

Matlab与信号处理

周治国

2018.9

第二章

Matlab基础知识

序言

致力于加快工程技术和科学研究的步伐——这句话总结了MathWorks坚持超过三十年的使命。

在这期间，MathWorks有幸见证了工程师和科学家使用MATLAB和Simulink在多个应用领域中的无数变革和突破：汽车行业的电气化和不断提高的自动化；日益精确的气象建模和预测；航空航天领域持续提高的性能和安全指标；由神经学家破解的大脑和身体奥秘；无线通信技术的普及；电力网络的可靠性，等等。

与此同时，MATLAB和Simulink也帮助了无数大学生在工程技术和科学研究课程里学习关键的技术理念并应用于实际问题中，培养他们成为栋梁之才，更好地投入科研、教学以及工业应用中，指引他们致力于学习、探索先进的技术，融合并应用于创新实践中。

如今，工程技术和科研创新的步伐令人惊叹。创新进程以大量的数据为驱动，结合相应的计算硬件和用于提取信息的机器学习算法。软件和算法几乎无处不在——从孩子的玩具到家用设备，从机器人和制造体系到每一种运输方式——让这些系统更具功能性、灵活性、自主性。最重要的是，工程师和科学家推动了这些进程，他们洞悉问题，创造技术，设计革新系统。

为了支持创新的步伐，MATLAB发展成为一个广泛而统一的计算技术平台，将成熟的技术方法（比如控制设计和信号处理）融入令人激动的新兴领域，例如深度学习、机器人、物联网开发等。对于现在的智能连接系统，Simulink平台可以让您实现模拟系统，优化设计，并自动生成嵌入式代码。

“科学与工程计算技术丛书”系列主题反映了MATLAB和Simulink汇集的领域——大规模编程、机器学习、科学计算、机器人等。我们高兴地看到“科学与工程计算技术丛书”支持MathWorks一直以来追求的目标：助您加速工程技术和科学研究。

期待着您的创新！

目录

- 1、Matlab概述
- 2、Matlab工作环境
- 3、Matlab程序控制结构
- 4、变量、数值与表达式
- 5、数组与矩阵
- 6、数据分析
- 7、图形绘制

1、Matlab概述

MATLAB的名字是由Matrix和Laboratory两词的前三个字母组合而成。在国际上三十几个数学类科技应用软件中，MATLAB在数值计算方面独占鳌头。

• Matlab的发展历史

- MATLAB的由来是20世纪70年代后期的事情：时任美国新墨西哥大学计算机科学系主任的Cleve Moler教授出于减轻学生编程负担的动机，为学生设计了一组调用LINPACK和EISPACK库程序的“通俗易懂”的接口，此即用FORTRAN编写的萌芽状态的MATLAB。
- 经几年的校际流传，在Little的推动下，由Little、Moler、Steve Bangert合作，于1984年成立了MathWorks公司，并把MATLAB正式推向市场。从这时起，MATLAB的内核采用C语言编写，而且除原有的数值计算能力外，还新增了数据图视功能。

1、Matlab概述

- Matlab的发展历史

- MATLAB以商品形式出现后，仅短短几年，就以其良好的开放性和运行的可靠性，使原先控制领域里的封闭式软件包（如英国的UMIST，瑞典的LUND和SIMNON，德国的KEDDC）纷纷淘汰，而改以MATLAB为平台加以重建。在时间进入20世纪九十年代的时候，MATLAB已经成为国际控制界公认的标准计算软件。
- 从2006年开始 MATLAB 分别在每年的3月和9月进行两次产品发布，每次发布都涵盖了产品家族中的所有模块，包含已有产品新特性和bug修订，以及新产品的发布。其中3月发布的产品称为“a”，9月发布的产品称为“b”。

1、Matlab概述

• Matlab的组成与功能

- MATLAB是由美国Math Works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。
- MATLAB系统主要包括五个部分：桌面工具和开发环境，数字函数库、语言、图形处理、外部接口。其中桌面工具包括MATLAB桌面和命令窗口、编辑器和调试器、代码分析器和浏览帮助、工作空间、文件的浏览器。
- MATLAB的函数库包括大量的算法，从初等函数到复杂的高等函数。MATLAB语言是一种高级的基于矩阵和数组的语言，具有程序流控制、函数、数据结构、输入输出和面向编程等特色。在图形处理中，MATLAB具有方便的数据可视化功能，同时，MATLAB语言有能够和一些高级语言进行交互的函数库。

2、Matlab工作环境

- 命令行窗口
- 帮助系统窗口
- 图形窗口
- M文件编辑窗口
- 当前文件夹
- 搜索路径

2、Matlab工作环境

- 命令行窗口

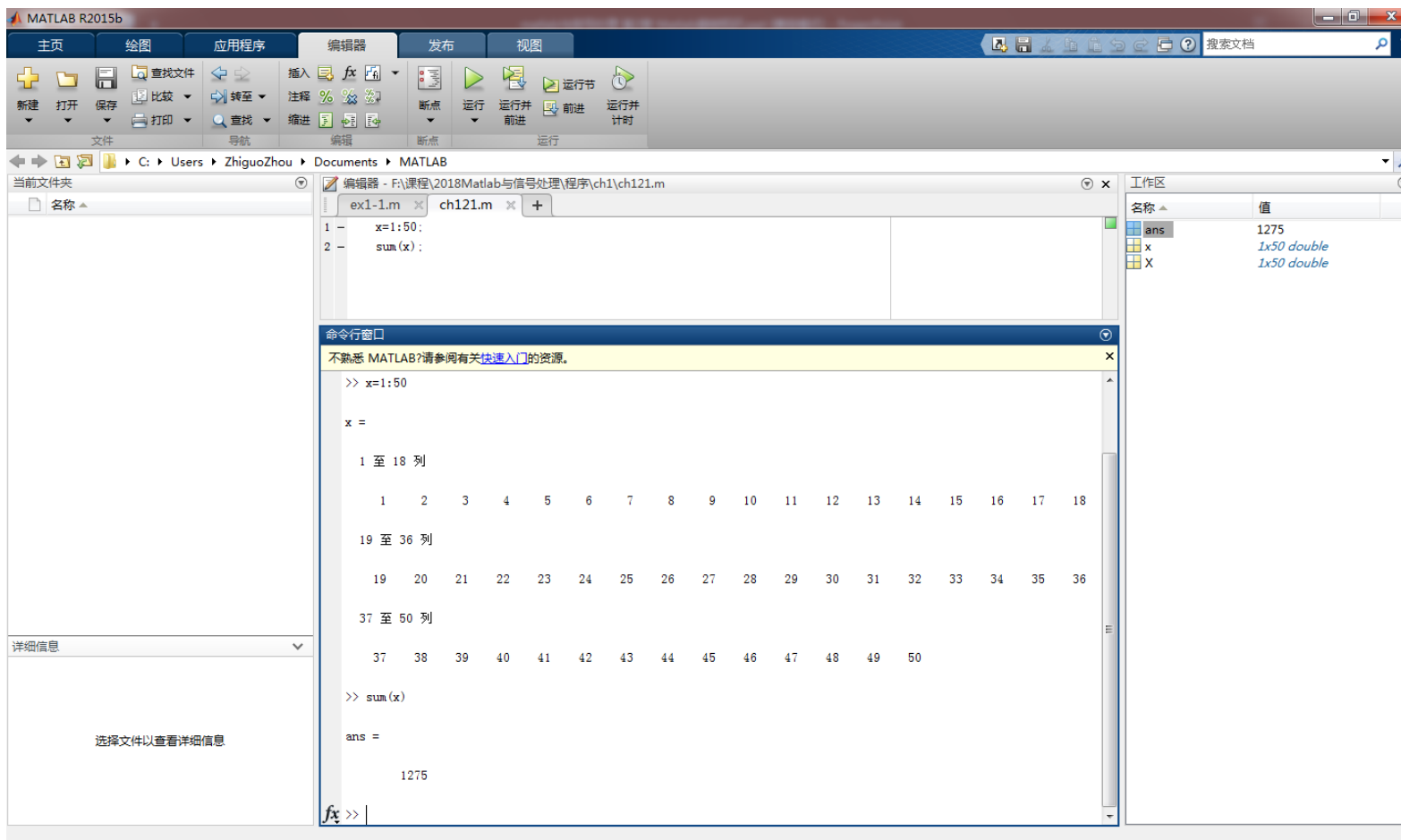
1.2.1 命令行窗口

MATLAB各种操作命令都是从命令行窗口开始，用户可以在命令行窗口中输入MATLAB命令，实现其相应的功能。启动MATLAB，单击MATLAB图标，进入到用户界面，此命令行窗口主要包括文本的编辑区域和菜单栏，如图1-1所示。

2、Matlab工作环境

- 命令行窗口

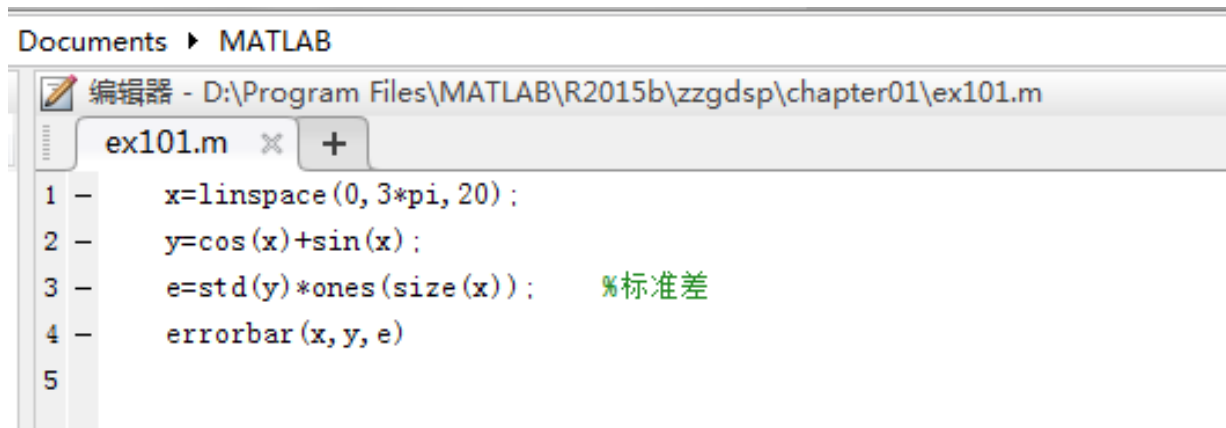
ex101.m



2、Matlab工作环境

- 命令行窗口

ex101.m



```
Documents > MATLAB
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex101.m
ex101.m x +
1 - x=linspace(0, 3*pi, 20);
2 - y=cos(x)+sin(x);
3 - e=std(y)*ones(size(x)); %标准差
4 - errorbar(x,y,e)
5
```

表1-1 命令行窗口常用的命令功能

命令	功能
clc	删去一页命令行窗口，光标回屏幕左上角
clear	从工作空间清除所有变量
clf	清除图形窗口内容
who	列出当前工作空间中的变量
whos	列出当前工作空间中的变量及信息或用工具栏上的Workspace浏览器
delete	从磁盘删除指定文件
which	查找指定文件的路径
clear all	从工作空间清除所有变量和函数
help	查询所列命令的帮助信息
save name	保存工作空间变量到文件name.mat
save name x y	保存工作空间变量x y到文件name.mat
load name	加载name文件中的所有变量到工作空间
load name x y	加载name文件中的变量x y到工作空间
diary name1.m	保存工作空间一段文本到文件name1.m
diary off	关闭日志功能
type name.m	在工作空间查看name.m文件内容
what	列出当前目录下的m文件和mat文件

2、Matlab工作环境

- 帮助系统窗口

1.2.2 帮助系统窗口

有效地使用帮助系统所提供的信息，是用户掌握好MATLAB应用的最佳途径。MATLAB的帮助系统分为联机帮助系统和命令行窗口查询帮助系统。

常用的帮助信息有：`help`，`demo`，`doc`，`who`，`whos`，`what`，`which`，`lookfor`，`helpbrowser`，`helpdesk`，`exit`，`web`等。例如：在窗口中输入“`help fft`”就可以获得函数“`fft`”的帮助信息。

2、Matlab工作环境

- 帮助系统窗口

命令行窗口

```
>> help fft
```

fft Discrete Fourier transform.

fft(X) is the discrete Fourier transform (DFI) of vector X. For matrices, the **fft** operation is applied to each column. For N-D arrays, the **fft** operation operates on the first non-singleton dimension.

fft(X,N) is the N-point **fft**, padded with zeros if X has less than N points and truncated if it has more.

fft(X,[],DIM) or **fft(X,N,DIM)** applies the **fft** operation across the dimension DIM.

For length N input vector x, the DFI is a length N vector X, with elements

$$X(k) = \sum_{n=1}^N x(n) \exp(-j*2*\pi*(k-1)*(n-1)/N), \quad 1 \leq k \leq N.$$

The inverse DFI (computed by IFFI) is given by

$$x(n) = (1/N) \sum_{k=1}^N X(k) \exp(j*2*\pi*(k-1)*(n-1)/N), \quad 1 \leq n \leq N.$$

See also [fft2](#), [fftn](#), [fftshift](#), [fftw](#), [ifft](#), [ifft2](#), [ifftn](#).

[fft 的参考页](#)

[名为 fft 的其他函数](#)

2、Matlab工作环境

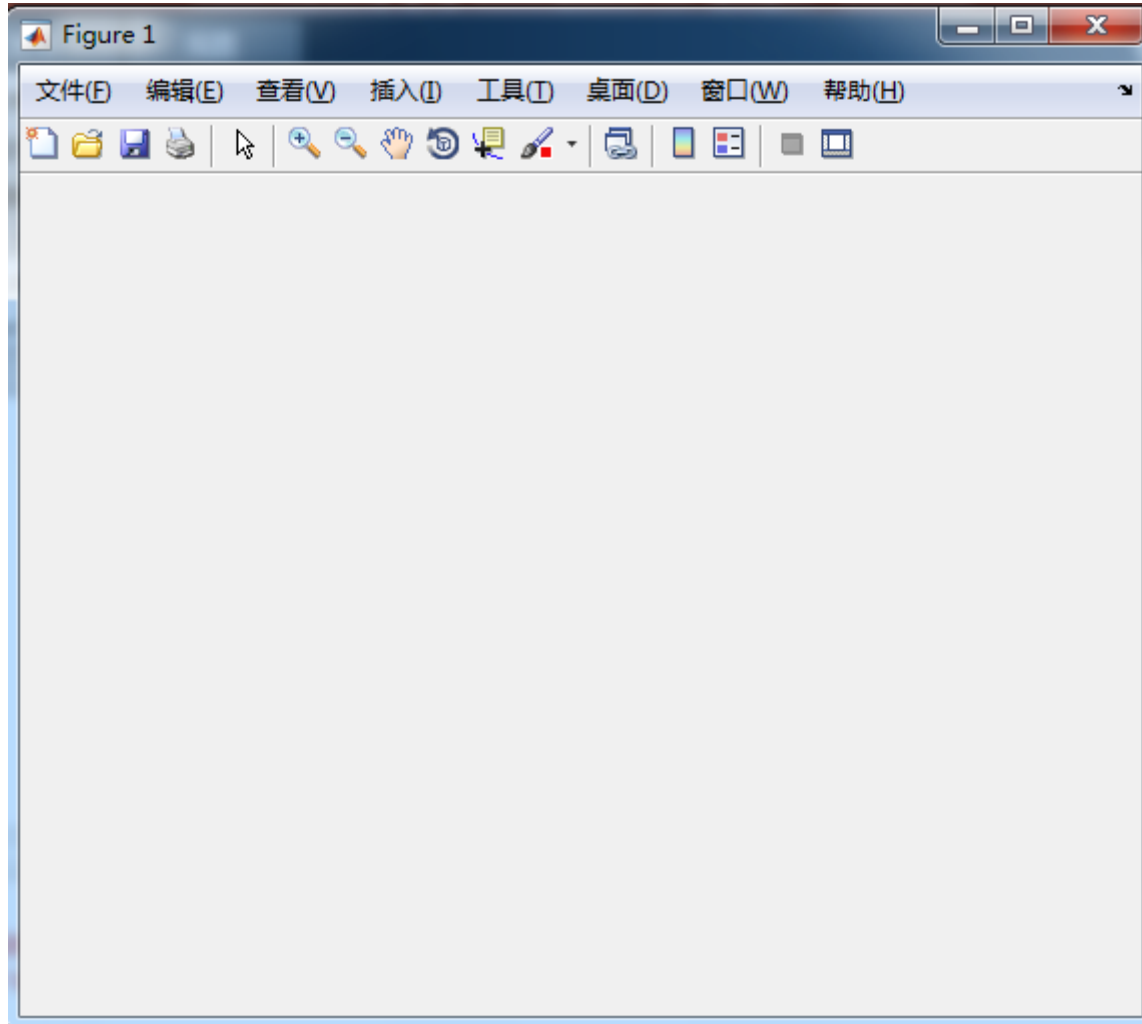
- 图形窗口

1.2.3 图形窗口

图形窗口用来显示MATLAB所绘制的图形，这些图形可以是二维图形，也可以是三维图形。用户可以通过选择“新建”→“图形”命令进入图形窗口，如图1-5所示。也可以通过运行程序自动弹出图形窗口，如图1-6所示。

2、Matlab工作环境

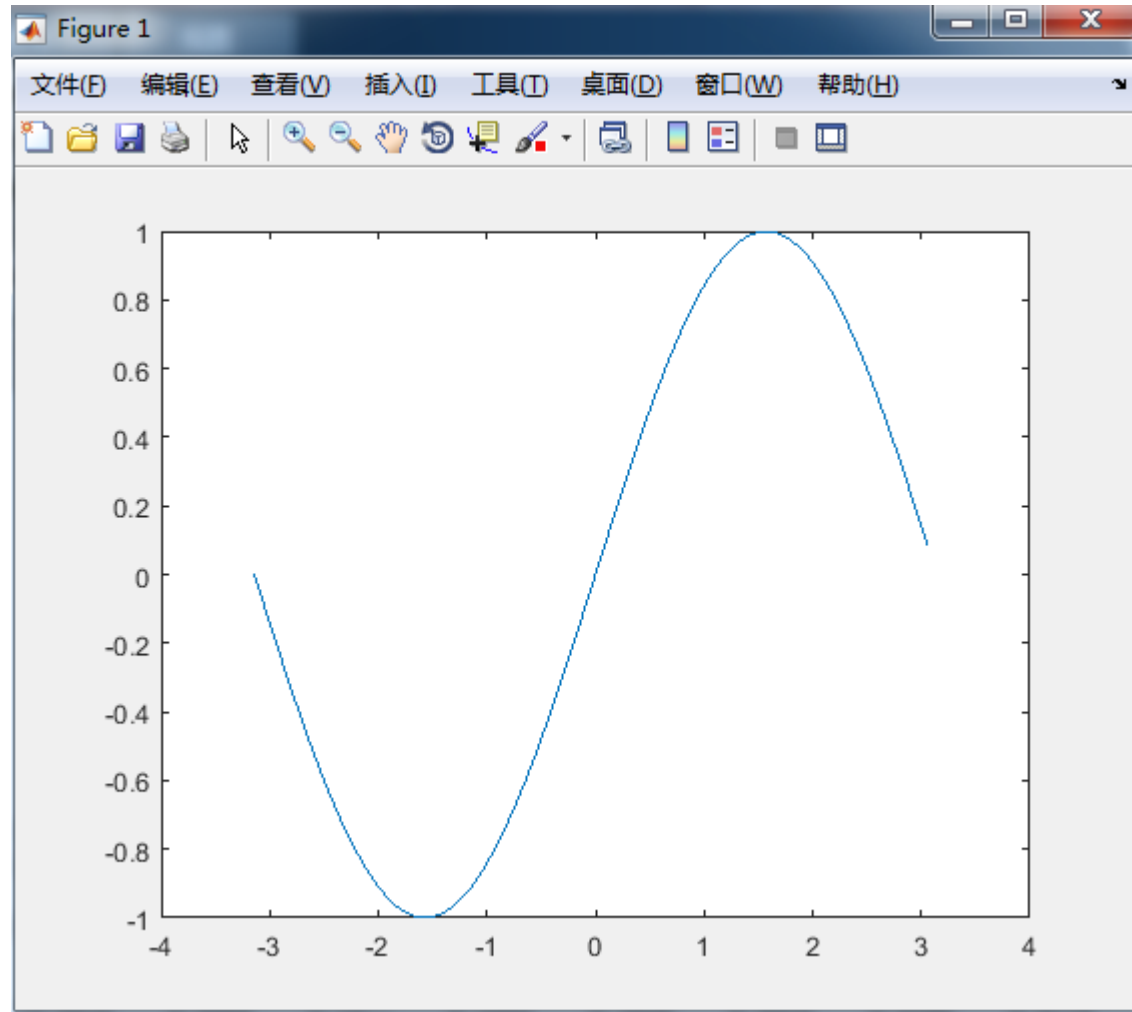
- 图形窗口



2、Matlab工作环境

- 图形窗口

```
x=-pi:0.1:pi  
y=sin(x)  
plot(x,y)
```



2、Matlab工作环境

- M文件编辑窗口

1.2.4 M文件编辑窗口

MATLAB对命令文件的执行等同于从命令窗口中顺序执行文件中的所有指令。命令文件可以访问MATLAB工作空间里的任何变量和数据。

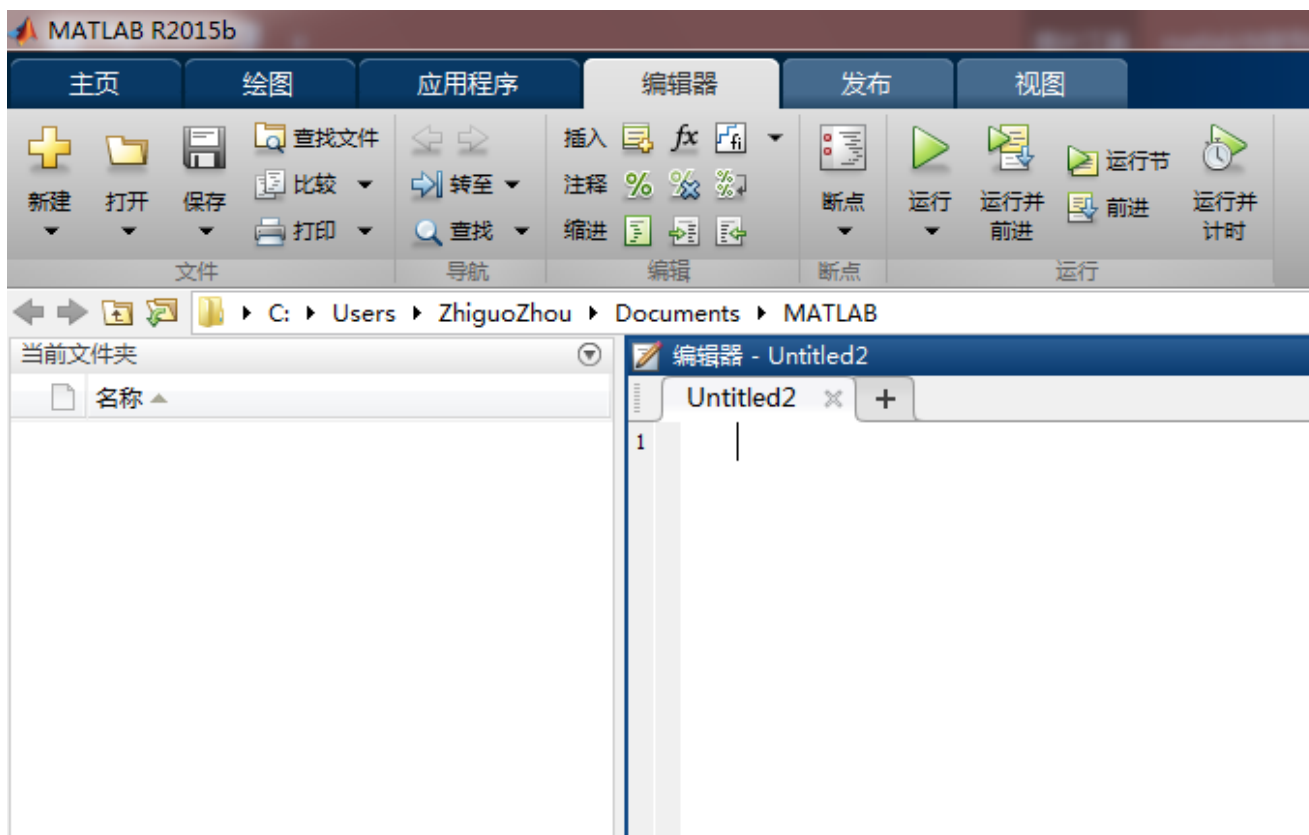
命令文件运行过程中产生的所有变量都等同于从MATLAB工作空间中创建这些变量。因此，任何其他命令文件和函数都可以自由地访问这些变量。这些变量一旦产生就一直保存在内存中，只有对它们重新赋值，它们的原有值才会变化。关机后，这里变量全部消失。另外，在命令窗口中运行Clear命令，也可以把这些变量从工作空间中删除。当然，在MATLAB的工作空间窗口中也可以用鼠标选择想要删除的变量，将这些变量从工作空间中删除。

M文件的类型是普通的文本文件，我们可以使用系统认可的文本文件编辑器来建立M文件。如DOS下的Edit，Windows的记事本和Word等。M文件主要功能如下：

2、Matlab工作环境

- M文件编辑窗口

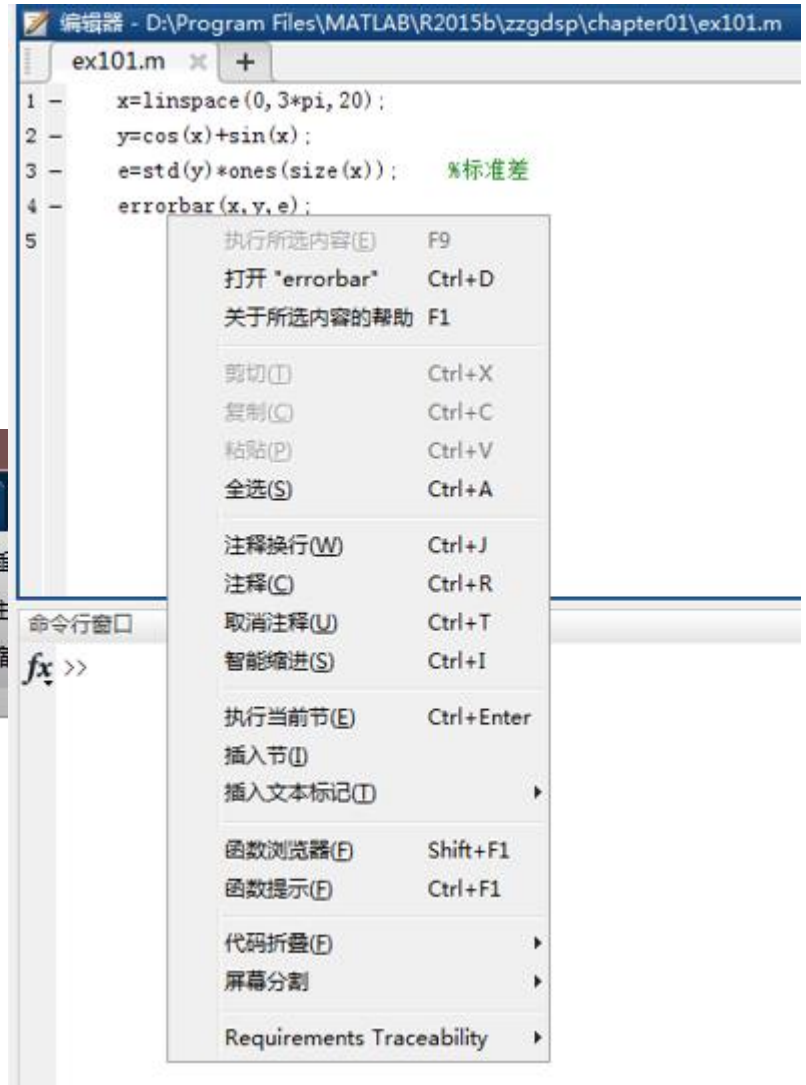
M文件的类型是普通的文本文件



2、Matlab工作环境

- M文件编辑窗口

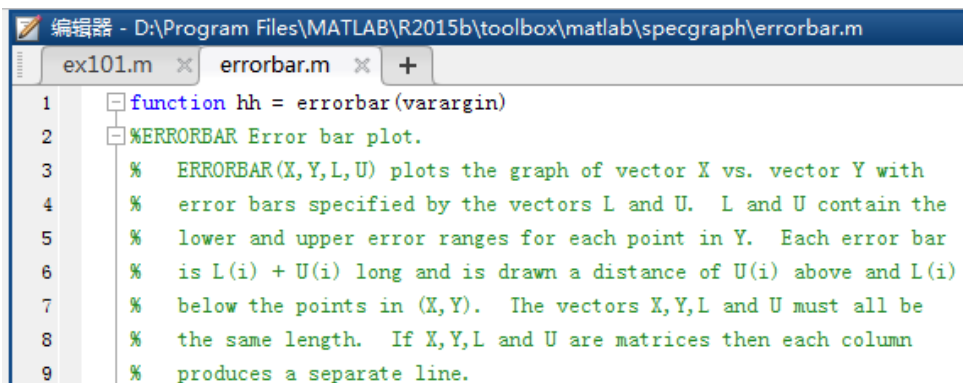
- (1) 编辑功能
- (2) 调试功能
- (3) 右键功能



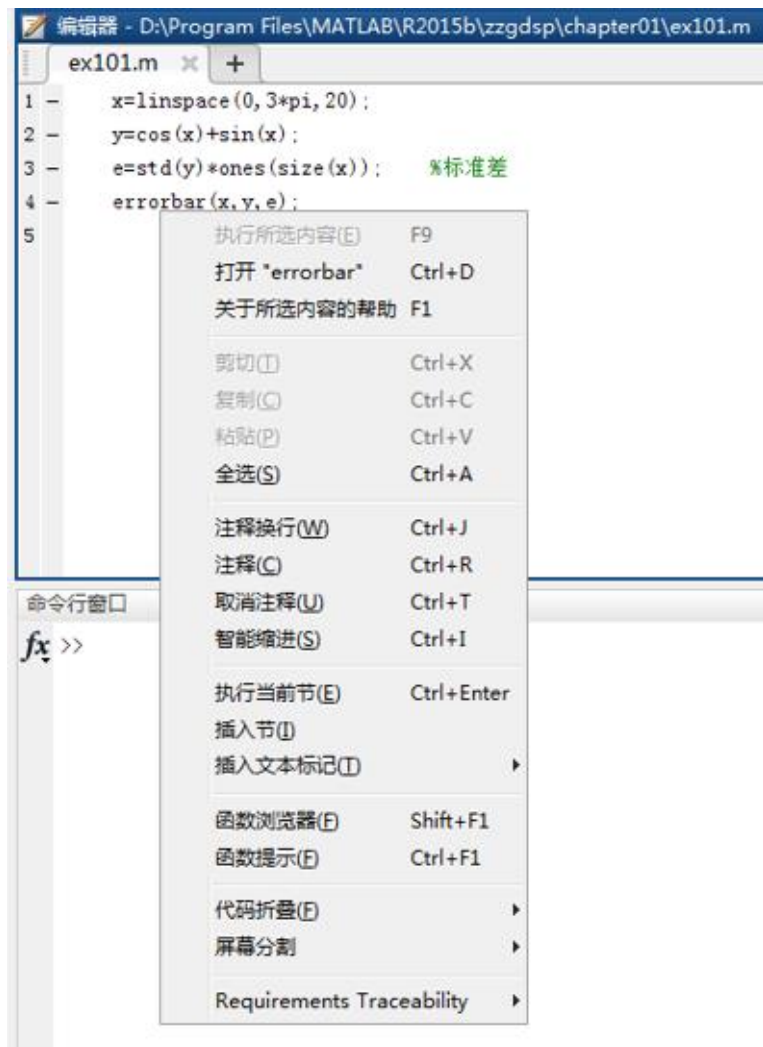
2、Matlab工作环境

- M文件编辑窗口

- (1) 编辑功能
- (2) 调试功能
- (3) 右键功能



```
1 function hh = errorbar(varargin)
2 %ERRORBAR Error bar plot.
3 % ERRORBAR(X,Y,L,U) plots the graph of vector X vs. vector Y with
4 % error bars specified by the vectors L and U. L and U contain the
5 % lower and upper error ranges for each point in Y. Each error bar
6 % is L(i) + U(i) long and is drawn a distance of U(i) above and L(i)
7 % below the points in (X,Y). The vectors X,Y,L and U must all be
8 % the same length. If X,Y,L and U are matrices then each column
9 % produces a separate line.
```



```
1 x=linspace(0,3*pi,20);
2 y=cos(x)+sin(x);
3 e=std(y)*ones(size(x)); %标准差
4 errorbar(x,y,e);
5
```

命令窗口
fx >>

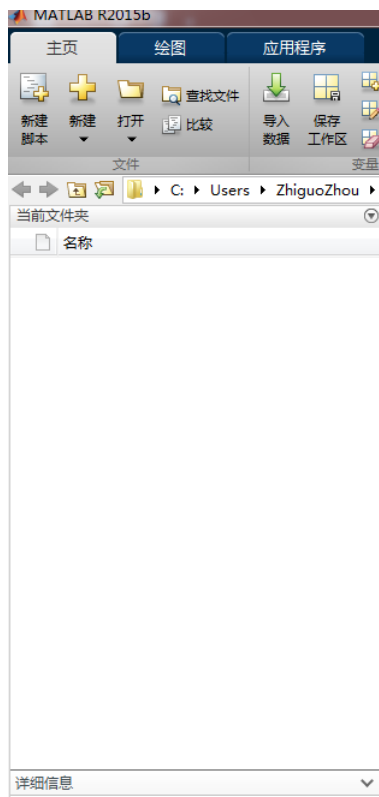
执行所选内容(E)	F9
打开 "errorbar"	Ctrl+D
关于所选内容的帮助	F1
剪切(T)	Ctrl+X
复制(C)	Ctrl+C
粘贴(P)	Ctrl+V
全选(S)	Ctrl+A
注释换行(W)	Ctrl+J
注释(C)	Ctrl+R
取消注释(U)	Ctrl+T
智能缩进(S)	Ctrl+I
执行当前节(E)	Ctrl+Enter
插入节(I)	
插入文本标记(I)	
函数浏览器(F)	Shift+F1
函数提示(F)	Ctrl+F1
代码折叠(F)	
屏幕分割	
Requirements Traceability	

2、Matlab工作环境

- 当前文件夹

1.2.5 当前文件夹

当前路径窗口显示当前用户所在的路径，可以在其中对MATLAB路径下的文件进行搜索、浏览、打开等操作，如图1-9所示。

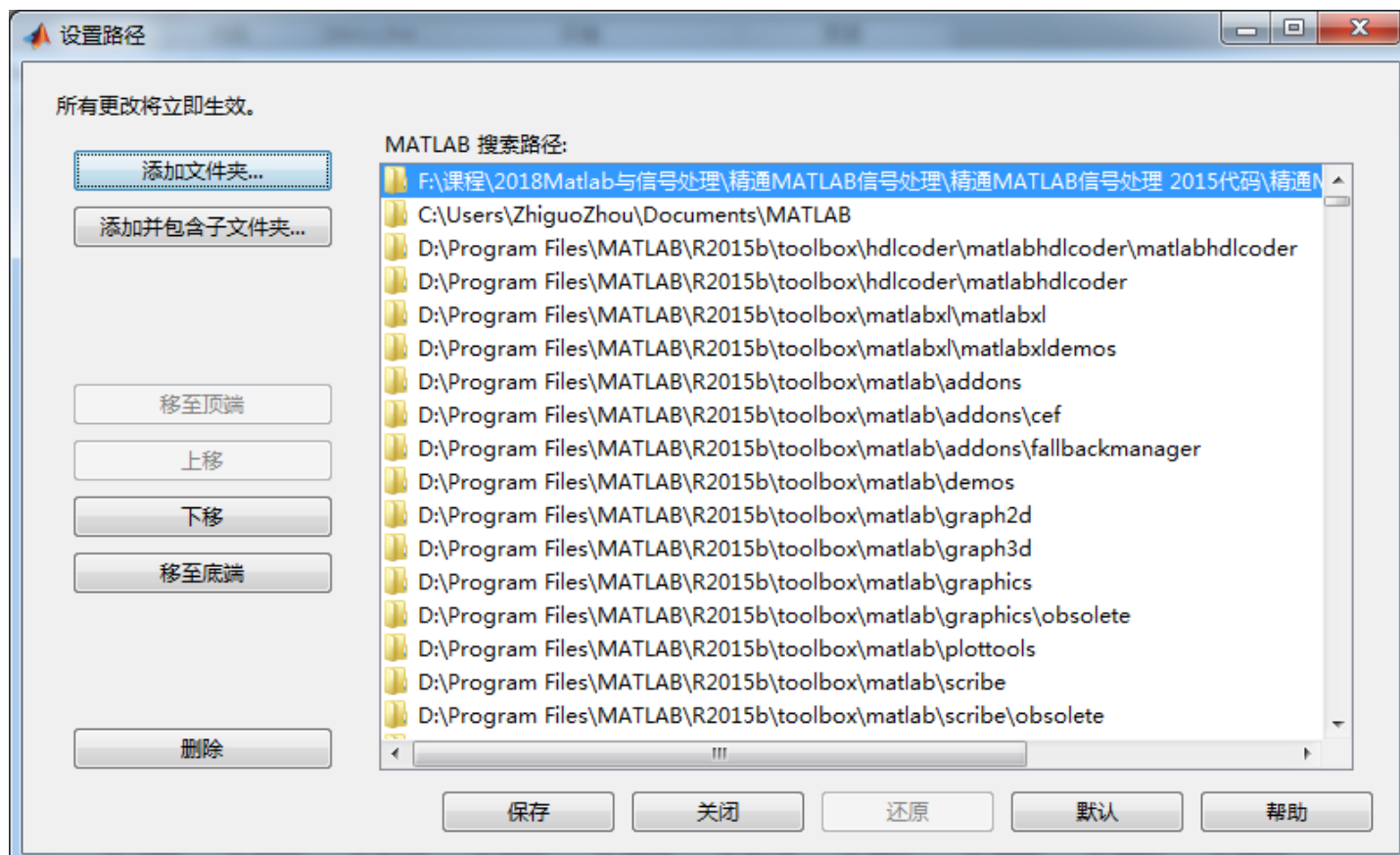


2、Matlab工作环境

• 搜索路径

注意中英文字符

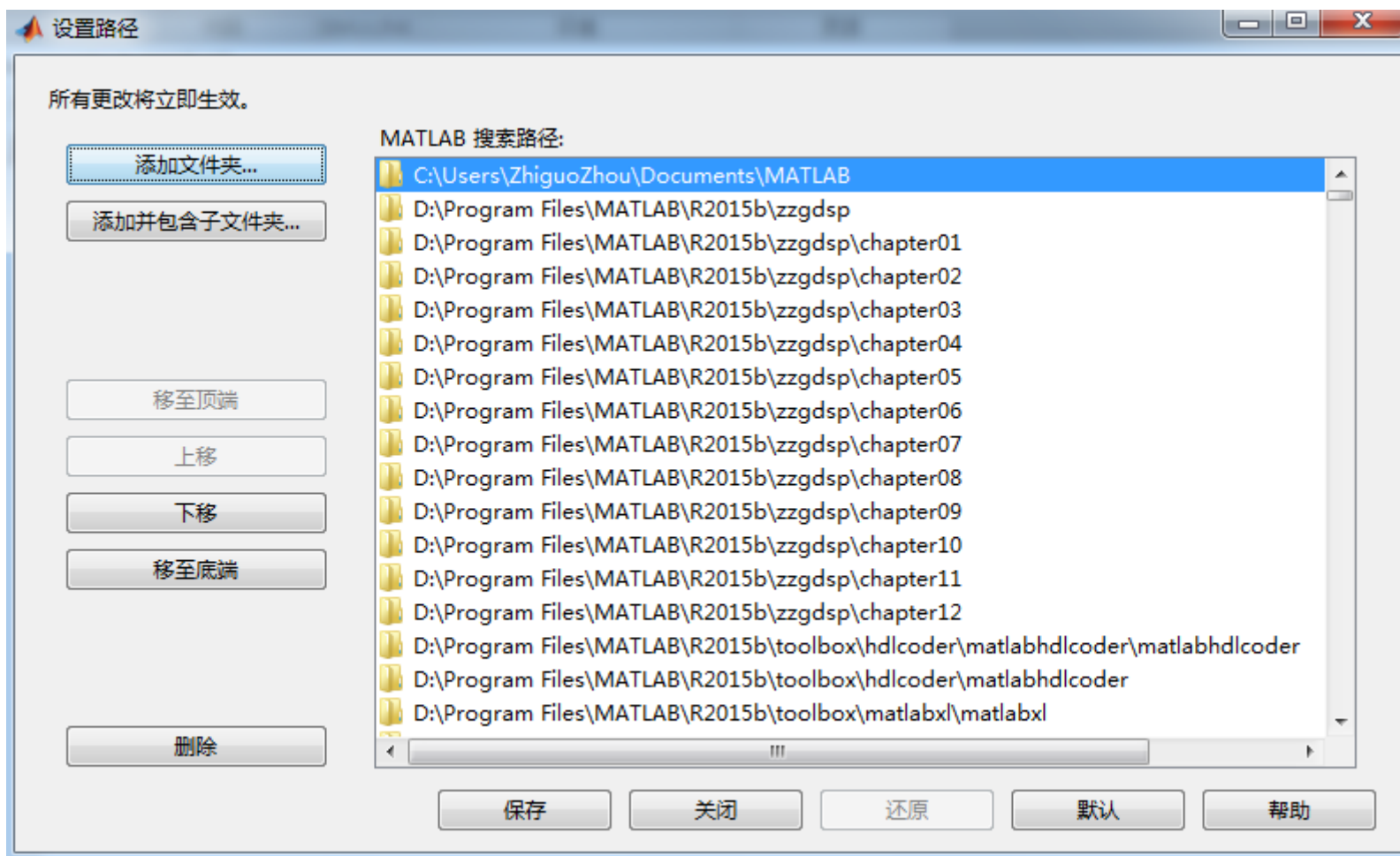
用户可以通过选择菜单栏中的“Set path”，或者在命令窗口输入`path`或`editpath`指令来查看MATLAB的搜索目录，如图1-10所示。



2、Matlab工作环境

- 搜索路径

不要有中文



3、程序控制结构

1.3 MATLAB程序控制结构

在MATLAB中，程序流程控制包含控制程序的基本结构和语法，结构化的程序主要有三种基本的程序结构。MATLAB语言的程序结构与其他高级语言是一致的，分为顺序结构、选择结构、循环结构。

MATLAB语言也给出了丰富的流程控制语句，以实现具体的程序设计。在M文件中，通过对流程控制语句的组合使用，实现多种复杂功能。MATLAB语言的流程控制语句主要有for、while、if-else-end及switch-case 4种语句。

3、程序控制结构

• 顺序结构

顺序结构是指所有组成程序源代码的语句按照由上至下的次序依次执行，直到程序的最后一个语句。这种程序优点是容易编制；缺点是结构单一，能够实现的功能有限。

在MATLAB语言的函数中，变量主要有输入变量、输出变量和函数内所使用的变量。

1. 数据输入

从键盘输入数据，则可以使用函数input来进行，该函数的调用格式为

```
A=input(提示信息,选项);
```

其中，提示信息为一个字符串，用于提示用户输入什么样的数据。

如果在input函数调用时采用's'选项，则允许用户输入一个字符串。例如，想输入一个人的姓名，可采用以下命令：

```
xm=input('What's your name?', 's');
```

2. 数据输出

MATLAB提供的命令窗口输出函数主要有函数disp，其调用格式为

```
disp(输出项);
```

其中，输出项可以为字符串，也可以为矩阵。

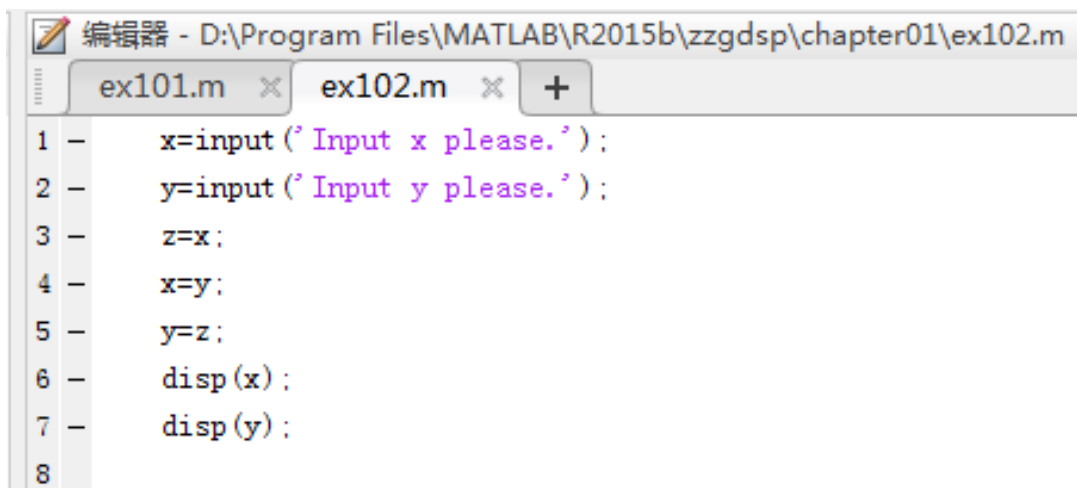
3、程序控制结构

- 顺序结构

(1) 数据输入

(2) 数据输出

ex102.m



```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex102.m
ex101.m x ex102.m x +
1 - x=input(' Input x please. ');
2 - y=input(' Input y please. ');
3 - z=x;
4 - x=y;
5 - y=z;
6 - disp(x);
7 - disp(y);
8
```

1.3.2 选择结构

在MATLAB中，选择结构依照不同的条件进行判断，然后根据判断的结果选择某种方法来解决某个问题。

if语句，在MATLAB中，if语句有3种格式。

(1) 单分支if语句

```
if 条件
    语句组
end
```

当条件成立时，执行语句组，执行完之后继续执行if语句的后继语句，若条件不成立，则直接执行if语句的后继语句。

(2) 条件判断语句也是程序设计语言中流程控制语句之一。使用该语句，可以选择执行指定的命令，MATLAB语言中的条件判断语句是if-else-end语句。

双分支if语句

```
if 条件
    语句组1
else
    语句组2
end
```

当条件成立时，执行语句组1，否则执行语句组2，语句组1或语句组2执行后，再执行if语句的后继语句。

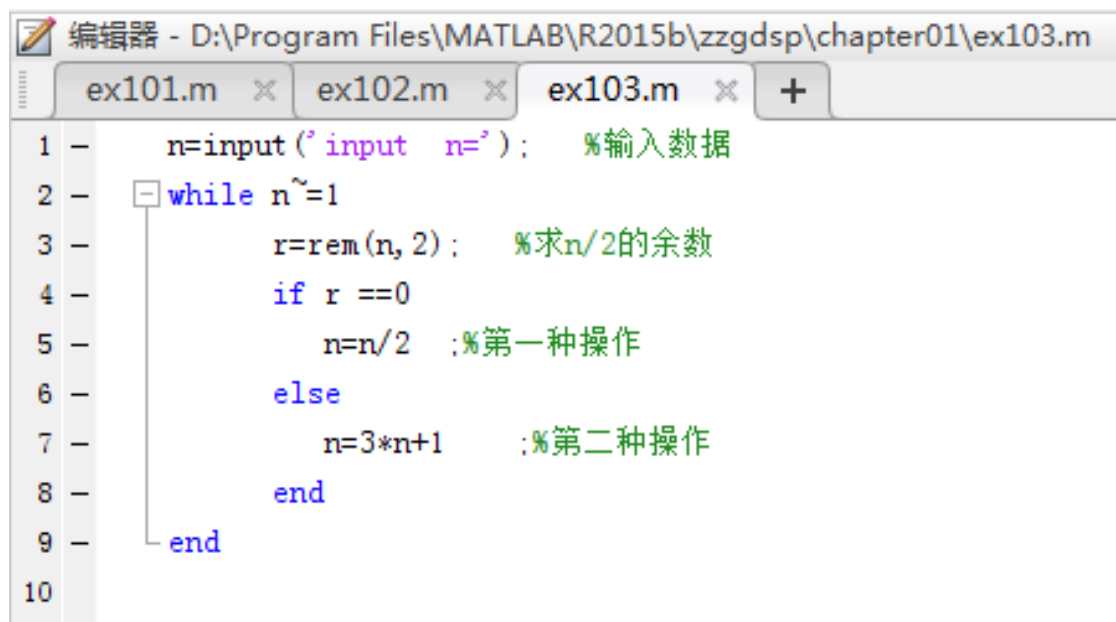
在程序设计中，也经常碰到需要进行多重逻辑选择的问题，这时可以采用if-else-end语句的嵌套形式：

3、程序控制结构

- 选择结构

(1) 单分支if语句

ex103.m



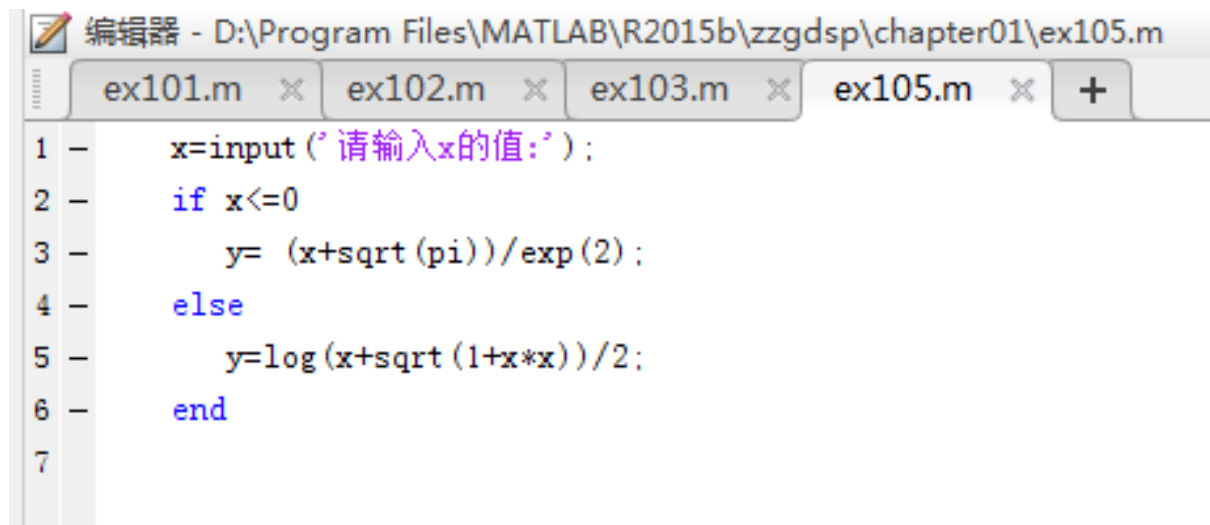
```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex103.m
ex101.m x ex102.m x ex103.m x +
1 - n=input('input n='); %输入数据
2 - while n~=1
3 -     r=rem(n,2); %求n/2的余数
4 -     if r ==0
5 -         n=n/2 ;%第一种操作
6 -     else
7 -         n=3*n+1 ;%第二种操作
8 -     end
9 - end
10
```

3、程序控制结构

- 选择结构

(2) 双分支if语句

ex105.m



```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex105.m
ex101.m × ex102.m × ex103.m × ex105.m × +
1 - x=input('请输入x的值:');
2 - if x<=0
3 -     y= (x+sqrt(pi))/exp(2);
4 - else
5 -     y=log(x+sqrt(1+x*x))/2;
6 - end
7
```

3、程序控制结构

- 选择结构

(2) 双分支 if 语句

ex106.m

```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex106.m
ex101.m x ex102.m x ex103.m x ex105.m x ex106.m x +
1 - A=input('请输入三角形的三条边:');
2 -     if A(1)+A(2)>A(3) && A(1)+A(3)>A(2) && A(2)+A(3)>A(1)
3 -         p=(A(1)+A(2)+A(3))/2;
4 -         s=sqrt(p*(p-A(1))*(p-A(2))*(p-A(3)));
5 -         disp(s);
6 -     else
7 -         disp('不能构成一个三角形。')
8 -     end
9
```

```
>> ex106
请输入三角形的三条边: [3 4 5]
6
```

3、程序控制结构

- 选择结构 (3) 多分支if语句

ex107.m

```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex107.m
ex101.m x ex102.m x ex103.m x ex105.m x ex106.m x ex107.m x
1 - c=input(' ','s');
2 -     if c>='A' && c<='Z'
3 -         disp(char(abs(c)+1));
4 -     elseif c>='a' && c<='z'
5 -         disp(char(abs(c)-1));
6 -     else
7 -         disp(c);
8 -     end
9
```

```
ex107
a
-
>> ex107
A
B
```


3、程序控制结构

- 选择结构

ex108.m

```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex108.m
ex108.m x +
1 - price=input('请输入商品价格');
2 - switch fix(price/100)
3 -     case {0,1}           %价格小于200
4 -         rate=0;
5 -     case {2,3,4}       %价格大于等于200但小于500
6 -         rate=0.1/100;
7 -     case num2cell(5:9) %价格大于等于500但小于1000
8 -         rate=0.2/100;
9 -     case num2cell(10:24) %价格大于等于1000但小于2500
10 -         rate=0.3/100;
11
12 - end
13 - price=price*(1-rate); %输出商品实际销售价格
14
```

3、程序控制结构

1.3.3 循环结构

在MATLAB中，循环结构就是在程序中某一条语句或多条语句重复多次的运行。

在MATLAB中，包含两种循环结构：循环次数不确定的while循环；循环次数确定的for循环。这两种循环结构不完全相同，各有特色。

for循环语句是流程控制语句的基础，使用该循环语句可以以指定的次数重复执行循环体内的语句。

(1) for语句

for语句的格式为

```
for 循环变量=表达式1:表达式2:表达式3
    循环体语句
end
```

其中，表达式1的值为循环变量的初值，表达式2的值为步长，表达式3的值为循环变量的终值。步长为1时，表达式2可以省略。

(2) while循环结构

while语句可以实现“当”型的循环结构，其语句格式为

```
while (表达式)
    MATLAB语句
end
```

其中，循环判断语句为某种形式的逻辑判断表达式，当该表达式的值为真时，执行循环体内的语句；当表达式的逻辑值为假时，退出当前的循环体。

在while循环语句中，在语句内必须有可以修改循环控制变量的命令，否则该循环语言将陷入死循环中，除非循环语句中有控制退出循环的命令，如break命令、continue命令。当程序流程运行至该命令时，则不论循环控制变量是否满足循环判断语句都退出当前循环，执行循环后的其他语句。

3、程序控制结构

- 循环结构

- (1) for语句

- (2) while循环结构

3、程序控制结构

- 程序流程控制

(1) break命令

(2) continue命令

(3) try命令

1. break命令

在MATLAB中，break命令通常用于for或while循环语句中，与if语句一起使用，中止本次循环，跳出最内层循环。

2. continue命令

通常用于for或while循环语句中，与if语句一起使用，达到跳过本次循环，去执行下一轮循环的目的。

3. try命令

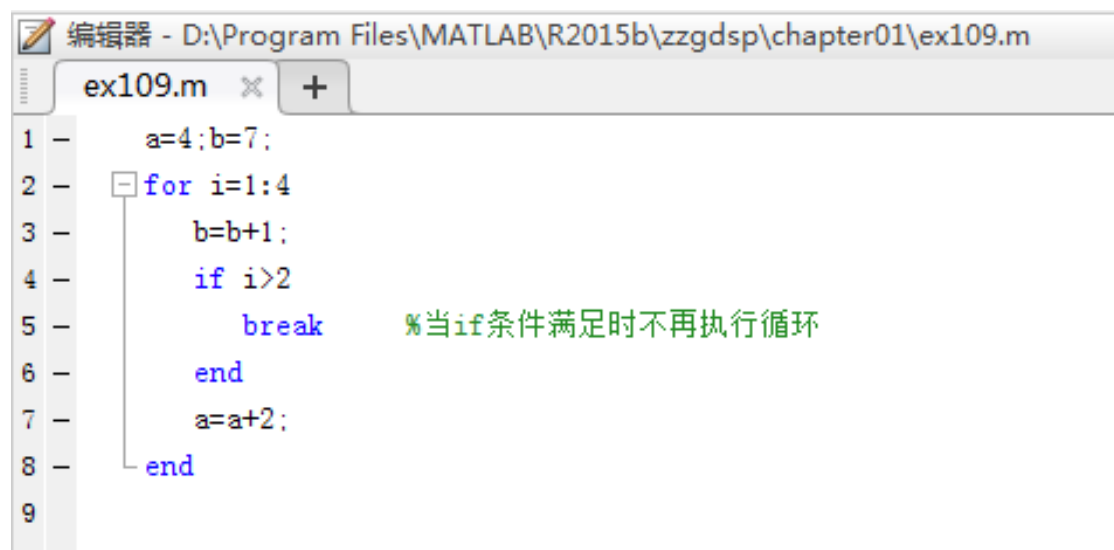
try语句是MATLAB特有的语句，它先试探性地执行语句1，如果出错，则将错误信息存入系统保留变量lasterr中，然后再执行语句2；如果不出错，则转向执行end后面的语句。此语句可以提高程序的容错能力，增加编程的灵活性。该指令的一般结构是

3、程序控制结构

- 程序流程控制

ex109.m

(1) break命令



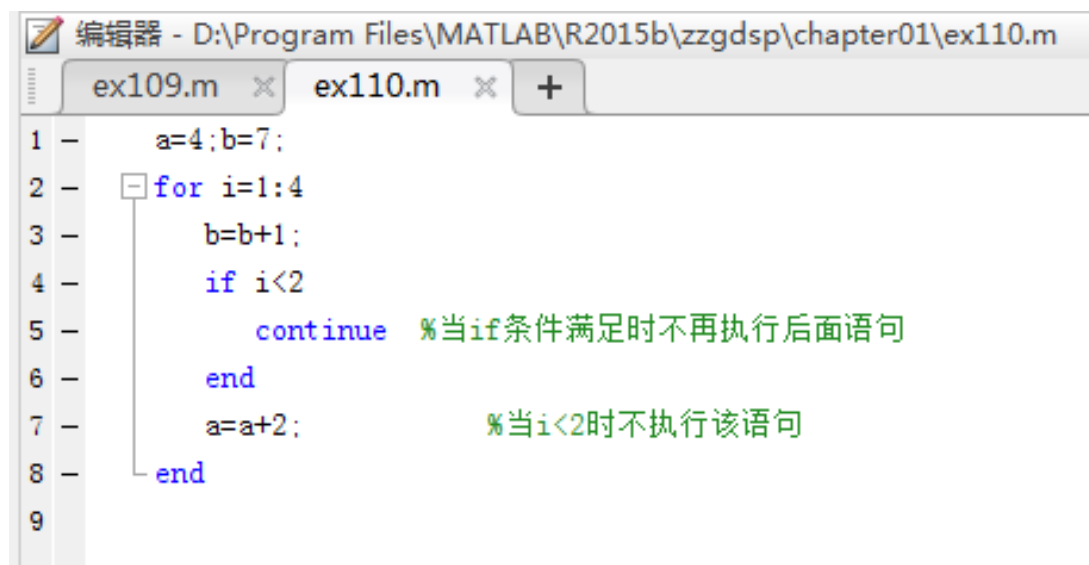
```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex109.m
ex109.m x +
1 - a=4;b=7;
2 - for i=1:4
3 -     b=b+1;
4 -     if i>2
5 -         break %当if条件满足时不再执行循环
6 -     end
7 -     a=a+2;
8 - end
9
```

3、程序控制结构

- 程序流程控制

ex110.m

(2) continue命令



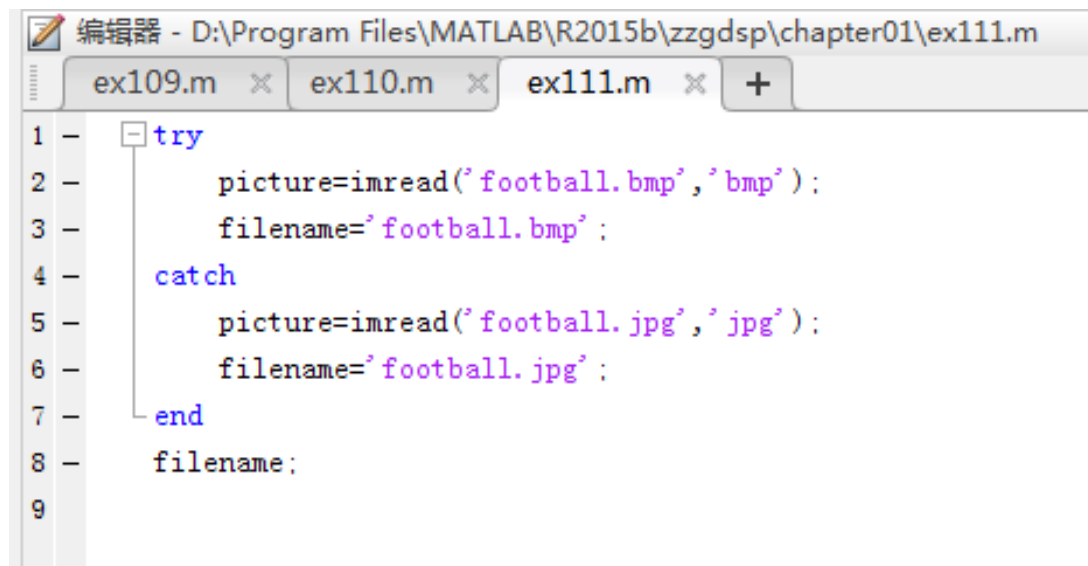
```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex110.m
ex109.m x ex110.m x +
1 - a=4;b=7;
2 - for i=1:4
3 -     b=b+1;
4 -     if i<2
5 -         continue %当if条件满足时不再执行后面语句
6 -     end
7 -     a=a+2; %当i<2时不执行该语句
8 - end
9
```

3、程序控制结构

- 程序流程控制

ex111.m

(3) try命令



```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex111.m
ex109.m x ex110.m x ex111.m x +
1 - try
2 -     picture=imread('football.bmp','bmp');
3 -     filename='football.bmp';
4 - catch
5 -     picture=imread('football.jpg','jpg');
6 -     filename='football.jpg';
7 - end
8 - filename;
9
```

4、变量、数值与表达式

- 变量

- (1) 特殊变量
- (2) 变量规范
- (3) 关键字
- (4) 作用域

1.4 变量、数值与表达式

1.4.1 变量

“变量”来源于数学，是计算机语言中能储存计算结果和能表示值抽象概念。表1-2给出了系统自定义的一些特殊的变量。

表1-2 系统中的特殊变量

特殊变量	说明
ans	默认变量名
π	圆周率
realmin	最小的正实浮点数
realmax	最大的正实浮点数
bitmax	最大正整浮点数
inf	无穷大
eps	浮点运算相对精度
nan	非数，即结果不能确定

在MATLAB中，当遇到一个新的变量名时，会自动产生一个变量并分配一个合适的存储空间，不需要对变量进行类型声明或维数声明。如果变量已经存在，则自动用新内容替换该变量的原有内容；若需要还会分配新的存储空间。例如：

4、变量、数值与表达式

- 变量

(1) 变量名区分大小写。如Price与price为两个不同的变量名，SIN不代表正弦函数。

(2) 变量名最多能包含63个字符，如果超出限制范围，从第64个字符开始，其后的字符都将被忽略。

(3) 变量名必须以字母开头，其后可以是任意数字、字母或下划线。

(4) 不允许出现标点符号，因为很多标点符号在MATLAB中具有特殊的意义。例如，“CB”与“C,B”会产生完全不同的结果，系统会认为“C,B”中间的逗号为分隔符，表示两个变量。

4、变量、数值与表达式

• 变量

在MATLAB语言的函数中，变量主要有输入变量、输出变量和函数内所使用的变量。

输入变量相当于函数入口数据，是一个函数操作的主要对象。某种程度上讲，函数的作用就是对输入变量进行加工以实现一定的功能。

函数的输入变量为形式参数，即只传递变量的值而不传递变量的地址，函数对输入变量的一切操作和修改，如果不依靠输出变量传出，将不会改变工作空间中该变量的值。

MATLAB语言提供了函数nargin和函数varargin来控制输入变量的个数，以实现不定个数参数输入的操作。

例如：

```
function y = bar(varargin)
```

也就是说，调用bar这个函数，可以传递1个参数，2个参数……或者不给参数。varargin表示输入变量的个数。

4、变量、数值与表达式

- 变量

ex113.m

```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex113.m
ex113.m x varargin.m x nargin.m x +
1 function[num1,num2,num3]=ex113(varargin)
2     num1=0;
3     num2=0;
4     num3=0;
5     if nargin==1
6         num1=1;
7     elseif nargin == 2
8         num2=2;
9     elseif nargin == 3
10        num3=3;
11    end
12
```

```
>> [num1,num2,num3]=ex113(1)
```

```
num1 =
```

```
1
```

```
num2 =
```

```
0
```

```
num3 =
```

```
0
```

```
>> [num1,num2,num3]=ex113(1,2,3)
```

```
num1 =
```

```
0
```

```
num2 =
```

```
0
```

```
num3 =
```

```
3
```

4、变量、数值与表达式

- 变量

函数对于函数变量而言，还应当指出其作用域的问题。在MATLAB语言中，函数内定义的变量均被视为局部变量，即不加载到工作空间中，如果希望使用全局变量，则应当使用命令global定义，而且在任何使用该全局变量的函数中都应加以定义。

1.4.2 数值

在MATLAB中，数值均采用习惯的十进制，可以带小数点及正负号。例如，以下写法都是合法的。

```
109      -35.9      -0.009      0.004
```

科学计数法采用字符e来表示10的幂。例如：

```
9.45e2    1.26e3    -2.1e-5
```

虚数的扩展名为i或者j。例如：

```
2i        3ej        -3.14j
```

在采用IEEE浮点算法的计算机上，实数的数值范围大致为 $-10e308 \sim 10e308$ 。

在MATLAB中输入同一数值，有时会发现，在命令行窗口中显示数据的形式有所不同。例如，0.3有时显示为0.3，但有时会显示为0.300。这是因为数据显示的格式不同。

在一般情况下，MATLAB内部每一个数据元素都是用双精度数来表示和存储的，数据输出时用户可以用format命令设置或改变数据输出格式。表1-3给出了不同种类的数据显示格式。

4、变量、数值与表达式

- 数值
 - (1) 格式
 - (2) 精度

表1-3 数据显示格式

格式	说明
format	设置输出格式
format short	表示短格式（默认显示格式），只显示5位。例如：3.1416
format long	表示长格式，双精度数15位，单精度数7位。例如：3.14159265358979
format short e	表示短格式e方式，只显示5位。例如：3.1416e+000
format long e	表示长格式e方式。例如：3.141592653589793e+000
format short g	表示短格式g方式（自动选择最佳表示格式），只显示5位。例如：3.1416
format long g	表示长格式g方式。例如：3.14159265358979
format compact	表示压缩格式。变量与数据之间在显示时不留空行
format loose	表示自由格式。变量与数据之间在显示时留空行
format hex	表示十六进制格式表示。例如：400921fb54442d18

4、变量、数值与表达式

1.4.3 表达式

ex116.m

在MATLAB中，数学表达式的运算操作尽量设计得接近于习惯，不同与其他编程语言在有些情况下一次只能处理一个数据，MATLAB允许快捷、方便地对整个矩阵进行操作。MATLAB表达式采用熟悉的数学运算符和优先级，如表1-4所示（表中运算符的优先级从上到下依次升高）。

表1-4 MATLAB的运算符优先级与表达式

运算	MATLAB运算符	MATLAB表达式
加	+	a+b
减	-	a-b
乘	* (., *)	a*b
除	/	a/b
幂	^ (., ^)	a^b
复数矩阵的（共轭）转置	' (., ')	
小括号指定优先级	()	(a+b)*c

MATLAB与经典的数学表达式也有所区别，例如，对矩阵进行右除与左除操作的结果是不同的。下面通过一个简单的例子演示复数矩阵的转置与共轭转置操作以及区别。

5、数组与矩阵

- 数组的创建与操作
- 常见的数组运算
- 矩阵的表述
- 矩阵的拼接
- Matlab矩阵寻访
- Matlab矩阵的运算

5、数组与矩阵

1.5 数组与矩阵

数值数组(简称为数组)是MATLAB中最重要的一种内建数据类型,是MATLAB软件定义的运算规则,其目的是为了数据管理方便、操作简单、指令形式自然和执行计算的有效。

1.5.1 数组的创建与操作

行数组： n 个元素排成一行又称为行向量（row vector）；列数组： m 个元素排成一列又称为列向量（column vector）。

用方括号 `[]` 创建一维数组就是将整个数组放在方括号里，行数组元素用空格或逗号分隔，列数组元素用分号分隔，标点符号一定要在英文状态下输入。

【例1-17】 创建数组示例。

ex117.m

```
clear all
A= []
B= [6 3 4 3 2 1]
C= [6,5,4,3,2,1]
D= [6;3;4;3;2;1]
E=B' %转置
```

运行结果如下：

```
A =
    []
B =
     6     3     4     3     2     1
C =
     6     5     4     3     2     1
D =
     6
     3
```

5、数组与矩阵

1.5.2 常见的数组运算

1. 数组的算数运算

两个一维数组之间进行运算的要求如下：

- (1) 两个数组都为行数组(或都为列数组)；
- (2) 数组元素个数相同。

表1-5给出了数组常用的运算格式。

Ex123-128.m

表1-5 数组常用的运算格式

格式	说明	格式	说明
$x + y$	数组加法	$x ./y$	数组左除
$x - y$	数组减法	$x ./y$	数组右除
$x .*y$	数组乘法	$x.^y$	数组求幂

5、数组与矩阵

1.5.3 矩阵的表示

MATLAB的强大功能之一体现在能直接处理向量或矩阵。当然首要任务是输入待处理的向量或矩阵。对于数组的创建有如下4种方法：

- (1) 直接输入法。
- (2) 载入外部数据文件。
- (3) 利用MATLAB内置函数创建矩阵。
- (4) 利用M文件创建和保存数组。

1. 直接输入法

最简单的建立矩阵的方法是从键盘直接输入矩阵的元素。将矩阵的元素用方括号括起来，按矩阵行的顺序输入各元素，同一行的各元素之间用空格或逗号分隔，不同行的元素之间用分号分隔。

如果只输入一行就形成一个数组（又称作向量）。矩阵或数组中的元素可以是任何MATLAB表达式，可以是实数，也可以是复数。使用此方法创建矩阵需要注意以下规则：

- (1) 矩阵元素必须在“[]”内；
- (2) 矩阵的同行元素之间用空格(或“,”) 隔开；
- (3) 矩阵的行与行之间用“;”(或回车符)隔开。

5、数组与矩阵

2. 载入外部数据文件

在MATLAB中，Load函数用于载入生成的包含矩阵的二进制文件，或者读取包含数值数据的文本文件。文本文件中的数字应排列成矩形，每行只能包含矩阵的一行元素，元素与元素之间用空格分隔，每行元素的个数必须相等。

例如，用Windows自带的记事本或用MATLAB的文本调试编辑器创建一个包含下列数字的文本文件。

```
1 2 3 4 5 6 7 8 9 0
```

把该文件命名为data.txt，并保存在MATLAB的目录下。如需读取该文件，可在命令窗口中输入：

```
>>load data.txt
```

系统将读取该文件并创建一个变量data，包含上面的这个矩阵。在MATLAB工作空间中可以查看这个变量。

3. 利用MATLAB内置函数创建矩阵

在MATLAB中，系统内置特殊函数可以用于创建矩阵，通过这些函数，可以方便地得到想要的特殊矩阵。系统内置创建矩阵特殊的函数如表1-6所示。

表1-6 系统内置创建矩阵特殊的函数

函数名	功能介绍
ones()	产生全为1的矩阵
zeros()	产生全为0的矩阵
eye()	产生单位阵
rand()	产生在(0, 1)区间均匀分布的随机阵
randn()	产生均值为0, 方差为1的标准正态分布随机矩阵
compan	伴随矩阵
gallery	Higham检验矩阵
hadamard	Hadamard阵
hankel	Hankel阵
hilb	Hilbert阵
invhilb	逆Hilbert阵
magic	魔方阵
pascal	Pascal阵
rosser	经典对称特征值
toeplitz	Toeplitz阵
vander	Vander阵
wilksion	wilksion特征值检验矩阵

4. 利用M文件创建和保存矩阵

此方法需要用MATLAB自带的文本编辑调试器或其他文本编辑器来创建一个文件，代码和在MATLAB命令窗口中输入的命令一样，然后以.m格式保存该文件。

1.5.4 矩阵的拼接

两个或者两个以上的单个矩阵，按一定的方向进行连接，生成新的矩阵就是矩阵的拼接。矩阵的拼接就是一种创建矩阵的特殊方法，区别在于基础元素是原始矩阵，目标是新的合并矩阵。

1. 基本拼接

矩阵的拼接有按照水平方向拼接和按照垂直方向拼接两种。例如，对矩阵A和B进行拼接，拼接表达式分别如下：

水平方向拼接： $C = [A \ B]$ 或 $C = [A, B]$ ；

垂直方向拼接： $C = [A; B]$ 。

Ex133-4.m

【例1-33】 把3阶魔术矩阵和3阶单位矩阵在水平方向上拼接成为一个的新矩阵，垂直方向上拼接成为另一个的新矩阵。

```
clear all;
c= magic(3)           %3阶魔术矩阵
d = eye (3)          %3阶单位矩阵
E = [c,d]            %水平方向上拼接
F = [c;d]            %垂直方向上拼接
```

运行结果如下：

```
c =
     8     1     6
     3     5     7
     4     9     2
```

1.5.5 MATLAB矩阵寻访

在MATLAB中，矩阵寻访的主要方法有：下标寻访、单元素寻访、多元素寻访，下面进行介绍。

1. 下标寻访

MATLAB中矩阵的下标表示与常用的数学习惯相同，使用分别表示行和列的“双下标”(Row-Column Index)，矩阵中的元素都有对应的“第几行，第几列”。这种表示方法简单直观，几何概念比较清晰。

【例1-35】 利用上下标来寻访矩阵元素。

Ex135-7.m

```
>>a=[1 2 3;4 5 6;7 8 9]
>>a(1,1)
>>a(2,2)
>>a(3,3)
```

运行结果如下：

```
a =
     1     2     3
     4     5     6
     7     8     9

ans =
     1

ans =
     5

ans =
     9
```

1.5.6 MATLAB矩阵的运算

在MATLAB中，矩阵的运算包括： $+$ （加）、 $-$ （减）、 $*$ （乘）、 $/$ （右除）、 \backslash （左除）、 $^$ （乘方）等。

1. 矩阵加减运算

两个矩阵相加或减是指有相同的行和列的两矩阵对应元素相加减。允许参与运算的两矩阵之一是标量（常量）。标量与矩阵的所有元素分别进行加减操作。

由 $A+B$ 和 $A-B$ 实现矩阵的加减运算。

【例1-38】 对矩阵A和B进行加减运算。

Ex138-41.m

```
A= [5 4 6;8 9 7;3 6 4]
B= [9 1 7;5 6 6;5 6 8]
C=A+B
D=A-B
```

运行结果如下：

```
A =
     5     4     6
     8     9     7
     3     6     4
B =
     9     1     7
     5     6     6
     5     6     8
```

2. 矩阵乘法运算

矩阵A的列数必须等于矩阵B的行数，若A为 $m \times n$ 矩阵，B为 $n \times p$ 矩阵，则 $C=A*B$ 为 $m \times p$ 矩阵。标量可与任何矩阵相乘，即矩阵的所有元素都与标量相乘。

3. 矩阵除法运算

矩阵除法运算： \backslash 和 $/$ ，分别表示左除和右除。 $A \backslash B$ 等效于A矩阵的逆左乘B矩阵，而 B/A 等效于A矩阵的逆右乘B矩阵。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。对于矩阵运算，一般 $A \backslash B \neq B/A$ 。

4. 矩阵乘方运算

若A为方阵，x为标量，一个矩阵的乘方运算可以表示为 A^x 。

6、数据分析

- 平均值和中值
- 数据比较
- 期望
- 方差
- 协方差和相关系数

6、数据分析

数据分析是指用适当的统计方法对收集来的数据资料进行分析，以求最大化地开发数据资料的功能，发挥数据资料的作用；数据分析是为了提取有用信息和形成结论而对数据加以详细研究和概括总结的过程。

1.6.1 平均值与中值

在MATLAB中，可以利用函数mean求算术平均值，该函数的调用方法为

mean(X)：X为向量，返回X中各元素的平均值。

mean(A)：A为矩阵，返回A中各列元素的平均值构成的向量。

mean(A, dim)：给出维数内的平均值。

dim为维数，返回A中。

当X为向量时，算术平均值的数学含义是 $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ ，即样本均值。

【例1-42】 利用函数mean求算术平均值。

ex142.m

```
A = [1 2 4 5; 2 3 5 6; 1 3 1 5]
mean(A)
mean(A, 1)
```

运行结果如下：

```
A =
     1     2     4     5
     2     3     5     6
     1     3     1     5
ans =
  1.3333   2.6667   3.3333   5.3333
```

1.6.2 数据比较

在MATLAB中，函数`sort`可以用于排序，该函数的调用方法为：

`Y=sort(X)`：X为向量，返回X按由小到大排序后的向量。

`Y=sort(A)`：A为矩阵，返回A的各列按由小到大排序后的矩阵。

`[Y,I]=sort(A)`：Y为排序的结果，I中元素表示Y中对应元素在A中的位置。

`sort(A,dim)`：在给定的维数dim内排序。

【例1-43】 利用函数`sort`排序。

Ex143-5.m

```
A = [1 2 3;4 5 6;3 7 8]
sort(A)
[Y,I] = sort(A)
```

运行结果如下：

```
A =
     1     2     3
     4     5     6
     3     7     8
ans =
     1     2     3
     3     5     6
     4     7     8

Y =
     1     2     3
     3     5     6
     4     7     8

I =
     1     1     1
     3     2     2
     2     3     3
```


6、数据分析

1.6.3 期望

函数`mean`用于计算样本均值。

随机抽取6个滚珠测得直径如下：

15.70 15.21 14.90 15.91 15.32 15.32

【例1-46】 试求样本平均值。

ex146.m

```
x = [15.70 15.21 14.90 15.91 15.32 15.32];  
mean(x)           %计算样本均值
```

运行结果如下：

```
ans =  
    15.3933
```

6、数据分析

1.6.4 方差

函数var用于求样本方差，该函数的调用方法如下：

D=var(X)： $\text{var}(X) = s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2$ ，若X为向量，则返回向量的样本方差。

D=var(A)：A为矩阵，则D为A的列向量的样本方差构成的行向量。

D=var(X, 1)：返回向量（矩阵）X的简单方差即（置前因子为 $\frac{1}{n}$ 的方差）。

D=var(X, w)：返回向量（矩阵）X的以w为权重的方差。

函数Std用于求标准差，该函数的调用方法如下：

std(X)：返回向量（矩阵）X的样本标准差（置前因子为 $\frac{1}{n-1}$ ），即： $\text{std} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n x_i - \bar{X}}$ 。

std(X, 1)：返回向量（矩阵）X的标准差（置前因子为 $\frac{1}{n}$ ）。

std(X, 0)：与std(X)相同。

std(X, flag, dim)：返回向量（矩阵）中维数为dim的标准差值，其中flag=0时，置前因子为 $\frac{1}{n-1}$ ；否则置前因子为 $\frac{1}{n}$ 。

ex147-8.m

6、数据分析

1.6.5 协方差与相关系数

ex149-51.m

函数cov用于求协方差，该函数的调用方法如下：

cov(X)：求向量X的协方差。

cov(A)：求矩阵A的协方差矩阵，该协方差矩阵的对角线元素是A的各列的方差，即 $\text{var}(A)=\text{diag}(\text{cov}(A))$ 。

函数corrcoef用于求相关系数，该函数的调用方法如下：

corrcoef(X,Y)：返回列向量X,Y的相关系数，等同于 $\text{corrcoef}([X,Y])$ 。

corrcoef(A)：返回矩阵A的列向量的相关系数矩阵。

7、图形绘制

1.7.1 二维图形的绘制

在MATLAB中绘制二维图形，通常采用以下步骤：

- (1) 准备数据；
- (2) 设置当前绘图区；
- (3) 绘制图形；
- (4) 设置图形中曲线和标记点格式；
- (5) 设置坐标轴和网格线属性；
- (6) 标注图形；
- (7) 保存和导出图形。

7、图形绘制

- 二维图形绘制

ex152.m

ex153.m

ex154.m

ex155.m

ex156.m

ex157.m

7、图形绘制

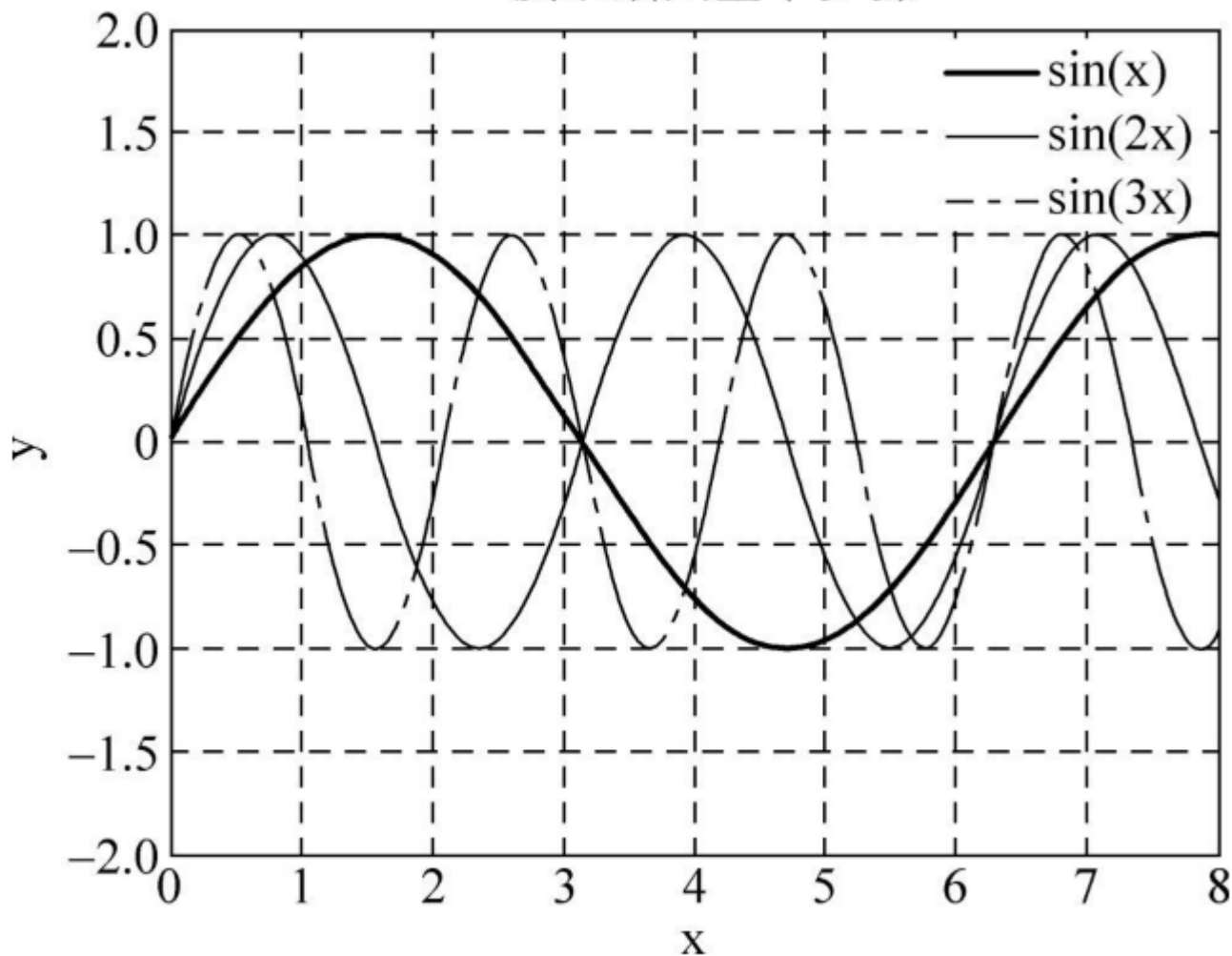
【例1-52】 演示绘图步骤。在同一坐标轴上绘制 $\sin(x)$ 、 $\sin(2x)$ 和 $\sin(3x)$ 这三条曲线。

```
clear all;
%准备数据
x=0:0.01:3*pi;
y1=sin(x);
y2=sin(2*x);
y3=sin(3*x);
%设置当前绘图区
figure;
%绘图
plot(x,y1,x,y2,x,y3);
%设置坐标轴和网格线属性
axis([0 8 -2 2]);
grid on;
%标注图形
xlabel('x');
ylabel('y');
title('演示绘图基本步骤')
legend('sin(x)', 'sin(2x)', 'sin(3x)')
legend('sin(x)', 'sin(2x)', 'sin(3x)')
```

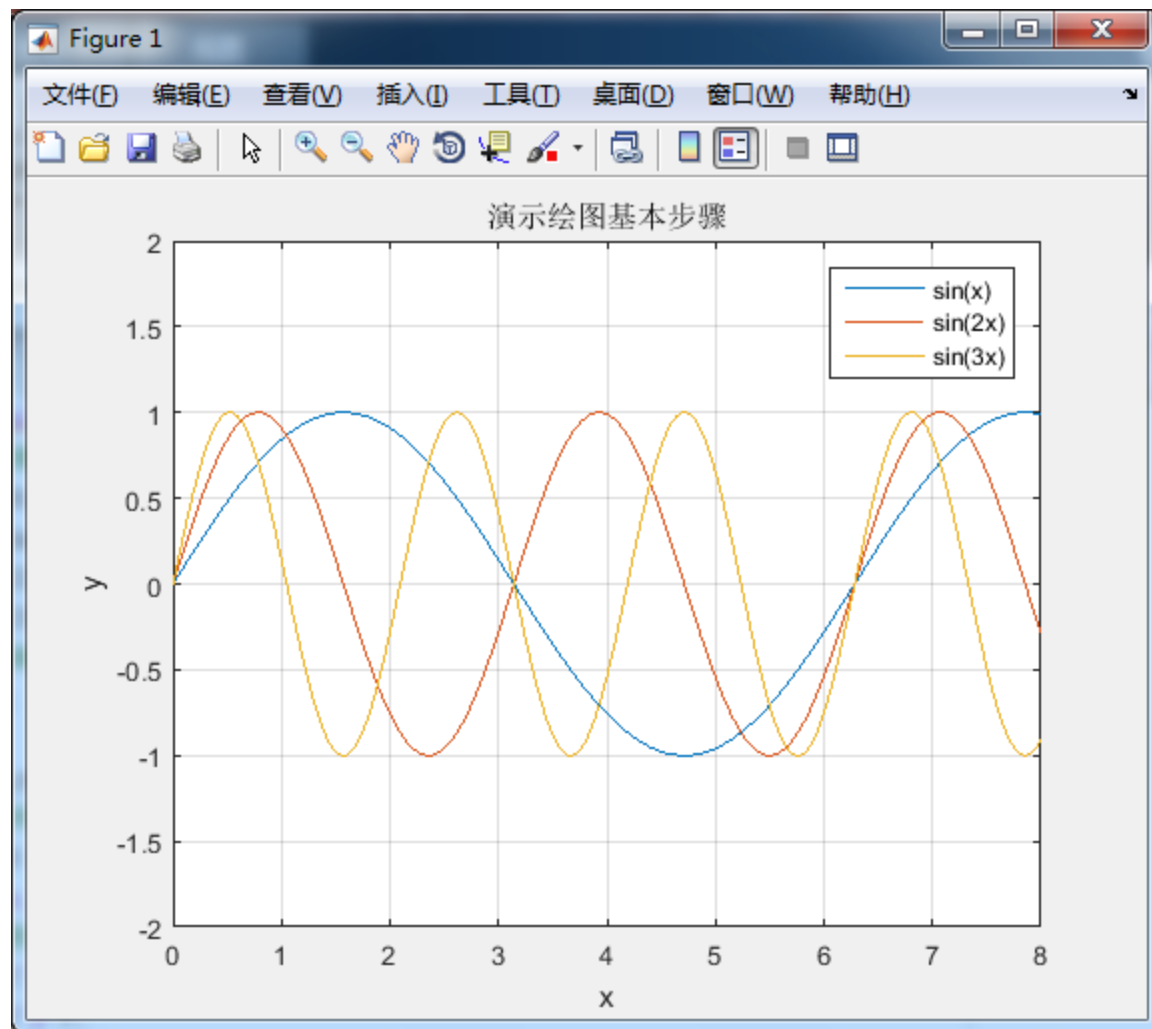
7、图形绘制

运行结果如图1-13所示。

演示绘图基本步骤



```
编辑器 - D:\Program Files\MATLAB\R2015b\zgzdsp\chapter01\ex152.m
ex152.m x +
1 - clear all;
2   %准备数据
3 - x=0:0.01:3*pi;
4 - y1=sin(x);
5 - y2=sin(2*x);
6 - y3=sin(3*x);
7   %设置当前绘图区
8 - figure;
9   %绘图
10 - plot(x, y1, x, y2, x, y3);
11  %设置坐标轴和网格线属性
12 - axis([0 8 -2 2]);
13 - grid on;
14  %标注图形
15 - xlabel('x');
16 - ylabel('y');
17 - title('演示绘图基本步骤')
18 - legend('sin(x)', 'sin(2x)', 'sin(3x)')
19 - legend('sin(x)', 'sin(2x)', 'sin(3x)')
20
```

1. 绘制二维曲线图

在MATLAB中，主要的二维绘图函数如下：

- (1) plot: x轴和y轴均为线性刻度。
- (2) loglog: x轴和y轴均为对数刻度。
- (3) semilogx: x轴为对数刻度，y轴为线性刻度。
- (4) semilogy: x轴为线性刻度，y轴为对数刻度。
- (5) plotyy: 绘制双纵坐标图形。
- (6) polar: 绘制极坐标图。
- (7) grid: 在图形窗口添加网格(grid on)或去掉网格(grid off)。
- (8) zoom: 对图形进行放大缩小操作(zoom on容许或zoom off不容许)。
- (9) ginput: 用鼠标获取图形中点的位置。

其中，plot是最基本的二维绘图函数，其调用格式有：

1) plot(Y)

(1) 若Y为实向量，则以该向量元素的下标为横坐标，以Y的各元素值为纵坐标，绘制二维曲线。

(2) 若Y为复数向量，则等效于plot(real(Y), imag(Y))。

(3) 若Y为实矩阵，则按列绘制每列元素值相对其下标的二维曲线，曲线的条数等于Y的列数。

(4) 若Y为复数矩阵，则按列分别以元素实部和虚部为横、纵坐标绘制多条二维曲线。

2) `plot(X, Y)`

(1) 若X、Y为长度相等的向量，则绘制以X和Y为横、纵坐标的二维曲线。

(2) 若X为向量，Y是有一维与X同维的矩阵，则以X为横坐标，与X同维的Y的一维为纵坐标。曲线条数与Y的另一维相同。

(3) 若X、Y为同维矩阵，则绘制以X和Y对应的列元素为横、纵坐标的多条二维曲线，曲线条数与矩阵的列数相同。

3) `plot(X1, Y1, X2, Y2, ..., Xn, Yn)`

每一对参数 X_i 和 Y_i 的取值和所绘图形与2)中相同。

4) `plot(X1, Y1, LineSpec, ...)`

以LineSpec指定的属性，绘制所有 X_n 、 Y_n 对应的曲线。

5) `plot(X1, Y1, 'PropertyName', PropertyValue, ...)`

由plot绘制的所有曲线，按照设置的属性值进行绘制。

6) `h=plot(...)`

调用函数plot时，同时返回每条曲线的图形句柄h。

【例1-53】 用不同线型和颜色在同一坐标内绘制曲线 $y=2e^{-0.5x}\sin(2\pi x)$ 及其包络线。

运行程序如下：

```
x=(0:pi/100:2*pi)';  
y1=2*exp(-0.5*x)* [1,-1];  
y2=2*exp(-0.5*x) .* sin(2*pi*x);  
x1=(0:12)/2;  
y3=2*exp(-0.5*x1) .* sin(2*pi*x1);  
plot(x,y1,'g:',x,y2,'b--',x1,y3,'rp');
```

运行结果如图1-14所示。

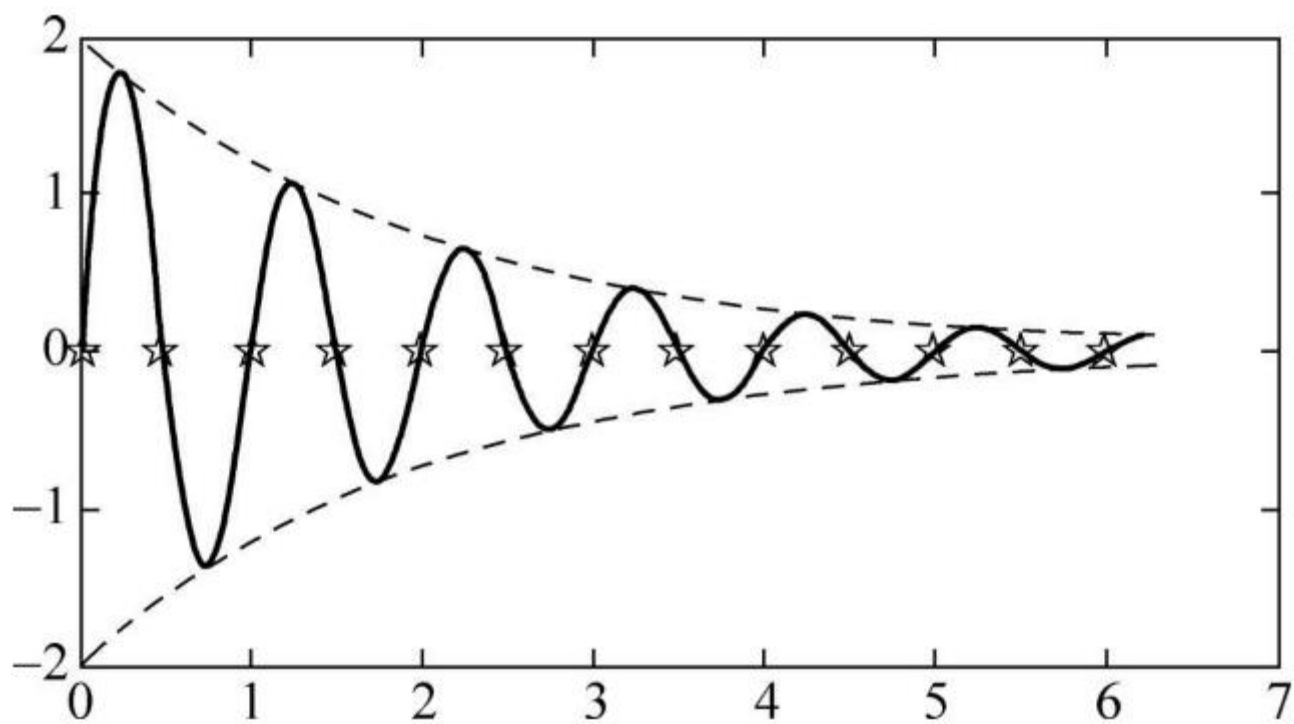
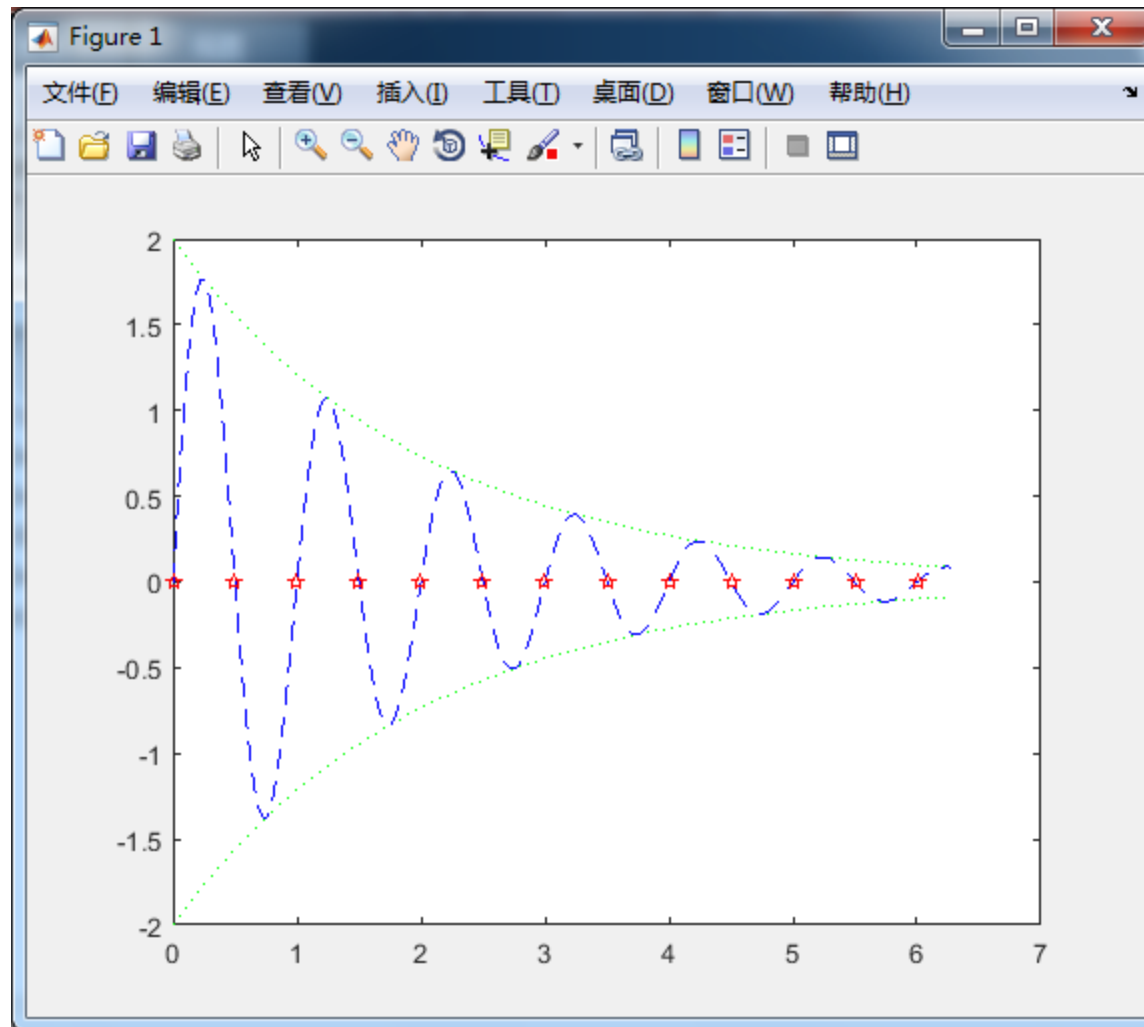


图1-14 二维曲线

```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex154.m
ex154.m x +
1 - x=(0:pi/100:2*pi)';
2 - y1=2*exp(-0.5*x)*[1,-1];
3 - y2=2*exp(-0.5*x).*sin(2*pi*x);
4 - x1=(0:12)/2;
5 - y3=2*exp(-0.5*x1).*sin(2*pi*x1);
6 - plot(x,y1,'g:',x,y2,'b--',x1,y3,'rp');
7 |
```



2. 绘制对数曲线图

MATLAB提供了绘制对数和半对数坐标曲线的函数，调用格式为

```
semilogx(x1,y1,选项1,x2,y2,选项2,...)  
semilogy(x1,y1,选项1,x2,y2,选项2,...)  
loglog(x1,y1,选项1,x2,y2,选项2,...)
```

【例1-54】 绘制 e^x 的图形。

运行程序如下：

```
x=logspace(-1,2);  
loglog(x,exp(x),'-s')  
grid on
```

运行结果如图1-15所示。

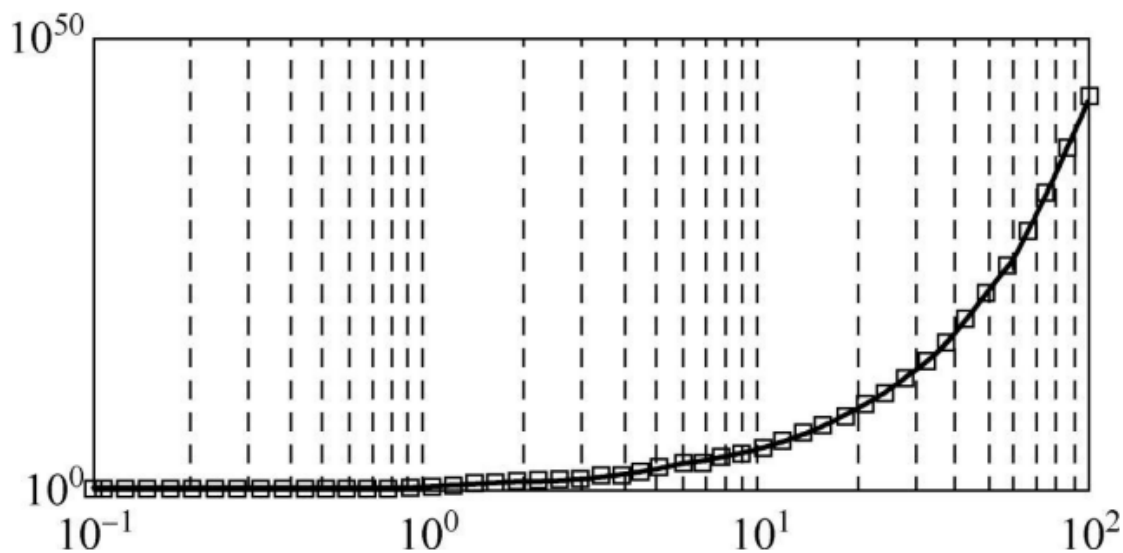


图1-15 对数坐标二维图

3. 绘制双纵坐标曲线图

在MATLAB中，可以使用绘图函数`plotyy`绘制出具有不同纵坐标的两个图形。

调用格式为

```
plotyy(x1,y1,x2,y2,'fun1','fun2')
```

其中`x1, y1`对应一条曲线，`x2, y2`对应另一条曲线。横坐标的标度相同，纵坐标有两个，左纵坐标用于`x1, y1`数据对，右纵坐标用于`x2, y2`数据对。

【例1-55】 双纵坐标图形实现。

运行程序如下：

```
x=0:0.01:5;  
y=exp(x);  
plotyy(x,y,x,y,'semilogy','plot')
```

运行结果如图1-16所示。

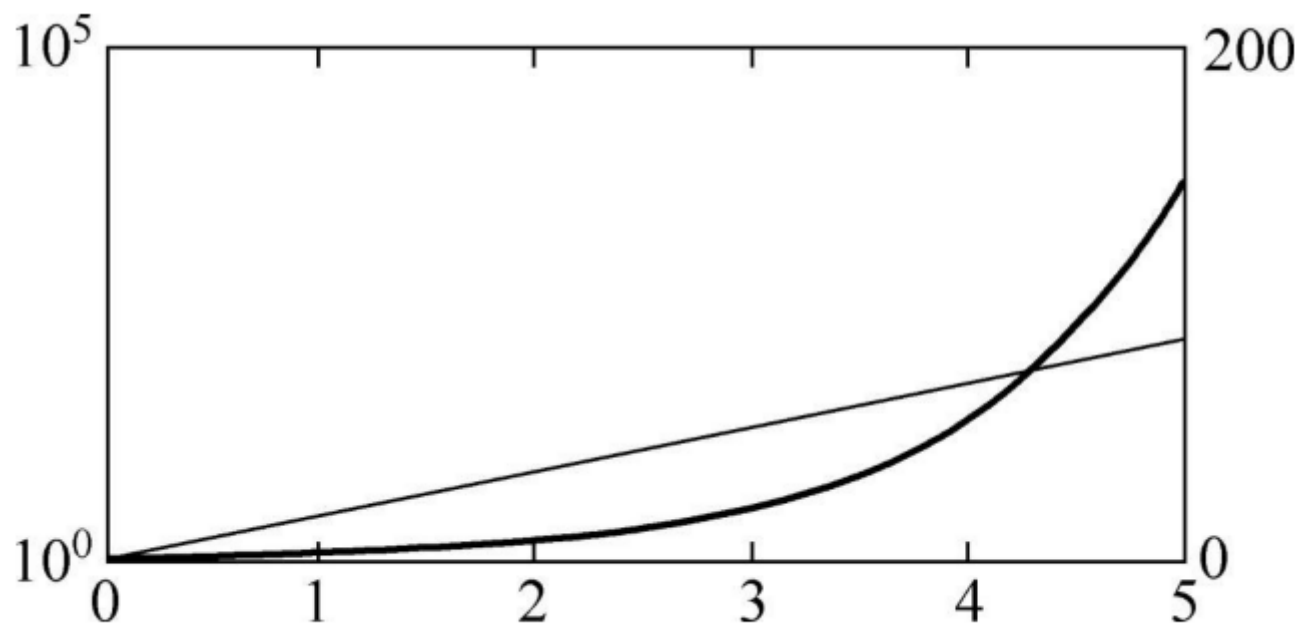


图1-16 双纵坐标图形

4. 绘制其他类型的曲线图

在MATLAB中, 还有其他绘图函数, 可以绘制不同类型的二维图形, 如表1-8所示。

表1-8 其他类型二维图形函数

函 数	二维图的形状	备 注
bar(x, y)	条形图	x是横坐标, y是纵坐标
fplot(y, [a b])	精确绘图	y代表某个函数, [a b] 表示需要精确绘图的范围
polar(θ, r)	极坐标图	θ 是角度, r代表以 θ 为变量的函数
stairs(x, y)	阶梯图	x是横坐标, y是纵坐标
stem(x, y)	针状图	x是横坐标, y是纵坐标
fill(x, y, 'b')	实心图	x是横坐标, y是纵坐标, 'b' 代表颜色
scatter(x, y, s, c)	散点图	x是圆圈标记点的面积, c是标记点颜色
pie(x)	饼图	x为向量

函数polar用来绘制极坐标图, 其调用格式为

```
polar(theta, rho, 选项)
```

其中theta为极坐标极角, rho为极坐标矢径, 选项的内容与函数plot相似。

【例1-56】 绘制 $r=\sin(x)\cos(x)$ 的极坐标图。

运行程序如下:

函数polar用来绘制极坐标图，其调用格式为

```
polar(theta, rho, 选项)
```

其中theta为极坐标极角，rho为极坐标矢径，选项的内容与函数plot相似。

【例1-56】 绘制 $r=\sin(x)\cos(x)$ 的极坐标图。

运行程序如下：

```
t=0:pi/50:2*pi;  
r=sin(t).*cos(t);  
polar(t,r,'-*');
```

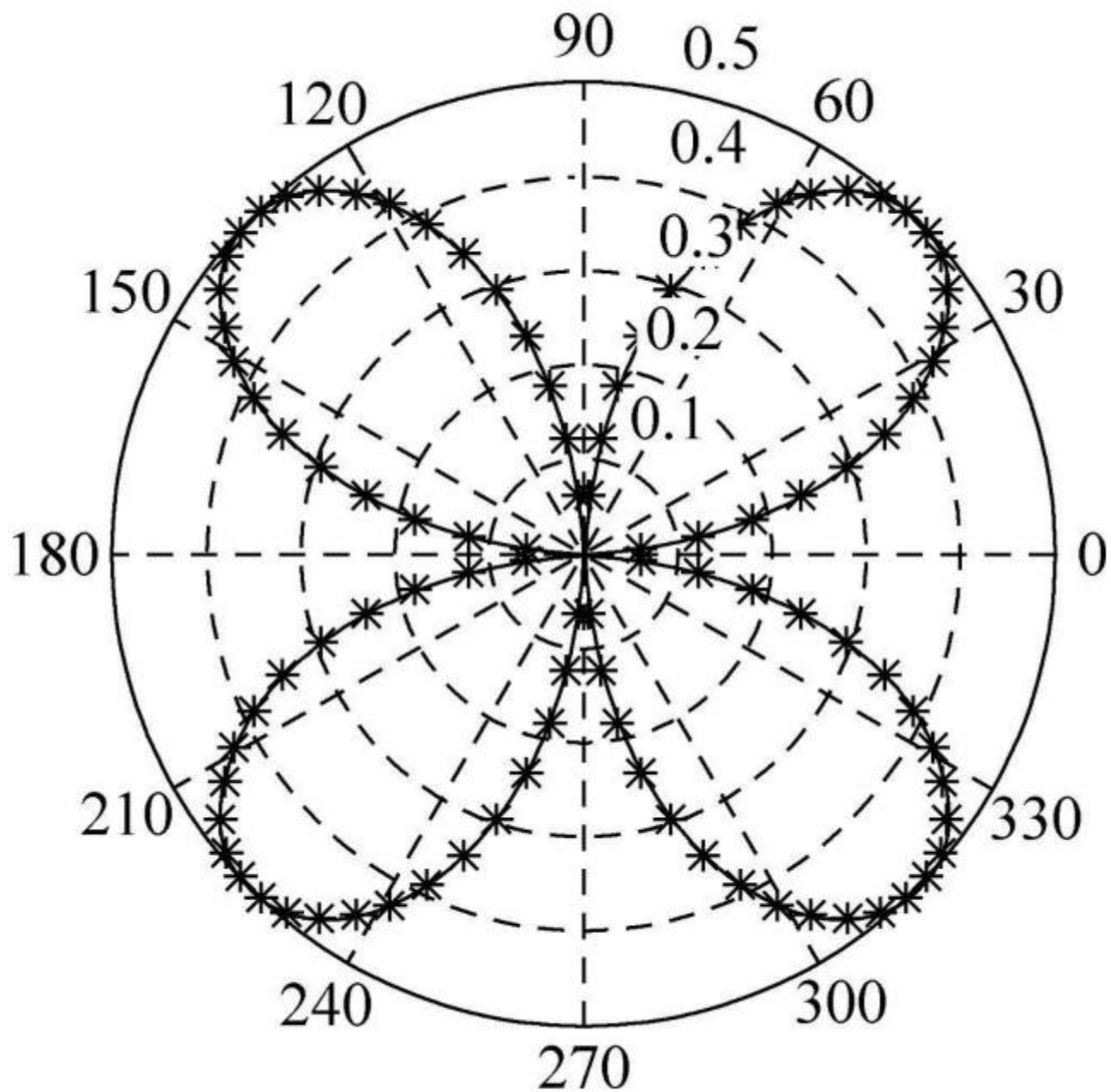


图1-17 极坐标二维图

【例1-57】 其他二维图形画图。

运行程序如下：

```
figure
subplot(221)
x=-2.9:0.2:2.9           %条形图
bar(x,exp(-x.^2))
subplot(222)
x=0:0.1:4                %针状图
y=(x.^0.8). *exp(-x)
stem(x,y)
subplot(223)
x=0:0.25:10              %阶梯图
stairs(x,sin(2*x)+sin(x))
subplot(224)
x=[43 78 88 43 21]      %饼图
pie(x)
```

运行结果如图1-18所示。

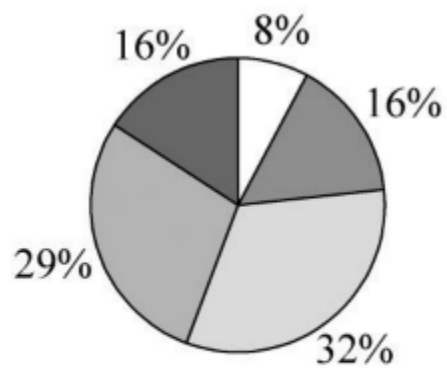
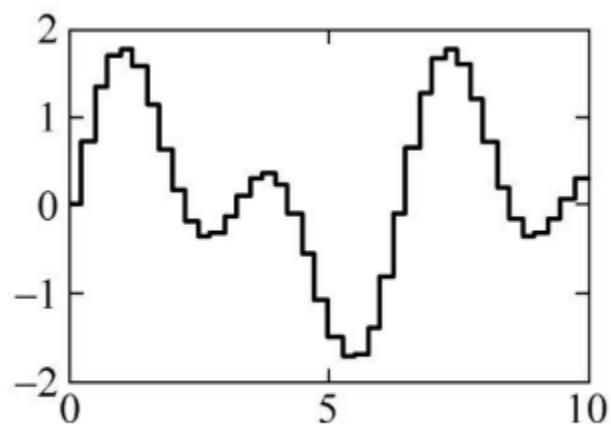
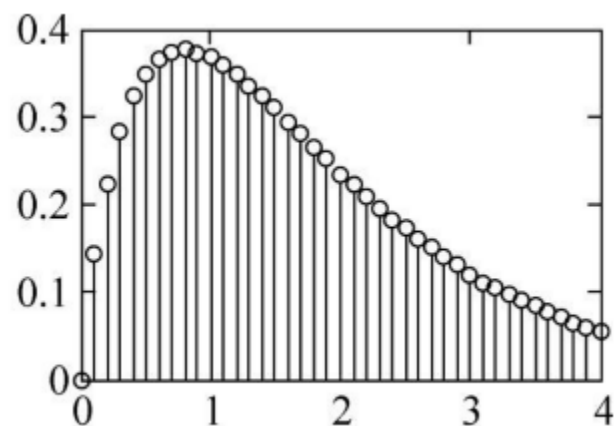
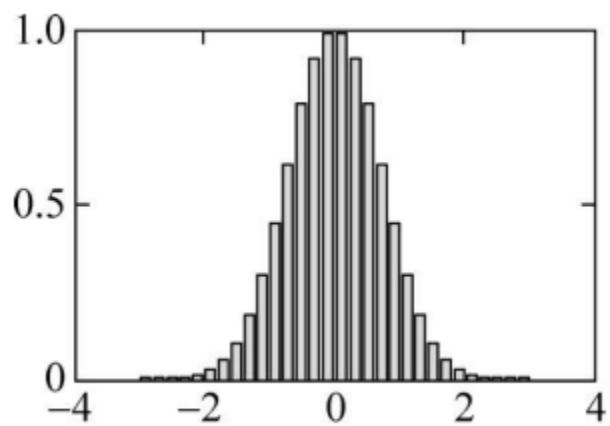
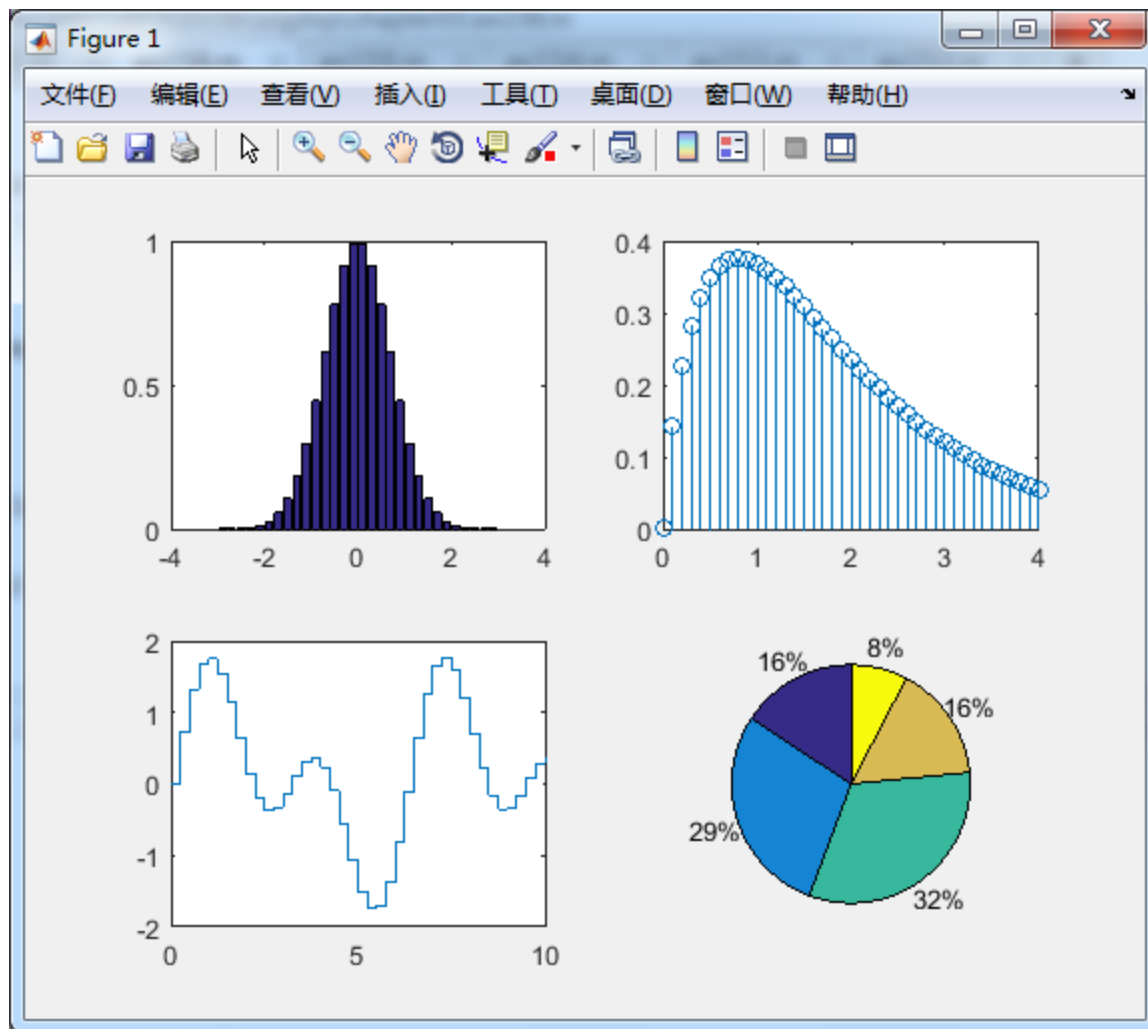


图1-18 其他类型的二维图


```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex158.m
ex158.m x +
1 - figure
2 - subplot(221)
3 - x=-2.9:0.2:2.9;           %条形图
4 - bar(x,exp(-x.^2))
5 - subplot(222)
6 - x=0:0.1:4;             %针状图
7 - y=(x.^0.8).*exp(-x);
8 - stem(x,y)
9 - subplot(223)
10 - x=0:0.25:10;          %阶梯图
11 - stairs(x,sin(2*x)+sin(x))
12 - subplot(224)
13 - x=[43 78 88 43 21];   %饼图
14 - pie(x)
15
```



7、图形绘制

1.7.2 图形绘制和编辑

1. 色彩和线型

在MATLAB中为区别位于同一窗口中的多条曲线，可以改变曲线的颜色和线型等图形属性。plot函数可以接受字符串输入变量，这些字符串输入变量用来指定不同的颜色、线型和标记符号（各数据点上的显示符号），如表1-9所示。

表1-9 plot绘图函数的常用参数颜色参数

颜色参数	颜色	线型参数	线型	标记符号	标 记
y	黄	-	实线	.	圆点
b	蓝	-	虚线	o	圆圈
g	绿	-.	点划线	+	加号
m	洋红(magenta)	:	点线	*	星号
w	白			x	叉号
c	青(cyan)			s 或 'square'	方块
k	黑			d 或 'diamond'	菱形
r	红			^	朝上三角符号
				v	朝下三角符号
				<	朝左三角符号
				>	朝右三角符号
				p	五角星(pentagon)
				h	六角星(hexagon)

【例1-58】 绘制两条不同颜色，不同线型的曲线。

7、图形绘制

- 图形绘制和编辑

ex158.m

ex159.m

ex160.m

ex161.m

ex162.m

ex163.m

ex164.m

ex165.m

ex166.m

ex167.m

ex168.m

ex169.m

ex170.m

ex171.m

ex172.m

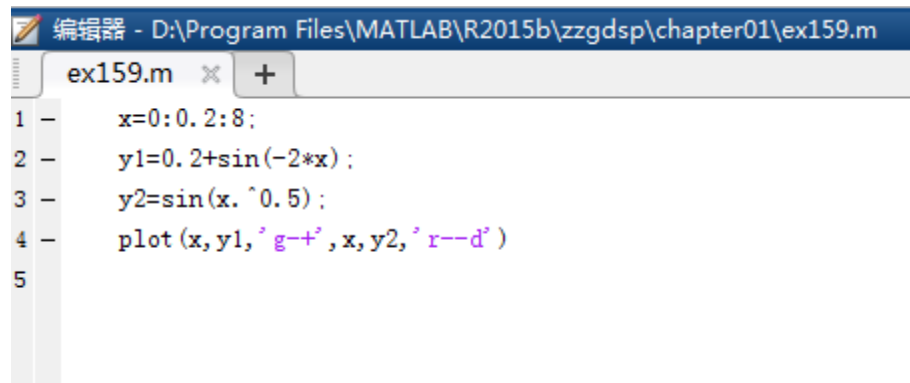
ex173.m

ex174.m

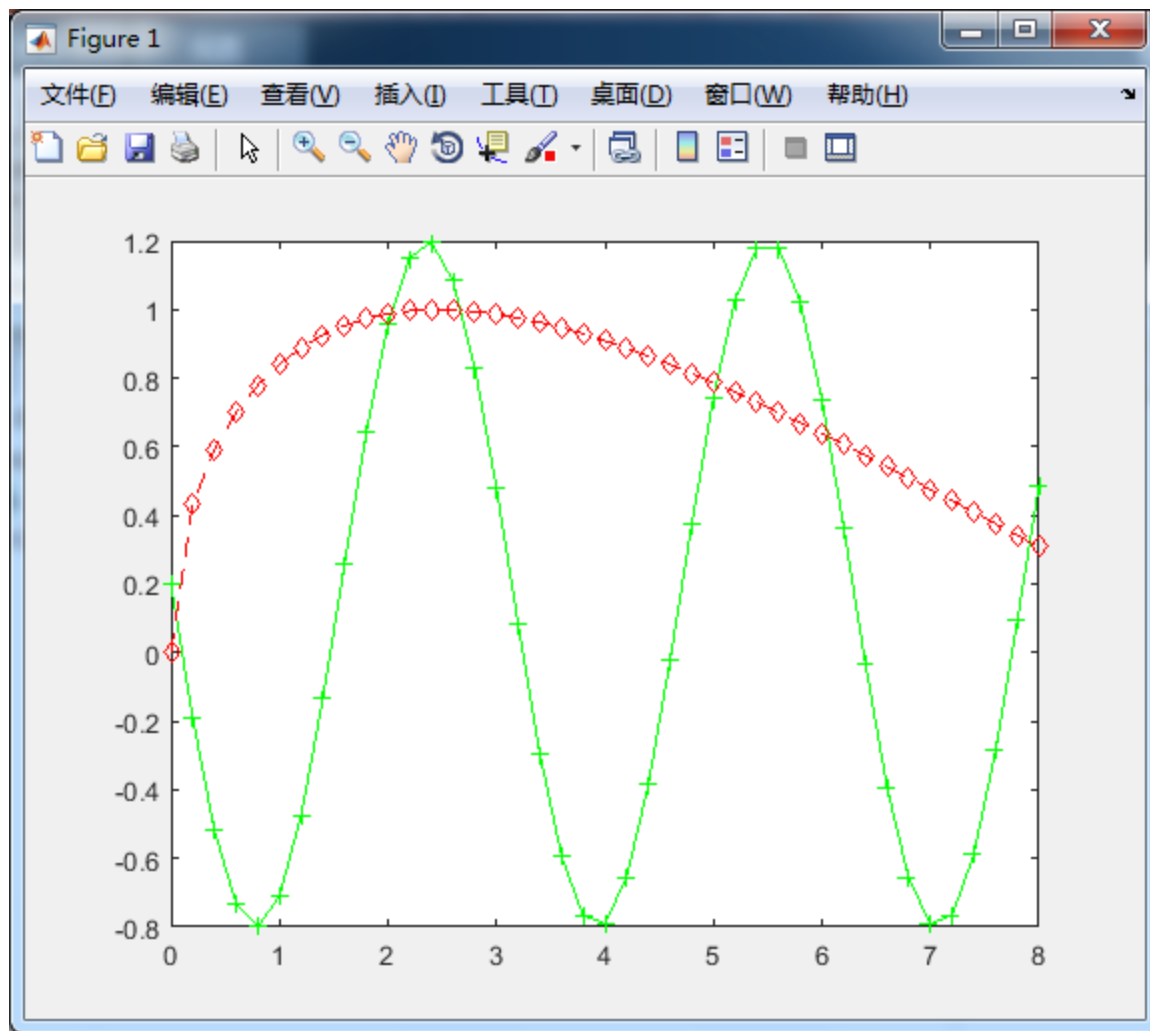
ex175.m

【例1-58】 绘制两条不同颜色，不同线型的曲线。

运行程序如下：



```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex159.m
ex159.m x +
1 - x=0:0.2:8;
2 - y1=0.2+sin(-2*x);
3 - y2=sin(x.^0.5);
4 - plot(x,y1,'g-+',x,y2,'r--d')
5
```



7、图形绘制

- 图形绘制和编辑

- 2. 图形的标住与修饰

在MATLAB中，提供了一些图形函数，专门对所画出的图形进行修饰，以使其更加美观，更便于应用。图形绘制以后，需要对图形进行标注、说明等修饰性的处理，以增加图的可读性，使之反映出更多的信息。下面将分别介绍这些函数。

在MATLAB中，`axis`函数用于根据需要适当调整坐标轴，该函数调用格式是：

`axis([xmin xmax ymin ymax])`：此函数将所画的X轴的大小范围限定在{xmin, xmax}之间，Y轴的大小范围限定在{ymin, ymax}之间。

`axis(str)`：将坐标轴的状态设定为字符串参数str所指定的状态。参数str是由一对单引号所包起来的字符串，它表明了将坐标轴调整为哪一种状态。各种常用字符串的含义如

表1-10所示。

`variable=axis`: 变量`variable`保存的是一个向量值, 显然这个向量值能够以`axis(variable)`的形式应用于设定坐标轴的大小范围。

`[s1, s2, s3] =axis('state')`: 将当前所使用的坐标轴的状态存储到向量 `[s1, s2, s3]` 中。`s1`表示是否自动设定坐标轴的范围, 取值为'auto'或'manual'; `s2`表示是否关闭坐标轴, 取值为'on'或'off'; `s3`表示所使用的坐标轴的种类, 取值为'xy'或'ij'。表1-10给出了函数`axis`的用法。

表1-10 函数`axis`的用法

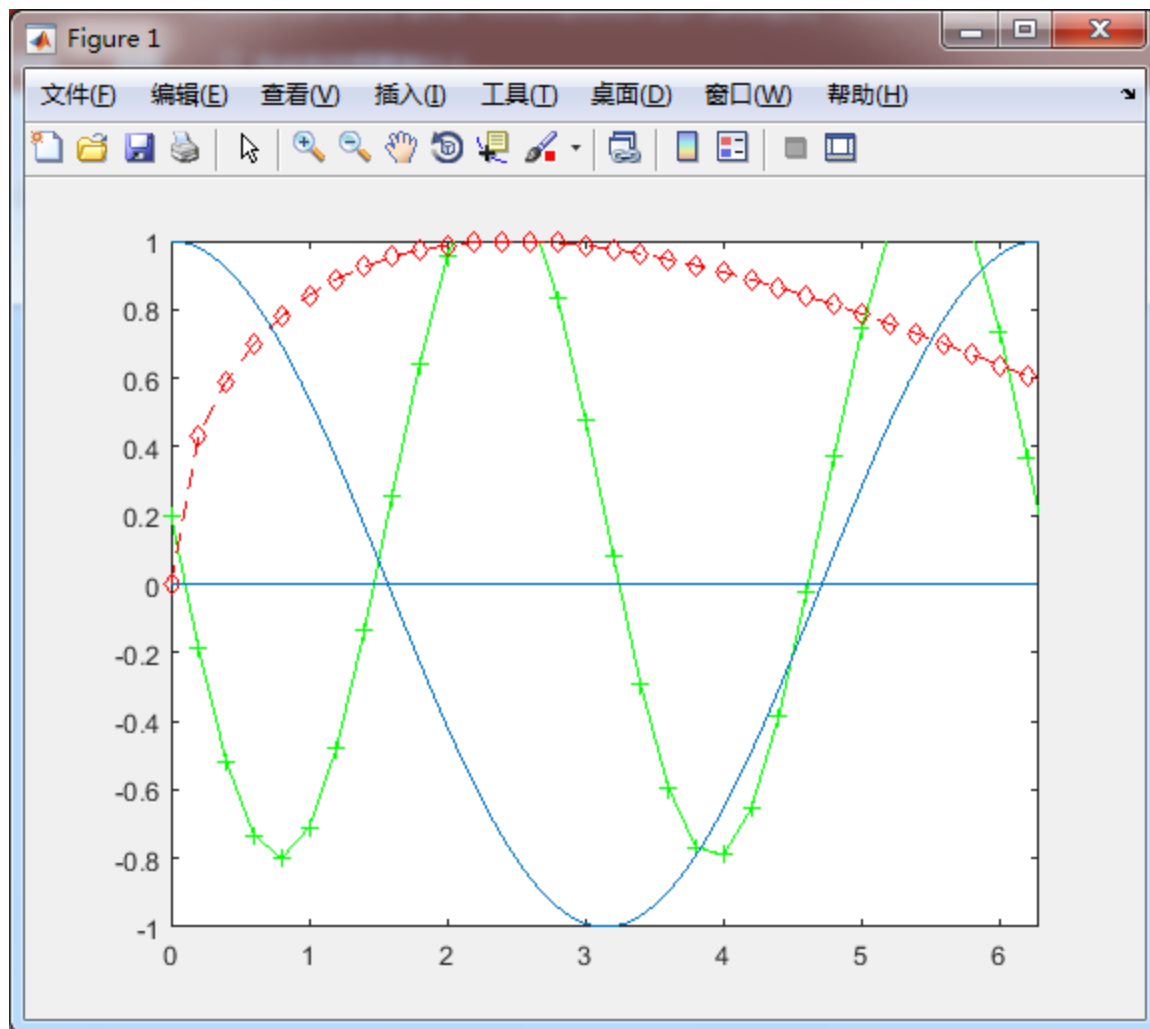
命 令	功 能
<code>axis([xmin xmax ymin ymax])</code>	表示按照用户给出的x轴和y轴的最大、最小值选择坐标系
<code>axis('auto')</code>	表示自动设置坐标系: <code>xmin=min(x)</code> ; <code>xmax=max(x)</code> ; <code>ymin=min(y)</code> ; <code>ymax=max(y)</code> ;
<code>axis('xy')</code>	表示使用笛卡儿坐标系
<code>axis('ij')</code>	表示使用matrix坐标系。即坐标原点在左上方, x坐标从左向右增大, y坐标从上向下增大
<code>axis('square')</code>	表示将当前图形设置为正方形图形
<code>axis('equal')</code>	表示将x, y坐标轴的单位刻度设置为相等
<code>axis('normal')</code>	表示关闭 <code>axis equal</code> 和 <code>axis square</code> 命令
<code>axis('off')</code>	表示关闭网络线、xy坐标的用 <code>label</code> 命令所加的注释, 但保留用图形中 <code>text</code> 命令和 <code>gtext</code> 命令所添加的文本说明
<code>axis('on')</code>	表示打开网络线、xy坐标的用 <code>label</code> 命令所加的注释

【例1-59】 利用函数axis调整 $y=\cos x$ 的坐标轴范围。

```
x=0:pi/100:2*pi;  
y=cos(x);  
line([0,2*pi],[0,0])  
hold on;  
plot(x,y)  
axis([0 2*pi -1 1])
```

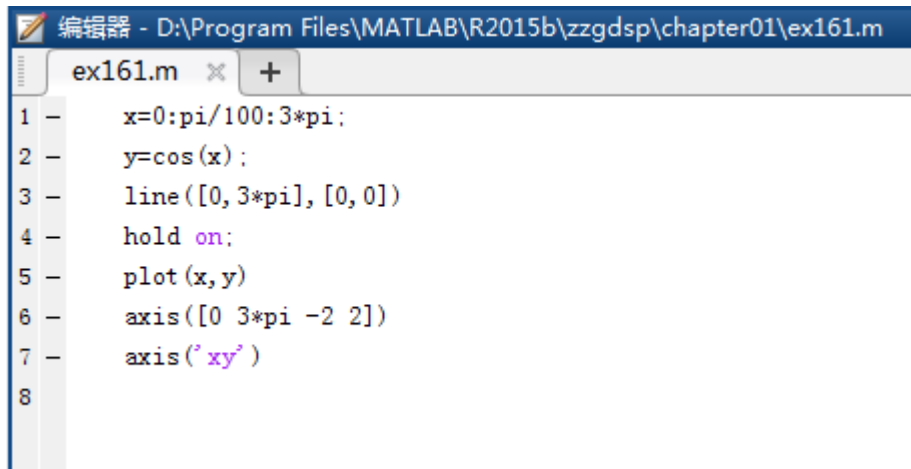
运行结果如图1-20所示。

```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex160.m
ex159.m x ex160.m x +
1 - x=0:pi/100:2*pi;
2 - y=cos(x);
3 - line([0,2*pi],[0,0])
4 - hold on;
5 - plot(x,y)
6 - axis([0 2*pi -1 1])
7 - |
```



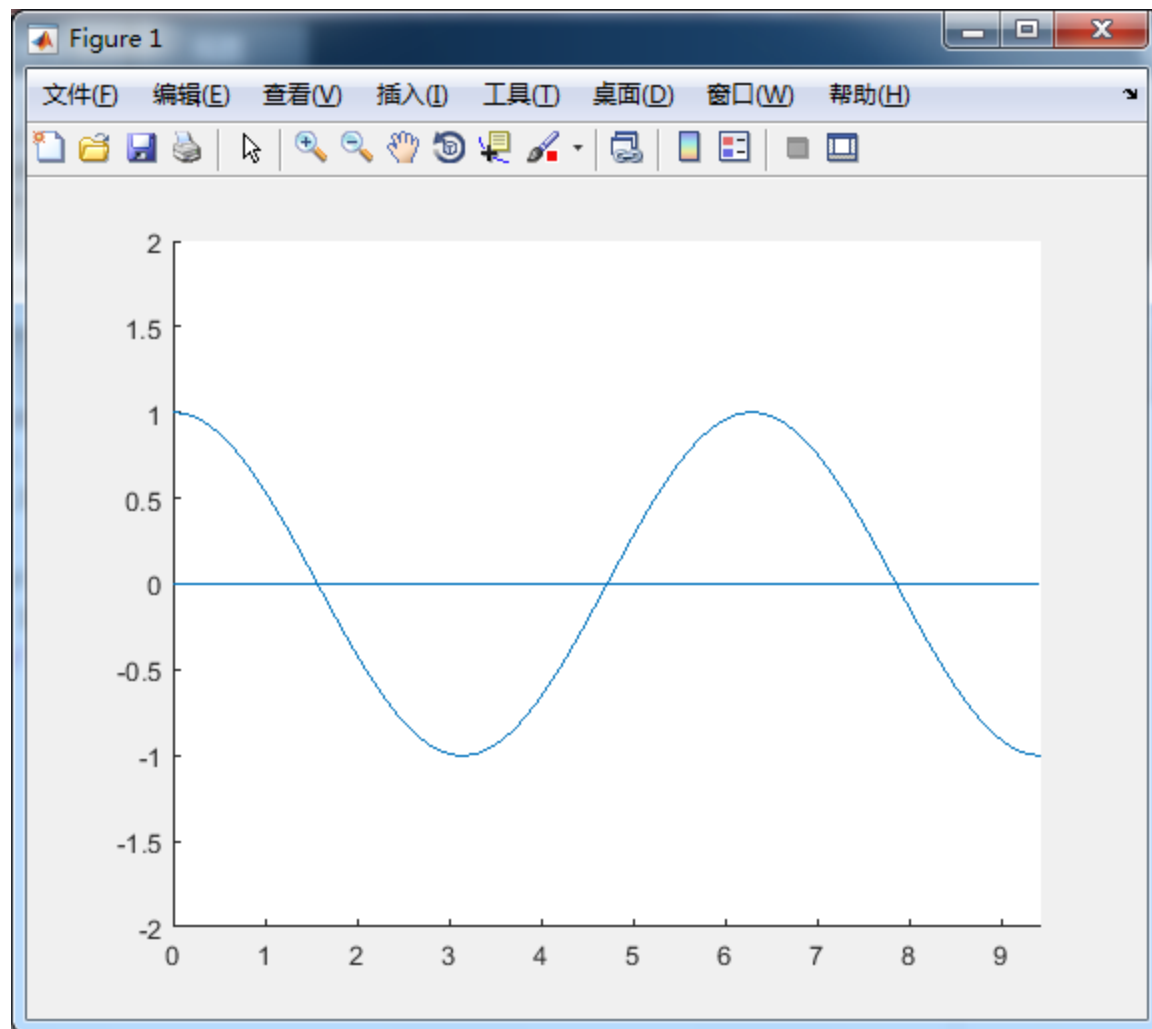
【例1-60】 利用axis函数为 $y=\cos x$ 绘制笛卡儿坐标系。

```
x=0:pi/100:3*pi;  
y=cos(x);  
line([0,3*pi],[0,0])  
hold on;  
plot(x,y)  
axis([0 3*pi -2 2])  
axis('xy')
```



The screenshot shows a MATLAB editor window titled "编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex161.m". The window contains a single tab labeled "ex161.m" with a close button and a plus sign. The code is displayed as follows:

```
1 - x=0:pi/100:3*pi;  
2 - y=cos(x);  
3 - line([0,3*pi],[0,0])  
4 - hold on;  
5 - plot(x,y)  
6 - axis([0 3*pi -2 2])  
7 - axis('xy')  
8
```



7、图形绘制

1.7.3 三维图形的绘图

在MATLAB中绘制三维图形，通常采用以下步骤：

- (1) 数据准备；
- (2) 图形窗口和绘图区选择；
- (3) 绘图；
- (4) 设置视角；
- (5) 设置颜色表；
- (6) 设置光照效果；
- (7) 设置坐标轴刻度和比例；
- (8) 标注图形；
- (9) 保存、打印或导出。

下面我们根据绘制三维图形的基本步骤分别介绍创建图形的各种函数。

7、图形绘制

- 三维图形绘制

ex176.m

ex177.m

ex178.m

ex179.m

ex180.m

ex181.m

ex182.m

ex183.m

ex184.m

ex185.m

1. 三维折线及曲线的绘制

在MATLAB中，plot3命令的功能及使用方法与plot命令的功能及使用方法相似，它们的区别在于前者绘制出的是三维图形。该函数的调用方法如下：

```
plot3(x,y,z)
plot3(x,y,z,option)
```

其中，选项参数option指明了所绘图中线条的线性、颜色以及各个数据点的表示记号。plot3命令使用逐点连线的方法绘制三维折线，当各个数据点的间距较小时，我们也可利用它来绘制三维曲线。

【例1-76】 利用plot3函数绘制一条三维螺旋线。

```
t=0:pi/50:8*pi;
x=sin(t);
y=cos(t);
z=t;
plot3(x,y,z)
```

运行结果如图1-38所示。

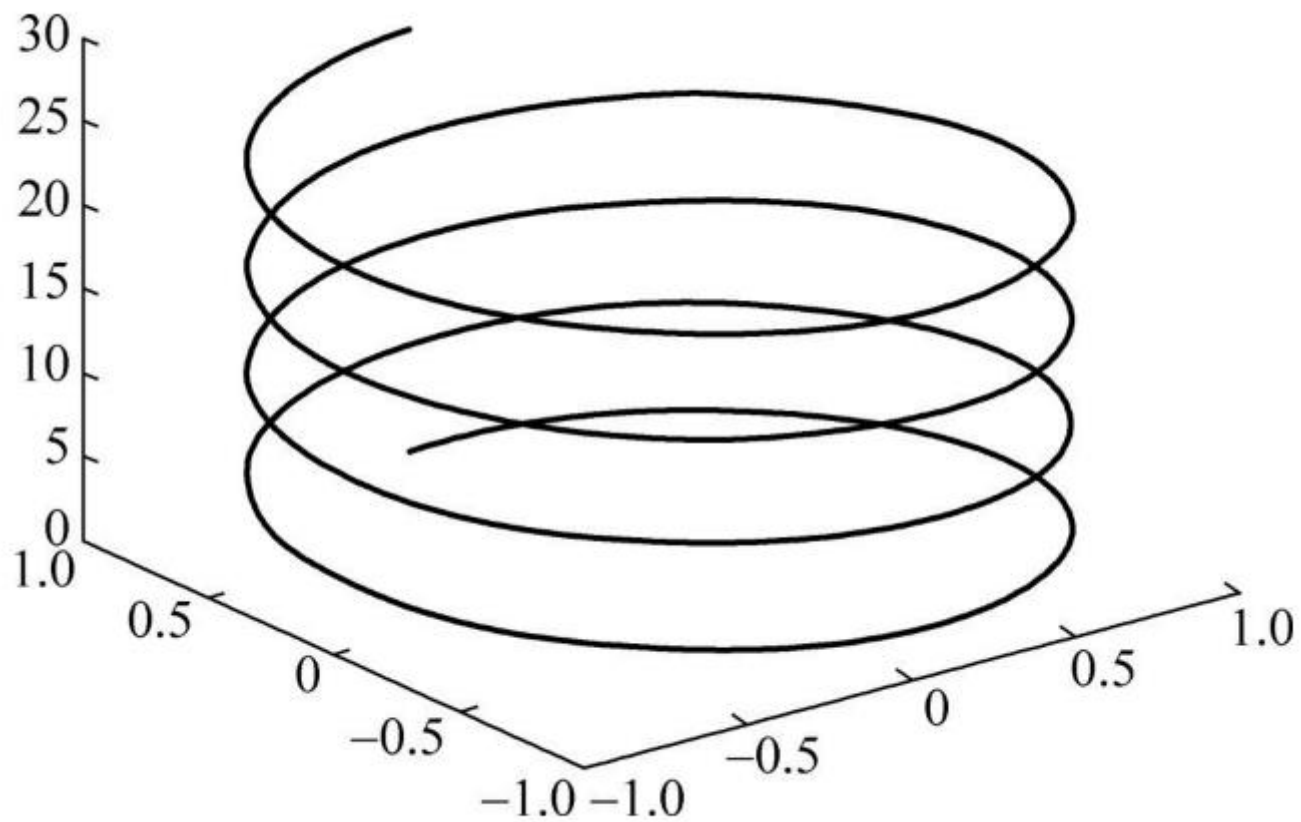


图1-38 三维螺旋线

2. 三维图形坐标标记的函数

MATLAB提供了下述三个用于三维图形坐标标记的函数，并提供了用于图形标题说明的语句。这些函数的调用方法如下：

`xlabel(str)`：将字符串str水平放置于X轴。

`ylabel(str)`：将字符串str水平放置于Y轴。

`zlabel(str)`：将字符串str水平放置于Z轴。

`title(str)`：将字符串str水平放置于图形的顶部。

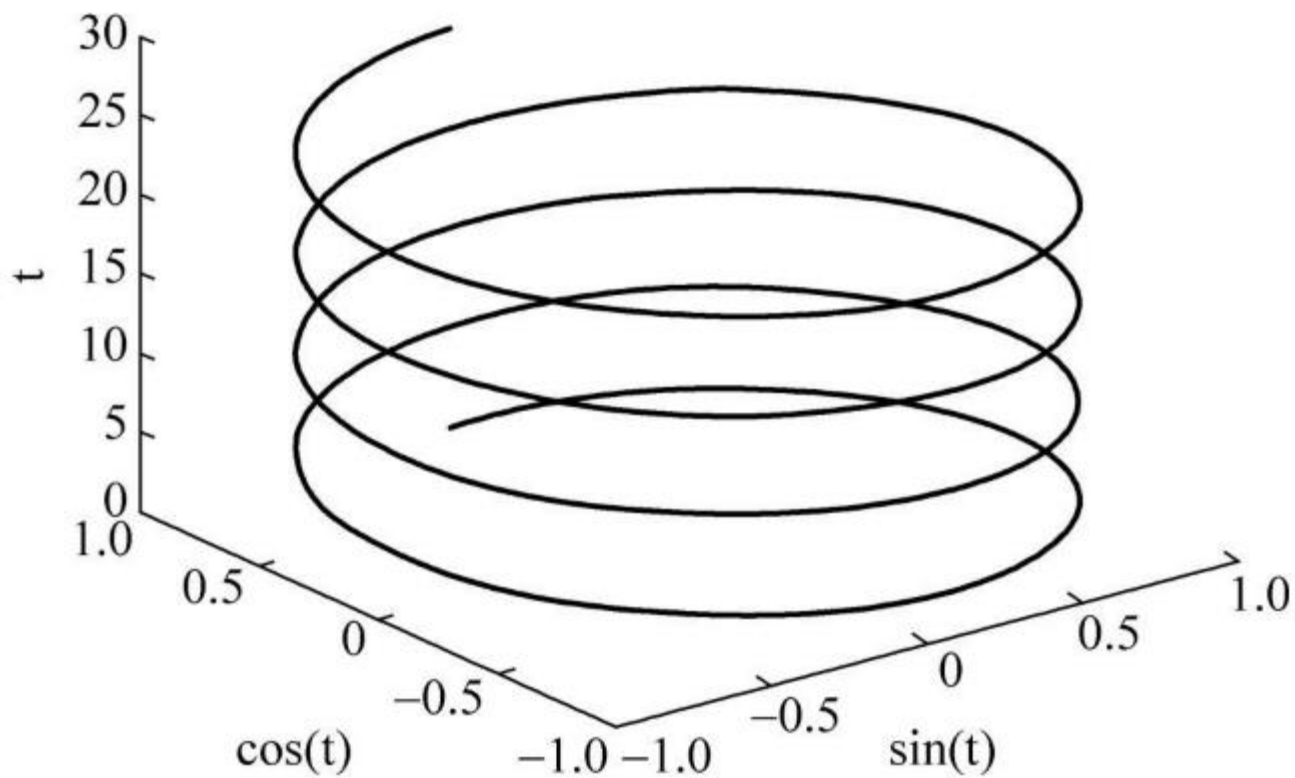
【例1-77】 利用函数为 $x=\sin t$ 、 $y=\cos t$ 的三维螺旋线图形添加标题。

```
t=0:pi/50:8*pi;  
x=sin(t);  
y=cos(t);  
z=t;  
plot3(x,y,z);
```

```
xlabel('sin(t)');  
ylabel('cos(t)');  
zlabel('t');  
title('三维螺旋线');
```

运行结果如图1-39所示。

三维螺旋线



3. 三维网格曲面的绘制

三维网格曲面是由一些四边形相互连接在一起所构成的一种曲面，这些四边形的4条边所围成的颜色与图形窗口的背景色相同，并且无色调的变化，呈现的是一种线架图的形式。在MATLAB中，函数mesh用于栅格数据点的产生，可用于绘制三维网格曲面图；函数hidden用于隐藏线的显示和关闭。这些函数的调用方法如下：

`[X, Y] = meshgrid(x, y)`：表示由x向量和y向量值通过复制的方法产生绘制三维图形时所需的栅格数据X矩阵和Y矩阵。在使用该命令的时候，需要说明以下两点：(1) 向量x和y分别代表三维图形在x轴、y轴方向上的取值数据点；(2) x和y分别是1个向量，而X和Y分别代表1个矩阵。

```
mesh(X, Y, Z, C)
mesh(X, Y, Z)
mesh(x, y, Z, C)
mesh(x, y, Z)
mesh(Z, C)
mesh(Z)
```

其中，在命令格式`mesh(X, Y, Z, C)`和`mesh(X, Y, Z)`中，参数X, Y, Z都为矩阵值，并且X矩阵的每一个行向量都是相同的，Y矩阵的每一个列向量也都是相同的。参数C表示网格曲面的颜色分布情况，若省略该参数则表示网格曲面的颜色分布与Z方向上的高度值成正比；在命令格式`mesh(x, y, Z, C)`和`mesh(x, y, Z)`中，参数x和y为长度分别是n和m向量值，而参数Z是维数为 $m \times n$ 的矩阵；在命令格式`[Z, C]`和`mesh(Z)`中，若参数Z是维数为 $m \times n$ 的矩阵，则绘图时的栅格数据点的取法是`x=1:n`和`y=1:m`。另外，MATLAB中还有两个mesh的派生函数：函数`meshc`用于在绘图的同时，在X-Y平面上绘制函数的等值线；函数`meshz`则用于在网格图基础上，在图形的底部外侧绘制平行z轴的边框线。

`hidden on`：表示去掉网格曲面的隐藏线。

`hidden off`：表示显示网格曲面的隐藏线。

【例1-79】 在笛卡儿坐标系中绘制函数的网格曲面图。

```
x=-7:0.5:7;  
y=x;  
[X,Y]=meshgrid(x,y);  
Q=sqrt(X.^+Y.^)+eps;  
Z=cos(Q) ./Q;  
mesh(X,Y,Z)  
grid on  
axis([-10 10 -10 10 -1 1 ])
```

运行结果如图1-41所示。

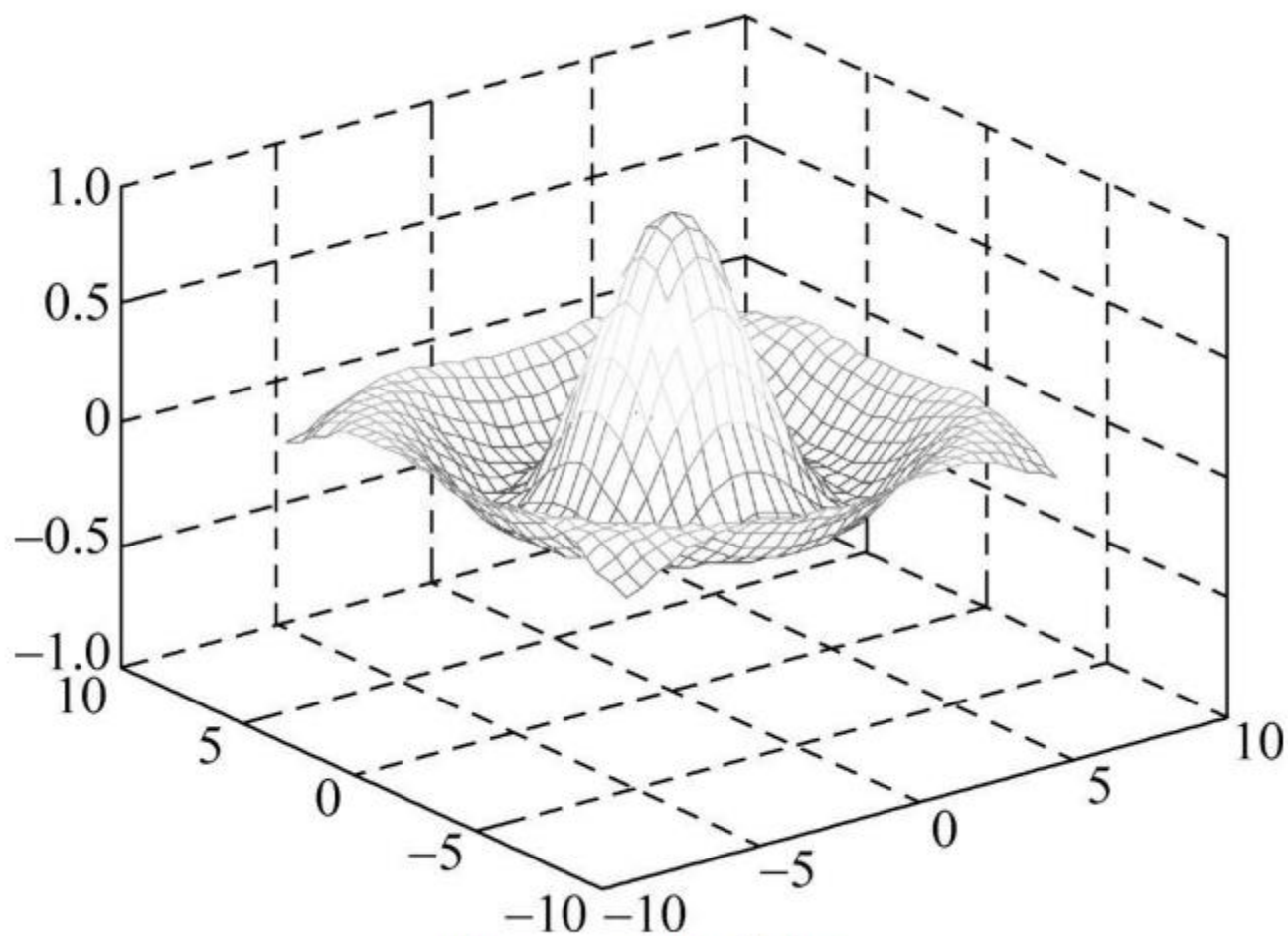
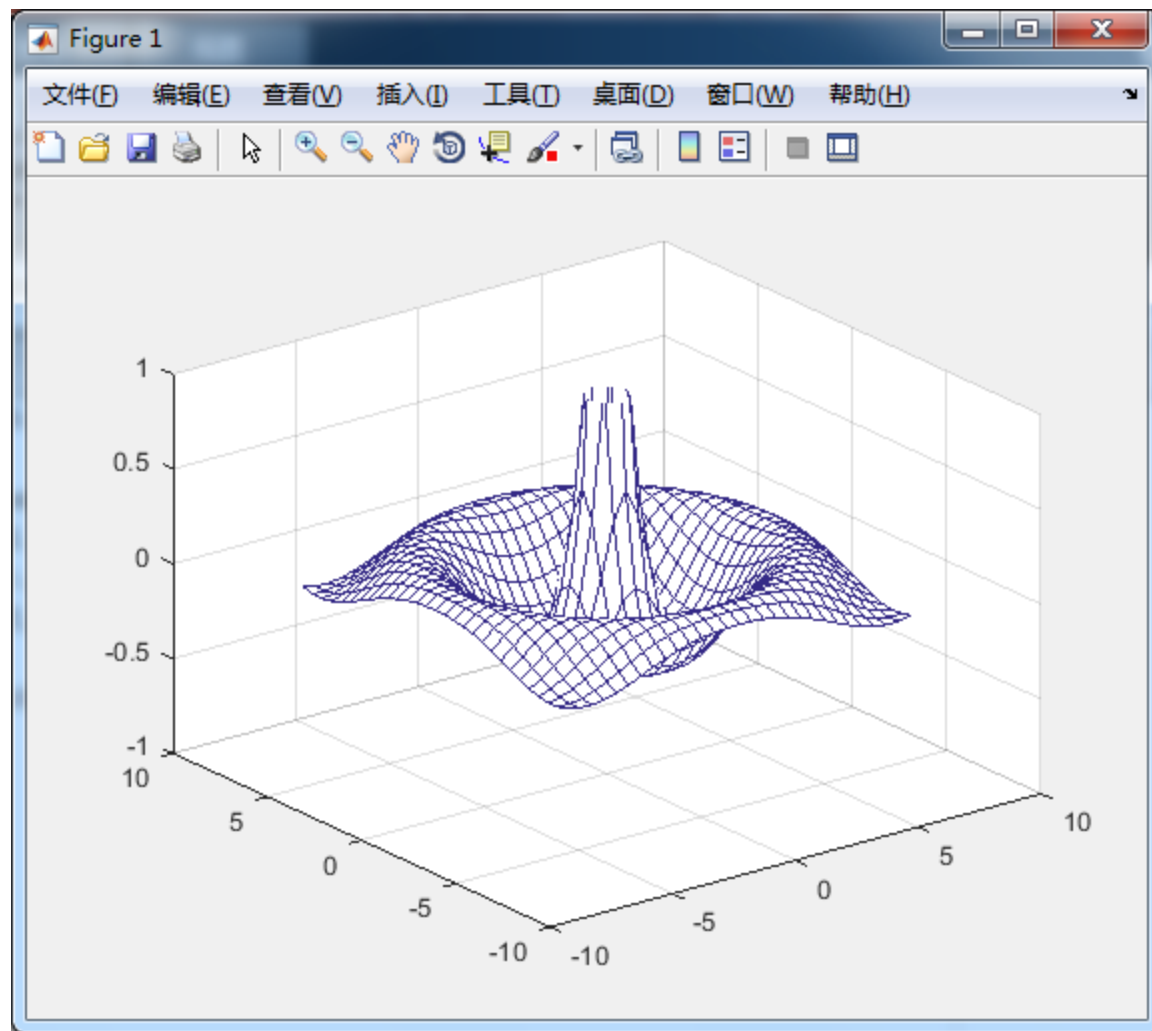


图1-41 函数的网格曲面

```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex180.m
ex161.m x ex181.m x ex179.m x ex180.m x +
1 - x=-7:0.5:7;
2 - y=x;
3 - [X,Y]=meshgrid(x,y);
4 - Q=sqrt(X.^2+Y.^2)+eps;
5 - Z=cos(Q)./Q;
6 - mesh(X,Y,Z)
7 - grid on
8 - axis([-10 10 -10 10 -1 1 ])
9
```



【例1-80】 利用meshc和meshz绘制三维网格图。

```
close all  
clear
```

```
[X,Y] = meshgrid(-2:. 4:2);  
Z = 2*X.^-3*Y.^;  
subplot(2,2,1)  
plot3(X,Y,Z)  
subplot(2,2,2)  
mesh(X,Y,Z)  
subplot(2,2,3)  
meshc(X,Y,Z)  
subplot(2,2,4)  
meshz(X,Y,Z)
```

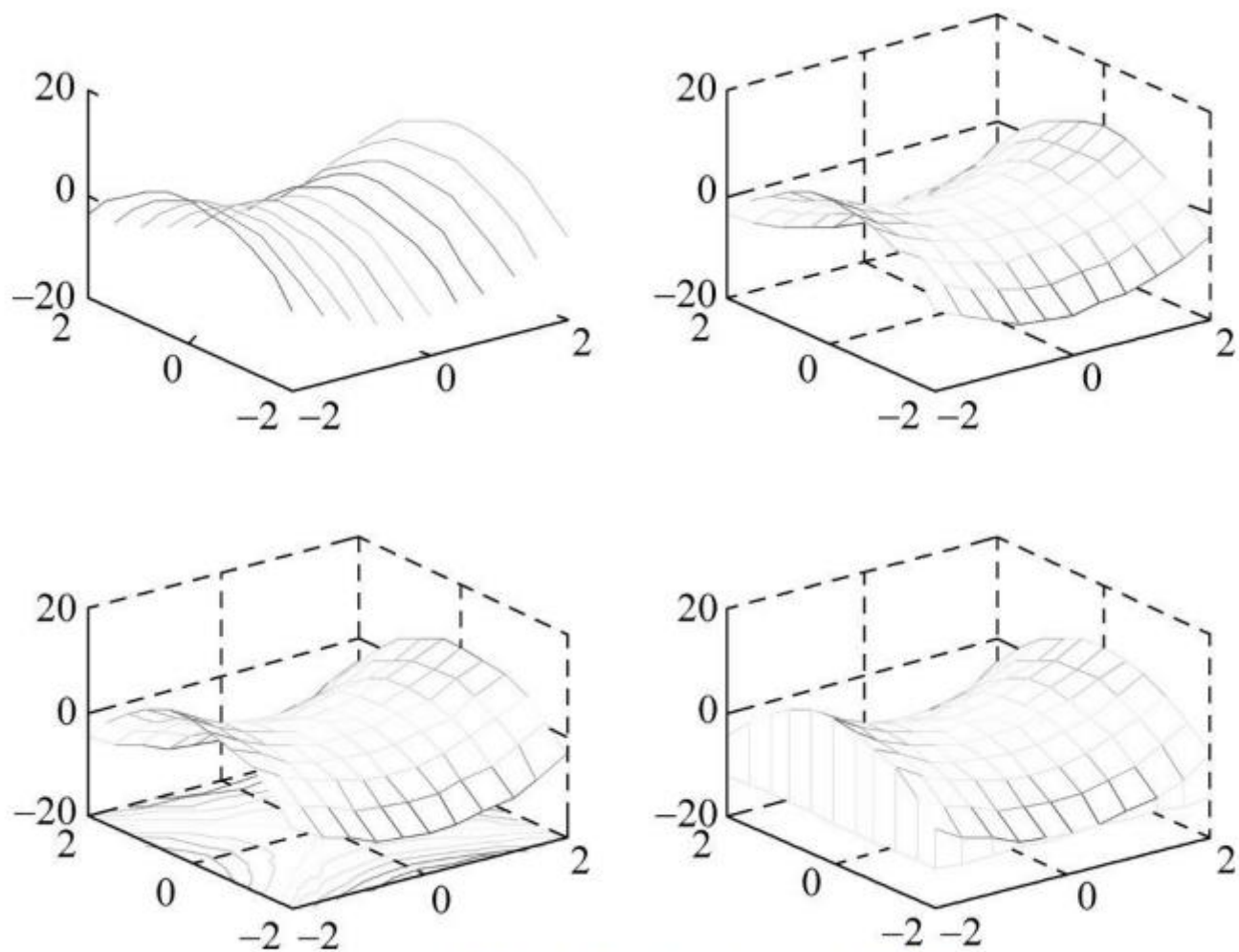
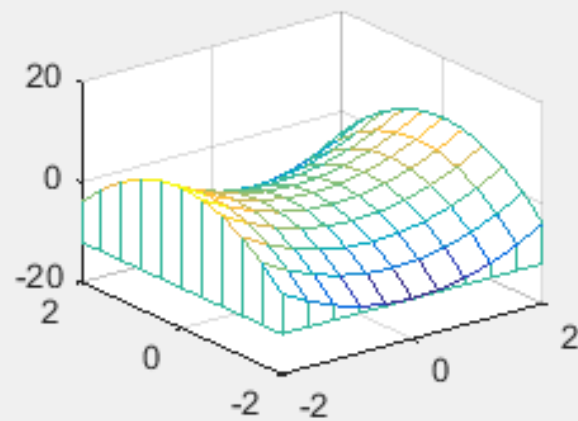
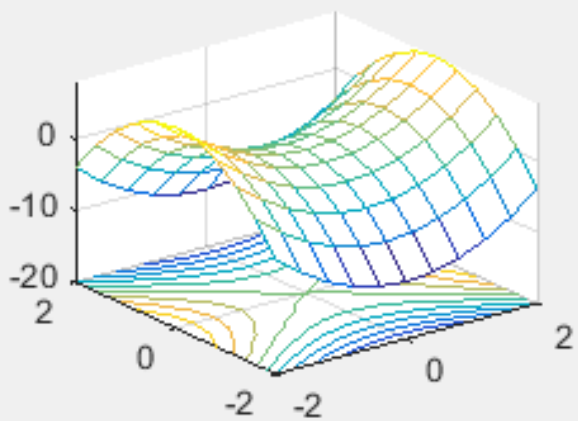
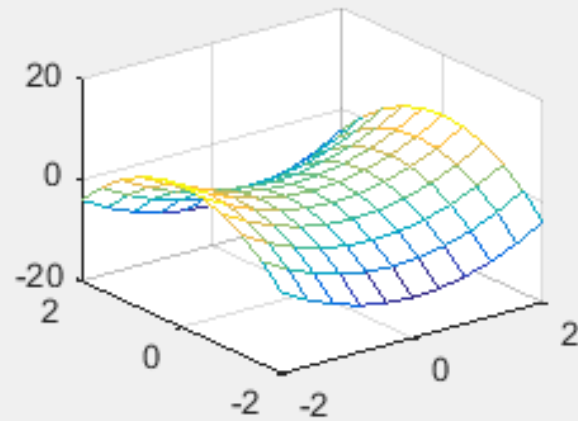
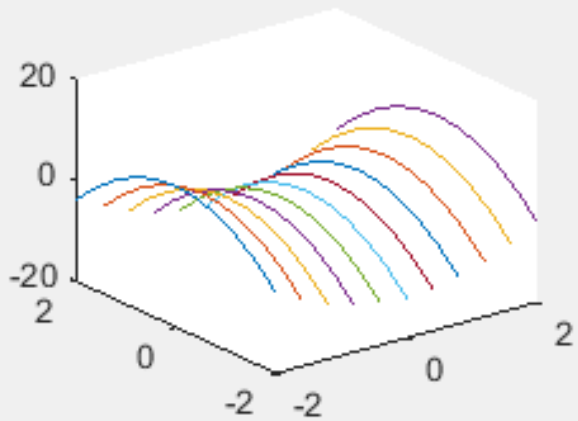


图1-42 利用meshc和meshz绘制的三维网格

```
编辑器 - D:\Program Files\MATLAB\R2015b\zzgdsp\chapter01\ex181.m
ex161.m x ex181.m x +
1 - close all
2 - clear
3 - [X,Y] = meshgrid(-2:.4:2);
4 - Z = 2*X.^2-3*Y.^2;
5 - subplot(2,2,1)
6 - plot3(X,Y,Z)
7 - subplot(2,2,2)
8 - mesh(X,Y,Z)
9 - subplot(2,2,3)
10 - meshc(X,Y,Z)
11 - subplot(2,2,4)
12 - meshz(X,Y,Z)
13
```

Figure 1

文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)



4. 三维阴影曲面的绘制

三维阴影曲面也是由很多个较小的四边形构成的，但是各个四边形的4条边是无色的(即为绘图窗口的底色)，其内部却分布着不同的颜色，也可认为是各个四边形带有阴影效果。

MATLAB提供了3条用于绘制这种三维阴影曲面的命令：函数surf和surfc用于基本的三维阴影曲面的绘制；函数surfl用于具有光照效果的阴影曲面绘制。这些函数的调用方法如下：

```
surf(X,Y,Z,C)
surf(X,Y,Z)
surf(x,y,Z,C)
surf(x,y,Z)
surf(Z,C)
surf(Z)
```

其中，surf命令与mesh命令的使用方法及参数含义相同，其区别是前者绘制的是三维阴影曲面，而后者绘制的是三维网格曲面。在surf命令中，各个四边形表面的颜色分布方式可由shading命令来指示。

shading faceted: 表示截面式颜色分布方式。

shading interp: 表示插补式颜色分布方式。

shading flat: 表示平面式颜色分布方式。

```
surfc(X,Y,Z,C)
surfc(X,Y,Z)
surfc(x,y,Z,C)
surfc(x,y,Z)
surfc(Z,C)
surfc(Z)
```

其中，`surfc`命令与`surf`命令的使用方法及参数含义相同，其区别是前者除了绘制出三维阴影曲面外，在XY坐标平面上还绘制有曲面在Z轴方向上的等高线，而后者仅绘制出三维阴影曲面。

```
surf1(X,Y,Z,s)
surf1(X,Y,Z)
surf1(Z,s)
surf1(Z)
```

其中，这4种`surf1`命令与前面介绍的`surf`命令的使用方法及参数含义相类似，其区别是前者绘制出的三维阴影曲面具有光照效果，而后者绘制出的三维阴影曲面无光照效果。向量参数`s`表示光源的坐标位置，`s=[sx,xy,xz]`。注意，若默认`s`，则表示光源位置设在观测角的反时针 45° 处，它是默认的光源位置。

【例1-81】 采用函数`shading faceted`来设置三维阴影曲面效果。

```
x=-7:0.5:7;
y=x;
[X,Y]=meshgrid(x,y);
Q=sqrt(X.^+Y.^)+eps;
Z=2*cos(Q). /Q;
surf(X,Y,Z)
grid on
axis([-10 10 -10 10 -0.5 1.5])
shading faceted
```

运行结果如图1-43所示。

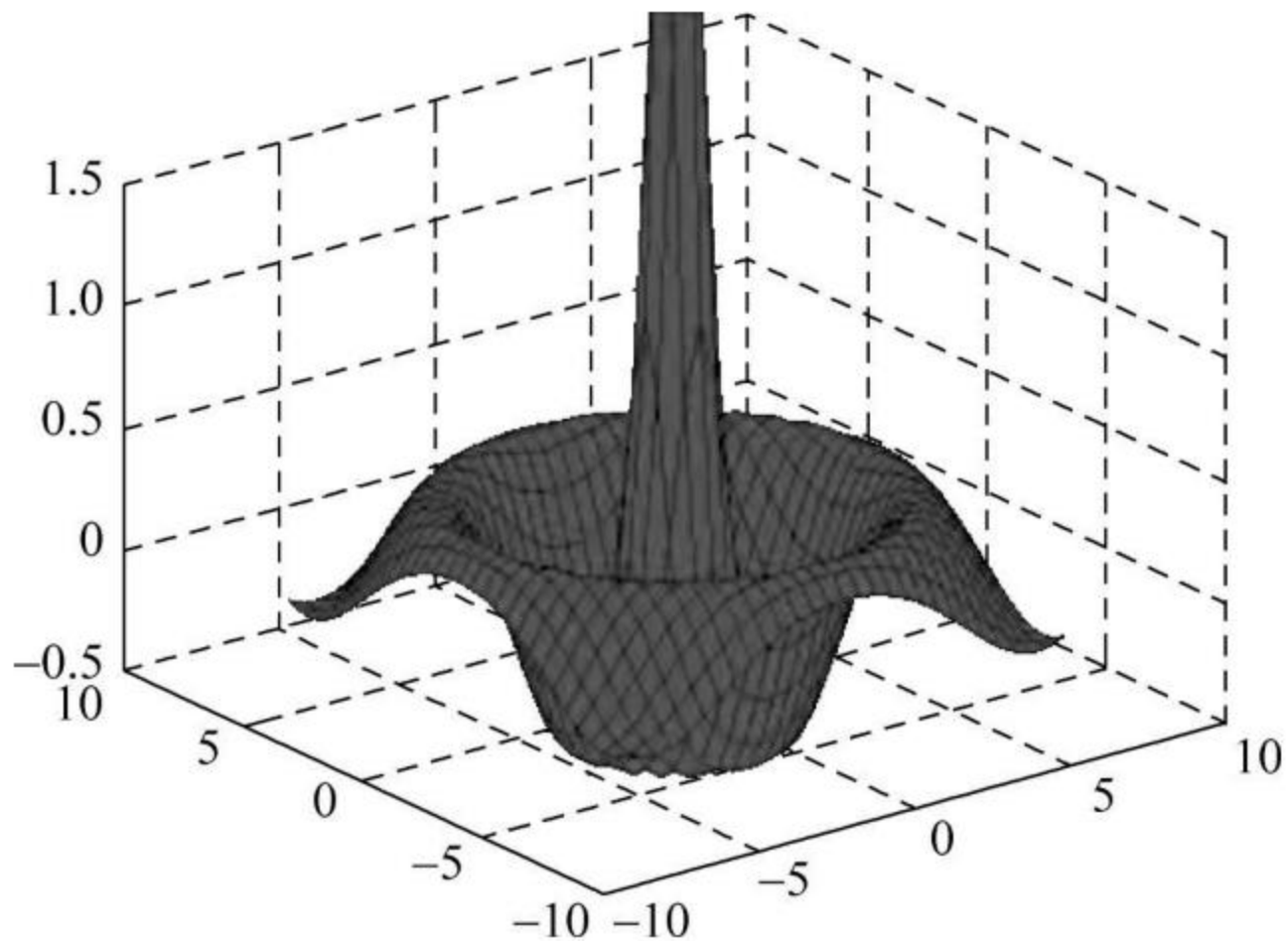


图1-43 截面式颜色分布方式

本章小结

本章首先介绍MATLAB的产生和发展历程及其特点，接着讲述了桌面操作结构、查询帮助命令等相关内容。希望读者仔细阅读，可以对MATLAB这个功能强大的数学软件有一个大致了解，为后面核心技术与工程应用的学习做好铺垫。

MATLAB提供了一套完整而易于使用的编程语言，本章首先介绍了M文件的相关知识，接着讲述了MATLAB程序控制结构、程序流程控制语句及其他常用命令，最后介绍了变量、数值及表达式。读者要掌握好本章的内容，还需要更多的知识，包括相关书籍和MATLAB帮助文件等。

MATLAB为工程人员提供了方便、强大的数值计算功能。同一般的计算机语言不同的是，MATLAB是一种边解释边执行的语言，因此工程人员用其解决实际数学问题，并不需要更多的编程技术。

数据可视化的目的在于通过图形从一堆杂乱的离散数据中观察数据间的内在关系，感受由图形所传递的内在本质。本实验要求结合信号系统专业知识绘制时域频域图，进一步了解图形绘制功能。本章着眼于MATLAB的图形命令，主要内容包括：绘图步骤，二维和三维图形的创建，参数编辑和图形编辑窗口，图形的输出等。