

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕХНОЛОГИИ И ДИЗАЙНА»

Кафедра машиноведения

СИСТЕМЫ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ

(задания к контрольной работе)

методические указания по дисциплине
«Системы компьютерной математики»
для студентов направления
15.03.02 – Технологические машины и оборудование
заочной формы обучения

Составители:
А. В. Марковец
И. М. Беспалова
Е. В. Анашкина

Санкт-Петербург
2015

СОДЕРЖАНИЕ

Введение	4
Содержание дисциплины	4
Список литературы и другие информационные источники	5
Порядок выполнения	6
Содержание отчета о выполнении контрольной работы	7
Расчетные задания к контрольной работе	7
Краткие сведения о среде MATLAB	10
Общие сведения о системах компьютерной математики	10
Рабочая среда системы MATLAB	12
Использование переменных в среде MATLAB	18
Использование математических функций в среде MATLAB.....	21
Массивы в среде MATLAB.....	25
Вычисления с применением массивов	32
Построение графиков в среде MATLAB (краткие сведения)	36

Введение

Целью дисциплины “Системы компьютерной математики” является формирование компетенций в области применения современных ЭВМ для проведения математических вычислений с использованием современного программного обеспечения.

При изучении данной дисциплины ставятся следующие задачи:

- рассмотреть возможности и области применения современных программных систем для выполнения математических расчетов на ЭВМ;
- раскрыть принципы работы в системах компьютерной математики при проведении вычислений, подготовке и оформлению результатов расчета;
- показать особенности выполнения вычислений со скалярными величинами и матрицами (массивами) в системах компьютерной математики;
- предоставить обучающимся возможности для формирования умений и навыков работы в системах компьютерной математики на примере системы инженерных и научных расчетов MATLAB.

Содержание дисциплины

Учебный модуль 1. Математическое обеспечение и системы компьютерной математики

Тема 1.1. Обзор систем компьютерной математики. Назначение. Области применения, возможности и особенности работы.

Тема 1.2. Пользовательский интерфейс системы MATLAB. Основные приемы работы в командной строке: ввод команд и выражений, создание, доступ и изменение значений переменных.

Тема 1.3. Использование математических операторов и функций для выполнения вычислений с действительными и комплексными числами.

Учебный модуль 2. Векторы и матрицы

Тема 2.1. Формирование одномерных (векторы) и двумерных (матрицы) числовых массивов в среде MATLAB.

Тема 2.2. Редактирование данных в массивах.

Тема 2.3. Особенности вычисления с массивами в среде MATLAB: матричные и поэлементные операции с массивами.

Учебный модуль 3. Визуализация вычислений в среде MATLAB

Тема 3.1. Построение графиков функций одной переменной. Приемы оформления графиков.

Тема 3.2. Построение графиков функций двух переменных, поверхностей. Приемы оформления

Тема 3.3. Построение специальных видов графиков (диаграммы, графики в полярных системах координат, графики векторов, траекторий, годографы и т.п.).

Учебный модуль 4. Разработка программ в среде MATLAB

Тема 4.1. Понятие об m-файлах: сценарии и функции. Приемы разработки сценариев и функций в среде MATLAB. Локальные, глобальные переменные.

Тема 4.2. Разработка программ с использованием условных операторов, операторов цикла, ветвления

Тема 4.3. Использование операций ввода-вывода в программах. Функции работы с файлами.

Список литературы и другие информационные источники

а) основная учебная литература

1 Поршневу, С. В. MATLAB 7. Основы работы и программирования: учебник / С. В. Поршневу. – М.: ООО «Бином-Пресс», 2006. – 320 с.

2 Дьяконов, В. П. MATLAB: учебный курс / В. П. Дьяконов. – СПб: Питер, 2001.

3 Марковец, А. В. Лабораторный практикум № 141 по курсу «Вычислительные методы для инженеров». Ч. 1 / А. В. Марковец, И. М. Беспалова. – СПб.: СПГУТД, 2007. – 84 с.

б) дополнительная литература и другие информационные источники

4 Кетков, Ю. Л. MATLAB 7: программирование, численные методы / Ю. Л. Кетков, А. Ю. Кетков, М. М. Шульц. – СПб.: БХВ-Петербург, 2005. – 752 с.

5 Ануфриев, И. Е. MATLAB 7 / И. Е. Ануфриев, А. Б. Смирнов, Е. Н. Смирнова. – СПб.: БХВ-Петербург, 2005. – 1104 с.

6 www.exponenta.ru – образовательный математический сайт.

Порядок выполнения

Контрольная работа состоит из теоретической и расчетной частей. Перед выполнением контрольной работы необходимо изучить теоретические основы дисциплины, используя приведенный ниже список тем (в квадратных скобках указаны ссылки на литературу).

Темы для изучения

1. Система инженерных и научных расчетов MATLAB.

1.1. Графический интерфейс пользователя и простейшие вычисления в системе MATLAB. Командное окно MATLAB. Рабочее пространство системы MATLAB. Формат представления вещественных чисел. Комплексные числа. Элементарные функции. Специальные символы, переменные и константы.

Литература: [1, § 1.1–1.4, 1.6]; [2, с. 126].

1.2. Числовые массивы в MATLAB. Формирование одномерных числовых массивов. Двумерные массивы чисел: векторы и матрицы. Особенности вычислений с массивами. Операции отношения и логические операции над числами.

Литература: [1, § 1.5, 3.1–3.4]; [2, с. 294].

1.3. Визуализация вычислений в среде MATLAB. Построение графиков функций, зависящих от одной переменной. Оформление графиков и графических окон. Трехмерная графика. Специальная графика.

Литература: [1, § 2.1–2.3, § 4.1–4.3]; [2, с. 157].

1.4. Программирование в системе MATLAB. М-файлы: сценарии и функции. Синтаксис определения и вызова М-функций – входные и выходные параметры функции. Конструкции управления, операторы цикла. Локальные, глобальные и статические переменные. Функции ввода-вывода. Операции записи и чтения файлов.

Литература: [1, § 7.1–7.6]; [2, с. 467].

Содержание отчета о выполнении контрольной работы

Отчет о выполнении контрольной работы должен содержать:

- Основные теоретические сведения по изучаемым темам.
- Результаты выполнения расчетных заданий.

Расчетные задания к контрольной работе

Номер варианта соответствует последней цифре номера зачетной книжки (кроме заданий 1 и 2, которые выполняются для всех вариантов).

1. Описать назначение следующих команд пакета MATLAB:

`clc`, `clear`, `who`, `whos`, `load`, `save`, `cd`, `help`, `format`

(выполняют все студенты).

2. Заполнить табл. 1 (выполняют все студенты).

Таблица 1

Написание функции		Название функции
математическое	в пакете MATLAB	
Тригонометрические функции		
$\sin(x)$		
$\cos(x)$		
$\operatorname{tg}(x)$		

Написание функции		Название функции
математическое	в пакете MATLAB	
$\operatorname{ctg}(x)$		
Обратные тригонометрические функции		
$\arcsin(x)$		
$\arccos(x)$		
$\operatorname{arctg}(x)$		
$\operatorname{arcctg}(x)$		
Гиперболические функции		
$\operatorname{sh}(x)$		
$\operatorname{ch}(x)$		
$\operatorname{th}(x)$		
$\operatorname{cth}(x)$		
Обратные гиперболические функции		
$\operatorname{arsh}(x)$		
$\operatorname{arch}(x)$		
$\operatorname{arth}(x)$		
$\operatorname{arcth}(x)$		
Другие математические функции		
e^x		
$\ln(x)$		
$\lg(x)$		
\sqrt{x}		
a^x		
$ x $		
$\operatorname{sign}(x)$		

3. Записать последовательность команд и выражений для построения в системе MATLAB графика функции. График функции построить линией черного цвета, толщиной 2. Нанести на график сетку и название осей.

Номер варианта	Функция	Номер варианта	Функция
0	$y = x^2 - \cos(x)$	5	$y = x \cdot \ln x^2 - e^{-x^2}$
1	$y = \frac{1}{1+x^2} - \frac{e^2}{2}$	6	$y = \frac{x}{2 + \sin(x)} - e^{-x^2}$
2	$y = x^3 + x^2 - 11$	7	$y = x^3 + 3e^{-x} \sin(x) + 5$

Номер варианта	Функция	Номер варианта	Функция
3	$y = \frac{\cos(x)}{2+x} - \ln x + e^{-x}$	8	$y = \sin^2(x) + 5e^{-x^2} - 18$
4	$y = 0,5x^4 + x^2 \ln x $	9	$y = \ln x - \cos^2(2x)$

4. При проведении эксперимента задавалась величина x и определялась координата y . Для определения координаты y было произведено пять опытов. Результаты эксперимента представлены в таблице. Пользуясь возможностями пакета MATLAB, определить максимальное y_{\max} , минимальное y_{\min} , среднее \bar{y} и среднеквадратическое отклонение σ величины y в каждом опыте. Построить на графике зависимость \bar{y} от x . Значения \bar{y} на графике отобразить точками черного цвета, не соединяя их линиями.

$x, \text{мм}$	$y, \text{мм}^*$				
	Опыт 1	Опыт 2	Опыт 3	Опыт 4	Опыт 5
5	0,139	0,142	0,141	0,143	0,140
10	0,299	0,310	0,311	0,298	0,297
15	0,238	0,238	0,245	0,237	0,237
20	-0,112	-0,104	-0,103	-0,104	-0,115
25	-0,804	-0,791	-0,803	-0,790	-0,791

* Выполняют все студенты.

5. Найти в системе MATLAB решение двух систем линейных уравнений.

Номер варианта	Система уравнений 1	Система уравнений 2
0	$\begin{cases} 1,5x_1 - 0,8x_2 + 4,25x_3 = 5,1 \\ 1,2x_1 + 7,18x_2 - 3,2x_3 = 4,2 \\ 0,5x_1 - 1,5x_2 + 7,1x_3 = -1,2 \end{cases}$	$\begin{cases} 5x_1 + x_3 = 11 \\ x_1 + 3x_2 - x_3 = 4 \\ -3x_1 + 2x_2 + 10x_3 = 6 \end{cases}$
1	$\begin{cases} 6,7x_1 - 0,6x_2 + 0,83x_3 = 6,8 \\ 0,8x_1 + 1,1x_2 - 7,2x_3 = 5,2 \\ 1,2x_1 - 5,4x_2 + 0,54x_3 = -3,2 \end{cases}$	$\begin{cases} 2x_1 - x_3 = -3 \\ -x_1 + 3x_2 + x_3 = 2 \\ x_1 - x_2 + 4x_3 = 3 \end{cases}$
2	$\begin{cases} -1,32x_1 + 2,15x_2 + 7,6x_3 = -1,4 \\ 2,62x_1 + 6,1x_2 - 4,12x_3 = 5,6 \\ 8,3x_1 - 2,84x_2 - 1,5x_3 = -6,5 \end{cases}$	$\begin{cases} 2x_1 - x_3 = 1 \\ x_1 - 3x_2 + x_3 = 2 \\ x_1 + x_2 + 3x_3 = 4 \end{cases}$

Номер варианта	Система уравнений 1	Система уравнений 2
3	$\begin{cases} 0,51x_1 - 10,2x_2 - 3,62x_3 = -2,05 \\ 3,09x_1 + 1,23x_2 - 4,64x_3 = -5,6 \\ 3,2x_1 - 2,31x_2 - 8,4x_3 = 6,1 \end{cases}$	$\begin{cases} 2x_1 - x_2 = -2 \\ 2x_1 + 5x_2 - 2x_3 = -4 \\ x_1 - x_2 + 3x_3 = 2 \end{cases}$
4	$\begin{cases} 7,12x_1 - 6,66x_2 + 2,6x_3 = -3,1 \\ -1,76x_1 + 6,5x_2 - 0,87x_3 = 2,85 \\ 0,65x_1 + 0,87x_2 - 8,7x_3 = 5,56 \end{cases}$	$\begin{cases} 3x_1 - x_2 + x_3 = 1 \\ 2x_2 - x_3 = 3 \\ -x_1 + x_2 + 5x_3 = -5 \end{cases}$
5	$\begin{cases} 6,4x_1 - 0,73x_2 + 2,1x_3 = 3,8 \\ -1,07x_1 + 3,8x_2 - 1,5x_3 = -1,2 \\ 2,7x_1 - 3,1x_2 + 4,2x_3 = -7,5 \end{cases}$	$\begin{cases} 4x_1 + x_2 - x_3 = 7 \\ 2x_1 + 3x_2 = 7 \\ x_1 - x_2 + 5x_3 = 11 \end{cases}$
6	$\begin{cases} 9,21x_1 - 1,84x_2 + 0,7x_3 = -3,2 \\ -6,17x_1 + 8,5x_2 - 2,87x_3 = -3,75 \\ 0,7x_1 + 0,87x_2 - 8,7x_3 = 2,64 \end{cases}$	$\begin{cases} 2x_1 - x_3 = 1 \\ x_1 - 4x_2 + 2x_3 = -5 \\ x_1 + x_2 + 3x_3 = 6 \end{cases}$
7	$\begin{cases} 4,3x_1 - 1,2x_2 + 10,3x_3 = 4,2 \\ 0,21x_1 + 6,2x_2 + 3,54x_3 = 5,1 \\ -0,31x_1 - 0,52x_2 + 3,6x_3 = -2,1 \end{cases}$	$\begin{cases} 2x_1 - x_2 = 3 \\ 5x_2 + 2x_3 = 7 \\ x_1 - x_2 + 3x_3 = 4 \end{cases}$
8	$\begin{cases} 6,9x_1 + 2,3x_2 + 1,21x_3 = 3,1 \\ -x_1 + 2,3x_2 - 3,4x_3 = -2,3 \\ 0,21x_1 + 0,43x_2 - 6,3x_3 = 3,6 \end{cases}$	$\begin{cases} 3x_1 + x_2 - x_3 = -2 \\ x_1 + 5x_2 - x_3 = 8 \\ 2x_1 + 3x_3 = 1 \end{cases}$
9	$\begin{cases} 12,4x_1 - 0,56x_2 + 4,2x_3 = 6,3 \\ -0,56x_1 + 4,4x_2 + 1,5x_3 = 1,5 \\ 1,5x_1 + 2,1x_2 - 2,8x_3 = 1,7 \end{cases}$	$\begin{cases} 3x_1 - x_3 = 7 \\ 2x_1 - 5x_2 + x_3 = -2 \\ 2x_1 + 2x_2 + 5x_3 = 1 \end{cases}$

Краткие сведения о среде MATLAB

Общие сведения о системах компьютерной математики

В практике инженерной деятельности часто возникают задачи связанные с выполнением математических расчетов. Сложность производимых вычислений, как правило, определяется сложностью решаемой задачи. В ряде простых случаев требуется выполнение расчетов по заданным точным аналитическим зависимостям (прямые вычисления). В более сложных случаях возникает необходимость выполнения многократно повторяющихся вычислений, расчетов по

приближенным зависимостям с оценкой точности. Применение ЭВМ позволяет автоматизировать выполнение математических операций, ускорить расчеты и увеличить точность вычислений.

Как правило, выполнение математических вычислений с помощью ЭВМ предполагает этапы разработки и отладки программы. Указанные этапы, особенно в случае решения сложных задач являются достаточно трудоемкими, и требует от инженера серьезных знаний, навыков и опыта в области программирования.

Другим инструментом для решения вычислительных задач на ЭВМ являются так называемые системы компьютерной математики (СКМ). Эти системы позволяют облегчить процесс решения вычислительных задач, оформления результатов в виде удобном для дальнейшего анализа (построение графиков, диаграмм, таблиц и т.п.). Как правило, СКМ имеют интерфейс, позволяющий вводить операторы, математические выражения, команды в виде более привычном и известном нам по курсу математики.

На рынке современных математических систем в настоящее время присутствует целый ряд крупных фирм: Macsyma, Inc., Waterloo Maple Software, Inc., Wolfram Research, Inc., MathWorks, Inc., MathSoft, Inc., SciFace GmbH и др. К разработке каждой такой математической системы привлекаются сотни профессионалов из известных университетов и крупных научных центров, а также высококвалифицированные программисты и эксперты в области проектирования сложных программных систем. Ими разработано ряд программных продуктов, включающие существенные математические понятия и обладающие богатым набором методов для решения общих и специальных математических и научно-технических задач.

Характерными представителями СКМ являются: MATLAB, MATHCAD, Maple, Mathematica, Macsyma, MuPAD, S-PLUS и др. По способу взаимодействия с пользователем эти системы условно можно разделить на две группы. К первой группе можно отнести СКМ, в которых вычисления выполняются подобно тому, как это делается и в языках программирования высокого уровня, но

с тем отличием, что имеют значительное количество встроенных и хорошо документированных математических библиотек, графический интерфейс для построения различных графиков. Примером такой СКМ является MATLAB. К другой группе СКМ следует отнести программы, в которых запись математических выражений, операторов и функций представлена в виде, привычном для математиков. В таких системах за счет использования панелей инструментов имеется возможность вводить греческие символы, символы математических операций (интегралов, производных, деления, умножения, возведения в степень и др). Примером такой СКМ является MATHCAD.

Отличительной чертой некоторых СКМ является возможность проводить вычисления в символьном виде. Это позволяет получать результат не в числовой, а в аналитической форме, упрощать математические выражения. Характерным представителем такой СКМ является программа Maple.

В дальнейшем основное внимание уделено изучению СКМ MATLAB и на примере ее изучения основных вычислительных алгоритмов, используемых в практике инженерных расчетов. Система MATLAB (сокращение от MATrix LABoratory – матричная лаборатория) представляет собой интерактивную компьютерную систему для выполнения инженерных и научных расчетов, ориентированную на работу с массивами данных. MATLAB – это чрезвычайно развитая система, которая содержит большое количество процедур и функций, необходимых инженеру и научному работнику для осуществления сложных численных расчетов, моделирования поведения технических и физических систем, оформления результатов этих расчетов в наглядном виде.

Рабочая среда системы MATLAB

При запуске MATLAB на экране открывается рабочая среда MATLAB, изображенная на *рис. 1*.

Основными элементами рабочей среды являются:

- строка меню;
- панель инструментов с кнопками и раскрывающимся списком;

- окна **Workspace** и **Current Directory** для просмотра переменных и содержимого текущего каталога (текущей папки);

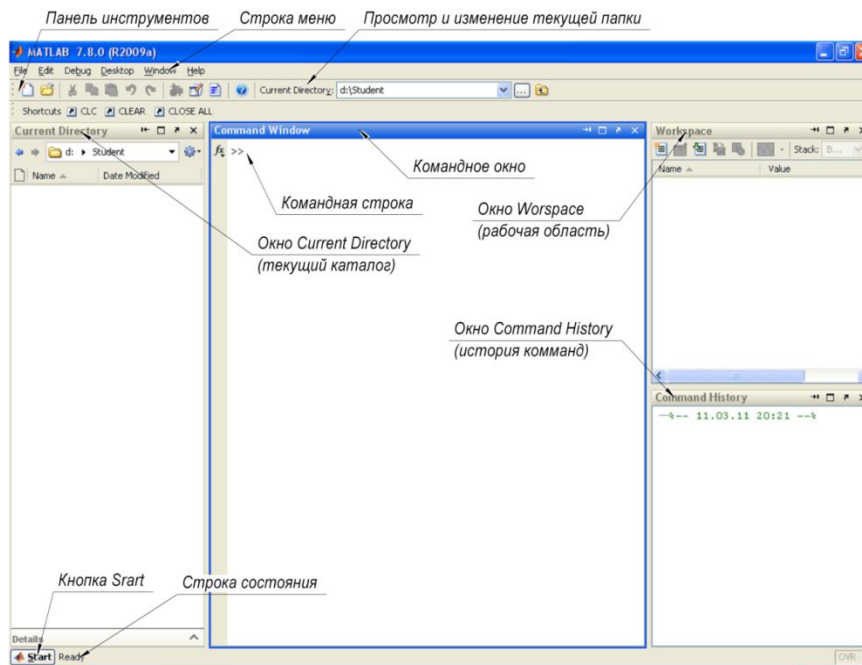


Рис. 1. Рабочая среда MATLAB

- окно **Command Window**, служащее для ввода команд и вывода результата;

- окно **Command History**, предназначенное для просмотра и повторного выполнения ранее введенных команд (окно **Command History** может быть не пустым, если до этого пакет **MATLAB** использовался);

- строка состояния с кнопкой **Start**, открывающей меню доступа ко всем основным средствам **MATLAB**.

- Окно **Command Window** состоит из следующих элементов:

- заголовка с названием окна и кнопками справа;

- рабочей области с командной строкой, в которой находится мигающий вертикальный курсор;

- полос скроллинга.

Все команды следует набирать в командной строке после символа **>>**. Набор команды или выражения должен заканчиваться нажатием на **<Enter>**, для того чтобы программа **MATLAB** выполнила эту команду или вычислила

выражение. В командной строке можно выполнять простейшие вычисления. Например, набрав $1+2$ и нажав **<Enter>**, в командном окне **MATLAB** можно получить следующий результат:

```
>> 1+2
ans =
     3
>>
```

Таким образом, программа **MATLAB** вычислила сумму. Так как мы не указали системе, куда следует сохранить результат вычисления, то **MATLAB** записал результат в специальную переменную **ans** и вывел ее значение, равное **3**, в командное окно. Переменная **ans**, если она не была создана ранее, всегда создается автоматически, когда пользователь не указал явно имя переменной куда следует записать результат вычисляемого выражения или выполняемой команды.

В окне **Command Window** ниже ответа (рис. 2) расположена командная строка с мигающим курсором, обозначающая, что среда **MATLAB** готова к дальнейшим вычислениям.

Если требуется продолжить работу с предыдущим выражением, например, вычислить $(1+2)/4.5$, то проще всего воспользоваться уже имеющимся результатом, который хранится в переменной **ans**, например:

```
>>ans/4.5
ans =
    0.6667
```

Как видно из представленного примера, значение переменной **ans** изменилось.

Рассмотрим следующий пример. Пусть требуется вычислить значение следующего математического выражения:

$$\frac{\left(15\frac{1}{7} - 3\frac{1}{12} + 1,45\right) \cdot 11,73 - 6,051}{3,02 \cdot 8,47}$$

При вычислении данного выражения необходимо учитывать, что запись $15\frac{1}{7}$ следует читать как “*пятнадцать и одна седьмая*”, что не соответствует операции умножения числа 15 на дробь $\frac{1}{7}$, т. е. $15\frac{1}{7} \neq 15 \cdot \frac{1}{7}$. В данном случае выражение $15\frac{1}{7}$ эквивалентно следующему математическому выражению

$$15\frac{1}{7} = 15 + \frac{1}{7}.$$

С учетом данного замечания, рассматриваемое математическое выражение в командном окне среды MATLAB можно записать в следующем виде:

```
>>( ( 15+1/7 - ( 3+1/12 ) + 1.45 ) * 11.73 - 6.051 ) / ( 3.02 * 8.47 )
```

По окончании ввода этого выражения следует нажать клавишу <Enter>.

Результат вычислений представлен на *рис. 2*.

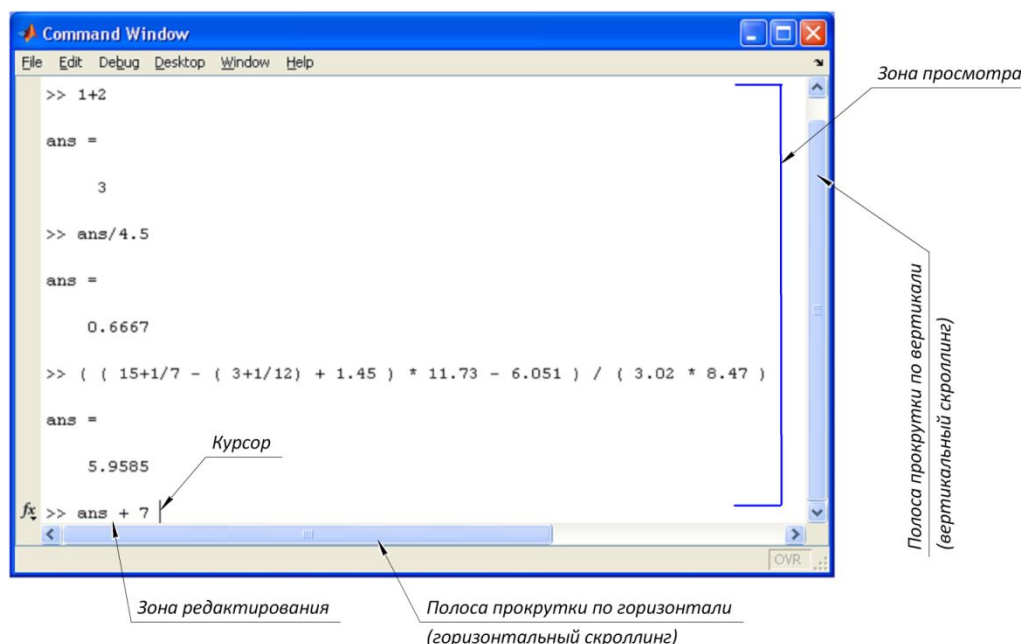


Рис. 2. Окно Command Window в процессе выполнения вычислений

Как видно из *рис. 2*, вся информация, размещенная в командном окне, располагается в двух зонах: *зоне просмотра* и *зоне редактирования*. Информацию (команды, выражения), размещенную в *зоне просмотра*, исправить (отредактировать) *нельзя*. В любую строку *зоны просмотра* с помощью мыши может

быть помещен курсор, однако при попытке ввода данных с клавиатуры курсор автоматически будет перемещен в зону редактирования. В зоне просмотра с помощью мыши можно лишь выделять любую информацию и копировать ее в буфер обмена операционной системы Windows. После этого содержимое буфера обмена может быть вставлено в строку редактирования, в документ Microsoft Word или другую программу, допускающую работу с буфером обмена Windows.

Расположенные на клавиатуре клавиши “↑” (стрелка вверх) и “↓” (стрелка вниз), используемые в других программах, как правило, для управления движением курсора вверх-вниз, в системе MATLAB работают иначе. Эти клавиши позволяют вернуть в строку ввода (зону редактирования) введенные ранее с клавиатуры команды и другую информацию. Введенные при этом команды и выражения могут быть отредактированы и запущены на выполнение (после нажатия клавиши <Enter>).

Вводимая в командную строку информация запоминается системой MATLAB в специальной области памяти, которую называют *стеком команд*. Содержимое этой области памяти отображается, также, в окне Command History. Перемещение по вертикали в зоне просмотра возможно с использованием полосы вертикальной прокрутки (полосы скроллинга), а также, с помощью клавиш <PageUp> и <PageDown>.

Основным типом данных, с которым производятся вычисления в среде MATLAB являются *вещественные числа*. Этот основной тип данных называется double. Вещественные числа задаются в системе MATLAB с использованием *экспоненциальной формы записи*. Эта форма записи представляет действительные числа в виде *мантиссы* и *порядка*. Она удобна при представлении очень больших и очень маленьких чисел, а также, для унификации их написания. Экспоненциальный формат записи вещественных чисел имеет вид

$$N = M \cdot n^p$$

где N – записываемое число, M – мантисса, $n = 10$ – основание степени, p – порядок числа (целое).

Например, запись

$$-2.851038547e + 012$$

обозначает следующее:

$$\underbrace{-}_{\text{знак числа}} \underbrace{2.851038547}_{\text{мантисса}} \underbrace{e}_{\text{обозначение основания степени, равно } 10} \underbrace{+}_{\text{знак порядка}} \underbrace{012}_{\text{порядок}}$$

Таким образом, записано число: $-2.851038547 \cdot 10^{12}$

В среде **MATLAB** вещественные числа представлены типом данных, который называется **double**. Под мантиссу и показатель степени (на машинном уровне используется двоичная система записи) отводится 8 байт памяти. В результате для десятичных чисел достигается точность порядка 15 значащих цифр.

Следует иметь в виду, что при записи вещественных чисел в MATLAB для отделения целой части числа от дробной используется не запятая (“,”), как это принято в математике, а десятичная точка (“.”).

При выводе значений вычисленных выражений в командном окне по умолчанию используется формат **short**, при котором показывается только четыре десятичные цифры после запятой. Например:

```
>> 100/3
ans =
    33.3333
```

Если при отображении слишком большого или малого числа результат не укладывается в формат **short**, тогда результат выводится в экспоненциальной форме:

```
>> 100000/3
ans =
    3.3333e+004
```

Первоначальная установка формата сохраняется и при дальнейших вычислениях – для небольших чисел вывод результата снова будет происходить в формате **short**.

Короткий формат **short** вывода результатов вычислений, установленный в **MATLAB** по умолчанию, может быть изменен пользователем. Задавать формат вывода можно непосредственно из командной строки при помощи команды

`format`. Например, для установки длинного с плавающей точкой формата вывода результатов вычислений следует ввести команду `format long e` в командной строке:

```
>> format long e
>> 1.25/3.11
ans =
    4 .019292604501608e-001
```

Независимо от того, какой формат вывода установлен вычисления MATLAB производить с *двойной точностью*.

Использование переменных в среде MATLAB

Для организации вычислений в среде MATLAB допускается использование переменных. Объявлять используемые переменные заранее, как это требуется в ряде других языков программирования, не требуется. По умолчанию (если пользователь не указал явно) все переменные относятся к вещественным типа `double`. Имена переменных должны начинаться с буквы и состоять из букв, цифр и знаков подчеркивания. Например:

```
x, y, x1, y1, x_1, y_1, Rab, Lab, R_ab, L_ab.
```

Недопустимым является использование имен переменных, содержащих буквы кириллицы, использование в именах переменных пробелов, знаков математических операций, недопустима ситуация когда имя переменной начинается с цифры.

Имя переменной может содержать до 30 символов и не должно совпадать с именами функций, команд среды MATLAB и системных переменных. При этом **различается регистр символов** в именах переменных (большие и маленькие буквы). Так, например, следующие имена

```
amenu, Amenu, aMenu
```

в среде MATLAB обозначают разные переменные.

Знак равенства (“=”) закреплен за операцией присваивания, т. е. выражение вида

```
>> x = 10
```

означает “присвоить переменной с именем X значение равно 10”.

При задании значений переменным допустимо использовать экспоненциальную форму записи. В этом случае важно помнить, что символ **e** обозначает не число **e** (основание натурального логарифма), а основание степени, равное 10. В следующем примере создается переменная **z** со значением $-7.5 \cdot 10^2$:

```
>> z = -7.5e+2
```

Ввод команд и математических выражений в командном окне должен завершаться нажатием клавиши «**Enter**». Символ «**;**» служит для отделения команд и выражений друг от друга когда они вводятся в одной строке, а также, для предотвращения вывода результата вычислений на экран в командное окно.

Рассмотрим следующий пример:

```
>> x = 10; y = 25.8;  
>> R = 0.5*(x^2 + y^2)  
R =  
 382.8200  
>>
```

В данном случае в первой строке выполняются две операции – переменным **x** и **y** присваиваются соответствующие значения. Так как две операции записаны в одной строке то они отделяются друг от друга символом “**;**”. Этот же символ предотвращает вывод результата операции в командное окно после нажатия клавиши **Enter**. Во второй строке вычисляется следующее математическое выражение

$$R = \frac{1}{2}(x^2 + y^2)$$

Символ “**^**” используется для записи операции возведения числа в степень. В конце данного выражения отсутствует точка с запятой и именно поэтому результат вычислений отображен в командном окне (см. рис. 3, а). После вычислений появится опять символ ввода “**>>**”.

В случае если в последнем выражении поставить в конце точку с запятой, то результат вычислений будет присвоен переменной **R**, но не будет выведен в командное окно (см. рис. 3, б).

```

>> x = 10; y = 25.8;
>> R = 0.5*(x^2 + y^2)

R =

    382.8200

>>

```

a

б

Рис. 3. Использование точки с запятой в арифметических выражениях

MATLAB различает прописные и строчные буквы. Попытка набора команды прописными буквами приведет к ошибке.

Все значения переменных, вычисленные в течении текущего сеанса работы, сохраняются в специально зарезервированной области памяти, называемым рабочим пространством системы MATLAB (Workspace).

Содержимое рабочего пространства отображается в окне *Workspace* (см. рис. 1). Узнать о содержимом рабочего пространства можно также с помощью команд *who* или *whos*. Разница в применении указанных команд хорошо видна из примеров, представленных на *рис. 4*.

```

>> who
Your variables are:
R x y

>> whos
      Name      Size      Bytes  Class  Attributes
      R         1x1         8  double
      x         1x1         8  double
      y         1x1         8  double

>>

```

a

б

Рис. 4. Использование команд *who* и *whos*

В первом случае (см. рис. 4, *a*) отображаются только имена переменных, хранящихся в данный момент в рабочей области, а во втором (см. рис. 4, *б*) – показывается дополнительная информация о размере переменной, количестве байтов, и типе переменной.

Командой

```
>> clc
```

можно стереть видимое содержимое командного окна (очистить окно), однако это не затронет содержимого рабочего пространства. Если после этого набрать имя ранее созданной переменной, например,

>> R

и нажать клавишу **Enter**, то на экране будет показано значение этой переменной.

Удаление всех созданных в текущем сеансе работы системы переменных из рабочей области памяти осуществляется с помощью команды

>> clear

Восстановить удаленные значения нельзя.

Если требуется удалить лишь некоторые переменные из рабочей области, то необходимо указать имена этих переменных в качестве параметров команды clear

>> clear x y

В этой команде имена стираемых переменных отделяются друг от друга пробелом (не запятой).

Система **MATLAB** имеет несколько имен переменных, которые используются самой системой и входят в состав зарезервированных:

i, *j* – мнимая единица (корень квадратный из -1);

pi – число π (сохраняется в виде 3.141592653589793);

inf – обозначение машинной бесконечности;

NaN – обозначение неопределенного результата (например, типа 0/0 или *inf/inf*);

eps – погрешность операций над числами с плавающей запятой;

ans – результат последней операции без знака присваивания;

realmax и *realmin* – максимально и минимально возможные величины числа, которое может быть использованы.

Эти переменные можно использовать в математических выражениях.

Использование математических функций в среде MATLAB

В языке **MATLAB** предусмотрены следующие элементарные арифметические функции (функций на самом деле намного больше, ниже приводятся наиболее употребимые).

Тригонометрические функции

$\sin(Z)$ – синус числа Z ;	$\text{asin}(Z)$ – арксинус (в радианах, в диапазоне от $-\pi/2$ к $+\pi/2$);
$\cos(Z)$ – косинус;	$\text{acos}(Z)$ – арккосинус (в диапазоне от 0 к π);
$\tan(Z)$ – тангенс;	$\text{atan}(Z)$ – арктангенс (в диапазоне от $-\pi/2$ к $+\pi/2$);
$\cot(Z)$ – котангенс;	$\text{acot}(Z)$ – арккотангенс;

Экспоненциальные функции

$\exp(Z)$ – экспонента числа Z ;

$\log(Z)$ – натуральный логарифм;

$\log_{10}(Z)$ – десятичный логарифм;

$\text{sqrt}(Z)$ – квадратный корень из числа Z ;

$\text{abs}(Z)$ – модуль числа Z .

Целочисленные функции

$\text{fix}(Z)$ – округление к ближайшему целому в сторону нуля;

$\text{floor}(Z)$ – округление к ближайшему целому в сторону отрицательной бесконечности;

$\text{ceil}(Z)$ – округление к ближайшему целому в сторону положительной бесконечности;

$\text{round}(Z)$ – обычное округление числа Z к ближайшему целому;

$\text{mod}(X, Y)$ – целочисленное деление X на Y ;

$\text{rem}(X, Y)$ – вычисление остатка от деления X на Y ;

$\text{sign}(Z)$ – вычисление знака числа Z (0 при $Z=0$, -1 при $Z<0$, 1 при $Z>0$).

Для изменения порядка действий при вычислениях используются круглые скобки, например

```
>> 4.5^2*(7.23 - 3.14*10.4)
```

```
ans =
```

```
-514.8765
```

```
>> 4.5^2*7.23 - 3.14*10.4
```

```
ans =
```

```
113.7515
```

В первом случае вычислено выражение $4.5^2 * (7.23 - 3.14 * 10.4)$, а во втором $4.5^2 * 7.23 - 3.14 * 10.4$.

При записи выражений необходимо учитывать порядок вычислений. Особенно в случаях, когда имеется операция деления. Например, для выражения вида

$$\frac{a+b}{a-b} \cdot z$$

Правильной будет следующая запись

```
>> (a+b)/(a-b)*z.
```

В тех случаях, когда есть сомнения в порядке вычислений записываемого выражения лучше проставить требуемый порядок с помощью круглых скобок. Использование скобок другого вида (квадратных, фигурных недопустимо).

Если требуется вычислить значение выражения, похожего на предыдущее, то необязательно снова набирать его в командной строке. Можно воспользоваться тем, что **MATLAB** запоминает все вводимые команды. Для повторного занесения их в командную строку служат клавиши <↑>, <↓>.

При нажатии клавиши <↑> в командной строке появится введенное ранее выражение. Можно ввести в него необходимые изменения (для перемещения по строке с выражением служат клавиши <←>, <→>, <Home>, <End>) и вычислить измененное выражение, нажав <Enter>.

При вычислениях возможны некоторые исключительные ситуации. При делении положительного числа на ноль в **MATLAB** получается *inf* (бесконечность), а при делении отрицательного числа на ноль получается *-inf* (минус бесконечность) и выдается предупреждение:

```
>> 1/0
Warning: Divide by zero
ans =
     Inf
```

При делении нуля на ноль получается **NaN** (не число) и также выдается предупреждение:

```
>> 0/0
Warning: Divide by zero
ans =
     NaN
```

При вычислении, например, $\sqrt{-1}$ MATLAB автоматически переходит в область комплексных чисел:

```
>> sqrt(-1.0)
ans =
    0 + 1.0000i
```

При наборе комплексных чисел в командной строке MATLAB можно использовать либо i , либо j , а сами числа при умножении, делении и возведении в степень необходимо заключать в круглые скобки:

```
>> (2.1 + 3.2i)*2+(4.2 + 1.7i)^2
ans =
    18.9500 + 20.6800i
```

Если не использовать скобки, то умножаться или возводиться в степень будет только мнимая часть и получится неверный результат.

Для вычисления комплексно-сопряженного числа применяется апостроф, который следует набирать сразу за числом, без пробела:

```
>> 2 - 3i'
ans =
    2.0000 + 3.0000i
```

Если необходимо найти комплексно-сопряженное выражение, то исходное выражение должно быть заключено в круглые скобки:

```
>> ((3.2 + 1.5i)*2 + 4.2 + 7.9i)'
ans =
    10.6000 - 10.9000i
```

MATLAB позволяет использовать комплексные числа в качестве аргументов встроенных элементарных функций:

```
>> sin(2 + 3i)
ans =
    9.1545 - 4.1689i
```

Рассмотрим функции для работы с комплексными числами. К ним относятся следующие:

- `abs()`, `angle()` — модуль r и фаза φ (в радианах от $-\pi$ до π) комплексного числа $z = a + i \cdot b = R \cdot (\cos \varphi + i \cdot \sin \varphi) = R e^{-i\varphi}$ (если число не комплексное, то функция `abs(x)` соответствует $|x|$);

- `conj()` — возвращает комплексно-сопряженное число;

- `imag()`, `real()` — мнимая и действительная части комплексного числа.

Более подробные сведения можно получить из обширной интерактивной справочной системы, выбрав в меню **Help** рабочей среды пункт **MATLAB Help**.

Массивы в среде MATLAB

Все данные MATLAB представляет в виде массивов. *Массив* — упорядоченная, пронумерованная совокупность однородных данных. У массива должно быть *имя*. Массивы различаются по *размерности*: одномерные, двумерные, многомерные. Одномерный массив принято называть *вектором*, а двумерный массив — *матрицей*. Вектор может быть записан в столбик (*вектор-столбец*) и в строку (*вектор-строка*). *Размером* массива называют число элементов вдоль каждого из измерений. Доступ к элементам осуществляется при помощи *индекса*. В MATLAB нумерация элементов массивов начинается с *единицы*. Это значит, что индексы должны быть больше или равны единице.

Ввод вектор-столбца в командной строке осуществляется в квадратных скобках, элементы вектор-столбца разделяются точкой с запятой:

```
>> a = [1.3; 5.4; 6.9]
a =
    1.3000
    5.4000
    6.9000
```

Для подавления вывода вектора на экран необходимо завершить выражение точкой с запятой:

```
>> b = [7.1; 3.5; 8.2];
```

Ввод вектор-строки осуществляется в квадратных скобках, однако в отличие от вектор-столбца элементы следует разделять пробелами или запятыми:

```
>> s1 = [3 4 9 2]
s1 =
    3 4 9 2
```

Из нескольких вектор-столбцов можно составить один, используя квадратные скобки и разделяя исходные вектор-столбцы точкой с запятой:

```
>> v1 = [1; 2];
>> v2 = [3; 4; 5];
>> v = [v1; v2]
```



```
v =  
    1  
    2  
    3  
    4  
    5
```

Для сцепления (соединения) вектор-строк также применяются квадратные скобки, но сцепляемые вектор-строки отделяются пробелами или запятыми:

```
>> g1 = [pi pi/2] ;  
>> g2 = [pi/3 pi/4 pi/5];  
>> g = [g1 g2]  
g =  
    3.1416  1.5708  1.0472  0.7854  0.6283
```

Возможен ввод элементов массива в виде арифметических выражений:

```
>> m = [(12+2)/(3+4) sqrt(16) exp(5)];  
m =  
    2.0000  4.0000  148.4132
```

Возможно задание массивов с комплексными элементами:

```
>> f = [1+2i 7+3i 2-5i];
```

Для формирования упорядоченной числовой последовательности, которая идет от начального значения с заданным шагом и завершается конечным значением, в MATLAB используется оператор : (двоеточие) в виде:

Переменная = Начальное значение : Шаг : Конечное значение

Операция формирования диапазона работает следующим образом. Сначала она включает в формируемый массив нижнюю границу диапазона (начальное значение). Затем она к числовому значению прибавляет шаг. Если сумма не превосходит верхней границы диапазона (конечного значения), то она включается в качестве элемента в формируемый массив. Это все повторяется до тех пор, пока очередное значение не превысит верхнюю границу диапазона.

Пример:

```
>> x = 0.6:-0.2:0;  
x =  
    0.6000  0.4000  0.2000  0.0000  
>> y = 1:2:10;  
y =
```

```
1 3 5 7 9
```

Отсюда видно, что конечное значение в массиве не всегда совпадает с заданной в операции двоеточие верхней границей диапазона.

Если шаг равен единице, его можно не указывать.

```
>> x = 1:5;  
x =  
    1 2 3 4 5
```

Доступ к элементам вектора-столбца или вектор-строки осуществляется при помощи индекса, заключаемого в круглые скобки после имени массива, в котором хранится вектор. Если среди переменных рабочей среды есть массив *v*, определенный вектор-строкой

```
>> v = [1.3 3.6 7.4 8.2 0.9];
```

то для обращения, например, к четвертому элементу используется *индексация*:

```
>> h = v(4)  
h =  
    8.2000
```

Указание элемента массива в левой части оператора присваивания приводит к изменению в массиве

```
>> v(2) = 555  
v =  
    1.3000 555.0000 7.4000 8.2000 0.9000
```

Из элементов массива можно формировать новые массивы, например:

```
>> u = [v(3); v(2); v(1)]  
u =  
    7.4000  
   555.0000  
    1.3000
```

В массив можно добавить элементы, например

```
>> u(4) = 15  
v =  
    7.4000  
   555.0000  
    1.3000  
   15.0000
```

Для помещения определенных элементов вектора в другой вектор в заданном порядке служит *индексация при помощи вектора*. Например, запись в

вектор-строку W четвертого, второго и пятого элементов V производится следующим образом:

```
>> ind =[4 2 5];
```

```
>> w = v(ind)
```

```
w =
```

```
8.2000 555.0000 0.9000
```

Элементы массива могут входить в выражения:

```
>> p = w(1)*5+w(4)
```

```
p =
```

```
5.6
```

Ввод матрицы из командной строки осуществляется в квадратных скобках, элементы матриц в каждой строке разделяются пробелом или запятой, а строки разделяются точкой с запятой:

```
>> A=[1 3 5; 7 9 0]
```

```
A =
```

```
1 3 5
```

```
7 9 0
```

Другой способ ввода матриц заключается в том, что матрицу можно трактовать как вектор-строку, каждый из элементов которой является вектором столбцом:

```
>> B=[[3; 4; 6] [-1; 2; 8] [7; 0; 9]]
```

```
B =
```

```
3 -1 7
```

```
4 2 0
```

```
6 8 9
```

или как вектор-столбец, каждый элемент которого является вектор-строкой:

```
>> B=[[3 -1 7]; [4 2 0]; [6 8 9]]
```

Доступ к элементам матриц осуществляется при помощи двух индексов – номеров строки и столбца, заключенных в круглые скобки:

```
>> B(2,3)
```

```
ans =
```

```
0
```

Элементы матриц могут входить в состав выражений:

```
>> B(1,1)+B(3,1)
```

```
ans =
```

```
9
```

В качестве индексов могут выступать векторы, содержащие номера нужных строк и столбцов. Например, для выделения элементов первой и второй строк второго и третьего столбцов матрицы **B** достаточно ввести команды:

```
>> i=[1 2]; j=[2 3]; B1=B(i, j)
B1 =
    -1  7
     2  0
```

Используя индексы элементов, можно создавать матрицы, задавая элементы по отдельности в любом порядке. При задании элементов возможно использование арифметических операций и различных функций, например

```
>> D(1,1)=2; D(1,2)=7.5; D(2,1)=D(1,1)+2*D(1,2); D(2,2)=sin(pi/2);
```

Существуют специальные команды для формирования массивов различных видов.

Формирование массивов, состоящих из единиц или нулей, выполняется соответственно с помощью функций `ones()` и `zeros()`, аргументами которых являются число строк и число столбцов массива:

```
>> M1 = ones(2,3)
M1 =
     1  1  1
     1  1  1
>> M2 = zeros(1,4)
M2 =
     0  0  0  0
```

Использование одного аргумента в функциях `ones()` и `zeros()` приводит к созданию квадратных матриц.

Для создания единичной матрицы используется функция `eye()`:

```
>> M3 = eye(3)
M3 =
     1  0  0
     0  1  0
     0  0  1
```

Для заполнения массивов элементами, случайно распределенными между нулем и единицей, используется функция `rand()`, аргументами которой являются число строк и число столбцов массива. При наличии одного входного аргумента формируется квадратная матрица.

Формирование диагональной матрицы, у которой все внедиагональные элементы равны нулю, применяется функция `diag()`. В качестве аргумента этой функции используется вектор-строка или вектор-столбец, элементы которого располагаются по диагонали создаваемой матрицы.

```
>> d = [1; 2; 3];
>> M4 = diag(d)
M4 =
    1  0  0
    0  2  0
    0  0  3
```

MATLAB предоставляет удобный способ обращения к блокам последовательно расположенных элементов массива. Для этого служит *индексация при помощи знака двоеточия*. Предположим, что в заданной вектор-строке `w` из семи элементов требуется заменить нулями элементы со второго по шестой. Индексация при помощи двоеточия позволяет просто и наглядно решить поставленную задачу:

```
>> w = [0.1 2.9 3.3 5.1 2.6 7.1 9.8];
>> w(2:6) = 0;
w =
    0.1000  0  0  0  0  9.8000
```

Индексация при помощи двоеточия оказывается удобной при выделении части из большого объема данных в новый массив:

```
>> w = [0.1 2.9 3.3 5.1 2.6 7.1 9.8];
>> w1 = w(3 : 5)
w1 =
    3.3000  5.1000  2.6000
>> m = [1 2 0 2; 4 10 20 5; 2 30 40 6];
m =
    1  2  0  2
    4 10 20  5
    2 30 40  6
>> m1 = m(2 : 3, 2 : 3)
m1 =
    10 20
    30 40
```

Запись `w(5:end)` означает, что берутся элементы вектора `w`, начиная с пятого и заканчивая последним. Если же ввести просто `w(end)`, то получим последний элемент массива:

```
>> w(end)
ans =
    9.8000
```

Для выделения строки или столбца массива следует в качестве одного из индексов использовать номер строки или столбца, а другой индекс заменить на двоеточие без указания пределов:

```
>> m2 = m(2,:)
m2 =
    4 10 20 5
```

Для удаления строки, столбца матрицы или элемента вектора следует присвоить удаляемому элементу пустой массив []. Например, для удаления первой строки матрицы *m* нужно записать

```
>> m(1,:) = []
m =
    4 10 20 5
    2 30 40 6
```

Для удаления нескольких столбцов следует присвоить им пустой массив:

```
>> m(:,2:3)=[ ]
m =
    4 5
    2 6
```

Логическое индексирование позволяет выбрать из массивов элементы, удовлетворяющие определенным условиям, которые заданы логическим выражением. Для выделения из матрицы *m* элементов, которые больше 10, нужно записать

```
>> f = m(m>10)
f =
    30
    20
    40
```

При этом индексы элементов остаются неизвестными. Для поиска индексов элементов, удовлетворяющих условию, используется функция `find()`.

```
>> [i,j]=find(m>10)
i =
    3
    2
    3
j =
```

2
3
3

Вычисления с применением массивов

В системе MATLAB допускаются групповые вычисления над всем массивом сразу. При этом нет необходимости выполнять каждую отдельную операцию над отдельным элементом массива. Это позволяет компактно создавать выражения, при вычислении которых реально выполняется значительный объем работы.

Начнем рассмотрение с арифметических операций. Над массивами одинакового размера допускаются операции сложения и вычитания, обозначаемые стандартными знаками + и -. Например:

```
>> A = [1 1 1; 2 2 2; 3 3 3];  
>> B = [0 0 0; 7 7 7; 1 2 3];  
>> C = A+B  
C =  
    1    1    1  
    9    9    9  
    4    5    6
```

Эти операции выполняются поэлементно и порождают массивы тех же размеров, что и исходные операнды. Если используются массивы разных размеров, то выдается сообщение об ошибке. Исключение составляют случаи, когда одним из операндов является скаляр (число или переменная размером 1x1), например:

```
>> C+1  
ans =  
    2    2    2  
   10   10   10  
    5    6    7
```

Для поэлементного перемножения и поэлементного деления массивов одинаковых размеров применяются операции, обозначаемые комбинациями двух символов: .* (точка умножить) и ./ (точка поделить). Использование комбинации символов объясняется тем, что символами * и / в системе MATLAB обозначены специальные операции линейной алгебры над векторами и матри-

цами. Кроме операции $./$, называемой операцией правого поэлементного деления, есть еще операция левого поэлементного деления \backslash . Выражение $A./B$ приводит к получению матрицы с элементами $A(k,m)/B(k,m)$, а выражение $A.\backslash B$ приводит к матрице с элементами $B(k,m)/A(k,m)$.

При помощи $.^{\wedge}$ осуществляется поэлементное возведение в степень:

```
>> v1 = [2 -3 4 1];
>> p = v1.^2
p =
    4  9 16  1
```

Показателем степени может быть вектор той же длины, что и возводимый в степень. При этом каждый элемент первого вектора возводится в степень, равную соответствующему элементу второго вектора:

```
>> v2 = [3 2 2 9];
>> p1 = v1.^v2
p1 =
    8.0000  9.0000 16.0000  1.0000
```

Кроме указанных операций над массивами, можно производить вычисления с использованием математических функций. Например:

```
>> x = 0:0.01:2*pi;
>> y = sin(x);
```

В данном случае вычисляется массив y значений функции $\sin(x)$ для значений x , лежащих в пределах от 0 до 2π с шагом 0,01. При этом выполняется большой объем вычислений, так как значений x много и для каждого x необходимо вычислить y .

Знак $*$ закреплен за перемножением матриц и векторов в смысле линейной алгебры. Эта операция выполнима лишь только тогда, когда число столбцов в левом операнде равно числу строк в правом операнде:

```
>> A1 = [1 2 3; 4 5 6; 7 8 9]
A1 =
    1  2  3
    4  5  6
    7  8  9
>> v = [1; 2; 3]
v =
    1
    2
```



```

3
>> B1 = A1*v
B1 =
    14
    32
    50

```

С помощью операции \ (левого деления) можно решать системы линейных уравнений. Например, пусть задана система линейных уравнений

$$\begin{cases} 2x_1 + 3x_2 = 18; \\ 5x_1 + 6x_2 = 39. \end{cases}$$

Эту систему можно переписать в матричном виде, т. е.

$$A x = b,$$

где

$$A = \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad b = \begin{bmatrix} 18 \\ 39 \end{bmatrix}.$$

Для нахождения x в системе MATLAB достаточно написать

```

>> A = [2 3; 5 6]; b = [18;39];
>> x = A \ b
x =
     3
     4

```

Для замены строк матрицы столбцами используется операция транспонирования, например

```

>> D = A1'
D =
     1     4     7
     2     5     8
     3     6     9

```

Перемножение элементов векторов, а также элементов столбцов матриц осуществляется при помощи функции prod():

```

>> z = [3; 2; 1; 4; 6; 5];
>> p = prod(z)
p =
    720

```

Функция sum() предназначена для суммирования элементов вектора или элементов столбцов матриц.

```

>> z1 = [3 2 1; 4 6 5];

```

```
>> s = sum(z)
s =
    7 8 6
```

Для вычисления среднего арифметического элементов вектора, используется функция `mean()`.

```
>> mean(z)
ans =
    3.5000
```

Для нахождения минимума и максимума из элементов вектора или элементов столбцов матриц служат встроенные функции `min()` и `max()`:

```
>> m = min(z)
m =
    1
>> M = max(z)
M =
    6
```

Чтобы определить максимальный или минимальный элемент матрицы, нужно использовать соответствующие функции дважды

```
>> M1 = max(max(D))           >> m1 = min(min(D))
M1 =                           m1 =
    9                             1
```

или

```
>> M1 = max(D(:))           >> m1 = min(D(:))
M1 =                           m1 =
    9                             1
```

При обращении к функции `min()` с двумя векторами в качестве входных аргументов получится вектор, каждый элемент которого есть минимум из двух элементов исходных векторов с одинаковыми номерами.

```
>> p = [3 12 8];
>> s = [4 10 7];
>> min(p, s)
ans =
    3 10 7
```

Часто необходимо знать не только значение минимального или максимального элемента в массиве, но и его индекс (порядковый номер). Для этого необходимо вызвать функцию `min()` с двумя выходными аргументами:

```
>> [m, k] = min(z)
m =
```

```
1
k =
3
```

В результате переменной *m* будет присвоено значение минимального элемента массива *Z*, а номер минимального элемента занесен в переменную *k*.

В число основных функций для работы с массивами входит функция упорядочения элементов вектора или элементов столбцов матрицы по возрастанию `sort()`.

```
>> r = [9.4 -2.3 -5.2 7.1 0.8 1.3];
>> R = sort(r)
R =
-5.2000 -2.3000 0.8000 1.3000 7.1000 9.4000
```

Можно упорядочить элементы по убыванию, используя функцию `sort()`:

```
>> R1 = -sort(-r)
R1 =
9.4000 7.1000 1.3000 0.8000 -2.3000 -5.2000
```

Упорядочение элементов в порядке возрастания их модулей производится с привлечением функции `abs()`:

```
>> R2 = sort(abs(r))
R2 =
0.8000 1.3000 2.3000 5.2000 7.1000 9.4000
```

Вызов `sort()` с двумя выходными аргументами приводит к образованию массива индексов соответствия элементов упорядоченного и исходного массивов:

```
>> [rs, ind] = sort(r)
rs =
-5.2000 -2.3000 0.8000 1.3000 7.1000 9.4000
ind = 3 2 5 6 4 1
```

Построение графиков в среде MATLAB (краткие сведения)

В результате вычислений в системе MATLAB обычно получается большой массив данных, который трудно анализировать без наглядной визуализации. Графические возможности системы MATLAB являются мощными и разнообразными. В первую очередь целесообразно изучить наиболее простые в использовании возможности. Их часто называют *высокоуровневой графикой*. Это

название отражает тот факт, что пользователю нет необходимости вникать во все детали работы с графикой.

Например, чтобы построить график функции одной вещественной переменной $y = \sin(x)$, $x \in [0, 4\pi]$, необходимо:

- 1) задать вектор значений аргумента x ;
- 2) вычислить вектор y значений функции $\sin(x)$;
- 3) использовать команду `plot()` для построения графика.

Таким образом, для построения графика функции $y = \sin(x)$ необходимо ввести в командном окне следующие выражения:

```
>> x=0:0.01:4*pi;  
>> y=sin(x);  
>> plot(x,y)
```

Команды для задания вектора x и вычисления функции y лучше завершать точкой с запятой для подавления вывода в командное окно значений x и y . После команды `plot()` точку с запятой ставить не нужно, так как эта команда ничего не выводит в командное окно.

MATLAB показывает графические объекты в специальных *графических окнах*, имеющих в заголовке слово **Figure** (фигура, изображение). После выполнения команды на экране появится графическое окно с заголовком **Figure 1**, содержащее график функции (рис. 5).

При построении графиков функций очень большую часть работы MATLAB берет на себя. После введения в командной строке одной функции `plot()` система сама создает графическое окно, строит оси координат, вычисляет диапазоны изменения переменных x и y , ставит на осях метки и соответствующие им числовые значения, проводит через опорные точки график функции некоторым выбранным по умолчанию цветом, в заголовке графического окна надписывает номер графика в текущем сеансе работы.

В общем случае функция

```
plot(x1, y1, 's1', x2, y2, 's2', ...)
```

позволяет объединить в одном графическом окне несколько графиков функций $y_1(x_1)$, $y_2(x_2)$,... проведя их со стилями s_1 , s_2 ,... и т. д.

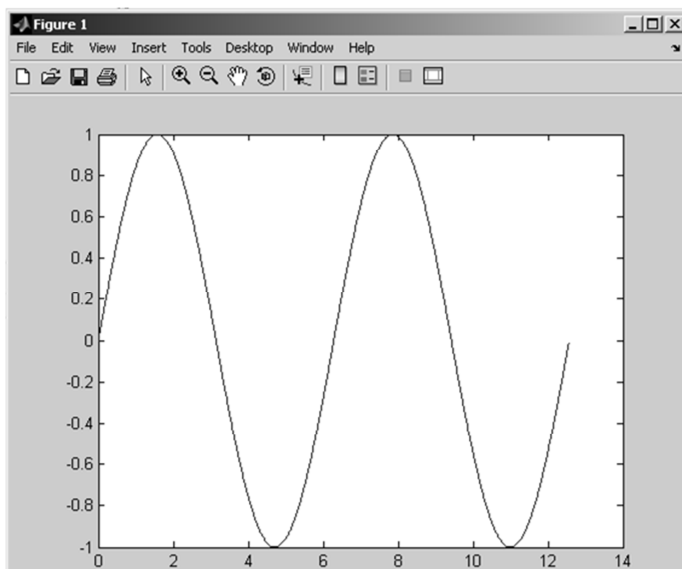


Рис. 5. График функции $y = \sin(x)$

Стили s_1, s_2, \dots задаются в виде набора *символов*, заключенных в одиночные кавычки (апострофы). Один из этих символов задает цвет, а го́й - тип линии или маркера. В табл. 2 приведены возможные цвета и типы линий и маркеров с указанием результата.

Таблица 2. Свойства линий и маркеров

Цвет		Тип маркера		Тип линии	
y	Желтый	.	Точка	-	Сплошная
m	Розовый	o	Кружок	:	Пунктирная
c	Голубой	x	Крестик	-.	Штрих-пунктирная
r	Красный	+	Знак «плюс»	--	Штриховая
g	Зеленый	*	Звездочка		
b	Синий	s	Квадрат		
w	Белый	d	Ромб		
k	Черный	v	Треугольник вершиной вниз		
		^	Треугольник вершиной вверх		
		<	Треугольник вершиной влево		
		>	Треугольник вершиной вправо		
		p	Пятиконечная звезда		
		h	Шестиконечная звезда		

Например, для построения графика $y = \sin(x)$ красной штриховой линией, а графика $z = \cos(x)$ маркерами в виде звездочек синего цвета на месте вычисляемых точек (рис. б) следует использовать команды

```
>> x=0:0.1:4*pi;
```

```
>> y=sin(x);z=cos(x);  
>> plot(x,y,'r--',x,z,'b*')
```

Удобство использования графиков во многом зависит от дополнительных элементов оформления: координатной сетки, подписей к осям, заголовка и легенды.

Сетка наносится командой `grid on` (или просто `grid`), а удаляется командой `grid off` (или повторением команды `grid`).

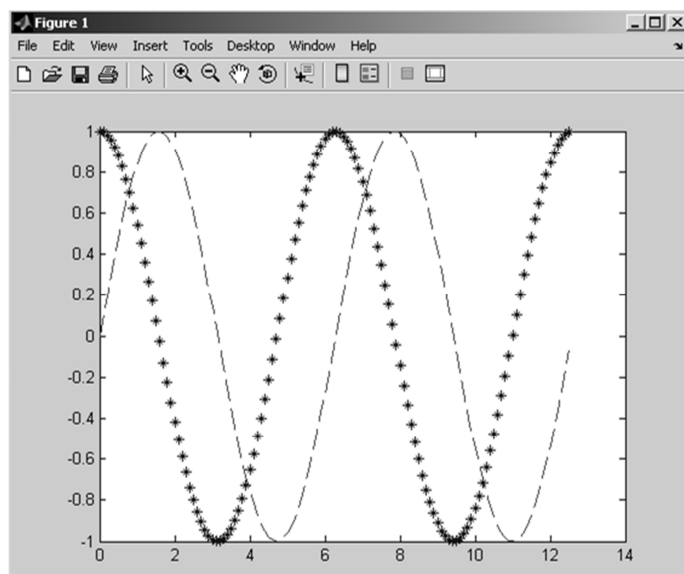


Рис. 6. Изменение параметров линий

Для размещения подписей к осям служат функции `xlabel()` и `ylabel()`. Функция `xlabel('название оси')` предназначена для проставления названия горизонтальной оси, функция `ylabel('название оси')` — то же для вертикальной оси (причем эти надписи ориентированы вдоль осей координат).

Общий заголовок для графика проставляется функцией `title('заголовок')`.

Если требуется разместить какую-либо надпись в произвольном месте рисунка, применяем функцию `text()`:

```
text(a,b,'надпись')
```

Надпись, задаваемая функцией `text()`, помещается, начиная от точки с координатами, указанными первыми двумя аргументами `a` и `b`. По умолчанию координаты задаются в тех же единицах измерения, что и координаты, указанные на горизонтальной и вертикальной осях.

Для размещения легенды на графике используется команда

legend ('график 1','график2',...)

При размещении легенды следует учесть, что порядок и количество аргументов команды `legend()` должны соответствовать линиям на графике.

Применяя все эти средства для оформления графиков, представленных на *рис. 6*, получим:

```
>> title('Графики функций y=sin x и z=cos x');  
>> xlabel('x')  
>> ylabel('sin x, cos x')  
>> legend('sin x','cos x')  
>> grid on;
```

Результаты оформления графиков представлены на *рис. 7*.

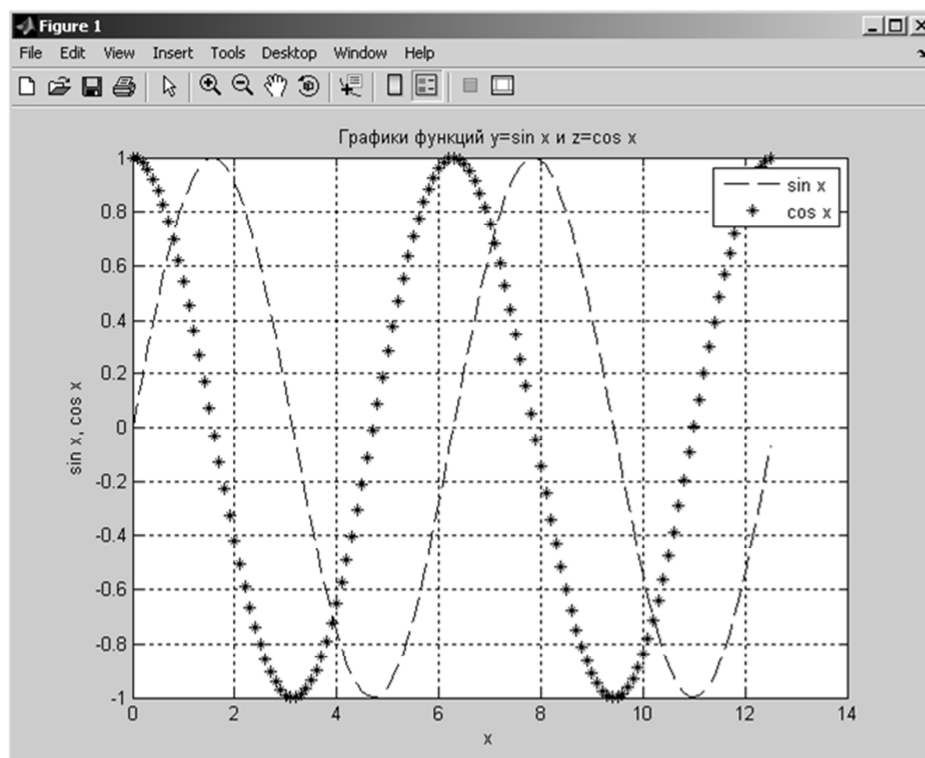


Рис. 7. Оформление графического окна

Система **MATLAB** устанавливает пределы на горизонтальной оси равными тем значениям, что указаны пользователем для независимой переменной. Для зависимой переменной по вертикальной оси **MATLAB** самостоятельно вычисляет диапазон изменения значений функции. В результате график функции оказывается как бы вписанным в прямоугольник.

Если необходимо отказаться от этой особенности масштабирования при построении графиков, то можно явным образом задать свои пределы изменения переменных по осям координат. Это делается с помощью функции

`axis([xmin, xmax, ymin, ymax])`

причем команду на выполнение этой функции можно вводить с клавиатуры сколько угодно раз уже после построения графика функции, чтобы, глядя на получающиеся визуальные изображения, добиться наилучшего восприятия. Такое масштабирование позволяет получить подробные изображения тех частей графика, которые вызывают наибольший интерес в конкретном исследовании.

В методических указаниях представлена краткая информация о возможностях и основных принципах использования среды **MATLAB** для выполнения математических вычислений в режиме командного окна. Более подробные сведения и примеры применения системы **MATLAB** для решения различных вычислительных задач можно найти в литературе (см. представленный выше список) и на специализированных сайтах в сети интернет.