

## 第2章 MATLAB启动

首先描述如何启动和如何退出 MATLAB，这一点很重要。然后描述如何进行分配和计算，也演示如何储存结果、获得帮助和定义你自己的函数。本章的部分内容对 MATLAB老用户可以跳过不读，但是，建议快速浏览本章，也建议参见附录 A “MATLAB初步”。

### 2.1 启动和退出MATLAB

不同的计算机系统，MATLAB的启动也不一样。在Windows 和Macintosh系统中，程序通常通过点击一个图标而启动。在UNIX系统中，程序是通过在命令行系统提示符后键入如下字符启动：

```
matlab
```

如果上述工作有问题，可请教系统管理员。当启动 MATLAB时，如果 **matlabrc.m**和 **startup.m**文件存在，则执行这些文件。在这些文件中，为满足个人需要，用户可以给定命令以调整 MATLAB，例如，**constants**用于设置图形等。在一个多用户系统上，系统管理员存储 **matlabrc.m**文件，但你也能为自己的使用创建文件 **startup.m**，参见例 2.20(c)。

要退出MATLAB，键入quit或exit。

#### 命令集1 退出和中断

exit, quit	结束 MATLAB会话。程序完成，如果没有明确保存，则变量中的数据丢失。参见 2.8节。
Ctrl-c	中断一个 MATLAB任务。例如，当 MATLAB正在计算或打印时，中断一个任务，但会话并没有结束。

除此之外，对一些系统有指定的菜单选择。例如，在 Windows和Macintosh系统中，在文件菜单下可以找到选项 quit。

当编辑或执行MATLAB时，下列的快捷键十分有用。通常因为不同的平台使用不同的键，因此，给定了一些替换键。在你的系统上试一下这些键，注意哪些键组合使用。

#### 命令集2 特殊的功能键

或 Ctrl_p	恢复前面的命令。
或 Ctrl_n	恢复当前命令之后键入的命令。
或 Ctrl_f	向右移动一个字符。
或 Ctrl_b	向左移动一个字符。
Delete, Backspace	删除字符。
Ctrl_l 或 Ctrl_	向左移动一个字。
Ctrl_r 或 Ctrl_	向右移动一个字。

Ctrl_a 或 Home	移动到行的第一个字符。
Ctrl_e	移动到行尾。
Ctrl_k	删除到行尾。
cedit	在不同的快捷键间转换。键入help cedit可得到更多的信息。

## 2.2 MATLAB中矩阵和 multidimensional 矩阵介绍

MATLAB中数据的基本格式是矩阵。二维矩阵是一个带有以行和列排列的元素的矩形表。如果有 $m$ 行、 $n$ 列，这个矩阵的大小就是 $m \times n$ 。多维矩阵的维数大于2，就是说其大小为 $m \times n \times \dots \times p$ 。

### 例2.1

一个 $2 \times 3$ 的矩阵如下：

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

第1行是(1 2 3)，第2列是 $\begin{pmatrix} 2 \\ 5 \end{pmatrix}$ 。

矩阵的元素，即数 $a_{ij}$ ，通常是实数，但也可以是复数。一个 $a_{ij}$ 是指第 $i$ 行、第 $j$ 列的数。在例2.1中，有 $a_{21}=4$ 。在本章，仅涉及数值矩阵，即矩阵包含的仅是数字。包含字符文本的矩阵（见命令集4）和细胞矩阵，在5.5节中介绍，这些矩阵包含了不同类型的数据。

当矩阵仅由一行组成时，它是一个特例，就是一个行向量。如果矩阵仅有一列，就是一个列向量。向量是矩阵的特例。向量中元素的数量是向量的长度。

如果矩阵的维数是 $1 \times 1$ ，它是一个标量，即是一个数。

在MATLAB中，一个变量可以通过给它分配一个值来定义，如下所示：

```
variable =expression
```

在expression之后按回车键。表达式可以由数字、变量、操作符和函数等组成。

定义一个变量的另一个方法是输入 expression项，然后MATLAB对预定义变量ans(answer的缩写)分配这个表达式值。

二维矩阵的分配可以有多种方法实现。最简单的方法是由方括号 [] (参见help paren) 包围的逐行给定元素。如果定义一个标量，则方括号就不需要了。

相同行中的元素是由一行或多个空格 ' ' 或一个逗号 ' ,' 分隔，列由分号 ' ; ' 或回车键分隔。没有结尾分号的每个命令在屏幕上显示出其结果。若结尾带分号，就执行计算，但计算结果并不显示。在MATLAB中使用的标点符号的一览表可以通过输入help punc得到。

一个变量的值可以通过输入它的名字和按回车键获得，MATLAB以显示这个变量的名字和值作为回答。如果这个变量并不存在，就显示一个错误信息。显示一个变量内容的另一个方法可参见5.1.3节。

一个矩阵或一个向量的指定元素是由指定它的索引来决定。例如：二维矩阵：

```
variable (rowIndex, columnIndex)
```

如果这个变量是一个向量，就只允许有一个索引。如何处理多维矩阵将在例 2.3中讨论。

## 例2.2

(a) 一个标量的指定。如果写入  $x=7$ ，则在屏幕上打印如下：

```
x=
    7
```

(b) 如果仅写入 7，则结果变为：

```
ans=
    7
```

(c)  $2 \times 3$  维情况下一个矩阵的定义可以通过逐行给出其元素：

```
A=[1  2  3
   4  5  6]
```

还在屏幕上给出如下结果：

```
A=
    1    2    3
    4    5    6
```

(d) 也可以在同一行上用分号来分隔行以给出所有的元素：

```
A=[ 1  2  3 4  5  6];
```

在命令后的一个分号禁止打印出结果。

(e) 一个行向量和一个列向量的定义：

```
rowvec=[1.2  3.2  4];
colvec=[2.7; 3.4; 9.2];
```

(f) 显示一个变量的值。输入 `colvec`，MATLAB显示：

```
colvec=
    2.7000
    3.4000
   -9.2000
```

(g) 逐个元素地分配矩阵：

```
B(1,1)=1;
B(1,2)=7;
B(2,1)=-5;
B(2,2)=0
```

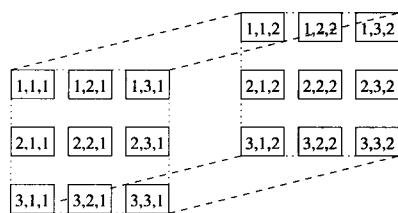
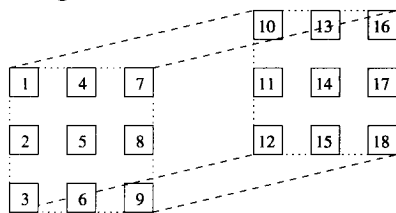
得到的结果是：

```
B=
    1    7
   -5    0
```

三维矩阵和其他数据结构在函数自变量中依次使用行、列和页维数次序。对于多维矩阵有两个索引原理，最自然的就是矩阵索引，它给出了每个元素在每一维中的一个位置，例如，在上例中的行和列的索引。图 2-1 给出了可视化的  $3 \times 3 \times 2$  的三维矩阵的不同元素的索引。

另一个索引原理是线性索引方法。一些命令把整个矩阵说明为一个长列元素，例子之一是 `reshape` (详见命令集 37)。如果所有的元素被放在一行上，那么对各元素给定的一个线性索引说明了一个指定矩阵的索引给出的某个位置，如图 2-2 所示。

创建一个多维矩阵有多种方法。

图2-1 一个 $3 \times 3 \times 2$ 矩阵的元素索引图2-2 一个 $3 \times 3 \times 2$ 矩阵的线性索引

## 例2.3

(a) 有两个二维矩阵A和B

A=

1	2	3
4	5	6

B=

11	12	13
14	15	16

可以很容易地构造一个三维矩阵C：

 $C(:, :, 1) = A;$  $C(:, :, 2) = B;$ 

给出：

 $C(:, :, 1) =$ 

1	2	3
4	5	6

 $C(:, :, 2) =$ 

11	12	13
14	15	16

(b) 为改变C中的一个元素，可以输入：

 $C(1, 1, 1) = 100;$ 

得到：

 $C(:, :, 1) =$ 

100	2	3
4	5	6

MATLAB允许在同一命令行上定义多个变量，也可以在按回车键之前通过输入三个点“... ”以便在下一行继续输入。

## 例2.4

(a) 在一行上的几个命令：

 $x=7; y=4.6735567; z=x^y;$ 

(b) 一个长的命令可以分成几行书写：

```
mat1 = [1.2 1.1 -1.1 1.4 1.1 -1.1 -1.2 ...
-1.1 -1.3 1.7];
```

MATLAB记忆不同矩阵变量的维数。为了获得一个变量的维数，可以使用命令size和length。

现在，令  $A$  是一个  $m \times n \times \dots \times p$  矩阵， $x$  是  $m \times 1$  矩阵（一个列向量）或是  $1 \times n$  矩阵（一个行向量），MATLAB 有如下的命令集：

### 命令集3 变量大小

<code>size(A)</code>	给出包含 $A$ 的维数的一个行向量。在这个返回向量中的第一个元素是行数，随后是列数、页数等。
<code>[m,n,...,p]= size(A)</code>	给出 $A$ 的维数、 $m$ 行数和 $n$ 列数，即两个标量。如果给出的自变量少，则后边的维数加入最后一个自变量。
<code>size(A,dim)</code>	在数组 $dim$ 中给出 $A$ 的维数。
<code>size(x)</code>	给出向量 $x$ 的大小或长度的一个行向量。如果 $x$ 是一个列向量，则第1个元素是 $m$ ，第2个元素是1。如果 $x$ 是一个行向量，则第1个元素是1，第2个元素是 $n$ 。
<code>length(x)</code>	给出一个向量的长度，即如果 $x$ 是一个行向量，那么这个向量的长度就是 $n$ ；如果 $x$ 是一个列向量，那么这个向量的长度就是 $m$ 。
<code>length(A)</code>	给出 $m, n, \dots, p$ 的最大数。
<code>ndims</code>	返回多维矩阵 $A$ 的维数。这个函数等价于 <code>length(size(A))</code> 。
<code>sub2ind(size,m,n,...)</code>	给出维数为 <code>size</code> 的一个矩阵的线性索引号 $m, n, \dots$ 。
<code>[m,n,...]= ind2sub (size,ind)</code>	用线性索引 <code>ind</code> 对元素给出索引 $(m, n, \dots)$ ，要把图像矩阵说明为多维，其参数 <code>size</code> 必须是一个向量。

**注意** 有时用字 `dimemsion` 代替维数和长度，通常它可以表示与一个向量范数相关的长度和与一个矩阵范数相关的矩阵的维数。可是，在本书中，维数和长度的使用与 `size` 和 `length` 有相同的含义。当这个长度概念以其他意义使用时，就要说明。

### 例2.5

(a) 命令 `thesize1=size(A)`，式中的  $A$  与例2.2(c) 中的相同，结果为：

```
thesize1=
      2      3
```

命令 `thesize2=size(C)`，式中的  $C$  是例2.3中的三维矩阵，得到：

```
thesize2=
      2      3      3
```

(b) 检查图2-2中的信息是否正确，可以使用：

```
sub2ind([3 3 2],1,2,2)
```

```
ans =
    13
```

## 2.3 MATLAB中的变量

在MATLAB中, 变量名可以有19个字符。字母A~Z、a~z、数字和下划线‘\_’可以作为变量名, 但第一个字符必须是一个字母。预定义函数名也可以像一个变量名那样使用, 但函数只有在变量由命令clear删除后才能使用, 所以, 不主张这样使用。

MATLAB是区分大小字母的, 如矩阵a和A是不一样的。MATLAB命令通常是用小写字母书写。例如, 命令abs(A)给出了A的绝对值, 但ABS(A)会导致在屏幕上显示如下错误信息:

```
??? Undefined variable or function ABS; Caps Lock may be on
```

在变量使用之前, 用户不需要指定一个变量的数据类型, 也不必声明变量。MATLAB有许多不同的数据类型, 这对决定变量的大小和形式是有价值的, 特别适合于混合数据类型、矩阵、细胞矩阵、结构和对象。

对于每一种数据类型, 有一个名字相同的、可以把变量转换到那种类型的函数。所用的不同的基本数据类型如下所示。

### 命令集4 数据类型和转换函数

double	是一个双精度浮点数, 每个存储的双精度数用 64位。
char	用于存储字符, 每个存储的字符用 16位。
sparse	用于存储稀疏矩阵, 由一个 sparse使用的内存是4+(非零元素数*16)。
uint8	是一个无符号的8位整数。数学函数并不对使用到的这种数据类型进行定义, 如存储图像。

混合数据类型在第5章和第12章中描述。

在MATLAB中, 有许多功能可以帮助找出一个变量是否是一个特殊类型。也有一个特殊的逻辑向量, 它是由命令 repmat生成(见第4.1节)。

### 命令集5 逻辑函数

iscell(x)	如果x是一个细胞矩阵, 返回1; 否则为0。可参见第5.5节。
isfield(x)	如果x在一个结构中是一个域, 返回1; 否则为0。可参见第12.5节。
isfinite(x)	返回一个与x相同大小的向量, 这个x包含有限元的位置为1, 其他位置为0。
islogical(x)	如果x是一个逻辑向量, 返回1; 否则为0。
isnumeric(x)	如果x是一个数值向量, 返回1; 否则为0。
isstr(x)	如果x是一个字符串, 返回1; 否则为0。可参见第5.1节。
isstruct(x)	如果x是一个结构, 返回1; 否则为0。可参见第12.5节。
isobject(x)	如果x是一个对象, 返回1; 否则为0。可参见第12.6节。
logical(x)	返回一个可以使用的逻辑向量, 例如逻辑索引或逻辑测试。

#### 例2.6

如果一个函数是要应用到一个矩阵中的隔一个元素上, 那么可以这样做:

```
data = rand(1,10)
```

```
data =
```

```
    0.6700    0.2009    0.2731    0.6262    0.5369    0.0595
    0.0890    0.2713    0.4091    0.4740
```

使用`repmat`创建`x`：

```
x = repmat([1 0],1,5)
```

```
x =
```

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

```
filter = logical(x)
```

```
filter =
```

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

为使用滤波器，给出下面的命令：

```
halfdata = data(filter)
```

```
halfdata =
```

```
    0.6700    0.2731    0.5369    0.0890    0.4091
```

为使用与函数`round`一起的过滤器，可输入：

```
result = round(data(filter))
```

```
result =
```

```
    1    0    1    0    0
```

命令`repmat`产生块矩阵，第 4.1 节将进一步对此进行描述。

在MATLAB中有许多如下的预定义变量：

#### 命令集6 MATLAB中预定义变量

<code>ans</code>	分配最新计算表达式的值，这个表达式并没有给定一个名字。
<code>eps</code>	返回机器精度，定义 1 与最接近可代表的浮点数之间的差。 <code>eps</code> 数在一些命令中用作偏差。用户可以设定一个新的 <code>eps</code> 值，但要注意这个 <code>eps</code> 值不能由命令 <code>clear</code> 恢复。
<code>realmax</code>	返回计算机能处理的最大浮点数。
<code>realmin</code>	返回计算机能处理的最小的非零浮点数。
<code>pi</code>	返回 $\pi$ ，即 3.141592653589793，如果 <code>eps</code> 足够小，那么用 16 位十进制数来表示其精度。
<code>inf</code>	定义为 $1/0$ 。当出现被零除时，MATLAB 就返回 <code>inf</code> ，并不中断执行而继续计算。
<code>NaN</code>	定义为“Not a Number”，这个非数值要么是 % 类型，要么是 <code>nf/inf</code> 。

<code>i, j</code>	定义为 $\sqrt{-1}$ ，虚数单位。可以为 <i>i</i> 和 <i>j</i> 分配其他值，它们将不再是预定义常数。可以由 <code>clear</code> 命令恢复。
<code>nargin</code>	给出在一个函数调用中输入自变量的个数，可参见第 12.3 节。
<code>nargout</code>	给出在一个函数调用中输出自变量的个数，可参见第 12.3 节。

为寻找哪个变量可以被定义，可以使用下列命令集：

#### 命令集7 变量列表

<code>who</code>	列出已定义的变量。
<code>Who global</code>	与 <code>who</code> 相同，但仅列出全局变量。参见第 12.3 节。
<code>who a*</code>	给出所有以 <i>a</i> 开头的变量的一个列表。
<code>whos</code>	给出比命令 <code>who</code> 更详细的列表，如显示矩阵的维数。
<code>whos global</code>	与 <code>whos</code> 相同，但仅列出全局变量，可参见第 12.3 节。
<code>exist(namestr)</code>	根据在字符串 <code>namestr</code> 中的变量的定义，返回不同的值。关于字符串更多的信息可参见第 5 章。这里要注意的重要的一点是变量名应在引号 ‘ ’ 之间给出。函数返回值的情况是： <ol style="list-style-type: none"> <li>1) 表示 <code>namestr</code> 是一个变量名；</li> <li>2) 表示 <code>namestr</code> 是一个 M 文件名(参见第 2.9 节)；</li> <li>3) 表示 <code>namestr</code> 是一个 MEX 文件名(参见第 15 章)；</li> <li>4) 表示 <code>namestr</code> 是一个编译的 SIMULINK 函数；</li> <li>5) 表示 <code>namestr</code> 是一个预定义的 MATLAB 函数名。</li> </ol>
<code>inmem</code>	返回一个带字符串的细胞向量，这个字符串包含目前在内存中的函数、M 文件。如果给出两个输出参数，则第二个包含了一个目前在内存中的 MEX 文件的列表。
<code>workspace</code>	对由 <code>whos</code> 得到的信息给出一个图形界面。命令 <code>clear</code> 被集成在这个环境中。由 <code>help workspace</code> 给出的信息，也在 UNIX 下工作。

如果不被用户删除或重命名，每个被定义的变量将在整个过程中保留。要删除变量，系统会劝告如要处理大矩阵可用命令 `clear`。

#### 命令集8 删除变量和合并

<code>clear</code>	删除所有变量并恢复除 <code>eps</code> 外的所有预定义变量。可参见命令集 111 当运行文件时的 <code>clear</code> 。
<code>clear name</code>	仅删除变量 <code>name</code> 。
<code>clear name1 name2</code>	删除变量 <code>name1</code> 、 <code>name2</code> 、...
...	
<code>clear a</code>	删除所有 <i>a</i> 开头的变量。
<code>clear value</code>	根据 <code>value</code> 给出不同的结果。键入 <code>help clear</code> 可得到更多的细节。



pack

重组和压缩已分配的内存碎块。当MATLAB的内存满后，可以使用命令pack而不是清除任何变量来得到更多的空间。将会产生如下情况：

- 所有变量都会保存在磁盘上的一个临时文件 **pack.tmp** 中；
- 删除主内存中的内容；
- 所有变量将从 **pack.tmp** 加载到主内存中；
- 删除文件 **pack.tmp**。

pack filename 用文件filename作临时文件，重组和压缩已分配内存。

键入 ~~help~~ **pack**、~~help~~ **pack** 和 ~~help~~ **pack** 可以获得更多的有关如何在MATLAB节省内存的信息。

注意 MATLAB中的命令实际上可以看作函数，把字符串看作自变量，这就意味着有两种描述：

command argument

command('argument')

这两种描述是等价的。例如，clear name与clear('name')得到相同的结果，who global与who('global')等价。其他的例题可在本书的其他几个地方找到，例如 axis square与axis('square')在第13.3节中可找到。由于命令自变量可以通过使用字符串控制命令而产生，使用函数/字符串公式的可能性使得MATLAB编程语言十分灵活，参见第5章。

## 2.4 算术表达式和数学函数

在MATLAB中通常的惯例是书写数字。对十进制数，使用科学记数法可以书写十分大和十分小的数。例如3.14和1.23E - 6，这里，后者代表  $1.23 \times 10^{-6}$ 。

MATLAB有算术运算符的扩展集，它们是：

- 1) ^ 幂
- 2) \* 乘
- / 右除(正常除)
- \ 左除
- 3) + 加
- 减

这是按序给出的运算，1是最高优先级。在带相同优先级的运算符表达式中，按从左到右的顺序执行。圆括号()能够用于改变优先级次序。

在第3.3节中，将看到两种不同的除法是有用的。对于数量右除  $2/5$  得0.4与左除  $5 \backslash 2$  是相同的，斜线号“靠着”的表达式或数字是分母。

### 例2.7

如果书写  $a/b+c$ ，MATLAB读作  $\frac{a}{b}+c$ ，但表达式  $a/(b+c)$  被读作  $\frac{a}{b+c}$ 。

如果使用左除  $a \backslash (b+c)$ ，MATLAB把它译成  $\frac{b+c}{a}$ 。

MATLAB包含了预定义数学函数，它们可以用于算术表达式中。如果自变量是复数，那么，

多数情况下是答案。

MATLAB也能计算含有预定义变量的表达式：例如，一个表达式可以用作一个函数的自变量。

预定义数学函数在命令集9中列出。即使这些函数是为数量自变量描述的，将在第3.6节看到它们既能处理向量，也能处理矩阵。要注意的是所有三角函数都需要其自变量以弧度表示。

### 命令集9 数学函数

<code>abs(x)</code>	求 $x$ 的绝对值，即 $ x $ 。
<code>sign(x)</code>	求 $x$ 的符号，如果是正的得1；负的得-1；零得0。
<code>sqrt(x)</code>	求 $x$ 的平方根，即 $\sqrt{x}$ 。
<code>pow2(x, f)</code>	求 $x \times 2^f$ 。把 $f$ 加到 $x$ 的浮点格式下的指数上计算是一种十分有效的运算。
<code>exp(x)</code>	求 $x$ 的指数函数，即 $e^x$ 。
<code>log(x)</code>	求 $x$ 的自然对数，即 $\ln x$ 。
<code>log10(x)</code>	求 $x$ 以10为底的对数，即 $\log_{10} x$ 。
<code>log2(x)</code>	求 $x$ 以2为底的对数，即 $\log_2 x$ 。
<code>sin(x)</code>	求正弦 $x$ ， $x$ 为弧度。
<code>cos(x)</code>	求余弦 $x$ ， $x$ 为弧度。
<code>tan(x)</code>	求正切 $x$ ， $x$ 为弧度。
<code>cot(x)</code>	求余切 $x$ ，即 $1/(\tan x)$ ， $x$ 为弧度。
<code>asin(x)</code>	求反正弦，即 $\sin^{-1} x$ 。
<code>acos(x)</code>	求反余弦，即 $\cos^{-1} x$ 。
<code>atan(x)</code>	求反正切，即 $\tan^{-1} x$ 。
<code>atan2(x, y)</code>	求四象限反正切 $(x/y)$ ，其结果在 $[-\pi, \pi]$ 区间内。
<code>acot(x)</code>	求反余切 $x$ =四象限反正切 $(1/x)$ 。
<code>sec(x)</code>	求正割 $x$ ，即 $1/(\cos x)$ 。
<code>csc(x)</code>	求余割 $x$ ，即 $1/(\sin x)$ 。
<code>asec(x)</code>	求 $\sec^{-1} x = \arccos(1/x)$ 。
<code>acsc(x)</code>	求 $\csc^{-1} x = \arcsin(1/x)$ 。
<code>sinh(x)</code>	求双曲正弦 $x$ 。
<code>cosh(x)</code>	求双曲余弦 $x$ 。
<code>tanh(x)</code>	求双曲正切 $x$ 。
<code>coth(x)</code>	求双曲余切 $x$ ，即 $1/(\tanh x)$ 。
<code>asinh(x)</code>	求 $\sinh^{-1} x = \ln(x + \sqrt{1+x^2})$ 。
<code>acosh(x)</code>	求 $\cosh^{-1} x = \ln(x + \sqrt{1-x^2})$ 。
<code>atanh(x)</code>	求 $\tanh^{-1} x = 0.5 \ln((1+x)/(1-x))$ 。
<code>acoth(x)</code>	求 $\coth^{-1} x = 0.5 \ln((x+1)/(x-1))$ 。
<code>sech(x)</code>	求双曲正割 $x$ ，即 $1/(\cosh x)$ 。
<code>csch(x)</code>	求双曲余割 $x$ ，即 $1/(\sinh x)$ 。
<code>asech(x)</code>	求 $\operatorname{sech}^{-1} x = \ln((1+\sqrt{1-x^2})/x)$ 。
<code>acsch(x)</code>	求 $\operatorname{csch}^{-1} x = \ln((1+\sqrt{1+x^2})/x)$ 。

## 例2.8

(a) 如果键入 `sinepi=sin(pi)`，就得到：

```
sinepi =
    1.22466e - 16
```

这个结果并不是精确地为0，因为`pi`是  $\pi$  的近似值，在计算中有舍入误差。

(b) `logarithm = log10(100)`

```
logarithm =
    2
```

(c) `e = exp(1)`

```
e =
    2.7183
```

在MATLAB中有几个命令用于数的取整。在命令集10中， $x$ 是一个浮点数或是一个带浮点元素的矩阵。

## 命令集10 取整命令和有关命令

<code>round(x)</code>	求最接近 $x$ 的整数。如果 $x$ 是一个向量，则适用于所有元素。
<code>fix(x)</code>	求0方向最接近的整数。即负向上四舍五入，正向下四舍五入。
<code>floor(x)</code>	求小于或等于 $x$ 的最接近的整数。
<code>ceil(x)</code>	求大于或等于 $x$ 的最接近的整数。
<code>rem(x, y)</code>	求整除 $x/y$ 的余数。
<code>gcd(x, y)</code>	求整数 $x$ 和 $y$ 的最大公因子。
<code>[g, c, d]=gcd(xy)</code>	求 $g, c, d$ ，满足 $g=xc+yd$ 。
<code>lcm(x, y)</code>	求正整数 $x$ 和 $y$ 的最小公倍数，也能用于决定最小公因子。
<code>[t, n]=rat(x)</code>	由有理数 $t/n$ 求 $x$ 的近似值，这里的 $t$ 和 $n$ 是整数，相对误差小于 $10^{-6}$ 。也可参见 <code>rats</code> (第5.1.2节)，它给出了对应的字符串。
<code>[t, n]=rat(x, tol)</code>	与上相同，但相对误差小于 $tol$ 。
<code>rat(x)</code>	求 $x$ 的连续的分数表达式。
<code>rat(x, tol)</code>	求带相对误差 $tol$ 的 $x$ 的连续的分数表达式。

## 例2.9

(a) 取整有几种方法，命令是：

```
x = -1.49;
rdx = round(x), fixx = fix(x), flx = floor(x), clx = ceil(x)

return

rdx =
    -1

fixx =
    -1
```

```
flx =
    -2
```

```
clx =
    -1
```

(b)由有理数 $t/n$ 求 $\sqrt{2}$ 的近似值：

```
[t,n] = rat(sqrt(2))
```

```
t =
    1393
```

```
n =
    985
```

为了与真值比较，键入`differ=sqrt(2) - t/n`，得：

```
differ=
    3.6440e-07
```

由此可见，差值不大。如果在函数 `rat` 对参数 `tol` 指定一个较小的值，那么这个差值将会是较小的。

在MATLAB中，大多数情况下是允许复数值表达的。加入变量 `i` 和 `j` 返回虚数单位，即  $\sqrt{-1}$  的值，能用于产生复数。这是可以用名字 `i` 和 `j` 作为变量的名字，一个新的复数单位可以由此产生：

```
ii=sqrt(-1);
```

由于空格是分隔元素的，因此在书写复数元素时要慎用空格。参见下面例题 2.10(c)。

#### 例2.10

(a)  $z = 3 + 4i$

```
z =
    3.0000 + 4.0000i
```

(b)一个较复杂的表达式：

```
w=r*exp(i*theta); comp=z*w;
```

式中，`r`和`theta`是一个已经定义的变量。

(c) 向量也可以是复数；

```
complexvector=[1-i 2-2i 3 -3i ]
```

返回：

```
complexvector=
    1.0000 -1.0000i    2.0000-2.0000i    3.0000 -3.0000i
```

注意，在3与-3i之间的空格使MATLAB读取它们时看作为两个分隔的复数。如果键入下列内容，可以看到此结果：

```
length(complexvector)
```

给出的结果是：

```
ans=
    4
```

MATLAB中有一些处理复数和函数的命令。

### 命令集11 有关复数的函数

<code>real(z)</code>	求 $z$ 的实部。
<code>imag(z)</code>	求 $z$ 的虚部。
<code>abs(z)</code>	求 $z$ 的绝对值, 即 $ z $ 。
<code>conj(z)</code>	求 $z$ 的复数共扼, 即 $\bar{z}$ 。
<code>angle(z)</code>	求 $z$ 的相角, 即 $z=x+iy=re^{i\theta}$ 中的 $\theta$ 。
<code>unwrap(v)</code>	求与 $v$ 相同长度的向量。这里, 两个相邻元素间的相角差已经改变, 因此, 差最大为 $\pi$
<code>unwrap(v, k)</code>	求出如上的一个向量, 但用转移偏差 $k$ 代替 $\pi$
<code>cplxpair(v)</code>	给出一个 $v$ 中各元素按实部递增排序, 并使其复数组合成复数共扼对的一个向量。在一个共扼对中, 负虚部在前, 实元素排在向量的后部。如果 $v$ 的一个元素在 $v$ 中没有它自己的复数共扼, 则显示一个错误信息。

#### 例2.11

令复数 $z$ 为:

```
z=1+2i ;
```

(a)  $z$ 的实部和虚部由下面求出:

```
realpart = real(z), imagpart = imag(z)
```

```
realpart =  
1
```

```
imagpart =  
2
```

(b) 复数共扼由 `conjugate=conj(z)` 求出:

```
conjugate=  
1.0000 - 2.0000i
```

$z$ 的绝对值由 `absz=abs(z)` 求出:

```
absz=  
2.2361
```

(c) 一个复数的自变量, 即复平面中的相角由 `arg=angle(z)` 求出:

```
arg=  
1.1071
```

MATLAB也用于坐标系之间转换的函数。这些函数既能对向量、也能对矩阵进行运算, 其结果将有与输入自变量相同的维数。

### 命令集12 坐标转换

<code>[theta, r]=</code>	将笛卡尔坐标转换为极坐标, 极坐标 $\theta$ 和 $r$ 是由卡笛尔 $x$
<code>cart2pol(x, y)</code>	和 $y$ 得到。

`[x, y]=pol2cart(theta, r)` 将极坐标转换为笛卡尔坐标，笛卡尔坐标 $x$ 和 $y$ 是从极坐标 $theta$ 和 $r$ 得到。

`[alpha, theta, r]=cart2sph(x, y, z)` 将笛卡尔坐标转换为球坐标，角 $alpha$ 、 $theta$ 和长度 $r$ 是从笛卡尔坐标 $x$ 、 $y$ 和 $z$ 得到。

`[x, y, z]=sph2cart(alpha, theta, r)` 将球坐标转换为笛卡尔坐标 $x$ 、 $y$ 和 $z$ 。

在MATLAB中也有更高级的预定义数学函数。

### 命令集13 特殊的数学函数

`legendre(n, x)` 返回一个 $n+1$ 长度的向量，代表与 $n$ 次legendre函数相关的值和对 $x$ 的0到 $n$ 次计算的值。如果 $x=x$ 是一个向量，则这个命令返回一个矩阵，列是为 $x$ 的每一个元素计算的Legendre函数值。 $x$ 的每个元素都必须在 $[-1, 1]$ 范围内。

`bessel(n, x)` 求第1类贝塞尔函数， $n$ 和 $x$ 可以是向量，但 $n$ 必须是步长为1的递增，范围为 $[0, 1000]$ 。这个命令根据 $x$ 是否是复型而调用不同的程序，但这些程序能被直接调用。输入`help besse`可以得到更多的信息。

`bessely(n, x)` 求与`bessel`有相同自变量的第2类贝塞尔函数。

`gamma(x)` 求 函数，即对正 $x$ ：

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$$

要获得负 $x$ 的定义，键入`help gamma`

`gammainc(x, a)` 求不完全 函数

$$\frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t} dt$$

`gammainc(x)` 求 函数的自然对数。用`log(gamma(x))`可以避免上下溢出。

`beta(x, y)` 求B函数，即：

$$\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

自变量 $x$ 必须在 $[0, 1]$ 间隔内，如果由三个自变量调用此函数，可以使用下面的命令`betainc`。

`betainc(x, a, b)` 求不完全 函数，类似地定义不完全 函数。

`betaln(x, y)` 求 函数的自然对数。

`expint(x)` 求  $\int_x^{\infty} \frac{e^{-t}}{t} dt$

erf(x)

求误差函数，即积分：

$$\frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

erfinv(y)

求逆误差函数。

erfc(x)

求互补误差函数  $1 - \text{erf}(x)$ 。

erfcx(x)

求成比例的互补误差函数。键入 `help erfc` 可得到更多信息。

[k, e]=ellipke(m)

求第1和第2类  $0 < m < 1$  的完全椭圆函数

[j1, j2, j3]=ellipj(x, m)

求Jacobi椭圆函数。

此外，用户可以定义自己的函数，参见第 2.9 节。一些特殊的数学函数在第 10.4 节中叙述。

## 2.5 计算浮点运算次数和时间管理

当相互比较不同算法时，计算一个会话期或一个部分会话期中的算术运算的次数是十分有用的。为了获得浮点运算 (flop) 的近似次数，使用命令 `flops`。显示部分 MATLAB 程序所占时间的计时方法将在第 12.7 节中描述。

### 命令集14 浮点运算计数器

`flops`

返回完成浮点运算的近似次数。最新的计数器值是 0，这是 MATLAB 启动时的缺省值。

加法和减法运算时，如果是实数运算，计入 1 个运算；如果是复数运算，计入 2 个运算。乘法和除法时，如果是实数运算，计入 1 个运算；如果是复数运算，计入 6 个运算。调用初等函数时，如果自变量是实数，计入 1 个运算；如果是复数，计数较多，具体次数随具体的函数而定。

`flops(0)`

计数器重置为零。

#### 例2.12

计算运算的次数。命令如下：

```
flops(0); x = 10 + 20 + 30*40/50;
numflops = flops
```

如期望的一样，其结果为：

```
numflops=
    4
```

MATLAB 能够告诉你日期和时间，并且给出计算机的有关信息。与命令 `flops` 一起使用，这些命令可用于分析一个算法的有效性。

### 命令集15 时间和日期(一)

`tic`启动一个可用命令 `toc` 读的时钟。`toc`读时钟，即显示开启时钟以来的时间。如果时钟没有运行 `toc` 返回 0 值。

<code>clock</code>	返回用十进制数表示日期和时间的具有 6 个元素的行向量。前 5 个元素是整数，秒由几个十进制小数表示。命令 <code>fix(clock)</code> 四舍五入至最接近整数的秒。
<code>etime(t1, t2)</code>	计算 <code>t1</code> 和 <code>t2</code> 时间间隔内所消耗的时间，以秒计算。 <code>t1</code> 和 <code>t2</code> 是表示日期和时间的 6 个元素的行向量。
<code>cputime</code>	返回以秒计的 MATLAB 自启动以来所用的 CPU 时间。

## 例2.13

下列方法可进行时间操作：

写入下式保存当前时间

```
t1=clock
```

写入 MATLAB 命令

```
timedifference=etime(t1, clock)
```

得到完成上述两条命令所用的间隔时间。

MATLAB 有处理日期的内部函数。一些系列函数的使用组成了一个描述日期的整数部分和一个描述时间的小数部分。

## 命令集 16 时间和日期(二)

<code>date</code>	以日—月—年字符串形式返回当前日期。
<code>calendar(yyyy, mm)</code>	显示当年 <code>yyyy</code> 当月 <code>mm</code> 按周排列的 $6 \times 7$ 矩阵形式的日历。
<code>datenum(yyyy, mm, dd)</code>	给出当年 <code>yyyy</code> 当月 <code>mm</code> 当日 <code>dd</code> 的序列数字。日期 0000—01—01 为 1 天。
<code>datestr(d, form)</code>	返回 <code>form</code> 格式的序列数字表示的日期，见表 2-1。
<code>datetick(axis, form)</code>	用于在图中的坐标轴上写数据。
<code>datevec(d)</code>	如果 <code>d</code> 是一个由诸如 <code>datestr</code> 返回的格式表示的序列数字或日期，则返回一个向量 <code>[yyyy mm dd hh mmise]</code> 。
<code>eomday(yyyy, mm)</code>	返回当年当月的天数。
<code>now</code>	返回当天和当时的序列数字。
<code>[daynr dayname]= weekday(day)</code>	返回 <code>dayname</code> 以告知当天 <code>day</code> 是否星期天、星期一等。 <code>daynr</code> 等于一周中当天 <code>day</code> 的序列数字。这里的 <code>day</code> 是一个字符形式或序列数字的日期。

在 `datestr` 中可以使用的不同的日期格式如表 2-1 所示。用 1986 年 4 月 26 日 2 时 14 分作为例



子列表介绍。

表2-1 日期格式

0	格式 <code>dd mmm yyyy HH:MM:SS</code>	<code>26—Apr—1986 02:14:00</code>
1	格式 <code>dd mmm yyyy</code>	<code>26—Apr—1986</code>
2	格式 <code>mm/dd/yy</code>	<code>04/26/86</code>
3	格式 <code>mmm</code>	<code>Apr</code>
4	格式 <code>m</code>	<code>A</code>
5	格式 <code>m#</code>	<code>4</code>
6	格式 <code>mm/dd</code>	<code>04/26</code>
7	格式 <code>dd</code>	<code>26</code>
8	格式 <code>ddd</code>	<code>Sat</code>
9	格式 <code>d</code>	<code>S</code>
10	格式 <code>yyyy</code>	<code>1986</code>
11	格式 <code>yy</code>	<code>86</code>
12	格式 <code>mmmyy</code>	<code>Apr86</code>
13	格式 <code>HH:MM:SS</code>	<code>02:14:00</code>
14	格式 <code>HH:MM:SS PM</code>	<code>2:14:00 AM</code>
15	格式 <code>HH:MM</code>	<code>02:14</code>
16	格式 <code>HH:MM PM</code>	<code>2:14 AM</code>
17	格式 <code>QQ-YY</code> ，这里 <code>QQ</code> 表示几刻钟	<code>Q2—86</code>
18	格式 <code>QQ</code>	<code>Q2</code>

#### 例2.14

为了获取指定日期是星期几，使用如下语句并得到相应结果：

```
[dnr dname] = weekday('26--April--1986')
```

```
dnr =
```

```
7
```

```
dname =
```

```
Sat
```

在MATLAB中，一周是从星期天开始的，这使得星期六的星期数为7。

## 2.6 输出格式

在屏幕上，通常以不带小数的整数格式或带四位小数的短浮点数格式显示结果。

如果一个矩阵中所有的元素都是整数，那么它们将以整数格式显示。但是，如果有一个或一个以上的元素是非整数，则所有的元素都以浮点数格式显示。

输出格式在计算中不影响精度，MATLAB总是按高精度完成计算。对大多数的计算机而言，MATLAB在计算中使用16位小数。

命令`format`用于改变输出格式。在Windows和Macintosh版本中，输出格式也能通过命令窗体中的下拉菜单控制。

## 命令集17 数字输出格式

<code>format deformat</code>	将输出格式改为由 <i>deformat</i> 定义的格式，这类格式可以是如下之一： <code>short</code> 、 <code>long</code> 、 <code>short e</code> 、 <code>long e</code> 、 <code>hex</code> 、 <code>+</code> 、 <code>bank</code> 、 <code>rat</code> 。也有 <code>compact</code> 或 <code>loose</code> ，它给出了一个较紧缩或较宽松的输出格式，但并不影响数值输出格式。
<code>more on</code>	本书中的所有例题都使用 <code>format compact</code> 。当全屏时停止显示，在进一步删除显示之前等待键盘输入。在窗体底部，MATLAB 打印— <i>more</i> —以指示有更多的信息要显示。
<code>more off</code>	不考虑窗体是否足够大而给以输出。
<code>more(n)</code>	如果输出多于 <i>n</i> 行，则显示输出 <i>n</i> 行。

## 例2.15

设  $p=1+1/3$ ，先定义格式，然后在屏幕上显示 *p*：

<code>format short</code>	得 1.3333	4位小数
<code>format long</code>	得 1.33333333333333	14位小数
<code>format short e</code>	得 1.3333e+00	4位小数
<code>format long e</code>	得 1.33333333333333e+00	15位小数
<code>format hex</code>	得 3ff5555555555555	16进制数
<code>format +</code>	得 +	正：+ 负：- 或零：0
<code>format bank</code>	得 1.33	美元和美分
<code>format rat</code>	得 4/3	作为一个有理数

MATLAB 在输出中使用的空格数是可以减少的。当输出行长于窗体时。也可以引导输出。本书在实例中使用了 `format compact` 来减少空格数。

## 2.7 帮助命令和示范

帮助总是可以通过使用下列命令中的任何一条获得。

## 命令集18 帮助命令

<code>help</code>	给出大约 20 个主题的列表，每个主题给出了基本信息。这些主题以目录给出，有关每个主题的信息通过 <code>help dir</code> 给出，这里的 <code>dir</code> 是目录。
<code>help command</code>	对指定的命令给出帮助。
<code>help dir</code>	给出目录 <code>dir</code> 的内容。
<code>hthelp</code>	对一个超链接数据库打开一个 MATLAB GUI。这个命令用 <code>helpdesk</code> 替换，也可能在今后的版本中删除。
<code>http</code>	是链接到 <code>hthelp</code> 中的帮助文件的一个预处理器。
<code>loadhtml</code>	通过 <code>hthelp</code> 和 <code>http</code> 用于加载、中断和显示 HTML 文件。

helpdesk	在MATLAB帮助桌面上启动一个带索引页的 Web浏览器。
doc command	在MATLAB帮助桌面上得到命令 <i>command</i> 的帮助页。
web URL	将浏览器指向 <i>URL</i> , 如果需要则打开一个浏览器。参见 <code>help web</code> 可以得到更多信息。
lookfor text	在所有的M文件的第1行中查找字符串 <i>text</i> 。
demo	给出一个MATLAB的不同命令、函数和应用领域的示范。命令 <code>demo</code> 运行MATLAB Expo , 它显示选择不同示范实例的一个菜单。也可以找到几个简单的游戏。
expo	运行MATLAB Expo ; 也可参见 <code>demo</code> 。
info	给出有关MATLAB的信息。例如 , 什么种类的计算机能够运行MATLAB , 如何得到更多的有关 MATLAB最近的进展和新版的信息等等。
whatsnew	给出新版本中新命令的有关信息。
subscribe	使之变成一个MATLAB的预约用户。
why	解释为什么一些事情会出错。

MATLAB Help Desk是一个基于HTML的帮助系统。在这个系统里,通过 `help` 得到的大量信息是以超链接形式一起提供文献、图片和公式。它也含有一个比由 `lookfor` 提供的更好的查找工具。重要的是要注意 MATLAB Help Desk使用了Java Script, 因此, Java Script必须是使能的。如果给出 `helpdesk` 命令, 浏览器并不如图 2-3那样, 而是仅显示 MATLAB 标记, 这可能是Java Script并没有生效。在UNIX 中, 可以通过一个名为 `docopt.m` 的M文件决定浏览器的选择。在Windows中, MATLAB使用与 `.htm` 文件相关的程序。

### 例2.16

(a) 命令 `help size` 得到信息:

*SIZE* 矩阵的维数。

$D=SIZE(X)$ , 对于  $M \times N$  的矩阵  $X$ , 返回两个元素的行向量  $D=[M, N]$ , 其中包含了矩阵中行数和列数。对于  $N-D$  数组,  $SIZE(X)$  返回一个  $1 \times N$  数组长度的向量。

$[M, N]=SIZE(X)$  返回输出变量中的行数和列数。  $[M1, M2, M3, \dots, MN]=SIZE(X)$  返回  $X$  中头  $N$  个数组的长度。

$M=SIZE(X, DIM)$  返回由标量  $DIM$  指定的数组长度。例如,  $SIZE(X, 1)$  返回行的数量。

也可参见 `LENGTH`、`NDIMS`。

### 重载方法

#### Overloaded methods

`help zpk/size.m`

`help tf/size.m`

`help ss/size.m`

注意, 尽管MATLAB不接受大写字母给出的命令, 但在帮助文本中用大写字母书写命令。

(b) 为启动MATLAB Help Desk用命令 `helpdesk`, MATLAB启动如图2-3所示的一个浏览器(或使用一个已经运行的浏览器)。



图2-3 启动MATLAB Help Desk后显示的索引页

(c) 查找有关正弦函数信息，可键入：`lookfor sin` 则得到：

ACOS	反余弦
ACOSH	反双曲余弦
ASIN	反正弦
ASINH	反双曲正弦
COS	余弦
COSH	双曲余弦
SIN	正弦
SINH	双曲正弦
TFFUNC	调制的余弦高斯脉冲的时间域和频率域
DST	离散正弦变换
IDST	反离散正弦变换

学习MATLAB的一个很好的办法是用命令 `demo` 运行一个示范程序，然后用命令 `help` 检查插入的命令，最后开始使用此命令。

MATLAB有许多可以给出有关你正在运行MATLAB的计算机的信息。

#### 命令集19 计算机信息

<code>[str, n]= computer</code>	给出正在运行MATLAB的计算机的一个描述。字符串 <code>str</code> 因计算机或操作系统而定，例如，VAX、Sun、PC和Macintosh等。 <code>n</code> 是当前安装的MATLAB允许一个矩阵中元素的总数量。
<code>isieee</code>	对带IEEE算法的计算机，返回1，例如，IBMPC和Macintosh。对不带IEEE算法的计算机，返回0，例如VAX和Cray。

version	返回一个带当前MATLAB版本号的字符串。
ver	显示当前MATLAB和工具箱的版本号。
hostid	返回MATLAB服务器主机识别号。
getenv(str)	返回与str连接的文本。这里，str是一个符号或一个环境变量的名字。
terminal	设置MATLAB终端类型。

### 例2.17

在Sun工作站上运行Solaris 2，命令为：

```
[comp, numb]=computer
```

返回

```
comp =  
      SOL2  
numb =  
      268435455
```

## 2.8 保存和装载

MATLAB能保持屏幕上显示的日记，这是由命令 `diary` 完成的，图形输出则是例外。有关打印或保存图形，可参见第 13.7 节。

### 命令集20 会话日记

<code>diary filename</code>	在文件 <b>filename</b> 中保存下一个会话。
<code>diary off</code>	停止记录。
<code>diary on</code>	开始记录。继续在当前日记文件中。
<code>diary</code>	在文件 <b>diary</b> 中存储下一个会话，但也是在启停日记之间的一个转换器。

产生的ASCII文件以后能被编辑，且包括文档。可是，用命令 `diary` 保存的值和结果通常不能由MATLAB在下阶段阅读。

为了保存各种变量和它们的内容以便能在下阶段使用，应该使用命令 `save` 和 `load`。表中的文件名 **filename** 确定了MATLAB是如何说明这个文件的。所有以 `.mat` 结尾的文件是二进制文件，所有以其他形式结尾的文件，包括 **filename.**，都是ASCII文件。

### 命令集21 保存和装入变量

<code>save</code>	在文件 <b>matlab.mat</b> 中保存所有的变量
<code>save filename</code>	在文件 <b>filename.mat</b> 中保存所有的变量。如果在文件名后以点结束 <b>filename.</b> ，或加了另外一个后缀，那么MATLAB不能加后缀 <b>.mat</b> 。
<code>save filename v1 v2</code>	在文件 <b>filename.mat</b> 中保存变量 <b>v1</b> 、 <b>v2</b> 、...等。

```

save filename v -ascii 在文件filename.mat中以可读的ASCII格式保存变量
                        v的值，写8位小数。
save filename V -ascii 在文件filename.mat中以可读的ASCII格式、带16
-double              位小数的双精度保存变量v的值。
load                  从文件filename.mat中装入所有变量。
load filename         把文件filename.mat中所有变量装入MATLAB。如
                        果filename没有一个后缀，那么这个文件由save生
                        成。如果此文件有一个后缀，但只有两位，那么应
                        该使用选项-mat。
                        如果filename含有‘.’和后缀，例如temp.dat，那
                        么数据就从相应的ASCII文件加载到MATLAB中，
                        作为一个不带‘.’和后缀的名为filename的矩阵，
                        此文件可由save file var-ascii 或直接通过使用一个
                        编辑器来产生或作为另一个程序的输出文件。

```

### 例2.18

假设ASCII文件A.dat是通过一个编辑器或一个程序创建的，它含有下列数据：

```

1 4 5
4 2 9

```

在MATLAB中，可以通过输入下列命令而得到一个矩阵：

```

load A.dat, A
A =
     1     4     5
     4     2     9

```

当文件名是以一个字符串保存时，在这个文件上保存数据的例子可参见例 5.8(b)，也可参见第15章。在第15章中，讨论了许多高级文件的处理及如何在FORTRAN和C程序中使用文件。

## 2.9 命令文件和函数文件

为了代替在MATLAB提示符下输入MATLAB命令的语句，可以把这些命令写入一个文本文件，这个文本文件可用一个编辑器创建。每当用户输入这个文件名和它的自变量时，这些命令就由MATLAB执行。MATLAB从文件而不是从终端读取命令，当文件中最后一个命令被执行时，MATLAB能再从终端读取命令。MATLAB将首先在当前工作目录下寻找此文件，如果它不在当前目录下，那么在该路径下的所有目录中搜索。该路径保持在matlabpath中，可参见命令集22。如果想执行一个没有放在可以自动搜索处的一个文件，你可使用命令run，参见第12章。

M文件是一种文件：

```
filename.m
```

即，它必须有后缀.m。

一个M文件包含许多连续的MATLAB命令，它也可以引用其他的M文件，可以递归，也

就是说可以自己引用自己。

在MATLAB的实用盘“utility-disk”上有大量的预定义M文件，例如，**cond.m**、**demo.m**、**length.m**和**hilb.m**。要了解这些文件的名称，可以使用命令 **what**，列出由用户定义的和在MATLAB目录中存放的M文件。

命令**dir**可以代替**what**命令，这个命令属于MATLAB文件命令集。

## 命令集22 系统命令

<code>what dirname</code>	列出当前目录下所有的 MATLAB文件。如果给定 <b>dirname</b> ，就列出目录 <b>dirname</b> 下的文件。
<code>dir</code>	列出一个目录或子目录中的所有文件。这个命令可以用不同的路径名和程序单。
<code>ls</code>	以不同的输出格式列出文件。
<code>pwd</code>	列出当前的工作目录。
<code>delete filename</code>	删除文件 <b>filename</b> 。
<code>cd</code>	改变当前目录。
<code>type filename</code>	显示文件 <b>filename</b> 的内容。如果没有指定后缀，MATLAB就读 <b>filename.m</b> 。
<code>edit file</code>	打开一个编辑器。如果给定 <b>file</b> ，那么这个文件就在编辑器中打开。
<code>copyfile (file1, file2)</code>	<b>file1</b> 复制到 <b>file2</b> 。有关错误处理可参见 <b>help copyfile</b> 。
<code>which filename</code>	显示由 <b>filename</b> 指定的函数的搜索路径。
<code>path</code>	显示MATLAB的目录搜索路径。如果给出带自变量的命令，就改变搜索路径。输入 <b>help path</b> 可以获得更多的信息。
<code>matlabpath</code>	当一个新的搜索路径给定时，作为路径工作，但没有错误处理。
<code>genpath(directory)</code>	返回一个新的搜索路径，这个路径是由老的和在MATLABROOT/toolbox下所有路径一起组成。如果 <b>directory</b> 给定，那么在 <b>directory</b> 下所有的目录都被增加替换。
<code>pathsep</code>	列出分隔标志，这个标志分开了 <b>matlabpath</b> 中的所有目录。
<code>partialpath</code>	列出本地搜索路径。
<code>editpath</code>	给出一个图形用户界面。在那里，可以从 MATLAB的搜索路径增加和编辑目录。
<code>addpath(dir1, dir2, ..., flag)</code>	在MATLAB的搜索路径的开头增加目录 <b>dir1</b> 、 <b>dir2</b> 、...。如果字符串 <b>flag</b> 给定，且是始端，那么目录被加在始端；如果它是末端，则加在末端。
<code>rmpath dir</code>	从MATLAB的搜索路径中移去目录 <b>dir</b> 。
<code>pathtool</code>	这是一个修改搜索路径的图形工具。尽管 <b>help</b>



<code>path2rc</code>	pathtool建议它不要在 UNIX下工作，但它还是在 UNIX下工作。
<code>dbtype filename</code>	在文件 <b>pathdef.m</b> 中保存当前的搜索路径，当启动 MATLAB 时，可以从这个文件中读取搜索路径。
<code>dbtype filename r1:rn</code>	带行号显示文件 <b>filename</b> 的内容。如果在 <b>filename</b> 中没有给定后缀，MATLAB 使用后缀 <b>.m</b> 。
<code>lasterr</code>	带行号地显示 <b>filename</b> 中 <b>r1</b> 行到 <b>rn</b> 行。
<code>lastwarn</code>	重复上次的错误信息。
<code>!</code>	重复上次的警告信息。
<code>isstudent</code>	把这一行看作操作系统的命令，在 Macintosh 版本中不能使用。
<code>isdir dirname</code>	如果使用的是 MATLAB 学生版 (由 Prentice Hall 发布的)，返回 1；否则返回 0。
<code>isppc</code>	如果 <b>dirname</b> 是一个目录，返回 1；否则返回 0。
<code>isunix</code>	如果在 Macintosh Power PC 上使用 MATLAB，返回 1；否则返回 0。
<code>isvms</code>	如果在 UNIX 系统上使用 MATLAB，返回 1；否则返回 0。
<code>vms</code>	如果在 VMS 上使用 MATLAB，返回 1；否则返回 0。
<code>dos</code>	从 MATLAB 中运行一个 VMS DCL 命令，与 <b>!</b> 相同，但有可能从这个命令变量中存储数据。输入 <code>help vms</code> 可以获得更多的信息。这个命令仅在 VMS 系统上应用。
<code>applescript</code>	从 MATLAB 中运行一个 DOS 命令。如同 <b>!</b> ，见上一条，但有可能从这个命令变量中存储数据。输入 <code>help dos</code> 可以获得更多的信息。这个命令仅在 PC 上使用。
<code>unix</code>	加载一个 Apple Script 文件并运行它，参见 <code>help desk</code> 了解更多的信息。这个命令仅在 Macintosh 上使用。
<code>tempdir</code>	从 MATLAB 中执行一个 UNIX 操作系统命令，如同 <b>!</b> 。
<code>tempname</code>	输入 <code>help uni</code> 可以得到更多的信息。
<code>matlabroot</code>	返回一个表示系统中临时目录名的字符串
	返回一个以 'tp' 开头的字符串，MATLAB 将检查这个字符串是否为系统的临时目录中的一个文件名。因为这样，这个字符串可以用作一个临时文件的名字。
	返回带指向 MATLAB 安装所在目录的搜索路径的一个字符串。

创建一个名为 `matlab` 的目录以保存 M 文件是一个好的主意，MATLAB 在这个目录中自动地寻找和发现文件。

#### 例 2.19

(a) 查看包含在 MATLAB 中的文件 `sec.m` 的内容，输入：



```

type sec
function y=sec(z)
% SEC 正割
% SEC(X)是x的元素的正割。
% Copyright (c) 1984-98 by The MathWorks, Inc.
% $Revision: 5.3 $ $Date: 1997/11/21 23:28:33 $

```

```
y = 1./cos(z);
```

M文件中的注释，即以%开始的行，在 MATLAB中用作文档。这些注释应该是更好地提供信息，而不是象在这个例子中混淆 **X**和 $z$ 。基本运算符./在第3.5节中定义。

(b) 如果写入 `help sec` 就显示出文件 `sec.m` 的开始注释行：

```
SEC 正割。
```

```
SEC(X) 是x的元素的正割。
```

这是一个函数文件的例子，一个用户定义的函数，即是一个特殊类型的 M文件。除了第二类M文件外，称之为命令文件，一个函数能有一个或几个自变量或参数，它们用分号‘；’分隔。在例题sec中，有一个名为 $z$ 的参数。所有的参数必须由括号‘()’括起来。命令文件也称为script文件。输入 `help script` 将得到更多的有关信息。

MATLAB中的函数与C中的函数或FORTRAN的子程序非常相似。函数文件有如下特征：

- 函数文件的第一行必须包含字 `function`，命令文件没有这种要求。因此，没有这样第一行的M文件是命令文件。

- 第一行必须指定函数名、输入变量(参数)和输出变量(参数)。输入参数是从 MATLAB的工作空间复制到函数工作空间的变量。第一行举例如下：

```
function output = name(input)
```

- 一个函数可以有0个、一个或几个输入参数和返回值。

- 与FORTRAN中的子程序、Pascal中的过程或C/C++中的无返回值的函数等价。

```
function = name(inputArguments)
```

在MATLAB中，一个函数M文件可以这样调用：

```
filename(inputArguments)
```

建议函数取名如同文件名一样。调用时所用的变量并不需要与函数文件中定义的变量有相同的名字。

当输入 `help name` 后，就显示第一行之后的注释。这与命令文件一样使用相似的规定。

函数文件和命令文件的执行如同普通的 MATLAB命令。当输入文件名时(如果有自变量就一起附带参数，就执行文件中的语句。

所有的M文件都是普通的ASCII文件，都能通过文本编辑器创建。无论何时，如当编辑文本时，一个操作系统命令要被引用，这时从 MATLAB中使用命令！来输入和测试 M文件是很方便的，参见命令集22。

## 例2.20

(a) 假设某个矩阵经常被使用，它被创建并按第2.8节存储起来，然后随时可以装入。一个替换的方法就是在一个 M文件中创建该矩阵。下面的 MATLAB命令存放在文件 `Thematrix.m` 中，产生一个可以经常使用的矩阵：

```
A=[ -9   -3  -16; 13   7   16  3   3   10]
```

通过输入 `Thematrix`，矩阵 `A` 就根据上面的那一行进行分配。通过输入下面的命令就可以显示这个矩阵的情况：

```
whos
```

Name	Size	Bytes	Class
A	3x3	72	double array

Grand total is 9 elements using 72 bytes

(b) 假设下列函数是存放在文件 `average.m` 中：

```
function y=average(A)
```

% 本函数计算 `A` 中所有元素的平均值，其结果为一个标量。

```
[m, n]=size(A);
y=sum(sum(A))/m/n;
```

如果定义了例(a)中的矩阵，输入下面的命令：

```
average_value = average(A)
```

得到的结果是：

```
average_value =
    2.6667
```

书写函数 `average` 的另一个方法是：

```
function y=average(A)
```

% 本函数计算 `A` 中所有元素的平均值，其结果为一个标量

```
y=mean(A(:));
```

在第6章定义了命令 `sum` 和 `mean`，在第4.3节讲述了冒号的概念 `A(:)` 的含义。

(c) `startup.m` 是一个特殊的用户定义的命令文件。如果此文件放在你的 MATLAB 工作目录里，那么每当你启动 MATLAB 时，它就自动地运行。在这个文件里，你可以列出自己预先定义的内容和设置，如下例中，当你开始一个新奇的 MATLAB 工作时，MATLAB 向你致意。

```
% 我的第一个 startup.m
disp('Welcome to MATLAB!')
```

有了这样一个定义的命令文件，下次启动 MATLAB 时就可以看到：

```
< M A T L A B (R) >
(c) Copyright 1984-98 The MathWorks, Inc.
All Rights Reserved
Version 5.2.0.3084
Jan 17 1998
```

```
To get started, type one of these: helpwin, helpdesk, or demo.
For product information, type tour or visit www.mathworks.com.
```

```
Welcome to MATLAB!
>>
```

有关 M 文件的更多信息可参见第 12.3 节。