quad \text{a)} \quad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 & 0 & 7 & 1 \\ 1 & 0 & 2 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\text{б)} \quad A = \begin{bmatrix} 1 & 2 & 3 & -1 & 0 & 0 \\ 1 & 2 & 3 & 0 & -1 & 0 \\ 1 & 2 & 3 & 0 & 0 & -1 \\ 0 & 0 & 7 & 2 & 2 & 2 \\ 0 & 7 & 0 & 2 & 2 & 2 \\ 7 & 0 & 0 & 2 & 2 & 2 \end{bmatrix}$$

4) Решить поставленную задачу.

Варианты:

1. Написать файл-функцию, которая по заданному вектору определяет номер его элемента с наибольшим отклонением от среднего арифметического всех элементов вектора.

2. Написать файл-функцию, возвращающую сумму всех элементов вектора с нечетными индексами.

3. Написать файл-функцию, вычисляющую максимальное значение среди диагональных элементов заданной матрицы.

4. Написать файл-функцию, переставляющую первый столбец квадратной матрицы с ее диагональю.
5. Написать файл-функцию, которая суммирует все внедиагональные элементы заданной матрицы.
6. Написать файл-функцию, заменяющую максимальный элемент вектора средним значением всех его элементов.
7. Написать файл-функцию, заменяющую элемент матрицы с индексами 1,1 произведением всех элементов матрицы.
8. Написать файл-функцию, возвращающую сумму всех элементов вектора с четными индексами.
9. Написать файл-функцию, переставляющую первую строку квадратной матрицы с ее диагональю.
10. Написать файл-функцию, заменяющую элемент матрицы с индексами 1,1 суммой квадратов всех элементов матрицы.
11. Написать файл-функцию, которая по заданному вектору определяет номер его элемента с наименьшим отклонением от среднего арифметического всех элементов вектора.
12. Написать файл-функцию, вычисляющую минимальное значение среди диагональных элементов заданной матрицы.
13. Написать файл-функцию, переставляющую третий столбец квадратной матрицы с ее диагональю.
14. Написать файл-функцию, возвращающую сумму всех элементов вектора с нечетными индексами.
15. Написать файл-функцию, заменяющую элемент матрицы с индексами 3,1 суммой квадратов всех элементов матрицы.

4.1.3 Контрольные вопросы

- 1) Как задается место хранения рабочих файлов в Matlab?
- 2) Какие типы М-файлов существуют?

- 3) Какое расширение имеют m-файлы в MatLab?
- 4) Чем отличается файл-сценарий от файл-функции?
- 5) В какие скобки помещается список входных аргументов m-функции?
- 6) В какие скобки помещается список выходных аргументов m-функции?
- 7) Как создаются, открываются, сохраняются и запускаются на исполнение m-файлы?
- 8) Какую команду нужно ввести в командное окно, чтобы вызвать редактор m-файлов системы MatLab?
- 9) Как выполнить несколько строк из окна редактирования в среде Matlab?
- 10) Какой тип m-файлов в системе Matlab допускает наличие входные и выходные аргументов, предназначен для расширения возможностей языка Matlab?

1.2 Лабораторная работа №7. Интерактивный ввод и вывод данных в m-сценариях

1.2.1 Теоретические сведения

Для организации диалогового ввода и вывода используются следующие операторы, представленные в таблице 1.

Таблица 1– Операторы диалогового ввода/вывода

Оператор	Синтаксис	Назначение
INPUT	$x = \text{input} ('<\text{приглашение}>')$	Для ввода данных с клавиатуры
DISP	$\text{disp} (<\text{переменная или текст в апострофах}>)$	Для вывода на дисплей

Функция *input* выдает запрос и присваивает переменной *x* введенное с клавиатуры значение, при этом если вводится арифметическое выражение, то функция его вычисляет. Для ввода символьного выражения необходимо добавить строку 's' к списку параметров функции:

```
name = input ('Input your name, please: ', 's')
```

Если выводимое значение – число, то вначале его преобразуют к строковому типу при помощи функций *int2str* или *num2str*. Конкатенацию строк производят как для одномерных векторов-строк.

Приведем простой пример диалоговой программы:

```
1 - name = input ('Hello! What is your name?\n', 's');
2 - y = input (['Very good, ', name, '. And how old are you?\n']);
3 - disp(['Resume: Mr(s) ', name, ' is ', int2str(y), ' years old.'])
```

В первой строке данного *m*-сценария в командное окно выводится запрос и ответ на него пользователя вводится в строчную переменную *name*.

Во второй строке выводится запрос, использующий значение переменной *s*, и ответ пользователя помещается в переменную *y*.

В третьей строке выводится строка, являющаяся объединением конкретных строк и значений переменных, полученных на предыдущих этапах.

Данный *m*-сценарий с названием *hup* при вызове организует диалог следующего характера:

```
>> hup
Hello! What is your name?
Andy
Very good? Andy. And old are you?
21
Resume: Mr(s) Andy is 21 years old.
```

Вставка комментариев в m-функцию

Комментарии могут занимать отдельные строки, начинающиеся с символа %, после которого следует текст комментария. Также комментарии можно располагать в конце любой строки кода, поскольку интерпретатор М-языка, встретив знак %, считает все символы после него просто комментарием (а не командами, подлежащими переводу в машинную форму и исполнению).

Особую роль в системе Matlab имеют комментарии, располагающиеся в смежном наборе строк сразу за заголовком определения функции. Весь этот набор

строк выводится в командное окно системы Matlab при исполнении команды *help имя_M-функции*.

Поскольку такую команду в первую очередь будут применять пользователи функции (а не разработчики), то желательно расположить в этих комментариях описательную информацию и сведения о правильном вызове этой функции.

Отладка М-функций

Теперь подробно остановимся на вопросе об отладке М-функций, то есть на приёмах, с помощью которых можно выявить месторасположение ошибок и их причину. Система Matlab осуществляет серьёзную помощь в этом процессе. В частности, при возникновении ошибки в процессе выполнения М-функции, в командное окно выводится приблизительное диагностическое сообщение и номер строки, в котором, по мнению Matlab, произошла ошибка.

Другим, более развитым способом отладки функции является применение точек останова и пошаговое выполнения тела функции. Для этого применяют встроенные возможности редактора-отладчика системы Matlab. То, что уже многократно применяемый нами редактор (в нём набираем текст функций и с помощью меню сохраняем в файле) заодно является и отладчиком, говорит даже заголовок его окна *Matlab Editor/Debugger* (так как *debugger* в переводе с английского означает "отладчик").

Чтобы поставить "точку останова" на какой-либо строке кода функции, туда нужно поместить курсор и нажать клавишу F12 (повторное нажатие этой клавиши убирает точку останова). Вместо нажатия этой клавиши можно выполнить команду меню *Debug / Set/Clear Breakpoint*, но всё же быстрее это можно выполнить нажатием клавиши. После этого в строке слева появляется красный кружок, указывающий на то, что в данной строке проставлена точка останова. После этого, не закрывая окна Редактора/Отладчика (*Editor/Debugger*), переключаем фокус ввода с клавиатуры в командное окно Matlab и запускаем обычным образом функцию на выполнение. После этого и произойдёт останов выполнения функции прямо на строке, в которой поставлена точка останова (*Breakpoint*).

Теперь мы можем просматривать фактические значения входных параметров функции, текущие значения глобальных и локальных переменных, а также значения выражений. Чтобы посмотреть значение переменной, достаточно подвести курсор к её имени в тексте функции, после чего на экране появится всплывающий жёлтый прямоугольник со значением переменной внутри него (рисунок 2):

Далее, нажимая клавишу F10 мы можем выполнять функцию построчно, каждый раз проверяя результаты такой пошаговой работы функции. В результате всегда можно "окружить ошибку" и выявить её причину.

Изменив текст функции и устранив выявленную ошибку, запускаем функцию на выполнение, в результате чего либо убеждаемся в её правильной работе, либо находим новую ошибку. Желательно продумать методику отладки, запуская функцию на выполнение с разными значениями аргументов и разными значениями глобальных функций. В результате такого итерационного отладочного процесса приходят к правильно работающим функциям.

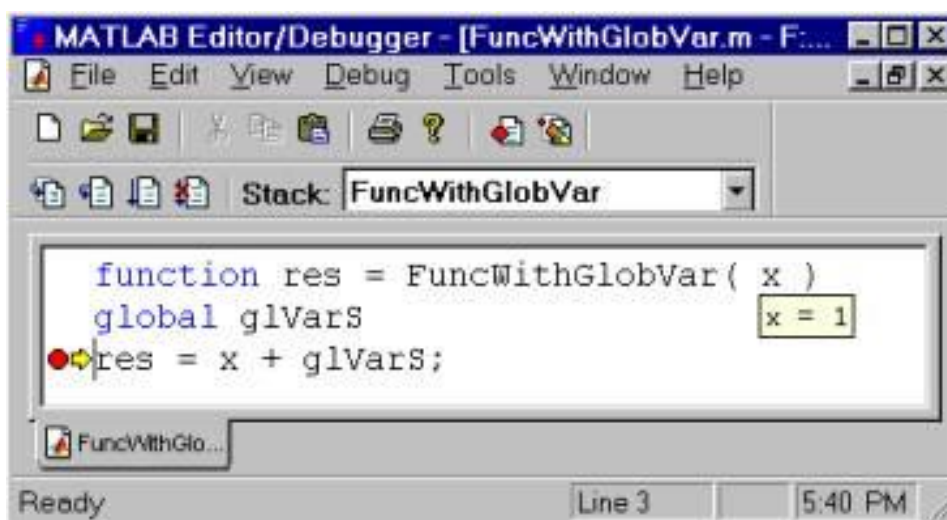


Рисунок 2 – Отладка программы

1.2.2 Практическая часть

1) Написать m-сценарий, организующий с пользователем диалог следующего вида:

Варианты 1, 3, 5, 7, 9, 11, 13, 15:

Hello! What is your name?	(вопрос, задаваемый из скрипта)
Nick	(ответ пользователя)
Very good, Nick! And how old are you?	(вопрос, задаваемый из скрипта)
19	(ответ пользователя)
Thank you, Nick! I think, that you was born in 1988.	(вывод сообщения)

Варианты 2, 4, 6, 8, 10, 12, 14:

Hello! What is your name?	(вопрос, задаваемый из скрипта)
Mary	(ответ пользователя)
Very good, Mary! In which year was you born?	(вопрос, задаваемый из скрипта)
1989	(ответ пользователя)
Thank you, Mary! I think, that you are 18 years old.	(вывод сообщения)

2) Добавьте в любую m-функцию из предыдущей лабораторной работы различные виды комментариев (строчные, блочные и содержащие информацию для команды help). Проверьте действие добавленных комментариев с помощью команды help.

1.2.3 Контрольные вопросы

- 1) Какие операторы используются для организации диалогового ввода и вывода?
- 2) Является ли правильным утверждение, что переменные, определенные в файл-функции, после ее выполнения становятся доступны в рабочем пространстве и могут использоваться в других файл-функциях?
- 3) Как происходит отладка программ?
- 4) Как осуществить конкатенацию строк?

- 5) Как поставить "точку останова" на какой-либо строке кода функции?
- 6) Как проверить результаты пошаговой работы функции?
- 7) Какой цвет по умолчанию использует редактор m-файлов для выделения синтаксических ошибок в коде программы?
- 8) Какие функции преобразуют число к строковому типу?
- 9) Какие виды комментариев существуют в Matlab?
- 10) Какой символ используется для обозначения комментариев в языке Matlab?

1.3 Лабораторная работа №8. Программирование

1.3.1 Теоретические сведения

Язык программирования Matlab достаточно простой, он содержит основной набор конструкций: операторы ветвления и циклы. Простота языка программирования окупается огромным количеством встроенных функций, которые позволяют решать задачи из различных областей.

Цикл *for* используется для повторения операторов в случае, когда число повторений заранее известно. В цикле *for* используется счетчик цикла, его начальное значение, шаг и конечное значение указываются через двоеточие. Блок операторов, размещенный внутри цикла, должен заканчиваться словом *end*.

Рассмотрим файл-программу для вывода графиков функции $f(x, \beta) = e^{\beta x} \sin x$ на отрезке $[-2, 2]$, для значений параметра $\beta \in [-0.5, 0.5]$:

Листинг 5. Графики функции при различных значениях параметра

```
x = [-2:0.01:2];  
for beta = -0.5:0.1:0.5  
y = exp(beta*x).*sin(x);  
plot(x,y)  
hold on  
end  
hold off
```

Если шаг равен единице, то его указывать не обязательно. Например, для вычисления суммы $\sum_{k=1}^{10} \frac{x^k}{k!}$ при различных значениях x потребуется файл-функция, текст которой приведен на листинге 6. Обратите внимание, что *sum10* может быть вызвана как от числа, так и от массива значений, благодаря применению поэлементных операций.

Листинг 6. Файл-функция для вычисления суммы

```
function s = sum10(x)
s = 0;
for k = 1:10
s = s+x.^k/factorial(k);
end
```

Цикл *for* подходит для повторения заданного числа определенных действий. В том случае, когда число повторов заранее неизвестно и определяется в ходе выполнения блока операторов следует организовать цикл *while*. Цикл *while* работает, пока выполнено условие цикла.

В системе Matlab могут применяться следующие операторы сравнения, приведенные в таблице 2.

Таблица 2 – Операторы сравнения

Символ	Назначение	Имя функции
<	Меньше	lt
>=	Больше или равно	ge
>	Больше	gt
<=	Меньше или равно	le
==	Равно	eq
~=	Не равно	ne

Операции (==, ~=) проводят сравнение вещественных и мнимых частей комплексных чисел, а операции (>, <, >=, <=) – только вещественных частей.

Логические операции можно записывать в виде функций (таблица 3).

Таблица 3 – Логические операции

Символ	Назначение	Имя функции
&	Логическое «и»	and
	Логическое «или»	or
~	Отрицание	not

Результатом логических операций являются числа 0 (false) и 1(true).

Файл-функция *negsum* (листинг 7) находит сумму всех первых отрицательных элементов вектора.

Листинг 7. Файл-функция negsum

```
function s = negsum(x)
```

```
s = 0;
```

```
k = 1;
```

```
while x(k)<0
```

```
s = s+x(k);
```

```
k = k+1;
```

```
end
```

Файл-функция *negsum* имеет один недостаток: если все элементы массива — отрицательные числа, то k становится больше длины массива x , что приводит к ошибке, например:

```
” b = [-2 -7 -1 -9 -2 -5 -4];
```

```
” s = negsum(b)
```

```
??? Index exceeds matrix dimensions.
```

Кроме проверки значения $x(k)$ следует позаботиться о том, чтобы значение k не превосходило длины вектора x . Вход в цикл должен осуществляться только при одновременном выполнении условий $k \leq \text{length}(x)$ и $x(k) < 0$, т.е. необходимо применить логический оператор "и". Замените условие цикла на составное: $k \leq \text{length}(x) \ \& \ x(k) < 0$. Если первое из условий не выполняется, то второе условие проверяться не будет, именно поэтому выбран такой порядок операндов. Теперь файл-функция *negsum* работает верно для любых векторов.

Ниже приведены логические операции по мере убывания их приоритета:

- отрицание \sim ;
- операторы отношения $>$, $<$, $>=$, $<=$, $==$, $\sim=$;
- логическое "и" $\&$;
- логическое "или" $|$.

Для изменения порядка выполнения логических операторов используются круглые скобки.

Циклы могут быть вложены друг в друга. Например, для поиска суммы элементов матрицы, расположенных выше главной диагонали, следует использовать два цикла *for*, причем начальное значение счетчика внутреннего цикла зависит от текущего значения счетчика внешнего цикла (листинг 8).

Листинг 8. Использование вложенных циклов

```
function s = upsum(A)
[n m] = size(A);
s = 0;
for i = 1:n
for j = i+1:m
s = s+A(i,j);
end
end
```

Ветвление в ходе работы программы осуществляется при помощи конструкции *if-elseif-else*. Самый простой вариант ее использования (без *elseif* и *else*) реализован в файл-функции *possum* (листинг 9), которая предназначена для нахождения суммы всех положительных элементов вектора.

Листинг 9. Файл-функция для суммирования положительных элементов вектора

```
function s = possum(x)
s = 0;
for k = 1:length(x)
if x(k)>0
s = s + x(k);
```

end

end

Если ход программы должен изменяться в зависимости от нескольких условий, то следует использовать полную конструкцию *if-elseif-else*. Каждая из ветвей *elseif* в этом случае должна содержать условие выполнения блока операторов, размещенных после нее. Важно понимать, что условия проверяются подряд, первое выполненное условие приводит к работе соответствующего блока, выходу из конструкции *if-elseif-else* и переходу к оператору, следующему за *end*. У последней ветви *else* не должно быть никакого условия. Операторы, находящиеся между *else* и *end*, работают в том случае, если все условия оказались невыполненными.

Предположим, что требуется написать файл-функцию для вычисления

кусочно-заданной функции:
$$f(x) = \begin{cases} 1 - e^{-1-x}, & x < -1 \\ x^2 - x - 2, & -1 \leq x \leq 2. \\ 2 - x, & x > 2 \end{cases}$$

Первое условие $x < -1$ проверяется в ветви *if*. Обратите внимание, что условие $-1 \leq x$ не требуется включать в следующую ветвь *elseif* (листинг 10), поскольку в эту ветвь программа заходит, если предыдущее условие ($x < -1$) оказалось не выполнено. Условие $x > 2$ проверять не надо – если не выполнены два предыдущих условия, то x будет больше двух.

Листинг 10. Файл-функция для вычисления кусочно-заданной функции

```
function f = pwf(x)
```

```
if x < -1
```

```
f = 1 - exp(-1 - x);
```

```
elseif x <= 2
```

```
f = x^2 - x - 2;
```

```
else
```

```
f = 2 - x;
```

```
end
```

Ход работы программы может определяться значением некоторой переменной (переключателя). Такой альтернативный способ ветвления программы основан на использовании оператора переключения *switch*. Переменная-переключатель помещается после *switch* через пробел. Оператор *switch* содержит блоки, начинающиеся со слова *case*, после каждого *case* записывается через пробел то значение переключателя, при котором выполняется данный блок. Последний блок начинается со слова *otherwise*, его операторы работают в том случае, когда ни один из блоков *case* не был выполнен. Если хотя бы один из блоков *case* выполнен, то происходит выход из оператора *switch* и переход к оператору, следующему за *end*.

Предположим, что требуется найти количество единиц и минус единиц в заданном массиве и, кроме того, найти сумму всех элементов, отличных от единицы и минус единицы. Следует перебрать все элементы массива в цикле, причем в роли переменной-переключателя будет выступать текущий элемент массива. Листинг 11 содержит файл-функцию, которая по заданному массиву возвращает число минус единиц в первом выходном аргументе, число единиц – во втором, а сумму – в третьем.

Листинг 11. Файл-функция mpsum

```
function [m,p,s] = mpsum(x)
m = 0;
p = 0;
s = 0;
for i = 1:length(x)
switch x(i)
case -1
m = m+1;
case 1
p = p+1;
otherwise
s = s+x(i);
end
```

end

Блок *case* может быть выполнен не только при одном определенном значении переключателя, но и в том случае, когда переключатель принимает одно из нескольких допустимых значений. В этом случае значения указываются после слова *case* в фигурных скобках через запятую, например: *case* {1,2,3}.

Досрочное завершение цикла *while* или *for* осуществляется при помощи оператора *break*. Пусть, например, требуется по заданному массиву *x* образовать новый массив *y* по правилу $y(k)=x(k+1)/x(k)$ до первого нулевого элемента $x(k)$, т.е. до тех пор, пока имеет смысл операция деления. Номер первого нулевого элемента в массиве *x* заранее неизвестен, более того, в массиве *x* может и не быть нулей. Решение задачи состоит в последовательном вычислении элементов массива *y* и прекращении вычислений при обнаружении нулевого элемента в *x*. Файл-функция, приведенная на листинге 12, демонстрирует работу оператора *break*.

Листинг 12. Использование оператора break для выхода из цикла

```
function y = div(x)
for k = 1:length(x)-1
if x(k) == 0
break
end
y(k) = x(k+1)/x(k);
end
```

1.3.2 Практическая часть

Контрольные задания

1) Написать файл-функцию для вычисления кусочно-заданной функции.

Варианты:

$$1. f(x) = \begin{cases} -1 & -3 \leq x \leq -1 \\ x & -1 < x \leq 1 \\ e^{1-x} & 1 < x \leq 3 \end{cases}$$

$$2. f(x) = \begin{cases} \frac{3 + \sin^2 x}{1 + x^2} & x \leq 0 \\ 2x^2 \cos^2 x & x > 0 \end{cases}$$

$$3. f(x) = \begin{cases} \ln x & 1 \leq x \leq e \\ \frac{x}{e} & e < x \leq 9 \\ 9e^{8-x} & 9 < x \leq 12 \end{cases}$$

$$4. f(x) = \begin{cases} \sin x & -2\pi \leq x \leq 0 \\ -x^3 & 0 < x \leq 1 \\ \cos \pi x & 1 < x \leq 3\pi \end{cases}$$

$$5. f(x) = \begin{cases} \sqrt{1+2x^2 + \sin^2 x} & x \leq 0 \\ \frac{2+x}{\sqrt[3]{2+e^{-0.5x}}} & x > 0 \end{cases}$$

$$6. f(x) = \begin{cases} |x| & -2 \leq x \leq 1 \\ \sin \frac{\pi}{2} x & 1 < x \leq 2 \\ (2-x)^3 & 2 < x \leq 3 \end{cases}$$

$$7. f(x) = \begin{cases} (x-1)^2 & -2 \leq x \leq 1 \\ \cos \frac{\pi}{2} x & 1 < x \leq 3 \\ 1 - e^{3-x} & 3 < x \leq 8 \end{cases}$$

$$8. f(x) = \begin{cases} \frac{\sqrt{1+|x|}}{2+|x|} & x \leq 0 \\ \frac{1+x}{2+\cos^3 x} & x > 0 \end{cases}$$

$$9. f(x) = \begin{cases} e^{x+1} & -2 \leq x \leq -1 \\ x^2 & -1 < x \leq 1 \\ (2-x)^3 & 1 < x \leq 2 \end{cases}$$

$$10. f(x) = \begin{cases} \frac{1+x}{\sqrt[3]{1+x+x^2}} & x \leq 0 \\ \frac{1+\cos^4 x}{3+x} & x > 0 \end{cases}$$

$$11. f(x) = \begin{cases} \sqrt{x} & 0 \leq x \leq 1 \\ 1 & 1 < x \leq 3 \\ (x-4)^2 & 3 < x \leq 5 \end{cases}$$

$$12. f(x) = \begin{cases} e^x & -2 \leq x \leq -1 \\ \frac{|x|}{e} & -1 < x \leq 1 \\ e^{-x} & 1 < x \leq 2 \end{cases}$$

$$13. f(x) = \begin{cases} \sqrt[3]{1+x^2} & x \leq 0 \\ \sin^2 x + \frac{1+x}{1+e^x} & x > 0 \end{cases}$$

$$14. f(x) = \begin{cases} \arcsin x - 1 & 0 \leq x \leq 1 \\ \frac{\pi}{2} - x & 1 < x \leq \frac{\pi}{2} \\ \cos x & \frac{\pi}{2} < x \leq \pi \end{cases}$$

$$15. f(x) = \begin{cases} x^2 \log_2 x & 1 \leq x \leq 2 \\ \frac{x^3}{2} & 2 < x \leq 3 \\ \frac{x^x}{2} & 3 < x \leq 3.5 \end{cases}$$

2) Написать файл-функцию для решения поставленной задачи.

Варианты:

1. Вычислить произведение элементов вектора, не превосходящих среднее арифметическое значение его элементов.

2. Подсчитать число нулей и единиц в заданной матрице.

3. Определить количество положительных элементов вектора, расположенных между его максимальным и минимальным элементами.

4. Просуммировать отрицательные элементы матрицы, лежащие ниже главной диагонали.

5. Заменить положительные элементы вектора суммой всех его отрицательных элементов.

6. Заполнить квадратную матрицу A , каждый элемент которой a_{ij}

определяется следующим образом:
$$a_{ij} = \begin{cases} i - j, & i > j \\ i + j, & i = j \\ i^2 + j^2, & i < j \end{cases} .$$

7. Для заданного значения x . Вычислить сумму:
$$s(x) = \sum_{i=1}^n \sum_{j=1}^m \frac{x^{i+j}}{(i+j)^2} .$$

8. Для матрицы $A = (a_{ij})$ размера n на m найти значение выражения:

$$w(x) = \sum_{i=1}^n \prod_{j=1}^m a_{ij} .$$

9. По заданному x найти максимальное значение n , для которого сумма

$$s(x) = \sum_{k=1}^n kx^k \text{ не превосходит } 100.$$

10. Для заданного значения x . Вычислить сумму
$$s(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$
 с заданной

точностью ε . Суммировать следует пока модуль отношения текущего слагаемого к уже накопленной части суммы превосходит ε . Сравнить результат с точным значением, построив графики e^x и $s(x)$ для $x \in [0, 5]$.

11. Просуммировать положительные элементы матрицы, лежащие ниже главной диагонали.

12. Заменить отрицательные элементы вектора суммой всех его положительных элементов.

13. Определить количество отрицательных элементов вектора, расположенных между его максимальным и минимальным элементами.

14. Вычислить сумму элементов вектора, не превосходящих среднее арифметическое значение его элементов

15. Заменить положительные элементы вектора числом «7».

1.3.3 Контрольные вопросы

- 1) Операторы ветвления.
- 2) Операторы цикла.
- 3) Какой оператор обеспечивает досрочное завершение цикла?
- 4) Какие операторы сравнения могут применяться в Matlab?
- 5) Назовите логические операции в Matlab.
- 6) Каков приоритет логических и арифметических операторов?
- 7) Какой оператор служит для прекращения текущей итерации в цикле и перехода к следующей?
- 8) Чем отличается подфункция от вложенной функции?
- 9) Какие функции возвращают число входных и выходных аргументов?
- 10) Могут ли операторы отношения использоваться в выражениях, вводимых в командном окне системы MatLab, наряду с арифметическими операторами?

2 Символьные вычисления в среде Matlab

2.1 Лабораторная работа №9. Аналитические преобразования.

Решение уравнений и систем уравнений

2.1.1 Теоретические сведения

Возможности встроенного пакета символьных вычислений и операции Symbolic Math Toolbox достаточно обширны, рассмотрим лишь некоторые его возможности. Для создания символьных переменных используется функция *sym*.

Например,

```
>> x = sym('x')
```

```
x =
```

```
x
```

```
>> a = sym('alpha')
```

```
a =
```

```
alpha
```

```
>> s = sym('5/3')
```

```
s =
```

```
5/3
```

Для создания одновременно нескольких символьных переменных используется команда *syms*:

```
>> syms x y z
```

Символьную функцию можно создать без предварительного объявления переменных при помощи *sym*, входным аргументом которой является строка с выражением, заключенная в апострофы:

```
>> sym('символьное выражение')
```

Например,

```
>> f = sym('a*x^2+b*x+c')
```

```
f =
```

```
a*x^2+b*x+c
```

Запись формулы для выражения в одну строку не всегда удобна, более естественный вид выражения выводит в командное окно функция *pretty* (рисунок 3).

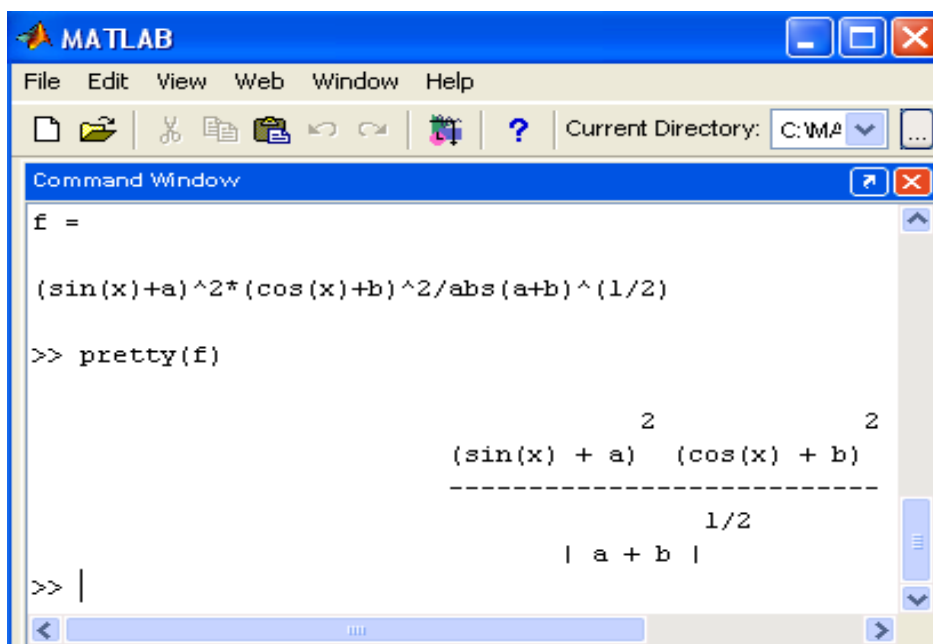


Рисунок 3 – Демонстрация работы функции *pretty*

Символьные выражения и манипуляции над ними

Рассмотрим некоторые простые преобразования, которые можно совершать над символьными объектами (таблица 4).

Таблица 4 – Алгебраические преобразования символьных выражений

Название функции	Назначение
<code>expand</code>	Раскрытие скобок
<code>factor</code>	Разложение на множители, каноническое представление числа
<code>collect</code>	Вычисляет коэффициенты при степенях независимой переменной
<code>subs(<куда>, <вместо чего>, <что>)</code>	Подстановка
<code>simplify</code>	Упрощение выражения
<code>simple</code>	

Рассмотрим примеры использования данных функций.

Функция *factor*:

```
>> factor(sym('45320'))
```

```
ans =
```

```
(2)^3*(5)*(11)*(103)
```

```
>> syms a b
```

```
>> factor(`a^2-b^2`)
```

```
ans =
```

```
(a-b)*(a+b)
```

Функция *expand*:

```
>> sym s
```

```
>> A = s+2;
```

```
>> B = s+3;
```

```
>> C = A*B
```

```
C =
```

```
(s+2)*(s+3)
```

```
>> c = expand(C)
```

```
C =
```

```
s^2+5*s+6
```

Функция *simplify*:

```
>> sym s
```

```
>> H = sym(`(s^3+2*s^2+5*s+10)/(s^2+5)`)
```

```
>> simplify(H)
```

```
H =
```

```
s+2
```

Функции *subs* и *collect*:

```
>> sym s
```

```
>> H = (s+3)/(s^2+6*s+8);
```

```
>> G = subs(H, `s`, `s+2`)
```

```
G =
```

```
(s+5)/((s+2)^2+6*(s+2)+8)
```

```
>> collect(G)
```

$$(s+5)/(s^2+10*s+24)$$

Для нахождения значения символьной функции в точке применяют функцию *subs*:

```
>> E = x^3-14*x^2+65*x-100;
```

```
>> F = subs(E,'x','7.1')
```

```
F =
```

```
13.6710
```

Построение графика символьной функции

Для построения графика символьной функции используется команда *ezplot*. Интервал, на котором изображается график, по умолчанию равен $[-2\pi, 2\pi]$.

```
>> sym x
```

```
>> f = sym('x^3+4*x^2-7*x-10');
```

```
>> ezplot(f,-1,3)
```

Оформление графика может быть произведено обычным образом.

Решение уравнений и систем

Для решения алгебраических уравнений и систем используется команда *solve*:

Например,

Решить уравнение: $x^2 - 4x + 3 = 0$

```
>> x = solve('x^2-4*x+3=0')
```

```
x =
```

```
1
```

```
3
```

Решить систему уравнений:
$$\begin{cases} x^2 + xy + y = 3 \\ x^2 - 4x + 3 = 0 \end{cases}$$

```
>> syms x y
```

```
>> [x,y]=solve('x^2+x*y+y=3','x^2-4*x+3=0')
```

```
x =
```

```
1
```

3

y =

1

-3/2

2.1.2 Практическая часть

Контрольные задания

1) Упростить выражение.

Варианты:

1. $1 + \frac{1 - \cos^2 x + \operatorname{tg}^2 x \cos^2 x}{\sin^2 x}$

2. $\frac{4x(x + \sqrt{x^2 - 1})^2}{(x + \sqrt{x^2 - 1})^4 - 1}$

3. $\frac{2 \sin x - \sin 2x}{2 \sin x + \sin 2x}$

4. $(\sqrt{1 - x^2} + 1) : \left(\frac{1}{\sqrt{1 + x}} + \sqrt{1 + x} \right)$

5. $\frac{2}{\sin x} - \frac{2}{\sin 3x} - \frac{4 \cos 2x}{\sin 3x}$

6. $\frac{\left(m^2 - \frac{1}{n^2}\right)^m \left(n + \frac{1}{m}\right)^{n-m}}{\left(n^2 - \frac{1}{m^2}\right)^n \left(m - \frac{1}{n}\right)^{m-n}}$

7. $\frac{1 + \operatorname{tg} x \operatorname{tg} 2x}{\operatorname{tg} x + \operatorname{ctg} x}$

8. $\frac{a^3 - 2a^2 + 5a + 26}{a^3 - 5a^2 + 17a - 13}$

9. $\frac{2(1 + \sin 2x - \cos 2x)}{\sin x(\sin x + \cos x)}$

10. $\frac{\sqrt{x+1}}{x\sqrt{x}+x+\sqrt{x}} : \frac{1}{x^2-\sqrt{x}}$
11. $\frac{1-\sin^4 2x-\cos^4 2x}{2\sin^4 2x} + 1$
12. $\left(\frac{a\sqrt{a}+b\sqrt{b}}{\sqrt{a}+\sqrt{b}}-\sqrt{ab}\right)\left(\frac{\sqrt{a}+\sqrt{b}}{a-b}\right)^2$
13. $z^{\frac{p-3}{p^2+3p}} : z^{\frac{12}{9-p^2}} \cdot z^{\frac{3}{3p-p^2}}$
14. $\frac{\sqrt{5-2\sqrt{6}}}{(\sqrt[4]{3}+\sqrt[4]{2})(\sqrt[4]{3}-\sqrt[4]{2})}$
15. $\frac{(a-b)^3(\sqrt{a}+\sqrt{b})^{-3}+2a\sqrt{a}+b\sqrt{b}}{a\sqrt{a}+b\sqrt{b}} + \frac{3(\sqrt{ab}-b)}{a-b}$

2) Решить уравнение.

Варианты:

- | | |
|--|--|
| 1. $\frac{x+2}{x+1} + \frac{2-x}{1-x} = \frac{4}{1-x}$ | 8. $\frac{3}{1+x+x^2} = 3-x-x^2$ |
| 2. $\frac{6}{x^2-1} - \frac{2}{x-1} = 2 - \frac{x+4}{x-1}$ | 9. $\frac{x^2-x}{x^2-x+1} - \frac{x^2-x+2}{x^2-x-2} = 1$ |
| 3. $\sqrt{3x-2} = 2\sqrt{x+2} - 2$ | 10. $(x+1)^5 + (x-1)^5 = 32x$ |
| 4. $3\left(x + \frac{1}{x^2}\right) - 7\left(1 + \frac{1}{x}\right) = 0$ | 11. $(x^2 - 6x)^2 - 2(x-3)^2 = 81$ |
| 5. $\frac{(3+x)(2+x)(1+x)}{(3-x)(2-x)(1-x)} = 0$ | 12. $\frac{x^2+1}{x+1} + \frac{x^2+2}{x-2} = -2$ |
| 6. $\frac{x-2}{x-1} + \frac{x+2}{x+1} = \frac{x-4}{x-3} + \frac{x+4}{x+3} - \frac{28}{15}$ | 13. $\sqrt[3]{x+7} - \sqrt{x+3} = 0$ |
| 7. $\frac{x^2+1}{x} + \frac{x}{x^2+1} = 2,9$ | 14. $\sqrt{x-2} + \sqrt{4-x} = x^2 - 6x + 11$ |
| | 15. $(x-2)^6 + (x-4)^6 = 64$ |

3) Решить системы уравнений.

Варианты:

$$1. \begin{cases} x + y = \frac{\pi}{3} \\ \cos(x + y) + \cos(x - y) = \frac{1}{2} \end{cases}$$

$$2. \begin{cases} x^2 + y^2 = 4 \\ x - y = 5 \end{cases}$$

$$3. \begin{cases} 3x + 4 \sin y = -11 \\ -2x + 5 \sin y = \frac{7}{2} \end{cases}$$

$$4. \begin{cases} x^2 + y^2 = 1 \\ x - y = 4 \end{cases}$$

$$5. \begin{cases} \sqrt{x} + \sqrt{y+1} = 1 \\ \sqrt{x+1} + \sqrt{y} = 1 \end{cases}$$

$$6. \begin{cases} x^2 + y^2 = 7 \\ x - y = 9 \end{cases}$$

$$7. \begin{cases} \cos 2y \cdot \sqrt{\sin x} = 0 \\ \cos 2y + 4 \sin^2 x - 3 = 0 \end{cases}$$

$$8. \begin{cases} x^2 + y^2 = 9 \\ x - y = 8 \end{cases}$$

$$9. \begin{cases} \sqrt{\sin x - \cos y} = \cos x \\ \sin x + \cos y = \sin^2 x \end{cases}$$

$$10. \begin{cases} x^2 + y^2 = 5 \\ x - y = 7 \end{cases}$$

$$11. \begin{cases} \cos x + \cos y = 1 \\ 4 \sin x \sin y = 3 \end{cases}$$

$$12. \begin{cases} x^2 + y^2 = 8 \\ x - y = 9 \end{cases}$$

$$13. \begin{cases} (x - y)(x^2 + y^2) = 5 \\ (x + y)(x^2 - y^2) = 9 \end{cases}$$

$$14. \begin{cases} 2^x + 2y = 1 \\ 3y - 6y^2 = 2^{x-1} \end{cases}$$

$$15. \begin{cases} \frac{x}{y} + \frac{y}{z} + \frac{z}{x} = 3 \\ \frac{x}{z} + \frac{y}{x} + \frac{z}{y} = 3 \\ x + y + z = 3 \end{cases}$$

5.1.3 Контрольные вопросы

1) Как объявить символьную переменную?

2) С помощью какой команды можно выполнить упрощение алгебраического выражения?

3) Чем отличаются функции *simple* и *simplify*?

- 4) Что выполняет команда *collect*?
- 5) Как решить уравнение в системе Matlab?
- 6) Как построить символьную функцию?
- 7) С помощью какой команды можно получить каноническое разложение числа?
- 8) Как решить систему уравнений в системе Matlab?
- 9) Как найти значение символьной функции в точке?
- 10) Какая функция выполняет операцию раскрытие скобок?

2.2 Лабораторная работа №10. Дифференцирование и интегрирование

2.2.1 Теоретические сведения

Символьное дифференцирование

Производные от символьных функций вычисляются при помощи функции *diff*:

```
>> syms a x
>> y = sin(a*x)
>> diff(y)          % вычисляем производную по переменной x
ans =
cos(a*x)*a
```

Нахождение производных высших порядков:

```
>> diff(y,2)       % вычисляем производную второго порядка по переменной x
ans =
-sin(a*x)*a^2
```

или

```
>> diff(diff(y))   % вычисляем производную второго порядка по переменной x
ans =
-sin(a*x)*a^2
```

Нахождение частных производных:

```
>> diff(y, a)      % вычисляем частную производную по переменной a
ans =
```



```
cos(a*x)*x
```

```
>> diff(diff(y, a), x)      % вычисляем частную производную  $y''_{ax}$ 
```

```
ans =
```

```
cos(a*x) - sin(a*x)*x*a
```

Символьное интегрирование

Неопределенные интегралы от символьных функций вычисляются при помощи функции *int*, в качестве входных аргументов указываются символьная функция и переменная, по которой происходит интегрирование:

```
>> sym x
```

```
>> int('x^5+sin(4*x)+log(x)')      % вычисляем неопределенный интеграл
```

```
ans =
```

```
x*(log(x) - 1) - cos(4*x)/4 + (x^6)/6
```

```
>> int('x^5 + sin(4*x)', 0, 3)      % вычисляем определенный интеграл
```

```
ans =
```

```
487/4 - cos(12)/4
```

Вычисление пределов

Для вычисления пределов используется функция *limit*:

```
>> sym x
```

```
>> limit((x-2)/(x^2-4), inf)      % вычисляем предел при  $x \rightarrow \infty$ 
```

```
ans =
```

```
0
```

```
>> limit((x-2)/(x^2-4), 2)      % вычисляем предел при  $x \rightarrow 2$ 
```

```
ans =
```

```
1/4
```

Вычисление односторонних пределов:

```
>> limit(x/abs(x), x, 0, 'left')      % вычисляем предел слева
```

```
ans =
```

```
-1
```

```
>> limit(x/abs(x), x, 0, 'right')      % вычисляем предел справа
```

ans =

1

2.2.2 Практическая часть

Контрольные задания

1) Найти производную функции.

Варианты:

$$1. y = \sin \sqrt{3} + \frac{1 \sin^2 3x}{3 \cos 6x}$$

$$2. y = \arcsin 3x^2 + \left(\frac{x^6 - 3}{7x - 2} \right)^3$$

$$3. y = \arctg 2x + \ln \frac{3x^2 - 1}{x^3 + 2}$$

$$4. y = \operatorname{ctg} \sqrt[3]{5} + \frac{1 \cos^2 4x}{8 \sin 8x}$$

$$5. y = \frac{\cos \sin 5 \cdot \sin^2 2x}{2 \cos 4x}$$

$$6. y = \frac{x^2 - 2}{x - 3} + \operatorname{tg}(x - x^2)$$

$$7. y = \frac{\cos \ln 7 \cdot \sin^2 7x}{7 \cos 14x}$$

$$8. y = \cos(\operatorname{ctg} 2) - \frac{1 \cos^2 8x}{16 \sin 16x}$$

$$9. y = \operatorname{ctg}(\cos 2) + \frac{1 \sin^2 6x}{6 \cos 12x}$$

$$10. y = \frac{2 \sin x}{1 - x^2} + (3x^2 - x + 1)^3$$

$$11. y = \cos(x^2 + 5x) + \frac{x + 1}{x^2 + 1}$$

$$12. y = \frac{\sin \cos 3 \cdot \cos^2 2x}{4 \sin 4x}$$

$$13. y = \sin(\ln x) + \frac{x^2}{x - 3}$$

$$14. y = \cos \ln 2 - \frac{1 \cos^2 3x}{3 \sin 6x}$$

$$15. y = \operatorname{tg} \lg \frac{1}{3} + \frac{1 \sin^2 4x}{4 \cos 8x}$$

2) Вычислить неопределенный и определенный интеграл.

Варианты:

$$1. \text{ а) } \int \frac{x \cos x dx}{\sin^3 x}$$

$$\text{ б) } \int_0^2 \frac{x^3}{x^2 + 4} dx$$

$$2. \text{ a) } \int x \sin^2 x dx$$

$$3. \text{ a) } \int \frac{x dx}{\sin^2 x}$$

$$4. \text{ a) } \int \frac{x dx}{\cos^2 x}$$

$$5. \text{ a) } \int (\sqrt{2} - 8x) \sin 3x dx$$

$$6. \text{ a) } \int (7x - 10) \sin 4x dx$$

$$7. \text{ a) } \int (4x + 3) \sin 5x dx$$

$$8. \text{ a) } \int \left(\frac{5}{x} - \sqrt[5]{x^3} \right) dx$$

$$9. \text{ a) } \int \left(\frac{3}{5x} + 4e^x \right) dx$$

$$10. \text{ a) } \int (x^2 + 4^x) dx$$

$$11. \text{ a) } \int \left(\frac{5}{x^2 + 1} + \sin(x) \right) dx$$

$$12. \text{ a) } \int (x^3 - \cos(x)) dx$$

$$13. \text{ a) } \int \left(\frac{4}{3} x^2 + \frac{1}{8x} \right) dx$$

$$14. \text{ a) } \int \left(2x^2 - \frac{3}{\cos^2(x)} \right) dx$$

$$\text{б) } \int_{\pi}^{2\pi} \frac{x + \cos x}{x^2 + 2 \sin x} dx$$

$$\text{б) } \int_0^1 \frac{x}{x^4 + 1} dx$$

$$\text{б) } \int_0^1 \frac{x^3}{x^2 + 1} dx$$

$$\text{б) } \int_1^3 \frac{1 - \sqrt{x}}{\sqrt{x}(x+1)} dx$$

$$\text{б) } \int_{\sqrt{3}}^{\sqrt{8}} \frac{1}{x\sqrt{x^2 + 1}} dx$$

$$\text{б) } \int_1^e \frac{1 + \ln x}{x} dx$$

$$\text{б) } \int_{-2}^2 (2x - x^2) dx$$

$$\text{б) } \int_1^2 (5x - x^2) dx$$

$$\text{б) } \int_0^5 (2x - 6x^2) dx$$

$$\text{б) } \int_{-1}^3 (3x^2 - 5x) dx$$

$$\text{б) } \int_1^3 (x^2 - 2x) dx$$

$$\text{б) } \int_{-2}^4 (2x^2 + 3x) dx$$

$$\text{б) } \int_5^7 (3x^2 + 10x) dx$$

$$15. \text{ a) } \int \left(5x + \frac{2}{\sqrt{1-x^2}} \right) dx$$

$$\text{б) } \int_{-1}^5 (x^2 + 4x) dx$$

3) Вычислить пределы.

Варианты:

$$1. \text{ a) } \lim_{x \rightarrow 0} \frac{\sin 5x}{\sin 3x}$$

$$\text{б) } \lim_{x \rightarrow \infty} \frac{3x^3 - 2x + 5}{8x - 7x^3}$$

$$2. \text{ a) } \lim_{x \rightarrow 1} \frac{\cos \frac{\pi}{2} x}{x - 1}$$

$$\text{б) } \lim_{x \rightarrow \infty} \frac{x^3 - 4x^2 + 1}{x^4 + 2x + 1}$$

$$3. \text{ a) } \lim_{x \rightarrow -2} \frac{x^3 + 3x^2 + 2x}{x^2 - x - 6}$$

$$\text{б) } \lim_{x \rightarrow \infty} \frac{4 - 3x - x^4}{8x^4 - 1}$$

$$4. \text{ a) } \lim_{x \rightarrow 2} \frac{x - 2}{\sqrt{x+3} - \sqrt{7-x}}$$

$$\text{б) } \lim_{x \rightarrow \infty} \frac{3x^3 - 14x + 8}{2x^3 - 4}$$

$$5. \text{ a) } \lim_{x \rightarrow 1} \frac{7x^2 + 26x - 8}{2x^2 + x - 28}$$

$$\text{б) } \lim_{x \rightarrow 0} \frac{\sin 3x - x}{\operatorname{tg} 2x}$$

$$6. \text{ a) } \lim_{x \rightarrow 0} \frac{\sqrt{x+4} - 2}{x}$$

$$\text{б) } \lim_{x \rightarrow 0} \frac{\operatorname{tg} 5x}{\sin 3x}$$

$$7. \text{ a) } \lim_{x \rightarrow \infty} \frac{4x^3 + 3x^2 - 1}{5x^3 + x^4 + 2}$$

$$\text{б) } \lim_{x \rightarrow 0} \frac{2 \sin 3x - x}{x - 3x^2}$$

$$8. \text{ a) } \lim_{x \rightarrow 7} \frac{\sqrt{x-3} + 2}{x^2 + 1}$$

$$\text{б) } \lim_{x \rightarrow 0} \frac{4x^4 - 8x}{3x^3 + 2x + 1}$$

$$9. \text{ a) } \lim_{x \rightarrow 5} \frac{x^2 - 4}{\sqrt{x-1} + 1}$$

$$\text{б) } \lim_{x \rightarrow 0} \frac{\cos 4x - 1}{x^2 - x}$$

$$10. \text{ a) } \lim_{x \rightarrow 3} \frac{2x^2 - 5x - 3}{3x^2 - 4x - 15}$$

$$\text{б) } \lim_{x \rightarrow 0} \frac{\operatorname{ctg} 3x}{\operatorname{ctg} 5x}$$

$$11. \text{ a) } \lim_{x \rightarrow 2} \frac{2^{x-2}}{5x^2 - 1}$$

$$\text{б) } \lim_{x \rightarrow 1} \frac{5x - 4x^2}{3x^3 - 2x + 5}$$

$$12. \text{ a) } \lim_{x \rightarrow 0} \frac{\operatorname{tg} 2x}{\sin 3x}$$

$$\text{б) } \lim_{x \rightarrow 5} \frac{\sqrt{x-1} - \sqrt{9-x}}{x-5}$$

$$13. \quad \text{a) } \lim_{x \rightarrow \infty} \frac{4 - 3x - x^4}{8x^4 - 1}$$

$$\text{б) } \lim_{x \rightarrow \frac{1}{2}} \frac{8x^3 - 1}{6x^2 - 5x + 1}$$

$$14. \quad \text{a) } \lim_{x \rightarrow 0} \frac{1 - \cos 10x}{\sin \frac{x}{2}}$$

$$\text{б) } \lim_{x \rightarrow 3} \frac{\sqrt{x^2 - 5}}{x + 3}$$

$$15. \quad \text{a) } \lim_{x \rightarrow 4} \frac{\ln(x - 3)}{x^2 - x + 5}$$

$$\text{б) } \lim_{x \rightarrow 0} \frac{\sin 10x - \sin 2x}{\text{tg} 3x}$$

5.2.3 Контрольные вопросы

- 1) Как называется пакет расширения системы MatLab для работы с символьными выражениями?
- 2) Какая функция служит для вычисления неопределенного интеграла в символьном виде?
- 3) Какая функция служит для вычисления определенного интеграла?
- 4) Какая функция служит для вычисления пределов?
- 5) Что необходимо сделать для вычисления производных?
- 6) Как вычислить левосторонний предел?
- 7) Как вычислить правосторонний предел?
- 8) Как найти частную производную от символьной функции?
- 9) Как вычислить производные высших порядков?
- 10) Объектами какого класса являются символьные переменные?

2.3 Лабораторная работа №11. Исследование функции

2.3.1 Теоретические сведения

План исследования функции:

- 1) Найти область определения функции и точки разрыва функции.
- 2) Точки пересечения с осями координат.
- 3) Найти асимптоты графика функции (создать файл-функцию или скрипт для вычисления и построения асимптот):

а) вертикальные асимптоты: $x = a$, если есть точки разрыва.

б) наклонные асимптоты $y = kx + b$, где

$$k = \lim_{x \rightarrow \infty} \frac{f(x)}{x}, \quad b = \lim_{x \rightarrow \infty} (f(x) - kx).$$

Если $k = 0$, и b конечное число, то $y = b$ горизонтальная асимптота.

4) Найти точки экстремума. Промежутки монотонности функции.

5) Найти точки перегиба. Промежутки выпуклости и вогнутости графика.

6) Построить график.

2.3.2 Практическая часть

Контрольные задания

1) Провести полное исследование функции и построить график.

На графике должно быть отражено:

– график функции,

– асимптоты пунктирными линиями,

– точки экстремума и точки перегиба,

– подписаны оси, график, точки экстремума и перегиба.

Варианты:

1. $f(x) = \frac{x^3}{x^2 - 1}$

2. $f(x) = \frac{x^3}{3 - x^2}$

3. $f(x) = \frac{x^3}{2(x+1)^2}$

4. $f(x) = \frac{x^4}{x^3 - 1}$

5. $y = \frac{x}{1 - x^2}$

6. $y = \frac{x^2 + 1}{x^2 - 1}$

7. $y = \frac{x^2 + 2x - 1}{x - 1}$

8. $y = \frac{9x}{x^2 - 9}$

9. $y = \frac{(x-1)^2}{x^2 + 2x}$

10. $y = \frac{2x^2 + 4x}{4x - 1}$

11. $y = \frac{x^2 + x}{x - 1}$

12. $y = \frac{x^2 - 1}{x^2 + 1}$

14. $y = \frac{x}{x^2 + 1}$

13. $y = \frac{x}{16 - x^2}$

15. $y = \frac{4 - x^2}{1 + x^2}$

2) Вычислить площадь фигуры, ограниченной линиями. Построить фигуру. Точки пересечения графиков функций найдите с помощью решения уравнения в математическом пакете.

Варианты:

1. $y = 4x - x^2, x + y = 4$

2. $y = x^2 - 2x - 4, y = 12 + 2x - x^2$

3. $y = 16/x^2, y = 2x, x = 4$

4. $y = \sqrt{x}, x + y = 2, x + 3y = 0$

5. $y^3 = x, y = 1, x + 8 = 0$

6. $y = 2x - x^2, x + y = 0$

7. $y(x^2 + 1) = 1, y = x^2/2$

8. $y = 4 - x^2, y = 0$

9. $y = \ln x, y = 0, x = e$

10. $y = x^2, y = 2 - x^2$

11. $xy = 4, y = 0, x = 0, x = 4, y = 4$

12. $y = x^2, y = \sqrt{x}$

13. $y = x^3, y = 1, x = 0$

14. $y = x^2 + 1, y = 0, x = 1, x = 2$

15. $y = e^x, y = 0, x = 0, x = 1$

2.3.3 Контрольные вопросы

- 1) Как найти точки разрыва функции?
- 2) Как построить вертикальные асимптоты?

- 3) Как построить горизонтальные асимптоты?
- 4) Как найти точки экстремума функции?
- 5) Как найти точки перегиба функции?
- 6) Как отметить на графике точки экстремума и точки перегиба?
- 7) Как найти наклонную асимптоту?
- 8) Как построить наклонную асимптоту?
- 9) Как найти точки пересечения графиков функций?
- 10) Как перевести символьные значения в числовые?

Список использованных источников

- 1 Ануфриев, И.Е. Самоучитель MatLab 5.3/6.x / И.Е. Ануфриев. – СПб: БХВ-Петербург, 2002. – 736 с.
- 2 Гультаев, А.П. Визуальное моделирование в среде MATLAB: учебный курс/ А.П. Гультаев. – СПб.: Питер, 2000.– 432 с.
- 3 Дьяконов, В.П. Matlab: учебный курс / В.П. Дьяконов. – СПб.: Питер, 2001. – 592 с.
- 4 Дьяконов, В.П. Matlab 6.5: Основы применения / В.П. Дьяконов. – М.: СОЛН-Пресс, 2005
- 5 Иглин, С.П. Математические расчеты на базе MATLAB/ С.П. Иглин. – СПб.: БХВ-Петербург, 2005. – 640 с.
- 6 Колокольникова, А.И. Спецразделы информатики: введение в MatLab: учебное пособие [Электронный ресурс] / А.И. Колокольникова, А.Г. Киренберг. – М., Берлин: Директ-Медиа, 2014. – 73 с. – Режим доступа: http://biblioclub.ru/index.php?page=book_view&book_id=275268
- 7 Конев, В.Ю. Основные функции пакета MATLAB: учебное пособие/ В.Ю. Конев, Л.А. Мироновский. – 2-ое издание. – СПб.: ГААПСПб., 1994. – 76 с.
- 8 Кривилев, А.В. Основы компьютерной математики с использованием системы MATLAB: учебное пособие / А.В. Кривилев. – М.: Лекс-Книга, 2005. – 496 с.
- 9 Потемкин, В.Г. MATLAB 6: среда проектирования инженерных приложений/ В.Г. Потемкин. – М.: Диалог-МИФИ, 2003. – 448 с.
- 10 Потемкин, В. Г. Система MATLAB: справочное пособие/ В. Г. Потемкин. – М.: ДИАЛОГ-МИФИ, 1997.– 350 с.