КЫРГЫЗСКО-РОССИЙСКИЙ СЛАВЯНСКИЙ УНИВЕРСИТЕТ

ЕСТЕСТВЕННО-ТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра прикладной математики и информатики

УДК 004 ББК 32.973-01 К 96

> Рецензенты: д-р. техн. наук, доцент *Н.М. Лыченко* канд. техн. наук, доцент *Е.Л. Миркин*

Рекомендовано к изданию кафедрой прикладной математики и информатики

Н.Л. КУЧЕРЕНКО

MATLAB:

ТИПЫДАННЫХ, МАССИВЫ, РАБОТА С ФАЙЛАМИ, ГРАФИКА, ИНТЕРФЕЙС

Учебно-методическое пособие

Кучеренко Н.Л.

К 96 МАТLАВ: ТИПЫ ДАННЫХ, МАССИВЫ, РАБОТА С ФАЙЛАМИ, ГРАФИКА, ИНТЕРФЕЙС: Учебно-методическое пособие. – Бишкек: КРСУ, 2011. – 95 с.

ISBN 978-9967-05-766-1

В пособии приведены структура системы MATLAB, список операторов и функций. Рассмотрена работа с массивами, файлами, графика, а также вопросы конструирования интерфейса с помощью графических объектов MATLAB.

К 2404090000-11

УДК 004 ББК 32.973-01

Бишкек 2011

ISBN 978-9967-05-766-1

© КРСУ, 2011

СОДЕРЖАНИЕ

ВВЕДЕНИЕ
Запуск и режимы работы системы МАТLAB 7
Справочная и демонстрационная системы МАТLAB.
Операторы и функции
Типы данных
Работа с массивами
Массивы записей 15
Массивы ячеек
Работа с файлами и переменными рабочей области
ГРАФИЧЕСКИЕ СРЕДСТВА СИСТЕМЫ MATLAB
ДВУМЕРНЫЕ ГРАФИКИ
График в линейном масштабе26
График в полярных координатах
ТРЕХМЕРНЫЕ ГРАФИКИ
Построение линий и точек в трехмерном пространстве
Формирование сетки на плоскости
в виде двумерных массивов X и Y
Трехмерная сетчатая поверхность
Затененная сетчатая поверхность
Затененная поверхность с подсветкой 34
Масштабирование осей
Нанесение сетки
Управление режимом сохранения
текущего графического окна 36
Разбиение графического окна на подокна
Управление масштабом графика
Палитра цветов

Установление соответствия между палитрой цветов	
и масштабированием осей	
Затенение поверхностей	39
Линии уровня для трехмерной поверхности	
Заголовки для двух- и трехмерных графиков	41
Обозначения осей	41
Добавление текста к текущему графику	41
Пояснение к графику	
Шкала палитры	
График с указанием интервала погрешности	
Построение гистограммы	45
Дискретный график	
Ступенчатый график	
Гистограмма в полярных координатах	
Графики векторов	
Поле градиентов функции	
Движение точки по траектории	
Закраска многоугольников	
Сечения функции трех переменных	
Трехмерная поверхность	
Вычисление матрицы управления углом просмотра	57
Управление положением точки просмотра	
ИНТЕРФЕЙС И ОБЪЕКТЫ МАТLAB	59
Создание приложений в среде GUIDE	63
Приложение 1	67
ЛИТЕРАТУРА	

введение

МАТLAB – среда проектирования инженерных приложений и научных расчетов, созданная фирмой The MathWorks Inc. МАТLAB – мощное и универсальное средство решения задач из самых различных областей человеческой деятельности: оптимизационные задачи, матричный анализ, задачи математической физики, картография, нейронные сети, нечеткая логика и др.

Язык системы MATLAB сравним с языком BASIC, который постепенно трансформировался в Visual Basic (ядро системы MATLAB 6 практически полностью воспроизведено на языке Visual Basic for Applications). Язык MATLAB является языком четвертого поколения (4GLs), не уступающим по своей эффективности процедурным языкам третьего поколения. Это достигается применением механизма ускорения Just-In-Time-Accelerator. Огромное преимущество MATLAB – это открытость кода.

Основное отличие системы MATLAB от аналогичных систем таких, как MathCAD, Maple, Mathematica – это векторная обработка данных, которая обеспечивает высокую скорость вычислений. Вторая важная особенность – модульный принцип построения системы, который привел к созданию большого числа пакетов прикладных программ (ППП). В настоящее время семейство MATLAB включает в себя более 50 ППП, а также свыше 250 приложений.

В состав системы входят (см. рис. 1):

- 1. Application Development Tools инструментальные средства (ИС) разработки приложений (ИС Notebook, ИС Excel Link, ИС Guide, ИС Runtime Server)
- 2. Standalone Applications независимо исполняемые приложения
- 3. Data Access Tools инструментальные средства доступа к данным, включают в себя программное обеспечение для работы с базами данных
- 4. **Data Sources** сбор средств в режиме реального времени от агенства Bloomberg корпорации IDC и серверов Yahoo
- 5. Student Products студенческие версии
- 6. Simulink подсистема моделирования динамических процессов
- 7. **Blocksets** наборы блоков динамических систем, относящихся к разным приложениям
- 8. Stateflow моделирование событий, возникающих при эксплуатации технической системы
- 9. Code Generation Tools инструментальное средство генерации кодов, позволяющее создавать независимо исполняемые коды на С или Ada.





Особое место в составе программных средств MATLAB занимают пакеты прикладных программ (ППП) для поддержки разработки приложений в конкретных предметных областях.

Математика представлена следующими пакетами:

- 1. Symbollic Math Toolbox символьная математика
- 2. Extended Symbolic Math Toolbox символьная математика с ядром системы MapleV
- 3. Optimization Toolbox оптимизация
- 4. Partial Differential Equation Toolbox решение уравнений в частных производных
- 5. Statistics Toolbox статистика
- 6. Curve Fitting Toolbox аппроксимация и сглаживание экспериментальных данных
- 7. Spline Toolbox сплайновая аппроксимация
- 8. Fuzzy Logic Toolbox нечеткая логика
- 9. Neural Network Toolbox анализ и проектирование нейронных сетей

Системы связи и передачи сигналов:

- 1. Communications Toolbox анализ систем связи и коммуникаций
- 2. Communications Blockset набор блоков для моделирования систем связи и коммуникаций
- 3. CDMA Reference Blockset набор блоков стандарта CDMA

Обработка сигналов и изображений:

- 1. Signal Processing Toolbox обработка сигналов
- 2. Filter Design Toolbox проектирование фильтров
- 3. Fixed-Point Blockset набор блоков для реализации операций с фиксированной точкой
- 4. **DSP(Digital Signal Processing)** набор блоков цифровой обработки сигралов и данных
- 5. Image Processing Toolbox обработка изображений
- 6. Wavelet Toolbox импульсная декомпозиция сигналов и изображений

Команды, позволяющие определить версию и просмотреть список ППП

>> version	версия системы Matlab
>>version – java	версия виртуальной машины Java
>>ver	список и версия используемых ППП
>>ver <имя ППП>	версия заданного ППП

ЗАПУСК И РЕЖИМЫ РАБОТЫ СИСТЕМЫ МАТLAB

MATLAB запускается с рабочего стола Windows или через

Start→Programs→ MATLAB. В результате на дисплее открывается рабочий стол системы (см. рис. 2)





Все настройки системы, касающиеся рабочего стола, организации кэш-памяти при работе с ППП, организации вывода на печать, управления файлами и др. устанавливаются в окне File—Preferences

В системе MATLAВ возможны два режима работы:

- 1. режим командной строки
- 2. режим редактора М-файлов

Пример: (режим командной строки):

- >>A=2;
- >>a=3; %MATLAB отличает прописные и строчные буквы
- >>b=a+A % точка с запятой не ставится, если результат необходимо отобразить на экране

Результат выведется в командное окно :>>b=5

Эту же программу можно запустить из редактора, для этого требуется

создать новый файл: **File→New →М-файл**

и запустить его на выполнение из редактора:

Debug→Save and Run

Если файл уже записан на диск (например, с именем file1.m), то его можно запустить на выполнение как из редактора: Debug→Run, так и из командной строки, набрав его имя :

>>file1₊

Необходимо так же помнить:

- 1. чтобы запускаемый файл находился в установленной текущей директории (Current Directory)
- 2. чтобы в начале работы нового приложения очищались все переменные рабочей области: clear all;

СПРАВОЧНАЯ И ДЕМОНСТРАЦИОННАЯ СИСТЕМЫ МАТLAB. ОПЕРАТОРЫ И ФУНКЦИИ

В состав системы MATLAB входит мощная подсистема справки Help и демонстрационная подсистема Demo. Они имеют стандартный для Windows-приложений характер, поэтому работа с ними не приводится в данном пособии.

Просмотр операторов, конструкций языка и системных функций возможен просто из командной строки:

>>help <имя >

<имя> может иметь следующие значения:

- general команды общего назначения
- **ор***s* операторы и специальные символы;
- lang конструкции языка программирования;
- strfun строковые функции;
- iofun функции ввода/вывода;
- timefun функции времени и дат;
- datatypes типы и структуры данных;
- **arith** арифметические операции;
- **relop** операции отношений;
- slash деление матриц;

- winfun использование системных интерфейсов Windows (COM/DDE);
- **audiovideo** аудио и видео поддержка;
- imagesci работа с изображениями и ввод/вывод научных данных;
- function_handle дескрипторы функций.
- Основные математические и матричные функции:
 - elmat команды создания элементарных матриц и операций с ними;
 - elfun элементарные математические функции;
 - **specfun** специальные математические функции;
 - matfun матричные функции линейной алгебры;
 - datafun анализ данных и преобразования Фурье;
 - polyfun полиномиальные функции и функции интерполяции;
 - **funfun** функции функций и функции решения обыкновенных дифференциальных уравнений;
 - sparfun функции разреженных матриц.

Команды графики:

- graph2d команды двумерной графики;
- graph3d команды трехмерной графики;
- specgraph команды специальной графики;
- graphics дескрипторная графика пользовательского интерфейса
- **uitools** инструментальные средства пользовательского интерфейса.

Например, строка в командном окне

>>help elfun – позволяет вывести список элементарных функций МАТ-LAB

>>help if – справка об операторе if

Список операторов и функций системы МАТLAB приведен в приложении 1.

ТИПЫ ДАННЫХ

В системе MATLAB определено 15 базовых типов данных, каждый из которых является тем или иным видом массива, другими словами, все объекты системы MATLAB являются массивами. Двумерный массив минимального размера – это пустой массив размера 0*0, максимальный размер и размерности многомерного массива не ограничены. Все вычисления в системе MATLAB производятся с удвоенной точностью. Представленные ниже на схеме типы данных (кроме типа user class) являются встроенными, для них можно переопределить методы обработки в рамках объектно-ориентированного подхода, реализованного в системе MATLAB





Классы ARRAY и NUMERIC не приписываются никаким данным и применяются только в целях классификации.

Тип logical – логический массив, т.е. массив значений из 1(true) и 0 (false); любое ненулевое значение преобразуется в значение 1. Только двумерный логический массив может быть разряженным (sparse).

Тип char – массив символов или строка; каждый символ длиной 16 бит. **Тип NUMERIC** – массивы 8,16, 32 и 64-битовых целых чисел со знаком (int) или без знака (uint).

Тип single – массив чисел обычной точности.

Тип double – массив чисел удвоенной точности.

Тип cell – массив ячеек, позволяет объединять массивы различных типов и размеров в один массив.

Тип structure – массив записей, предназначен для хранения разнородных данных. Данные хранятся в специальных полях, имеющих имена.

Тип function handle – дескрипторы функций, может быть передан в списке входных аргументов и использован для вычислений при помощи функции feval.

Тип java class – тип данных Java, можно использовать уже определенные в Java API – типы, а также собственные типы, написанные на Java. **Тип user class** – тип данных, определяемых пользователем.

РАБОТА С МАССИВАМИ

Векторные вычисления позволяют упростить работу с векторами и матрицами. В MATLAB существует несколько способов задания массива и многочисленные функции работы с массивами (см. приложение 1). Рассмотрим некоторые из них:

A=[];	пустой массив
M=1:5	M = [1 2 3 4 5]
X=1:0.1:2	X=[1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2]
Y=[2 3 7]	Y=[2 3 7]
Z=[1 4 7; 5 0 2]	$\mathbf{Z} = \begin{bmatrix} 1 & 4 & 7 \\ 5 & 0 & 2 \end{bmatrix}$
B=[1 2]; C=[B B+1]	C=[1 2 2 3]
B=[1 2]; D=B+3	D=[4 5] сложение вектора с числом
A=[1 4]; S=A'	$S = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$ транспонирование
F=[2 4; 5 6]; E=[0 1]'; G=F*E	$G = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$ умножение матриц
F=[2 4 5]; P=F.*2	Р=[4 8 10] поэлементное умножение на
	число
F=[2 4 5]; P=[3 5 0]; H=F.*P	H=[6 20 0] поэлементное умножение
	векторов
Y=[2 3 4]; p=Y.^2	р=[4 9 16] поэлементное возведение в
	степень
W=[1 2 3]; D=12./W	D=[12 6 4] деление числа на элементы
	вектора
R=[pi pi/2]; G=sin(R)	G=[0 1] функция от массива
T=[2 3]; U=[1 2]; k=sum(T.*U)	k=8 скалярное произведение векторов
	$\lceil 10 \rceil$
M=[2; 3; 1]; N=[1; 2; 4]; p=cros	$\mathbf{s}(\mathbf{M},\mathbf{N}) \qquad \mathbf{p} = \begin{vmatrix} \mathbf{p} \\ -7 \end{vmatrix}$

векторное произведение векторов

A=[1 2 3]; h=prod(A)	h=6 произведение элементов массива		$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$
S=sum(A)	S=6 сумма элементов массива	Y=zeros(3);	Y=
Если F=[2 3 4; 1 5 9; 1 4 8] ;			
G = max(F)	G= [2 5 9] – максимальные элементы по		
	столбцам		нулей
GF=max(max(F))	GF=9 – максимальный элемент матрицы		$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$
d=det(F)	d=7 – вычисление определителя матрицы	I=eye(3);	$I = \begin{vmatrix} 0 & 1 & 0 \end{vmatrix}$ единичная матрица
r=rank(F)	r=3 – вычисление ранга матрицы		
Обращение к элементам масс	ива:		
	$\begin{bmatrix} 2 & 3 & 1 \end{bmatrix}$	U=ones(2,3);	$U = \begin{vmatrix} 1 & 1 & 1 \end{vmatrix}$ матрица из единиц
C=[2 3 1; 6 5 7; 2 4 9];	$C = \begin{vmatrix} 6 & 5 & 7 \end{vmatrix}$		
	$\begin{bmatrix} 2 & 4 & 9 \end{bmatrix}$	M=magic(3)	$M = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 7 & 7 \end{bmatrix}$ магическая матрина
H=C(2,3) или H=C(8)	H= C(2,3)=C(8)=7 элементы матрицы		
	нумеруются по столбцам.		$\begin{bmatrix} 4 & 9 & 2 \end{bmatrix}$
Удаление строк и столбцов:	Пусть C= $\begin{bmatrix} 6 & 5 & 7 \end{bmatrix}$, тогда	\mathbf{D} -rond $(2,3)$	$\mathbf{P} = \begin{bmatrix} 0.9501 & 0.6068 & 0.8913 \end{bmatrix}$
	2 4 0	\mathbf{K} -1 and (2,3)	$\mathbf{K} = \begin{bmatrix} 0.3501 & 0.0000 & 0.0013 \\ 0.2211 & 0.4000 & 0.5013 \end{bmatrix}$
			0.2311 0.4860 0.7621
F=C(1:2, :)	$F = \begin{bmatrix} 2 & 3 & 1 \end{bmatrix}$		
	6 5 7		матрица чисел, распределенных случай-
P=C(2, 2:end)	P=[57]		ным образом между нулем и единицей
	[3 1]		$\begin{bmatrix} -1.5937 & -0.3999 & 0.7119 \end{bmatrix}$
H=C(:,2:3)	$H = \begin{vmatrix} z & z \\ z & z \end{vmatrix}$	P=randn(3,4)	$P = \begin{bmatrix} -1 & 4410 & 0.6900 & 1.2902 \end{bmatrix}$
Характеристики массивов:			
d=size(H)	d=[3 2]; размерность массива Н		матрица нормально распределенных
L=length(H)	L=3; длина вектора, в данном случае		чисел
	размер столбца		
p=isequal(H)	p=0; истинно, если массив пустой и		
	ложно, если иначе		МАССИВЫ ЗАПИСЕИ
n=isnumeric(H)	n=1; истинно, если массив числовой		
w=isequal(H,C)	w=0; истинно, если массивы идентичны	Массив записе	й – это массив, имеющий именованные поля. Для
		работы с массивом заг	писей предназначены следующие функции:
Матрицы специального вида:		Struct	Создание массива записей
A=zeros(2,3);	А= 0 0 0 прямоугольная матри-	Fieldnames	Получить имена полей
~ / //		Getheld	получить содержимое поля
	иа из нулей	Setfield	у становить содержимое поля
		Rmfield	удалить поле

Isfield	Истинно, если это поле массива записей
Isstruct	Истинно, если это массив записей
Deal	Обмен данными между массивами

Например, создадим массив записей student, в котором будут храниться сведения о студентах.

Массив записей будет иметь следующие поля

	.name .begin	Петров А.С. 2006
student	.year .group	4 ЕПМ
	.test	5 4 5 4 5 5
		3 4 5 3 4 4

Массив записей можно сформировать 2 способами:

 с помощью оператора присваивания student.name='Петров А.С.'; student.begin=2006; student.year=4; student.group='ЕПМ'; student.test = [5 4 5 4 5 5;3 4 5 3 4 4];

При заполнении данных одной записи автоматически формируется структура массива (поля).

Чтобы расширить массив, достаточно добавить индекс к имени структуры. Например, вторую запись необходимо создать следующим образом:

student(2).name ='Иванов П.Р.'; student(2).begin = 2006; student(2).year= 3; student(2).group ='ИВТ'; student(2).test = [3 3 5 4 5 5;5 4 5 5 4 4];

 с помощью функции struct student=struct('name','Петров А.С.', 'begin', 2006,' year', 4,' group', 'ЕПМ', 'test',[5 4 5 4 5 5; 3 4 5 3 4 4]);

Доступ к полям массива записей	
str1=student(2).name	
str1=getfield(student,{2},'name')	
N=student(1).test(2,4)	
student(1).test(2,1)=3;	

str1=Иванов П.Р. str1=Иванов П.Р. N=3 присваивание значения

Поле массива записей может само включать другой массив записей.

МАССИВЫ ЯЧЕЕК

Массив ячеек – наиболее универсальный тип данных МАТLAB, это массив, элементами которого являются ячейки, содержащие любой тип массива, в том числе и массив ячеек. Массивы ячеек используются для работы с данными, которые различаются по размерам и типам, но по каким-либо причинам должны быть представлены единым массивом.

Для работы с массивом ячеек предназначены следующие функции:

cell	Создание массива ячеек
celldisp	Показать содержимое массива ячеек
cellplot	Показать графическую структуру массива ячеек
num2cell	Преобразовать числовой массив в массив ячеек
cell2struct	Преобразовать массив ячеек в структуру
struct2cell	Преобразовать структуру в массив ячеек
iscell	Истинно, если это массив ячеек
sort	Сортировка элементов массива
sortrows	Сортировка элементов массива с сохранением
	целостности строк
deal	Обмен данными между массивами

Например, массив ячеек S должен иметь следующую структуру:

cell 1,1 1 2 4 4 8 12	cell 1,2 'IVANOV'
cell 2,1 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.□ 2.9 3.□	cell 2,2 8+7i

Создать массив ячеек можно двумя способами:

1. используя оператор присваивания $S(1,1) = \{ [1 2 4; 2 8 12] \};$ S(1,2)={'IVANOV'}; $S(2.1) = \{2:0.1:3\};$ $S(2,2) = \{8+7i\};$ 2. используя функцию cell d=cell(2,3) – создать пустой массив ячеек размером 2x3 Данные из массива ячеек можно извлечь, используя индексацию содержимого:

 $G=S{1,1}(2,2)$ G=8name='IVANOV' name=S{1,2}

РАБОТА С ФАЙЛАМИ И ПЕРЕМЕННЫМИ РАБОЧЕЙ ОБЛАСТИ

Рабочая область (Workspace) - это область памяти, в которой размещены переменные системы MATLAB. Содержимое этой области можно просмотреть либо в окне Workspace, либо из командной строки, используя команды who и whos. Команда who выводит только список имен переменных в алфавитном порядке, а команда whos – информацию о размерах, объеме памяти массивов и типе переменной. Можно изменить значение переменной, открыв редактор данных из Workspace (двойной щелчок мышкой по изображению переменной), либо командой:

openvar<список переменных, которые нужно просмотреть или изменить>.

МАТLAB при работе оперирует следующими видами файлов:

.m – обычный текстовый файл, содержащий программу

.mat – двоичный файл. которой создает МАТLAB при записи рабочей области на диск

.fig - файл, создаваемый инструментальным средством GUIDE, содержит графический интерфейс приложения

.mdl – файл, содержащий модель SIMULINK

М-файлы бывают двух видов:

- 1. файлы-программы (Script-файлы), содержащие последовательность команд
- 2. файлы-функции (Function-файлы), содержащие функции, определяемые пользователем

Файлы-программы не имеют заголовка, начинаются сразу с вычислений и используются, если к данной программе не будет обращений из других модулей. М-файл необходимо оформить в виде файлафункции, если в дальнейшем к нему есть обращение из других модулей или его необходимо компилировать.

Файлы-функции начинаются со строчки:

function [список выходных параметров] = имя функции (список входных параметров)

При этом допускается использование системных констант:

varargin - переменное число входных параметров (Variable length input argument list)

varargout – переменное число выходных параметров

nargin – системная константа, число входных параметров функции nargout – системная константа, число выходных параметров функции Например, функция s sum, суммирует (переменная sum) элементы матрицы А, находит максимальный элемент в каждом столбце (вектор **k max**) и максимальный элемент матрицы (переменная **s max**):

function [sum, k max, s max]=s sum(A)

sum=0:

% п-размер матрицы А: количество строк ,количество столбцов n=size(A):

- for i=1:n(1).
 - for i=1:n(2),
 - sum=sum+A(i,j);
- end

end

k max=max(A);

s max=max(max(A));

Чтобы обратиться к функции s_sum, необходимо в командной строке (или модуле) задать матрицу и вызвать функцию s sum:

 $>>a = [6 \ 2 \ 9; \ 1 \ 8 \ 5];$ >>[s, k, m]=s sum(a)Результат выполнения программы: >>s = 31k = 6 8 9 m = 9

Если обратиться к функции без выходных параметров, то результатом выполнения программы будет значение первого выходного параметра, в нашем случае это сумма элементов матрицы:

>>s_sum(a)

>>ans=31

Большинство стандартных функций Matlab допускают обращение к ним с различным числом входных и выходных параметров. Например, к функции fminbnd – нахождения минимума функции, можно обратиться с 3-мя и более входными аргументами и одним или более выходными аргументами.

Рассмотрим файл-функцию с переменным числом входных параметров на следующем примере. Пусть на плоскости задано произвольное количество кругов (координаты центра и радиус: x1, y1, r1; x2, y2, r2...) и точка с координатами (px,py). Требуется определить лежит ли эта точка внутри какого-либо круга и вывести количество кругов, в которые попала точка и номера этих кругов.

Создадим файл-функцию point и запишем ее в одноименный файл:

function [where,varargout]=point(varargin)

 % Массив varargin всегда указывается последним!

 % выделение координат точки из первых двух ячеек

 px=varargin{1};

 py=varargin{2};

 % нахождение числа заданных кругов

 Ncircle=length(varargin)-2;

 for i=1:Ncircle

 Xcircle(i)=varargin{i+2}(1);

 Ycircle(i)=varargin{i+2}(2);

 Rcircle(i)=varargin{i+2}(3);

 end

 % Полагаем where=0 и Nc=0, т.е. пока нет ни одного нужного

круга

Nc=0;where=0;
% Перебор кругов в цикле for i=1:Ncircle
% dist – расстояние от точки до центра текущего круга dist=sqrt((px-Xcircle(i))^2+(py-Ycircle(i))^2);
% Сравнение расстояния с радиусом круга if dist<=Rcircle(i)

% Требуемый круг найден % where – логическая переменная where=1, если точка по-

пала хоть в один из заданных кругов

% Nc – счетчик кругов, в которые попала точка

% Вектор Nums содержит номера кругов, в которые попала точка

```
where=1; Nc=Nc+1; Nums(Nc)=i;
end
end
% количество выходных параметров определяется
% системной константой nargout
switch nargout
case(2)
varargout(1)={Nc};
case(3)
varargout(1)={Nc}
varargout(2)={Nums(:)}; end
```

Обратимся к фунции **point** с 3-мя выходными аргументами [w,n,nums]=point(0,0,[2,3,4],[5,6,1],[-1,-1,3])

Результат работы программы:

w=1 точка (0,0) попала хотя бы в один круг

n=2 точка (0,0) попала в два круга

nums = 1 3 точка попала в 1-й и 3-й круг.

Если введем более трех выходных параметров, то на экране появится сообщение: "Error using ==> point. Too many output arguments."

Дополним файл-функцию **point** проверкой входных аргументов на правильность (первые два параметра должны быть вещественными числами, а остальные – векторами длиной 3 и последний аргумент каждого вектора (радиус) должен быть больше нуля). В конце файла выведем данные в графическое окно.

function [where,varargout]=point(varargin)

if ~isnumeric(varargin{1})|~isreal(varargin{1})|...

max(size(varargin{1}))~=1

error('Аргумент №1 должен быть вещественным числом') end

if ~isnumeric(varargin{2})|~isreal(varargin{2})|...

max(size(varargin{2}))~=1

error('**Аргумент №2 должен быть вещественным числом**') end

for i=3:length(varargin)

if ~isnumeric(varargin{i})|~isreal(varargin{i})|... min(size(varargin{i}))~=1|length(varargin{i})~=3|...

 $varargin{i}(3) < 0$ str1='Аргумент №'; str2=num2str(i): str3='должен быть вещ. вектором длиной 3 с 3-м элементом >=0'; strerror=strcat(str1,str2,str3); error(strerror) end end % координаты заданной точки px=varargin{1}; $py=varargin{2};$ % нахождение числа заданных кругов Ncircle=nargin-2; % или Ncircle=length(varargin)-2; for i=1:Ncircle $Xcircle(i)=varargin{i+2}(1);$ $Ycircle(i)=varargin{i+2}(2);$ $Rcircle(i)=varargin{i+2}(3);$ end % Полагаем where=0 и Nc=0, т.е. пока нет ни одного нужного круга Nc=0:where=0: % Перебор кругов в цикле for i=1:Ncircle % dist – расстояние от точки до центра текущего круга dist=sqrt((px-Xcircle(i))^2+(py-Ycircle(i))^2); % Сравнение расстояния с радиусом круга if dist<=Rcircle(i) % Требуемый круг найден % where – логическая переменная where=1, если точка попала хоть в один % из заданных кругов % Nc - счетчик кругов, в которые попала точка % Вектор Nums содержит номера кругов, в которые попала точка Nc=Nc+1;where=1; Nums(Nc)=i; end

end

% количество выходных параметров определяется системной % константой nargout switch nargout case(2) $varargout(1) = \{Nc\};$ case(3) $varargout(1) = \{Nc\};$ varargout(2)={Nums}; end % Вывод в графическое окно figure; %Создание окна % Построение окружностей t = [0:pi/20:2*pi];for i=1:Ncircle x=Rcircle(i)*cos(t)+Xcircle(i); y=Rcircle(i)*sin(t)+Ycircle(i); plot(x,y)% графики должны выводиться в одно графическое окно hold on end % Вывод точки красным цветом plot(px,py,'*r') hold off axis equal % одинаковый масштаб по оси х и у Обратимся к файл-функции point с двумя выходными параметрами w и п: $[w,n]=point(3,4,[0\ 0\ 1],[4\ 4\ 6],[-1\ 4\ 5])$ В командном окне получим: w=1 – точка (3, 4) попала хотя бы в один круг n=2 – точка (3, 4) попала в два круга

и появится графическое окно с изображением заданных кругов и заданной точки (см. рис. 3)



Рис. 3. Графическое окно с изображением заданных кругов и точки

Таблица 1

Команда	Действие
1	2
load	Считывает данные из файла matlab.mat, если он
	создан командой save, иначе сообщение об ошиб-
	ке
load <имя ф-ла>	Загружает переменные из МАТ-файла с полным
	именем <имя ф-ла>
load <имя ф-ла> Х Ү	Загружает в рабочую область только указанные
	переменные
load <имя ф-ла> -ascii	Загрузка файла в формате ASCII
load <имя ф-ла> -mat	Загрузка файла в формате МАТ
save	Выгружает все переменные рабочей области в
	двоичном формате в специальный файл mat-
	lab.mat

Команды работы с файлами и переменными Workspace

save <имя ф-ла>	Сохраняет все переменные рабочей области в
	двоичном файле с именем <имя ф-ла>
save <имя ф-ла> <пе-	Выгружает указанные переменные рабочей об-
ременные>	ласти в двоичном файл с именем
	<имя ф-ла>
save <имя ф-ла> <оп-	Записывает данные в формате, заданный опциями
ции>	(-mat,-append,-ascii и др.)
clear	Удаляет все переменные из рабочей области
clear <имя>	Удаляет только М-, МЕХ-файл или переменную с
	данным именем
clear <имя1> <имя2>	Удаляет из рабочей области памяти переменные
	и функции с указанными именами
1	2
1 clear global <имя>	2 Удаляет глобальную переменную из рабочей об-
1 clear global <имя>	2 Удаляет глобальную переменную из рабочей об- ласти глобальных переменных
1 clear global <имя> clear <ключ>	2 Удаляет глобальную переменную из рабочей об- ласти глобальных переменных Удаляет переменные и функции из рабочей об-
1 clear global <имя> clear <ключ>	2 Удаляет глобальную переменную из рабочей об- ласти глобальных переменных Удаляет переменные и функции из рабочей об- ласти в зависимости от значения ключа (all,
1 clear global <имя> clear <ключ>	2 Удаляет глобальную переменную из рабочей об- ласти глобальных переменных Удаляет переменные и функции из рабочей об- ласти в зависимости от значения ключа (all, classes, functions, global, import, variables)
1 clear global <имя> clear <ключ> open <имя>	2 Удаляет глобальную переменную из рабочей области глобальных переменных Удаляет переменные и функции из рабочей области в зависимости от значения ключа (all, classes, functions, global, import, variables) Открывает переменную или файл с указанным
1 clear global <имя> clear <ключ> open <имя>	2 Удаляет глобальную переменную из рабочей области глобальных переменных Удаляет переменные и функции из рабочей области в зависимости от значения ключа (all, classes, functions, global, import, variables) Открывает переменную или файл с указанным именем

Другие команды для работы с файлами и переменными рабочей области см. в приложении 1.

Например, текстовый файл dat.txt имеет вид: 12 10 5 1 2 3 необходимо представленные данные загрузить в переменные x и y s=load('dat.txt') x=s(1,:) y=s(2,:) Результат работы программы в командном окне:

Часто при обращении к стандартным программам предварительно требуется создать вспомогательный файл.

Например, необходимо создать программно файл-функцию a1.m, вычисляющую sin(x) на интервале от 0 до 20 с шагом 0.1.

clear all; % очистить все переменные

% создать массив строк

fstr=['x=0:0.1:20;',...

 $y=\sin(x)$;

% в переменную strpath загрузить имя текущего каталога strpath=eval('cd');

% фунция streat объединяет строки, в результате

% переменная fname будет содержать полное имя файла fname=strcat(strpath,'\a1.m');

% переменная fid содержит указатель на файл для записи fid=fopen(fname,'w');

% выводит строку 'function a1(x,y)' в файл с указателем fid % согласно формату

fprintf(fid, '% s n', function a1(x,y)');

fprintf(fid,'%s\r\n',fstr); % выводит строки fstr в файл с указателем fid fclose(fid): % закрывает файл с указателем fid

Результат выполнения программы:

В текущем каталоге запишется файл с именем "a1.m", содержимое которого следующее: function a1(x, y)

x=0:0.1:20

y = sin(x)

ГРАФИЧЕСКИЕ СРЕДСТВА СИСТЕМЫ МАТLAB

Одно из достоинств системы Matlab – обилие средств графики, начиная с команд построения простых графиков функций в различных системах координат, кончая презентационными и комбинированными графиками с элементами анимации, а также средствами проектирования графического пользовательского интерфейса GUI.

ДВУХМЕРНЫЕ ГРАФИКИ

plot	графики в линейном масштабе
loglog	графики в логарифмическом масштабе
semilogx	графики в полулогарифмическом масштабе по оси х
semilogy	графики в полулогарифмическом масштабе по оси у
polar	графики в полярных координатах
plotyy	график с двумя вертикальными осями

График в линейном масштабе

Синтаксис: plot(y) plot(x,y)plot(x1,y1,s1,x2,y2,s2,...) plot(x,y,s)

Команда plot(y) – строит график одномерного массива у в зависимости от номера элемента

Команда plot(x,y) – строит график одномерного массива у в зависимости от одномерного массива х, если массив у двумерный, то строятся графики для столбцов массива у в зависимости от элементов массива х; если оба массива х и у двумерные, то строятся зависимости столбцов массива у от столбцов массива х.

Команда plot(x,y,s) – строковая переменная s содержит способ отображения линии графика и может включать до трех символов следующей таблицы:

Таблица 2

Способы отображения линий

Тип линии	Тип точки		Цвет
Непрерывная -	Точка	•	Желтый
			У
Штриховая	Плюс	+	Фиолетовый
			m
Двойной пунктир :	Звездочка	*	Голубой
			c
Штрих-пунктирная	Кружок	0	Красный
			r
	Крестик	Х	Зеленый
			og v
	Квадрат	S	Синии
			b
	Ромо	d	Белыи
			W
	Греугольник (вниз)	v	Черныи
			K
	Треугольник (вверх)	~	
	<u>Треугольник (вправо)</u>	>	
	Треугольник (влево)	<	
	Пятиугольник	р	
	Шестиугольник	h	

Команда plot(x1, y1, s1, x2, y2, s2, ...) – позволяет построить несколько графиков в одних координатных осях

ПРИМЕР (см. рис. 4). Функции y=sin(x), y=cos(x) и y=sin(x)cos(x), построенные в одних координатных осях.

clear all x=-5:0.1:7; y1=sin(x); y2=cos(x); y3=sin(x).*cos(x); plot(x,y1,'-vr',x,y2,'-.k',x,y3,':s');



Рис. 4. Графики, построенные с помощью команды **plot График в полярных координатах** *Синтаксис:* polar(phi,r); polar(phi,r,s)

Команды polar(phi,r) и polar(phi,r,s) – строят график функции r=r(phi) в полярных координатах, задаваемых углом phi и радиусом г. Строковая переменная s содержит способ отображения линии (см. таблицу 2).

ПРИМЕР (см. рис. 5). График функции **r=sin**(**4φ**) в полярных координатах clear all; phi=0:0.01:2*pi; polar(phi, sin(4*phi),'pk'); title('r=sin(4*phi)');



Рис. 5. График, построенный с помощью команды polar

ТРЕХМЕРНЫЕ ГРАФИКИ

plot3	Построение линий и точек в трехмерном пространстве
contour	Изображение линий уровня для трехмерной поверхности
contourc	Формирование массива описания линий уровня
contour3	Изображение трехмерных линий уровня
meshgrid	Формирование двумерных массивов Х и Ү
mesh	Трехмерная сетчатая поверхность
meshc	Трехмерная сетчатая поверхность с проекцией линий постоян-
	ного уровня
meshz	Трехмерная сетчатая поверхность с плоскостью отсчета на ну- левом уровне
surf	Цветная сетчатая поверхность
surfc	Цветная сетчатая поверхность с проекцией линий постоянного
	уровня
surfl	Цветная сетчатая поверхность с подсветкой

Построение линий и точек в трехмерном пространстве Синтаксис:

 $\begin{array}{ll} plot3~(x,\,y,\,z) & plot3(X,Y,Z) & plot3~(x,\,y,\,z,\,s) \\ plot3~(x1,\,y1,\,z1,\,s1,\,x2,\,y2,\,z2,\,s2,\ldots) \end{array}$

Команда plot3(x,y,z) – строит точки с координатами x(i), y(i), z(i) и соединяет их прямой линией.

Команда plot3(X,Y,Z) – где X,Y,Z-двумерные массивы одинакового размера, строит точки X(i,:), Y(i,:), Z(i,:) для каждого столбца и соединяет их прямыми линиями.

Команда plot3(x,y,z,s) – строит график функции z(x,y), где цвет и форма точек и линий задается с помощью строкой переменной s (см. таблицу 2). Команда plot3(x1, y1, z1, s1, x2, y2, z2, s2,...) – позволяет построить на одном графике несколько функций.

Для построения поверхностей удобно задать на плоскости XOY сетку с узлами x(i), y(i), это можно осуществить с помощью функции **meshgrid**.

Формирование сетки на плоскости в виде двумерных массивов Х и Ү

Синтаксис: [X,Y]=meshgrid(x,y) [X,Y]=meshgrid(x)

Функция [X,Y]=meshgrid(x, y) задает сетку на плоскости XOY в виде двумерных массивов X,Y, которые определяются одномерными массивами x,y

Функция [X,Y]=meshgrid(x) – упрощенная форма записи для функции [X,Y]=meshgrid(x,x)

ПРИМЕР (см. рис. 6). Построение поверхности $z=x \cdot exp(-x^2-y^2)$ с помощью команды plot3 x=-2:0.1:2;y=x;[X,Y]=meshgrid(x,y); $Z=X.*exp(-X.^2-Y.^2);$

Z=X.*exp(-X.^ plot3(X,Y,Z);



Рис. 6. Поверхность, построена с помощью команды plot3

Трехмерная сетчатая поверхностьСинтаксис:mesh(X,Y,Z,C)meshc(X,Y,Z,C)meshc(X,Y,Z,C)meshz(X,Y,Z,C)

mesh(x,y,Z,C)	meshc(x,y,Z,C)	meshz(x,y,Z,C)	
mesh(Z,C)	meshc(Z,C)	meshz(Z,C)	
mesh(x,y,Z)	meshc(x,y,Z)	meshz(x,y,Z)	
mesh(Z)	meshc(Z)	meshz(Z)	h=mesh()

Команда mesh (X, Y, Z, C) – выводит на экран сетчатую поверхность для значений массива Z, определяемых на множестве значений массивов X и Y. Цвета узлов поверхности задаются массивом C. Цвета ребер определяются свойством **EdgeColor** объекта **surface.**

Применение функции shading после обращения к функции mesh изменяет спецификации свойств EdgeColor.

Таблица З

Спецификации свойств EdgeColor при применении функций shading и mesh

Сройство	Π	Ірименяемая функц	КИЛ
Своиство	mesh	shading flat	shading interp
EdgeColor	flat	flat	interp
FaceColor	Цвет фона	Цвет фона	Цвет фона

Команда mesh (x, y, Z, C) выполняет ту же функцию, но вместо двумерных массивов X, Y использует их проекции. В этом случае узлы сетчатой поверхности определяются тройками $\{x(j), y(i), Z(i, j)\}$, где вектор x определяет столбцы массива Z, а y – строки.

Команда mesh(Z, C) использует сетку, которая определяется одномерными массивами x=1:n и y=1:m.

Команды mesh(X, Y, Z), mesh (x, y, Z), mesh (Z) используют в качестве массива цвета C=Z, то есть цвет в этом случае пропорционален высоте поверхности.

Группа команд meshc (...) в дополнение к трехмерным поверхностям строит проекции линий уровня.

Группа команд meshz (...) в дополнение к трехмерным поверхностям строит плоскость отсчета на нулевом уровне, закрывая поверхность, лежащую ниже этого уровня.

Функция h = mesh (...) возвращает дескриптор (указатель) h для графического объекта surface.

ПРИМЕР (см. рис. 7). Построение поверхности $z=x \cdot exp(-x^2-y^2)$ и линий уровня с помощью функции **meshc**

[X,Y]=meshgrid (-2:0.1:2); Z=X.*exp(-X.^2-Y.^2); Meshc (X,Y,Z);



Рис. 7. Поверхность, построенная с помощью команды meshc

Затененная сетчатая поверхность			
Синтакси	с:		
surf(X,Y,Z,C)	surfc(X,Y,Z,C)	surf(x,y,Z,C)	surfc(x,y,Z,C)
surf(Z,C)	surfc(Z,C)	surf(x,y,Z)	surfc(x,y,Z)
surf(Z)	surfc(Z)		

Команда surf (X, Y, Z, C) – выводит на экран сетчатую поверхность для значений массива Z, определяемых на множестве значений массивов X и Y. Цвет ячейки определяется массивом C. Цвет ребер – черный, определяется свойством EdgeColor, специфицирован как [0 0 0]. Можно задать одинаковый цвет для всех ребер в виде вектора [r g b]. Если применить спецификацию **none**, то ребра не будут прорисовываться.

Применение функции shading после обращения к функции surf изменяет спецификации свойств EdgeColor и FaceColor согласно следующей таблице:

Таблица 4

Спецификации свойств EdgeColor и FaceColor

Свойство	Применяемая функция		
	surf	shading flat	shading interp
EdgeColor	[0 0 0]	None	None
FaceColor	flat	flat	interp

Остальные команды определяются аналогично команде mesh (...).

ПРИМЕР (см. рис. 8). Построение поверхности $z=x \cdot exp(-x^2-y^2)$ с помощью функции **surf**

[X,Y]=meshgrid(-2:0.1:2); Z=X.*exp(-X.^2-Y.^2); surf(X,Y,Z); colormap(bone) shading interp colorbar



Рис. 8. Поверхность, построенная с помощью команды surf			
Затенен	ная поверхность с подсветкой		
Синтаксис:			
surfl(X,Y,Z,s)	surfl(Z,s)		
surfl(x,y,Z,s,k)	surfl(Z,s,k)		
surfl(X,Y,Z)	surfl(Z)		

Команда surfl(X,Y,Z,s) выводит на экран затененную поверхность с подсветкой для значений массива Z, определенных на множестве массивов X и Y. Направление на источник света может быть задано с помощью вектора s=[Sx,Sy,Sz] в декартовых координатах или вектора s=[az,elev] в сферических координатах. По умолчанию азимут az= -37.5° , возвышение elev= 30° . Подсветка учитывает модели рассеяния, отражения и зеркального эффекта освещения поверхности.

Команда surfl (x, y, Z, s, k) позволяет управлять параметрами рассеяния, отражения и зеркального эффекта, используя вектор k=[ka,kd,ks,spread], который учитывает эффекты отраженного света ka, диффузного отражения kd, зеркального отражения ks и зеркального распространения spread. По умолчанию k=[0.55 0.6 0.4 10].

Команда surfl (X, Y, Z) использует значения параметров s и k по умолчанию.

Команды surfl (Z,...) строят графики, не учитывая истинных значений массивов X и Y.

Алгоритм **surfl** вычисляет нормали к поверхности, поэтому необходимо, чтобы входные матрицы имели размер, по крайней мере, 3х3.

ПРИМЕР (см. рис. 9). Изображение функции $z=ln(x^2+y^2)$ с подсветкой: [X,Y]=meshgrid(-:1/8:3); Z=log(X.^2+Y.^2); surfl(X,Y,Z); title('ln(x^2+y^2)'); shading interp colormap (gray)



Рис. 9. Поверхность, построенная с помощью команды surfl

Масштабирование осей

Синтаксис axis([xmin xmax ymin ymax])

axis([xmin xmax ymin ymax zmin zmax])				
axis('auto')	axis(axis)	v=axis	axis('ij')	
axis('xy')	axis('square')	axis('egual')	axis('image')	
axis('normal')	axis('off')	axis('on')		
[s1,s2,s3] = axis('sta	te') axis(s1,s2,s3)			

Команда axis([xmin xmax ymin ymax]) устанавливает масштаб по осям х,у для активного графического окна.

Команда axis([xmin xmax ymin ymax]) устанавливает масштаб по осям x,y,z для активного графического окна.

Команда axis('auto') возвращает масштаб к значениям принятым по умолчанию.

Команда axis(axis) фиксирует текущие значения масштабов для последующих графиков, как если бы был включен режим hold. Функция v=axis возвращает вектор-строку масштабов по осям для активного графика.

Команда axis('ij') перемещает начало отсчета в левый верхний угол (матричная система координат).

Команда axis('xy') возвращает декартову систему координат, начало отсчета находится в левом нижнем углу.

Команда axis('square') устанавливает одинаковый диапазон изменения переменных по осям.

Команда axis('equal') устанавливает масштаб, задающий одинаковые расстояния между метками по оси х и оси у.

Команда axis('image') устанавливает масштаб, задающий квадратные размеры пикселей.

Команда axis('normal') восстанавливает полноразмерный масштаб, отменяя масштабы, установленные командами axis('square') и axis('equal').

Команда axis('off') убирает с осей их обозначения и маркеры.

Команда axis('on') восстанавливает на осях их обозначения и маркеры. Функция [s1, s2, s3]= axis('state') возвращает строку, определяющую вектор состояния объекта axes: s1='auto'|'off'; s2='on'|'off'; s3='xy'|'ij' Команда axis (s1, s2, s3) устанавливает параметры объекта axes в соответствии с вектором [s1,s2,s3], по умолчанию ['auto','on','xy'].

Нанесение сетки

Синтаксис: grid on grid off grid

Команда grid on наносит координатную сетку на текущие оси. Команда grid off удаляет координатную сетку. Команда grid выполняет роль переключателя с одной функции на другую.

Управление режимом сохранения текущего графического окна *Синтаксис:* hold on hold off hold

Команда hold on включает режим сохранения текущего графика и свойств объекта **axes**, т.е. последующие команды приведут к добавлению графиков в графическом окне.

Команда hold off выключает режим сохранения графика. **Команда hold** выполняет роль переключателя с одной функции на другую.

Разбиение	графического окн	а на подокна
Синтаксис:		
subplot (m,n,p)	subplot(h)	subplot(mnp)

Команды subplot(m,n,p), subplot(mnp) производят разбивку графического окна на несколько подокон, создавая при этом новые объекты axes; значение m – указывает на сколько частей разбивается окно по горизонтали, n – по вертикали, а p – номер подокна, куда будет выводится очередной график.

Команда subplot(h) выбирает подокно с дескриптором h.

ПРИМЕР (см. рис. 10). Построение нескольких графиков в отдельных подокнах графического окна x=-10:0.1:10; y1=sin(x).*x; $y2=log(x.^2);$ y3=cos(x).*tan(x/10); subplot(2,2,1), plot(x,y1), set(gca,'fontsize',12), title('y1=sin(x)*x'), grid on; subplot(2,2,3), plot(x,y2), set(gca,'fontsize',12), title('y=lg(x^2)'); subplot(2,2,2), plot(x,y3), set(gca,'fontsize',12), title('y=cos(x)*tg(x/10)');

subplot(2,2,4), plot(y1,x);



Рис. 10. Разбиение графического окна на подокна

Управление масштабом графика

Синтаксис: Zoom on zoom Zoom off zoom out

Команда zoom on включает режим масштабирования активного графика. При нажатии левой мыши вблизи интересующей Вас точки, масштаб графика увеличивается в 2 раза; при нажатии правой клавиши масштаб в 2 раза уменьшается. Удерживая левую клавишу, можно выделить интересующую область окна.

Команда zoom off выключает режим масштабирования.

Команда zoom реализует переключение от одного режима к другому. Команда zoom out возвращает график в исходное состояние.

Палитра цветов

Синтаксис:	
colormap(C)	C= colormap
colormap('default')	

Палитра цветов С – это матрица размера $m \times 3$ действительных чисел в диапазоне [0,1]. Строка к палитре сформирована из трех чисел, которые указывают интенсивность красного, зеленого и синего цветов, т.е. $C(\kappa, :) = [\Psi(\kappa) \quad g(\kappa) \quad B(\kappa)].$

Команда colormap (C) устанавливает палитру соответственно матрице C, где C – матрица размера mx3 действительных чисел в диапазоне от 0 до 1 (RGB).

Команда colormap('default') устанавливает штатную палитру, соответствующую модели HSV (Hue-Saturation-Value, оттенок-насыщенностьзначение). Последовательность цветов этой палитры соответствует цветам радуги. Более того в Matlab реализованы и другие палитры: bone (серая с оттенком синего), соррег (линейная в оттенках меди) и другие (см. приложение).

Установление соответствия между палитрой цветов и масштабированием осей

Синтаксис:	
caxis([cmin cmax])	caxis('auto')
v=caxis	caxis(caxis)

Команда caxis позволяет еще одним способом установить свойства Clim и ClimMode объекту axes.

Команда caxis([cmin cmax]) устанавливает диапазон [cmin cmax] из текущей палитры цветов для масштабирования данных, которые используются объектами **surface** и **patch**.

Команда caxis('auto') устанавливает штатное масштабирование данных палитрой цветов.

Команда v=caxis возвращает вектор-строку [cmin cmax].

Команда caxis(caxis) фиксирует текущую палитру для последующих графиков.

Затенение поверхностей

Синтаксис: shading faceted shading flat shading interp

Команды группы shading устанавливают способ затенения графических объектов surface и patch.

Команда shading faceted устанавливает равномерную раскраску ячеек с нанесением черных граней.

Команда shading flat устанавливает раскраску каждой ячейки или грани определенным цветом, который зависит от цвета узлов сетки.

Команда shading interp устанавливает раскраску каждой ячейки или грани цветом, который определяется билинейной интерполяцией цветов в узлах сетки.

ПРИМЕР (см. рис. 11). Поверхность $z=y^2 \exp(-x^2-y^2)$; (см. так же рис. 8 и рис. 9).

[X,Y]=meshgrid(-3:0.1:3); Z=Y.^2*exp(-X.^2-Y.^2); surfl(X,Y,Z); shading faceted colormap(bone)



Рис. 11. Поверхность, затененная с помощью команды shading faceted

Линии уровня для трехмерной поверхности

Синтаксис:

contour(Z)	contour(x,y,Z)	contour(Z,n)	contour(x,y,Z,n)
contour(Z,v)	contour(x,y,Z,v)	contour(,'тип	линии')
C = contour()	[C,h]=contour()		

Команда contour(Z) рисует двумерные линии уровня для массива данных Z, определяющего поверхность в 3-х мерном пространстве без учета диапазона изменения координат x и y.

Команда contour(x,y,Z) рисует двумерные линии уровня для массива данных Z, определяющего поверхность в 3-х мерном пространстве с учетом диапазона изменения координат x и y.

Команды contour(Z,n) и contour(x,y,Z,n) рисуют п линий уровня для массива данных Z; по умолчанию n=10.

Команды contour(Z,v) и contour(x,y,Z, v) рисуют линии уровня для заданных значений, которые указаны в векторе v.

Команда contour(...,'тип линии') рисует линии уровня, тип и цвет которых определяется параметром 'тип линии' команды plot.

Функция C=contour(...) возвращает массив С описания линий уровня для последующего использования функцией clabel.

Функция [C,h]=contour(...) возвращает массив С и вектор-столбец дескрипторов h графических объектов line каждой линии уровня.

Формирование массива описания линий уровня осуществляют функции **C=contourc(...)**.

Для изображения трехмерных линий уровня используются команды contour3(...).

ПРИМЕР (см. рис. 12). Построение линий уровня для поверхности $z=x \exp(-x^2-y^2)$ с помощью функции **C=contour(Z)** и команды **clabel** x=-2:0.1:2;

y=x; [X,Y]=meshgrid(x,y); Z=X.*exp(-X.^2-Y.^2); surfl(X,Y,Z); C=contour(Z); clabel(C);



Рис. 12. Линии уровня поверхности $z=x \exp(-x^2-y^2)$

Заголовки для двух- и трехмерных графиков *Синтаксис:* title('текст')

Команда title('текст') размещает текст над графиком (см. рис. 13)

Обозначения осей

Синтаксис: xlabel('текст') ylabel('текст') zlabel('текст')

Команды xlabel('текст'), ylabel('текст'), zlabel('текст') размещают текст вдоль осей х,у,z соответственно. Повторное использование приводит к замене старого текста новым.

Добавление текста к текущему графику Синтаксис:

text(x,y,'TekcT') gtext('TekcT')

Команда text(x,y, 'текст') размещает начало текста в заданной позиции (x,y).

Команда gtext('текст') размещает текст в позиции, указанной мышкой.

Пояснение к графику

Синтаксис: legend('текст1', 'текст2,...) legend('тип линии 1', 'текст1', 'тип линии 1', 'текст2,...) legend(h,...)

Команда legend('текст1', 'текст2,...) добавляет к текущему графику пояснения в виде указанных текстовых строк.

Команда legend('тип линии 1', 'текст1', 'тип линии 1', 'текст2,...) позволяет специфицировать тип линии, которая выносится в пояснение, так, как это делается в команде plot.

Команда legend(h,...) добавляет пояснение к графику с дескриптором h.

ПРИМЕР (см. рис. 13). Построение функций Бесселя 1,3 и 5-го порядка на одном графике. x=0:0.2:12; hp=plot(x,bessel(1,x),'-',x,bessel(3,x),'--',x,bessel(5,x),'-.'); title('ФУНКЦИИ БЕССЕЛЯ'); gtext('функции Бесселя'); set(gca,'fontsize',12) set(gca,'linewidth',[3]); legend('порядка 1','порядка 3','порядка 5') команд **plot,title, gtext, legend**



Рис. 13. Построение графиков с использованием

Шкала палитры

Синтаксис: colorbar('vert') colorba colorbar(h) colorba

colorbar('horiz') colorbar

Команды colorbar('vert'), colorbar('horiz') добавляют к текущему графику вертикальную и горизонтальную палитры соответственно.

Команды colorbar(h) добавляет к графику с дескриптором h шкалу палитры.

Команды colorbar без аргументов размещает на текущем графике новую вертикальную шкалу палитры или обновляет прежнюю (см. рис. 8).

Столбцовые диаграммы

 Синтаксис:

 bar(y)
 bar (x, y)
 [xb,yx]=bar(...)

 bar(y, 'тип линии')
 bar(x,y, 'тип линии')

Команда bar(у) выводит график элементов одномерного массива в виде столбцовой диаграммы.

Команда bar(x,y) выводит график элементов массива у в виде столбцов в позициях, определяемых массивом х. Если массивы х,у двумерные, одинаковых размеров, то каждая диаграмма определяется соответствующей парой столбцов и они надстраиваются одна над другой.

Команды bar(y,'тип линии'), **bar(x,y,'тип линии')** позволяют задать тип линии, используемых для построения диаграмм, по аналогии с командой plot.

Функция [xb,yb]=bar(...), не выводит графика, а формирует такие массивы xb и yb, которые позволяют построить столбцовую диаграмму с помощью команды plot (xb,yb).

ПРИМЕР (см. рис. 14). График функции y=sin(x)*exp(-x²) в виде столбцовой диаграммы. x=-2.9:0.2:2.9; bar(x,sin(x).*exp(-x.*x))



Рис. 14. График, построенный с помощью команды bar

График с указанием интервала погрешности Синтаксис:

errorbar(x,y,u,w,'тип линии') errorbar(y,u,w'тип линии') errorbar(x,y,q) errorbar(y,q)

Команда errorbar(x,y,u,w,'тип линии') строит график функции у в зависимости от x с указанием интервала погрешности, который определяется массивами u,w.

Массивы **x**, **y**, **u**, **w** должны быть одного размера. Погрешности в каждой точке (x(i),y(i)) определяются отклонениями вниз u(i) и вверх w(i) относительно точки графика, так что суммарная погрешность равна u(i)+w(i). Строковая переменная **'тип линии'** аналогична используемой в команде plot.

Если **х**, **у**, **u**, **w** – двумерные массивы, то в этом случае каждому столбцу соответствует свой график.

Команда errorbar (**x**, **y**, **q**) строит график функции с погрешностью ±**q** относительно точки графика.

Команды errorbar (y, u, w, 'тип линии') и errorbar (y, q) строят график в зависимости от номера элемента. **ПРИМЕР** (см. рис. 15). График функции $y=sin(x) \cdot tg(x)$ с интервалом погрешности $\pm q$, которое определяется как стандартное отклонение. x=1:10;y=sin(x).*tan(x)q=std(y)*ones(size(x))/5;

errorbar(x,y,q)



Рис. 15. Использование команды errorbar

Построение гистограммы

Синтаксис:

hist(y)	hist(y,x)
hist(y,n)	[y,x]=hist(y,)

Команды hist (...) подсчитывают и отображают на графике количество элементов массива у, значения которых попадают в заданный интервал; для этого весь диапазон значений у делится на n интервалов (по умолчанию 10) и подсчитывается количество элементов в каждом интервале

Команды hist(у) выводит гистограмму для 10 интервалов

Команды hist(y,n) выводит гистограмму для n интервалов Команды hist(y,x) выводит гистограмму с учетом диапазона изменения

переменной х.

Функция [y,x] = hist(y,...) формирует такие массивы у и x, что bar(x,y) является гистограммой.

ПРИМЕР (см. рис. 16). Гистограмма для 10000 случайных чисел, распределенных по нормальному закону.



Рис. 16. График, построенный с помощью команды hist

Дискретный график	
Синтаксис:	
stem(y)	stem(x, y)
stem(y,'тип линии')	stem(x, y,'тип линии')

Команда stem(y) выводит график элементов одномерного массива у в виде вертикальных линий, которые заканчиваются в точках графика. Команда stem(x,y) выводит график элементов массива у в виде вертикальных линий в позициях, определяемых массивом x, элементы которого должны быть упорядочены в порядке возрастания.

Команды stem(y,'тип линии'), stem(x,y, 'тип линии') позволяют задать тип линий, используемых для построения дискретного графика, по аналогии с командой plot. **ПРИМЕР** (см. рис. 17). Значения амплитуд для функции x=exp(-at)sin(bt).

t=0:3:200; a=0.02; b=0.5; x=exp(-a*t).*sin(b*t); stem(t,x,'color',[0 0 0]), grid set(gca,'FontName','TimeET'); xlabel('BPEMЯ'), ylabel('АМПЛИТУДА')



Рис. 17. График, построенный с помощью команды stem

Ступенчатый график

Синтаксис: stairs(y) stairs(x,y) [xb,yb]=stairs(...)

Команда stairs(у) выводит график элементов одномерного массива у в виде ступенчатой функции, аналогично столбцовой диаграмме, но без вертикальных линий. Такие графики используются при выводе процессов в дискретно-непрерывных системах.

Команда stairs(x,y) выводит график элементов одномерного массива у в виде ступенчатой функции в позициях, определяемых массивом х, элементы которого должны быть упорядочены в порядке возрастания. **Функция [xb,yb]=stairs(...)** не строит график, а формирует такие массивы xb,yb, которые позволяют построить ступенчатую функцию с помощью команды plot(xb,yb).

ПРИМЕР (см. рис. 18). Ступенчатый график x=-2*pi:0.5:2*pi; stairs(x, sin(x));



Рис. 18. График, построенный с помощью команды stairs

Гистограм	има в полярных координатах
Синтаксис:	
rose(phi)	rose(phi,n)
rose(phi,x)	[phi,r]=rose(phi,)

Команды гоse(...) подсчитывают и отображают на графике количество угловых элементов в массиве phi, значения которых попадают в заданный интервал, для этого весь диапазон значений phi делится на п интервалов (по умолчанию 20) и подсчитывается количество угловых элементов в каждом интервале. Такая гистограмма носит название розы ветров.

Команда rose(phi) строит розу ветров для 20 интервалов.

Команда rose(phi, n) строит розу ветров для n интервалов.

Команда rose(phi, x) строит розу ветров с учетом диапазона изменения переменной х.

Функция [phi,r]=rose(phi) формирует массивы phi и r, такие что polar(phi,r) является гистограммой в полярных координатах.

ПРИМЕР (см. рис. 19). Измерение направления ветра через каждый час в течение 12 ч задано массивом :

w=[45 90 90 45 360 335 360 270 335 270 335 335];

Необходимо построить розу ветров для данного периода наблюдений.

На графике видно, что преимущественное направление розы ветров в течение заданного периода составляет 330⁰.

w=[45 90 90 45 360 335 360 270 335 270 335 335]; w=w*pi/180;

set(gca,'fontsize',[14]);

rose(w);



Рис. 19. График, построенный с помощью команды rose

Графики векторов

Синтаксис:compass(z)feathercompass(x,y)feathercompass(y,'тип линии')feathercompass(x,y, 'тип линии')feather[hc,hb]=compass(...)feather

feather(z) feather (x,y) feather (y,'тип линии') feather(x,y, 'тип линии') **Команда compass**(z) – выводит график комплексных элементов одномерного массива z в виде векторов-стрелок, исходящих из начала координат. **Команда compass**(x,y) равносильна команде compass(x+i*y)

Команды compass(y, 'тип линии'), compass(x,y, 'тип линии') позволяют задать тип линий, используемых для построения векторов-стрелок, по аналогии с командой plot.

Функция [hc,hb] = compass(...) не выводит графика, а формирует массивы hc, hb, которые позволяют построить столбцовую диаграмму с помощью команды plot(hc,hb).

Команды feather(...) аналогичны командам compass(...), но выводят вектора-стрелки не из одной точки, а из равноотстоящих точек горизонтальной оси.

ПРИМЕР (см. рис. 20). Измерения направления и скорости ветра через каждый час в течение 12 ч заданы массивами:

w=[45 90 90 45 360 335 360 270 335 270 335 335] и

knots=[6 6 8 6 3 9 6 8 9 10 14 12].

Построить соответствующую диаграмму ветров для данного периода наблюдений.

w=[45 90 90 45 360 335 360 270 335 270 335 335]; knots=[6 6 8 6 3 9 6 8 9 10 14 12];

rw=w*pi/180;

% pol2cart-преобразует полярные координаты в декартовы [x, y]=pol2cart(rw, knots); h=polar(rw, knots, ':w'); hold set(gca.'fontsize',[14]); compass(x, y)



Рис. 20. Диаграмма ветров, построенная с помощью команды compass

ПРИМЕР (см. рис. 21). Векторы заданны в полярной системе координат через угол theta и радиус-вектор г. Необходимо построить их в декартовой системе координат.

clear all

theta = (-90:10:90)*pi/180; r = 2*ones(size(theta)); % функция pol2cart преобразует полярные координаты в декартовы. [p,v] = pol2cart(theta,r); feather(p,v);



Рис. 21. Использование команды feather

Поле градиентов функции

Синтаксис:quiver(X,Y,DX,DY)quiver(1)quiver(DX,DY)quiver(1)quiver(x,y,dx,dy,s)quiver(1)quiver(..., тип линии')quiver(1)

quiver(x,y,DX,DY) quiver(dx,dy,s) **Команда quiver (X, Y, DX, DY)** формирует и выводит на экран поле градиентов функции в виде стрелок для каждой пары элементов массивов X и Y, а пары элементов DXи DY используются для указания направления и размера стрелки.

Команда quiver (x,y,DX,DY), где x и y – одномерные массивы размеров length(x)=n и length(y)=m, где [m,n]=size(DX)=size(DY), формирует и выводит на экран поле градиентов для каждой точки; стрелки задаются четверками $\{x(i),y(i),DX(i,j),DY(i,j)\}$. При этом x соответствует столбцам DX и DY, а y – строкам.

Команда quiver(DX,DY) использует массивы x=1:n и y=1:m.

Команды quiver (x,y,dx,dy,s) и quiver (dx,dy,s) использует скаляр s как коэффициент масштаба стрелки, например s=2 – размер стрелки увеличивается вдвое.

Команда quiver(...,'тип линии') позволяет задать тип и цвет линии по аналогии с командой plot.

ПРИМЕР (см. рис. 22). Поле направлений функции $z=\exp(-x^2-y^2)$; [x, y]=meshgrid(-2:0.2:2); $z=\exp(-x.^2-y.^2)$; [dx, dy]=gradient(z, 0.5, 0.5); colormap(winter); set (gca,'fontsize',[14]); contour (x, y, z),hold on quiver (x, y, dx, dy)



Рис. 22. Поле направлений, построенное с помощью команды quiver

Движение точки по траектории

Синтаксис:

comet(y) comet(x,y) comet(x,y,p) comet(axes_handle,...)

Команда comet (у) рисует движение точки по траектории, заданной одномерным массивом у в виде головы и хвоста кометы.

Команда comet (x, y) рисует движение точки по траектории, заданной массивами x и y.

Команда comet (**x**, **y**, **p**) управляет длиной хвоста кометы p*length(y) с помощью параметра р (по умолчанию p=0.1).

Команда comet (axes_handle,...) строит траекторию движения в осях, указанных указателем axes_handle вместо текущих осей (gca).

ПРИМЕР (см. рис. 23). Разность двух функций в виде траектории движения точки

t=-pi:pi/200:pi; comet(t,tan(sin(t))-sin(tan(t)))



Рис. 23. Траектория движения точки, построенная командой comet

Закраска многоугольников

Синтаксис: fill (x, y, 'цвет') fill (x, y, c) fill (X, Y, C) fill (X1, Y1, C1, X2, Y2, C2,...) h = fill(...) Команда fill(x,y,'цвет') закрашивает многоугольник, заданный одномерными массивами x, y, цветом который может быть задан либо одним из символов 'r','g','b','c','m','y','w','k', либо вектором [r g b]. Вершины многоугольника имеют координаты (x_i, y_i).

Команда fill(x,y, c), (где с – вектор той же длины, что и x, y) закрашивает многоугольник цветом, задаваемым вектором с. Элементы вектора с масштабируются функцией caxis и используются как индексы текущей палитры для задания цветов в вершинах многоугольника. Цвет внутри многоугольника определяется билинейной интерполяцией цветов в узлах.

Команда fill(X,Y,C), где X и Y – массивы одинаковых размеров, строит для каждого столбца свой многоугольник. Если C – массив-строка, количество элементов которого равно числу столбцов массивов X и Y, то каждый многоугольник будет закрашен собственным цветом, что соответствует применению команды shading flat. Если же C имеет те же размеры, что и массивы X и Y, то закраска реализуется методом интерполяции; это соответствует применению команды shading interpolated. Команда fill (X1, Y1, C1, X2, Y2, C2,...) позволяет выполнить закраску конечного количества многоугольников.

Функция h = fill(...) возвращает вектор-столбец дескрипторов для графических объектов patch, которыми и являются закрашенные многоугольники. Команда fill(...) задает свойству Facecolor объекта **patch** одно из значений **'flat', 'interp'** или [**r g b**].

ПРИМЕР (см. рис. 24). Закраска многоугольников

t=0:pi/3:2*pi; x=2*cos(t); y=2*sin(t); c=[0.9 0.9 0.4,... 0.9 0.3 0.1 0.0]; colormap(bone); fill(x,y,c); axis('square')



Рис. 24. Многоугольник, закрашенный с помощью команды fill

Сечения функции трех переменных

Синтаксис:

slice (x, y, z, V, xi, yi, zi, n)	slice (X, Y, Z, V, xi, yi, zi, n)
slice (V, xi, yi, zi, n)	h=slice ()

Команда slice (x, y, z, V, xi, yi, zi, n) строит плоские сечения функции от трех переменных V (x, y, z) вдоль осей x, y, z; позиции сечений определяются векторами xi, yi, zi. Размер массива V равен mxnxp, где m=length(y), n=length(x), p=length(z).

Команда slice (**X**, **Y**, **Z**, **V**, **xi**, **yi**, **zi**, **n**) вместо одномерных массивов использует двумерные массивы X,Y,Z, которые вычисляются с помощью функции meshgrid.

Команда slice (V, xi, yi, zi, n) использует для задания области построения массивы x=1:n, y=1:m, z=1:p.

Функция h=slice(...) возвращает вектор-столбец дескрипторов для графических объектов surface, которыми являются сечения трехмерной функции.

```
ПРИМЕР (см. рис. 25). Сечения функции V = \exp(-x^2 - y^2 - z^2);
```

x=-2:0.2:2; y=-2:0.25:2; z=-2:0.16:2; colormap(colorcube); [X, Y, Z]=meshgrid (x, y, z) V=exp(-X.^2-Y.^2-Z.^2); slice(x,y,z,V,[2],[2],... [-.75 0.5],length(x)),grid colorbar



Рис. 25. Сечения функции V=exp(-x²-y²-z²), построенные с помощью команды **slice**

Трехмерная поверхность

Команда waterfall (X, Y, Z, C) строит поверхность для значений массива Z, определенных на множестве значений массивов X и Y. Она аналогична команде mesh, но не прорисовывает ребер сетки.

Команда waterfall (x, y, Z, C) выполняет ту же функцию, но вместо двумерных массивов X,Y использует их одномерные проекции, так что если lengt(x)=n, a length(y)=m, то [m,n]=size(Z).

Команда waterfall (\mathbf{Z} , \mathbf{C}) использует сетку, которая определяется одномерными массивами x=1:n и y=1:m.

Команды waterfall (X, Y, Z), waterfall (x, y, Z), waterfall (Z) используют в качестве массива цвета C=Z, то есть цвет в этом случае пропорционален высоте поверхности.

Функция h= waterfall(...) возвращает вектор дескрипторов h для графических объектов patch. **ПРИМЕР** (см. рис. 26). Поверхность z=x*exp(-x²-y²). [X,Y]=meshgrid([-2:0.1:2]); Z=X.*exp(-X.^2-Y.^2); waterfall(X, Y, Z)



Рис. 26. Трехмерная поверхность, построенная с помощью команды waterfall

Вычисление матрицы управления углом просмотра *Синтаксис:* T=viewmtx(az,el) T=viewmtx(az,el,phi) T=viewmtx(az,el,phi,xc)

Функция T=viewmtx (az, el) вычисляет матрицу управления углом просмотра, или обобщенную матрицу преобразований для аффинного изображения. Переменные аz и el определяют соответственно углы азимута и возвышения точки просмотра. Положительные значения угла азимута соответствуют вращению вокруг оси z против часовой стрелки. Положительные значения угла возвышения соответствуют точке просмотра, расположенной сверху над объектом, а отрицательные – снизу под объектом. Функция T=viewmtx (az, el, phi) вычисляет матрицу управления углом просмотра для перспективного изображения. Угол phi задает поворот системы координат относительно оси х и тем самым позволяет управлять степенью искажения перспективы в соответствии со следующей таблицей:

Таблица 5

Степень искажения перспективы в зависимости от угла поворота системы координат относительно оси ОХ

phi	Описание
0°	Аффинное изображение
10 ^o	Телескопическое изображение
25°	Нормальное фотоизображение
60°	Широкоугольное изображение

Функция T=viewmtx (az, el, phi, vt) вычисляет матрицу управления углом просмотра для перспективного изображения, используя в качестве дополнительного параметра вектор координат vt наблюдаемой точки (точки наведения). Координаты рассматриваются как нормализованные в диапазоне [0,1]; по умолчанию вектор vt=[0 0 0].

Управление положением точки просмотра

Синтаксис:		
T=view(az,el)	view(2)	
T=view([az,el])	view(3)	
T=view([x v z])	view(T)	[az.el]= view



Рис. 27. Положение точки просмотра через углы азимута и возвышения

Команды view(az,el), T=view([az,el]) задают положение точки просмотра, из которой наблюдается объект, используя углы азимута и возвышения (см. рис. 27).

Команда view([x y z]) задают положение точки просмотра в декартовой системе координат.

Команда view(2) устанавливает штатное положение точки просмотра для двумерной графики: $az=0^{\circ}$, $el=90^{\circ}$.

Команда view(3) устанавливает штатное положение точки просмотра для трехмерной графики: $az=-37.5^{\circ}$, $el=30^{\circ}$.

Команда view(T) устанавливает положение точки просмотра в соответствии с обобщенной матрицей преобразований, вычисленной с помощью функции viewmtx.

Функция [az el]=view присваивает текущие значения углов азимута и возвышения соответственно переменным az и el.

ПРИМЕР. Тетраэдр, прорисованный с различных точек просмотра.

```
 \begin{array}{l} x = [0 \ 1 \ 1 \ 0 \ 0 \ 0.5 \ 1 \ 0 \ 0.5 \ 1]; \\ y = [0 \ 0 \ 1 \ 1 \ 0 \ 0.5 \ 1 \ 1 \ 0.5 \ 0]; \\ z = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]; \\ for j=1:3 \\ A = viewmtx(30*j,30); \\ [m,n] = size(x); \\ x4d = [x(:),y(:),z(:),ones(m*n,1)]'; \\ x2d = A*x4d; \\ x2 = zeros(m,n); y2 = zeros(m,n); \\ x2(:) = x2d(1,:); \\ y2(:) = x2d(2,:); \\ plot(x2,y2); \\ pause(3); \\ end \end{array}
```

ИНТЕРФЕЙС И ОБЪЕКТЫ МАТLAВ

МАТLAВ является объектно-ориентированной системой, что, в частности, позволяет просто и эффективно реализовать графический интерфейс приложений. МАТLAВ имеет свои готовые объекты, иерархия которых приведена на рисунке 28:



Рис. 28. Иерархия объектов Matlab

 Root –экран

 Figure – графическое окно

 Axes – оси

 Uicontrol – элементы интерфейса (кнопки, текст, редакторские окна, скролинг и др.)

 Uimenu – меню

 Uicontextmenu – контекстное меню

 Image – рисунок

 Ligth – источник света

 Line – линия графика

 Patch – полигональный объект

 Rectangle – прямоугольник

 Surface – поверхность

 Text – текст

Каждый объект имеет свой набор свойств, которые можно просмотреть с помощью команды: get(имя объекта).

Получить значение какого-либо свойства можно, набрав следующую строку:

переменная=get(указатель объекта, 'название свойства');

Например, get(uicontrol); – получить все свойства объекта uicontrol win=figure; – открыть графическое окно

s=get(win,'color') – получить в переменную s цвет графического окна win

p=get(0,'screensize') – записать в переменную **p** размер экрана

Чтобы установить какому-либо свойству объекта определенное значение, необходимо ввести следующую команду:

set (указатель объекта, ' название свойства', значение свойства);

Создание любого графического объекта в MATLAB сопровождается появлением числового указателя на него. Функции **gcf, gca** и **gco** возвращают указатели на текущие ОКНО, ОСИ, ОБЪЕКТ.

ПРИМЕР: Построение графиков функций.

win1=figure('color',[0.8 0.8 0.8], ...% открыть графическое окно серого цвета,

'units','normal',...% в нормализованных координатах (от 0 до 1 по оси X и Y)

'menu', 'none',... % окно не содержит меню

'numbertitle', 'off',... % заголовок окна не нумеруется 'name', 'GRAPH of FUNCTION'); % имя окна

as=axes('units','normal'.... % создать оси в нормализованных координатах

'pos',[0.1 0.2 0.6 0.7]); % позиция осей: левый нижний угол, ширина и высота

set(as,'xgrid','on'); % установить свойству 'xrid' осей 'as' значение 'on' set(as,'ygrid','on'); % установить свойству 'yrid' осей 'as' значение 'on'

% program – массив строк, содержащий команды, которые % будут выполняться при нажатии кнопки 'GRAPHIC'

program=['xstr=get(edit_rang,"string");',...% в переменную xstr занести строку из поля edit rang

'ystr=get(edit_func,"string");',... % в переменную ystr занести строку из поля edit_func

'x=eval(xstr);',...% выполнить, что содержится в строке xstr 'y=eval(ystr);',...% выполнить, что содержится в строке ystr 'line(x,y)']: % вывести график в заданные оси координат

% текстовое поле, содержит надпись "rang" text func=uicontrol('units','normal'.... 'style', 'text',... 'pos',[0.72 0.75 0.25 0.03],... 'String', 'rang'); edit rang=uicontrol('units','normal',... 'style', 'edit',... % поле для ввода изменения независимой переменной, 'pos',[0.72 0.7 0.25 0.05],...% позиция поля в окне 'String','-3:0.1:3'); % в начале в поле выводится (-3,3)пустая строка % text func – текстовое поле, содержит надпись "function" text func=uicontrol('units','normal',... 'style', 'text',... 'pos',[0.72 0.65 0.25 0.03],... 'String', 'function'); % edit func – поле для ввода функции, например, х.^2 edit func=uicontrol('units','normal',... 'style', 'edit',... 'pos',[0.72 0.6 0.25 0.05],... 'String', 'x.^2'); % btn1 - кнопка 'GRAPHIC', при нажатии на которую строится график btn1=uicontrol('units','normal',... 'style', 'push',... % создание кнопки 'pos',[0.1 0.05 0.15 0.07],... % позиция кнопки 'String', 'GRAPHIC',.... % название кнопки 'Call', 'eval(program)'); %вызов команд, выполняемых при нажатии кнопки. % btn2 – кнопка 'CLEAR' обновляет оси координат btn2=uicontrol('units','normal',... 'style', 'push',... 'pos',[0.4 0.05 0.15 0.07],... 'String', 'CLEAR',... 'Call'.'cla'): btn3=uicontrol('units','normal'.... % btn3 - кнопка 'CLOSE' закрывает текущее графическое окно 'pos'.[0.7 0.05 0.15 0.07].... 'String', 'CLOSE',.... 'Call', 'close(gcf)'); В результате выполнения программы откроется следующее гра-

фическое окно, в котором можно строить графики различных функций в

заданном диапазоне (см. рис. 29).



Рис. 29. Графическое окно программы

СОЗДАНИЕ ПРИЛОЖЕНИЙ В СРЕДЕ GUIDE

Для размещения элементов интерфейса в пределах графического окна и задания, связанных с ними команд, в Matlab существует инструментальное средство GUIDE.

Вызов GUIDE осуществляется в командной строке:

>>guide

Появляется редактор окна приложения, заголовок которого **untitled.fig.** Это означает, что открывается новый файл, отвечающий за графический интерфейс вашего приложения (см. рис. 30).



Рис. 30. Редактор приложения и панель инструментов элементов интерфейса

Графический объект, выбранный в панели элементов интерфейса, обладает рядом свойств, которые можно задавать с помощью редактора свойств **Property Inspector** (см. панель инструментов для управления приложением). Когда нужный интерфейс создан, его записывают в файл <имя.fig>. Автоматически создается М-файл с тем же именем, в котором находится «заготовка» будущей программы. Каждая такая «заготовка» представляет собой файл-функцию, содержащую подфункции, обрабатывающие события (Callback,ButtonDownFcn и др.), соот-ветствующие нажатию кнопок, установке флажков и др. Подфункции необходимо «наполнить» нужным содержимым. Имена элементов управления (см. пример ниже radiobut_xgrid, radiobut_ygrid, graph, clearbut и др.) задаются с помощью свойства Tag в Property Inspector.

Например, для интерфейса, изображенного на рис. 31, файл-функция, строящая график y=exp(x²), может иметь вид:



Рис. 31. Графический интерфейс приложения, хранящийся в файле graphic.fig

function varargout = graphic(varargin)

```
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
```

else

gui mainfcn(gui State, varargin{:}); end function graphic OpeningFcn(hObject, eventdata, handles, varargin) handles.output = hObject; guidata(hObject, handles); function varargout = graphic OutputFcn(hObject, eventdata, handles) varargout $\{1\}$ = handles.output; function radiobut xgrid Callback(hObject, eventdata, handles) % функция, обрабатывающая переключатель "xrid" if get(handles.radiobut xgrid,'Value') set(gca,'Xgrid','on') else set(gca,'Xgrid','off') end function radiobut_ygrid_Callback(hObject, eventdata, handles) % функция, обрабатывающая переключатель "vrid" if get(handles.radiobut ygrid,'Value') set(gca,'Ygrid','on') else set(gca,'Ygrid','off') end function graph Callback(hObject, eventdata, handles) % функция, обрабатывающая нажатие кнопки "graphic" x = [-2:0.1:5]; $y = exp(-x.^{2});$ plot(x,y); function clearbut_Callback(hObject, eventdata, handles) % функция, обрабатывающая нажатие кнопки "clear" cla:

ПРИЛОЖЕНИЕ 1

Операторы системы MATLAB

КОМАНДЫ ОБЩЕГО НАЗНАЧЕНИЯ		
Операторы	Назначение	
Ha	стройка параметров операционной системы	
matlabrc	Запуск системы MATLAB	
matlabroot	Определение корневого каталога	
diary	Ведение дневника сеанса работы	
printopt	Задание опций печати	
quit	Завершение работы в системе МАТLAВ	
exit	Завершение работы в системе MATLAB	
	Управление операционной системой	
!	Выполнить команду операционной DOS	
dos	Выполнить команду DOS и возвратить результат	
getenv	Получить значение переменной среды окружения	
cd	Изменить текущий каталог	
pwd	Путь доступа к текущему каталогу	
fileattrib	Получить/установить атрибуты файлов и каталогов	
copyfile	Копировать файл и каталог	
movefile	Переместить файл и каталог	
delete	Удалить файл	
dir	Содержимое текущего каталога	
filesep	Разделитель каталогов для используемой операционной	
	системы и возвратить результат	
isdir	Проверить, является ли аргумент каталогом	
mkdir	Создать каталог	
rmdir	Удалить каталог	
tempdir	Имя рабочего каталога DOS	
tempname	Имя временного файла	
system	Выполнить команду операционной системы и возвратить	
	результат	
perl	Выполнить команду языка Perl и возвратить результат	
unix	Выполнить команду ОС UNIX и возвратить результат	
computer	Определить тип используемого компьютера	
isunix	Проверить, является ли версия MATLAB версией для ОС	
ispc	Проверить, является ли версия MATLAB версией для PC (Windows)	

Управление командным окном		
clc	Очистить командное окно	
home	Поместить курсор в начальную позицию	
echo	Эхо-команда	
more	Управление выводом страниц	
format	Управление форматом вывода чисел в командное окно	
type	Вывод в командное окно содержимого ASCII-файла	
what	Список файлов в текущем каталоге	
type	Просмотр текста М-файла	
edit	Редактирование текста М-файла	
lookfor	Поиск М-файла по ключу	
ctrl+S	Контекстный поиск символов в командном окне	
tab completion	Дополнение имени до полного <символы>+Tab(+Tab)	
	Управление рабочей областью	
who	Список текущих переменных	
whos	Список текущих переменных с подробностями	
load	Считывание переменных с М-файла	
save	Запись переменных в М-файл	
clear	Удаление переменных и функций из памяти	
inmem	Список функций, загруженных в рабочую область памяти	
pack	Дефрагментация рабочей области памяти	
workspace	Активизировать окно рабочей области Workspace	
	Управление текущим каталогом	
fileparts	Выделить составляющие пути доступа	
fullfile	Сформировать полное имя файла из частей	
what	Вывод списка файлов	
which	Вывод списка файлов	
copyfile	Скопировать файл	
open	Открыть файл	
delete	Удалить файл	
filebrowser	Активизировать окно текущего каталога Current Directory	
Управление путями доступа		
matlabpath	Определить путь доступа	
pathdef	Определить путь доступа	
path	Определить/установить путь доступа	
pathsep	Определить разделитель в списке путей доступа	
partialpath	Определить частичный путь доступа	
genpath	Выделить подкаталоги	
path2rc	Сохранить список путей доступа	
addpath	Добавить каталог к пути доступа	

rmpath	Удалить каталог из пути доступа
rehash	Обновить кэш-память
editpath	Открыть окно Set Path
pathtool	Открыть окно Set Path
	Справочные команды
vertion	Справка об используемой версии МАТLАВ и виртуальной
	машины JAVA
ver	Справка о составе текущей версии системы MATLAB и ее
	расширений
license	Определение номера лицензии
info	Информация о системе МАТLАВ и фирме производителе
whatsnew	Документация Releanse Notes в окне help
help	Справка об операторе в командной строке
helpwin	Справка об операторе в окне help
helpdesk	Доступ к HTML-документации в окне help
helpbrowser	Доступ к HTML-документации в окне help
doc	Просмотр HTML-документов в окне help
web	Открыть web-браузер
finddemo	Определить расположение демонстрационных примерах в
	окне help
demo	Справка о демонстрационных примерах в окне help
	К ПАССЫ ОБЪЕКТОВ
	Операторы создания объектов
logical	Создать логическую переменную
char	Создать массив символов
single	Создать массив чисел одинарной точности
double	Создать массив чисел двумерной точности
int8	Преобразовать в 8-битовоецелое со знаком
int16	Преобразовать в 16-битовоецелое со знаком
int32	Преобразовать в 32-битовоецелое со знаком
int64	Преобразовать в 64-битовоецелое со знаком
uint8	Преобразовать в 8-битовоецелое без знака
uint16	Преобразовать в 16-битовоецелое без знака
uint32	Преобразовать в 32-битовоецелое без знака
uint64	Преобразовать в 64-битовоецелое без знака
cell	Создать массив ячеек
struct	Создать массив записей
inline	Созлать объект inline

ОПЕРАТОРЫ, СПЕЦИАЛЬНЫЕ СИМВОЛЫ,		
ПЕРЕМЕННЫЕ И КОНСТАНТЫ		
Арифметические операторы		
+ plus	Сложение	
+ uplus	Унарное сложение	
- minus	Вычитание	
- uminus	Унарное вычитание	
* mtimes	Умножение матриц	
.* times	Поэлементарное умножение для массива	
^ mpower	Возведение матрицы в степень	
.^ power	Возведение в степень для массивов	
\ mldivide	Левое деление матриц	
/ mrdivide	Правое деление матриц	
.\ ldivide	Левое деление для массивов	
./ rdivide	Правое деление для массивов	
kron	Тензорное произведение векторов	
	Операторы отношения	
== eq	Тождественно	
~= ne	Не тождественно	
< lt	Меньше	
> gt	Больше	
<= le	Меньше или равно	
>= ge	Больше или равно	
	Логические операторы	
& and	Логическое И	
or	Логическое ИЛИ	
	Логическое ИЛИ для операндов типа logical	
~ not	Логическое НЕТ	
xor	Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ	
&&	Логическое И для операндов типа logical	
all	Истинно, если все элементы вектора не равны нулю	
any	Истинно, если хотя бы 1 элемент вектора не равен нулю	
Специальные символы		
:	Сечение массива	
()	Указание последовательности выполнения операций	
[]	Формирование массива	
{}	Многомерные массивы	
•	Десятичная точка	
•	Выделение поля структуры	
	Указание на каталог-родитель	
•••	Продолжение строки	
,	Разделитель	

;	Подавление вывода результата
%	Комментарий
!	Вызов команды операционной системы
=	Присваивание
6	Кавычка
.'transpose	Транспонирование элементов массива
'ctranspose	Транспонирование элементов матрицы
[,] horzcat	Объединение элементов в строку
[;] vertcat	Объединение элементов в столбец
(),{},.subsasgn	Присваивание подмассива
(),{ },.subsref	Ссылка на подмассив
subsindex	Индекс подмассива
	Операторы поразрядной обработки
bitand	Поразрядное И
bitcmp	Биты дополнения
Bitor	Поразрядное ИЛИ
bitmax	Максимальное число разрядов
bitxor	Поразрядное ИСКЛЮЧАЮЩЕЕ ИЛИ
bitset	Задать бит
bitget	Узнать бит
bitshift	Поразрядный сдвиг
	Операторы обработки множеств
union	Объединение множеств
unique	Выделение множеств
intersect	Пересечение множеств
setdiff	Разность множеств
setxor	ИСКЛЮЧАЮЩЕЕ ИЛИ для множеств
ismember	Истинно, если это элемент множества
	Специальные переменные и константы
ans	Результат выполнения последней операции
eps	Машинная точность
realmax	Наибольшее число с плавающей точкой
realmin	Наименьшее число с плавающей точкой
pi	$\pi = 3,1415926535 89793 e + 000$
i, j	Мнимая единица
inf	Бесконечное значение
NaN	Нечисловое значение
isnan	Истинно, если нечисловое значение
isinf	Истинно, если бесконечное значение
isfinite	Истинно, если конечное значение
flops	Количество операций с плавающей точкой

ФУНКНИИ ВЫЧИСЛЕНИЯ ВРЕМЕНИ И ЛАТ	
	Текущее время и дата
clock	Текущее время и дата в виде вектора
date	Текущая дата в форме строки
now	Текущее время и дата в виде числа
	Основные функции
datenum	Последовательный номер даты с 01.01.0000
datestr	Строковое представление даты
datevec	Векторное представление даты
calendar	Календарь текущего месяца
weekday	День недели
eomday	Последний день месяца
datetick	Форматирование меток осей датой
cputime	Время работы центрального процессора в секундах
tic	Начало отсчета
toc	Конец отсчета
etime	Интервал использованного времени
MACC	ИВЫ, МАТРИЦЫ И ОПЕРАЦИИ НАД НИМИ
	Массивы и матрицы специального вида
zeros	Формирование массива нулей
ones	Формирование массива единиц
eye	Формирование единичной матрицы
repmat	Формирование многомерного массива на основе данного
rand	Формирование массива элементов, распределенных по
	равномерному закону
randn	Формирование массива элементов, распределенных по
	нормальному закону
linspace	Формирование линейного массива равноотстоящих узлов
logspace	Формирование узлов логарифмической сетки
meshgrid	Формирование узлов двухмерной и трехмерной сеток
:	Формирование векторов и подматриц
	Характеристики массивов
size	Размер массива
length	Длина вектора
ndims	Количество размерностей массива
isequal	Истинно, если два массива идентичны
isempty	Истинно, если массив пустой
isnumeric	Истинно, если массив числовой
islogical	Истинно, если массив имеет тип данных logical
logical	Преобразовать данные к типу logical

Операции над матрицами и массивами		
reshape	Преобразование размеров и размерностей матриц	
diag	Формирование или извлечение диагоналей матрицы	
tril	Формирование нижней треугольной матрицы	
triu	Формирование верхней треугольной матрицы	
fliplr	Отражение матрицы относительно вертикальной оси	
flipud	Отражение матрицы относительно горизонтальной оси	
flipdim	Отражение многомерного массива относительно указан-	
	ной размерности	
rot90	Поворот матрицы на 90 °	
find	Определить индексы ненулевых элементов	
end	Последний индекс многомерной матрицы	
sub2ind	Преобразование многомерной нумерации в последова-	
ind2sub	Преобразование последовательной в многомерную	
ОПЕРАЦИИ НАД МНОГОМЕРНЫМ И МАССИВАМИ		
oot	Многомерные массивы	
cat ndima	Розмериости массивы	
namis	Размерность массива	
nagria	Стенерировать сетку для многомерной функции	
permute	Перестановка размерностеи массива	
ipermute	Обратная перестановка размерностеи массива	
shiftdim	Изменить размерность массива	
squeeze	Удалить одну из размерностей	
	Операции над массивами ячеек	
	Массивы ячеек	
cell	Создать массив ячеек	
celldisp	показать содержимое массива ячеек	
cellplot	Показать графическую структуру массива ячеек	
deal	Установить соответствие входов с выходами	
iscell	Истинно, если это массив ячеек	
cell2struct	Преобразовать массив ячеек в массив структур	
num2cell	Преобразовать числовой массив в массив ячеек	
struct2cell	Преобразовать массив структур в массив ячеек	
	Операции над массивами записей	
struct	Создать массив записей	
fieldnames	Получить имена полей	
getfield	Получить содержание полей	
setfield	Установить содержимое полей	
rmfield	Удалить поле	

isfield	Истинно, если это поле массива записей	
isstruct	Истинно, если это массив записей	
	ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ	
	ПРОГРАММИРОВАНИЕ	
class	Создать объект или возвратить класс объекта	
methods	Показать методы данного класса	
isa	Истинно, если объект принадлежит данному классу	
iIsobject	Истинно, если это объект	
inferiorto	Отношение низшего класса	
superiorto	Отношение высшего класса	
	Переопределение методов	
minus	Переопределить метод для а-b	
plus	Переопределить метод для а+b	
times	Переопределить метод для а.*b	
mtimes	Переопределить метод для а*b	
mldivide	Переопределить метод для а\b	
mrdivide	Переопределить метод для а/b	
rdivide	Переопределить метод для а.\b	
ldivide	Переопределить метод для а./b	
power	Переопределить метод для а.^b	
mpower	Переопределить метод для а^b	
uminus	Переопределить метод для –а	
uplus	Переопределить метод для +а	
horzcat	Переопределить метод для [a b]	
vertcat	Переопределить метод для [a;b]	
le	Переопределить метод для a<=b	
lt	Переопределить метод для a <b< th=""></b<>	
gt	Переопределить метод для а>b	
ge	Переопределить метод для а>=b	
eq	Переопределить метод для а==b	
ne	Переопределить метод для а~=b	
not	Переопределить метод для ~а	
and	Переопределить метод для а&б	
0f	Переопределить метод для ap	
subsasgn	Персопределить метод для $a(1)=0$, $a\{1\}=0$, a .neu $=0$	
subsrei	Переопределить метод для а(1), а(1), а(1), а.неш	
transnoso	Переопределить метод для а.о	
ctranspose	Переопределить метод для а.	
subsinder	Переопределить метод для а	
SUDSILIUCA	переопределить метод для х(а)	

Отладка М-файлов	
debug	Просмотреть список команд отладки
dbstop	Задать контрольную точку
dbclear	Удалить контрольную точку
dbcont	Продолжить выполнение
dbdown	Прейти к сетке вызываемых М-функций сверху в низ
dbsteck	Вывести стек вызываемых М-функций
dbstatus	Просмотреть список контрольных точек
dbstep	Выполнить одну или несколько команд отладки
dbtype	Распечатать М-файл с пронумерованными строками
dbup	Перейти в стеке вызываемых М-функций снизу вверх
dbquit	Завершить отладку
Профилирование М-файлов	
profile	Профиль времени исполнения М-файла
profreport	Отчет о профиле

depfun	Определить список вызываемых функций для данных М-		
	файла		
depdir	Определить каталоги размещения вызываемых функций		
	для данного М-файла		
	Класс объектов INLINE		
	Массивы записей		
inline	Конструктор функции INLINE		
argnames	Получить имя аргументов функции INLINE		
char	Преобразовать объект INLINE в массив символов полей		
formula	Получить формулу функции INLINE		
vectorize	Представить функцию INLINE в векторную форму		
	Импорт и экспорт файлов		
uiimport	Запустить мастер импорта		
clipboard	Обмен массивами символов с буфером обмена		
finfo	Определить тип файла		
importdata	Загрузить данные из файла		
open	Открыть файл с заданным расширением		
saveas	Сохранить графическое окно в файле с заданным расши-		
	рением		
load	Прочитать переменные из М-файла		
save	Записать переменные в М-файл		
	Файлы в текстовом формате		
csvread	Прочитать числовой файл с разделителем запятая		
csvwrite	Записать числовой файл с разделителем запятая		
dlmread	Прочитать числовой файл с разделителем в виде ASCII-		
	кода		

dlmwrite	Прочитать числовой файл с разделителем в виде ASCII-	
	кода	
textread	Прочитать текстовый файл	
Файлы электронных таблиц		
xlsinfo	Получить информацию о таблице Microsoft Excel	
xlsread	Прочитать электронную таблицу Microsoft Excel	
wklinfo	Получить информацию о таблице Lotus 1-2-3	
wklread	Прочитать электронную таблицу Lotus 1-2-3	
wklwrite	Записать массив чисел в электронную таблицу Lotus 1-2-3	
	Графические файлы	
iminfo	Получить информацию о графическом файле	
imread	Считать графическое изображение из файла	
imwrite	Записать графическое изображение в файл	
	Аудиофайлы	
wavinfo	Получить информацию о звуковом файле с расширением	
	wav	
wavread	Считать звуковой файл с расширением Wav	
wavwrite	Записать звуковой файл с расширением Wav	
Видеофайлы		
aviinfo	Получить информацию о видео файле с расширением avi	
aviread	Считать видеофайл	
aviwrite	Создать видеофайл	
addframe	Добавить кадр к видеофайлу	
close	Закрыть видеофайл	
	Файлы в формате научных данных	
cdfinfo	Получить информацию о CDF-файле в формате Common	
	Data Format	
cdfread	Считать данные из CDF-файла	
fitsinfo	Получить информацию о FITS-файле в формате Flexible	
	Image Transport System	
fitsread	Считать данные из FITS-файла	
hdf	Интерфейс с библиотеками научных данных в форматах	
	hdf и hdf-eos	
hdfinfo	Получить информацию о файлах в форматах hdf и hdf-eos	
hdfread	Считать данные из файлов в форматах hdf и hdf-eos	
hdftool	Графический интерфейс для считывания данных из фай-	
	лов в форматах hdf и hdf-eos	
c)ПЕРАНИИ ВВОЛА–ВЫВОЛА ФАЙЛОВ	
	Открытие и закрытие файлов	
fonen	Открыть файл	
fclose	Закрыть файл	

	Двоичные файлы
fread	Прочитать двоичные данные из файла
fwrite	Записать двоичные данные из файла
	Форматированные файлы
fscanf	Прочитать форматированные данные из файла
fprintf	Записать форматированные данные в файл
fgetl	Прочитать строку файла, удалив символ конца строки
fgets	Прочитать строку файла, сохранив символ конца строки
input	Интерактивный вход
	Позиционирование файла
ferror	Запросить информацию об ошибке ввода-вывода
feof	Проверить признак конца файла
fseek	Установить указатель в заданную позицию
ftell	Запросить позицию указателя в файле
frewind	Установить указатель в начало файла
	МАТЕМАТИЧЕСКИЕ ФУНКЦИИ
	Специальные математические функции
airy	Функция Эйри
besselj	Функция Бесселя первого рода
bessely	Функция Бесселя второго рода
besselh	Функция Бесселя третьего рода
besseli	Модифицированная функция Бесселя первого рода
besselk	Модифицированная функция Бесселя второго рода
beta	Полная бета-функция
betainc	неполная бета-функция
betaln	Натуральный логарифм полной бета-функции
ellipj	Эллиптическая функция Якоби
ellipke	Полные эллиптические интегралы
erf	Функция ошибки
erfc	Остаточная функция ошибки
erfcx	Масштабированная остаточная функция ошибки
erfinf	Обратная функция ошибки
expint	Интегральная показательная функция
gamma	Полная гамма-функция
gammainc	неполная гамма-функция
gammaln	Натуральный логарифм полной гамма-функции
legendre	Функция Лежандра
cross	Векторное произведение векторов
dot	Скалярное произведение векторов

Теоретико-числовые функции	
fix	Округление числа до целого в сторону нуля
floor	Округление числа до ближайшего наименьшего целого
	числа
round	Округление числа до ближайшего наибольшего целого
	числа
mod(x,y)	Возвращает $x - n.*y$,где $n = floor(x./y)$, если $y \neq 0$.
rem	Возвращает x – n.*у,где n = fix(x./y), если у≠0
factor	Разложение числа на простые множители
isprime	Истинно, если число простое
primes	Формирование списка простых чисел
gcd	Наибольший общий делитель
icm	Наименьшее общее кратное
rat	Приближение числа в виде рациональной дроби
rats	Вычисление в поле рациональных чисел
perms	Формирование всех перестановок элементов вектора
nchoose	Вычисление числа сочетаний, C_n^k
factorial	Вычисление факториала числа
Фу	нкция преобразования систем координат
cart2sph	Преобразование декартовой системы в сферическую
cart2pol	Преобразование декартовой системы в полярную
pol2cart	Преобразование полярной системы в декартовую
sph2cart	Преобразование сферической системы в декартову
	ЛИНЕЙНАЯ АЛГЕБРА
	Коллекция матриц
company	Сопровождающая матрица
gallery	Пакет тестовых матриц
hadamard	Матрица Адамара
hankel	Матрица Ганкеля
hild	Матрица Гильберта
invhild	Матрица, обратная матрице Гильберта
magic	Магический квадрат
pascal	Матрица Паскаля
rosser	Матрица Рессера
teoplitz	Матрица Теплица
vander	Матрица Вандермонда
wilkinson	Матрица Уилкинсона

	Матричный анализ
norm	Вычисление норм векторов и матриц
rank	Вычисление ранга матрицы
det	Вычисление определителя матрицы
trace	Вычисление следа матрицы
null	Вычисление нуль-пространства матрицы
orth	Вычисление ортонормального базиса матрицы
rref	Приведение к треугольной форме
subspace	Вычисление угла между подпространствами
	Решение систем линейных уравнений
\ /	Решатели систем линейных уравнений
inv	Вычисление обратных матриц
cond	Вычисление числа обусловленности по отношению к за-
	даче обращения матрицы
chol	Разложение Холецкого
cholinc	Неполное разложение Холецкого
cholupdate	Разложение Холецкого матрицы А+-х*х`
lu	Lu-разложение
luinc	Неполное Lu-разложение
qr	Qr-разложение
qrupdate	Разложение Холецкого для матрицыА+u*v`
pinv	Псевдообращение по Муру-Пенроузу
iscov	Метод наименьших квадратов в присутствии шумов
Col	бственные значения и сингулярные числа
eig	Полная проблема собственных значений
svd	Сингулярное разложение
eigs	Вычисление отдельных собственных значений
svds	Вычисление отдельных сингулярных чисел
poly	Вычисление характеристического полинома
polyeig	Решение полиномиальных матричных уравнений
condeig	Вычисление числа обусловленности по отношению к за-
	даче на собственные значения
hess	Приведение матрицы к форме Хессенберга
qz	Обобщенная проблема собственных значений
schur	Приведение матрицы к форме Шура
	Вычисление функций от матриц
expm	Вычисление матричной экспоненты
logm	Вычисление логарифма матрицы
sqrtm	1
	Вычисление функции A^2
funm	Вычисление произвольных функций матриц

Утилиты	
qrdelete	Удалить столбец из QR-разложения
qrinsert	Добавить столбец в QR-разложение
rsf2csf	Преобразование действительной формы Шура в ком-
	плексную
cdf2rdf	Преобразование комплексной формы Шура в действи-
	тельную
balance	Масштабирование матриц
planerot	Формирование матрицы вращения Гивенса
	Полиномы и операции над ними
polyval	Вычисление полинома
polyvalm	Вычисление матричного полинома
poly	Вычисление характеристического полинома
residua	Разложение на простые дроби
roots	Вычисление корней полинома
polyfit	Аппроксимация данных полиномом
polyder	Вычисление производной полинома
polyint	Вычисление интеграла от полинома
conv	Умножение полиномов
deconv	Деление полиномов
РАБ	ОТА С РАЗРЕЖЕННЫМИ МАТРИЦАМИ
	Элементарные разреженные матрицы
sparse	Формирование разреженной матрицы
speye	Единичная разреженная матрица
sprand	Случайная разреженная матрица с элементами, распреде-
	ленными по равномерному закону
sprandn	Случайная разреженная матрица с элементами, распреде-
	ленными по нормальному закону
sprandsym	Случайная разреженная симметричная матрица
spdiags	Форматирование диагоналей разреженной матрицы
	Характеристики разреженных матриц
normest	Оценка 2-нормы разреженной матрицы
condest	Оценка числа обусловленности по 1-норме
sprank	Вычисление структурного ранга
	Преобразование разреженных матриц
full	Преобразование разреженной матрицы в полную
find	Определение индексов ненулевых элементов
spconvert	Восстановление разреженной матрицы из внешнего AS-
	СП-формата
	Работа с ненулевыми элементами
nnz	Количество ненулевых элементов
nonzeros	Формирование вектора ненулевых элементов

nzmax	Количество ячеек памяти для размещения ненулевых
	элементов
spones	Формирование матриц связности
spalloc	Выделить память для разреженной матрицы
issparse	Истинно, если матрица разреженная
spfun	Вычисление функции только для нулевых элементов
Оп	ерации над графом разреженной матрицы
etree	Вычисление дерева структуры
etreeplot	Построение дерева структуры
treelayout	Разметка дерева структуры
treeplot	Построение дерева структуры
	Алгоритмы упорядочения
colmmd	Упорядочение по разреженности столбцов
symmmd	Симметрическая упорядоченность
symrcm	RCM-упорядоченность
colperm	Упорядочение столбцов с учетом их разреженности
Randperm	Формирование случайных перестановок
dmperm	DM-декомпозиция разреженной матрицы
Решение систем линейных уравнений с разреженными матрицами	
pcg	Метод сопряженных градиентов
bicg	Двунаправленный метод сопряженных градиентов
bicgstab	Устойчивый двунаправленный метод
cgs	Квадратичный метод сопряженных градиентов
gmres	Метод минимизации обобщенной невязки
qmr	Квазиминимизация невязки
	Визуализация разреженных матриц
gplot	Построение графа структуры
spy	Визуализация структуры разреженной матрицы
	Вспомогательные операции
spparms	Установка параметров для алгоритмов обработки
symbfact	Характеристики разложения Холецкого
spaugment	Формирование расширенной матрицы для метода наи-
	меньших квадратов
	ОБРАБОТКА СТРОК
	Основные функции
blanks	Сформировать строку пробелов
cellstr	Преобразовать массив символов в массив ячеек для строк
char	Сформировать массив символов
deblank	Удалить пробелы в конце строки
double	Преобразовать символы строки в числовые коды

	Проверка строк	
ischar	Истинно, если это массив символов	
iscellstr	Истинно, если это массив ячеек для строк	
isletter	Истинно, если это символ алфавита	
isspace	Истинно, если это пробел	
	Операции над строками	
strcat	Горизонтальное объединение строк	
strvcat	вертикальное объединение строк	
strcmp	Сравнить строки	
strcmp	Сравнить п символов строк	
findstr	Найти заданную строку в составе другой строки	
strjust	Выровнять массив символов	
strmatch	Найти все совпадения	
strrep	Заменить одну строку другой	
strtok	Найти часть строки, ограниченную разделителями	
lower	Перевести символы строки в нижний регистр	
upper	Перевести символы строки в верхний регистр	
	Преобразование строк	
num2str	Преобразование числа в строку	
int2str	Преобразование целого в строку	
mat2str	Преобразование матрицы в строку	
str2mat	Объединение строк в матрицу	
str2num	Преобразование строки в арифметическое выражение и	
	его вычисление	
sprint	Записать форматированные данные	
sscanf	Прочитать строку с учетом формата	
	ОБРАЗОВАНИЕ СИСТЕМ СЧИСЛЕНИЯ	
hex2num	Преооразовать шестнадцатеричное число в число удво- енной точности	
hex2dec	Преобразовать шестнадцатеричное число в десятичное	
	число	
dec2hex	Преобразовать десятичное число в шестнадцатеричное	
	число	
bin2dec	Преобразовать двоичную строку в десятичное число	
dec2bin	Преобразовать десятичное число в двоичную строку	
base2dec	Преобразовать В-строку в десятичное число	
dec2base	Преобразовать десятичное число в В-строку	
АЛГОРИТМЫ ОБРАБОТКИ ЛАННЫХ		
	Базовые операции	
max	Максимальный компонент массива	
min	Минимальный компонент массива	

mean	Компонент средних значений массива
median	Компонент срединных значений массива
std	Компонент стандартных значений массива
sort	Сортировка по возрастанию
sortrows	Сортировка строк по возрастанию
sum	Суммирование элементов массива
prod	Произведение элементов массива
cumsum	Суммирование с накапливанием
cumprod	Произведение с накапливанием
ЧИСЛЕННО	Е ИНТЕГРИРОВАНИЕ И КОНЕЧНЫЕ РАЗНОСТИ
	Численное интегрирование
cumtrapz	Численное интегрирование метолом трапеций с накопле-
· · · · · · · · · · · · · · · · · · ·	нием
trapz	Численное интегрирование методом трапеций
quad	Численное интегрирование методом квадратур
quadl	Численное интегрирование методом Лобатто адаптивных
•	квадратур
dblquad	Вычисление двойного интеграла
triplequad	Вычисление тройного интеграла
Вычисление конечных разностей	
diff	Аппроксимация производных
gradient	Вычисление градиента функции
del2	Аппроксимация Лапласиана
РЕШАТ	ГЕЛИ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ
Обь	икновенные дифференциальные уравнения
ode113	Нежесткие ОДУ, метод переменных состояний
ode15s	жесткие ОДУ, метод переменных состояний
ode23	Нежесткие ОДУ, метод низкого порядка
ode23s	Жесткие ОДУ, метод Рунге-Кутты 3-го порядка
ode23t	Уметено жесткие ОДУ и смешанные дифференциально-
-	алгебраические уравнения, метод трапеций
ode23td	Жесткие ОДУ, метод низкого порядка
ode45	Нежесткие ОДУ, метод Рунге-Кутты 4-го порядка
dde4c	Решение ОДУ с запаздывающим аргументом
bvp4c	Решение ОДУ с двухточечными граничными условиями
Уран	внение в частных производных 1-го порядка
pdepe	Решение уравнений параболического и эллиптического
	типов с начальными граничными условиями
	Формирование выходов решателя
deval	Решение дифференциального уравнения
odeplot	Формирование процессов как функций времени

odephas2	Двухмерная фазовая плоскость
odephas3	трехмерная фазовая плоскость
odeprint	Командное окно вывода на печать
dvpinit	Формирование начальных приближений для функции
	BVP4C
pdeval	Интерполяция решения, полученного с помощью функ-
	ции PDEPE
	<u> </u>
ВЫЧИ	СЛЕНИЕ МИНИМУМОВ И НУЛЕИ ФУНКЦИИ,
	ОПТИМИЗАЦИЯ
Вычисление минимумов и нулей функций	
fmin	Вычисление минимума функции однои переменнои
fmins	Вычисление минимума функции нескольких переменных
fzero	Нахождение нулей нелинейной функции одной перемен-
	НОИ
e • 1	Оптимизация
Iminsearch	Безусловная нелинеиная минимизация функции многих
	переменных, метод нелдера-мида.
optimset	Создать/изменить опции оптимизации
	ОКСИМАЦИИ И ИНТЕГНОЛИЦИИ ДАННЫХ
Алгоритмы intorn1	
interp1	Бистрая одномериая интерполяция
interpre	Лаумериая таблициая интерполяция
interp2	Трехмериод табличная интерполяция
interps	Прехмерная таблиция интерполяция
interppi	Аппроксимация периодицеской функции
ariddata	Интерполяния на не равномерной сетке
griuuata griddata2	Трехмериод интерноляция из неравномериой сетке
griddatan	Прехмерная интерполяция на неравномерной сетке
griuuatan	Интерноляция на не равномерной сетке
pemp	мами Эрмита
nnvol	мами Эрмита
spline	Аппроксимация кусочно-гладкими полиномами
spinie	титерноляция кубическим спланном
	ГЕОМЕТРИЧЕСКИЙ АНАЛИЗ ЛАННЫХ
Алгоритмы	Пометти пеский инжино динных
delaunav	Триангуляция Лелона
delaunav3	Трехмерная триангуляция Лелона
delaunayo	N-мерная триангуляция Делона
dsearch	Триангуляция Ледона для ближайшей точки
dsearchnn	N-мерная триангуляция Лепона для ближайшей точки
ubcai ciiiii	ту мерния триан уляция делона для олиманшей точки

tsearch	Поиск наилучшей триангуляция	
tsearchn	Поиск наилучшей N-мерной триангуляция	
convhull	Вычисление выпуклой оболочки	
convhulln	Вычисление N-мерной выпуклой оболочки	
voronoi	Вычисление диаграммы Воронова	
voronoin	Вычисление N-мерной диаграммы Воронова	
inpolygon	Истинно, если точка внутри полигона	
rectint	Область пересечения треугольника	
polyarea	Область многоугольника	
ПРЕОГІ		
IIFEOD	АНАЛИЗ СИГНАЛОВ	
fft	Пискретное одномерное преобразование Фурье	
fft?	Дискретное одномерное преобразование Фурке	
fftn	Малиние дискретное преобразование Фурье	
ifft	Обратное одномерное преобразование Фурьс	
ifft?	Обратное двумерное преобразование Фурье	
ifftn	Обратное Мериое преобразование Фурке	
fftchift	Преобразование Фурье со сленгом постоящой состав-	
mishin	преобразование Фурве со едвигом постоянной состав-	
ifftshift	Обратное преобразование Фурье со слвигом постоянной	
mushit	составляющей	
	Свертка и фильтрация	
filter	Лискретная олномерная фильтрация	
filter2	Лискретная двумерная фильтрация	
conv	Свертка одномерных массивов	
conv2	Свертка двумерных массивов	
convn	Свертка N-мерных массивов	
deconv	Операция, обратная свертке	
detrend	Удаление линейного тренда	
	Корреляционный анализ	
corrcoef	Вычисление коэффициентов корреляции	
cov	Вычисление матриц ковариаций	
	ГРАФИКА	
	Задание осей координат	
axis	Масштабирование и вывод осей координат	
grid	Управление выводом сетки	
hold	Управление режимом сохранения графического окна	
subplot	Разбиение графического окна	
zoom	Изменение масштаба в графическом окне	

Надписи и обозначения к графикам	
xlabel	Обозначение оси х
ylabel	Обозначение оси у
zlabel	Обозначение оси z
clabel	Маркировка линий уровня
colorbar	Шкала палитры
title	Заголовок графика
text	Добавление к текущему графику текста
gtext	Размещение текста на графике с помощью мыши
legend	Пояснение к графику
Создание твердой копии и сохранение графика	
orient	Размещение твердой копии на странице
print	Вывод графика на печать или в файл
printopt	Задание опций печати по умолчанию
ДВУМЕРНАЯ ГРАФИКА	
	Элементарная графика
plot	График в линейном масштабе
loglog	График в логарифмическом масштабе
semilogx	График в полулогарифмическом масштабе по оси х
semilogy	График в полулогарифмическом масштабе по оси у
polar	График в полярных координатах
plotyy	График с двумя вертикальными осями
	Специальная графика
area	Закраска областей графика
bar	Столбцовая диаграмма
barh	Столбцовая диаграмма с горизонтальным расположением
comet	Движение точки по траектории
compass	График векторов-стрелок, исходящих из начала координат
arrorbar	График с указанием интервала погрешностей
feather	График векторов-стрелок, исходящих из равноотстоящих
	точек горизонтальной оси
fill	Закраска многоугольников
hist	Построение гистограммы
pareto	График результатов профилирования программы
pie	Круговая диаграмма
plotmatrix	График матрицы
auiver	График поля направления
ribbon	Изображение линий на трехмерном графике
stairs	Ступенчатый график
stem	Спафик лискретных значений
stem	трафик дискретивіх значений

ТРЕХМЕРНАЯ ГРАФИКА		
	Элементарная графика	
plot3	Построение линий и точек в трехмерном пространстве	
contour	Изображение линий уровня для трехмерной поверхности	
contour	Формирование массива описания линий уровня	
contour3	Изображение трехмерных линий уровня	
meshgrid	Формирование двумерных массивов Х и Ү	
mesh	Трехмерная сетчатая поверхность	
meshc	Трехмерная сетчатая поверхность с проекцией линий по-	
	стоянного уровня	
meshz	Трехмерная сетчатая поверхность с плоскостью отсчета	
	на нулевом уровне	
surf	Цветная сетчатая поверхность	
surfc	Цветная сетчатая поверхность с проекцией линий посто-	
	янного уровня	
surfl	Цветная сетчатая поверхность с подсветкой	
Специальная графика		
bar3	Трехмерная столбцевая диаграмм	
bar3h	Трехмерная столбцевая диаграмм с горизонтальным рас-	
	положением	
comet3	Движение точки по траектории в трехмерном простран-	
	стве	
contour	График линий уровня с раскрашенными областями	
fill3	Раскраска многоугольников в трехмерном пространстве	
pie3	Секторная диаграмма	
quiver3	График для направлений в трехмерном пространстве	
slice	Сечение функции в трех переменных	
stem3	График дискретных значений в трехмерном пространстве	
trimash	Трехмерная поверхность с треугольными ячейками	
trisurf	Трехмерная сетчатая поверхность с треугольными ячей-	
	ками	
waterfall	Трехмерная поверхность без прорисовки ребер сетки	
УПРАВЛЕН	ИЕ СВОЙСТВАМ И ГРАФИЧЕСКИХ ОБЪЕКТОВ	
	Управление цветом	
caxis	Установления соответствия между палитрой цветов и	
	масштабированием осей	
colormap	Палитра цветов	
colstyle	Выделить цвет и стиль для графика из заданного массива	
pcolor	Палитра псевдоцветов	

rgbplot	Изображение палитры	
spinmap	Вращение палитры	
hsv2rgb	Преобразование hsv-палитры в rgb-палитру	
rgb2hsv	Преобразование rgb-палитры в hsv-палитру	
shading	Затенение поверхностей	
brighten	Управление яркостью	
contrast	Палитра серого с повышенной контрастностью	
hidden	Управление удалением невидимых линий	
whitebg	Управление цветом фона	
colordef	Определить схему цветов для графического объекта	
	Figure	
	Палитры цветов	
autumn	Палитра с оттенками красного и желтого	
bone	Серая палитра с оттенками синего	
colorcube	RGB-палитра с оттенками серого	
cool	Палитра с оттенками голубого и фиолетового	
copper	Линейная палитра в оттенках меди	
hot	Палитра с чередованием черного, красного, желтого и	
	белого	
flag	Палитра с чередованием красного, белого, синего и черного	
grey	Линейная палитра в оттенках серого	
hsv	Палитра радуги	
jet	Разновидность hsv-палитры	
lines	Палитра, определяемая свойством ColorOrder	
pink	Розовая палитра с оттенками пастели	
prism	Палитра с чередованием красного, оранжевого, желтого,	
· ·	зеленого, синего, и фиолетового	
spring	Палитра с оттенками желтого и фиолетового	
summer	Палитра с оттенками желтого и зеленого	
white	Палитра с оттошкоми ролиборо и ролоноро	
winter	Палитра с оттенками голуоого и зеленого	
diffuso	Эффект лиффузиого расседния	
lighting	Управление половеткой	
material	Эффект рассеяния материала поверхности	
specular	Эффект зеркального отражения	
surfnorm	Построение нормалей к поверхности	
	Управление углом просмотра	
view	Управлением точки просмотра	
viewmtx	Вычисление матрицы управления углом просмотра	
rotate3d	Интерактивные повороты трехмерного объекта	
L		

ОБЪЕМНЫЕ ТЕЛА	
Объемные графические объекты	
cylinder	Объемный цилиндр
ellipsoid	Объемный эллипсоид
patch	Закрашенный многоугольник
sphere	Объемная сфера
surf2patch	Преобразовать графический объект Surface в графиче- ский объект Patch
	Визуализация объемных тел
coneplot	Построить векторы скоростей в виде конуса трехмер-
contourslice	Построить линии постоянного уровня в сечениях объема
curl	Вычислить ротор и угловую скорость векторного поля
divergence	Вычислить дивергенцию векторного поля
flow	Функции скорости потока струи в бесконечном объеме
interpstreamspeed	Интерполировать линии тока, используя значения ско-
isocaps	Вычислить геометрию торца поверхности постоянного уровня
iso	Вычислить нормали в вершинах поверхности постоянного уровня
iso	Извлечь данные о поверхности постоянного уровня из описания объемного тела
reducepatch	Редукция количества объектов Patch
reducevolume	Редукция количества элементов для описания объем- ного тела
shrinkface	Сокращение размеров объектов Patch
slice	Изобразить плоскости срезов в объеме
smooth3	Сглаживание данных по трем измерениям
stream2	Вычислить линии тока в двух измерениях
stream3	Вычислить линии тока в трех измерениях
streamline	Изобразить линии тока
streamparticles	Изобразить частицы потока для объемных данных
streamribbon	Изобразить ленты потока для объемных данных
streamslice	Изобразить линии потока для объемных данных
streamtube	Изобразить струи потока для объемных данных
surf2patch	Преобразовать объект Surface в объект Patch
subvolume	Выделить подмножество из объемных данных
volumebounds	Возвратить предельные значения координат и цветов для объемных данных

ПОСТРОЕНИЕ ГРАФИКОВ л ня объектов ini ine и функций с лескриштором		
Дли ОДДЕКТОД ПАЛИЕ И ФУЛКЦИИ С ДЕСКГИПТОРОМ		
emlot	Построение графика функции	
ezplot	Построение параметрической кривой в трехмерном	
ezpiote	пространстве	
ann al an		
ezpolar	построение графика функции в полярных координа-	
ozcontour		
ezcontourf	Построение закращении у областей с линиями уровия	
ozmosh	Построение траумарии и сатиати и пораруностай	
ezmeshe	Построение трехмерных сетчатых поверхностей	
ezinesite	ниями уровня	
ezsurf	Построение трехмерных пертных сетчатых поверуно-	
V25011	стей	
ezsurfc	Построение трехмерных цветных сетчатых поверхно-	
0100110	стей с линиями уровня	
fplot	Построение графика явной функции	
•		
	ДЕСКРИПТОРНАЯ ГРАФИКА	
f" ann a	Ооъекты дескрипторной графики	
ngure	Графический объект Figure	
line	Графический объект Ахез	
nostangla	Графический объект Lille	
toyt	Графический объект Кестанде	
surface	Графический объект Техт	
notch	Графический объект Sufface	
light	Графический объект Гасп	
imaga	Графический объект Light	
Cor		
gigure	Открыть графическое окно	
gref	Получить дескриптор графического объекта Figure	
clf	Очистить графическое окно	
sho	показать графическое окно	
close	закрыть графическое окно	
refresh	Обновить графическое окно	
C	оздание и управление осями координат	
axes	Создать оси координат	
box	Окружить оси прямоугольником или параллелепипе-	
	дом	
cla	Очистить оси координат	

gca	Получить дескриптор графического объекта Axes
hold	сохранить оси координат
ishold	Истинно, если оси координат сохранены
Оп	ерации над графическими объектами
set	Установить свойства графического объекта
get	Получить свойства графического объекта
reset	Восстановить штатные значения свойств
delete	Удалить графический объект
gco	Получить дескриптор текущего объекта
ginput	Снять координаты точки с помощью мыши
drawnow	Выполнить очередь задержанных графических команд
findall	Найти все графические объекты
allchild	Найти всех потомков графического объекта
findobj	Найти объекты с заданными свойствами
copyobj	Скопировать сам объект и порожденные им графиче-
	ские объекты
setappdata	Приписать данные графическому объекту
getappdata	Получить данные графическому объекту
rmappdata	Удалить данные приписанные графическому объекту
isappdata	Истинно, если данные приписаны графическому объ-
	екту
Утилиты	
closereq	Запрос на закрытие графического окна
ishandle	Истинно, если это дескриптор
newplot	Восстановление штатных значений свойства NextPlot
	графинеский интермейс
	ГГАФИЧЕСКИИ ИНТЕГФЕИС
uicontrol	Создать управляющий интерфейс
uimenu	Создать управляющий интерфене
uicontextmenu	Создать контекстное меню
ulcontextinent	
	Средство проектирования GUI
guide	Средство проектирования GUI Вызов интерактивного средства GUIDE
guide propedit	Средство проектирования GUI Вызов интерактивного средства GUIDE Редактор свойств
guide propedit align	Средство проектирования GUI Вызов интерактивного средства GUIDE Редактор свойств Средство вырывания и расположения объектов
guide propedit align cbedit	Средство проектирования GUI Вызов интерактивного средства GUIDE Редактор свойств Средство вырывания и расположения объектов Редактор ответных вызовов
guide propedit align cbedit menuedit	Средство проектирования GUI Вызов интерактивного средства GUIDE Редактор свойств Средство вырывания и расположения объектов Редактор ответных вызовов Редактор меню
guide propedit align cbedit menuedit	Средство проектирования GUI Вызов интерактивного средства GUIDE Редактор свойств Средство вырывания и расположения объектов Редактор ответных вызовов Редактор меню Диалоговые панели
guide propedit align cbedit menuedit dialog	Средство проектирования GUI Вызов интерактивного средства GUIDE Редактор свойств Средство вырывания и расположения объектов Редактор ответных вызовов Редактор меню Диалоговые панели Создание шаблона диалоговой панели
guide propedit align cbedit menuedit dialog msgbox	Средство проектирования GUI Вызов интерактивного средства GUIDE Редактор свойств Средство вырывания и расположения объектов Редактор ответных вызовов Редактор меню Диалоговые панели Создание шаблона диалоговой панели Диалоговая панель сообщений
guide propedit align cbedit menuedit dialog msgbox errordlg	Средство проектирования GUI Вызов интерактивного средства GUIDE Редактор свойств Средство вырывания и расположения объектов Редактор ответных вызовов Редактор меню <u>Диалоговые панели</u> Создание шаблона диалоговой панели Диалоговая панель сообщений Диалоговая панель сообщений об ошибки

helpdlg	Диалоговая панель подсказок
inputdlg	Диалоговая панель ввода
listdlg	Диалоговая панель просмотра списка
questdlg	Диалоговая панель вопроса
pagedlg	Диалоговая панель расположения страниц
printdlg	Диалоговая панель печати
waitbar	Стандартная панель ожидания
uigetfile	Стандартная диалоговая панель открытия файла
uiputfile	Стандартная диалоговая панель записи файла
uisetcolor	Стандартная диалоговая панель выбора цвета
uisetfont	Стандартная диалоговая панель выбора шрифта
	Операции над объектами GUI
gcbo	Получить дескриптор повторно вызываемого объекта
gcbf	Получить дескриптор повторно вызываемого графи-
	ческого окна
waitforbuttonpress	Ожидание нажатия кнопки клавиатуры или мыши
rbbox	Растянуть прямоугольник с помощью мыши
selectmoveeresize	Выбор, перемещение, изменение размеров, копирова-
	ние объектов с помощью мыши
dragrect	Переместить треугольник с помощью мыши
waitfor	Блокировать выполнение в ожидание события
uiwait	Блокировать выполнение в ожидание возобновление
uiresume	Возобновить выполнение после блокирования
РАБОТА (С ГРАФИЧЕСКИМИ ИЗОБРАЖЕНИЯМИ
	Базовые операции
image	Вывод графического изображения
imagesc	Масштабирование и вывод графического изображения
imfifo	Информация о структуре графического файла
imread	Read image from graphics file
imwrite	Write image to graphics file
im2java	Convert image to Java image
Утилиты	
brighten	Brighten or darken color map
colorbar	Display color bar
colormap	Color look-up table
contrast	Gray scale color map to enhance image contrast
gray	Linear gray-scale color map

АНИМАЦИЯ И ВИДЕОКЛИПЫ			
	Анимационные возможности		
capture	Захват графической фигуры		
getframe	Создать фрейм для анимации		
moviein	Выделить память под фреймы анимации		
movie	Выполнить анимацию		
rotate	Вращение графического объекта		
frame2im	Преобразование фрейма в графический обзор		
im2frame	Преобразование графического обзора в фрейм		
]	РАБОТА С ОБЪЕКТАМИ ЗВУКА		
	Объекты звука		
audioplayer	Windows audio player object		
audiorecorder	Windows audio recorder object		
Воспроизведение зву	ка		
sound	Озвучить одномерный массив чисел		
soudsc	Масштабировать и озвучить одномерный массив чи-		
	сел		
wavplay	Play sound using windows audio output device		
wavrecord	Record sound using windows audio input device		
Утилиты	Утилиты		
mu2lin	Преобразование μ -кодированного сигнала в линей-		
	ный		
lin2mu	Преобразование линейного сигнала		
	в μ -кодированый		
ИНТЕРФЕЙСЫ С ОС WINDOWS			
	Интерфейс DDE		
ddeadv	Установить консультативную связь		
ddeexec	Послать строку на выполнение		
ddeinit	Инициировать DDE-диалог		
ddepoke	Послать данные в приложении		
ddeterm	Завершить DDE-диалог		
ddeunadv	Завершить консультативную связь		
Интерфейс ActiveX			
actxcontrol	Создать элемент управления		
actxserver	Создать локальный или удаленный сервер		
active	Конструктор интерфейса ActiveX		
mwsamp	Пример скрипта, создающего объект ActiveX		
sampev	Пример обработчика событий для объекта ActiveX		

ЛИТЕРАТУРА

- 1. Потемкин В.Г. МАТLАВ: среда проектирования инженерных приложений. – М.: Диалог-МИФИ, 2003. – 447 с.
- 2. *Ануфриев И.Е.* Самоучитель MatLab 5.3/6.х. СПб.: БХВ-Петербург, 2004. 736 с.
- 3. Потемкин В.Г. Справочник по МАТLAB. СПб.: Питерпресс, 2002. 248 с.

Н.Л. Кучеренко

МАТLАВ: ТИПЫ ДАННЫХ, МАССИВЫ, РАБОТА С ФАЙЛАМИ, ГРАФИКА, ИНТЕРФЕЙС

Редактор В.В. Мокрынина Компьютерная верстка Ю.А. Ларичевой

Подписано в печать 23.03.11. Формат 60х84 ^{1/}₁₆ Офсетная печать. Объем 6,0 п. л. Тираж 60 экз. Заказ 169.

> Отпечатано в типографии КРСУ 720048, г. Бишкек, ул. Горького, 2