第 1 章　MATLAB 基础

1.1　MATLAB 简介

MATLAB 是一种高精度的科学计算语言。它将计算和可视化编程结合起来，给用户提供一个易于使用的环境。在这个环境里，用户可以用熟悉的数学符号来表示问题及其解决方法。它的典型运用包括数学与运算、算法开发、建模与仿真、数值分析、工程图形科学等。作为一个交互式系统，它的基本数据单元为数组，数组不需要固定的大小，因此用户可以灵活解决许多数学问题，特别是涉及矩阵和向量运算的问题。MATLAB 指令表达式类似于数学运算、工程应用中常用的格式。相对于 C 和 Fortran 这些高级语言，MATLAB 的语法规则更为简单。

MATLAB 最重要的特性是它提供了很多已有程序组以解决特定应用问题，也就是工具箱，如信号处理工具箱、控制系统工具箱、神经网络工具箱、模糊逻辑工具箱、通信工具箱、数据采集工具箱以及其他许多特定的工具箱。对于大多数用户，为了能够灵活而高效地使用工具箱，他们通常需要学习相关的专业知识。除了内置函数外，所有主要文档和 MATLAB 工具箱文件都是可读且可更改的。这些工具箱实际就是一套复杂函数，工具箱的应用进一步扩展了 MATLAB 软件的功能。为了能够解决特殊问题，用户可以改变源文件并加入自己的文件以建立新的工具箱。

在 MATLAB 软件成功安装后，以 R2011b 版本为例，首次进入显示的主界面如图 1-1 所示。

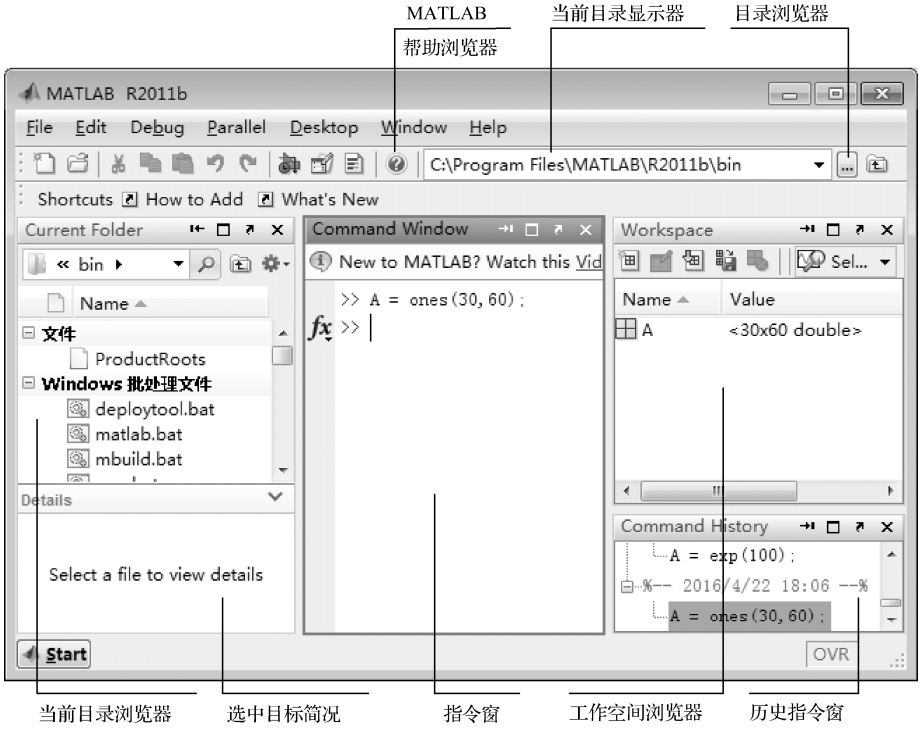


图 1-1　MATLAB 用户界面

（1）指令窗。MATLAB 操作的最主要窗口。在该窗内，可输入各种 MATLAB 指令、函数、表达式；显示除图形外的所有运算结果；运行错误时，显示相关提示。

（2）当前目录浏览器。在该浏览器中，展示着子目录、M 文件、MAT 文件和 MDL 文件等。对该界面上的 M 文件，可以直接进行复制、编辑和运行；界面上的 MAT 数据文件，可以直接送入 MATLAB 工作内存。

此外，在当前目录浏览器正下方，还有一个“选中目标简况”窗。该窗显示所选文件的概况信息。比如该窗会展示：M 文件的 H1 行内容，最基本的函数格式；所包含的内嵌函数和其他子函数。

（3）工作空间浏览器。该窗口罗列出 MATLAB 工作空间中所有的变量名、大小、字节数。在该窗中，可对变量进行观察、编辑、提取和保存。

（4）历史指令窗。该窗口记录已经运行过的指令、函数、表达式，以及它们运行的日期、时间。该窗中的所有指令、文字都允许复制、重运行及用于产生 M 文件。

1.2 MATLAB 基本操作

MATLAB 提供方便实用的功能键用以在当前或之前的输入命令窗口编辑、修改命令行。这些功能键如表 1-1 所示。

表 1-1 常用功能键

功 能 键	功 能	功 能 键	功 能
↑	重新加载之前的命令行	End	将光标移动到行尾
↓	重新加载之后的命令行	Ctrl+Home	将光标移动到命令窗口的上端
←	光标向左移动一个字符	Ctrl+End	将光标移动到命令窗口的下端
→	光标向右移动一个字符	Esc	撤销命令行
Ctrl+←	光标向左移动一个单词	Delete	删除光标处的字符
Ctrl+→	光标向右移动一个单词	Backspace	删除光标左侧的字符
Home	将光标移动到行头		

1.2.1 MATLAB 运算模式

MATLAB 的运算模式有两种：命令行运算模式和 M 文件运算模式。命令行运算模式是直接在命令窗口的提示“>>”后输入命令或运算表达式。按下“Enter”键后，MATLAB 将会进行运算并显示运算结果。这种方式比较适合实现一些简单的功能，比如简单的计算或画图等。M 文件运算模式是一种用 MATLAB 语言写成的计算机程序，它的扩展名为“.m”。在 MATLAB 的 M 文件编辑器中输入、编辑和调试代码，然后在命令窗口中输入所生成的文件名就可以运行它了。

当 M 文件运行时，MATLAB 将会使用它默认的搜索路径去寻找 M 文件。如果你想运行的 M 文件不在搜索路径中，它将不会被执行。我们可以使用 MATLAB 用户主界面中的“File”菜单下的“set path”命令设置，在 MATLAB 搜索路径中加入我们所需要的文件夹和目录。

1．命令行运算模式举例

可以在命令窗口中直接输入以下命令实现三角函数绘图。

```
clear;
x = -pi:0.1:pi;
y1 = sin(x);
y2 = cos(x);
plot(x,y1,x,y2);
title('cosine and sine functions');
xlabel('time');
ylabel('Amplitude');
legend('y = cos(x)','y = sin(x)');
grid on;
```

命令窗口操作运行结果如图 1-2 所示。

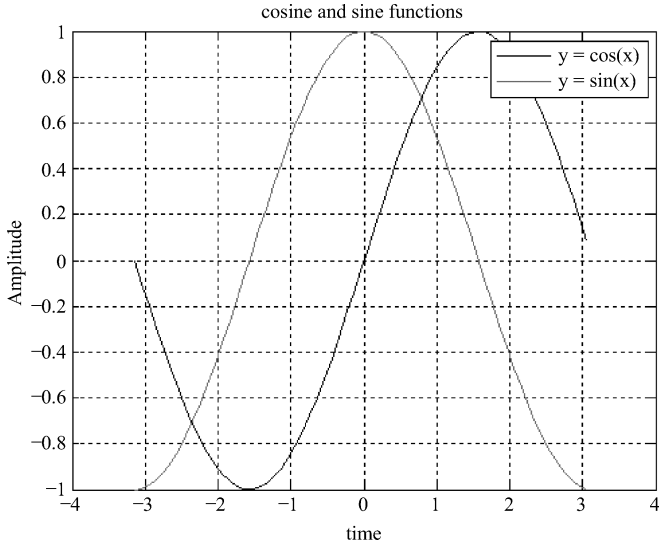


图 1-2　命令窗口操作运行结果

2．M 文件举例

通常 M 文件中会包含一定的注释语句，这些注释语句的出现位置可以放在解释对象的上面、下面或后面，甚至可以放在程序开始的地方，以下为一个 M 文件的例子，其中有多行注释语句。

```
% 这是一个 M 函数示例
function [y,pos] = findmax(a)
% findMax 可以找出矩阵 a 中最大值及其地址
% y = findmax(a):找出矩阵 a 中的最大值
% [y,pos] = findmax(a):找出矩阵 a 中的最大值，pos 为最大值地址
[y,p] = max(a(:));
[r,c] =ind2sub(size(a),p);
```

```
pos = [r,c];
```

1.2.2 数据类型和算术运算

1．数字、变量和表达式

每种编程语言都有自己的数字、变量和表达式使用约束，MATLAB 变量的描述是十进制形式，十进制的小数点和复数符号可以用在科学计数法中，如$1.3e^{-3}$。MATLAB 可支持 16 位有效数字，从$10e^{-308}$～$10e^{308}$。

数字是数学运算的最基本对象，MATLAB 中定义的数字的数据类型为整数、浮点数和复数三种，另外，还定义了 Inf 和 NaN 两个特殊数值。

MATLAB 支持 8 位、16 位、32 位和 64 位的有符号和无符号整数。这 8 种数据类型及其描述如表 1-2 所示。

表 1-2　整数类型

数 据 类 型	描　　述
uint 8	8 位无符号整数，范围为 $0\sim2^8-1$
int8	8 位有符号整数，范围为$-2^7\sim2^8-1$
uint16	16 位无符号整数，范围为 $0\sim2^{16}-1$
int16	16 位有符号整数，范围为$-2^{15}\sim2^{15}-1$
uint32	32 位无符号整数，范围为 $0\sim2^{32}-1$
int32	32 位有符号整数，范围为$-2^{31}\sim2^{31}-1$
uint64	64 位无符号整数，范围为 $0\sim2^{64}-1$
int64	64 位有符号整数，范围为$-2^{63}\sim2^{63}-1$

MATLAB 支持单精度和双精度两种浮点数，其中，双精度浮点数是 MATLAB 的默认数据类型。这两种数据类型及其描述如表 1-3 所示。

表 1-3　浮点数类型

数 据 类 型	描　　述
single	单精度浮点数，范围为$-3.40282\times10^{38}\sim3.40282\times10^{38}$
double	双精度浮点数，范围为$-1.79769\times10^{308}\sim1.79769\times10^{308}$

复数包含实部和虚部，在 MATLAB 中，用 i 或者 j 来表示虚部。

MATLAB 中，Inf 和-Inf 分别表示正无穷大和负无穷大。除法运算中除数为 0 或者结果溢出都有可能导致 Inf 或-Inf 的运算结果。类似 1/0 的结果是 Inf。用 NaN（Not a Number）表示一个既不是实数也不是复数的数值。类似 0/0、Inf/Inf 的结果都是 NaN。

MATLAB 用户自定义变量命名规则：MATLAB 的变量要求区分大小写，首字符要求是英文字母，长度不得超过 31 个字符，可包含英文字母、数字和下画线。但变量名称中不能包含空格或标点符号。

除了用户自定义变量以外，MATLAB 中有一些预定义变量，这些预定义变量具有相应的

初始值，其中比较常用的如表 1-4 所示。

表 1-4 MATLAB 默认预定义变量

预定义变量	含 义	预定义变量	含 义
NaN 或 nan	非数值	ans	最近结果
nargin	函数输入变量个数	Inf 或 inf	无穷大
nargout	函数输出变量个数	i 或 j	虚数单位
realmax	浮点数类型所能表示的正的最大值	pi	圆周率
realmin	浮点数类型所能表示的正的最小值	eps	浮点数精度值

表达式是由数字、算符、数字分组符号（括号）、用户变量和系统变量等组合所得到的。下面给出一个简单的表达式及其运行结果。

```
cosd(45)+tand(30)+sqrt(2)+1
ans =
    3.6987
```

2．算术运算符和表达式

MATLAB 中的算术运算符如表 1-5 所示。

表 1-5 MATLAB 中的算术运算符

预定义符号	含 义	预定义符号	含 义
+	加号	.^	按元素乘方
-	减号	*	矩阵相乘
.*	按元素相乘	/	解决线性等式 xA = B 中的 x
./	数组右除	\	解决线性等式 Ax = B 中的 x
.\	数组左除	^	矩阵乘方

MATLAB 中的表达式由变量名、运算符号和函数名组成。括号具有最高优先级。算术运算的优先级顺序为：乘方>相乘/相除>加法/减法。赋值运算符“=”和其他运算符的旁边可以加入空格，以提高程序的可读性。

算数运算符应用示例如下。

以下运算均针对于 ***A***=[1,2,3;5,6,7;3,4,9]，***B***=[2,3,5;5,2,7;4,6,9]进行描述。

（1）计算线性等式 $x\boldsymbol{A} = \boldsymbol{B}$ 中的 x 的值。

命令：

```
A=[1,2,3;5,6,7;3,4,9];
B=[2,3,5;5,2,7;4,6,9];
x=B/A
```

结果：

```
x =
```

```
    0.6250    0.1250    0.2500
   -6.0000    1.0000    2.0000
    1.3750    0.3750    0.2500
```

（2）计算 3./***A***。

命令：

```
A=[1,2,3;5,6,7;3,4,9];
C=3./A
```

结果：

```
C =
    3.0000    1.5000    1.0000
    0.6000    0.5000    0.4286
    1.0000    0.7500    0.3333
```

1.2.3 关系和逻辑运算符

MATLAB 与其他的计算机语言一样，不仅有算数运算符，还有关系运算符和逻辑运算符。关系运算符和逻辑运算符分别如表 1-6 和表 1-7 所示。

表 1-6 关系运算符

运 算 符	描 述	运 算 符	描 述
==	判断是否相等	<=	判断是否小于或等于
>=	判断是否大于或等于	<	判断是否小于
>	判断是否大于	~=	判断是否不相等
isequal	判断数组容量是否相等	isequalwithequalnans	判断数组容量是否相等，NaN 也可以作为判断量

表 1-7 逻辑运算符

运算符	描述	运算符	描述
&	“与”运算	any	判断是否任何数组元素非零
~	“非”运算	false	逻辑“0”（“假”）
|	“或”运算	find	寻找非零元素的位置
xor	“异或”运算	islogical	判断输入是否为逻辑数组
all	判断是否所有数组元素非零或真	logical	将数字值转为逻辑值

MATLAB 规定：在所有的关系和逻辑表达式中，所有的非零数值都表示“真”，只有 0 表示“假”。

关系运算和逻辑运算的结果为由“0”或“1”组成的逻辑数组。逻辑数组是一种具有特殊数值的数组，它会涉及数值运算和函数调用，同是它也可以表示逻辑结果。

关系运算符应用示例如下。

（1）在命令窗口输入“A=1∶9”，输入以下的命令可以得到对应关系运算的结果。

```
B=10-A,r0=(A<B),r1=(A==B)
```

结果：

```
A=1 2 3 4 5 6 7 8 9
B=9 8 7 6 5 4 3 2 1
r0=1 1 1 1 0 0 0 0 0
r1=0 0 0 0 1 0 0 0 0
```

（2）显示矩阵 ***A***=[−1,2,4; −2,5,3;9, −8,3]中值大于 3 的元素。

命令：

```
A=[-1,2,4;-2,5,3;9,-8,3];
A>3*ones(3)
```

结果：

```
0     0     1
0     1     0
1     0     0
```

若一个表达式包括运算变量、算数运算符、关系运算符和逻辑运算符等，则表达式的计算需要遵循一套优先级顺序，MATLAB 首先计算优先级高的运算，再计算优先级低的运算；若优先级相同，则按照从左到右的顺序依次计算。表 1-8 所示为按照优先级从高到低对运算符进行排序。

表 1-8 运算符的优先等级

运算符
圆括号（）
转置（.'），共轭转置（'），乘方（.^），矩阵乘方（^）
标量加法（+），减法（-），取反（～）
乘法（.*），矩阵乘法（*），右除（./），左除（.\），矩阵右除（./），矩阵左除（.\）
加法（+），减法（-），逻辑非（～）
冒号运算符（：）
小于（<），大于（>），小于或等于（<=），大于或等于（>=），等于（==），不等于（～=）
数组逻辑与（&）
数组逻辑非（|）
逻辑与（&&）
逻辑非（||）

MATLAB 支持两种不同的数值运算方式——数组和矩阵运算。其中数组运算指数组对应元素之间的运算，也称为点运算，MATLAB 支持多维数组。数组运算需要输入的数组有相同的大小，否则无法进行运算。而矩阵运算遵循线性变换，并不是简单的多维数组的运算，输入矩阵的维数之间的关系则取决于具体的运算操作，如右除需要两个矩阵有相同的列数，而矩阵相乘则需要前者的列数与后者的行数相等。英文句号“.”可用来区分数组运算和矩阵运

算，两种运算中的加减都是相同的，因此不需要将它们写成“.+”和“.-”。若数组或矩阵运算的操作数中有一个为标量而另外一个不是，那么 MALTAB 将会对其进行标量扩展，使其变为普通的数组或矩阵运算。

表 1-9 给出了数组和矩阵运算的一些区别，本质上的区别是对元素的操作还是对数据整体的操作，数组的概念在其他的编程语言中也有定义，数组与矩阵本质上具有一致性，都是对数据存储和处理的一种方式，这里的数组运算可以看作这个专有的名词，而不要理解为是一种针对数组的运算。

表 1-9 数组和矩阵运算符之间的区别（A、B 为非标量）

数 组 运 算		矩 阵 运 算	
运 算 符	描 述	运 算 符	描 述
.*	A.*B 表示 A 与 B 对应元素相乘	*	A*B 表示 A 与 B 按线性代数法则相乘
.^	A.^B 表示 A 中元素按 B 中对应元素次乘方	^	A^B，若 B 为标量，则为 A 中元素的 B 次乘方；若 B 为非标量，则运算涉及特征值和特征向量
./	A./B 表示 A 中元素除以 B 中对应元素	/	A/B 的结果为 xA=B 的解，A 与 B 必须有相同的列数
.\	A.\B 表示 B 中元素除以 A 中对应元素	\	A\B 表示结果为 Ax=B 的解，A 与 B 必须有相同的行数
.'	A.'表示 A 的转置	'	A'表示 A 的共轭转置

以下为数组与矩阵运算举例，其中的“命令”表示在命令窗口直接输入。这里需要注意，在输入矩阵时，由于矩阵具有多行多列，要注意元素之间的区分，同行元素间用逗号或空格隔开，不同的行由分号“;”或用回车键隔开。命令输入完毕，回车表示代码执行命令。

（1）命令：

```
a=[1,2,3]+[3 2 1]*i                          %%%  输入行向量
```

结果：

```
a=1.0000+3.0000i    2.0000+2.0000i    3.0000+1.0000i
```

（2）命令：

```
b=a'                                         %%%  对行向量 a 共轭转置
```

结果：

```
b=
        1.0000-3.0000i
        2.0000-2.0000i
        3.0000-1.0000i
```

（3）命令：

```
c=b*a                                        %%%  矩阵 a 与 b 相乘
```

结果：

```
c=
        10.0000              8.0000-4.0000i      6.0000-8.0000i
        8.0000+4.00000i      8.0000              8.0000-4.0000i
        6.0000+8.0000i       8.0000+4.0000i      10.0000
```

（4）命令：

```
d=a.'                                          %%%  转置
```

结果：

```
d=
         1.0000+3.0000i
         2.0000+2.0000i
         3.0000+1.0000i
```

（5）命令：

```
e=c*b                                          %%%  矩阵相乘
```

结果：

```
e=
        28.0000-84.0000i
        56.0000-56.0000i
        84.0000-28.0000i
```

（6）命令：

```
f=exp(c)                                       %%%  元素指数运算
```

结果：

```
f=
        1.0e+004*
        2.2026               -0.1948+0.2256i          -0.0059-0.0399i
        -0.1948-0.2256i       0.2981                  -0.1948+0.2256i
        -0.0059+0.0399i      -0.1948-0.2256i           2.2026
```

（7）命令：

```
g=expm(c)                                      %%%  矩阵指数运算
```

结果：

```
g=
        1.0e+011*
        5.1652+0.0000i       4.1322-2.0661i       3.0991-4.1322i
        4.1322+2.0661i       4.1322+0.0000i       4.1322-2.0661i
        3.0991+4.1322i       4.1322+2.0661i       5.1652+0.0000i
```

（8）对于如下矩阵：

$$\boldsymbol{A}=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix}$$

计算 A*B，A.*B，A.^B，并比较结果。

输入命令参考：

```
A = [1,2;3,4];
B = [1,3;2,4];
A*B                                  %%%  矩阵乘运算
A.*B                                 %%%  矩阵对应元素乘
A.^B                                 %%%  对应元素做乘方运算
```

结果：

```
ans =
     5    11
    11    25
ans =
     1     6
     6    16
ans =
     1     8
     9   256
```

注意，在每个表达式后加上分号“;”，那么结果不会显示出来。这可以加速程序运行。删除表达式后的分号，结果将会立刻在命令窗口中显示，便于查看运行结果。

1.2.4 数组及其操作

一维数组创建和读取示例：

```
x=rand(1,5)                          %%%  产生均匀分布的随机数
x([1 2 5])                           %%%  读取第 1、2、5 个元素
x=find(x>0.5)                        %%%  找出所有元素中大于 0.5 的元素
```

运行结果：

```
x =
    0.8147    0.9058    0.1270    0.9134    0.6324
ans =


    0.8147    0.9058    0.6324
x =
     1     2     4     5
```

二维数组创建示例：

```
A=[1 3;2 4]
```

运行结果：

```
A =
```

1 3
2 4

二维数组访问和赋值如表 1-10 所示。

表 1-10 二维数组访问和赋值

数组访问和赋值	含 义
A(i,j)	i 行 j 列元素
A(:,j)	A 中第 j 列
A(i,:)	A 中第 i 行
A(:,:)	等价于二维数组，如果是矩阵则与 A 相同
A(s)	单下标访问
A(i,j) = sa	将 sa 赋值给 A(i,j)
A(:) = D(:)	将 D 中所有元素赋值给 A
A(s) = sa	将 sa 赋值给 A(s)

1.2.5 保存结果

在程序运行后，我们通常可以得到一幅图像或一些有用的数据作为最终结果。例如，已经得到如图 1-3 所示的结果。

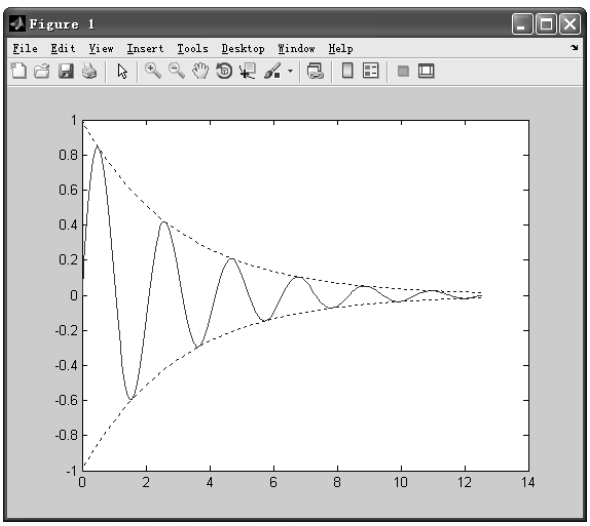


图 1-3 实验结果示例

可以执行菜单栏中的“Edit”→“Copy Figure”命令复制图片，也可以执行菜单栏中的“File”→“Save”命令保存图片，如图 1-4 和图 1-5 所示。

如果想要保存一些关键数据，可以在工作区找到变量，右键单击找到“Save As”命令。例如，要保存矩阵 A 的数据，可以按图 1-6 所示单击“Save As”命令。

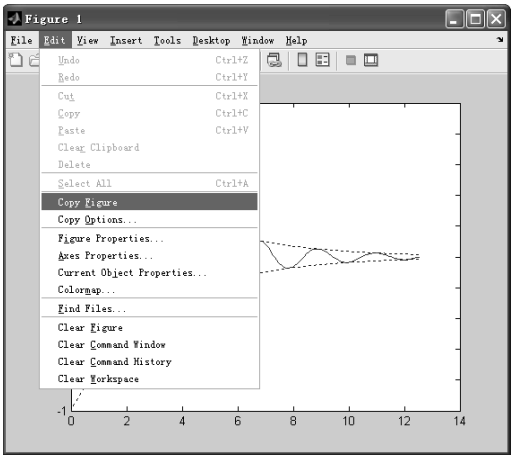


图 1-4　复制图片选项

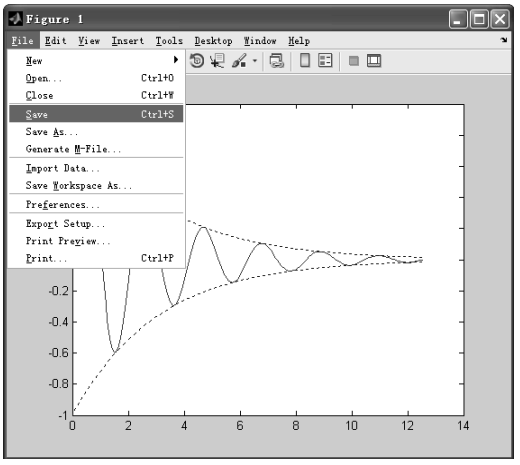


图 1-5　保存图片选项

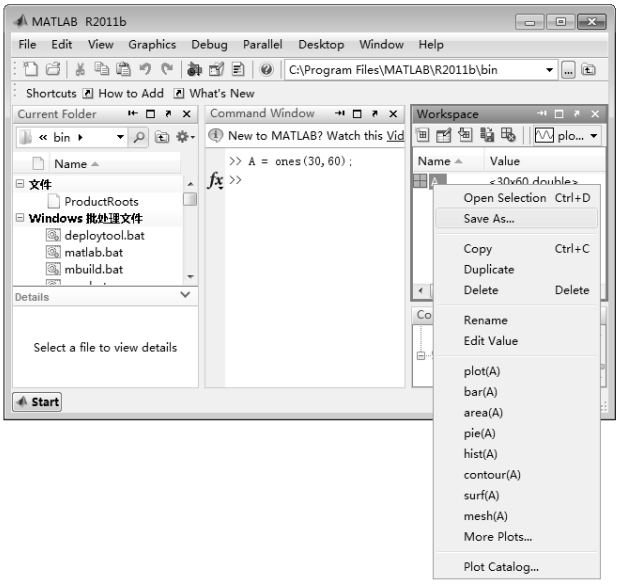


图 1-6　用“Save As”命令保存数据示例

1.2.6　使用“ Help ”选项

当不清楚某个函数或命令的使用方法时，可以使用 MATLAB 中的“Help”选项得到帮助。这里需要说明的是，不同版本 MATLAB 的帮助菜单展现形式不同，但都可以从软件界面打开帮助功能，如图 1-7 所示。

在应用中，打开“Help”选项后，会出现一个“Product Help”窗口（不同的版本会有差异）。可以搜索任何不理解的命令或函数。例如，输入名为“size”的函数就可以得到有关该函数的相关指导，如图 1-8 所示，可以看到帮助界面给出了该函数的使用方法，对同一个变量，用 size 查看尺寸时，可以有多种不同的输出形式和输出变量的个数。

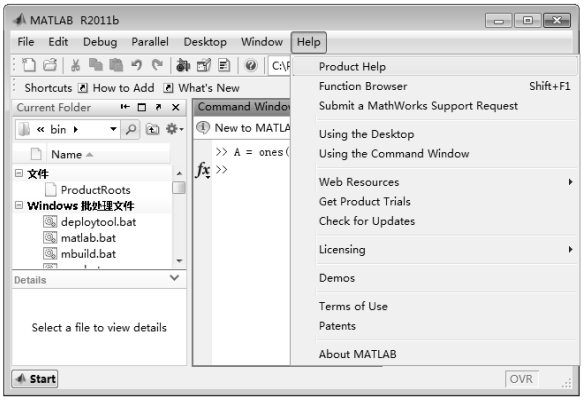
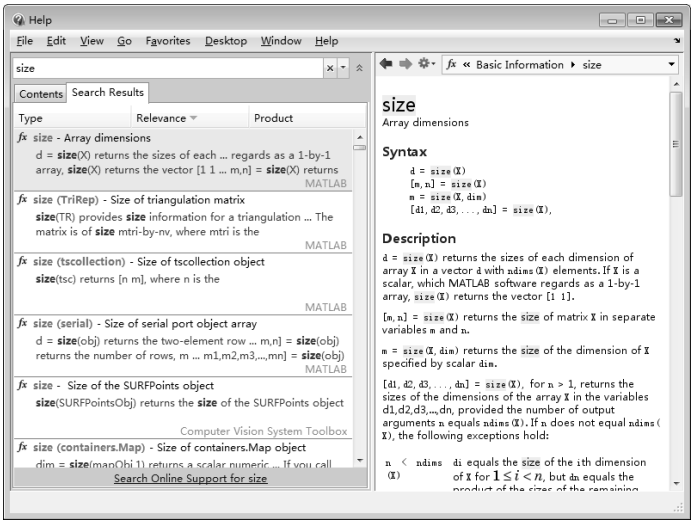


图 1-7 “Help”选项



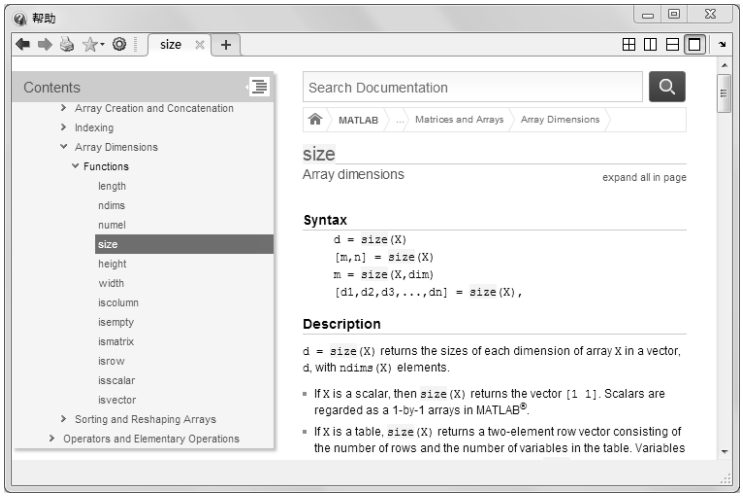


图 1-8 不同版本下“Help”使用界面

1.3 MATLAB 常用函数

在使用 MATLAB 进行仿真时，不可避免地要利用到软件中已经提供的常用函数，这里把一部分常用的函数以表格的形式列出，便于读者查阅。

1.3.1 矩阵计算

常用线性代数函数如表 1-11 所示。

表 1-11 常用线性代数函数

函 数	描 述	函 数	描 述
det	矩阵行列式	Rank	矩阵的秩
diag	生成对角矩阵或得到矩阵的对角线元素	trace	矩阵对角线之和
inv	矩阵的逆	rref	最简行矩阵
triu	提取上三角矩阵	tril	提取下三角矩阵
null	零空间	poly	生成矩阵特征多项式

示例：矩阵 $\boldsymbol{A}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&3&0\\0&0&0&6\end{bmatrix}$，计算$|\boldsymbol{A}|$、$\boldsymbol{A}^{-1}$和矩阵 $\boldsymbol{A}$ 的迹。

```
A = diag([1,1,3,6])
B = det(A)
C = inv(A)
D = trace(A)
```

结果：

```
A =
     1     0     0     0
     0     1     0     0
     0     0     3     0
     0     0     0     6
B =
    18
C =
    1.0000         0         0         0
         0    1.0000         0         0
         0         0    0.3333         0
         0         0         0    0.1667
D =
    11
```

1.3.2 矩阵分解

矩阵的分解主要包括：矩阵的 LU 分解、QR 分解、Cholesky 分解、奇异值分解、特征值分解、Schur 分解及 Jordan 标准型分解等。

1．LU 分解

矩阵的 LU 分解就是将一个矩阵表示为一个下三角矩阵 L 和一个上三角矩阵 U 的乘积形式。线性代数中已经证明，只要方阵 A 是非奇异的（即可逆的），LU 分解总是可以进行的。

当 L 为单位下三角矩阵而 U 为上三角矩阵时，此三角分解称为杜利特（Doolittle）分解。当 L 为下三角矩阵而 U 为单位上三角矩阵时，此三角分解称为克劳特（Crout）分解。显然，如果存在，矩阵的三角分解不是唯一的。

MATLAB 提供的 lu()函数用于对矩阵进行 LU 分解，其调用格式为：

[L,U]=lu(X)：产生一个上三角阵 U 和一个变换形式的下三角阵 L(行交换)，使之满足 X=LU。注意，这里的矩阵 X 必须是方阵。

[L,U,P]=lu(X)：产生一个上三角阵 U 和一个下三角阵 L 以及一个置换矩阵 P，使之满足 PX=LU。当然矩阵 X 同样必须是方阵。

示例：

```
A = [3,-1,1;-1,5,2;1,2,4]
[L,U]=lu(A)
```

结果：

```
A =
     3    -1     1
    -1     5     2
     1     2     4
L =
    1.0000         0         0
   -0.3333    1.0000         0
    0.3333    0.5000    1.0000
U =
    3.0000   -1.0000    1.0000
         0    4.6667    2.3333
         0         0    2.5000
```

2．QR 分解

对矩阵 X 进行 QR 分解，就是把 X 分解为一个正交矩阵 Q 和一个上三角矩阵 R 的乘积形式。QR 分解只能对方阵进行。MATLAB 的函数 qr()可用于对矩阵进行 QR 分解，其调用格式为：

[Q, R]=qr(X)：产生一个正交矩阵 Q 和一个上三角矩阵 R，使之满足 X = QR。

[Q, R, E]=qr(X)：产生一个正交矩阵 Q、一个上三角矩阵 R 及一个置换矩阵 E，使之满足 XE = QR。

示例：

```
A = [3,-1,1;-1,5,2;1,2,4]
[Q,R,E]= qr(A)
```

结果：

```
A =
     3    -1     1
    -1     5     2
     1     2     4
Q =
   -0.1826   -0.4647   -0.8664
    0.9129    0.2472   -0.3249
    0.3651   -0.8503    0.3791
R =
    5.4772    3.1038   -1.0954
         0   -3.3714   -2.4915
         0         0   -1.8954
E =
     0     0     1
     1     0     0
     0     1     0
```

3．Cholesky 分解

如果矩阵 X 是对称正定的，则 Cholesky 分解将矩阵 X 分解成一个下三角矩阵和上三角矩阵的乘积。设上三角矩阵为 R，则下三角矩阵为其转置，即 X = R'R。MATLAB 函数 chol(X) 用于对矩阵 X 进行 Cholesky 分解，其调用格式为：

R = chol(X)：产生一个上三角阵 R，使 R'R = X。若 X 为非对称正定，则提示输出一个出错信息。

[R, p] = chol(X)：这种命令格式不提示出错信息。当 X 为对称正定的，则 p = 0，R 与上述格式得到的结果相同；否则 p 为一个正整数。如果 X 为满秩矩阵，则 R 为一个阶数为 q = p-1 的上三角阵，且满足 R'R = X(1:q, 1:q)。

示例：

```
A = [3,-1,1; -1,5,2; 1,2,4]
B = chol(A)
```

结果：

```
A =
     3    -1     1
    -1     5     2
     1     2     4
B =
    1.7321   -0.5774    0.5774
         0    2.1602    1.0801
```

```
    0         0    1.5811
```

4．奇异值分解

任意一个 m×n 维的矩阵 X 可以分解为 X = USV'，U、V 均为酉矩阵，S 为 m×n 维的对角矩阵，其对角线元素为 X 的从大到小排序的非负奇异值。

```
[U,S,V] = svd(X)
```

示例：

```
A=[1 2 2;2 1 2;2 2 1]
[V,D] = eig(A)
```

结果：

```
A =
     1     2     2
     2     1     2
     2     2     1
V =
    0.6206    0.5306    0.5774
    0.1492   -0.8027    0.5774
   -0.7698    0.2722    0.5774
D =
   -1.0000         0         0
         0   -1.0000         0
         0         0    5.0000
```

5．特征值分解

特征值和特征向量在信号处理中十分重要，它通常用于判断矩阵的秩、决定线性空间的基底、判断系统稳定性以及系统分析中的信号分解。与特征分解相关的一些常用函数如下。

任意一个 n 阶方阵 X 可以分解为 XV = VD，其中 D 为 X 的特征值对角阵，V 为 X 的特征向量矩阵。

[V,D] = eig(X)

[V,D] = eig(X, Y)计算广义特征值矩阵 D 和广义特征值向量矩阵 V，使得 XV = YVD。

示例：

```
A=[1 2 2; 2 1 2; 2 2 1]
[V,D]=eig(A)
```

结果：

```
A=
     1     2     1
     2     1     2
     2     2     1
V=
```

```
        0.6015    0.5522    0.5774
        0.1775   -0.7970    0.5774
       -0.7789    0.2448    0.5774
D=
       -1.0000         0         0
             0   -1.0000         0
             0         0    5.0000
```

以上示例对于输入的矩阵 A，通过 eig()函数计算其特征分解，得到特征向量 V 和特征值 D，这里的特征向量和特征值是一一对应的。

6．Schur 分解

任意一个 n 阶方阵 X 可以分解为 X = URU'，其中 U 为酉矩阵，R 为上三角 schur 矩阵且其主对角线上的元素为 X 的特征值。

```
[U,R] = schur(X)
```

示例：

```
A =[1 2 2; 2 1 2; 2 2 1]
[b, c] = schur(A)
```

结果：

```
A =
     1     2     2
     2     1     2
     2     2     1
b =
    0.6206    0.5306    0.5774
    0.1492   -0.8027    0.5774
   -0.7698    0.2722    0.5774
c =
   -1.0000         0         0
         0   -1.0000         0
         0         0    5.0000
```

7．Jordan 标准型分解

MATLAB 中 Jordan 标准型分解用函数 jordan()来实现，其调用格式为：

```
[V,J] = jordan(A)
```

示例：

```
A=[1 2 2; 2 1 2; 2 2 1];
[V,J] = jordan(A)
```

结果：

```
V =
     1    -1    -1
     1     1     0
     1     0     1
J =
     5     0     0
     0    -1     0
     0     0    -1
```

1.3.3 数学计算函数

在仿真计算中，不可避免地要利用到各种数学表达式，数学表达式可以分解为多个基本的数学运算，一些常用函数的形式如表 1-12 所示。

表 1-12 常用函数的形式

三角函数和双曲线函数			
函 数	描 述	函 数	描 述
sin	正弦（弧度）	asin	反正弦（弧度）
cos	余弦（弧度）	acos	反余弦（弧度）
sinh	双曲正弦（弧度）	asinh	反双曲正弦（弧度）
cosh	双曲余弦（弧度）	acosh	反双曲余弦（弧度）
tan	正切（弧度）	atan	反正弦（弧度）
cot	余切（弧度）	acot	反余切（弧度）
sec	正割（弧度）	asec	反正割（弧度）
csc	余割（弧度）	acsc	反余割（弧度）
sech	双曲正割（弧度）	asech	反双曲正割（弧度）
coth	双曲余切（弧度）	acoth	反双曲余切（弧度）
指数函数			
exp	指数	log10	常用对数（以 10 为底）
pow2	以 2 为底的指数	log	自然对数（以 e 为底）
log2	以 2 为底的对数并将浮点数分解成指数和尾数部分	sqrt	开平方根
取整函数和辅助函数			
ceil	向正无穷方向取整	floor	向负无穷方向取整
fix	向零取整	round	向最近整数取整
mod	除法求余（与除数同号）	rem	除法求余（与被除数同号）
sign	符号函数		

1.3.4 复数与复矩阵

MATLAB 允许使用复数，它通常以实部和虚部表示。基础复数运算函数如表 1-13 所示。

表 1-13　基础复数运算函数

函　　数	描　　述
real(z)	计算 z 的实部
imag(z)	计算 z 的虚部
abs(z)	计算 z 的模值
angle(z)	计算 z 的相角

MATLAB 中定义了 i 和 j 为虚数单位，即 $i\times i=-1$，$j\times j=-1$。在此基础上，可以定义复数，这里给出复矩阵创建和运算的示例。

（1）给定 $\boldsymbol{A}=\begin{bmatrix}1-2i & 3-4i\\ 5-6i & 7-8i\end{bmatrix}$，$\boldsymbol{B}=\begin{bmatrix}1+2i & 5+6i\\ 3+4i & 7+8i\end{bmatrix}$，计算 $\boldsymbol{C}=\boldsymbol{A}\cdot\boldsymbol{B}$，并计算其实部、虚部、模值和相角（real(), imag(), abs(), angle()）。

参考程序：

```
A=[1-2i,3-4i;5-6i,7-8i];                    %%%  定义 A
B=[1+2i,5+6i;3+4i,7+8i];                    %%%  定义 B
C=A*B                                       %%%  计算 C
abs(C),real(C),imag(C),angle(C)
```

结果：

```
C =
1.0e+002 *
          0.3000                  0.7000 - 0.0800i
          0.7000 + 0.0800i        1.7400
ans =
          30.0000    70.4557
          70.4557   174.0000
ans =
          30    70
          70   174
ans =
          0    -8
          8     0
ans =
          0   -0.1138
          0.1138     0
```

（2）给定 $z_1=1+2i$，$z_2=3+4i$，$z_3=5e^{\frac{\pi}{6}i}$，编程计算 $z=z_1\cdot z_2/z_3$。

程序代码如下：

```
z1 = 1+2i;
z2 = 3+3i;
z3 = 5*exp(i*pi/6);
z=z1*z2/z3
```