第六章 复数数据、字符数据和附加画图类型

在第二章中，我们学习了 **MATLAB** 基础数据类型：double 和 char。**MATLAB** 还有许多的附加数据类型，在本章，我们将会了解它们中的一个。我们要讨论的附加数据类型是 **MATLAB** 支持的复数数据。我们也将学习如何使用 char 数据类型，以及如何把 **MATLAB** 数组扩展为多维数组。

本章还会涉及到 **MATLAB** 的附加画图类型。

6.1 复数数据

复数是指既包含实部又包含虚部的数。复数出现在许多的科研工作问题上。例如，在电器工程中，我们可以用复数代表交变电压，交变电流和阻抗。描述电器系统行为的公式经常用到复数。因为这是非常常见的，作为一个程师如果没有很好理解和运用复数，它无法工作。

复数的一般形式如下：

C=a+bi

其中 C 为复数，a 和 b 均为实数，i 代表$\sqrt{-1}$。a，b 分别为 C 的实部和虚部。由于复数有两个部分，所以它能在平面内标出。这个平面的横轴是实轴，纵轴是虚轴，所以复数在这个平面内为一个点，横轴为 a，纵轴为 b。用上面的方式表示一个复数，叫做直角坐标表示，为坐标的横轴与虚轴分别代表复数的实部与虚部。

复数有在一平面内另一种表达方式，既极坐标表示，公式如下，

$c = a + bi = z\angle\theta$

其中 z 代表向量的模，θ 代表辐角。直角坐标中的 a，b 和极坐标 z，θ 之间的关系为

$$a = z\cos\theta \tag{6.2}$$

$$b = z\sin\theta \tag{6.3}$$

$$z = \sqrt{a^2+b^2} \tag{6.4}$$

$$\theta = \tan^{-1}\frac{b}{a} \tag{6.5}$$

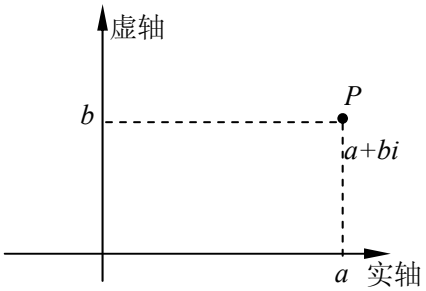


图 6.1 直角坐标系中复数

图 6.2 极坐标系中复数

MATLAB 用直角坐标表达复数。每一个复数应有一对实数（a，b）组成。第一个数（a）代表复数的实部，第二个数（b）代表复数的虚部。

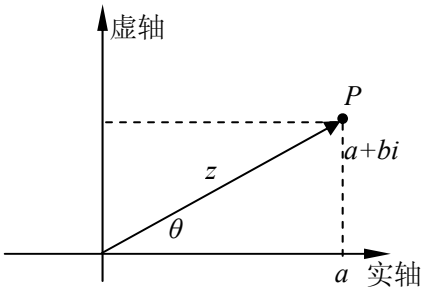
如果复数 $c_1=a_1+b_1i$ 和复数 $c_2=a_2+b_2i$，那么它们的加减乘除运算定义如下。

$$c_1 + c_2 = (a_1 + a_2) + (b_1 + b_2)i \quad (6.6)$$

$$c_1 - c_2 = (a_1 - a_2) + (b_1 - b_2)i \quad (6.7)$$

$$c_1 \times c_2 = (a_1a_2 - b_1b_2) + (a_1b_2 + b_1a_2)i \quad (6.8)$$

$$\frac{c_1}{c_2} = \frac{(a_1a_2+b_1b_2)}{(a_1^2 + b_2^2)} + \frac{(b_1a_2-a_1b_2)}{(a_2^2+b_2^2)}\ i \quad (6.9)$$



当两个复数进行二元运算，**MATLAB** 将会用上面的法则进行加法，减法，乘法和除法运算。

6.1.1 复变量（complex variables）

当复数值赋值于一个变量名，**MATLAB** 将自动创建一个复变量。创建复数的最简单方法是用 **MATLAB** 本自带的因有变量 i 或 j，它们都被预定义为$\sqrt{-1}$ 。例如下面的语句将复数 4+3i 赋值于 c_1。

```
>> c1 = 4 + 3*i
c1 =
    4.0000 + 3.0000i
```

函数 isreal 可以判断一个数组包是实数组还是复数组。如果一个数组中的所有元素只有虚部，那么这个数组是复数组，并且 isreal(array)将会返回一个 0。

6.1.2 带有关系运算符的复数的应用

用关系运算符==来判断两复数是否相等，或用关系运算符~=判断两复数是否不相等，这种情况是可能的。这些运算都会产生出我们所期望的结果。例如，如果 c1=4+3i 和 c2=4-3i，那么关系运算 c1==c2 将会产生 0，关系运算 c1~=c2 将会产生 1。

但是，比较运算符>，<，<=或>=将不会产生我们所期望的结果。当复数进行此类关系运算时，只对复数的实部进行比较。例如，如果 c1=4+i3 和 c2=4+i8，那么比较运算 c1>c2 将会产生 1，尽管 c1 的模要比 c2 的模小。

如果我们需要用这些运算对两复数进行比较，我们更加关心的是两复数的模，而不只是实部。复数的模可以由 abs 固有函数计算得到（在下一节介绍，或者由公式（6.4）得到）。

$$|c| = \sqrt{a^2+b^2} \quad (6.4)$$

如果我们对两复数进行比较，得到的结果将更加合理。abs(c1)>abs(c2)将会产生 0，因为 c1 的模大于 c2 的模。

常见编程错误

当我们应用关系运算符对复数运算时，一定要小心。关系运算符>，<，<=或>=只比较复数的实部，而不是它们的模。如果你要用这些关系运算符对一复数进行运算，比较两复数的模将更加常见。

6.1.3 复函数（complex function）

MATLAB 中有许多的函数支持复数的运算。这些函数可分为三大类。

1. 类型转换函数

这些函数把数据从复数据类型转换为实数数据类型（double）。函数 real 将复数的实部转化为 double 型数据，把复数的虚部抛弃。函数 imag 把函数的虚部转化为相应的实数。

函数	描述
conj(c)	计算 c 的共共轭复数。如果 c=a+bi，那么 conj(c)=a-bi。
real(c)	返回复数 c 的实部
imag(c)	返回复数 c 的虚部
isreal(c)	如果数组 c 中没有一个元素有虚部，函数 isreal(c)将返回 1。所以如果一个数组 c 是复数组成，那么~isreal(c)将返回 1。
abs(c)	返回复数 c 模
angle(c)	返回复数 c 的幅角，等价于 atan2(imag(c)，real(c))

表 6.1 常见的支持复数运算的 **MATLAB** 函数

2. 绝对值和幅角函数

这些函数把复数转化它的极坐标形式。函数 abs(c)用于计算复数 c 相应的绝对值，公式如下

$abs(c)=\sqrt{a^2+b^2}$

其中 c=a+bi。函数 angle(c)用下面的公式计算复数 c 的幅角

angle(c)=atan2(imag(c), real(c))

由它产生的角的取值范围为$-\pi<\theta\leq\pi$。

3. 数学函数

许多的数函数都可以对复数进行运算。这些函数包括指数函数，对数函数，三角函数，还有平方根函数。函数 sin，cos，log，sqrt 等既能对复数数据进行运算，又能对实数据进行运算。

一些支持复数运算的函数在表 6.1 中列出。

例 6.1

二次方程的求解（重写）

复数的价值体现在它能使运算简化。

例如，我们在例 3.2 中已解决的二次方程的求解问题，但它根据判别式用到 3 个选项的选择结构，由于复数的出现，负数的平方根的处理将不困难。所以能够大大简化我们的计算。编写一个普通的程序，解一元二次方程的根，不管是什么类型的。用复变量，而不用选择结构。

1. 陈述问题

编写一个程序，解一元二次方程的根，不管有两个不同的实根，还是用两个相同的实根或两个不同复根。不需要检测判别式。

2. 定义输入输出

本程序所需要方程式

$ax^2 + bx + c = 0$ (3.1)

的三个系数 a，b，c。输出是这个方程式的所有根。

3. 设计算法

这个程序从整体上可以分为三大步，即输入，计算，输出

Read the input data
Calculate the roots
Write out the roots

我们现在把每一步进行逐步细化。这时判别式的值对程序的执行过程不产生影响。伪代码如下：

```
Prompt the user for the coefficients a, b, and c.
Read a, b, and c
discriminant ← b^2 - 4 * a * c
x1 ←( -b + sqrt(discriminant) ) / (2*a)
x2 ←( -b - sqrt(discriminant) ) / (2*a)
Print 'x1 = ' , real(x1), ' + i ', imag(x1)
Print 'x2 = ' , real(x2), ' + i ', imag(x2)
```

4. 将算法转化为 **MATLAB** 语句

```
% Script file: calc_roots2.m
%
% Purpose:
% This program solves for the roots of a quadratic equation
% of the form a*x**2 + b*x + c = 0. It calculates the answers
% regardless of the type of roots that the equation possesses.
%
% Record of revisions:
% Date Programmer Description of change
% ==== ========== ====================
% 12/06/98 S. J. Chapman Original code
%
% Define variables:
% a -- Coefficient of x^2 term of equation
% b -- Coefficient of x term of equation
% c -- Constant term of equation
% discriminant -- Discriminant of the equation
% x1 -- First solution of equation
% x2 -- Second solution of equation
% Prompt the user for the coefficients of the equation
disp ('This program solves for the roots of a quadratic ');
disp ('equation of the form A*X^2 + B*X + C = 0. ');
a = input ('Enter the coefficient A: ');
```

```
b = input ('Enter the coefficient B: ');
c = input ('Enter the coefficient C: ');
% Calculate discriminant
discriminant = b^2 - 4 * a * c;
% Solve for the roots
x1 = ( -b + sqrt(discriminant) ) / ( 2 * a );
x2 = ( -b - sqrt(discriminant) ) / ( 2 * a );
% Display results
disp ('The roots of this equation are:');
fprintf ('x1 = (%f) +i (%f)\n', real(x1), imag(x1));
fprintf ('x2 = (%f) +i (%f)\n', real(x2), imag(x2));
```

5.检测程序下一步，我们必须输入检测来检测程序。我们要有三组数据进行检测，其判别式分别大于 0，等于 0，小于 0。根据方程式（3。1），用下面的方程式验证程序。

$x^2 + 5x + 6 = 0$ x = -2, x = -3
$x^2 + 4x + 4 = 0$ x = -2
$x^2 + 2x + 5 = 0$ x = -1 ± 2i

我们把它们的系数分别输入程序，结果如下

```
>> calc_root2
This program solves for the roots of a quadratic
equation of the form A*X^2 + B*X + C = 0.
Enter the coefficient A: 1
Enter the coefficient B: 5
Enter the coefficient C: 6
The roots of this equation are:
x1 = (-2.000000) +i (0.000000)
x2 = (-3.000000) +i (0.000000)
>> calc_root2
This program solves for the roots of a quadratic
equation of the form A*X^2 + B*X + C = 0.
Enter the coefficient A: 1
Enter the coefficient B: 4
Enter the coefficient C: 4
The roots of this equation are:
x1 = (-2.000000) +i (0.000000)
x2 = (-2.000000) +i (0.000000)
>> calc_root2
This program solves for the roots of a quadratic
equation of the form A*X^2 + B*X + C = 0.
Enter the coefficient A: 1
Enter the coefficient B: 2
Enter the coefficient C: 5
The roots of this equation are:
x1 = (-1.000000) +i (2.000000)
x2 = (-1.000000) +i (-2.000000)
```

在三种不同的情况下，程序均给出了正确的结果。注意此程序与例 3.2 中的程序相比有多简单。复数数据的应用可大大简化我们的程序。

6.1.4 复数数据的作图

因为复数数据既包括实部又包括虚部，所以在 **MATLAB** 中复数数据的作图与普通实数据的作图有所区别。例如，考虑下面的函数

$$y(t) = e^{-0.2t} (cost + i \sin t) \qquad (6.10)$$

如果我们用传统的 plot 命令给这个函数作图，只有实数数据被作出来，而虚部将会被忽略。下面的语句得到图象如图 6.3 所示，注意出现了警告信息：数据的虚部被忽略

```
t = 0:pi/20:4*pi;
y = exp(-0.2*t) .* (cos(t) + i * sin(t));
plot(t, y);
title('\bfPlot of Complex Function vs Time');
xlabel('\bf\itt');
ylabel('\bf\ity(t)');
```

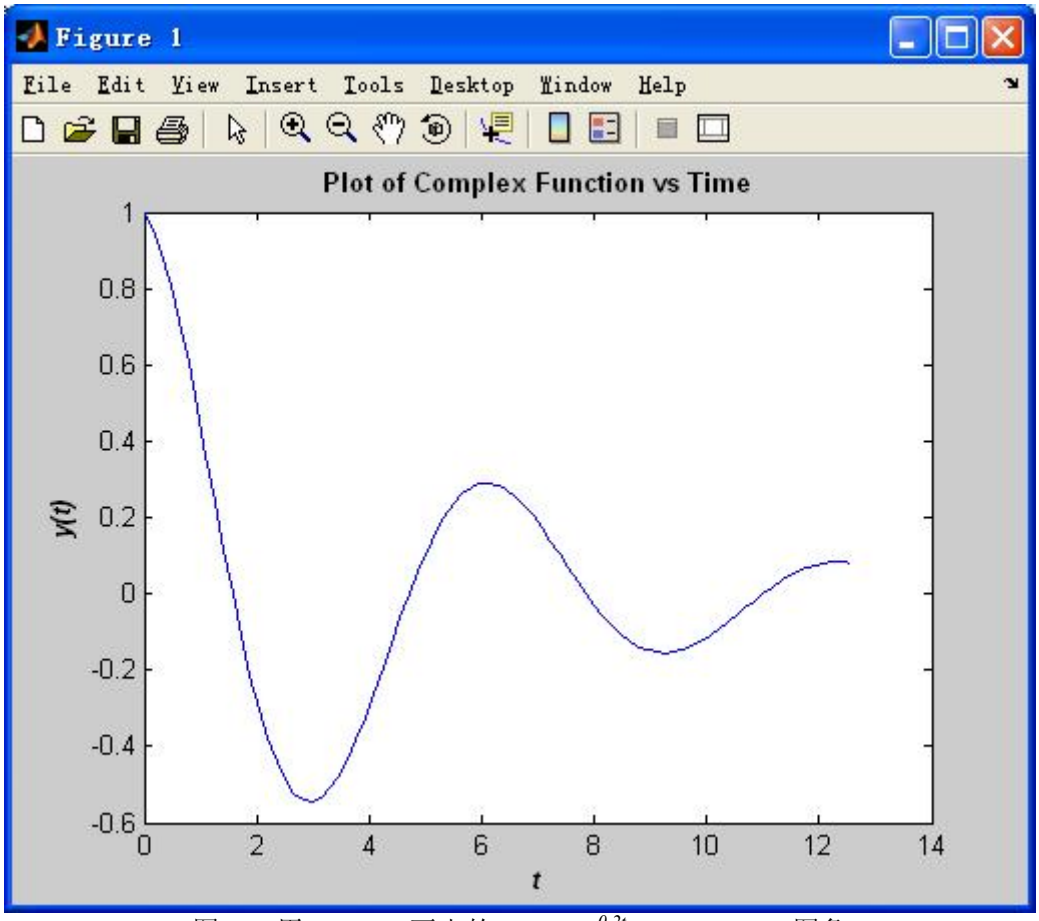


图 6.3 用 plot(t, y)画出的 $y(t) = e^{-0.2t}(\cos t + i \sin t)$图象

如果函数的实部和虚部都需要的话，那么用户可以有几种选择。我们可以用下面的语句，在相同的时间轴内画出函数的图象（图 6.4）。

```
t = 0:pi/20:4*pi;
y = exp(-0.2*t) .* (cos(t) + i * sin(t));
plot(t, real(y),'b-');
hold on;
plot(t, imag(y),'r-');
title('\bfPlot of Complex Function vs Time');
xlabel('\bf\itt');
ylabel('\bf\ity(t)');
legend('real','imaginary');
hold off;
```

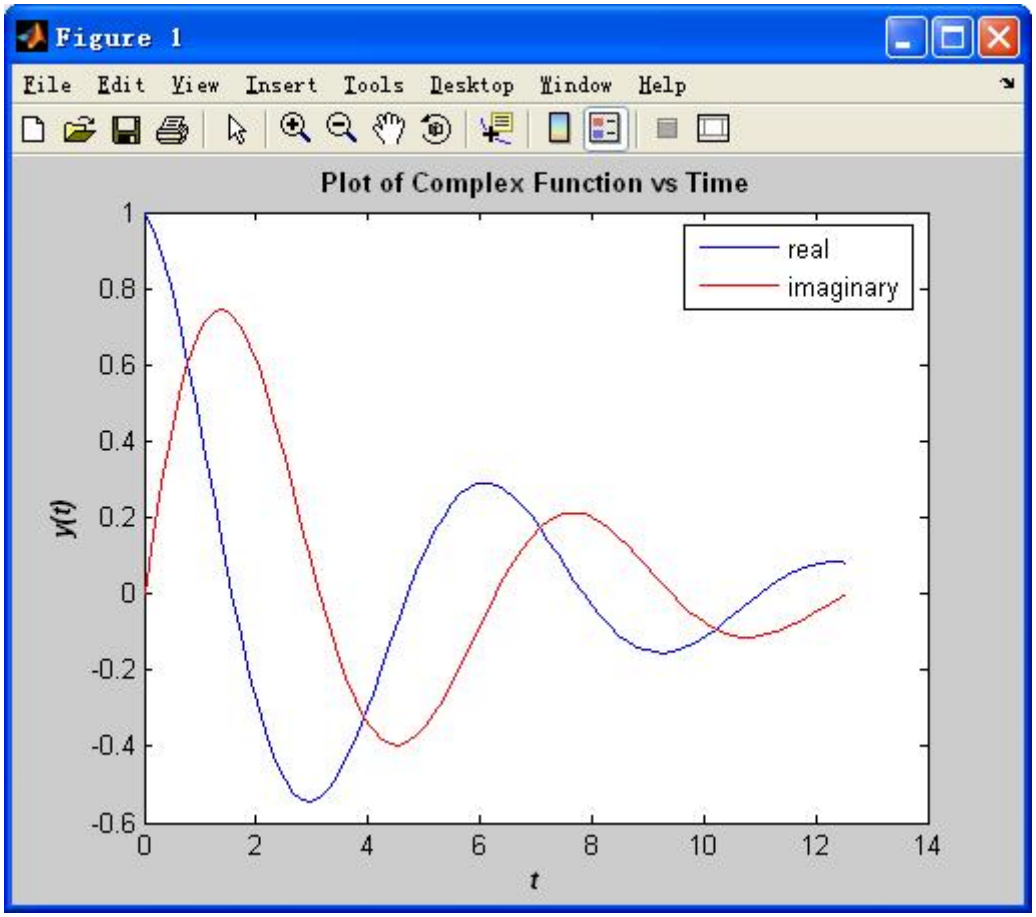


图 6.4 包含了 y(t)的实部和虚部

可选择的，函数的实部-虚部图可以被画出来。如果有一个复参数提供给 plot 函数它会自动产生一个函数的实部-虚部图。产生这类图的语句如下，产生的结果如图 6.5 所示。

```
t = 0:pi/20:4*pi;
y = exp(-0.2*t) .* (cos(t) + i * sin(t));
plot(y,'b-');
title('\bfPlot of Complex Function');
xlabel('\bfReal Part');
ylabel('\bfImaginary Part');
```

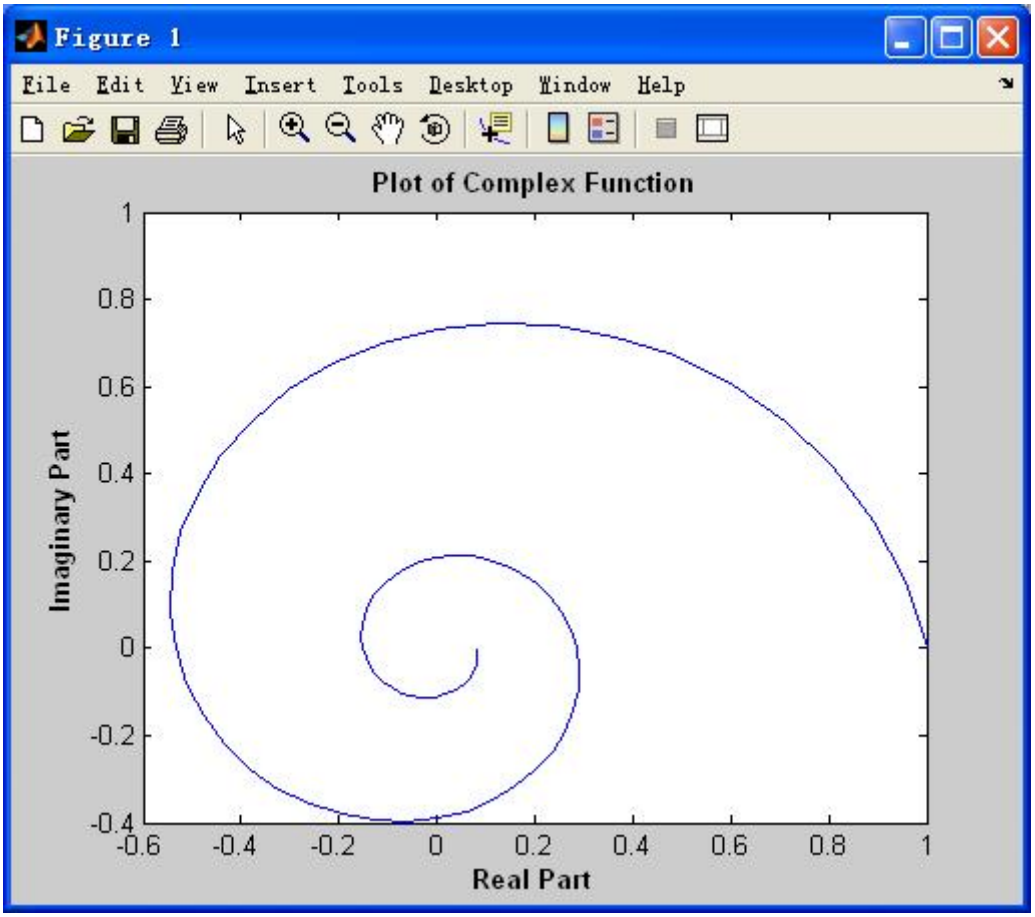


图 6.5 y(t)的的实部-虚部图

最后，我们可以画出函数的极坐标图。产生这类图语句如下，产生的结果如图图 6.6 所示。

```
t = 0:pi/20:4*pi;
y = exp(-0.2*t) .* (cos(t) + i * sin(t));
polar(angle(y),abs(y));
title('\bfPlot of Complex Function');
```

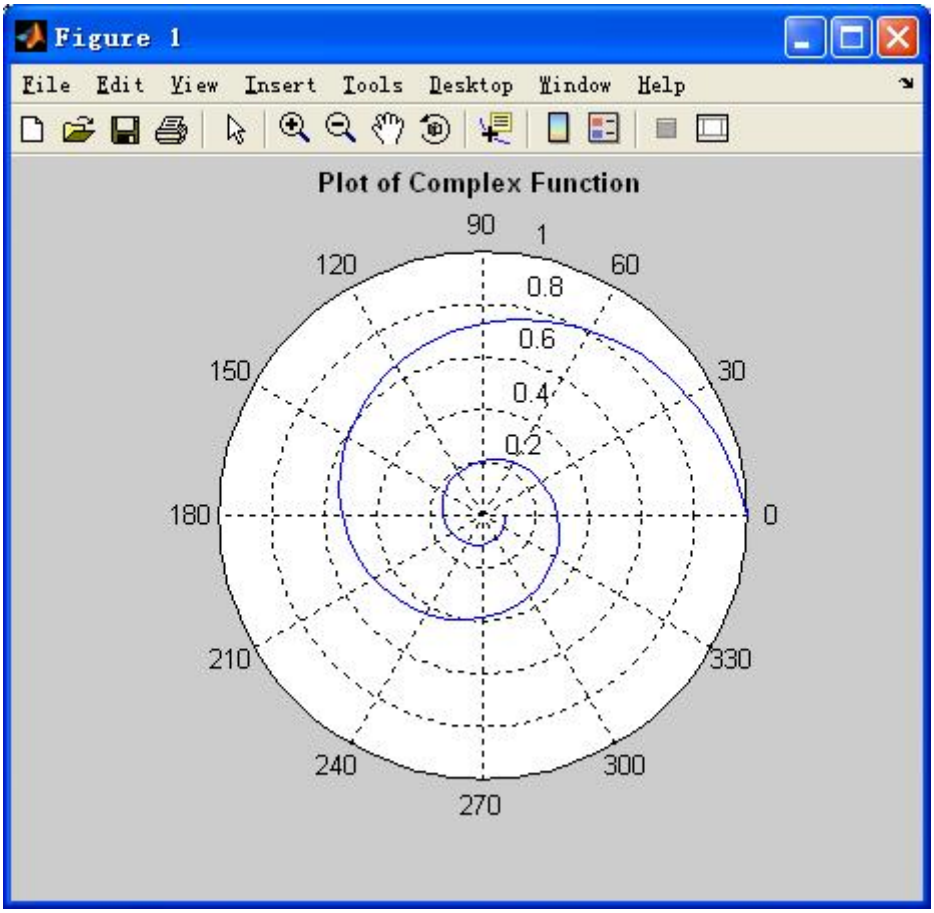


图 6.6 y(t)的极坐标图

6.2 字符串函数（string functions）

一个 **MATLAB** 字符串是一个 char 型数组。每一个字型占两个字节。当字符串被赋值于一个变量时，这个变量将被自动创建为字符变量。例如语句

```
str = 'This is a test';
```

将会创建一个含有 14 个元素的数组。用 whos 命令查看它属性。

```
>> whos
  Name        Size                    Bytes  Class
  str         1x14                       28  char array
Grand total is 14 elements using 28 bytes
```

一个专门的函数 ischar 常用来判断一个变量是否为字符数组。如果是的话，那么函数较会返回 1，如果不是，将会返回 0。

在下面的的小节中，我们将向大家介绍一些对字符串进行操作的函数。

6.2.1 字符转换函数

我们可以利用 double 函数把变量从字型转化为 double 型。所以，函数 double(str)产生的结果为

```
>> x = double(str)
x =
```

```
  Columns 1 through 12
    84   104   105   115    32   105   115    32    97    32   116   101
  Columns 13 through 14
   115   116
```

我们可以利用 char 函数把 double 型数据转化为字符型数据。所以函数 char(x)产生的结果为

```
>> x = char(x)
x =
This is a test
```

6.2.2 创建二维字符数组

我们可以创建二维字符数组，但一个数组中每一行的长度都必须相等。如果其中的一行比其他行短，那么这个字符数据将会无效，并产生一个错误。例如，下面的语句是非法的，因为他两行的长度不同。

```
name = ['Stephen J. Chapman'; 'Senior Engineer'];
```

创建二维字符数组的最简单的方法是用 char 函数。函数将会自动地寻找所有字符串中最长的那一个。

```
>> name = char('Stephen J. Chapman','Senior Engineer')
name =
Stephen J. Chapman
Senior Engineer
```

二维字符数组也可以用函数 strvcat，这个函数我会在下一节中介绍。

好的编程习惯
用 char 函数创建二维字符数组，我们就不用担心每一行的长度不相同了。

我们可以应用 deblank 函数去除多余空格。例如，下面的语句去除 name 数组中第二行的多余空格，产生的结果与原来的进行比较。

```
>> line2 = name(2,:)
line2 =
Senior Engineer
>> line2_trim = deblank(name(2,:))
line2_trim =
Senior Engineer
>> size(line2)
ans =
        1       18
>> size(line2_trim)
ans =
        1       15
```

6.2.3 字符串的连接

函数 strcat 水平连接两字符串，忽略所有字符串末端的空格，而字符串的空格保留。例如，下面的语句为

```
>> result = strcat('string 1    ','String 2')
result =
string 1String 2
```

产生的结果 string 1String 2。

函数 strvcat 用于竖直地连接两字符串，自动地把它转化为二维数组。这个函数将产生这样的结果

```
>> result = strvcat('Long String 1    ','String 2')
result =
Long String 1
String 2
```

6.2.4 字符串的比较

字符串与子字符串可以通过下面许多的方式进行比较。

- 两个字符串，或两个字符串的部分，看两者是否相同
- 两个独立的字符相比较看两者是否相同
- 检查字符串判断每一个字符是字母，还是空格

6.2.4.1 比较两字符串，看是否相同

你可以利用 **MATLAB** 函数比较两字符串整体是否相同。它们是

- strcmp 判断两字符串是否等价
- strcmpi 忽略大小写判断两字符串是否等价
- strncmp 判断两字符串前 n 个字符是否等价
- strncmpi 忽略大小写判断两字符串前 n 个字符是否等价

函数 strcmp 比较字符串，包括字符串前面或后面的空格。如果两字符串完全相同，那么这个函数将返回 1。否则，返回 0。strcmpi 与 strcmp 类似，但它忽略了大小写（即“a”与“A”看作相同的）

函数 strncmp 用来比较两字符串前 n 个字符串，包含开头的空格，如果这个 n 个字符是相同的，它们将会返回 1。否则它将会返回 0。函数 strncmpi 与它相类似，但忽略了大小写。

为了更好的理解这些函数，考虑下面的字符串

```
str1 = 'hello';
str2 = 'Hello';
str3 = 'help';
```

字符串 str1 和 str2 不相同，但它第一个字母大小不同。所以 strcmp 将返回 0，strcmpi 将返回 1。

```
>> c = strcmp(str1,str2)
c =
     0
>> c = strcmpi(str1,str2)
c =
     1
```

字符串 str1 和 str3 不相同，所以 strcmp 与 strcmpi 返回 0。但是 str1 和 str3 的前三个字符是相同，所以按照下面的方式调用将会返回 1。

```
>> c = strncmp(str1, str3, 2)
c =
     1
```

6.2.4.2 判断单个字符是否相等

我们可以利用 **MATLAB** 关系运算符对字符数组中的每一个元素进行检测，看是否相同，但是我们要保证它们的维数是相同的，或其中一个是标量。例如，你可以用相等运算符（==）来检测两字符串是否相匹配。

```
>> a = 'fate';
>> b = 'cake';
>> result = a == b
result =
     0     1     0     1
```

所有的关系运算符（>，>=，<，<=，==，~=）都是对字符所对应的 ASCII 值进行比较。

与 C 语言不同，**MATLAB** 中没有一个内建函数，对两字符串在整体进行“大于”或“小于”的关系运算。我们将会本节末创建一个类似的函数。

6.2.4.3 在一字符串内对字符进行判断

有两个函数可对一个字符串内的字符逐个进行分类。

- isletter 用来判断一个字符是否为字母
- isspace 判断一个字符是否为空白字符（空格，tab，换行符）

例如，我们要创建一个字符串 mystring，

```
mystring = 'Room 23a'
```

函数 isletter 检测字符串中的每一个字符，将产生一个与字符串 isletter 相同长度输出向量，一个字符对应一个 1。

```
>> a = isletter(mystring)
a =
     1     1     1     1     0     0     0     1
```

在 a 中前四个元素和最后一个元素是 1，因为它们对应的 mystring 中的字符是字母。函数 isspace 检测字符串中的每一个字符，将产生一个和字符串长度相同的输出变量，对应于空字符的向量元素为 0。

因为向量的第五个元素对应的是空格，所以向量的第五个元素的值为 1。

6.2.5 在一个字符串中查找/替换字符

MATLAB 提供了许多的函数，用来对字符串中的字符进行查找或替换。考虑字符串 test

```
test = 'This is a test!';
```

函数 findstr 返回短字符串在长字符串中所有的开始位置。例如为了寻找 test 内的所有“is”

```
>> position = findstr(test,'is')
position =
     3     6
```

字符串“is”在 test 内出现两次，开始位置分别为 3 和 6。

函数 strmatch 是另一种匹配函数。它用来查看二维数组行开头的字符，并返回那些以

指定的字符序列为开头行号。它的基本形式如下

```
result = strmatch(str,array);
```

例如，我们用 strvcat 创建一个二维数组，

```
array = strvcat('maxarray','min value','max value');
```

那么下面的语句将会返回开始字符为“max”的行数。

```
>> result = strmatch('max',array)
result =
     1
     3
```

函数 strrep 用于进行标准的查找和替换操作。它能找到一个字符串中的所有另一个字符串，并被第三个字符串替换。这个函数形式为

```
result = strrep(str,srch,repl)
```

其中 str 是被检测的字符串，srch 是要查找到的字符串，repl 是用于替代的字符串，例如，

```
>> result = strrep(test,'test','pest')
result =
This is a pest!
```

函数 strtok 返回输入字符串中第一次出现在分隔符前面的所有字符。默认的分隔符为一系列的空白字符。strtok 的形式如下

```
[token, remainder] = strtok(string,delim)
```

其中 string 是输入字符串，delim 是可选择的分隔符，token 代表输入字符串中第一次出现在分隔符前面的所有字符，remainder 代表这一行的其余部分。例如

```
>> [token, remainder] = strtok('This is a test!')
token =
This
remainder =
  is a test!
```

你可以利用函数 strtok 把一个句子转换为单词。例如，下面的代码从字符数组 input_string 中分离出每一个单词，并把每一个单词独立地存储在字符数组 all_words 的每一行中。

```
function all_words = word(input_string)
remainder = input_string
all_words = '';
while (any(remainder))
      [chopped, remainder] = strtok(remainder);
      all_words = strvcat(all_words, chopped);
end
```

6.2.6 大小写转换

函数 upper 和 lower 分别把一个字符串中所有转化大定和小写。例如

```
>> result = upper('This is test 1!')
result =
THIS IS TEST 1!
>> result = lower('This is test 2!')
result =
this is test 2!
```

注意在大小转换时，数字和符号不受影响。

6.2.7 字符串转换为数字

MATLAB 把由数字组成的字符串转化为数字要用到函数 eval。例如，字符串“3.141592”能用下面的语句把它转换为数字。

```
>> a = '3.141592';
>> b = eval(a)
b =
    3.1416
>> whos
  Name        Size                    Bytes  Class
  a           1x8                        16  char array
Grand total is 8 elements using 16 bytes
```

字符串可以用 sscanf 函数转化为数字。这个函数根据格式化转义字符转化为相应的数字。这个函数最简单的形式如下

```
value = sscanf(string, format)
```

其中，string 是要转化的字符串，format 是相应的转义字符。函数 sscanf 两种最普通的转义序是“%d”，“%g”，它们分别代表输出为整数或浮点数。这个函数更多的细节我们将在第 8 章介绍。在作图中，创建一个复杂的标题或标签，它是非常有用的。

下面的例子用于说明函数 sscanf 的应用。

```
>> value1 = sscanf('3.141593','%g')
value1 =
    3.1416
>> value2 = sscanf('3.141593','%d')
value2 =
     3
```

6.2.8 数字转化为字符串

MATLAB 中有许多的字符串/数字转换函数把数字转化为相应的字符串。我们在这里只看两个函数 num2str 和 int2str。考虑标量 x

```
x = 5317;
```

在默认的情况下，**MATLAB** 把数 x 作为一个 1×1 的 double 数组，它的值为 5317。函数 int2str（integer to string）所这个标量转化为 1×4 的字符数组，包含有字符串“5317”。

```
>> x = 5317;
>> y = int2str(x);
>> whos
  Name        Size                    Bytes  Class
  x           1x1                         8  double array
  y           1x4                         8  char array
Grand total is 5 elements using 16 bytes
```

函数 num2str 为输出字符串的格式提供更多的控制。第二个可选择的参数可以对输出字符串的数字个数进行设置或指定一个实际格式。例如

```
>> p = num2str(pi,7)
p =
3.141593
>> p = num2str(pi,'%10.5e')
p =
3.14159e+000
```

函数 int2str 和 num2str 对作图标签是非常有用的。例如，下面的语句用 num2str 生成图象的标签。

```
function plotlabel(x,y)
plot(x,y)
str1 = num2str(min(x));
str2 = num2str(max(x));
out = ['Value of f from ' str1 ' to ' str2];
xlabel(out);
```

还有一些转换函数，用于把数字值从十进制转化另一种数制，例如二进制或十六进制。例如函数 dec2hex 把一个十进制数转化为相应的十六进制字符串。此类的函数还有 hex2num，hex2dec，bin2dec，dec2bin，base2dec，你可以通过 **MATLAB** 网上帮助来获取这些函数的作用和使用方法。

MATLAB 函数 mat2str 可以把一个数组转化为相应的 **MATLAB** 能运算字符串。这个字符串可以是 eval 函数的输入，函数 eval 对这个字符串的运算和直接在命令窗口键入效果是一样的。例如，我们定义一个数组如下

```
>> a = [1 2 3; 4 5 6]
a =
     1     2     3
     4     5     6
```

函数 mat2str 运行得到的结果为

```
>> b = mat2str(a)
b =
[1 2 3;4 5 6]
```

最后，**MATLAB** 中有一个专门的函数 sprintf 等价于函数 fprintf，唯一不同的是它的输出是一个字符串。这个函数对字符串的格式化操作的完全支持。例如，

```
>> str = sprintf('The value of pi = %8.6f',pi)
str =
The value of pi = 3.141593
```

在图象中，用这些函数创建复杂的标题或标签将会非常的有用。

6.2.9 总结

普通的 **MATLAB** 字符串函数总结在表 6.2 中。

表 6.2 普通的 **MATLAB** 字符串函数

类别	函数	描述
普通		
	char	(1)把数字转化为相应的字符值 (2)把二维数组转化相应的字符串
	double	把字符转化为相应的 double 值
	blanks	创建一个由空格组成的字符串
	deblanks	去除字符串末端的空格
字符检测		
	ischar	如果是一个字符数组，那么将会返回 1
	isletter	如果是字母表中的字母，那么将会返回 1
	isspace	如果是空白字符，那么将会返回 1
字符串操作		
	strcat	连接字符串
	strvcat	竖直地连接字符串
	strcmp	如果两字符串相等，那么函数将会返回 1
	stricmp	忽略大小写如果两字符串相等，那么函数将会返回 1

strncmp	如果两字符串的前 n 个字母相等，那么函数将会返回 1
strncmpi	忽略大小，如果两字符串的前 n 个字母相同，那么数将会返回 1
findstr	在一个字符串中寻找另一个字符串
strfind	在一个字符串中寻找另一个字符串（版本 6.1 或以后的版本）
strjust	对齐字符串
strmatch	找字符串的区配
strrep	用一个字符串去替代另一个字符串
strtok	查找一字符串
upper	把字符串的所有字符转化为大写
lower	把字符串的所有字符转化为小写
数字转化为字符串	
int2str	把整数转化为相应的字符串形式
num2str	把数字转化为相应的字符串形式
mat2str	把矩阵转化为相应的字符串形式
sprintf	对一字符串进行格式化输出
字符串转化为数字	
str2double	把字符串转化相应的 double 型数据
str2num	把字符转化成数字
sscanf	从字符串中读取格式化数据
数制转换	
hex2num	把 IEEE 十六进制字符型型数据转化为 double 形数据
hex2dec	把十六制字符串转化为相应的十进制整数
dec2hex	把十进制数转化为相应的十六制字符串
bin2dec	把二进制字符串转化为相应的十进制整数
base2dec	把 base B 转化为相应的十进制数据
dec2base	把十进制转化为相应的 base B
hex2num	把 IEEE 十六进制字符型型数据转化为 double 形数据

例 6.2

字符串比较函数

在 C 语言中，函数 strcmp 根据 Ascii 码中字符顺序（我们称之为字典顺序）比较两字符，如果第一个字符串的字典顺序在第二个字符串字典之后，函数将会产生-1，如果两字符串相同那么将会产生 0，如果第一个字符串的字典顺序在第二个字符串字典之前那么函数将会返回+1。

创建一个 **MATLAB** 函数 c_strcmp 用来比较两字符串，其功能与 C 语言 strcmp 中的函数功能相类似。这个函数对字符串进行比较时，应忽略末尾的空格。注意这个函数必须控制两字符串不同长的情况。

答案：

1. 陈述问题

编写一个函数，比较两字符串 str1 和 str2，并返回以下结果

- -1　　如果 str1 在字典顺序比 str2 的晚
- 0　　如果两字符串的字典顺序相同
- +1　　如果 str1 的字典顺序比 str2 的早

2. 定义输入输出量

函数所需的输入量为两个字符串，str1 和 str2。这个函数的输出为-1、0 或 1。

3. 描述算法这个工程可分为以下四大步

Verify input strings

Pad strings to be equal length
Compare characters from beginning to end, looking for the first difference
Return a value based on the first difference

我们将以上每一大步分解成更小的更细的小块。第一，我们必须验证传递给函数的数据是正确的。函数必须有两个参数，且这两个参数必须为字符。这一步的伪代码为

```
% Check for a legal number of input arguments.
msg = nargchk(2, 2, nargin)
error(msg)
% Check to see if the arguments are strings
if either argument is not a string
    error('str1 and str2 must both be strings')
else
    (add code here)
end
```

下一步，我们要做的是把这两个字符串具有相同的长度。最简单的方法是用 strvcat 函数把这两个字符串联合成一个二维数组。注意在这个步骤中产生字符串末端空格只是为了两字符串相等，所以这些空格可以被省略。

```
% Pad strings
strings = strvcat(str1, str2)
```

现在我们要对字符串中的每一个字符进行比较，直到我们一个不同的字符出现，并基于这种不同返回相应的值。为了达到这个止目的，其中的方法是应用关系运算符比较两个字符串，产生一个由 0 和 1 组成的数组。然后我们可以寻找第一个 1，因为它两字符串在这里出现第一次不同。这一步的伪代码如下

```
% Compare strings
diff = strings(1, :) ~= strings(2, :)
if sum(diff) == 0
    % Strings match
    result = 0
else
    % Find first difference
    ival = find(diff)
    if strings(1, ival(1)) > strings(2, ival(1))
            result = 1
    else
            result = -1
    end
end
```

4. 把算法转化为相应的 **MATLAB** 语句

```
function result = c_strcmp(str1,str2)
%C_STRCMP Compare strings like C function "strcmp"
% Function C_STRCMP compares two strings, and returns
% a -1 of str1 < str2, a 0 if str1 == str2, and a
% +1 if str1 > str2.
% Define variables:
% diff                  -- Logical array of string differences
% msg                   -- Error message
% result                -- Result of function
% str1                  -- First string to compare
% str2                  -- Second string to compare
% strings               -- Padded array of strings
% Record of revisions:
% Date          Programmer          Description of change
% ====          ==========          =====================
% 10/18/98    S. J. Chapman      Original code
% Check for a legal number of input arguments.
```

```
msg = nargchk(2,2,nargin);
error(msg);
% Check to see if the arguments are strings
if ~(isstr(str1) & isstr(str2))
    error('Both str1 and str2 must both be strings!')
else
    % Pad strings
    strings = strvcat(str1,str2);
    % Compare strings
    diff = strings(1,:) ~= strings(2,:);
    if sum(diff) == 0
        % Strings match, so return a zero!
        result = 0;
    else
        % Find first difference between strings
        ival = find(diff);
        if strings(1,ival(1)) > strings(2,ival(1))
            result = 1;
        else
            result = -1;
        end
    end
end
```

5. 检测程序

我们必须用多个字符串对程序进行检测

```
>> result = c_strcmp('String 1','String 1')
result =
     0
>> result = c_strcmp('String 1','String 1          ')
result =
     0
>> result = c_strcmp('String 1','String 2')
result =
    -1
>> result = c_strcmp('String 1','String 0')
result =
     1
>> result = c_strcmp('String 1','str')
result =
    -1
```

第一次检测返回 0，是因为两字符串是相同的。第二次也返回 0，因为两字符串也是相等的，只是末端空格不同，末端空格被忽略。第三次检测返回-1，因为两字符串第一次的不同出现在第 8 位上，且在这个位置上“1”<“2”。第四次检测将返回 1，因为两字符串第一次的不同出现在第 8 位上，且在这个位置上，“1”>“0”。第五次检测将会返回-1，因为两字符串的第一个字符就不同，在 ascii 的序列上“S”<“s”。这个函数工作正常。

测试 6.1

本测试提供了一个快速的检查方式，看你是否掌握了 6.1 到 6.2 的基本内容。如果你对本测试有疑问，你可以重读 6.1 到 6.2，问你的老师，或和同学们一起讨论。在附录 B 中可以找到本测试的答案。

1. 下面的语句产生的 result 的值是多少？

(a)
```
x = 12 + i*5;
y = 5 - i*13;
result = x > y;
```

(b)
```
x = 12 + i*5;
```

```
        y = 5 - i*13;
        result = abs(x) > abs(y);
    (c) x = 12 + i*5;
        y = 5 - i*13;
        result = real(x) - imag(y);
```

2. 果 array 是一个复数组，那么函数 plot(array)将会产生怎样的结果。

3. 们如何将一个 char 数据类型的向量转化为相应的 double 型数据类型的数据向量。

从式 4 到 11，判断这些语句是否正确。如果它们正确，那么将产生什么结果？

4.
```
str1 = 'This is a test! ';
str2 = 'This line, too.';
res = strcat(str1, str2);
```
5.
```
str1 = 'Line 1';
str2 = 'Line 2';
res = strcati(str1,str2);
```
6.
```
str1 = 'This is a test!';
str2 = 'This line, too.';
res = [str1; str2];
```
7.
```
str1 = 'This is a test!';
str2 = 'This line, too.';
res = strvcat(str1, str2);
```
8.
```
str1 = 'This is a test!';
str2 = 'This line, too.';
res = strncmp(str1, str2, 5);
```
9.
```
str1 = 'This is a test!';
res = findstr(str1, 's');
```
10.
```
str1 = 'This is a test!';
str1(4:7) = upper(str1(4:7));
```
11.
```
str1 = 'This way to the egress.';
str2 = 'This way to the egret.';
res = strncmp(str1, str2);
```

6.3 多维数组

从，**MATLAB**5.0 版本开始支持多于二维的数组。这些多维数组用来显示多于二维的数据，或显示多个版本的二维图。例如，在一个三维空间中压力和速度的测量对于一些学科来说是非常重要的，例如空气动力学，流体力学。在这些领域中自然会用到多维数组。

多维数组是二维数组的扩展。每增加一维，它们所对应的每个元素就会多一个下角标。

我们可以轻易地创建一个多维数组。它既可以通过直接的赋值语句进行赋值，可用相同的函数进行创建（和一维二维中一样）。例如，假设你已经利用赋值语句创建了一个二维数组

```
>> a = [1 2 3 4; 5 6 7 8]
a =
     1     2     3     4
     5     6     7     8
```

这是一个 2×4 数组，每个元素被访问时，都应带有两个下标。这个数组可扩展为一个三维 2×4×3 数组，语句如下

```
>> a(:,:,2) = [9 10 11 12; 13 14 15 16];
>> a(:,:,3) = [17 18 19 20; 21 22 23 24]
a(:,:,1) =
     1     2     3     4
     5     6     7     8
a(:,:,2) =
```

```
     9    10    11    12
    13    14    15    16
a(:,:,3) =
    17    18    19    20
    21    22    23    24
```

在这个多维数组中的每一个元素都可以用它的函数名加上它的三个下标进行访问，数据下标的创建可以用克隆运算符。例如，a(2,2,2)的值为

```
>> a(2,2,2)
ans =
    14
```

向量 a(1,1,:)为

```
>> a(1,1,:)
ans(:,:,1) =
     1
ans(:,:,2) =
     9
ans(:,:,3) =
    17
```

多维数组也可以用与其他数据相同的函数进行创建

```
>> b = ones(4,4,2)
b(:,:,1) =
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
b(:,:,2) =
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
>> c = randn(2,2,3)
c(:,:,1) =
   -0.4326    0.1253
   -1.6656    0.2877
c(:,:,2) =
   -1.1465    1.1892
    1.1909   -0.0376
c(:,:,3) =
    0.3273   -0.1867
    0.1746    0.7258
```

多维数组的维数可以利用 ndims 函数得到，数组的大小可通过 size 函数得到。

```
>> ndims(c)
ans =
     3
>> size(c)
ans =
     2     2     3
```

如果你需要多数组编写应用程序，你可以通过阅读 **MATLAB** user's guide 来了解更多的多维数组函数的细节。

好的编程习惯

我们可以利用多维数组来解决自然界的多变量问题，如空气动力学和流体力学。

6.4 关于二维作图的补充说明

在前面的章节中，我们学习了如何创建线性图，对数图，线性-对数图和极坐标图。

MATLAB 提供了许多的画图类型，用来显示你的数据。本节将向你介绍它们其中的一些操作。

6.4.1 二维作图的附加类型

除了我们已经看到图象类型，**MATLAB** 还支持其他的图象。实际上，在 **MATLAB** 帮助工作台中列出超过 20 种类型的作图。例如针头图（Stem Plots），阶梯图（stair plots），条形图，饼图（pie plots），罗盘图（compass plots）。在针头图中的每一个值都用一个圆圈和垂直于 x 轴的直线连接而成。在阶梯图中的每一个值都是用连续的竖直的长条线来表示，形成阶梯状效果。条形图可分成水平条形图和竖直条形图。

饼图用不同的扇区代表不同的变量。最后罗盘图是另一种极坐标图它的每一值用箭头来表示。

针头图，阶梯图，条形图，饼图，罗盘图与普通的图象差不多，它的调用方式相同。例如，下面显示的是一个针头图的代码，产生的图象如图 6.7a 所示。

```
>> x = [1 2 3 4 5 6];
>> y = [2 6 8 7 8 5];
>> stem(x,y);
>> title('\bfExample of a Stem Plot');
>> xlabel('\bf\itx');
>> ylabel('\bf\ity');
>> axis([0 7 0 10]);
```

针头图，条形图，罗盘图可以调用 stair，bar，barh 和 compass 命令来创建，代码类似于上面的语句。这些图象的具体细节，例如它们选择性参数，可以通过 **MATLAB** 在线帮助系统得到。

函数 pie 与前面其他的画图有所不同。为了创建一个饼图，程序员把数组 x 传递给函数，函数计算出每一个元素占全部元素和的百分比。例如，如果数组 x 是[1 2 3 4]，那么 pie 函数将会计算出第一个元素 1 占全部元素和的 10%，第二个元素占 20%等等。这个函数将会占这个百分比画出相应的饼图。

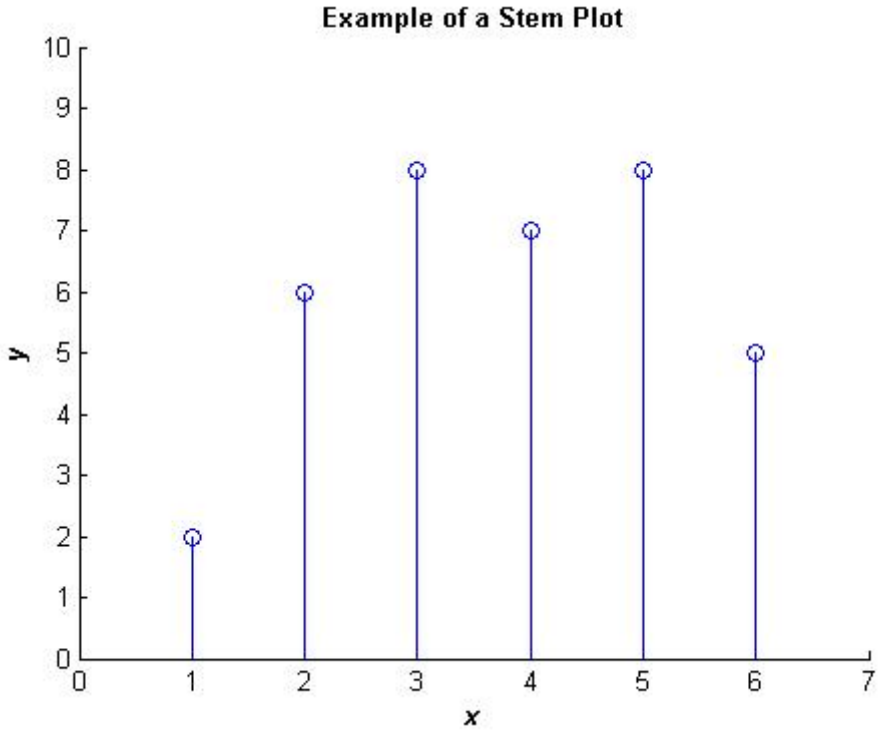
函数 pie 也支持选择性参数，它是 eplode。如果存在的话，explode 是一个逻辑数组，包含元素 1 和 0。如果 explode 的值为 1，那么它对应的扇区就从整体中分离出来。下面的代码将会创建出饼图 6.7e。注意下面的第二个扇区分离出来的。

```
data = [10 37 5 6 6];
explode = [0 1 0 0 0];
pie(data, explode);
title('\bfExample of a Pie Plot');
legend('One','Two','Three','Four','Five');
```

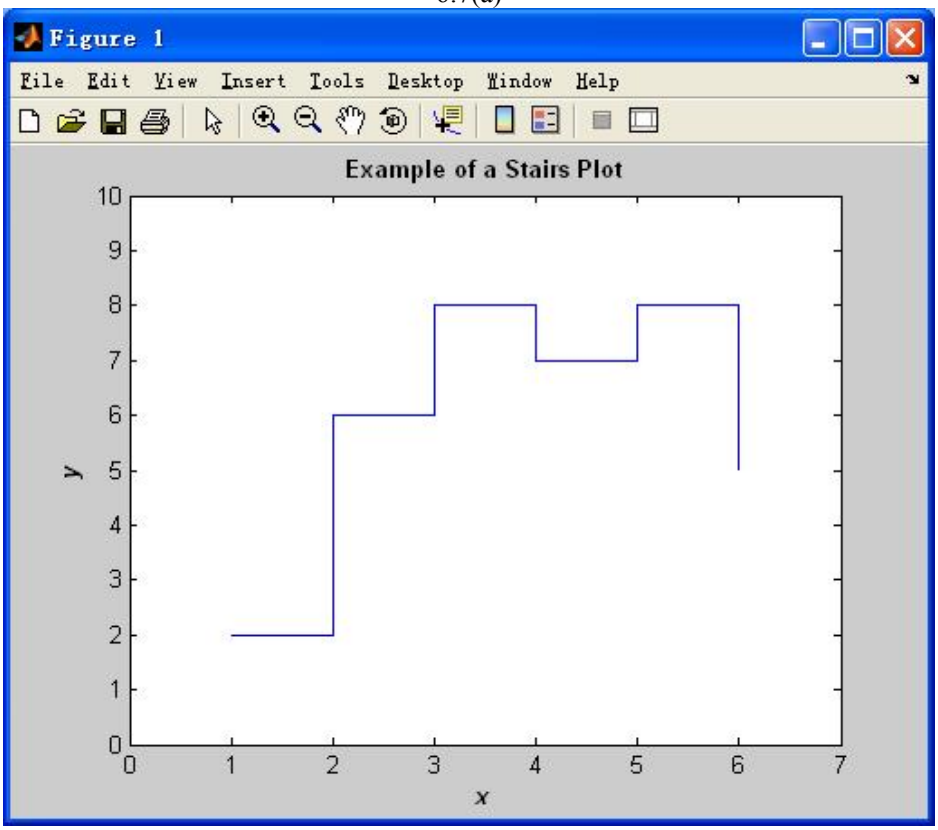
表 6.3 附加的二维作图类型

函数	描述
bar(x, y)	这个函数用于创建一个水平的条形图，x 代表第一个 X 轴的取值，y 代表对应于 Y 的取值
barh(x, y)	这个函数用于创建一个竖直的条形图，x 代表第一个 X 轴的取值，y 代表对应于 Y 的取值
compass(x, y)	这个函数用于创建一个极坐标图，它的每一个值都用箭头表示，从原点指向（x，y），注意：（x，y）是直角坐标系中的坐标。

pie(x) pie(x, explode)	这个函数用来创建一个饼状图，x 代表占总数的百分数。explode 用来判断是否还有剩余的百分数
stairs(x, y)	用来创建一个阶梯图，每一个阶梯的中心为点(x, y)
stem(x, y)	这个函数可以创建一个针头图，它的取值为(x,y)



6.7(a)



6.7(b)

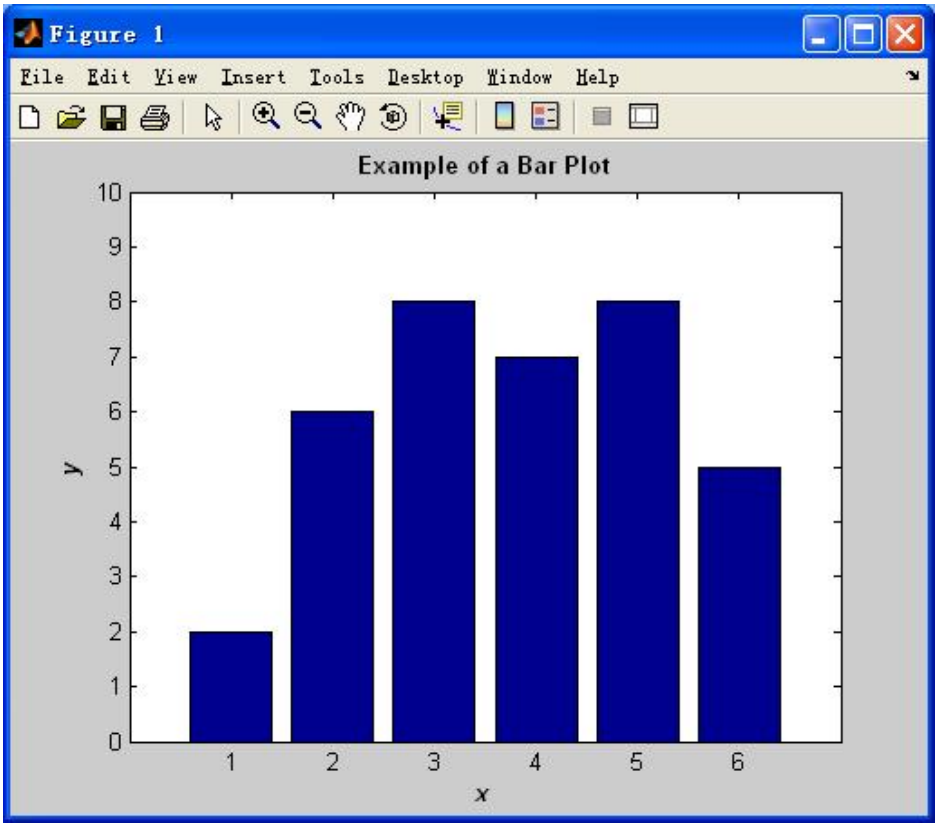


Figure 1
File Edit View Insert Tools Desktop Window Help
Example of a Bar Plot
y
x
0 1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6

6.7(c)

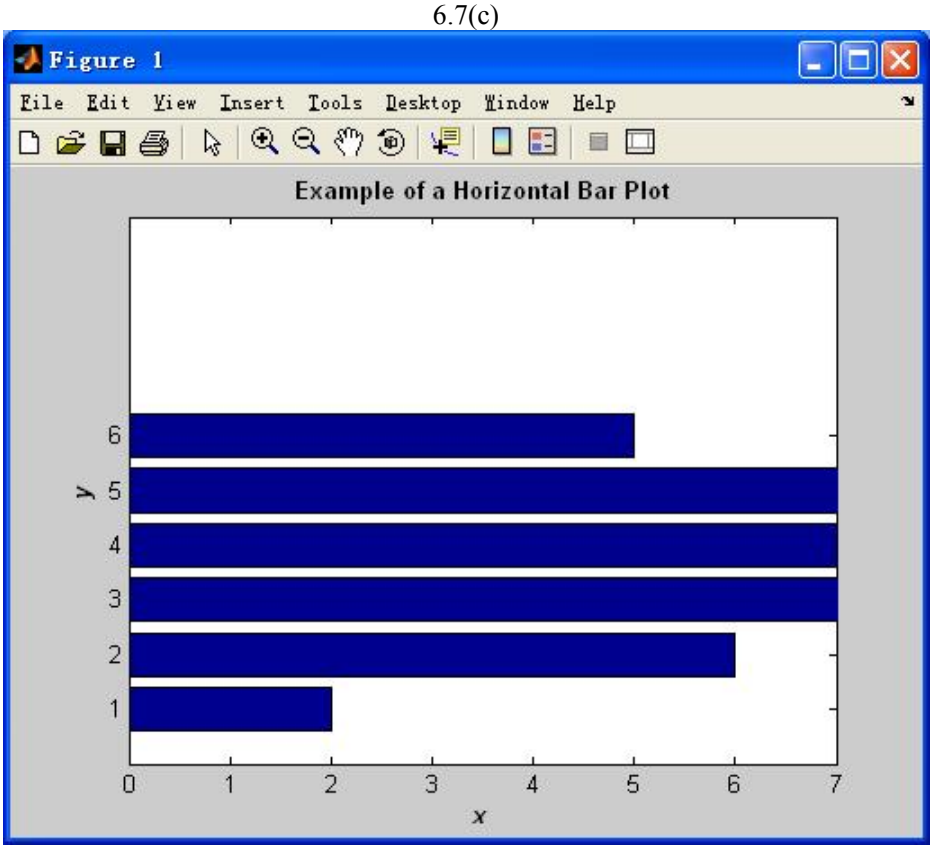
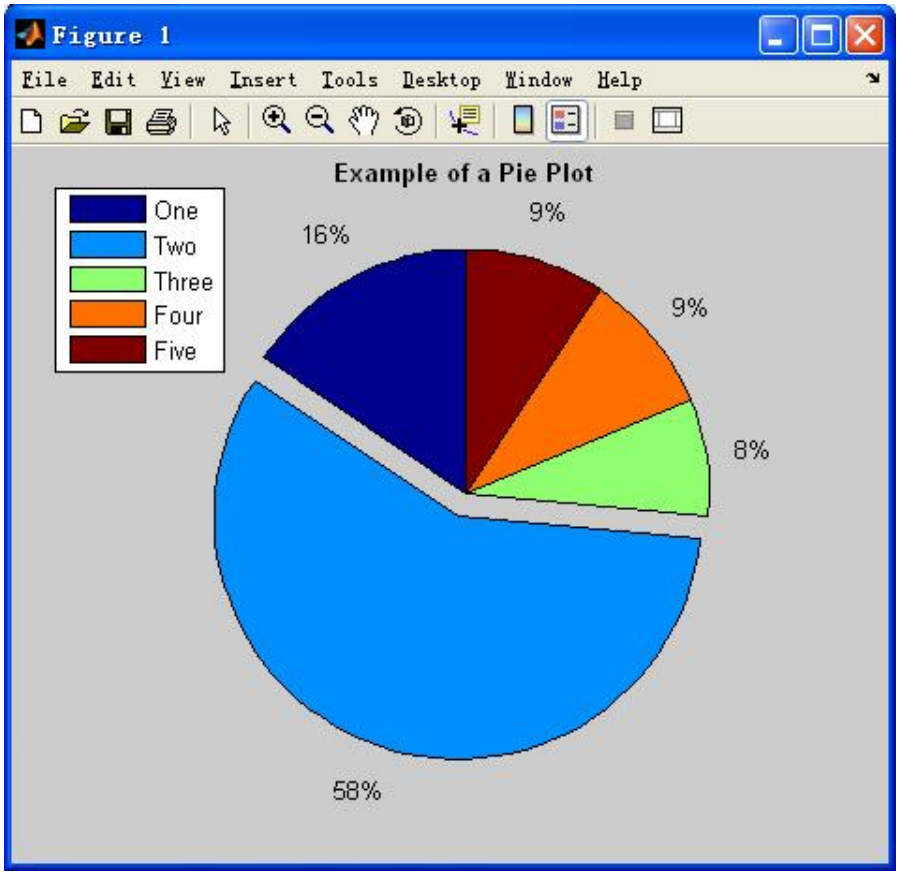
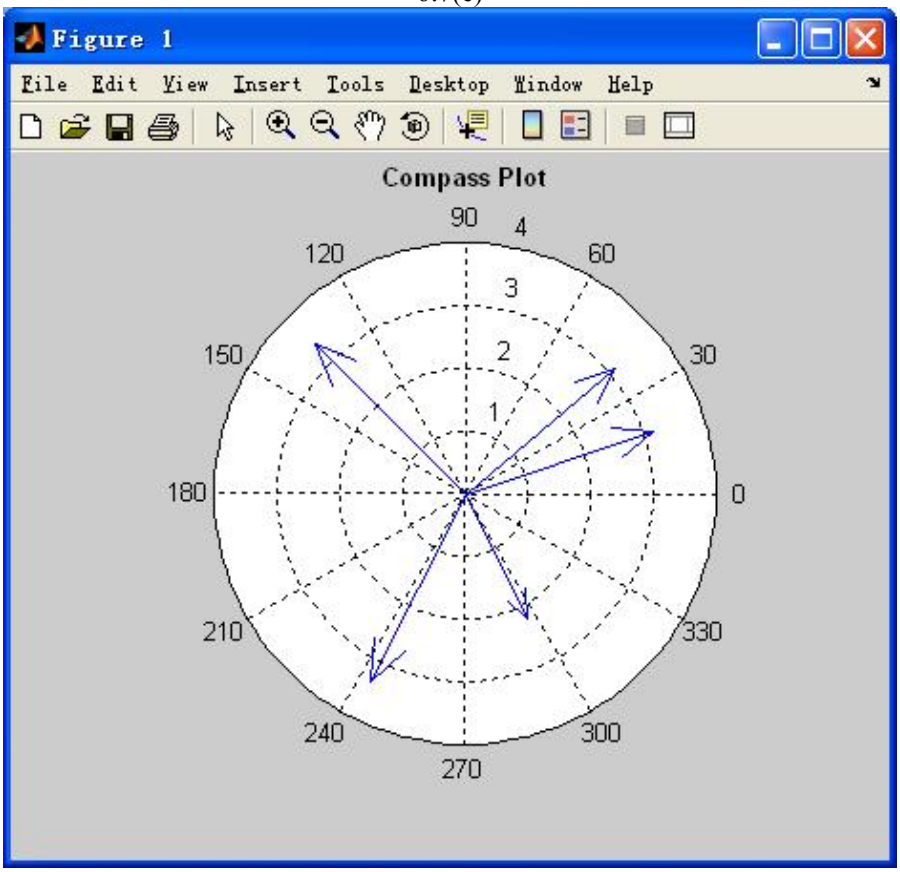


Figure 1
File Edit View Insert Tools Desktop Window Help
Example of a Horizontal Bar Plot
y
x
1 2 3 4 5 6
0 1 2 3 4 5 6 7

6.7(d)



6.7(e)



6.7(f)

图 6.7 各种图象

6.4.2 作图函数

在前面的所有作图，我们必须创建数组，并把这些数组传递给作图函数。**MATLAB** 提供了两个函数可以直接作出图象，而不需要创建中间数据数组。它们是函数数 ezplot 和 fplot。

ezplot 调用函数的形式如下

ezplot(fun);

ezplot(fun, [xmin xmax]);

ezplot(fun, [xmin xmax], figure);

其中，fun 代表一个字符串，用来表示要画的基本表达式。选择性参数[xmin，xmax]指定自变量的取值范围。如果它不存在的话，函数自变量的范围从-2π到 2π。选择性参数图来用指定图象数。

例如，下面语句打印出函数 f(x)=sinx/x，x 的取值范围在-4π 到 4π，输出图象如图 6.8 所示。

```
ezplot('sin(x)/x',[-4*pi 4*pi]);
title('Plot of sinx/x');
grid on;
```

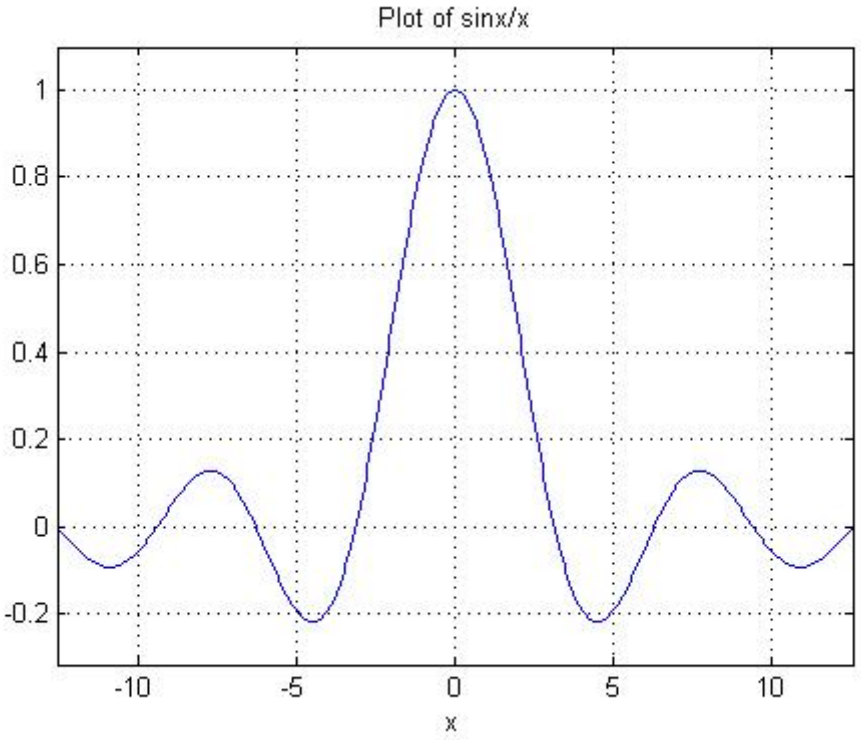


图 6.8 函数 sin(x)/x 的图象

函数 fplot 与 ezplot 相类似，但更加精确。前两个参数与函数 ezplot 中的相同，但是函数 fplot 还有其他优点。

1. 函数 fplot 是适应性的，它意味着在自变量范围内函数突然变化显示更多的点。
2. 函数 fplot 支持 T_EX 命令，用来指定坐标图的标题和坐标轴标签，而函数 ezplot 则不能。

在一般情况下，在画函数图象时，你应当使用函数 fplot。

函数 ezplot 和 fplot 是第五章中“函数的函数”的具体例子。

好的编程习惯

使用 fplot 函数直接打印函数，而不需创建中间数据数据。

6.4.3 柱状图

柱状图用来显示一系列数据数值的分布。为了创建一个柱状图，在一系列数值中范围被平均划分，并确定某一个范围中数值的个数。并把这个数目通过函数画出图来。

标准的 **MATLAB** 柱状图函数应为 hist。函数的形式如下：

```
hist (y)
hist(y, nbins)
his(y, x);
[n, xout] =hist(y, ...)
```

第一个函数创建并画出一个 10 等分的柱状图，而第二种形式创建的是以 nbins 为宽度的柱状图。第三个函数允许用户用数组 x 指定柱状图中长条的中心。这个函数创建柱状图长条都是以数组中的元素为中心的。这三种形式均能创建柱状图。这个函数的最后一种形式创建了一个柱状图并返回了一个数组 xcout，在数组 n 中的每一长条的数目，而实际上并没有创建一个图象。

例如，下面的语句创建一个包含有 1000 个符合正分布的随机数数组并产生了一个取值范围 15 等分的柱状图。产生的图象如图 6.9 所示。

```
y = randn(10000, 1);
hist(y, 15);
```

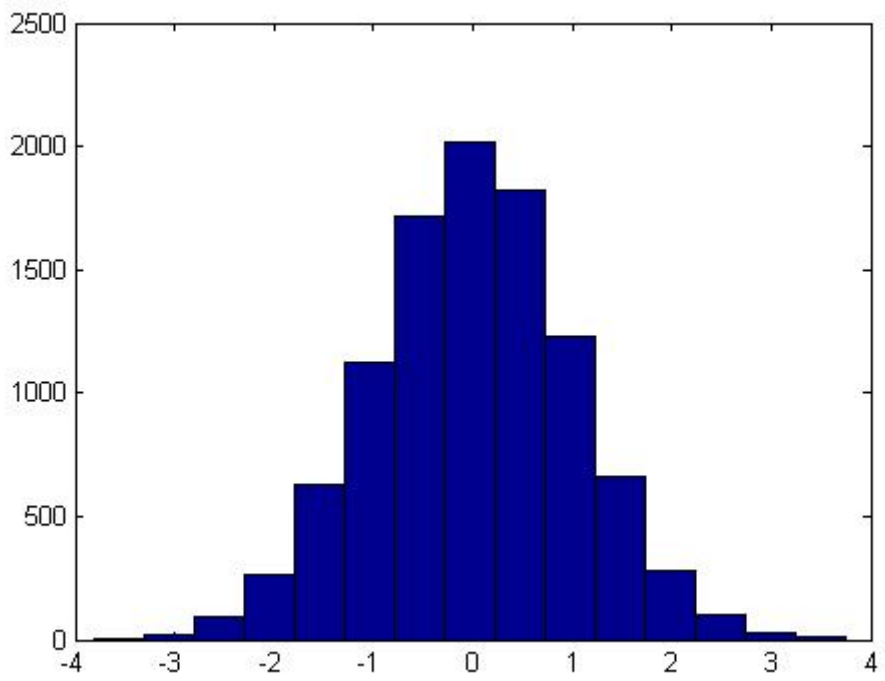


图 6.9 柱状图

MATLAB 提供了函数 rose 用来创建极坐标系中的柱状图。对于研究角度的分布非常有用。你将会在章末的练习中用到。

6.5 三维作图

MATLAB 中包括了丰富的三维图象，用来显示各式各样的数据。在一般情况下，三维

图象常用于显示以下两类数据。

1. 两个变量是同一自变量的函数，当你希望显示自变量重要性时，你可以用三维作图表示

2. 一个变量是另外两个变量的函数

6.5.1 三维曲线作图

我们可以用 plot3 函数进行三维曲线的作图。这个函数与二维 plot 函数非常相似，除了每一个点用 x，y，z 表示，而不用 x，y 表示。它的最简单的函数为

plot(x, y, z);

其中 x，y，z 是等大小的数组，它们组成了这个点的 3 维坐标。函数 plot3 提供了和 plot 函数相同的线型，大小和颜色，你直接利用前面学到的知识，画出一定的图形。作为一个三维曲线的例子，考虑下面的函数

$$x(t) = e^{-0.2t}\cos 2t$$
$$y(t) = e^{-0.2t}\sin 2t$$

这些函数表示在二维机械系统振动衰退情况，所以 x，y 代表在时刻 t 系统的位置。注意 x，y 有一相同的自变量 t。

我们可以创建一系列（x，y）并用二维函数 plot 画出（x，y）的图象，但是我们如果这样作了，时间的重要性就得不到体现。下面的语句创建了物体位置的一个二维图象，如图 6.10a。这个图不可能告诉我们振动变化的快慢。

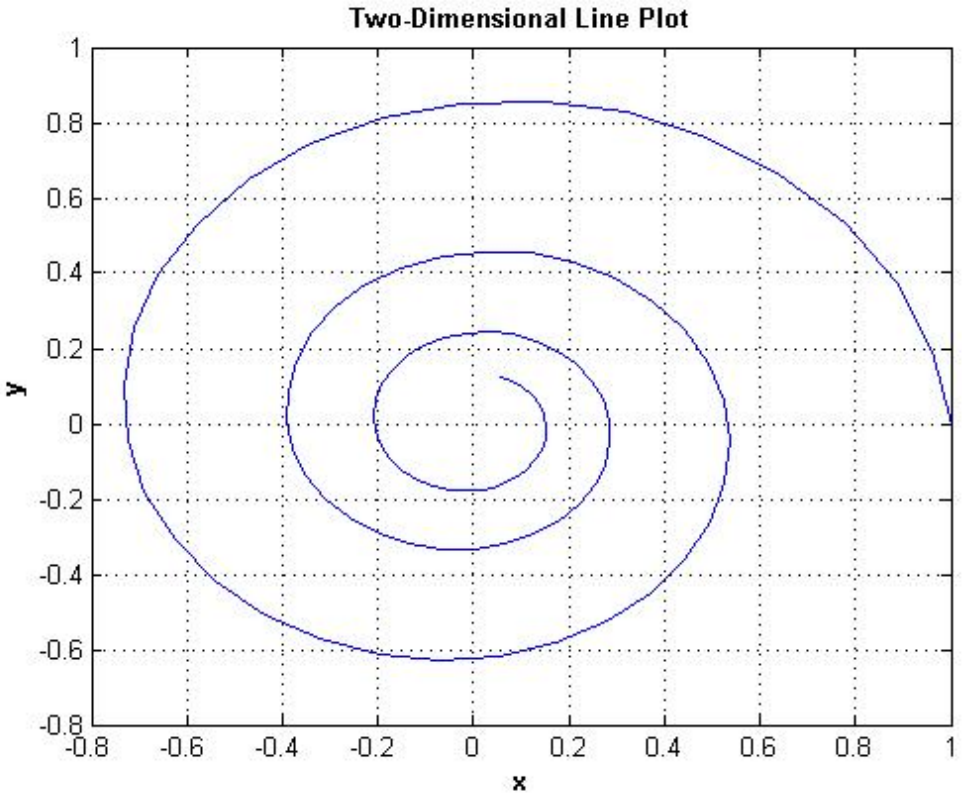


图 6.10(a)

我们可以利用 plot3 函数画出时间物体位置的三维图象。下面的语句将会创造 6.11 的三维图象。

```
t = 0:0.1:10;
x = exp(-0.2*t) .* cos(2*t);
y = exp(-0.2*t) .* sin(2*t);
plot3(x,y,t);
```

```
title('\bfThree-Dimensional Line Plot');
xlabel('\bfx');
ylabel('\bfy');
zlabel('\bfTime');
axis square;
grid on;
```

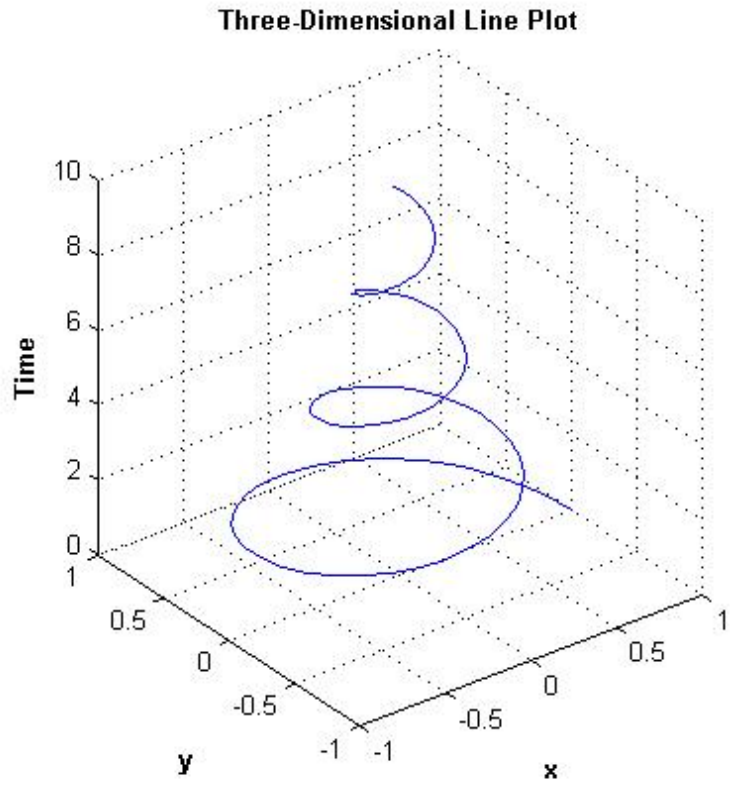


图 6.10(b)

(a)用二维图象代表物体的位置

(b)用三维图象来表示指定时间内的物体的位置产生的图象如图 6.0b 所示。注意这个图象显示了时间的独立性。

6.5.2 三维表面，网格，等高线图象

表面，网格，等高线图象是非常简单的方法来表示两变量的函数。例如，东西（x）南北（y）点上的温度。任何两变量函数的值都能用表面，网格，或等高线图象表示。有更多作图类型出现表 6。4 中，每一个图象的例子显示在图 6。11 中。

利用其中一个函数画图，用户必须创建三个等大小的数组。这个三个数组必须包括要画的每一点的 x，y，z 值。举一个简单的例子，假设我们要 4 个点（-1，-1，1），（1，-1，2），（-1，1，1）和（1，1，0），为了画出这 4 个点，我们必须创建三个数组

$$x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, y = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \text{ 和 } z = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}$$

数组 x 包括要画得每一点的 x 值，数组 y 包括要画得每一点的 y 值，数组 z 包括要画得每一点的 z 值。这些数组被传递到画图函数。

表 6.4 表面，网格，等高线图象函数

函数	描述
mesh(x, y, z)	这个函数创建一个三维网格图象。数组 x 包括要画得每一点的 x 值，数组 y 包括要画得每一点的 y 值，数组 z 包括要画得每一点的 z 值。
surf(x, y, z)	这个函数创建一个三维表面图象。x，y，z 代表的意义与上式相同。
contour(x, y, z)	这个函数创建一个三维等高线图象。x，y，z 代表的意义与上式相同。

MATLAB 函数 meshgrid 使函数图象数组 x，y 的创建变得十分容易。这些函数的形式为

[x y] = meshgrid(xstart:xinc:xend, ystart:yinc:yend);

xstart:xinc:xend 指定 x 值，ystart:yinc:yend 指定 y 值。

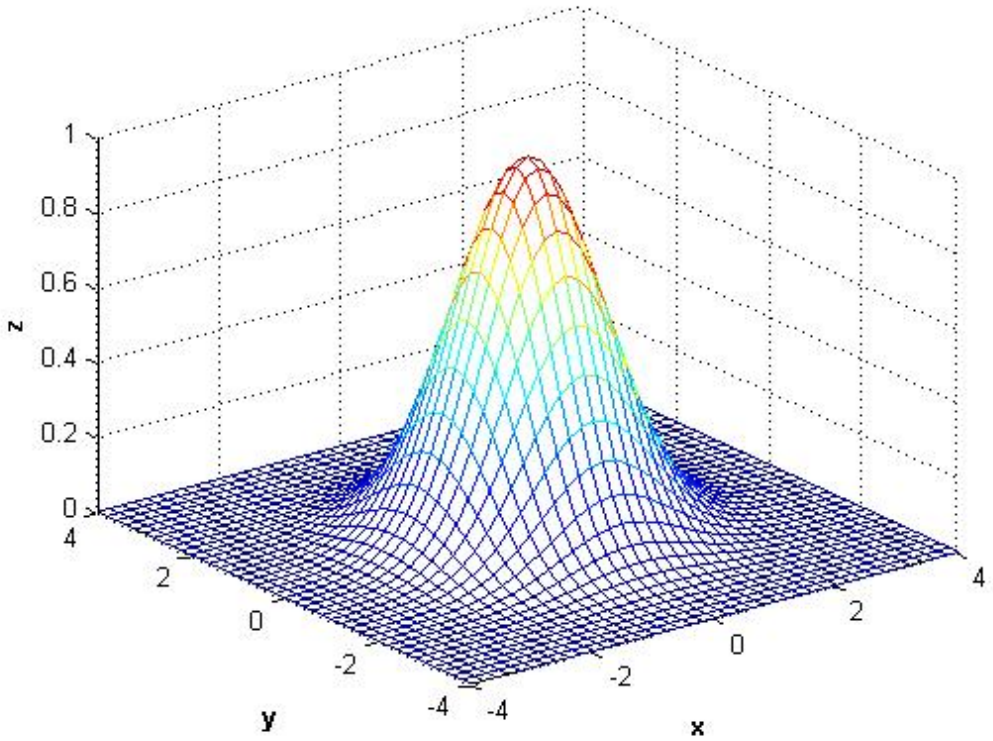
为了创建一个图象，我们要应用 meshgrid 来创建 x，y 的值，并通过函数计算（x，y）相对应的值。最后我们调用函数 mesh，surf 或 contour 来创建图象。例如，我们考虑下面函数的网格图象。

$$z(x, y) = e^{-0.5(x^2 + y^2)} \qquad (6.12)$$

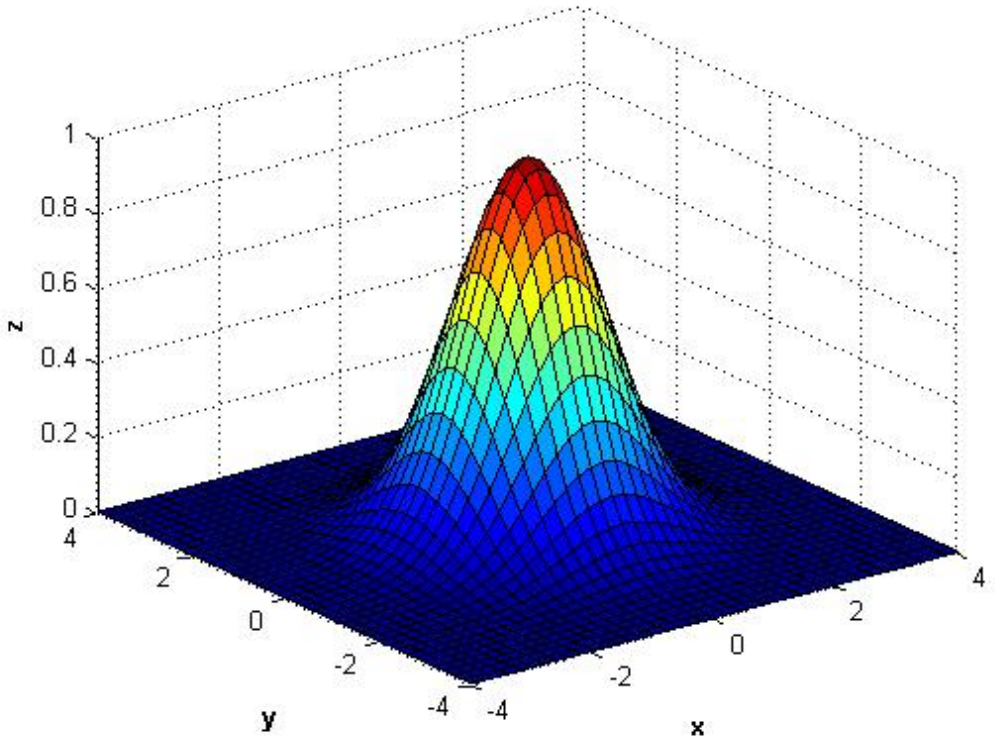
x，y 的取值分别为[-4, 4]，[-4,4]。下面语句将会创建这个图象，如图 6.11a 所示

```
[x,y] = meshgrid(-4:0.2:4,-4:0.2:4);
z = exp(-0.5*(x.^2+y.^2));
mesh(x,y,z);
xlabel('\bfx');
ylabel('\bfy');
zlabel('\bfz');
```

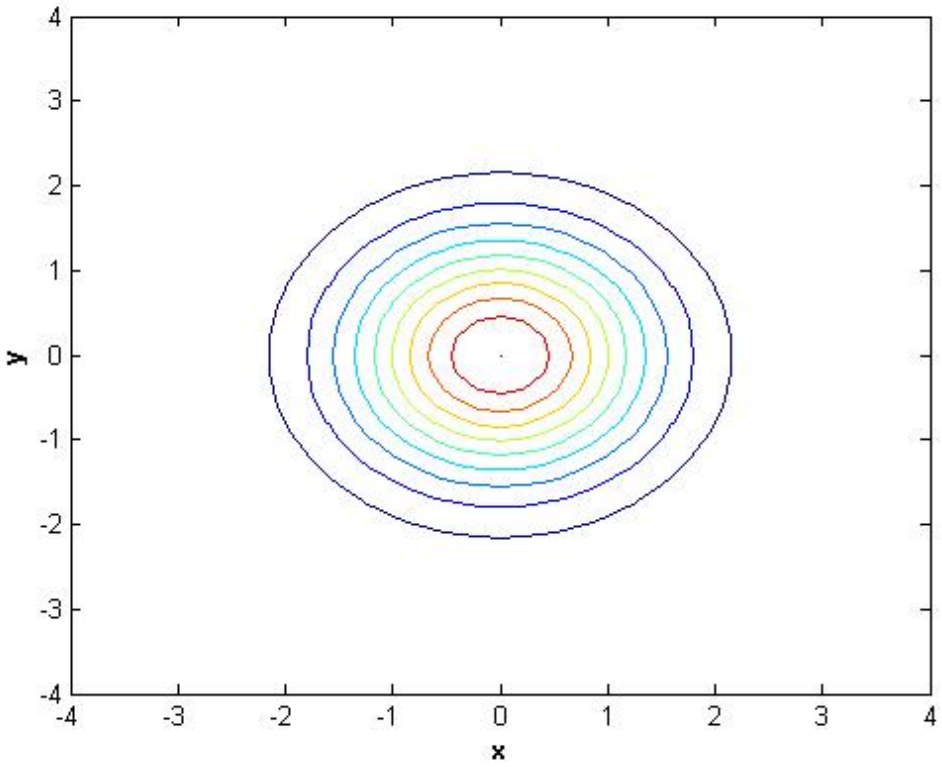
表面，等高线图象可以类似于 mesh 函数创建。



6.11(a)



6.11(b)



611(c)

图 6.11 (a)(b)(c)分别代表函数$z(x, y) = e^{-0.5(x^2 + y^2)}$ 网格图，表面图和等高线图。

6.6 总结

MATLAB 支持复数，作为 double 数据类型的扩展。复数用 i 和 j 定义，它们都被预定义为$\sqrt{-1}$ 。你可以直接应用复数进行运算，但要注意关系运算符只对复数的实部进行比较，而不对它的模进行比较。当你进行复数运算这是一个常见错误。

字符串函数是进行字符串操作。字符串是 char 型数组。这些函数允许用户对字符串进行各种各样的操作，例如连接，比较，替换，查找，大小写转换，数字与字符串之间的转换。

多维数组是指超过两维的数组。它们可用创建一维，二维数组的方法进行创建。多维数组用于解决自然界的一些问题。

MATLAB 中包含了许多的二维三维的作图方法，在本章中我们向大家介绍了针头图（stem Plots），阶梯图（stair plots），条形图，饼图（pie plots），罗盘图（compass plots），三维表面，网格，等高线图象。

6.6.1 好的编程习惯总结

下面是要我们遵守的指导原则

1. 用 char 函数创建二维字符数组，我们就不用担心每一行的长度不相同了。
2. 我们可以利用多维数组来解决自然界的多变量问题，如空气动力学和流体力学。
3. 使用 fplot 函数直接打印函数，而不需创建中间数据数据。

6.6.2 MATLAB 函数与命令总结

函数	描述
char	(1)把数字转化为相应的字符值
	(2)把二维数组转化相应的字符串
double	把字符转化为相应的 double 值
blanks	创建一个由空格组成的字符串
deblanks	去除字符串末端的空格
strcat	连接字符串
strvcat	竖直地连接字符串
strcmp	如果两字符串相等，那么函数将会返回 1
stricmp	忽略大小写如果两字符串相等，那么函数将会返回 1
strncmp	如果两字符串的前 n 个字母相等，那么函数将会返回 1
strncmpi	忽略大小，如果两字符串的前 n 个字母相同，那么数将会返回 1
findstr	在一个字符串中寻找另一个字符串
strfind	在一个字符串中寻找另一个字符串（版本 6。1 或以后的版本）
strjust	对齐字符串
strmatch	找字符串的区配
strrep	用一个字符串去替代另一个字符串
strtok	查找一字符串
upper	把字符串的所有字符转化为大写
lower	把字符串的所有字符转化为小写
int2str	把整数转化为相应的字符串形式

函数	描述
num2str	把数字转化为相应的字符串形式
mat2str	把矩阵转化为相应的字符串形式
sprintf	对一字符串进行格式化输出
str2double	把字符串转化相应的 double 型数据
str2num	把字符转化成数字
sscanf	从字符串中读取格式化数据
hex2num	把 IEEE 十六进制字符型型数据转化为 double 形数据
hex2dec	把十六制字符串转化为相应的十进制整数
dec2hex	把十进制数转化为相应的十六制字符串
bin2dec	把二进制字符串转化为相应的十进制整数
base2dec	把 baseb 转化为相应的十进制数据
dec2base	把十进制转化为相应的 baseb
bar(x, y)	这个函数用于创建一个水平的条形图，x 代表第一个 X 轴的取值，y 代表对应于 Y 的取值
barh(x, y)	这个函数用于创建一个竖直的条形图，x 代表第一个 X 轴的取值，y 代表对应于 Y 的取值
compass(x, y)	这个函数用于创建一个极坐标图，它的每一个值都用箭头表示，从原点指向（x，y），注意：（x，y）是直角坐标系中的坐标。
pie(x) pie(x, explode)	这个函数用来创建一个饼状图，x 代表占总数的百分数。explode 用来判断是否还有剩余的百分数
stairs(x, y)	用来创建一个阶梯图，每一个阶梯的中心为点(x, y)
stem(x, y)	这个函数可以创建一个针头图，它的取值为(x,y)

6.7 练习

6.1

图 6.12 显示的是一个 *RLC* 电路，它的激励产生一个 $120\angle 0^{o}$V 的电压。这个电路中的感抗为 $Z_L=j*2\pi fL$，其中 j 为$\sqrt{1}$ ，f 是电压的频率，单位为 Hz。这个电路的容抗为 $Z_C=-j*\frac{1}{2\pi fC}$ ，C 为电容。假设 R=100Ω，L=0.1mH，C=0.25nF。

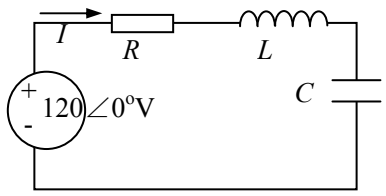


图 6.12 *RLC* 电路

根据基尔霍夫电压定律我们可以得到电流$I=\frac{120\angle 0^{o}\text{V}}{R+j2\pi fL-j\frac{1}{2\pi fC}}$

a. 计算并画出以频率为自变量的电流模函数的图象，要求频率从 100KHz 到 10MHz 变化。画出图要求有两个，一个是线性图，一个是对数线性图。保证包括标题和坐标标签。

b. 计算并画出以频率为自变量的电流相角函数的图象。要求频率从 100KHz 到 10MHz

变化。画出图要求有两个，一个是线性图，一个是对数线性图。保证包括标题和坐标标签。

c.画出以频率为自变量的电流相角，模函数的图象。要求在同一图象窗口内画出两个图象。用对数线性标度。

6.2

编一个函数 to_polar，可以接受一个复数，并返回两个输出参数——复数 C 的模 mag 和角度 theta。输出角度的单位为度。

6.3

编写一个函数 to_complex，可以接受两个输入参数——复数 C 的模 mag 和角度 theta，返回复数 C。

6.4

在一个稳定的交变电路中，通过一中性电路元件时符合欧姆定律为

$$V=IZ \qquad (6.13)$$

V 代表这个元件两端的电压，I 代表通过这个器件的电流，Z 代表这个元件的阻抗。注意这个三个数都是复数。而这些数一般用模与相角的形式表示。例如，电压可能为 $V=120\angle 30^{o}$。

编写一个程序，读取负载两端的电压，和负载的阻抗，并计算出电流值。输入应为模与相角，输出也应为这种格式。用练习 6.3 和 6.2 中的程序进行格式转换。

6.5

编写一个程序，接受一个复数 C 并在笛卡尔坐标系中用圆圈画出相应的点。图象应包括 x，y 轴，并要求生成一个由原点指向 C 的向量。

6.6

调用函数 plot(t，v)画出数学函数 $v(t)=10e^{(-0.2+j\pi)t}$ 在 0≤t≤10 时图象。这个图象是什么？

6.7

调用函数 plot(v)画出数学函数 $v(t)=10e^{(-0.2+j\pi)t}$ 在 0≤t≤10 时图象。这次图象是什么？

6.8

创建一个数学函数在 0≤t≤10 时的极坐标图。

6.9

调用函数 plot(t，v)画出数学函数 $v(t)=10e^{(-0.2+j\pi)t}$ 在 0≤t≤10 时图象，图象的每一维分别是函数的实部，虚部和时间。

6.10

欧拉公式。公式定义如下

$$e^{i\theta} = \cos\theta + i\sin\theta \qquad (6.14)$$

创建一个二维图象，θ 的取值 0 到 2π 之间。

创建一个三维图象，θ 的取值 0 到 2π 之间。

$$z = e^{x+iy}$$

6.11

创建函数 $z=e^{x+iy}$ 在-1≤x≤1 和-2π≤y≤2π 内的三维网图，表面图和等高线图。

6.12

编写一个程序，从用户接受一个字符串，并确定用户指定的字母出现在字符串多少次（提示：在 **MATLAB** 帮助工作台中查找 input 函数的“s”参数选项。）

6.13

修改上面的程序，确定用户指定的字母出现在字符串多少次，忽略字母的大小写。

6.14

编写一个程序，用 input 函数接受一个字符串，并对这些字符串分解成各种符号，然后对这些符号进行按升序排序，并把他们打印出来。

6.15

编写一个程序，用 input 函数接受一系列字符串，并对这些字符串按升序进行排序，并把他们打印出来。

6.16

编写一个程序，用 input 函数接受一系列字符串，并对这些字符串按升序进行排序，忽略大小写，并把他们打印出来。

6.17

MATLAB 中包含两个函数 upper 和 lower，它分别把字符串转化为大字和小写。创建一个新的函数 caps，让字符串每个单词的第一个字母大写，其余的为小写。

6.18

画出函数 $y=e^x \sin x$ 的针头图，阶梯图，条形图，罗盘图。确保所有图象中都有标题和坐标轴标签。

6.19

假设 George，Sam，Betty，Charlie 和 Suzie 去送别一位同事，他们为买礼物分别花了 $5，$10，$7，$15。创建一个饼图，Sam 花得钱占总数的百分比为多少？

6.20

用函数 fplot 画出函数 $f(x)=\frac{1}{\sqrt{x}}$ ，x 在 $0.1 \leqslant x \leqslant 10.0$ 中的图象。