

Министерство образования Республики Беларусь
Учреждение образования
«Гомельский государственный технический университет
имени П.О.Сухого»

Т.А. Трохова, Т.Л. Романькова, В.В. Кротенок

**СИСТЕМА МАТЕМАТИЧЕСКИХ
ВЫЧИСЛЕНИЙ МАТЛАВ**

ПРАКТИЧЕСКОЕ ПОСОБИЕ

для студентов технических специальностей
дневного отделения

Гомель 2004

УДК 681.3.06.

Авторы-составители: Трохова Т.А., Романькова Т.Л., Кротенок В.В.

Система математических вычислений MatLab: практическое пособие по курсу «Информатика» для студентов технических специальностей дневного отделения. - Гомель: Учреждение образования "ГГТУ имени П.О.Сухого", 2004. – 22 с.

Практическое пособие содержит описание основных приемов работы с системой компьютерной математики MatLab. Дана краткая характеристика интерфейса системы, рассмотрены структуры и типы данных, стандартные матричные функции, функции по решению уравнений и систем. Показаны возможности графического оформления результатов расчетов.

Практическое пособие предназначено для для студентов технических специальностей дневного отделения ГГТУ имени П.О.Сухого.

Рецензент:

© Учреждение образования "Гомельский государственный технический университет имени П.О.Сухого ", 2004

ВВЕДЕНИЕ

Система компьютерной математики MatLab является в настоящее время одним из наиболее эффективных инструментов для выполнения научно-технических расчетов, графической интерпретации полученных результатов и визуального моделирования. Эта система имеет удобный пользовательский интерфейс, развитый язык программирования, ядро символьных вычислений и множество дополнительных пакетов, таких, как Simulink, SimPowerSystem и другие.

Практическое пособие состоит из двух частей. Первая часть дает представление об интерфейсе и внутреннем языке системы, позволяет разобраться в работе с матрицами, научиться строить двумерные и трехмерные графики, решать уравнения и системы. Вторая часть рассматривает вопросы программирования, создания пользовательского графического интерфейса, М-файлов, решения дифференциальных уравнений и систем, аппроксимации экспериментальных данных. В этой же части рассматриваются основные принципы работы с пакетом Simulink.

Овладев навыками и приемами работы в системе MatLab, студенты технических специальностей университета смогут качественно и достаточно быстро проводить в ней нужные исследования и вычисления по курсовому и дипломному проектированию, применять систему для решения сложных инженерных задач.

1. ИНТЕРФЕЙС СИСТЕМЫ МАТЛАВ

1.1. Виды окон

Система MatLab обладает развитым многофункциональным интерфейсом, под управлением которого можно выполнить инженерные расчеты, их графическую интерпретацию, получить результаты в численном и аналитическом виде, создать пользовательскую среду с кнопками управления процессом вычислений и т. д.

Система имеет несколько режимов работы, каждый из которых поддерживается собственным диалоговым окном. Ниже приведен перечень основных режимов работы в системе с указанием соответствующих им окон.

- Командный режим – командное окно (MatLab Command Window)
- Программный режим – окно создания и редактирования М-файлов
- Графический режим – окно редактирования графиков (Figure)
- Режим создания графического интерфейса – окно графического интерфейса(GUI) – предназначен для разработки удобного

пользовательского интерфейса с применением кнопок, меню, окон ввода и вывода информации и т.д.

- Режим помощи – окно помощи (Help)
- Режим демонстрации – окон демонстрационных примеров (Demos)

При работе в любом из перечисленных режимов могут быть использованы дополнительные информационные окна, такие, как:

- Окно рабочей области (Workspace Browser) – предназначено для просмотра и редактирования содержимого рабочей области памяти, в нем указывается имя переменной (массива или структуры), ее размерность и тип.
- Окно быстрого доступа (Launch Pad) – позволяет организовать быстрый доступ к компонентам системы MatLab (элементам справки, демонстрационным примерам, специализированным пакетам и т.д.)
- Окно доступа к файлам и папкам (Current Directory) – предназначено для вызова нужного файла в текущем каталоге, изменения текущего каталога.
- Окно стека команд (Command History) – содержит перечень команд, введенных пользователем в командном режиме за текущий и предыдущий сеансы работы с системой.
- Окно редактора массивов (Array Editor) – появляется после двойного щелчка левой кнопкой мыши по имени массива в окне рабочей области Workspace Browser и служит для редактирования содержимого массива.

После первоначальной загрузки системы MatLab на экране появляется главное окно или рабочий стол системы, на котором пользователь может по желанию устанавливать нужные ему окна. На рисунке 1.1 приведены примеры различных интерфейсных окон рабочего стола системы.

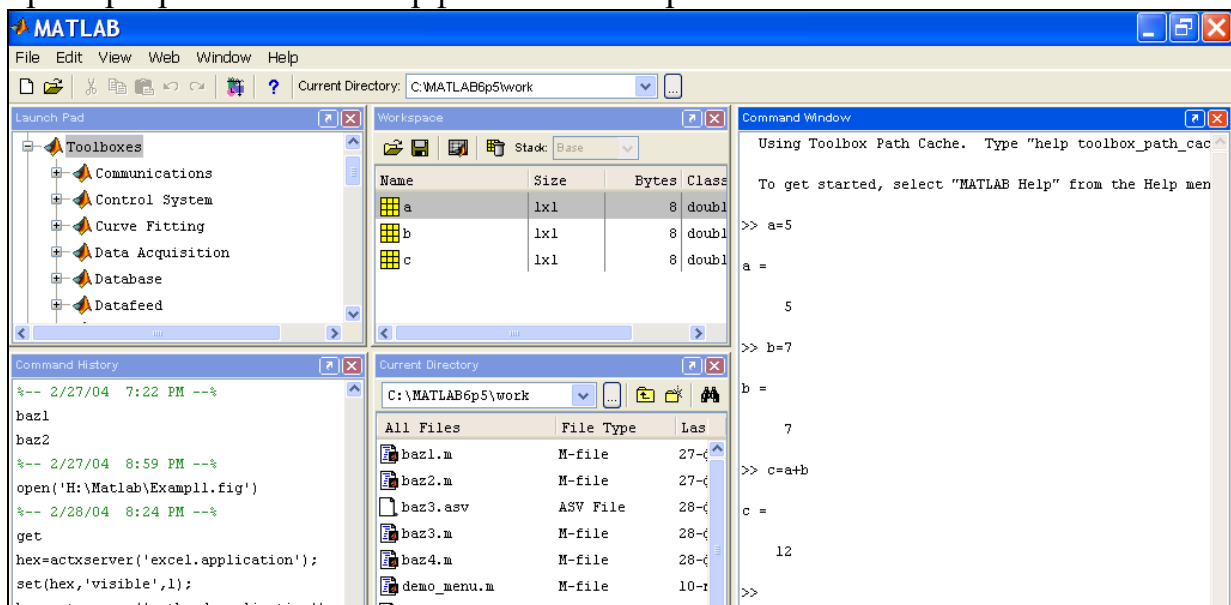


Рисунок 1.1 - Примеры различных интерфейсных окон рабочего стола системы.

Вне зависимости от обилия интерфейсных окон последовательность работы и приемы вычислений в системе MatLab достаточно просты и доступны для изучения.

1.2. Команды основного меню рабочего стола MatLab

Основное меню рабочего стола позволяет переключать режимы работы в системе, считывать и сохранять информацию во внешних файлах, изменять вид интерфейсных окон и т.д. Ниже приведены разделы команд основного меню с их краткой характеристикой.

Раздел **File** имеет следующие команды: New – создание новых файлов различных типов (М-файлы, графические файлы, интерфейсные файлы и т.д.); Open – открытие уже существующего файла; Close – закрытие отдельного окна; Import Data – ввод данных из других систем; Save Workspace As – сохранение данных рабочей области системы и т.д.

Раздел **Edit** позволяет выполнять стандартный набор команд по работе с буфером обмена, кроме того, имеет команды по очистке командного окна (Clear Command Window), окна стека команд (Clear Command History), окна рабочей области (Clear Workspace).

Раздел **View** дает возможность отображать на экране различные компоненты системы, задавать свойства их вывода.

Раздел **Web** вызывает браузер Интернет и делает попытку загрузки сайта компании MathWorks, если доступ в Интернет открыт.

Раздел **Window** выводит перечень всех открытых на текущий сеанс работы с системой дополнительных окон (графического окна, окна М-файла) и команду Close All, позволяющую закрыть все дополнительные окна.

Раздел **Help** содержит команды вызова браузера справочной системы MatLab, вызова окна с демонстрационными примерами (Demos).

1.3. Правила формирования и редактирования документа

Создание документа в системе MatLab может выполняться в различных режимах, ниже рассматриваются командный режим и режим М-файлов.

Документ при работе в командном режиме представляет собой последовательность команд пользователя и ответов системы, расположенных в окне MatLab Command Window. Символ >> в окне команд показывает, что система готова к диалогу с пользователем. Командная строка может содержать одну или несколько команд, она завершается нажатием клавиши Enter. Строка реакции системы называется строкой вывода, она показывает результаты выполнения команды либо в

стандартной переменной ответа *ans*, либо в переменной, заданной пользователем, например:

```
>> 5+3
ans=
8
>> b=5+3
b=
8
```

Переменная *ans* хранится в памяти и может использоваться в дальнейших вычислениях. Выражения или команды в строке разделяются символами «,» или «;». Отсутствие разделителя между командами равносильно запятой. Результат вычисления выражения или команды, за которой следует запятая, выводится на экран в строку вывода. Результат вычисления выражения или команды, за которой следует точка с запятой, на экран не выводится, но сохраняется в памяти, например:

```
>> x=5, y=x+7;
x=
5
```

Отработанная командная строка не может быть выполнена повторно путем возвращения в нее курсора мыши в командном окне. Для нового выполнения команды она должна быть вызвана из стека команд нажатием клавиш ↑ или ↓. Стек или «история» команд сохраняет команды текущего и предыдущего сеансов работы с системой. Стек команд выводится в окне Command History, из которого она может быть скопирована в командное окно через буфер обмена.

Если выражение не помещается в одной командной строке, то его можно перенести на следующую строку, а предыдущую закончить тремя точками.

M-файл представляет собой программу, состоящую из команд и выражений системы MatLab, хранящуюся на диске в виде файла с типом *.m*. Создать новый M-файл можно с помощью команд основного меню **File – New – M-file**, после чего на экране появляется окно редактора M-файлов. Окно содержит сервисно-командную область (три верхние строки) и область ввода и редактирования M-файла. Для удобства отладки строки команд программы пронумерованы. При наведении курсора мыши на имена переменных появляются всплывающие подсказки с начальными числовыми значениями переменных. Последовательность обработки M-файла такова:

- создание или редактирование M-файла;
- запись файла на диск с именем, содержащим тип *.m*;
- запустить программу на выполнение, указав имя файла в командной строке. На рисунке 1.2 приведен вид окон системы при работе в командном режиме и в режиме создания M-файлов. Подробнее приемы

программирования с использование М-файлов будут описаны во второй части пособия.

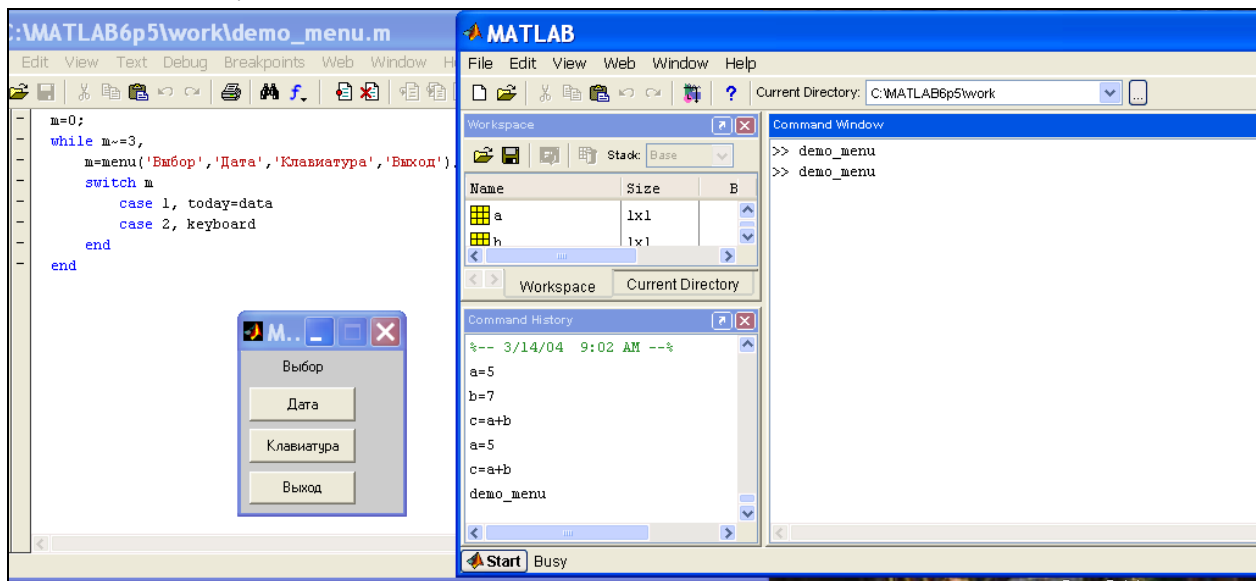


Рисунок 1.2 - Работа в командном режиме и в режиме создания М-файлов.

2. ЭЛЕМЕНТЫ ЯЗЫКА СИСТЕМЫ MATLAB

2.1. Основные понятия

Данные в системе Matlab представлены переменными, константами и выражениями.

Переменные – это имеющие имена объекты, способные хранить некоторые данные.

Переменные в Matlab не нужно предварительно описывать, указывая их тип. Тип переменной определяется ее значением. Так, например, переменная, в которую записана матрица, является матричной.

Имя переменной (идентификатор) должно начинаться с буквы, может содержать буквы латинского алфавита, цифры и символ подчеркивания, причем Matlab различает большие и малые буквы. Допустимы имена любой длины, но значимым является только 31 начальный символ. Имя любой переменной должно быть уникальным, т.е. оно не должно совпадать с именами переменных, функций и процедур системы.

Примеры допустимых имен переменных: V_1, speed, vremja25

Для того, чтобы задавать переменным определенные значения, в системе Matlab используется **оператор присваивания**, вводимый знаком равенства =.

Имя переменной = Выражение

Например, >>a=2+3

Если переменная для значения результата вычислений не указана, то Matlab использует системную переменную **ans**, хранящую результат последней операции.

Например, >>2+3

ans=

5

Ниже описаны основные **системные** переменные и константы, применяемые в Matlab:

- i или j – мнимая единица (корень квадратный из -1);
- π – число $\pi=3,1415926\dots$;
- eps – погрешность операций над числами с плавающей точкой (2^{-52});
- inf – значение машинной бесконечности;
- NaN – указание на нечисловой характер данных(Not-a-Number);
- ans – результат выполнения последней операции;
- end – наибольшее значение индекса размерности массива.

Для создания различных объектов языка используются также **специальные символы**. Такие как `:`, `()`, `[]`, `!`, `%` и др.

Символ `%` используется для ввода **комментариев** к вычислениям.

Например,

```
>>L=2*pi*r % Вычисление длины окружности
```

2.2. Структуры данных

В системе Matlab существуют следующие основные типы данных:

- массивы чисел (вектора, матрицы, многомерные массивы);
- строки символов;
- массивы ячеек;
- структуры.

Массив - это последовательность однотипных элементов, снабженных индексами (порядковыми номерами).

Вектором принято называть одномерный массив.

Матрица - это двумерный массив.

Так как Matlab – система, специально предназначенная для проведения сложных вычислений с векторами, матрицами, массивами, то по умолчанию все числовые переменные в Matlab считаются матрицами, так что скалярная величина есть матрица первого порядка, а векторы – это матрицы, состоящие из одного столбца или одной строки.

Элементами числовых массивов являются действительные или комплексные числа. **Действительные** числа могут быть целыми, дробными, с фиксированной и плавающей точкой. Возможно представление чисел в научном формате с указанием мантиссы и порядка числа. Дробная часть отделяется от целой точкой. Например, **5** **-45**
2.301 **125.35e-25**

Комплексные числа содержат действительную и мнимую части. Мнимая часть имеет множитель i или j , означающий корень квадратный из -1 . Например, $3i - 5j - 3.2 + 2i$

Чтобы задать **вектор-строку**, значения его элементов следует перечислить в квадратных скобках, разделяя пробелами.

Например, `>>V=[2 3 8 0]`

Чтобы задать **вектор-столбец**, значения его элементов следует перечислить в квадратных скобках, используя для разграничения строк точку с запятой или клавишу **Enter**.

Например,

<code>>>V=[2;3;8]</code>	или	<code>>>V=[</code>
<code>V=</code>		<code>2</code>
<code>2</code>		<code>3</code>
<code>3</code>		<code>8];</code>
<code>8</code>		

Векторы могут быть сформированы как **диапазоны** – при помощи двоеточий, разделяющих начальное значение n , шаг m и конечное значение k :

$n:m:k$

В результате будет сформирован вектор, первый элемент которого равен n , второй – $n+m$, третий - $n+m+m$ и т.д. Последний элемент будет не больше k для положительного шага m , и не меньше k – для отрицательного. Если величина шага отсутствует, то по умолчанию его значение равно единице:

$n:k$

Например,

<code>>>d=1:2:7</code>	<code>>>x=2:-2:-4</code>	<code>>>v=3:5</code>
<code>d=</code>	<code>x=</code>	<code>v=</code>
<code>1 3 5 7</code>	<code>2 0 -2 -4</code>	<code>3 4 5</code>

Обращение к элементу вектора происходит следующим образом: после имени вектора в круглых скобках указывается индекс. Например, $V(3)$ – это третий элемент вектора. В качестве индекса можно использовать также переменную или арифметическое выражение

Чтобы задать **матрицу**, значения ее элементов следует перечислить в квадратных скобках, разделяя элементы в строках пробелами или запятыми, а для разграничения строк использовать точку с запятой или клавишу **Enter**. Например,

`>>A=[3 1 -3; 2 0 1.5]`

`A =`

```
3.0000 1.0000 -3.0000
2.0000      0 1.5000
```

Возможен ввод элементов матриц и векторов в виде арифметических выражений. Например,

```
>> A=[2+sin(3.5) 3.4 4*5; 1 0 2/(3-5)]
```

```
A =  
    1.6492    3.4000   20.0000  
    1.0000         0   -1.0000
```

Для указания отдельного элемента матрицы после имени матрицы в круглых скобках указываются два индекса: номер строки и номер столбца. Например, $M(2,3)$ – это третий по столбцу и второй по строке элемент матрицы M . Для матрицы из предыдущего примера $A(2,1)=1$.

В качестве индекса можно также использовать системную переменную **end** – максимальное значение индекса:

```
>> A(end,2)  
ans =  
     8
```

Строка символов представляет собой набор символов, заключенных в апострофы. Например, $\gg S='Курить вредно!'$

```
S =  
    Курить вредно!
```

Для обращения к определенному символу строки нужно после имени строки в круглых скобках указать порядковый номер символа в строке. Например для строки $S(5)$ – буква т.

Для хранения разнородных объектов (массивов разных размерностей, разнотипных данных) в системе Matlab используется такой тип данных как массивы ячеек.

Массив ячеек – это совокупность компонентов(ячеек), снабженных индексами и содержащих любые типы массивов, включая массивы ячеек.

Чтобы задать **массив ячеек**, можно значения его элементов (ячеек) перечислить в фигурных скобках, разделяя элементы в строках пробелами или запятыми, а для разграничения строк использовать точку с запятой. Например,

```
>> V={'мама'; [1;2]; [1:0.5:2.5]};
```

Для обращения к ячейке необходимо после имени массива ячеек в фигурных скобках указать один или более индексов, в зависимости от размерности массива. Например, для массива из предыдущего примера получим следующие значения ячеек:

<code>>> V{1}</code>	<code>>> V{2}</code>	<code>>> V{3}</code>
ans = мама	ans = 1 2	ans = 1.0000 1.5000 2.0000 2.5000

Структура – это совокупность фиксированного числа компонентов, имеющих имена и называемых полями, которые могут быть разных типов. Обращение к полю структуры осуществляется следующим образом:

имя структуры . имя поля

Например, $S.name$ - поле name структуры S .

Для задания структур на языке Matlab можно использовать операторы присваивания, как показано в следующем примере:

```
>> S.im='Вова';
>> S.vozrast=5;
>> S.ves=20;
```

2.3. Формирование арифметических и логических выражений

Выражения в Matlab строятся из констант, переменных, стандартных функций, знаков операций и скобок.

В таблицах 2.1, 2.2 и 2.3 приводятся основные арифметические, логические операции, а также операции отношения.

Таблица 2.1 - Арифметические операции

Название	Знак операции	Пример
Сложение	+	$x+y$
Вычитание	-	$x-y$
Умножение	*	$x*y$
Возведение в степень	^	x^5
Обратное (справа налево) деление матриц	\	$x\backslash y$
Деление матриц слева направо	/	x/y

Таблица 2.2 - Операции отношения

Название	Знак операции	Пример
Равно	==	$x==y$
Не равно	~=	$x~=y$
Меньше	<	$x<y$
Больше	>	$x>y$
Меньше или равно	<=	$x<=y$
Больше или равно	>=	$x>=y$

Таблица 2.3 - Логические операции

Название	Знак операции	Пример
Логическое И	&	$x\&y$
Логическое ИЛИ		$x y$
Логическое НЕ	~	$\sim x$

В Matlab приоритет логических операций выше, чем арифметических, приоритет возведения в степень выше приоритетов умножения и деления, приоритет умножения и деления выше приоритета сложения и вычитания. Для изменения приоритета операций в математических выражениях используются круглые скобки. Степень вложения скобок не ограничивается.

Функция – это имеющий уникальное имя объект, выполняющий определенные преобразования своих аргументов и при этом возвращающий результаты этих преобразований. Функции в общем случае имеют список аргументов (параметров), заключенный в круглые скобки. Имена встроенных (стандартных) функций записываются строчными буквами.

В таблице 2.4 приводятся некоторые часто употребляемые математические функции.

Таблица 2.4 - Основные математические функции

Описание	Имя	Описание	Имя
Абсолютная величина	abs(x)	Косинус	cos(x)
Экспонента	exp(x)	Синус	sin(x)
Натуральный логарифм	log(x)	Тангенс	tan(x)
Знак числа	sign(x)	Котангенс	cot(x)
Десятичный логарифм	log10(x)	Арккосинус	acos(x)
Корень квадратный	sqrt(x)	Арксинус	asin(x)
Вещественная часть комплексного числа	real(x)	Арктангенс	atan(x)
Мнимая часть комплексного числа	imag(x)	Арккотангенс	acot(x)

Приведем примеры записи арифметических выражений в системе Matlab.

Запись в математике:	Запись в Matlab:
$\frac{\cos x^2}{x + \sin^3 x} + e^{-2.1}$	cos(x^2)/(x+sin(x)^3)+exp(-2.1)
$\frac{1.2 - 2x}{\lg(x + 3.2)} - \sqrt{ x - 5.8 }$	(1.2-2*x)/log10(x+3.2)-sqrt(abs(x-5.8))

3. МАТРИЧНЫЕ ВЫЧИСЛЕНИЯ И ЧИСЛЕННЫЙ АНАЛИЗ В MATLAB

3.1. Функции и операции по обработке матриц

В системе MatLab основной единицей данных является матрица, поэтому система имеет обширный набор стандартных функций и операций по обработке матриц, который позволяет:

- формировать новые матрицы стандартного вида;
- выполнять матричные арифметические операции;
- вычислять матричные характеристики и математические функции.

Для формирования новых матриц стандартного вида применяются следующие системные функции:

rand(M,N) – формирует прямоугольную матрицу размерностью M×N, элементами которой являются случайные числа в интервале (0.0; 1.0), функция rand без параметров формирует одно случайное число в том же интервале.

ones(M,N) формирует единичную матрицу размерностью M×N.

zeros(M,N) формирует матрицу размерностью M×N, состоящую из нулей.

diag(V) создает диагональную матрицу, в которой элементы вектора V являются элементами главной диагонали.

Матричные арифметические операции представлены следующими:

A+B , **A-B** матричное сложение и вычитание. Оба операнда этой операции должны иметь одинаковую размерность, если они являются матрицами. Один из операндов может быть скалярной величиной.

A*B матричное умножение. Операция выполняется по правилам матричного умножения, число столбцов матрицы A должно быть равно числу строк матрицы B .

A \ B левое деление матриц. Осуществляет решение системы линейных алгебраических уравнений $A*X=B$. Число столбцов A должно быть равно числу строк B .

A / B правое деление матриц. Осуществляет решение системы линейных алгебраических уравнений $X*A=B$.

X ^ P возведение матрицы в степень. Эта операция при скалярном значении P возводит квадратную матрицу X в степень P . Если X – скалярная величина, а P – квадратная матрица, то X^P возводит X в матричную степень P . Эта операция является ошибочной, если оба операнда – матрицы.

В MatLab существуют матричные операции, которые выполняются над каждым элементом матрицы, это такие операции, как:

- .* поэлементное матричное умножение.
- .\ поэлементное левое деление матриц.
- ./ поэлементное правое деление матриц.

Оба операнда этих операций должны иметь одинаковую размерность, или один из них должен являться скалярной величиной.

Операция «апостраф» ' вычисляет комплексно сопряженную транспонированную матрицу.

Операция «точка апостраф» .' вычисляет транспонированную матрицу.

Ниже приведены примеры выполнения матричных арифметических операций.

Формирование новой матрицы	Матричные операции	Поэлементные матричные операции
>> rand(2,3) ans = 0.9218 0.1763 0.9355 0.7382 0.4057 0.9169	>> x=[2 3 4; 5 6 7]; >> y=[9 8; 7 6; 5 4]; z=x*y z = 59 50 122 104	>> x=[2 3 4; 5 6 7]; >> y=[9 8 7 ; 6 5 4]; >> z=x .*y z = 18 24 28 30 30 28
>> ones(2,3) ans = 1 1 1 1 1 1	>> x=[2 3; 5 6]; x^3 ans = 158 201 335 426	>> f=y./x f = 4.5000 2.6667 1.7500

```

>> x=[2 3 4; 5 6 7];          1.2000  0.8333  0.5714
>> t=[3; 6; 9];              >> w=y.'
>> z=x*t                      w =
z =                             9  6
    60                          8  5
   114                          7  4

```

Система содержит стандартные функции, позволяющие вычислять различные характеристики матриц:

- det(A)** вычисляет определитель матрицы;
- trace(A)** вычисление следа матрицы;
- rank(A)** вычисление ранга матрицы;
- inv(A)** вычисление обратной матрицы.

Ниже приведены примеры вычисления этих характеристик для матриц A и B.

Определитель матрицы	Обратная матрица
<pre> >> x=[2 3 4; 5 6 7; 1 2 2]; >> d=det(x) d = 3 </pre>	<pre> >> x=[2 3 4; 5 6 7; 1 2 2]; >> inv(x) ans = -0.6667 0.6667 -1.0000 -1.0000 0 2.0000 1.3333 -0.3333 -1.0000 </pre>

Над массивами можно выполнять различные операции, заданные системными функциями.

- max(A)** вычисление максимального элемента массива;
- min(A)** вычисление минимального элемента массива;
- sum(A)** вычисление суммы элементов массива;
- prod(A)** вычисление произведения элементов массива;
- mean(A)** вычисление среднего значения элементов массива.

Порядок применения этих функций и результаты их выполнения рассмотрены на примерах.

Минимум и максимум	Сумма и произведение	Среднее значение
<pre> >> x=[2 3 4; 55 6 7; 1 12 2]; >> m=max(x) m = 55 12 7 >> m=max(max(x)) m = 55 >> z=min(x) z = 1 3 2 </pre>	<pre> >> x=[2 3 4; 5 6 7; 1 2 2]; >> s=sum(x) s = 8 11 13 >> s1=sum(sum(x)) s1 = 32 >> z=prod(x) z = 10 36 56 </pre>	<pre> > x=[2 3 4; 5 6 7; 2 3 7]; >> z=mean(x) z = 3 4 6 >> z1=mean(mean(x)) z1 = 4.3333 </pre>

Матрицы могут быть аргументами стандартных математических функций системы, применяемых не к отдельным элементам матрицы, а к матрице целиком. Ниже приведен перечень этих функций.

- expm(A)** вычисление матричной экспоненты от матрицы A;
- sqrtn(A)** вычисление квадратного корня из матрицы A;
- logm(A)** вычисление логорифма от матрицы A;
- funm(A)** вычисление функции от матрицы A.

В данном разделе приведены только основные матричные операции и функции, полный их перечень можно найти в [1] или обратившись к справке командой help funfun.

3.2. Численное интегрирование и дифференцирование в MatLab

Одной из распространенных задач численного анализа является вычисление определенных интегралов. В MatLab для этой цели используются различные методы, реализованные в следующих стандартных функциях.

- traz** вычисление интеграла методом трапеций;
- quad** вычисление интеграла методом Симпсона;
- quad8** вычисление интеграла методом Ньютона-Котеса;
- dblquad** вычисление интеграла по областям.

Подынтегральная функция может быть описана несколькими способами: в обычном виде $y=f(x)$; в форме символьной строки; в виде строкового объекта; с использованием M-файла.

Функция traz имеет следующий общий вид:

traz(X, Y),

где X – переменная, указывающая пределы интегрирования (диапазон, вектор); Y – подынтегральная функция. Если параметр X опущен, то шаг изменения аргумента считается равным 1. Например, нужно вычислить интеграл функции $y= \sin(x)$ в пределах от 1 до 2. Это можно сделать последовательностью команд вида:

```
>> x=1:0.1:2; y=sin(x);
>> trapz(x,y)
ans =
    0.9557
```

Функция quad имеет следующий общий вид:

quad(F, A, B, TOL),

где F – имя подынтегральной функции f(x); A, B – пределы интегрирования; TOL – абсолютная погрешность вычисления, необязательный параметр, если он не задан, то он имеет значение 10^{-6} . Имя функции может быть задано несколькими способами: выбором из списка простейших функций со знаком @, например, @sin; описанием функции как строки символов; описанием функции как строкового объекта с помощью стандартной функции

inline (S),

где S – символьное представление функции $f(x)$. Еще один способ задания функции $f(x)$ – создание М-файла, он будет рассматриваться во второй части практического пособия.

Ниже приведены примеры вычисления определенного интеграла.

Описание функции как строки символов	Описание функции как строкового объекта	Описание функции со знаком @
<code>>> quad('2*x+sin(x)',0,3)</code> ans = 10.9900	<code>>>y=inline('2*x+sin(x)');</code> <code>>> quad(y,0,3)</code> ans = 10.9900	<code>>> quad(@cos,0,3)</code> ans = 0.1411

Вычисление производной N -го порядка выполняется командой
diff(Y, N)/ diff(X, N),

где Y – дифференцируемая функция, X – переменная дифференцирования, N – порядок производной.

Ниже приведены примеры вычисления производных функций в точках и в диапазонах.

<code>>> x=1:0.1:2; y=sin(x);</code> <code>>> p=diff(y,1)/diff(x,1)</code> p = 0.0678	<code>>> x=1:0.1:2; y=sin(x);</code> <code>>> p2=diff(y,2)/diff(x,2)</code> p2 = -1.5337e+011
--	--

3.3. Решение уравнений и систем уравнений

Для решения уравнения $Y(x)=0$, где $Y(x)$ является полиномом, используется стандартная функция **roots** следующего общего вида:

roots(a),

где a – вектор коэффициентов перед неизвестными полинома размерностью $n+1$ (n – порядок полинома). Результатом работы этой функции будет вектор корней полинома размерностью n . Ниже приведены примеры поиска корней полиномиальных уравнений.

Решение уравнения $3x^3+x^2-10x-8=0$	Решение уравнения $3x^3+x^2-10x+8=0$	Решение уравнения $x^2-5x-3=0$
<code>>> v=[3 1 -10 -8];</code> <code>>> roots(v)</code> ans = 2.0000 -1.3333 -1.0000	<code>>> v=[3 1 -10 8];</code> <code>>> roots(v)</code> ans = -2.2936 0.9801 + 0.4495i 0.9801 - 0.4495i	<code>>> roots([1 -5 -3])</code> ans = 5.5414 -0.5414

Для решения нелинейного уравнения $f(x)=0$ используется стандартная функция **fzero** следующего упрощенного общего вида:

fzero(f, x0),

где f – имя функции $f(x)$ исходного уравнения, x_0 – начальное приближение корня. Если начальное приближение корня – скалярная величина, то результатом работы функции является один вещественный корень, если начальное приближение – массив или диапазон, то результат – вектор со множеством корней. Ниже приведены примеры использования функции `fzero`.

<p>Решение уравнения $\cos(x) - 0.1x = 0$ <code>>>fzero('cos(x)-0.1*x',1)</code> <code>ans =</code> 1.4276</p>	<p>Решение уравнения $\sin(x) - 0.5 = 0$ в диапазоне x <code>>> x0=2;y=inline('sin(x)-0.5'); fzero(y,x0)</code> <code>ans =</code> 2.6180</p>
--	--

Одним из удобных способов решения систем линейных уравнений является применение операции левого матричного деления. Если задана система линейных алгебраических уравнений вида:

$$AX=B,$$

где A – матрица коэффициентов перед неизвестными системы, B – вектор свободных членов, то решение системы может быть найдено в виде:

$$X=A \setminus B$$

То же самое решение может быть получено с помощью обратной матрицы, например:

$$X=\text{inv}(A)*B$$

Ниже приведены примеры решения систем линейных уравнений.

<p>Решить систему линейных уравнений $3x+5y=2$ $6x-8y=-3$ <code>>> A=[3 5;6 -8]; B=[2 ; -3];X=inv(A)*B</code> <code>X =</code> 0.0185 0.3889</p>	<p>Решить систему линейных уравнений $3a+7b=-13$ $5a+b=1$ <code>>> A=[3 7;5 1]; B=[13; 1];X=A\B</code> <code>X =</code> -0.1875 1.9375</p>
---	---

4. ГРАФИКА В МАТЛАВ

4.1. Двумерные графики

Графические объекты в Matlab строятся в специальном графическом окне (`figure`). Одновременно может быть открыто несколько таких окон, каждому из которых присваивается номер. Для перехода к имеющемуся окну с номером N или открытия нового графического окна необходимо ввести команду **figure (N)**. Кроме того, первое обращение к графической команде автоматически вызывает появление графического окна, которому присваивается первый номер.

Для построения графиков функций одной переменной в декартовой системе координат используются различные формы команды **plot**, которая рисует графики функций по ряду точек, соединяя их отрезками прямых,

т.е. осуществляя линейную интерполяцию функции в интервале между смежными точками.

- **plot(X,Y)** – строит график функции, координаты точек которой берутся из векторов **одинакового размера** X и Y. Если X или Y – матрица, то строится семейство графиков по данным, содержащимся в столбцах матрицы.

Ниже приводится пример, иллюстрирующий построение графика косинусоиды на интервале $[-2\pi, 2\pi]$, значения аргумента которой содержатся в векторе T, а соответствующие значения функции содержатся в векторе Z.

```
>>T=-2*pi/10:2*pi; Z=cos(T);
>>plot(T,Z)
```

Результатом выполнения этой группы команд будет графическое окно с выведенным в нем графиком функции, показанное на рисунке 4.1.

Группа команд

```
>> X=-2*pi/10:2*pi;
>> Y=[sin(X); cos(X)];
>> plot(X,Y)
```

осуществляет построение графиков двух функций в одном графическом окне.

- **plot(Y)** – строит график зависимости, значения ординат которой берутся из вектора Y, а значения абсцисс представляют собой индексы соответствующих элементов вектора.

- **plot(X,Y,S)** – аналогична команде **plot(X,Y)**, в которой формат линии графика можно задавать с помощью строковой константы S. Символы, которые могут использоваться в параметре S, приведены в таблице 4.1.

Таблица 4.1 - Символы, определяющие формат линий графика

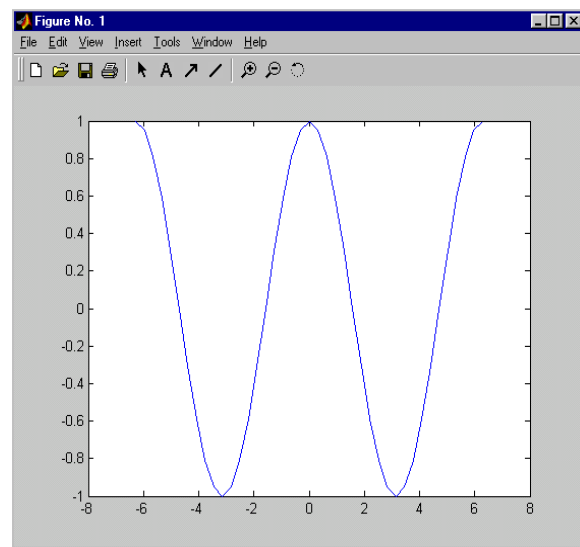


Рисунок 4.1 - График косинусоиды.

Тип линии		Тип точки		Цвет линии	
-	Сплошная	.	Точка	y	Желтый
:	Двойной пунктир	o	Окружность	m	Фиолетовый
-.	Штрих-пунктир	x	Крест	c	Голубой
--	Штриховая	+	Плюс	r	Красный
		*	Звездочка	g	Зеленый
		s	Квадрат	b	Синий
		d	Ромб	w	Белый
		v	Треугольник вниз	k	Черный

Например, для отображения графика функции $y=0.02x^3$ штриховой линией красного цвета с узловыми точками в виде ромбов можно выполнить следующую последовательность команд:

```
>> x=-5:0.5:2; plot(x,0.02*x.^3,'--dr')
```

- **plot(X1,Y1,S1,X2,Y2,S2,...)** – строит графики нескольких функций на одном рисунке, где X1,Y1 – абсциссы и ординаты 1-й кривой, X2, Y2 – абсциссы и ординаты 2-й кривой и т.д. Дополнительные параметры S1, S2 и т.д. позволяют задать стиль линий графиков.

Рассмотрим пример построения в окне с номером 2 графиков трех функций с различным форматом представления каждой из них:

```
>> figure (2)
```

```
>> x=-2*pi:pi/10:2*pi;
```

```
>> f1=cos(x); f2=sin(x); f3=sin(x+0.5).^3; plot(x,f1,'-k',x,f2, '-.xk', x, f3, '--ok')
```

Результат выполнения этих команд отражен на рисунке 4.2.

Для создания в графическом окне нескольких графических областей для вывода графиков применяется команда

subplot(m, n, p),

которая разбивает графическое окно на $m \times n$ областей, где m – число областей по вертикали, n – число областей по горизонтали, p – номер области, в которую будет выводиться текущий график (области отсчитываются последовательно по строкам).

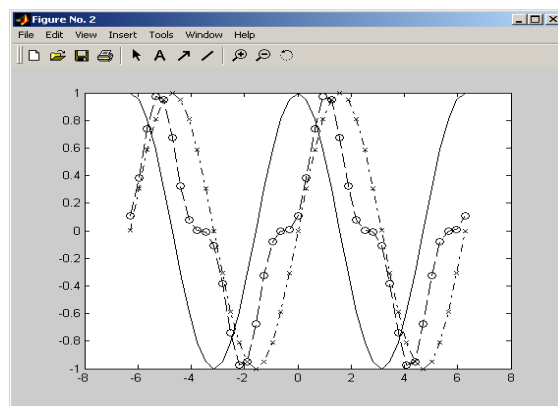


Рисунок.4.2. Построение графиков трех функций на одном рисунке разным стилем линий.

4.2. Трехмерные графики

Для построения трехмерных графиков требуется создать две матрицы с повторяющимися строками и столбцами, в которых хранятся аргументы функции. Для создания таких массивов служит функция **meshgrid**, которая обычно используется совместно с командами построения графиков трехмерных поверхностей.

[X,Y]=meshgrid(x,y) – преобразует векторы x и y в матрицы X и Y , которые могут быть использованы для вычисления функции двух переменных и построения трехмерных графиков. Строки выходного массива X являются копиями вектора x , а столбцы матрицы Y – копиями вектора y . Например,

```
>> [X,Y]=meshgrid([2 3 5],[4 6 8]);
```

После выполнения команды

X =	2	3	5	Y =	4	4	4
	2	3	5		6	6	6

Для построения графиков поверхностей используются следующие команды:

- **plot3(X,Y,Z)** – строит изображение трехмерной поверхности, соединяя узловые точки отрезками прямых. X , Y , Z – матрицы одинакового размера.
- **plot3(X,Y,Z,S)** – аналогична предыдущей команде, но со спецификацией стиля линий и точек. Здесь S – строковая константа, которая может содержать символы из таблицы 4.1.1.
- **mesh(X,Y,Z)** – выводит в графическое окно сетчатую поверхность $Z(X, Y)$. В этом случае используется функциональная окраска, при которой цвет задается высотой поверхности.
- **meshc(X,Y,Z)** – строит сетчатый график поверхности и изображение ее проекции в виде линий равного уровня.
- **meshz(X,Y,Z)** – строит график поверхности из многочисленных столбцов.
- **surf(X,Y,Z)** – строит цветной сетчатый график поверхности по данным матриц X , Y и Z с функциональной закраской ячеек цветом, который соответствует высоте поверхности.
- **surfc(X,Y,Z)** – аналогична команде **surf**, но обеспечивает дополнительное построение контурного графика проекции фигуры на опорную плоскость.
- **surf1(X,Y,Z)** – аналогична команде **surf**, но строит график поверхности с подсветкой от источника света.
- **contour3(X,Y,Z)** – строит контурные линии для поверхности, полученные при расслоении трехмерной фигуры рядом секущих плоскостей, расположенных параллельно опорной плоскости фигуры.

В приведенном ниже примере показаны возможности применения описанных команд для построения графика поверхности $Z = \sin(X) \cos(X)$.

```
>> [X,Y]=meshgrid(-3:0.1:3,-3:0.1:3); Z=sin(X).*cos(X);
>> subplot(3,2,1), plot3(X,Y,Z) % Рисунок 4.3 (а)
>> subplot(3,2,2), mesh(X,Y,Z) % Рисунок 4.3 (б)
>> subplot(3,2,3), surf(X,Y,Z) % Рисунок 4.3 (в)
>> subplot(3,2,4), surfc(X,Y,Z) % Рисунок 4.3 (г)
>> subplot(3,2,5), meshz(X,Y,Z) % Рисунок 4.3 (д)
>> subplot(3,2,6), contour3(X,Y,Z) % Рисунок 4.3 (е)
```

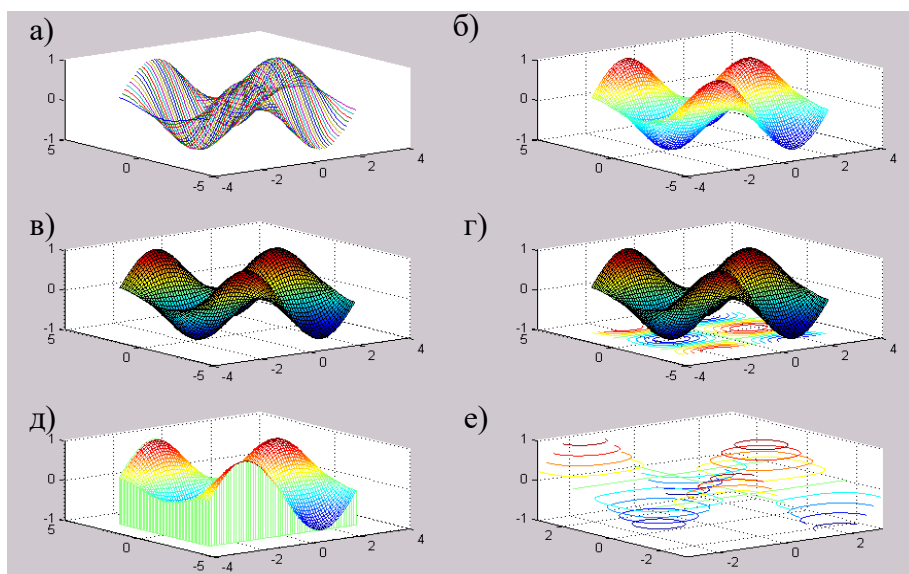






Рисунок 4.3 - Пример построения графиков поверхности различными командами.

4.3. Форматирование графиков

Система Matlab предоставляет возможность настраивать и корректировать свойства графиков как средствами интерфейса графического окна, так и путем задания соответствующих графических команд и параметров. В таблице 4.3.1. приведены некоторые простейшие приемы форматирования графиков.

Таблица 4.2 - Форматирование графиков

Действие	Средства графического интерфейса	Команда
Переход в режим редактирования.	Щелкнуть по кнопке Edit Plot в панели инструментов графического окна. 	
Форматирование линий и маркеров опорных точек графиков.	В режиме редактирования дважды щелкнуть по линии графика левой клавишей мыши. В появившемся окне Property Editor-Line установить все необходимые параметры линии (толщину, стиль, цвет и т.д.).	plot(X,Y,S), plot3(X,Y,Z,S) (Описание команд приводится в п.4.1 и п.4.2)
Форматирование осей графиков.	В режиме редактирования дважды щелкнуть по оси графика. В появившемся окне Property Editor-Axes установить все необходимые параметры осей.	axes – управляет свойствами осей. grid - включает и отключает координатную сетку. xlabel(S), ylabel(S), zlabel(S) – устанавливает надписи возле

	Заголовок графика и метки осей можно также с помощью команд Insert Title , Insert Xlabel , Insert Ylabel главного меню графического окна.	осей. Здесь S – строковая константа или переменная. title(S) – выводит заголовок графика
Нанесение надписей прямо на график.	Щелкнуть по кнопке Insert Text  , зафиксировать место надписи щелчком мыши и ввести нужный текст.	text(X,Y,S) – добавляет в двумерный график текст, заданный строкой S , так что начало текста расположено в точке с координатами (X, Y) . text(X,Y,Z,S) - добавляет текст в трехмерный график.
Нанесение линий и линий со стрелками прямо на график	Щелкнуть по одной из кнопок Insert Arrow или Insert Line  . Установить указатель мыши в нужное место графика и при нажатой левой клавише мыши нарисовать линию.	
Построение легенды	В главном меню графического окна выбрать команду Insert , а затем команду legend .	legend(S1,S2,S3,...) – добавляет к текущему графику легенду с пояснениями в виде строк, указанных в списке параметров.
Вывод шкалы цветов	В главном меню графического окна выбрать команду Insert , а затем команду Colorbar .	colorbar('vert') , colorbar('horiz') – выводит вертикальную или горизонтальную шкалу цветов.
Вращение графика	Щелкнуть по кнопке Rotate 3D  и вращать график с помощью мыши (может применяться также для двумерных графиков).	rotate3d – задает поворот трехмерной фигуры.