

**Государственный комитет Российской Федерации  
по высшему образованию**

**Самарский государственный аэрокосмический  
университет имени академика С.П.Королева**

**АВТОМАТИЗАЦИЯ ФИЗИЧЕСКОГО  
ЭКСПЕРИМЕНТА НА БАЗЕ ПАКЕТА  
МАТРИЧНЫХ ВЫЧИСЛЕНИЙ**

**Методические указания**

**Самара 1994**

Составитель С.В. Суханов

УДК 681.3

Автоматизация физического эксперимента на базе пакета матричных вычислений: Метод. указания / Самар. гос. аэрокосм. ун-т; Сост. С.В. Суханов. Самара, 1994. 20 с.

Указания содержат порядок работы с пакетом программ матричных вычислений MATLAB для персональных ЭВМ, совместимых с IBM PC. Описаны важнейшие функции пакета, язык командных файлов и другая информация, необходимая начинающему пользователю.

Указания предназначены для студентов факультета "Информатика" специальности 01.04 "Физика", выполняющих лабораторные работы по курсу "Автоматизация физического эксперимента", а также специальности 01.02 "Прикладная математика", выполняющих лабораторные работы и курсовые проекты по курсам: "Цифровая обработка сигналов", "Методы прикладной математики", "Планирование эксперимента и статистический анализ". Указания будут также полезны для широкого круга пользователей персональных ЭВМ. Подготовлены на кафедре технической кибернетики.

Печатаются по решению редакционно-издательского совета Самарского государственного аэрокосмического университета имени академика С.П. Королева

Рецензент В.И. Чекин

## 1. ОБЩИЕ СВЕДЕНИЯ

В зарубежных исследовательских центрах и университетах для компьютерной обработки результатов физических экспериментов широко применяется пакет MATLAB, программный продукт фирмы The MathWorks, Inc.

В настоящих методических указаниях даны элементарные сведения о пакете MATLAB, необходимые начинающему пользователю. Более подробные (с примерами) сведения об этом пакете рассмотрены в [1]. При написании методических указаний был использован текст [2], распространяемый в виде компьютерного файла на некоммерческой основе. Приводимые в тексте примеры представляют собой вывод на экран монитора информации, вводимой пользователем, и выводимой компьютером, поэтому знаки пунктуации, обеспечивающие синтаксическую правильность текста, даны с отступом.

### 1.1. Краткая характеристика пакета

Ядром пакета является интерпретатор, который понимает несложный интерактивный язык программирования высокого уровня, напоминающий Бейсик. Пользователь может работать с пакетом в двух режимах: непосредственно выполняя команды с клавиатуры; запуская свои программы. Язык пакета в основном реализует принципы модульного программирования; например, допускает написание многих командных файлов (программ) пользователя с возможностью вызова встроенных и библиотечных функций, а также других программ пользователя.

Главной отличительной чертой языка системы, отражающей общую направленность пакета MATLAB (что заметно из названия), является ориентация на операции матричной алгебры. Большая часть встроенных функций языка и библиотечных файлов реализует широкий круг операций с матрицами и векторами. Поэтому, в частности, в языке отсутствуют понятия массива и других типов данных, но есть понятия матрицы и вектора.

Большой репертуар встроенных функций и обширная библиотека функций позволяют реализовать вычисления для достаточно широкого круга математических задач и инженерных приложений. Простота программирования и отладки, унифицированная форма ввода/вывода позволяют пользователю максимально сосредоточиться на содержательной части своей задачи, что делает рассматриваемый программный продукт особенно полезным для студентов.

Графические возможности пакета достаточно удобны для пользователя, хотя и не отличаются особой гибкостью. Набор графических функций системы позволяет выводить на экран, принтер или в специальный графический метафайл графики зависимостей в различных системах координат и трехмерные поверхности. Отдельный

программный модуль - графический постпроцессор - позволяет из созданного графического метафайла выводить в пакетном режиме графическую информацию на различные графические устройства (дисплеи, принтеры, плоттеры, лазерные принтеры).

Для передачи данных (числовых и текстовых) из других программных продуктов в среду MATLAB и обратно (экспорт/импорт файлов) система имеет транслятор данных, который позволяет преобразовывать стандартные текстовые файлы, файлы электронных таблиц, файлы данных Фортрана и другие в файлы, загружаемые в среду MATLAB с помощью функции load.

## 1.2. Структура файлов пакета

### Файлы пакета

matlab.bat	- стартовый файл системы;
pcmatlab.exe	- интерпретатор языка, основной модуль системы;
pcmatlab.ove	- модуль перекрытий для основного модуля;
matlab.hlp	- информация о встроенных функциях системы;
graph.bat	- стартовый файл графического постпроцессора;
GPP.exe	- графический постпроцессор;
translate.exe	- транслятор файлов данных;
имя.ш	- библиотечные функции пакета.

### Файлы пользователя

имя.ш	- командные файлы; библиотечные функции (процедуры);
имя.mat	- файлы данных во внутреннем формате пакета;
имя.met	- графические метафайлы.

Пользователю рекомендуется иметь как минимум два каталога: один - для файлов пакета и другой - для своих файлов. Полезно разделить исполняемые файлы пакета и библиотеку функций. В рабочем каталоге пользователя, откуда запускается (вызывается) пакет, следует иметь копию файла matlab.bat. В данном файле в команде set MATLABPATH должны быть перечислены спецификации всех каталогов, в которых хранятся файлы пакета и пользователя.

## 1.3. Начало и окончание работы пакета

Для запуска пакета MATLAB достаточно исполнить стартовый файл matlab.bat, находясь в рабочем каталоге пользователя. Вообще, для запуска из любого другого каталога, не содержащего файла matlab.bat, надо указать полную спецификацию файла matlab.bat. При этом файлы, создаваемые пакетом и пользователем, по умолчанию попадут в тот каталог, откуда был выполнен запуск. Следует избегать смешивания

своих файлов с библиотечными функциями пакета. Если в файле `matlab.bat` не указаны имена каталогов, где операционная система должна искать нужные файлы, вероятно ситуация, когда запуск пакета не выполняется, так как нужные файлы не найдены.

Файл `matlab.bat` обеспечивает вызов модуля `rsmatlab.exe` и организует цикл, обеспечивающий при написании пользователем командных файлов автоматический выход из интерпретатора и вызов редактора текста, а при выходе из редактора - автоматический возврат в среду интерпретатора. При этом обеспечивается автоматическое сохранение состояния рабочей среды (переменные, функции и проч.), предшествовавшего сеансу редактирования. Пользователь может подключить свой редактор текста к пакету MATLAB, вставив его вызов в текст файла `matlab.bat`.

Начав работать, `rsmatlab.exe` демонстрирует свою "торговую марку" на экране, исполняет файл `matlab.m`, который вызывает файл `startup.m`. В этот файл пользователь может внести нужные сообщения и функции пакета MATLAB для настройки на выполнение определенной задачи. По окончании работы `startup.m` на экране появляется отклик системы " п ", а справа от него - мерцающий курсор. Система готова к работе

Для окончания работы нужно выполнить одну из следующих функций: `quit`, `exit` или нажать комбинацию из двух клавиш `Ctrl_Z`. Если пользователь желает продолжить в последствии работу "с того же места", то перед выходом из пакета надо записать текущее состояние рабочей области MATLAB с помощью функции `save имя_файла`. А в последствии, после запуска пакета, - выполнить функцию `load имя_файла`, чтобы загрузить содержимое рабочей области из файла `имя_файла.mat`.

#### 1.4. Подсказки и демонстрации

Начинающий пользователь на первом этапе может обращаться за подсказками (`help`) и демонстрациями (`demo`), причем в зависимости от версии пакета MATLAB информация, выдаваемая на экран, может быть на английском, русском или частично русском языках.

Функция `help` выводит на экран сначала список всех встроенных функций, символов и понятий, описываемых в файле `matlab.hlp`. Затем последовательно выводятся списки всех функций (файлов с расширением `.m`) из всех каталогов, которые указаны пользователем в команде `set MATLABPATH` файла `matlab.bat`.

Если пользователь желает получить краткую информацию о конкретной встроенной функции, библиотечной функции, функции "собственного производства" (правила оформления функций описаны в п. 1.5) или о своем командном файле, то он должен набрать после `help` соответствующее имя (имя файла без расширения). Для командных файлов (в том числе библиотечных функций) в качестве информации

выводятся на экран первые строки комментариев (текст после знака % ). Если пользователь наберет лишь одну, две или несколько букв имени встроенной функции, то будет выведена на экран справочная информация о всех встроенных функциях, имена которых начинаются с этих букв. Для имен файлов такой поиск не предусмотрен, поэтому имя файла или библиотечной функции надо набирать полностью и точно.

Функция `demo` вызывает на экран меню демонстрационных программ, которые иллюстрируют возможности пакета. Часть из этих программ ведет диалог на русском языке. Начинающему пользователю для ознакомления с вычислительными и графическими возможностями пакета рекомендуется выполнить каждую из демонстрационных программ. Тексты демонстрационных программ и библиотечных функций полезно изучать для освоения стиля программирования в среде пакета MATLAB, а фрагменты этих текстов пользователь может позаимствовать для своих разработок (список демонстрационных программ содержится в тексте библиотечной функции `demo`, файл `demo.m`).

### 1.5. Создание командных файлов

Командные файлы (программы) пользователя состоят из обращений к встроенным и библиотечным функциям пакета, другим командным файлам и функциям пользователя. Имя файла должно иметь расширение `.m`, только тогда возможен его вызов (по имени, без расширения). Командный файл рекомендуется начинать с одной или нескольких строк комментариев (текст после знака % ) до первой исполняемой функции, что даст возможность получить о нем справку с помощью функции `help`.

Важной разновидностью командных файлов являются функции. В виде функции имеет смысл оформлять небольшие отложенные программы (процедуры). Одно из главных отличий функции: используемые в ней переменные являются локальными, то есть действуют в пределах этой функции. В обычном командном файле все переменные глобальные. В функции же часть переменных может быть объявлена глобальными с помощью функции `global` ( см. п. 2.2.1).

Для того чтобы оформить свой командный файл как функцию, необходимо ввести первой строкой файла специальный заголовок функции:

```
function [список вых. перем.]=имя_функции(список вх. перем. )
```

Пример:

```
function [t,y]=ode(F,t0,tf,tol,tr)
```

```
% ode – решение обыкновенного
```

```
% дифференциального уравнения.
```

```
Операторы функции.
```

Тексты командных файлов (функций) пользователь может вводить и редактировать в режиме редактирования, выполнив функцию `edit`. При исполнении этой команды происходит выход из пакета MATLAB с сохранением в файле `edit.mat` информации о состоянии среды (значения переменных и проч.), вызов редактора текста, который пользователь указал в файле `matlab.bat`. Пользователь набирает нужное имя файла (с расширением `.m`) и редактирует этот файл. При выходе из редактора (не забудьте отредактированный файл записать на диск) автоматически загружается пакет MATLAB и восстанавливается состояние среды из файла `edit.mat`. Система готова к отладке (выполнению) созданного командного файла.

Если достаточен объем оперативной памяти, удобнее и быстрее вызывать текстовый редактор без выгрузки пакета MATLAB. Например, редактор `word` может быть вызван следующим образом:

```
!word имя файла.m
```

Эта команда сохраняется в стеке MATLAB, может быть многократно вызвана (клавишами управления курсора вниз и вверх) и использована без повторного набора. Символ `!` позволяет выполнять из среды MATLAB любые команды MS-DOS, запускать любые внешние файлы.

## 2. ФУНКЦИИ ПАКЕТА MATLAB

При выполнении функций пакета в непосредственном режиме с клавиатуры часто возникает необходимость повторного многократного вызова одних и тех же функций. MATLAB запоминает в специальном стеке ограниченного объема все инструкции пользователя, введенные с клавиатуры (наиболее "ранние" вытесняются из стека и теряются). Пользователю следует для вызова выполнявшихся ранее функций из стека использовать клавиши управления курсором (вверх и вниз).

### 2.1. Использование служебных символов и переменных

"[ ]" - квадратные скобки используются для задания векторов и матриц. Элементы вектора и элементы строки матрицы разделяются запятыми или пробелами, одна строка от другой отделяется точкой с запятой. Элементами матриц могут быть имена векторов и матриц. В этом случае контроль за соответствием числа строк и столбцов возлагается на пользователя. Например,

$A = []$  - пустая матрица;

$A = 7$  и  $A = [7]$  - эквивалентны;

$[3\ 5\ 6; 7\ 8\ 9]$  и  $[3,5,6;7,8,9]$  - эквивалентны;

[6.9 9.64 log(3.62)] - вектор из трех элементов;

A = [B,C;D] - задание через ранее определенные матрицы.

"( )" - круглые скобки могут использоваться для изменения порядка выполнения операций в выражениях, для задания аргументов функций, для задания конкретных значений индексов матриц и векторов. Примеры:

X(3) - третий элемент вектора X;

X([1 2 3]) и X([1.41,sqrt(3),4\*atan(1)]) - первые три элемента вектора X (с учетом округления для получения целых индексов);

X(V) = [X(V(1)), X(V(2)), ..., X(V(N))], - здесь X и V - векторы.

" . " - десятичная точка помимо разделения целой и дробной части числа используется для обозначения поэлементных операций.

" , " - запятая может использоваться для разделения элементов в строках матриц и векторов, для разделения индексов, аргументов функций и для разделения в строках программы.

" ; " - точка с запятой может использоваться внутри квадратных скобок для отделения одной строки матрицы от другой, а также для разделения в строках программы, что позволяет отключить вывод на экран результатов расчетов.

" % " - процент означает логический конец строки программы. Следующий за ним текст игнорируется, что позволяет использовать этот знак для помещения комментариев в текст программы.

" ! " - любой текст, следующий за восклицательным знаком, рассматривается как команда MS-DOS, которая выполняется без загрузки пакета MATLAB.

" : " - двоеточие удобно использовать в матрицах и циклах FOR при задании индексов, для создания целочисленных и нецелочисленных последовательностей (векторов). Примеры:

J:K - это [ J, J+1, J+2, ..., K ];

J:I:K - это [ J, J+I, J+2I, ..., K ];

0:pi/2:pi - это [0, pi/2, pi];

A(J:K) - это A(J), A(J+1), ..., A(K);

A(:,J) - это j-й столбец матрицы A.

" ' " - апостроф используется для задания значений текстовых переменных. Любой текст в апострофах рассматривается как вектор, компонентами которого являются ASCII - коды символов текста.

" = " - знак равенства используется как оператор присваивания.

Служебные переменные: ANS - при отсутствии левой части оператора (чему присваивать) значение вычисленного выражения присваивается этой переменной. NARGIN - переменная, содержащая число входных аргументов программы-функции. NARGOUT - переменная, содержащая число выходных аргументов программы-функции. PI = 4\*atan(1) = 3.1415926535897 - встроенная константа.

<i>Знаки арифметических и алгебраических операций:</i>	
+	- сложение; размерности матриц должны совпадать;
-	- вычитание; размерности матриц должны совпадать;
*	- матричное умножение;
.*	- поэлементное умножение;
/	- правое ( \ - левое ) матричное деление;
./	- поэлементное деление;
^	- возведение в степень;
.^	- поэлементное возведение в степень.
<i>Знаки логических операций:</i>	
&	- логическое "и";
	- логическое "или";
~	- логическое "не".
<i>Знаки отношений:</i>	
>	- больше;
<	- меньше;
>=	- больше или равно;
<=	- меньше или равно;
==	- равно (эквивалентно);
~=	- не равно.

## 2.2. Функции языка программирования

### 2.2.1. Управление ходом вычислений, областью переменных

<i>Условный оператор.</i>	
<p>Первая форма:</p> <pre>if условие, операторы else операторы end</pre>	<p>Вторая форма:</p> <pre>if условие, операторы elseif условие, операторы end</pre>

### Оператор цикла.

Первая форма:  
**for** *переменная* = *выражение*,  
*операторы*  
**end**

Вторая форма:  
**while** *выражение*,  
*операторы*  
**end**

В первой форме число итераций заранее известно; столбцы, вычисляемые в *выражении*, присваиваются поочередно *переменной* и выполняются *операторы* до **end**. Чаще, однако, используются скалярные переменные (векторы размерности 1). В этом случае удобно использовать *выражение* в форме  $X:Y:Z$  или  $X:Z$  (здесь по умолчанию  $Y=1$ ), где  $X$  - начальное,  $Z$  - конечное значение,  $Y$  - шаг.

Во второй форме число итераций заранее неизвестно; *операторы* выполняются, пока результат вычисления *выражения* имеет ненулевые элементы.

**BREAK** - эта функция используется для прерывания выполнения цикла, если это необходимо сделать до окончания перебора всех вариантов (напомним: оператор безусловного прехода типа **go to** в языке отсутствует).

Пример 1:

```
for I=1:N,
  for J=N:-0.1:1.0,
    A(I,J)=1/(I+J-1);
    if A(I,J)==0.5, break; end
  end
end
```

Пример 2:

```
N=1;
while A >= .01,
  A=1/N; N=N+1;
end
```

**Внимание!** Избежать использования оператора цикла в тривиальных случаях (при вычислении суммы или произведения) помогут две следующие функции. Их использование на порядок увеличивает быстродействие.

**SUM(X)** - это сумма элементов  $X$ , если  $X$  - вектор. Если  $X$  - матрица, то **sum(X)** - вектор-строка, элементы которой - суммы элементов столбцов матрицы  $X$ .

**PROD(X)** - это произведение элементов  $X$ , если  $X$  - вектор. Если  $X$  - матрица, то **prod(X)** - вектор-строка, элементы которой - произведения элементов столбцов матрицы  $X$ .

Следующие четыре функции помогут пользователю оценить трудоемкость вычислений, повысить гибкость работы своих программ.

**CASESEN** - отключает (а при повторном обращении включает) режим различения прописных и строчных букв.

**CLOCK** - возвращает вектор-строку:

```
A = clock
A = [<год> <месяц> <день> <часы> <минуты> <секунды>]
```

**EVAL(X)** - функция текстовой макроподстановки; например

```
x=4; st='2*sqrt(x)'; y=eval(st)
```

```
y=4.0000 .
```

**FLOPS** - возвращает текущее значение счетчика числа операций с плавающей точкой; **flops(0)** - сброс счетчика.

Следующие шесть функций необходимы для работы с программами (функциями) пользователя.

**ECHO** - позволяет включать/выключать режим эхо-печати в программных файлах пользователя. При включенном режиме на экран последовательно выводятся выполняемые операторы программы. Варианты использования:

```
echo on/off - включает/выключает режим в текущем файле;
```

```
echo on/off all - включает/выключает режим для всех файлов библиотечных функций;
```

```
echo имя on/off - включает/выключает режим для файла библиотечной функции имя.ш.
```

**ERROR('сообщение')** - выводит на экран текст *сообщения* и прерывает выполнение программы. Если аргументом является пустая матрица, никакие действия не выполняются.

**GLOBAL** - объявляет глобальные переменные. Например, функция **global X Y Z** объявляет глобальными во всех программных файлах переменные X, Y, Z. Причем эти переменные сохраняют свои значения при выходе из файла. По умолчанию же каждая переменная действует в пределах одного файла.

**KEYBOARD** - временно приостанавливает выполнение программы и передает управление пользователю. При этом режиме в начале строки на экране появляется перед мерцающим курсором двойной (а не одинарный, как обычно) символ "пп" и пользователь может выполнять с клавиатуры все функции MATLAB. Этот режим удобен для ввода данных с клавиатуры, а также при отладке. Для возврата управления прерванной программе надо нажать **Ctrl\_Z** .

**PAUSE** - приостанавливает выполнение программы до нажатия пользователем любой клавиши; **pause(n)** - прерывает выполнение программы на n секунд.

**RETURN** - возвращает управление вызывающей программе или интерпретатору.

Следующие семь функций обеспечивают работу с областью переменных.

**CLEAR** - удаляет все переменные из отведенной для хранения их значений области оперативной памяти; **clear X Y** удаляет только переменные X и Y; **clear function** - удаляет все скомпилированные библиотечные функции (MATLAB автоматически компилирует и хранит те функции, которые многократно используются в программе).

**EXIST('имя')** - возвращает 1, если имя переменной найдено в области переменных; возвращает 2, если на диске найден файл *имя*.ш, и возвращает 0, если не найдено ни то, ни другое.

**SAVE имя** - записывает на диск в файл с именем *имя.mat* имена и значения всех переменных из области переменных; **save имя X Y** - сохраняет на диске только переменные X и Y; **save** - сохраняет все переменные в файле с именем *matlab.mat*.

**LOAD имя** - восстанавливает с диска, из файла с именем *имя.mat* сохраненные переменные и их значения; **load** - восстанавливает область переменных из файла с именем *matlab.mat*.

**PACK** - удаляет все лишнее из области переменных.

**WHO** - выдает на экран содержимое области переменных (имена). Функция **whos** кроме имен выдает и размеры памяти, занимаемой каждой переменной.

В заключение раздела опишем три библиотечные функции, полезные для управления ходом вычислений.

**ETIME(T1,T0)** - возвращает время в секундах, прошедшее между двумя векторами (отсчетами) времени T0 и T1 (см. **clock**).

**FEVAL(F,x1,x2,...,x10)** - вычисляет значения функции. F - имя (строка символов) функции, обычно библиотечной функции пользователя, которая может использовать до 10 аргументов. Например, для файла *funk.ш* строка *F='funk'*, тогда **feval(F,9.64)** - это тоже самое, что **funk(9.64)**. Описанная функция удобна внутри функций, использующих имена других функций в качестве своих аргументов.

**MENU(S0,S1,...,S15)** - генерирует на экране меню с заданными пользователем вариантами (не более 15), возвращает номер выбранного пользователем варианта. Например, *S0 = 'Выбери цвет'* - строка заголовка меню, *S1 = 'красный'*, *S2 = 'синий'*, *S3 = 'зеленый'* и т.д. - варианты.

### 2.2.2. Ввод/вывод и работа с файлами

**CLC** - очистка текстового экрана.

**DISP(X)** - выводит на экран элементы матрицы X, но имя матрицы не печатает. Это основная функция пакета для вывода текста, для чего формируется текстовый вектор X.

**ECHO** - см. п. 2.2.1.

**ERROR** - см. п. 2.2.1.

**FORMAT** - устанавливает формат вывода числовых данных. Все вычисления в пакете выполняются с двойной точностью, а формат вывода чисел можно задать следующими способами:

**format short** (или просто **format**) - основной ( по умолчанию) - 5 значащих цифр, с фиксированной точкой;

**format long** - 15 значащих цифр, с фиксированной точкой;

`format short e` - 5 значащих цифр, с плавающей точкой;

`format long e` - 15 значащих цифр, с плавающей точкой;

`format hex` - шестнадцатеричные числа;

`format +` - компактный формат: вместо положительного числа выводится "+" (плюс), вместо отрицательного числа или нуля - "-" (минус). Мнимая часть комплексного числа игнорируется;

`format bank` - коммерческий формат с фиксированной точкой, 2 десятичных знака в дробной части (для копеек или центов);

`format compact` - компактная печать без пропуска пустых строк;

`format loose` - добавление пустых строк (по умолчанию).

`INPUT` - ввод данных с клавиатуры:

`input ('подсказка')` - печатает подсказку и ждет ввода числа (чисел) с клавиатуры, возвращает введенное число (вектор), то есть можно вводить векторы, элементы которых разделены пробелами или запятыми;

`input ('подсказка', 's')` - для ввода текстовой информации.

`PRN` - выводит на принтер содержимое графического экрана; эквивалентно в MS-DOS одновременному нажатию клавиш Shift и PrtSC.

Функции работы с файлами:

`CHDIR имя_директория` - меняет текущий директорию (аналогично MS-DOS).

`DELETE` - удаление файла с диска (аналогично MS-DOS).

`DIARY имя` - открывает на диске файл (протокол сеанса работы) с заданным именем и направляет в этот файл всю информацию, выводимую на экран (как ввод, так и вывод);

`diary` - последовательное выполнение этой функции (без имени файла) поочередно приостанавливает и возобновляет вывод данных в открытый ранее файл;

`diary off` - прекращает вывод и закрывает файл.

`DIR` - выводит список файлов директория (аналогично MS-DOS).

`EXIST` - см. п. 2.2.1. `LOAD` - см. п. 2.2.1. `META` - см. п. 2.2.4. `SAVE` - см. п.

## 2.2.1.

`TYPE имя_файла.расширение` - выводит на экран текст файла;

`type имя_файла` - выводит на экран текст файла `имя_файла.ш`.

`WHAT` - выдает список файлов с расширением `.ш` для текущего, а также для каждого из директориев, заданных параметром `MATLABPATH` (см. п. 1.2).

## 2.2.3. Преобразование данных

Целые и дробные числа:

`CEIL(X)` - округляет `X` до ближайшего большего целого.

`FIX(X)` - округляет `X` до целого в сторону нуля (положительные - до меньшего, отрицательные - до большего).

**FLOOR(X)** - округляет X до ближайшего меньшего целого.

**REM(X,Y)** - возвращает остаток от деления X на Y:

$$\text{rem}(X,Y)=X-Y*N, \text{ где } N= \text{fix}(X/Y)$$

**ROUND(X)** - округляет X до ближайшего целого.

**Комплексные и вещественные числа:**

В пакете MATLAB элементы матриц могут иметь комплексные значения. В качестве символа мнимой единицы используется **SQRT(-1)**. Вводить мнимые и комплексные числа можно различными способами, например:

$$A = S*\text{sqrt}(-1); \quad B = 2+3*\text{sqrt}(-1);$$

$$j = \text{sqrt}(-1); \quad A = S*j; \quad B = 2 + 3*j;$$

**ABS(X)** - возвращает абсолютную величину X, если X - вещественное; если X - комплексное, то **abs(X)** - модуль комплексного числа.

**CONJ(X)** - возвращает комплексно-сопряженное число.

**IMAG(X)** - мнимая часть числа X.

**REAL(X)** - вещественная часть числа X.

**ANGLE(X)** - библиотечная функция - возвращает фазу комплексного числа в радианах.

**Текстовые данные:**

Строка символов  $X = \text{'строка символов'}$  - это вектор, элементами которого являются ASCII-коды символов.

**ABS(x)** - превращает такой вектор в его десятичное ASCII представление, при этом элементы вектора - целые числа в диапазоне от 0 до 255.

**SPRINTF('формат',X)** - преобразует скалярную (не векторную и не матричную) числовую величину X в строку символов в соответствии с форматом (более подробную информацию о форматах см. Язык C, 'stdio', 'printf', см. также текст библиотечных функций **int2str** и **num2str**).

**FPRINTF('имя\_файла','формат',X)** - делает то же самое, что и **sprintf**, но добавляет преобразованную строку символов в указанный файл.

**INT2STR(N)** - преобразует целое число (скаляр) N в строчное представление (вектор ASCII-кодов). Эквивалентно **sprintf('%f',N)**.

**NUM2STR(X)** - преобразует число (скаляр) X в строчное представление с 4 цифрами дробной части (и, если необходимо, с плавающей точкой). Эквивалентно **sprintf('%4g',X)**.

Две последние библиотечные функции используют обращение к встроенной функции **sprintf**. Изменив в тексте функций спецификатор *формата*, можно получить строчное представление чисел в любом нужном формате.

Прочие функции, полезные при преобразовании данных:

**L=SIZE(X)** - возвращает двухэлементный вектор  $L=[M,N]$ -размер матрицы X; другое обращение:  $[M,N] = \text{size}(X)$ .

**L=LENGTH(X)** - библиотечная функция - возвращает длину вектора **x**, что эквивалентно **l=max(size(X))**.

<b>ONES</b> - генерирует матрицу единиц; <b>ZEROS</b> - генерирует матрицу нулей;		
<b>ones(N)</b>	<b>zeros(N)</b>	- размера <b>N*N</b> ;
<b>ones(M,N)</b>	<b>zeros(M,N)</b>	- размера <b>M*N</b> ;
<b>ones(A)</b>	<b>zeros(A)</b>	- размера матрицы <b>A</b> .

#### 2.2.4. Графика

**AXIS(V)** - "ручная" установка масштабов по осям **x** и **y**; обычно масштаб для каждого графика выбирается автоматически (по умолчанию).

**V=[x\_max,x\_min,y\_max,y\_min]** - вектор, задающий масштабы по осям. Функция возвращает вектор из четырех элементов с масштабами предыдущего построения графика: **Vold=axis(Vnew)**, где **Vold** - старый, **Vnew** - новый масштаб;

**axis** - отменяет автоматическое масштабирование, при этом действителен масштаб **Vold**. Повторное выполнение **axis** - вновь включает автоматическое масштабирование;

**axis('square')** - делает график квадратным;

**axis('normal')** - возврат к автоматической настройке формы и размера графиков в зависимости от их числа на экране и стандарта графического интерфейса компьютера.

**CLG** - очистка графического экрана.

**CONTOUR(Z)** - выводит на экран в виде линий уровня поверхность, заданную поточечно элементами матрицы **Z**, значения которых рассматриваются как высота над плоскостью;

**contour(Z,N)** - использует при этом ровно **N** линий уровня;

**contour(Z,V)** - использует при этом **length(V)** линий уровня, которые соответствуют уровням, заданным элементами вектора **V**;

**contour(Z,N,X,Y)** - векторы **X** и **Y** задают ту часть координатной плоскости, поверхность над которой будет выведена на экран.

**GRID** - накладывает на график координатную сетку.

**HOLD** - "задерживает" уже выведенные на графический экран кривые; в этом режиме масштаб сохраняется прежний, а последующие кривые будут накладываться на предыдущие. Повторное **hold** отключает этот режим. Возможная форма: **hold on** - включает; **hold off** - выключает.

LOGLOG - выводит график с логарифмической (с основанием 10) шкалой для осей  $x$  и  $y$ . См. plot.

MESH(Z) - выводит на экран трехмерную "сетчатую" поверхность, заданную поточечно элементами матрицы Z, значения которых рассматриваются как высота над плоскостью;

mesh(Z,M) - позволяет при этом задать "точку обзора" поверхности;  $M=[AZ EL]$  - двухэлементный вектор; AZ (азимут) - угол поворота вокруг вертикальной оси в градусах (положительный для поворота поверхности по часовой стрелке); EL (высота) - угол поворота вокруг горизонтальной оси в градусах (положительный, если точка обзора выше поверхности);

mesh(Z,S), а также mesh(Z,M,S) - используют трехэлементный вектор  $S=[sx sy sz]$ , который задает размеры фрагмента поверхности по каждой из трех осей  $x,y,z$  соответственно.

Примеры:

$M = [-37.5 \ 30]$  - используется по умолчанию;

$M = [0 \ 0]$  - "точка обзора" лежит на уровне первого столбца, со стороны элемента  $Z(m,1)$ ;

$AZ = 180$  - взгляд на матрицу "сзади";

$EL = 90$  - взгляд на матрицу "сверху".

Для облегчения визуализации функции двух переменных следует использовать библиотечную функцию meshdom.

META имя - открывает на диске графический метафайл с именем *имя.mesh* и выводит в него текущее содержимое графического экрана. В этот же файл можно направить и другие графики, которые будут позже построены на графическом экране. Для этого надо выполнить функцию meta без имени файла. Для вывода графиков на принтер в нужном пользователю формате графический метафайл должен быть обработан графическим постпроцессором GPP.exe.

PLOT(X,Y) - выводит график зависимости  $Y = f(X)$ . Векторы X и Y задают поточечно абсциссы и ординаты точек. Масштабы по осям  $x$  и  $y$  выбираются автоматически в зависимости от содержимого векторов X и Y. Если Y - матрица, то выводится несколько кривых на одном графике;

plot(X1,Y1, ..., XN,YN) - выводится несколько кривых, причем тип линий выбирается автоматически;

plot(X1,Y1,'код1', X2,Y2,'код2') - выводятся две кривые, тип линий для которых задан пользователем (см. таблицу);

plot(Y) - использует в качестве абсцисс индексы элементов вектора (матрицы); если элементы Y - комплексные числа, то plot(Y) эквивалентно  $plot(\text{real}(Y), \text{imag}(Y))$ , во всех остальных случаях мнимая часть игнорируется.

Типы линий и их коды (функция plot):

непрерывные:	поточечные:	цветные:
сплошная -	точки .	красная r
пунктирная --	плюсы +	зеленая g
из точек :	звездочки *	синяя b
штрих-пунктир -.	кружочки o	белая w
	крестики x	невидимая i

**POLAR(THETA,RO)** - выводит график зависимости  $RO = f(THETA)$  в полярных координатах, где THETA - угол (в радианах) поворота радиус-вектора, RO - его модуль (длина). Для вывода нескольких кривых линиями разного типа см. plot.

**SEMYLOGX** - выводит график с логарифмической (с основанием 10) шкалой для оси x и линейной шкалой для оси y. См. plot.

**SEMYLOGY** - выводит график с логарифмической (с основанием 10) шкалой для оси y и линейной шкалой для оси x. См. plot.

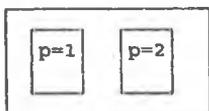
**SHG** - делает видимым на экране монитора графический экран; обратный переход к текстовому экрану - по нажатию любой клавиши.

**SUBPLOT(mnp)** - (выполняется перед функцией plot) задает разбиение графического экрана для вывода нескольких графиков меньшего размера, где mnp - константа из 3 цифр. Экран разбивается на  $m \cdot n$  окон, вывод графика идет в окно с номером p (развертка по строкам). После выполнения функций:

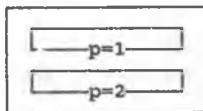
subplot(mnp), plot(X,Y)

последующие plot(X,Y) будут выводить графики в окна с номерами p+1, p+2 и т.д. Возврат к полному экрану возможен одним из следующих способов: clg, subplot(111) или subplot.

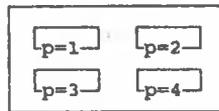
Примеры:



subplot(12p)  
1 <= p <= 2



subplot(21p)  
1 <= p <= 2



subplot(22p)  
1 <= p <= 4

**TITLE('текст')** - выводит текст (заголовок) в верхней части графического экрана.

**XLABEL('текст')** - выводит текст под осью x.

**YLABEL('текст')** - выводит текст слева от оси y.

### Библиотечные функции, связанные с графикой:

**BAR(X)** - выводит на графический экран "столбиковую" диаграмму (элементы вектора X - высоты "столбиков").

**HISTOGRAM(X,n)** - вычисляет и строит гистограмму распределения элементов вектора X, n - число интервалов. По умолчанию, в **histogram(X)** -  $n=10$ .

**[XX,YY] = MESHDOM(X,Y)** - генерирует матрицы XX и YY для области плоскости (x,y), задаваемой векторами X и Y, что дает пользователю возможность генерировать затем матрицу значений функции двух переменных и визуализировать ее в виде поверхности с помощью функции **mesh**. Пример:

```
x=-2.0:1.0:2.0; y=-1.0:1.0:1.0;
[xx,yy]=meshdom(x,y);
z=xx.*exp(-xx.^2-yy.^2); mesh(z);
```

в этом примере:  $x=[-2 \ -1 \ 0 \ 1 \ 2]$ ,  $y=[-1 \ 0 \ 1]$ ,

```
xx=          yy=
-2  -1  0  1  2          -1  -1  -1  -1  -1
-2  -1  0  1  2          0  0  0  0  0
-2  -1  0  1  2          1  1  1  1  1
```

### 2.3. Математические функции

Аргументом (X) математической функции может быть вектор или матрица (в том числе размерности  $1 \times 1$ ). Значения функции вычисляются от каждого элемента аргумента. Возвращаемый результат имеет ту же размерность, что и аргумент. Звездочкой (\*) помечены библиотечные функции.

- exp(X) - экспонента;
- sqrt(X) - квадратный корень;
- sign(X) - сигнум - знак числа;
- \* gamma(X) - гамма-функция;
- \* **bessel(ALFA,X)** - функция Бесселя первого рода.

*Тригонометрические:*

*Гиперболические:*

*Логарифмические:*

sin(X)	asin(X)	* sinh(X)	* asinh(X)	log(X)
cos(X)	acos(X)	* cosh(X)	* acosh(X)	* log10(X)
tan(X)	atan(X)	* tanh(X)	* atanh(X)	
	atan2(Y,X)			

## 2.4. Логические и реляционные функции

Помимо стандартных (см. п. 2.1) реализован ряд встроенных и библиотечных функций:

**ALL(X)** - возвращает строку, элементы которой равны 1, если все элементы соответствующего столбца не равны 0, иначе - 0.

**ANY(X)** - возвращает строку, элементы которой равны 1, если любой из элементов соответствующего столбца не равен 0, иначе - 0.

**MAX(X)** - возвращает строку максимумов для каждого из столбцов.  $[Y, I] = \max(X)$  - строка I содержит индексы максимумов. **max(X, Y)** - возвращает матрицу того же размера, что X и Y, элементами которой являются наибольшие из двух соответствующих элементов X и Y. Если числа комплексные, то сравниваются их модули.

**MIN(X)** - отыскание минимумов (аналогично max).

**FIND(X)** - возвращает строку индексов элементов вектора X, не равных 0. **find(X>100)** - строка индексов элементов, больших 100.

**SORT(X)** - сортирует каждый столбец матрицы X в возрастающем порядке.  $[Y, I] = \text{sort}(X)$  - матрица I содержит индексы сортировки.

**Y=ISEMPTY(X)** - (библиотечная функция) возвращает единицу, если X - пустая матрица, имеющая размер 0\*0, в противном случае - 0.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Матричные вычисления на персональной ЭВМ / Сост. С.В. Суханов; Самар. гос. аэрокосм. ун-т. Самара, 1994. 28 с.
2. Документация к пакету MATLAB / Пер. с англ.

## УКАЗАТЕЛЬ ФУНКЦИЙ

ABS	14	AXIS	15	CONTOUR	15
acos	18	BAR	17	cos	18
acosh	18	bessel	18	cosh	18
ALL	19	BREAK	10	DELETE	13
ANGLE	14	CASESEN	10	demo	5, 6
ANS	3	CEIL	13	DIARY	13
ANY	19	CHDIR	13	DIR	13
asin	18	CLC	12	DISP	12
asinh	18	CLEAR	11	ECHO	11, 12
atan	8, 18	CLG	15, 17	edit	7
atan2	18	CLOCK	10	else	9
atanh	18	CONJ	14	elseif	9

end	9, 10	IEMPTY	19	RETURN	11
ERROR	11, 12	KEYBOARD	11	ROUND	14
ETIME	12	LENGTH	15	save	5, 12, 13
EVAL	10, 11	load	4, 5, 12, 13	SHG	17
EXIST	12, 13	log	8, 18	sign	18
exit	5	log10	18	sin	18
exp	18	LOGLOG	16	sinh	18
FEVAL	12	max	15, 19	SIZE	15
FIND	19	MENU	12	SORT	19
FIX	13, 14	MESH	16, 18	SPRINTF	14
FLOOR	14	meshdom	16, 18	sqrt	8, 11, 14, 18
FLOPS	11	META	13, 16	startup	5
FOR	8, 10	MIN	19	SUBPLOT	17
FORMAT	12, 13	NARGIN	8	SUM	10
FPRINTF	14	num2str	14	tan	18
function	6, 11	ONES	15	tanh	18
gamma	18	PACK	12	TITLE	17
global	6, 11	PAUSE	11	TYPE	13
GPP	4, 16	pi	8	WHAT	13
GRID	15	plot	16, 17	while	10
help	5, 6	POLAR	17	WHO	12
HOLD	15	PROD	10	XLABEL	17
if	9, 10	PRTSC	13	YLABEL	17
IMAG	14, 16	quit	5	ZEROS	15
INPUT	13	REAL	14, 16		
int2str	14	REM	14		

## Содержание

1. Общие сведения	3
1.1. Краткая характеристика пакета	3
1.2. Структура файлов пакета	4
1.3. Начало и окончание работы пакета	4
1.4. Подсказки и демонстрации	5
1.5. Создание командных файлов	6
2. Функции пакета MATLAB	7
2.1. Использование служебных символов и переменных	7
2.2. Функции языка программирования	9
2.2.1. Управление ходом вычислений, областью переменных	9
2.2.2. Ввод/вывод и работа с файлами	12
2.2.3. Преобразование данных	13
2.2.4. Графика	15
2.3. Математические функции	18
2.4. Логические и реляционные функции	19
Библиографический список	19
Указатель функций	19