第1章　仿真软件——MATLAB

MATLAB 是由美国 MathWorks 公司发布的主要面对科学计算、可视化以及交互式程序设计的高科技计算环境。它将数值分析、矩阵计算、科学数据可视化以及非线性动态系统的建模和仿真等诸多强大功能集成在一个易于使用的视窗环境中,为科学研究、工程设计以及必须进行有效数值计算的众多科学领域提供了一种全面的解决方案,并在很大程度上摆脱了传统非交互式程序设计语言(如 C、Fortran)的编辑模式,代表了当今国际科学计算软件的先进水平。

1.1　MATLAB 的功能特点

MATLAB 是一种用于算法开发、数据可视化、数据分析以及数值计算的高级技术计算语言和交互式环境。使用它可以较使用传统的编程语言(如 C、C ++ 和 Fortran)更快地解决技术计算问题。它的应用范围非常广,包括工程计算、控制系统设计、信号和图像处理、通信、测试和测量、金融建模和分析以及计算生物学等众多应用领域。附加的工具箱扩展了 MATLAB 环境,以解决这些应用领域内特定类型的问题。

1. MATLAB 简介

在科学研究和工程应用中,为了克服一般语言对大量的数学运算,尤其当涉及矩阵运算时,编程难、调试麻烦等困难,美国 MathWorks 公司于 1967 年构思并开发了“Matrix Laboratory”(缩写 MATLAB,即矩阵实验室)软件包,经过不断更新和扩充,该公司于 1984 年推出了正式版的 MATLAB 1.0。特别是 1992 年推出了具有划时代意义的 MATLAB 4.0 版,并于 1993 年推出了其微机版,以配合当时日益流行的 Microsoft Windows 一起使用。到 2015 年为止先后推出了微机版的 MATLAB 4. x、MATLAB 5. x、MATLAB 6. x、MATLAB 7. x 和 MATLAB 8. x,使之应用范围越来越广。欲查看 MATLAB 版本更新一览表请扫描右边二维码。

用 MATLAB 编程运算与人进行科学计算的思路和表达方式完全一致,所以使用 MATLAB 进行数学运算就像在草稿纸上演算数学题一样方便,因此,在某种意义上说,MATLAB 既像一种万能的、科学的数学运算“演算纸”,又像一种万能的计算器一样方便快捷。MATLAB 大大降低了对使用者的数学基础和计算机语言知识的要求,即使用户不懂 C 或 FORTRAN 这样的程序设计语言,也可使用 MATLAB 轻易的再现 C 或 FORTRAN 语言几乎全部的功能,设计出功能强大、界面优美、稳定可靠的高质量程序来,而且编程效率和计算效率极高。

尽管 MATLAB 开始并不是为控制理论与系统的设计者们编写的,但以它“语言”化的数值计算、强大的矩阵处理及绘图功能、灵活的可扩充性和产业化的开发思路很快就为自动控制界研究人员所瞩目。目前,在自动控制、图像处理、语言处理、信号分析、振动理论、优化设计、时序分析、工程计算、生物医学工程和系统建模等领域,由著名专家与学者以 MATLAB 为基础开发的实用工具箱极大地丰富了 MATLAB 的内容。

MATLAB 包括拥有数百个内部函数的主包和几十种工具箱。工具箱又可以分为功能性工具箱和学科工具箱。功能工具箱用来扩充 MATLAB 的符号计算,可视化建模仿真,文字处理及实时

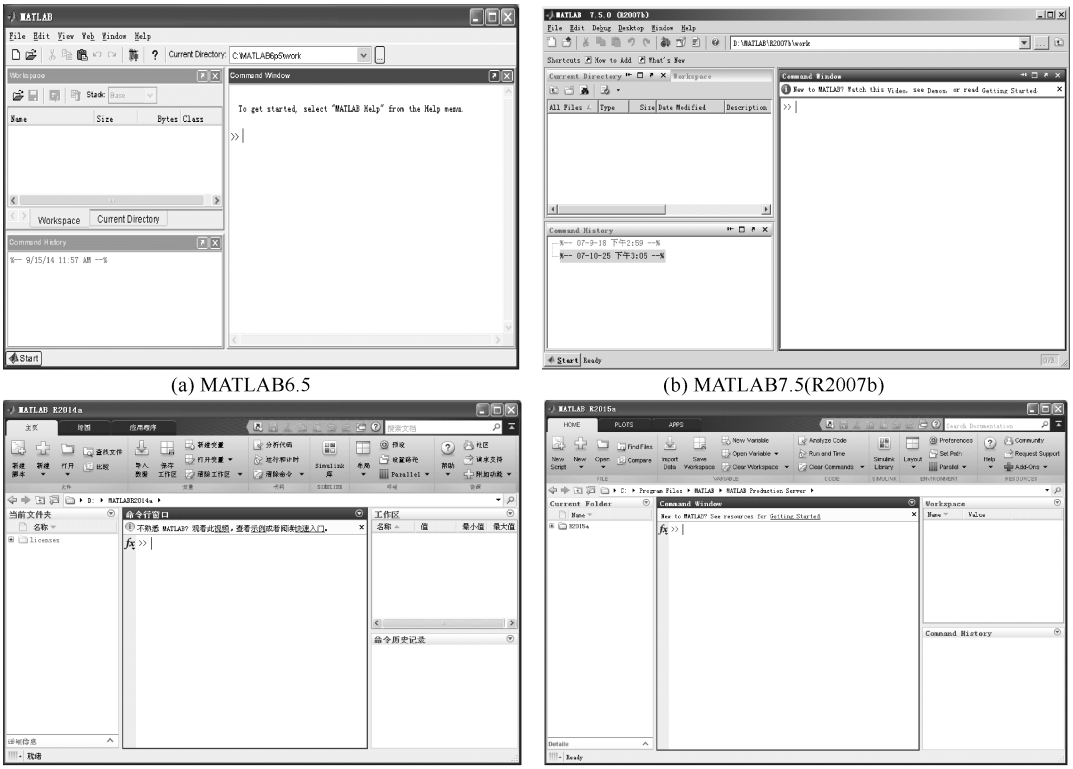
控制等功能。学科工具箱是专业性比较强的工具箱,如控制系统工具箱(Control System Toolbox),模糊逻辑工具箱(Fuzzy Logic Toolbox)和动态仿真工具箱(Simulink Toolbox)等。开放性使 MATLAB 广受用户欢迎,除内部函数外,所有 MATLAB 主包文件和各种工具箱都是可读可修改的文件,用户通过对源程序的修改或加入自己编写程序构造新的专用工具箱。较为常用的 MATLAB 工具箱可扫描右边二维码。

模型输入与仿真环境 Simulink 更使 MATLAB 为控制系统的仿真与 CAD 中的应用打开了崭新的局面,并使得 MATLAB 目前已经成为国际上最流行的控制系统计算机辅助设计的软件工具。MATLAB 不仅流行于控制界,在生物医学工程、语言处理、图像信号处理、雷达工程、信号分析、数学计算、金融统计和计算机技术等各行各业中都有极广泛的应用。

2. MATLAB 操作界面

一台计算机上可以同时安装多种 MATLAB 版本,各种版本之间相互独立运行互不干扰。使用 Windows XP 系统的用户需要安装 MATLAB 6.5 及以上的版本,否则不能正常使用。MATLAB 7.6 (R2008a)以上的版本基本都兼容 Windows 7 及以上系统。高版本的 MATLAB 同时支持 32 和 64 位操作系统,安装包 win32 和 win64 两个文件夹分别与之对应。

目前几种较为常用的 MATLAB 版本启动后的操作界面如图 1-1 所示。



(a) MATLAB6.5　(b) MATLAB7.5(R2007b)

(c) MATLAB8.3(R2014a)　(d) MATLAB8.5(R2015a)

图 1-1　MATLAB 操作界面

由图 1-1 可知,MATLAB 各种版本的操作界面略有不同。MATLAB 6.5 以前版本的操作界面通常由工作窗口、功能菜单和工具栏等组成。在 MATLAB 6.5 和 MATLAB 7.x 操作界面的左下角中新增加了开始(Start)按钮。而在 MATLAB 8.x 操作界面中,又新设置了主页(HOME)、绘图(PLOTS)和应用程序(APPS)等 3 个页面,同时取消了左下角的开始按钮并将其主要操作命令合并

到应用程序页面中。其中主页中包含一些常用的功能菜单和快捷按钮;绘图页面中包含所有绘图函数;应用程序页面包含常用工具箱中的各种交互操作界面命令,其更加方便、实用和灵活。

由于最新版的新增功能大多对于本课程涉及的内容没有太大影响,再加上最新版本安装程序大、启动和运行速度较慢。另外,尽管 MATLAB 新版本的内容和功能有所增加,但其使用方法基本同前。特别指出的是,MATLAB 8.3(R2014a)等虽已将主操作界面汉化,并支持中文,便于读者自学,但其大多子操作界面和子菜单仍为英文,且主要功能的使用方法仍同于 MATLAB 7.x。故本书以下仍以目前流行的经典版本 MATLAB 7.5(R2007b)为基础来进行叙述,但增加了新版本与以前版本有较大变化且涉及本课程内容的部分,使得本书所述内容对使用最新版本的用户仍可完全适用,同时也兼顾了当前仍在较低配置计算机上使用较低版本 MATLAB 6.5 的用户。

(1) MATLAB 的工作窗口

在默认状态下,MATLAB 的工作窗口由命令窗口(Command Window)、历史命令窗口(Command History)、工作空间浏览器窗口(Wotkspace)和当前工作目录窗口(Current Directory)等组成。

① 命令窗口(Command Window)

MATLAB 的命令窗口位于 MATLAB 操作界面的右方,它是 MATLAB 的主要操作窗口,MATLAB 的大部分命令和结果都需要在此窗口中进行操作和显示。

MATLAB 命令窗口中的“>>”标志为 MATLAB 的命令提示符,“|”标志为输入字符提示符。命令窗口中最上面的提示行是显示有关 MATLAB 的信息介绍和帮助等命令的。如果用户是第一次使用 MATLAB,则建议首先在命令提示符后键入 demo 命令,它将启动 MATLAB 的演示程序,用户可以在这些演示程序中领略到 MATLAB 所提供的强大的运算和绘图功能。

② 历史命令(Command History)窗口

在默认状态下,该命令窗口出现在 MATLAB 操作界面的左下方。这个窗口记录用户已经操作过的各种命令,用户可以对这些历史信息进行编辑、复制和剪切等操作。

③ 当前工作目录(Current Directory)窗口

在默认状态下,该窗口出现在 MATLAB 操作界面的左上方的前台。在这个窗口中,用户可以设置 MATLAB 的当前工作目录,并展示目录中的 M 文件等。同时,用户可以对这些 M 文件进行编辑等操作。

④ 工作空间(Workspace)浏览器窗口

在默认状态下,该窗口出现在 MATLAB 操作界面的左上方的后台。在这个窗口中,用户可以查看工作空间中所有变量的类别、名称和大小。用户可以在这个窗口中观察、编辑和提取这些变量。

(2) 开始按钮

开始按钮(Start)位于 MATLAB 操作界面的左下角,单击这个按钮后,会出现 MATLAB 的操作菜单。这个菜单上半部分的选项包含 MATLAB 的各种交互操作命令,下半部分的选项的主要功能是窗口设置、访问 MATLAB 公司的网页和查看帮助文件等。

(3) 功能菜单

为了更好地利用 MATLAB,在其操作界面中设置了以下多个功能菜单。

• File——文件操作菜单

New	新建 M 文件、图形、模型和图形用户界面
Open	打开 .m,.fig,.mat,.mdl,.cdr 等文件
Close Command Window	关闭命令窗口
Import Data	从其他文件导入数据
Save Workspace As	保存工作空间数据到相应的路径文件窗口中

Set Path	设置工作路径
Preferences	设置命令窗口的属性
Page Setup	页面设置
Print	设置打印机属性
Print Selection	选择打印
Exit MATLAB	退出 MATLAB 操作界面
• Edit——编辑菜单	
Undo	撤销上一步操作
Redo	重新执行上一步操作
Cut	剪切
Copy	复制
Paste	粘贴
Paste to Workspace…	粘贴到工作空间
Select All	全部选定
Delete	删除所选对象
Find	查找所需对象
Find Files	查找所需文件
Clear Command Window	清除命令窗口的内容
Clear Command History	清除历史窗口的内容
Clear Workspace	清除工作区的内容
• Debug——调试菜单	
Open M-Files when Debugging	调试时打开 M 文件
Step	单步调试
Step In	单步调试进入子函数
Step Out	单步调试跳出子函数
Continue	连续执行到下一断点
Clear Breakpoints in All Files	清除所有文件中的断点
Stop if Errors/Warnings	出错或报警时停止运行
Exit Debug Mode	退出调试模式
• Desktop——桌面菜单	
Unlock Command Window	命令窗口设为当前全屏活动窗口
Desktop Layout	桌面设计
Save Layout	保存桌面设计
Organize Layouts	组织桌面设计
Command Window	显示命令窗口
Command History	显示历史窗口
Current Directory	显示当前工作目录
Workspace	显示工作空间
Help	帮助窗口
Profiler	轮廓图窗口
Editor	编辑器

Figures	图形编辑器
Web Brower	Web 浏览器
Array Editor	矩阵编辑器
File Comparisons	文件比较
Toolbar	显示/隐藏工具栏
Shortcuts Toolbar	显示/隐藏快捷工具栏
Titles	显示/隐藏标题

• Window——窗口菜单

Close All Documents	关闭所有文档
Command Window	选定命令窗口为当前活动窗口
Command History	选定历史窗口为当前活动窗口
Current Directory	选定当前工作目录为当前活动窗口
Workspace	选定工作空间为当前活动窗口

(4) 工具栏

MATLAB 操作界面工具栏中的按钮“ ”分别用来快捷建立 M 文件编辑窗口和打开编辑文件窗口;按钮“ ” 对应的功能与 Windows 操作系统类似;按钮“ ”分别用来快捷启动 Simulink 库浏览窗口、GUIDE 模板窗口和轮廓图窗口;按钮“ ”分别用来快捷设置当前目录和返回到当前目录的父目录。

1.2 MATLAB 的基本操作

1.2.1 语言结构

MATLAB 命令窗口就是 MATLAB 语言的工作空间,因为 MATLAB 的各种功能的执行必须在此窗口下才能实现。所谓窗口命令,就是在上述命令窗口中输入的 MATLAB 语句,并直接执行它们完成相应的运算、绘图等。

MATLAB 语句的一般形式为

变量名 = 表达式

其中,等号右边的表达式可由操作符或其他字符、函数和变量名组成,它可以是 MATLAB 允许的数学或矩阵运算,也可以包含 MATLAB 下的函数调用;等号左边的变量名为 MATLAB 语句右边表达式的返回值语句所赋值的变量的名字。在调用函数时,MATLAB 允许一次返回多个结果,这时等号左边的变量名需用[ ]括起来,且各个变量名之间用逗号分隔开;等号和左边的变量名可以缺省,此时返回值自动赋给 ans。

MATLAB 中使用的算术运算符如表 1-1 所示。对于矩阵来说,这里左除和右除表示两种不同的除数矩阵和被除数矩阵的关系。对于标量,两种除法运算的结果相同,如 1/4 和 4\1 有相同的值 0.25。常用的十进制数符号如小数点、负号等,在 MATLAB 中也可以同样使用,表示 10 的幂次要用符号 e 或 E,如 3, -99,0.0001,1.6e-20,6.2e23。

在 MATLAB 中变量名必须以字母开头,之后可以是任意字母、数字或者下划线(不能超过 19 个字符),但变量中不能含有标点符号。变量名区分字母的大小写,同一名字的大写与小写被视为两个不同的变量。一般来说,在 MATLAB 下变量名可以为任意字符串,但 MATLAB 保留了一些特殊的字符串如表 1-2 所示。

表 1-1　MATLAB 中的算术运算符

算术运算符	意　义
+	加
-	减
*	乘
\	左除
/	右除
^	幂

表 1-2　MATLAB 中的特殊变量

特殊变量	取　值	特殊变量	取　值
ans	默认变量名	nargin	函数的输入变量数目
pi	圆周率(π = 3.1415926…)	nargout	函数的输出变量数目
i 或 j	基本虚数单位	realmin	系统所能表示的最小数值
inf 或 Inf	无限大,如 1/0	realmax	系统所能表示的最大数值
nan 或 NaN	不定量,如 0/0,∞/∞,0 * ∞	lasterr	存放最新的错误信息
eps	浮点相对精度	lastwarn	存放最新的警告信息

MATLAB 命令语句能即时执行,它不是输入完全部 MATLAB 命令语句并经过编译、连接形成可执行文件后才开始执行,而是每输入完一条命令,MATLAB 就立即对其处理,并得出中间结果,完成了 MATLAB 所有命令语句的输入,也就完成了它的执行,直接便可得到最终结果。从这一点来说,MATLAB 清晰地体现了类似“演算纸”的功能。例如,在 MATLAB 的命令窗口中直接输入以下命令:

```
>> a = 5;
>> b = 6;
>> c = a * b,
>> d = c + 2
```

其中由逗号结束的第 3 条命令和直接结束的第 4 条命令执行后,其结果分别显示如下

```
c =
    30
d =
    32
```

注意,以上各命令行中的“ >> ”标志为 MATLAB 的命令提示符,其后的内容才是用户输入的命令语句。每行命令输入完后,只有当用回车键进行确定后,命令才会被执行。

MATLAB 命令语句既可由分号结束,也可由逗号或直接结束,但它们的含义是不同的。如果用分号 “;”结束,则说明除了这一条命令外还有下一条命令等待输入,MATLAB 这时将不立即显示运行的中间结果,而等待下一条命令的输入,如上面例子的前两条命令;如果以逗号 “,”或直接结束,则将把左边返回的内容全部显示出来,如上面例子的后两条命令。当然在任何时候也可输入相应的变量名来查看其内容。例如:

```
>> a
```

结果显示:

```
a =
    5
```

在 MATLAB 中,几条命令语句也可出现在同一行中,但同一行中的两条命令间必须用分号或逗号将它们分割开来。例如:

```
>> a = 5;b = 6;c = a * b,d = c + 2
```

这时可得与上面相同的结果。

MATLAB 工作空间中的变量在退出 MATLAB 时会丢失。如果在退出 MATLAB 前想将工作空间中的变量保存到文件中,则可以调用 save 命令来完成,该命令的调用格式为

save 文件名 变量列表 其他选项

注意,这一命令中不能使用逗号,不同的元素之间只能用空格来分隔。例如,想把工作空间中

的 a,b,c 变量存到 mydat. mat 文件中去,则可用下面的命令来实现:

```
>> save mydat a b c
```

这里将自动地使用文件扩展名". mat"。如果想将整个工作空间中所有的变量全部存入该文件,则应采用下面的命令:

```
>> save mydat
```

当然这里的 mydat 也可省略,这时将工作空间中的所有变量自动地存入到文件 matlab. mat 中了。应该指出的是,这样存储的文件均是按照二进制数的形式进行的,所以得出的文件往往是不可读的。如果想按照 ASCII 码的格式来存储数据,则可以在命令后面加上一个控制参数:-ascii,该选项将变量以单精度的 ASCII 码形式存入文件中去;如果想获得高精度的数据,则可使用控制参数:-ascii -double。

MATLAB 提供的 load 命令可以从文件中把变量调出并重新装入到 MATLAB 的工作空间中去,该函数的调用格式与 save 命令相同。

当然工作空间中变量的保存和调出也可利用菜单项中的 File →Save Workspace As …和 File → Open 命令来完成;或利用 MATLAB 8. x 主页(HOME)页面中的快捷按键"  "和"  "来完成。

如果想查看目前的工作空间中都有哪些变量名,则可以使用 who 命令来完成。例如,当 MATLAB 的工作空间中有 a,b,c,d 四个变量名时,使用 who 命令将得出如下的结果:

```
>> who
   your variable are:
   a   b   c   d
```

想进一步了解这些变量的具体细节,则可以使用 whos 命令来查看。

了解了当前工作空间中的现有变量名之后,则可以使用 clear 命令来删除其中一些不再使用的变量名,这样可使得整个工作空间更简洁,节省一部分内存。例如,想删除工作空间中的 a,b 两个变量,则可以使用下面的命令:

```
>> clear a b
```

如果想删除整个工作空间中所有的变量,则可以使用以下命令:

```
>> clear
```

在 MATLAB 命令窗口中,利用上下方向键可以回调已输入的命令,向上和向下方向键"↑"和"↓"分别用于回调上一行和下一行命令。回调后的命令也可进行编辑等操作。

但仅靠一条一条地输入语句,MATLAB 难以实现复杂功能,为了实现诸如循环、条件、分支等功能,MATLAB 利用了 M 文件,M 文件由一系列的 MATLAB 语句组成。

MATLAB 实际上可以认为是一种解释性语言,用户可以在 MATLAB 工作环境下一条一条地键入命令,也可以直接键入用 MATLAB 语言编写的 M 文件名,或将它们结合起来使用。这样 MATLAB 软件对此命令或 M 文件中各条命令进行翻译,然后在 MATLAB 环境下对它进行处理,最后返回运算结果。所以说 MATLAB 语言的结构可用下式进行描述:

$$\text{MATLAB 语言} = \text{窗口命令} + \text{M 文件}$$

1.2.2 磁盘文件

因为 MATLAB 本身可以被认为是一种高效的语言,所以用它可编写出具有特殊意义的磁盘文件来。这些磁盘文件由一系列的 MATLAB 语句组成,它既可能是由一系列窗口命令语句构成的文本文件(也称为脚本文件,简称为 MATLAB 的程序),又可以是由各种控制语句和说明语句构成的函数文件(简称为 MATLAB 的函数)。由于它们都是由 ASCII 码构成的,其扩展名均为". m",故统

称为 M 文件。

由于 M 文件具有普通的文本格式,因而可以用任何编辑器建立和编辑。但一般最常用、而且最为方便的是使用 MATLAB 自带的编辑器,即利用 MATLAB 操作界面中的菜单命令 File→New→M-File 或 File→Open 打开的 M 文件编辑窗口对 M 文件进行建立和编辑。MATLAB 为了进一步方便用户对 M 文件的建立和编辑,在窗口中也设置了快捷工具“  ”和“  ”。

1. 文本文件

文本文件(脚本文件)由一系列的 MATLAB 语句组成,它类似于 DOS 下的批处理文件,在 MATLAB 命令窗口的提示符下直接键入文本文件名,回车后便可自动执行文本文件中的一系列命令,直至给出最终结果。文本文件在工作空间中运算的变量为全局变量。

【例 1-1】 利用 MATLAB 的文本文件,求以下方程

$$\begin{cases} y_1 = 3x_1 + x_2 + x_3 \\ y_2 = 3x_1 - x_2 - x_3 \end{cases}$$

在 $x_1 = -2, x_2 = 3, x_3 = 1$ 时的值。

解 ① 首先在 MATLAB 的操作界面中,利用菜单命令 File →New →M - File,打开 M 文件编辑器,然后在编辑器中根据例题中所给方程编写以下文本文件,并以 ex1_1_1 为文件名进行保存(后缀 . m 自动追加)。

```
% ex1_1_1. m
x1 = -2;x2 = 3;x3 = 1;
y1 = 3 * x1 + x2 + x3
y2 = 3 * x1 - x2 - x3
```

其中,带% 的语句为说明语句,不被 MATLAB 所执行,它可以在命令窗口中利用 help ex1_1_1 命令来显示% 后的内容。

② 当以上文本文件 ex1_1_1. m 建立后,在 MATLAB 命令窗口中输入

```
>> ex1_1_1
```

回车后显示:

```
y1 =
     -2
y2 =
    -10
```

由于文本文件中的变量为全局变量,故以上变量 x_1, x_2, x_3 的值,也可在文本文件外先给定,此时的文本文件为

```
% ex1_1_2. m
y1 = 3 * x1 + x2 + x3
y2 = 3 * x1 - x2 - x3
```

当以上文本文件 ex1_1_2. m 建立后,利用以下命令,同样可以得到以上结果。

```
>> x1 = -2;x2 = 3;x3 = 1;ex1_1_2
```

以上两种方式下,文本文件中变量的值都被保存下来,这与下面的函数文件是不同的。

对于 MATLAB 8. x,则利用其主页(HOME)中新建(New)菜单下的文本(Script)命令或主页(HOME)中新建文本(New Script)快捷工具“  ”打开 M 文本文件编辑窗口。

2. 函数文件

函数文件的功能是建立一个函数,且这个函数可以同 MATLAB 的库函数一样使用,它与文本文件不同,在一般情况下不能单独键入函数文件的文件名来运行一个函数文件,它必须由其他语句来调用,函数文件允许有多个输入参数和多个输出参数值。其基本格式如下:

function [f1,f2,f3,…] = fun(x,y,z,…)
注释说明语句
函数体语句

其中,x,y,z,…是形式输入参数;而 f1,f2,f3,…是返回的形式输出参数值;fun 是函数名。

实际上,函数名一般就是这个函数文件的磁盘文件名,注释语句段的内容同样可用 help 命令显示出来。

调用一个函数文件只需直接使用与这个函数一致的格式

[y1,y2,y3,…] = fun(a,b,c,…)

其中,a,b,c,…是相应的实际输入参数的值;而 y1,y2,y3,…是相应的实际输出参数的值。

【例 1-2】 利用 MATLAB 的函数文件,求以下方程

$$\begin{cases} y_1 = 3x_1 + x_2 + x_3 \\ y_2 = 3x_1 - x_2 - x_3 \end{cases}$$

在 $x_1 = -2$, $x_2 = 3$, $x_3 = 1$ 时的值。

解 ① 由于函数文件的建立与文本文件完全一样,故与例 1-1 一样首先根据例题中所给方程在 MATLAB 的 M 文件编辑器下,建立以下函数文件 ex1_2.m。

```
% ex1_2.m
function [b1,b2] = ex1_2(a1,a2,a3)
b1 = 3 * a1 + a2 + a3;
b2 = 3 * a1 - a2 - a3;
```

② 当以上函数文件 ex1_2.m 建立后,在 MATLAB 命令窗口中输入命令:

```
>> x1 = -2;x2 = 3;x3 = 1;[y1,y2] = ex1_2(x1,x2,x3)
```

结果显示:

```
y1 =
     -2
y2 =
    -10
```

对于 MATLAB 8.x,则利用其主页(HOME)中新建(New)菜单下的函数(function)命令,打开 M 函数文件编辑窗口,它与 M 文本文件编辑窗口略有不同,其区别在于已经设置其第一行由 function 开头,最后一行由 end 结尾的标准函数文件格式(end 也可删除不要)。

函数文件中定义的变量为局部变量,也就是说它只在函数内有效。即在该函数返回后,这些变量会自动在 MATLAB 工作空间中清除掉,这与文本文件是不同的,但可通过命令

global <变量>

来定义一个全局变量。

函数文件与文本文件另一个区别在于其第一行是由 function 开头的,且有函数名和输入形式参数与输出形式参数,没有这一行的磁盘文件就是文本文件。

1.2.3 库函数和数值显示格式

1. 库函数

为了方便用户,MATLAB 提供了丰富的库函数,库函数是根据系统已经编制好了的,提供用户直接使用的函数,MATLAB 中常用的基本数学函数,如表 1-3 所示。

表 1-3 MATLAB 的基本函数

函数名	含 义	函数名	含 义	函数名	含 义
sin( )	正弦函数	atan2( )	四象限反正切函数	sign( )	符号函数
cos( )	余弦函数	abs( )	绝对值或幅值函数	rand( )	随机数
tan( )	正切函数	sqrt( )	平方根	gamma( )	伽吗函数
asin( )	反正弦函数	exp( )	自然指数	angle(z)	复数 z 的相位函数
acos( )	反余弦函数	pow2( )	2 的指数	real(z)	复数 z 的实部
atan( )	反正切函数	log( )	以 e 为底的对数,即自然对数	imag(z)	复数 z 的虚部
sinh( )	双曲正弦函数	log2( )	以 2 为底的对数	conj(z)	复数 z 的共轭复数
cosh( )	双曲余弦函数	logl0( )	以 10 为底的对数	rat(x)	将实数 x 化为多项分数展开
tanh( )	双曲正切函数	floor( )	舍去正小数至最近整数	rats(x)	将实数 x 化为分数表示
asinh( )	反双曲正弦函数	ceil( )	加入正小数至最近整数	rem(x,y)	求 x 除以 y 的余数
acosh( )	反双曲余弦函数	round( )	四舍五入至最近整数	gcd(x,y)	整数 x 和 y 的最大公因数
atanh( )	反双曲正切函数	fix( )	舍去小数至最近整数	lcm(x,y)	整数 x 和 y 的最小公倍数

除了基本函数外,不同版本的 MATLAB 还增加了具有不同功能的库函数,也称工具箱或模块集。例如控制系统工具箱、模糊逻辑工具箱、神经网络工具箱和模型预测控制工具箱等。

对于各种函数的功能和调用方法可使用 MATLAB 的联机帮肋 help 来查询,例如:

```
>> help sin      % 得到正弦函数的使用信息;
>> help [        % 显示如何使用方括号。
```

2. 数值显示格式

尽管 MATLAB 计算中所有的量为双字长浮点数,但为了方便显示应遵循下面的规则。

在默认情况下,当结果为整数时,MATLAB 将它作为整数显示;当结果为实数时,MATLAB 以小数点后 4 位的精度近似显示;如果结果中的有效数字超出了这一范围,MATLAB 以科学计数法显示结果。MATLAB 可以使用 format 命令来改变显示格式,其调用格式为

format 控制参数

其中,控制参数决定显示格式,控制参数如表 1-4 表示。

表 1-4 format 命令的控制参数

控制参数	意 义	例 100/3
short	5 位有效数字,同默认显示	33.3333
long	长格式,15 位有效数字	33.33333333333334
short e	短格式,5 位有效数字的浮点数	3.3333e+001
long e	长格式,15 位有效数字的浮点数	3.333333333333334e+001
hex	十六进制格式	4040aaaaaaaaaab
bank	2 个十进制位	33.33
+	正、负或零	+
rat	有理格式	100/3

1.2.4 基本输入输出函数

MATLAB 的输入与输出函数包括命令窗口输入与输出及图形界面的输入与输出。除上面提到的用于机器间交换数据的函数语句 save 和 load 外,MATLAB 还允许计算机和用户之间进行数据

交换,允许对文件进行读写操作。如果用户想在计算的过程中给计算机输入一个参数,则可以使用input( )函数来进行,该函数的调用格式为

变量名 = input(提示信息,选项)

这里提示信息可以为一个字符串显示,它用来提示用户输入什么样的数据,input( )函数的返回值赋给等号左边的变量名。

例如,用户想输入 x 的值,则可以采用下面的命令来完成:

```
>> x = input(' Enter matrix x =>' )
```

执行该命令时首先给出 Enter matrix x => 提示信息,然后等待用户从键盘按 MATLAB 格式输入值,并把此值赋给 x。

如果在 input( )函数调用时采用了 's' 选项,则允许用户输入一个字符串,此时需用单引号将所输字符串括起来。

MATLAB 提供的命令窗口输出函数主要有 disp( )函数,其调用格式为

disp(变量名)

其中,变量名既可以是字符串,也可以是变量矩阵。例如

```
>> s = ' Hello World'
```

结果显示:

```
s =
    Hello World
>> disp(s)
```

结果显示:

```
    Hello World
```

可见用 disp( )函数显示的方式,和前面有所不同,它将不显示变量名字而其格式更紧密,且不留任何没有意义的空行。

MATLAB 提供了较实用的字符串处理及转换的函数,例如 int2str( )函数就可以方便地将一个整形数据转换成字符串形式,该函数的调用格式为

cstr = int2str(n)

其中,n 为一个整数,而该函数将返回一个相关的字符串 cstr。

例如 num 的数值为 num = 15,而在输出中还想给出其他说明性附加信息,则可利用语句:

```
>> num = 15;disp ([' The value of num is' ,int2str(num),' ! ok' ])
```

结果显示:

```
    The value of num is 15 ! ok
```

与 int2str( ) 函数的功能及调用方式相似,MATLAB 还提供了 num2str( )函数,可以将给出的实型数据转换成字符串的表达式,最终也可以将该字符串输出。例如给绘制的图形赋以数字的标题时可采用命令:

```
>> c = (70 - 32)/1.8;title([' Room temperature is' ,num2str(c),' degrees C' ])
```

则会在当前图形上加上题头标注:

```
    Room temperature is 21.1111 degrees C
```

1.2.5 外部程序调用

MATLAB 允许在其命令窗口中调用可执行文件,其调用方法是在 MATLAB 提示符下键入惊叹号“!”,后面直接跟该可执行文件即可。MATLAB 也允许采用这样的方式来直接使用 DOS 命令,如

磁盘复制命令 copy 可以由!copy 来直接使用,而文件列表命令 dir 可以由!dir 来调用。事实上,为了给用户提供更大的方便,MATLAB 已经把一些常用的 DOS 命令做成了相应的 MATLAB 命令,表 1-5 列出了 MATLAB 中提供的一些文件管理命令。

表 1-5　文件管理命令

命　令	注　释
what	列出当前目录下所有的 M 文件
dir	列当前目录下所有的文件
ls	与 dir 命令相同
type myfile	在命令窗口中显示文件 myfile. m 的内容
delete myfile	删除文件 myfile. m
cd path	进入子目录 path
which myfile	显示文件 myfile. m 所在的路径

当然由 C 或 FORTRAN 编译产生的可执行文件可采用上述方法直接调用,但此时 MATLAB 和该程序之间的数据传递是由读写文件的方式来完成的,这种调用格式虽然直观,但其缺点是速度相当慢;此外由于调用方式的原因,使用起来不是特别规范。故 MATLAB 还提供了对 C 或 FORTRAN 语言编写的程序的另一种调试方式,它是通过 MATLAB 提供的 MEX 功能来实现的。它由所调用的 C 或 FORTRAN 源码编译、连接而成 MEX 文件或 EXE 文件,这种可执行文件的速度较快,因为它和 MATLAB 之间的数据传递是通过指针来完成的,而不涉及对文件的读写,且其调用格式和 MATLAB 本身的函数调用格式完全一致,就如同这些子程序是 MATLAB 本身的程序一样。

1.3　MATLAB 的控制语句

MATLAB 是一个功能极强的高度集成化程序设计语言,它具备一般程序设计语言的基本语句结构,并且它的功能更强,由它编写出来的程序结构简单,可读性强。和其他高级语言一样,MATLAB 也提供了条件转移语句、循环语句等一些常用的控制语句,从而使得 MATLAB 语言的编程显得十分灵活。

1.3.1　循环语句

在实际计算中,经常会遇到许多有规律的重复计算,此时就要根据循环条件对某些语句重复执行。MATLAB 中包括两种循环语句:for 语句和 while 语句。

1. for 语句的基本格式

在 MATLAB 中,for 语句的基本命令格式为

```
for　循环变量 = 表达式 1:表达式 3:表达式 2
　　循环语句组
end
```

在 MATLAB 的循环语句基本格式中,循环变量可以取做任何 MATLAB 变量;表达式 1、表达式 2 和表达式 3 的定义和 C 语言相似,即首先将循环变量的初值赋成表达式 1 的值,然后判断循环变量的值,如果此时循环变量的值介于表达式 1 和表达式 2 的值之间,则执行循环体中的语句,否则结束循环语句的执行。执行完一次循环体中的语句之后,则会将循环变量自增一个表达式 3 的值,然后再判断循环变量是否介于表达式 1 和表达式 2 之间,如果满足就再执行循环体直至不满足为止,这时将结束循环语句的执行,而继续执行后面的语句。如果表达式 3 的值为 1,则可省略表达式 3。

【例 1-3】　求 $\sum_{i=1}^{100} i$ 的值。

解　MATLAB 程序 ex1_3_1. m 如下。

```
% ex1 _ 3_1. m
mysum = 0;
    for i = 1:100
        mysum = mysum + i;
    end
mysum
```

根据以上方法编写 MATLAT 的程序文件(M 文件)ex1_3_1. m,其运行结果显示

```
mysum  =
        5050
```

实际编程中,在 MATLAB 下采用循环语句会降低其执行速度,所以上面的程序可以由下面的命令来代替,以提高运行速度。

```
>> i = 1:100; mysum = sum(i)
```

其中,sum( )为内部函数,其作用是求出 i 向量的各个元素之和。

2. while 语句的基本结构

在 MATLAB 中,while 语句的基本命令格式为

```
while (条件式)
    循环体条件组
end
```

其执行方式为,若条件式中的条件成立,则执行循环体的内容;执行后再判断表达式是否仍然成立,如果表达式不成立,则跳出循环,向下继续执行。

例如,对于上面的例 1-3,如果改用 while 循环语句,则可以编写下面的程序:

```
% ex1 _ 3 _ 2. m
mysum  = 0; i = 1;
while (i < = 100)
    mysum  =  mysum  + i; i = i + 1;
end
mysum
```

MATLAB 提供的循环语句 for 和 while 是允许多级嵌套的,而且它们之间也允许相互嵌套,这和 C 语言等高级程序设计语言是一致的。

1.3.2 程序流控制语句

在程序设计语言中,经常会遇到提前中止循环、跳出子程序、显示执行过程等,此时就要用到以下控制程序流命令。

(1) echo 命令

一般来说当一个 M 文件运行时,文件中的命令不在屏幕上显示出来;而利用 echo 命令可以使 M 文件在运行时把其中的命令显示在工作空间中,这对于调试、演示等很有用。其命令格式为

```
echo on          % 显示其后所有执行的命令文件的指令;
echo off         % 不显示其后所有执行的命令文件的指令;
echo             % 在上述两种情况下进行切换;
```

```
echo filename on        % 显示由 filename 指定的 M 文件的执行命令;
echo filename off       % 不显示由 filename 指定的 M 文件的执行命令;
echo on all             % 显示其后的所有 M 文件的执行命令;
echo off all            % 不显示其后的所有 M 文件的执行命令。
```

(2) break 命令

在 MATLAB 中,break 命令经常与 for 或 while 等语句一起使用,其作用就是中止本次循环,跳出最内层的循环。使用 break 命令可以不必等到循环的自然结束,而是根据条件,遇到 break 命令后强行退出循环过程。

(3) continue 命令

在 MATLAB 中,continue 命令也经常与 for 或 while 等语句一起使用,其作用是结束本次循环,即跳过循环体下面尚未执行的命令,接着进行下一次是否执行循环的判断。

(4) pause 命令

pause()命令使用户暂停运行程序,当再按任一键时恢复执行。其中 pause(n)中的 n 为等待的秒数。

(5) return 命令

return 命令能使当前正在运行的函数正常退出,并返回调用它的函数,继续运行。

1.3.3 条件转移语句

在程序设计中,经常要根据一定的条件来执行不同的命令。当某些条件满足时,只执行其中的某个命令或某些命令。在 MATLAB 中,条件转移语句包括:if-else-end 语句和 switch-case-otherwise 语句。

1. if-else-end 语句的基本格式

在 MATLAB 中,最简单的条件结构,即 if-end 语句的命令格式为

```
if expression
    statements
end
```

当给出的条件式 expression 成立时,则执行该条件块结构中的语句内容 statements,执行完之后继续向下执行;若条件不成立,则跳出条件块而直接向下执行。

【例 1-4】 求满足 $\sum_{i=1}^{m} i > 1000$ 的最小 m 值。

解 MATLAB 程序 ex1_4.m 如下。

```
% ex1_4.m
mysum = 0;
for m = 1:1000
    mysum = mysum + m;
    if (mysum > 1000) break; end
end
m
```

运行结果显示:

```
m =
    45
```

MATLAB 还提供了其他两种条件结构：if-else-end 格式和 if-else if-end 格式，这两种格式的调用方法分别为

```
if expression
    statements1
else
    statements2
end
```

和

```
if expression1
    statements1
else if expression2
    statements2
else if expression3
    statements3
    ⋮
end
```

【例 1-5】 如果想对一个变量 x 自动赋值。当从键盘输入 y 或 Y 时(表示是)，x 自动赋为 1 值；当从键盘输入 n 或 N 时(表示否)，x 自动赋为 0 值；输入其他字符时中止程序。

解 要实现这样的功能，则可由下列的 while 循环程序来执行。

```
% ex1_5.m
ikey = 0;
while(ikey == 0)
  s1 = input('若给 x 赋值请输入[y/n]? ','s');
  if(s1 == 'y'|s1 == 'Y'),
     ikey = 1; x = 1
  else if (s1 == 'n'|s1 == 'N') ikey = 1; x = 0, end
     break
  end
end
```

2. switch-case-otherwise 语句的基本格式

MATLAB 中 switch-case-otherwise 语句的调用格式为

```
switch switch-expression
case case-expression1
        statements1;
case case-expression2
        statements2;
case case-expression3
        statements3;
        ⋮
otherwise
        statementsn;
end
```

switch-case-otherwise 语句中,switch-expression 给出了开关条件:当有 case-expression 与之匹配时,就执行其后的语句;如果没有 case-expression 与之匹配,就执行 otherwise 后面的语句。在执行过程中,只有一个 case 命令被执行。当执行完命令后,程序就跳出分支结构,执行 end 后面的命令。

例如,对于以下 MATLAB 函数文件 myfun. m。

```
function f = myfun( n)
switch n
case 0
      f = 1;
case 1
      f = 2;
otherwise
      f = 8;
end
```

在 MATLAB 命令窗口中输入以下命令:

```
>> y = myfun( 1)
```

结果显示:

```
y  =
    2
```

1.4 MATLAB 的绘图功能

MATLAB 被控制界广泛接受的另一个重要原因是,它提供了十分方便的一系列绘图命令。例如,线性坐标、对数坐标、半对数坐标及极坐标等命令,它还允许用户同时打开若干图形窗口,对图形标注文字说明等,它使得图形绘制和处理等复杂工作变得简单得令人难以置信。

1.4.1 二维图形

1. 基本形式

MATLAB 最基本的绘图函数为 plot( )。

如果 y 是一个 n 维行向量或列向量,那么 plot(y)将绘制一个 y 元素和 y 元素排列序号 1,2,…,n 之间关系的线性坐标图。如 y 是一个 $n \times m$ 维矩阵,那么 plot(y)将同时绘出每列元素与其排列序号 1,2,…,n 之间关系的 m 条曲线。例如

```
>> y = [0 0. 48 0. 84 1 0. 91 0. 6 0. 14]; plot( y)
```

则显示如图 1-2 所示的简单曲线。

如果 x 和 y 是两个等长向量,那么 plot(x,y) 将绘制一条 x 和 y 之间关系的线性坐标图。例如,利用以下命令可显示如图 1-3 所示正弦曲线。

```
>> x = 0:0. 01:2 * pi;y = sin( x) ;plot( x,y)
```

2. 多重线型

在同一图形中可以绘制多重线型,其基本命令格式为

plot(x1,y1,x2,y2,…,xn,yn)

其中,向量 x1,x2,…,xn 为 x 轴变量;向量 y1,y2,…,yn 为 y 轴变量。

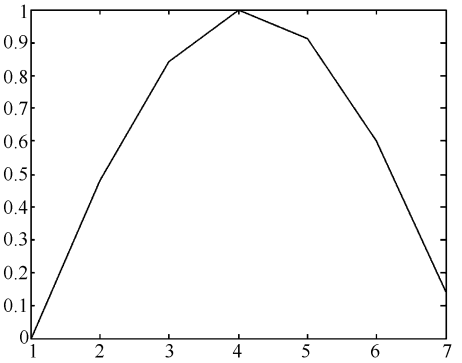


图 1-2　简单曲线

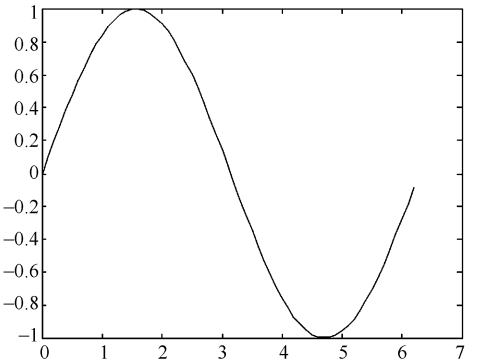


图 1-3　正弦曲线

以上命令可将 x1 对 y1,x2 对 y2,…,xn 对 yn 的图形绘制在一个图形中,而且分别采用不同的颜色或线型。例如,利用以下命令可显示如图 1-4 所示的正余弦曲线。

```
>> x = 0:0.1:2 * pi;plot(x,sin(x),x,cos(x))
```

当 plot( )命令作用于复数数据时,通常虚部是忽略的。然而有一个特殊情况,即当 plot( )只作用于单个复变量 z 时,则实际绘出实部对应于虚部的关系图形(复平面上的一个点)。即这时 plot(z) 等价于 plot(real(z),image(z)),其中 z 为矩阵中的一个复向量。

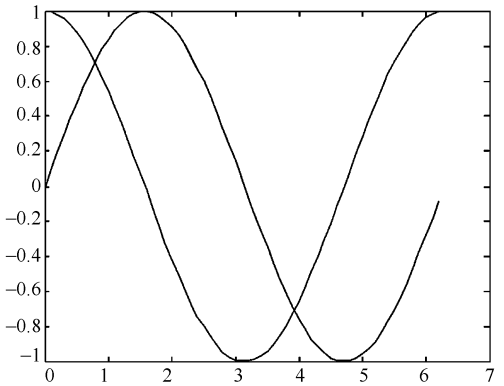


图 1-4　正余弦曲线

表 1-6　MATLAB 中绘图命令的各种选项

选项	意　义	选项	意义
-	实线	b	蓝
--	虚线	c	青色
—.	点划线	g	绿
:	点线	r	红
.	用点号绘制各数据点	k	黑
×	叉号线	m	洋红色
。	圆圈线	w	白
*	星号线	y	黄色

3. 图形修饰及文本标注

MATLAB 中对于同一图形中的多重线,不仅可分别定义其线型,而且可分别选择其颜色,带有选项的曲线绘制命令的调用格式为

plot(x1, y1,选项 1,x2, y2,选项 2,…, xn, yn,选项 n)

其中,向量 x1,x2,…,xn 为 x 轴变量;向量 y1, y2,…, yn 为 y 轴变量。选项如表 1-6 所示。

表 1-6 中的线型和颜色选项可以同时使用。例如

```
>> x = 0:0.1:2 * pi; plot(x,sin(x),'-g', x,cos(x),'-.r')
```

绘制完曲线后,MATLAB 还允许用户使用它提供的特殊绘图函数来对屏幕上已有的图形加注释、题头或坐标网格。例如

```
>> x = 0:0.1:2 * pi;y = sin(x);plot(x,y)
>> title('Figure Example')          % 给出题头
>> xlabel('This is x axis')         % x 轴的标注
>> ylabel('This is y axis')         % y 轴的标注
```

>> grid　　% 增加网格

除了在标准位置书写标题和轴标注以外，MATLAB 还允许在图形窗口的位置利用 line( ) 和 text( )命令画直线或写字符串，它们的调用格式分别为

line(x, y)　和　text(x, y,chstr,选项)

其中，line( )函数在给定的图形窗口上绘制一条由向量 x 和 y 定义的折线；text( )函数是在指定的点(x,y)处写一个 chstr 绘出的字符串；而选项决定 x,y 坐标的单位，如选项为‘sc’，则 x,y 表示规范化窗口的相对坐标，其范围为 0 到 1，即左下角坐标为(0,0)，而右上角的坐标为(1,1)。如省略选项，则 x,y 坐标的单位和图中是一致的。例如

>> text(2.5,0.7,′sin(x)′)

用 text( )命令可以在图形中的任意位置加上文本说明，但是必须知道其位置坐标；而利用另一个函数 gtext( )，则可以用鼠标来对要添加的文本字符串定位。在 MATLAB 的工作空间中键入下列命令：

>> gtext(′sin(x)′)

则在图中将出现一个十字叉，用鼠标将它移动到添加文本的位置，单击鼠标，gtext(′sin(x)′)命令中的文本字符串“sin(x)”就自动添加到指定的位置。

4. 图形控制

MATLAB 允许将一个图形窗口分割成 $n \times m$ 个部分，对每一部分可以用不同的坐标系单独绘制图形。窗口分割命令的调用格式为

subplot(n,m,k)

其中，n 和 m 分别表示将这个图形窗口分割的行和列数；k 表示每一部分的代号。例如，想将窗口分割成 4×3 个部分，则右下角的代号为 12，MATLAB 最多允许 9×9 的分割。

尽管 MATLAB 可以自动根据要绘制曲线数据的范围来选择合适的坐标系，使得曲线能够尽可能清晰地显示出来，但是，如果觉得自动选择的坐标仍不合适，则可以用手动的方式来选择新的坐标系。调用函数的格式为

axis([xmin, xmax, ymin, ymax])

另外，MATLAB 还提供了清除图形窗口命令 clf、保持当前窗口的图形命令 hold、放大和缩小窗口命令 zoom 等。

5. 特殊坐标图形

除了基本的绘图命令 plot( )外，MATLAB 还允许绘制极坐标曲线、对数坐标曲线、条形图和阶梯图等，其常用的函数如表 1-7 所示。

表 1-7　特殊二维曲线绘制函数

函数名	意　义	常用调用格式	函数名	意　义	常用调用格式
polar( )	极坐标图	polar(x,y)	comet( )	彗星状轨迹图	comet(x,y)
semilogx( )	x－半对数图	semilogx(x,y)	quiver( )	向量场图	quiver(x,y)
semilogy( )	y－半对数图	semilogy(x,y)	feather( )	羽毛图	feather(x,y)
loglog( )	对数图	logog(x,y)	compass( )	罗盘图	compass(x,y)
stairs( )	阶梯图	stairs(x,y)	stem( )	针图	stem(x,y)
bar( )	长条形图	bar(x,y)	fill( )	实心图	fill(x,y,c)
errorbar( )	误差限图	errorbar(x,y,ym,yM)	hist( )	累计图	hist(y,n)

表 1-7 中参数 x,y 分别表示横、纵坐标绘图数据；c 表示颜色选项；ym,yM 表示误差图的上下