目录

第二章 MATLAB 基础

在本章我将向大家介绍 MATLAB 的基本元素。在本章的章末，你将会编写简单的函数化的工具。

2.1 变量和数组

MATLAB 程序的基本数据单元是数组。一个数组是以行和列组织起来的数据集合，并且拥有一个数组名。数组中的单个数据是可以被访问的，访问的方法是数组名后带一个括号，括号内是这个数据所对应行标和列标。标量在 MATLAB 中也被当作数组来处理——它被看作只有一行一列的数组。

数组可以定义为向量或矩阵。向量一般来描述一维数组，而矩阵往往来描述二维或多维数组。在本书中，当我们讨论一维数组时用向量表示，当我们讨论二维或多维向量时用矩阵。如果在特殊情况下，同时遇到这两种数组，我们就把他们通称为“数组”。

数组的大小（size）由数组的行数和列数共同决定,注意行数在前。一个数组所包含的数据多少可由行数乘列数得到。例如，下列数组的大小为

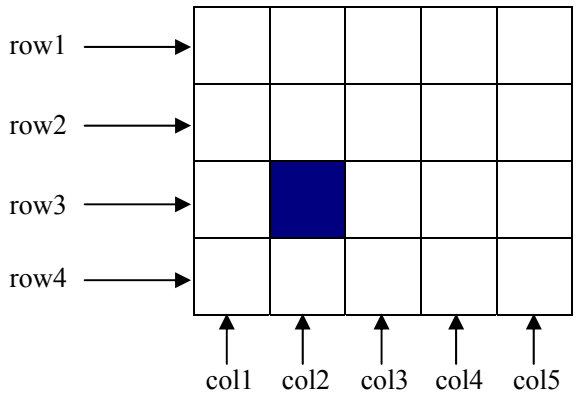


图 2.1 一个数组是以行和列组织起来的数据集合,此数组 arr 含有 20 个元素，共 4 行，5 列。阴影元素是 arr(3,2)

数组	大小
$A=\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}$	这是一个 3×2 矩阵，包含 6 个元素
$B=\begin{bmatrix}1 & 2 & 3 & 4\end{bmatrix}$	这是一个一维行向量，共有 4 个元素
$C=\begin{bmatrix}1\\ 2\\ 3\\ 4\end{bmatrix}$	这是一个一维行向量，共有 4 个元素

数组中的单个数据是可以被访问的，访问的方法是数组名后带一个括号，括号内是这个

数据所对应的行标和列标。如果这个数组是一个行向量或列向量，则只需要一个下标。例如上面的数组 A（2 1）为 3，C（2）为 2。一个 MATLAB 变量是一段包含一个数组的内存区，并且拥有一个用户指定的变量名。通过适当的命令和它的变量名随时可以就调用它和修改它。

MATLAB 的变量名必须以字母开头，后面可以跟字母，数字和下划线（_）.只有前 31 个字符是有效的；如果超过了 31 个字符，基余的字符将被忽略。如果声明两个变量，两变量名只有第 32 个字符不同，那么 MATLAB 将它们当作同一变量对待。

常见编程错误

确保你所声明的变量名前 31 个字符是独一无二的。否则，MATLAB 将无法辨认出它们的不同。

当你编写程序时，给变量起一个有意义的名字非常的重要。有意义的名字极大的提高了程序的可读性和可维护性。像 day，month 和 year 这样的名字意义非常明确，即使第一次看到也能理解。尽管空格不能用在 MATLAB 变量名中，但是可以用下划线代替空格创造出有意义的变量名。比如，changerate 可以写成 change_rate.

好的编程习惯

给你的变量起一个描述性的且易于记忆的变量名。例如，货币汇率可以 exchange_rate 为变量名。这种方法将使得你的程序更加明确且易于理解。

在你所写的程序的开头列出一数据字典（data dictionary）十分的重要.数据字典列举了你在本程序中用到的所有变量的定义。它的定义应包括本条目的所要描述的内容和它在执行时所在的单元。当编写程序时，编定数据字典看似没有必要。但是设想一下，在过了一段时间后，你或其他人要对此程序修改，这时数据字典就显得十分的有用。

好的编程习惯

给每个程序创建一个数据字典以增强程序的可维护性。

在 MATLAB 语言中是区分字母大小的，也就是说，大写字母和小写字母代表的东西是不同的。所以变量 NAME,Name，name 在 MATLAB 中是不同的。所以已用过的小写变量名与一个新建大写的变量名重名，这时使用时要特别地小心。在一般情况下，我们一律用小写字母来表示。

好的编程习惯

在每次用到一个变量时，我们要确保变量名的大小写的精确匹配。在变量名中只使用小写字母是一个好的编程习惯。

两个最常见的变量类型是 char 型和 double 型。double 型的变量包括由 64 位双精度浮点数构成的标量或数组。这种变量可以代表实数，虚数和复数。每个值的实部和虚部的变化范围为正负 10^{-308}~10^{308}，拥有 15 到 16 位有效数字。这是基本的数字数据类型。

无论什么时侯，你将一个数值赋值于一个变量名，那么 MATLAB 将自动建立一个 double 型变量。例如，下面语句创建了一个以 var 为变量名的 double 型变量，包含了一个 double 型的单个元素，存储了复数值（1+i）；

```
var=1+i;
```

char 型的变量包括由 16 位数值构成的标量或数组，每一个 16 位数代表一个字符。这个类型的经常用于字符串操作，当一个字符或字符串赋值于一个变量名时，系统会自动建立一个 char 型变量。例如，下面的这个语句创建了一个 char 型变量 comment，并存储了一个字符串在其内。当这个语执行后，系统将会建立一个 1×26 的字符串数组。

```
comment='this is a character string';
```

像 C 语言这样的语言中，变量类型和变量在使用之前必须强制声明。这种语言我们叫它**强类型语言**。相对地，像 MATLAB 这样的叫做**弱类型语言**。通过简单的赋值形式就可以创建变量，变量类型取决于创建时的类型。

2.2 MATLAB 变量的初始化

当变量初始化时，MATLAB 将会自动建立变量。有三种方式初始化 MATLAB 中的变量：

1．用赋值语句初始化变量

2．用 input 函数从键盘输入初始化变量

3．从文件读取一个数据

前两种方法我们在这里讨论，第三方法我们将在 2.7 节介绍。

2.2.1 用赋值语句初始化变量

最简单的创建和初始化一个变量的方法是用赋值语句赋予变量一个或多个值。赋值语句的一般形式如下

var = expression

var 是变量名，expression 可以是一个标量、一个数组或常量、其他变量和数学运算符号（+、-）的联合。这个表达式（expression）的值是通过一般的数学运算法则计算出来的，然后将产生的结果存储到变量 var 中。下面是一些用赋值语句初始化的变量：

```
var=40*i;
var2=var/5;
array=[1 2 3 4];
x=1;
y=2;
```

第一个例子创建了一个 double 类型的标量变量，存储了一个虚数 40i。第二个例子创建了一个表达式 var2，把 var/5 的值存储于内。第三个例子创建了一个数组 array，并存储了一个 4 元素的行向量于内。最后一个例子显示了多个赋值语句可写在同一行，中间用逗号或分号隔开。注意如果在赋值语句执行时变量已经存在，那么这个变量原有的值将被覆盖。

正如第三个例子显示的，数据数组也可以初始化变量。我们可以用是括号（）和分号建立数组。所有元素按行阶排序，换句话说，每一行的值从左向右，顶部的行置于最前，底部的行置于最后。在一行内单个数值可用空格或逗号隔开，而行与行之间要与则用分号隔开，或另起一行书写。下面的表达式都是合法的，能用于建立一个变量：

[3.4]	这个表达式创建了 1×1 数组(一个标量),包含数值 3.4. 这时括号可以省略.
[1.0 2.0 3.0]	这个表达式创建了 1×3 数组,一维行向量[1 2 3]
[1.0;2.0;3.0]	表达式创建了一个 3×1 数组,一维列向量 $\begin{bmatrix}1\\2\\3\end{bmatrix}$
[1,2,3;4,5,6]	这个表达式创建了一个 2×3 数组,矩阵 $\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix}$
[1,2,3 4,5,6]	这个表达式创建了一个 2×3 数组,矩阵 $\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix}$
[]	是个空数组,没有行,没有列(注意他与元素全为零的数组的区别)

注意一个数组每一行元素的个数必须完全相同,每一列元素的个数也必须完全相同.像[1 2 3;4 5]这样的表达式是非法的,因为第一行有 3 个元素,第二行有只有 2 个元素.

常见编程错误

每一行元素的个数必须完全相同,每一列元素的个数也必须完全相同.试图创建一个不同行(列)拥有不同数目元素的数组,在编译时将会出现错误.

用于初始化数组的表达式可以包括代数符号或事先已经定义好的数组.例如赋值语句

```
a=[0 1+7]
b=[a(2) 7 a]
```

定义了数组 a=[0 8]和数组 b=[8 7 0 8].

当我们创建一个数组时,不是每一个元素都必须定义.如果要定义一个特殊的数组,或只有一个或几个元素没有定义,那么之前的那些元素将会自动创建,并初始化为 0.例如,如果数组 c 事先没有定义,语句 c(2,3)=5 将会创建一矩阵 $c=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 5\end{bmatrix}$。相似地，指定一个值赋予一个存在的数组，但超过了这个数组的大小。例如，假设存在一数组 d=[1 2],下面这个语句

```
d(4)=4;
```

将会制造出数组 d=[1 2 0 4].

在每个赋值语句末的分号有特殊的目的：无论在何时一个表达式在赋值语句中被赋值，分号将会中止变量值的重复。如果句末没有分号，变量值将会自动显示在命令窗口(The Command Windows)中。

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

如果在赋值语句末有分号，这种重复将会消失。重复是一个用于检查你的工作极好的方法，但是它降低了运行速度。因此，我们在一般情况下总是禁止重复。尽管如此，重复计算的结果提供了一个强大的应急调试器。如果你不能确定一个特定的赋值语句结果是多少，这时你可以去掉这个语句后的分号，当这个语句编译时，结果会显示在命令窗口(The Command Windows)。

好的编程习惯

在 MATLAB 赋值语句后加上一个分号来禁止变量值在命令窗口(The Command Windows)的重复。这将大大提高编译的速度。

好的编程习惯

如果你在调试程序时需要检测一个语句的结果，可能把句后的分号去掉，这样结果将会出现在命令窗口(The Command Windows)。

2.2.2 用捷径表达式（short expressions）来赋值

创建一个小数组用一一列举出元素的方法是比较容易的，但是当我们创建包括成千上万个元素的数组怎么办？把每一个元素列举出来则不太现实。

MATLAB 提供一种专门的捷径标记法，这种方法用克隆运算符（colon operator）适用于上述情况。克隆运算符指定一系列的数值，它指定了这个系列数的第一值，步增和最后一个值。它的一般顺序始下

first:incr:last

first 代表数组的每一个值，incr 代表步增量，last 代表这个数组的最后一个值。如果步增量为 1，那么步增量可省略，而变成了 first:last 格式。

例如，表达式 1:2:10 是创建一个 1×5 行向量[1 3 5 7 9]的简便方法。

```
>> x=1:2:10
x =
     1     3     5     7     9
```

用克隆标记法初始化一个含有一百个元素的数组$\left[\frac{\pi}{100},\frac{2\pi}{100},6\ ,\pi\right]$，语句如下

```
Angles=(.01:.01:1)*pi
```

捷径表达式可以联合转置运算符（’）来初始化行向量，或更加复杂的矩阵。转置运算符可以在需要的情况下完成行和列的转换。因为这个表达式

```
f=[1:4]';
```

产生一个 4 元素行向量[1 2 3 4]，然后将其转换成 4 元素的列向量 $f=\begin{bmatrix}1\\2\\3\\4\end{bmatrix}$;相似地，表达式

```
g=1:4;
h=[g' g']
```

将会创建一个矩阵 $h=\begin{bmatrix}1&1\\2&2\\3&3\\4&4\end{bmatrix}$

2.2.3 用内置函数来初始化

数组也可以用 MATLAB 内置函数创始化。例如，函数 zeros 可以初始化任何大小的全为零的数组。用许多形式的 zeros 函数。如果这个函数的参数只是一个标量，那么 MATLAB 将会创建一个方阵，行数和列数均为这个参数。如果这个函数有两个标量参数，那么第一个参数代表行数，第二个参数代表列数。因为 size 函数所返回的一个数组的行数和列数，所以它可以联合 zeros 函数来创建一个相同大小零矩阵。下面是一些用到 zeros 函数的例子。

```
a=zeros(2);
b=zeros(2,3);
c=[1 2;3 4];
d=zeros(size(c))
```

这些语句产生下列的数组

$a=\begin{bmatrix}0&0\\0&0\end{bmatrix}$　　$b=\begin{bmatrix}0&0&0\\0&0&0\end{bmatrix}$

$c=\begin{bmatrix}1&2\\3&4\end{bmatrix}$　　$d=\begin{bmatrix}0&0\\0&0\end{bmatrix}$

相似地,ones 函数产生的数组包含的元素全为 1,eye 函数通常用来产生单位矩阵,只有对角线的元素为 1.其他元素为 0.表 2.1 列出一些用于初始化变量的函数.

表 2.1　用于创始化变量的 MATLAB 函数

函数	作用
zeros(n)	创建一个 n×n 零矩阵
zeros(n,m)	创建一个 n×m 零矩阵
zeros(size(arr))	创建一个与数组 arr 的零矩阵
ones(n)	创建一个 n×n 元素全为 1 矩阵
ones(n,m)	创建一个 n×m 元素全为 1 矩阵
eye(n)	创建一个 n×n 的单位矩阵
eye(n,m)	创建一个 n×m 的单位矩阵
length(arr)	返回一个向量的长度或二维数组中最长的那一维的长度
size(arr)	返回指定数组的行数和列数

2.2.4 用关键字 input 初始化变量

关键字 input 用来提示使用者和直接从键盘输入初始化变量.当脚本文件(Script files)时,它可以用来提示使用者输入.input 函数在命令窗口(The Command Windows)显示提示语句,并等待用户输入一个值.例如,下面的赋值语句:

```
my_val=input('enter an input value:')
```

当这个语句被编译时,MATLAB 打印出字符串 enter an input value:,然后等待用户回复.如果要只输入一个数,那么只需要直接键入,如果要输入一个数组,则必须带上中括号([]).不管怎样,当按下回车键时.在窗口输入的任何值都会被储入变量 my_val.如果只按下回车键,那么这个变量就存储了一个空矩阵.

如果 input 函数中有字符's'做为它的第二个参数,输入的数据就被当字符串.因此,语句

```
in1=input('enter data:');
enter data:1.23
```

把数值 1.23 存储到 in1 中.而语句

```
in2=input('enter data:','s')
enter data:123
```

把字符串 1.23 存储到 in2 中.

测试 2.1

本测试提供了一个快速的检查方式，看你是否掌握了 2.1 和 2.2 的基本内容。如果你对本测试有疑问，你可以重读 2.1 和 2.2，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.数组,矩阵,向量有什么区别?

2.回答关于下列矩阵的有关问题

$$C=\begin{bmatrix}1.1 & -3.2 & 3.4 & 0.6\\ 0.6 & 1.1 & -0.6 & 3.1\\ 1.3 & 0.6 & 5.5 & 0.0\end{bmatrix}$$

(a)C 的大小是多少?

(b)C(2,3)的值是多少?

(c)列出值为 0.6 的元素的下标

3.确定下列数组的大小，通过 whos 或工作空间窗口（The workspace browser）检查你的答案。注意在本练习中后面的数组可能要用到前面数组的定义。

(a) u=[10 20*i 10+20]

(b) v=[-1;20;3]

(c) w=[1 0 -9;2 -2 0;1 2 3]

(d) x=[u' v]

(e) y(3,3)=-7

(f) z=[zeros(4,1) ones(4,1) zeros(1,4)']

(g) v(4)=x(2,1)

4.w(2,1)的值是多少?

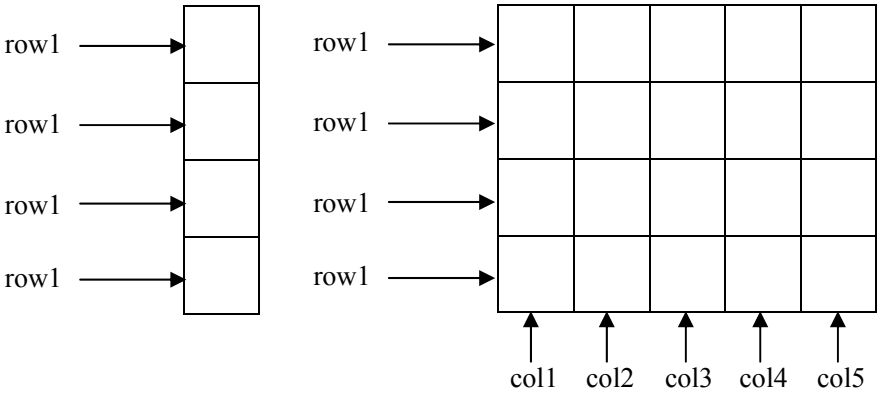
5.x(2,1)的值是多少?

6.y(2,1)的值是多少?

7.当语句（g）执行后，v(3)的值是多少?

2.3 多维数组

正如我们所看到的，MATLAB 的数组可能是一维或多维的。一维的数组可以形象地看作一系列的数垂直地罗列起来，用一个下标就可以调用数组中的元素（如图 a）。这样的数组适用于一个变量的函数，例如在规定的时间间隔后一系列的测量温度。



a1(irow)
(a)一维数组

a2(irow,icol)
(b)多维数组

许多数据的类型需要多变量的函数。例如，我要在 5 个不同的地方，每个地方测 4 次温度。在这种情况下，我们的 20 次测量结果在逻辑上分为五个不同的行，每行有 4 个测量结果（如图 b）.在这种情况下，我们就需要两个下标来调用这个数组特定的函数：第一个下标选择行，第二个下标选择列。这样的数组叫做**二维数组**。二维数组中元素的个数取决于这个数组的行数和列数。

出于问题的需要，MATLAB 允许我们创建多维数组。这些数组的每一维对应一个下标，和每一个单个元素都可以通过它的每一个下标被调用。在这个数组中元素的总和取决于每一维中元素的个数。例如，下面两个语句创建了一个 2×3×2 数组 c

```
>> c(:,:,1)=[1,2,3;4,5,6];
>> c(:,:,2)=[7,8,9;10,11,12];
>> whos c
  Name          Size                    Bytes  Class

  c             2x3x2                      96  double array

Grand total is 12 elements using 96 bytes
```

这个数组（2×3×2）包括 12 种元素，它的内容显示方法和其他数组的显示方法大体相同

```
>> c
c(:,:,1) =

     1     2     3
     4     5     6

c(:,:,2) =

     7     8     9
    10    11    12
```

2.3.1 多维数组在内存中的存储

一个有 m 行和 n 列的二维数组包括 m×n 个元素，这些元素在计算机的内存中将会占有 m×n 个连续的内存空间。这些数组的元素在内存中是如何排列的呢？MATLAB 以**列主导顺序**分配数组中的元素。也就是说，内存先分配第一列的元素，然后第二列，第三列，……以此类推，直到所有列都被分配完。图 2.3 说明 4×3 数组 a 的内存分配情况。正如我们所看到的，元素 a(1,2)是其实在内存分配的第五个位置。在下一节我们讨论用单一下标访问数

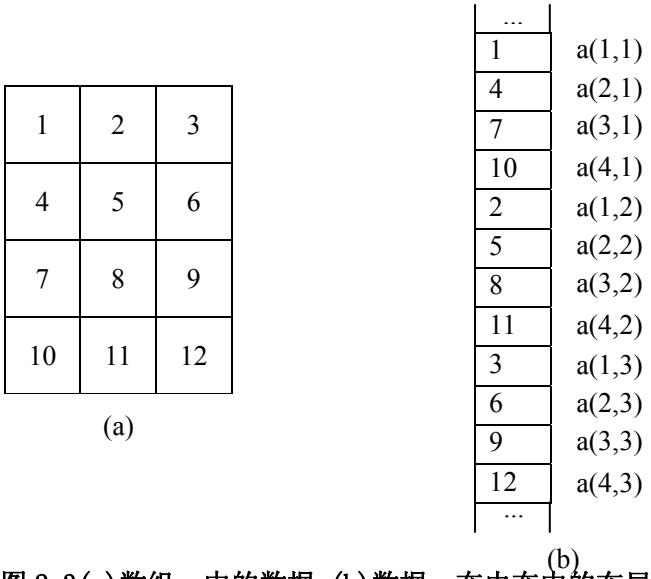


图 2.3(a)数组 a 中的数据 (b)数据 a 在内在中的布局

组元素和第八章低级 I/O 接口，内存分配元素的顺序将变得十分重要。

这种分配方式也适用于多维数组。数组的第一个下标增长最快，第二个仅次之，依此类推，最后一个变化最慢。例如，在一个 2×2×2 数组中，它的元素在内存中的分配顺序是（1，1，1），（2，1，1），（1，2，1），（2，2，1），（1，1，2），（2，1，2），（1，2，2），（2，2，2）。

2.3.1 用单个下标访问多标数组

MATLAB 的特性之一就是它允许使用者或程序员把一个多维数看作一个一维数组，这个一维数组的长度等于多维数组的元素数。如果用一个下标访问一个多维数组，那么元素的排列顺序就是内存的分配顺序。

例如，假设我们要声明一个 4×3 的数组如下：

```
>> a=[1 2 3
4 5 6
7 8 9
10 11 12]

a =

     1     2     3
     4     5     6
     7     8     9
    10    11    12
```

那么 a(5)的值为 5 和 a(1,2)的值相同，这是因为元素 a(1,2)排在内存第五个位置。

在一般情况下，我们不应使用 MATLAB 的这一特性。用单个下标访问多维数组可能会带很多的麻烦。

好的编程习惯

在访问多维数组时，总是使用合适的维数。

2.4 子数组

你可以选择和使用一个 MATLAB 函数的子集，好像他们是独立的数组一样。在数组名后面加括号，括号里面是所有要选择的元素的下标，这样就能选择这个函数的子集了。例如，

假设定义了一个数组 arr1 如下

```
arr1=[1.1 -2.2 3.3 -4.4 5.5]
```

那么 arr1(3)为 3.3,arr1([1 4])为数组[1.1 -4.4],arr1(1:2:5)为数组[1.1 3.3 5.5].

对于一个二维数组，克隆运算符可以用于下标来选择子数组。例如，假设

```
arr2=[1 2 3; -2 -3 -4;3 4 5]
```

将建立一个数组

$$\text{arr2}=\begin{bmatrix}1 & 2 & 3\\ -2 & -3 & -4\\ 3 & 4 & 5\end{bmatrix}$$

在这种定义下，子数组 arr2(1,:)为[1 2 3]，子数组 arr2(:,1:2:3)为 $\begin{bmatrix}1 & 3\\ -2 & -4\\ 3 & 5\end{bmatrix}$

2.4.1 end 函数

MATLAB 中有一个特殊的函数叫做 end 函数,对于创建子数组的下标非常的有用.当用到一个函数的下标时,end 函数将会返回下标的最大值.

例如,假设数组 arr3 定义如下:

```
arr3=[1 2 3 4 5 6 7 8];
```

那么 arr3(5:end)将会产生数组[5 6 7 8],arr3(end)将会产生值 8.

end 函数返回的值一般为所要下标的最大值.如果 end 函数显示有不同的下标,那它将在一个表达式内返回不同的值.例如,假设一个 3×4 数组 arr4 的定义如下:

```
arr4=[1 2 3 4;5 6 7 8;9 10 11 12]
```

那么表达式 arr4(2:end,2:end)将会返回 $\begin{bmatrix}6 & 7 & 8\\ 10 & 11 & 12\end{bmatrix}$.注意第一个 end 返回值为 3,第二个返回值为 4.

2.4.2 子数组在左边的赋值语句的使用

只要数组的形（行数和列数）和子数组的形相匹配，把子数组放于赋值语句的左边用来更新数组中的值。如果形不匹配，那么将会有错误产生。例如，下面有一个 3×4 数组定义如下：

```
>> arr4=[1 2 3 4;5 6 7 8;9 10 11 12]

arr4 =

     1     2     3     4
     5     6     7     8
     9    10    11    12
```

因为在等号左边的表达式的形（2×2）与 a 相匹配，那么下面的这个赋值语句是合法的。

```
>> arr4(1:2,[1 4])=[20 21;22 23]

arr4 =

    20     2     3    21
    22     6     7    23
     9    10    11    12
```

注意数组元素（1，1），（1，4）（2，1）和（2，4）得到了更新。相对而言，两边的形不相匹配，则表达式是非法的，例如下面这个表达式。

```
>> arr5(1:2,[1 4])=[20 21]
??? Subscripted assignment dimension mismatch.
```

常见编程错误

对于涉及子数组的赋值语句，等号两边的形必须相匹配。否则将会产生错误。

在 MATLAB 中用子数组赋值和用值直接赋值有很大的不同。如果用子数组赋值，那么只有相应的值得到更新，而其他的值保持不变。另一方面，直接赋值，则数组的原有内容全部删除并被新的值替代。例如，假设用一个数组 arr4 定义如下：

```
>> arr4=[1 2 3 4;5 6 7 8;9 10 11 12]

arr4 =

     1     2     3     4
     5     6     7     8
     9    10    11    12
```

下面的赋值语句，只更新特定的元素：

```
>> arr4(1:2,[1 4])=[20 21;22 23]

arr4 =

    20     2     3    21
    22     6     7    23
     9    10    11    12
```

相对地，下面的赋值语句更新了数组的全部内容，并改变了数组的形

```
>> arr4=[20 21;22 23]

arr4 =

    20    21
    22    23
```

好的编程习惯

确保将赋值于子数组和赋值于数组。MATLAB 将它们当作两个不同的情况来对待。

2.4.3 用一标量来给子数组赋值

位于赋值语句的右边的标量值总是能匹配左边数组的形。这个标量值将会被复制到左边语句中所对应的元素。例如，假设用一个数组 arr4 定义如下：

```
arr4=[1 2 3 4;5 6 7 8;9 10 11 12]
```

下面的表达式将一个值赋值于数组的 4 个元素。

```
>> arr4(1:2,1:2)=1

arr4 =

     1     1     3     4
     1     1     7     8
     9    10    11    12
```

2.5 特殊变量

在 MATLAB 中有许多预先定义好的特殊变量。在 MATLAB 中这些特殊变量可以随时使用，不用初始化。一些常见的预定义值列在表 2.2。

表 2.2 预定义特殊变量

pi	有 15 个有效值的 π
i,j	代表虚数 i($\sqrt{-1}$)
Inf	这个符号代表无穷大，它一般情况下是除以 0 产生的
NaN	这个符号代表没有这个数。它一般由数学运算得到的。例如，0 除以 0。
clock	这个特殊变量包含了当前的年，月，日，时，分，秒，是一个 6 元素行向量
date	当前的日期，使用的的字符形式，如 30-Dec-2007
eps	变量名是 epsilon 的简写。它代表计算能机辨别的两数之间的最小数
ans	常用于存储表达式的结果，如果一个结果没有明确的赋值给某个变量

这个些预定义值存储在一般的变量中，所以他们能被覆盖或改写。如果一个新值赋值于其中一个预定义变量，，那么这以后的计算中新值将会替代默认值。例如，考虑下面用于计算以半径为 10 的圆的周长的语句；

```
circl=2*pi*10
pi=3
circ2=2*pi*10
```

在第一个语句中，pi 有默认值 3.14159...，所以周长 6.28319 是正确的结果，第二个语句重定义 pi 为 3，所以第三个语句 circ2 为 60。在程序中修改预定义值会造成一些不正确的结果,并导致一些微小而难以发现的错误。设想一下，要在 1000 行的程序中找出一个像这样的错误是多么不容易。

常见编程错误

不要重定义有意义的预定义变量。否则将后患无穷，制造成出小而难以发现的错误。

测试 2.2

本测试提供了一个快速的检查方式，看你是否掌握了 2.3 到 2.5 的基本内容。如果你对本测试有疑问，你可以重读 2.3 和 2.5，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. c 数组的定义如下，写出下面子数组的内容。

```
c =
    1.1000   -3.2000    3.4000    0.6000
    0.6000    1.1000   -0.6000    3.1000
    1.3000    0.6000    5.5000         0
```

(a) c(2,:)　(b) c(:,end)　(c) c(1:2,2:end)　(d) c(6)
(e) c(4:end)　(f) c(1:2,2:4)　(g) c([1 4],2)　(h) c([2 2],[3 3])

2.当赋值语句执行后，下列数组的内容是多少？

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

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

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

3.当数组执行后，下列数组 a 的内容是多少？

(a)
```
a=eye(3,3);
b=[1 2 3];
a(2,:)=b;
```

(b) a=eye(3,3);
b=[4 5 6];
a(:,3)=b';
(c) a=eye(3,3);
b=[7 8 9];
a(3,:)=b([3 1 2]);
(d) a=eye(3,3);
b=[7 8 9];
a(3,:)=b([3 1 2]);

2.6 显示输出数据

在 MATLAB 中有许多的方法显示输出数据。最简单的方法是我们已经用过的去掉语句末的分号，它将显示在命令窗口(The Command Windows)中。在这里向大家介绍一些其他的方法。

2.6.1 改变默认格式

当数据重复在命令窗口(The Command Windows)时，整数以整形形式显示，其他值将以默认格式显示。MATLAB 的默认格式是精确到小数点后四位。如果一个数太大或太小，那么将会以科学记数法的形式显示。比如，语句

```
x=100.11
y=1001.1
z=0.00010011
```

它的输入格式为

```
x =
   100.1100

y =
   1.0011e+003

z =
   1.0011e-004
```

改变默认输出格式要用到 format 命令，可根据表 2.3 改变数据的输出格式

表 2.3 输出显示格式

format 命令	结果	例子
format short	保留小数点后 4 位（默认格式）	12.3457
format long	保留小数点后 14 位	12.345678901234567
format short e	带有 5 位有效数字科学记数法	1.2346e+00
format short g	总共有 5 个数字，可以用科学记数法，也可不用	12.346
format long e	带有 15 位有效数字科学记数法	1.234567890123457e+001
format long g	总共有 5 个数字，可以用科学记数法，也可不用	12.3456789012346
format bank	美元格式	12.35
format hex	用 16 进制表示	4028b0fcd32f707a
format rat	两个小整数的比	1000/81
format compact	隐藏多余的换行符	
format loose	使用多余的换行符	
format +	只显示这个数的正负	+

所有例子都以 12.345678901234567 为例子默认的格式可以改变格式以显示更多的有效数字，用科学计数法来显示，精确到小数点后两位，显示或隐藏多余的换行符。

2.6.2 disp 函数

另一种显示数据的方法是用 disp 函数。disp 需要一个数组参数，它将值将显示在命令窗口(The Command Windows)中。如果这个数组是字符型（char），那么包含在这个数组中的字符串将会打印在命令窗口(The Command Windows)中。

此函数可联合 num2str(将一个数转化为字符串)和 int2str(将一个整数转化为字符串)来产生新的信息，显示在命令窗口(The Command Windows)中。例如，下面的语句将“the value of pi=3.1416”显示在命令窗口(The Command Windows)中。第一句创建了一个字符型数组,第二句用于显示这个数组。

```
str=['the value of pi='     num2str(pi)];
disp(str);
```

2.6.3 用 fprintf 函数格式化输出数据

用 fprintf 函数显示数据是一种十分简便方法。fprintf 函数显示带有相关文本的一个或多个值，允许程序员控制显示数据的方式。它在命令窗口打印一个数据的一般格式如下：

fprint(format,data)

其中 format 用于代表一个描述打印数据方式的子符串，data 代表要打印的一个或多个标量或数组。format 包括两方面的内容，一方面是打印内容的文本的提示；另一方面是打印的格式。例如，函数

```
fprintf('The value of pi is %6.2f \n',pi)
```

将会打印出'The value of pi is 3.14',后面带有一个换行符。转义序列%6.2 代表在本函数中的第一个数据项将占有 6 个字符宽度，小数点后有 2 位小数。

fprintf 函数有一个重大的局限性，只能显示复数的实部。当我们的计算结果是复数时，这个局限性将会产生错误。在这种情况下，最好用 disp 显示数据。

表 2.4 fprintf 函数 format 字符中的特殊字符

format string	结果
%d	把值作为整数来处理
%e	用科学记数法来显示数据
%f	用于格式化浮点数，并显示这个数
%g	用科学记数格式，或浮点数格式，根据那个短，并显示之
\n	转到新的一行

例如，下列语句计算复数 x 的值，分别用 fprintf 和 disp 显示

```
x=2*(1-2*i)^3;
str=['disp: x = ' num2str(x)];
disp(str);
fprintf('fprintf: x = %8.4f\n',x);
```

打印的结果如下

```
disp: x = -22+4i
fprintf: x = -22.0000
```

注意 frpintf 忽略了虚部。

常见编程错误

fprintf 函数只能复数的实部，所以在有复数参加或产生的计算中，可能产生错误的结果。

2.7 数据文件

有许多的方法用于加载和保存 MATLAB 的数据文件，这些方法将会在第八章中介绍。在这里我们只向大家介绍最简章的 save 和 load 命令。

save 命令用于保存当前 MATLAB 工作区内的数据到一个硬盘文件。这个命令的基形式如下

save filename var1 var2 var3

filename 代表你要保存变量的那个文件，var1,var2 等是要保存的变量。在默认情况下，这个这个文件的扩展名为‘mat’,我们称之为 MAT 文件。如果在 filename 后面无变量，则工作区的所有内容将会被保存。

MATLAB 用一种特殊的复杂形式来存储数据，包括了许许多多的细节，例如变量名和变量类型，数组的大小，以及所有变量值。一个在任何一个平台上创建的 MAT 文件(pc, mac, unix)在另一个平台上都可以应用。它的缺点是 MAT 文件的存储格式不能被其他程序读取。如果一个数据必须由其他程序所读取，那么必须转化为 ASCII 码，并将这些数值写到一个以 ASCII 码为编码的文件中。但是，当以 ASCII 的形式存储，像变量名和变量类型这样的信息就会丢失，产生的数据结果将会更大。

例如，假设数组 x 的定义如下

```
x=[1.23 3.14 6.28; -5.1 7.00 0];
```

命令“save x.dat x -ascii”将会创建一个文件 x.dat，包括数据如下

```
1.2300000e+000   3.1400000e+000   6.2800000e+000
-5.1000000e+000   7.0000000e+000   0.0000000e+000
```

用这种格式定的数据能被其他语言编写的程序或扩展页读取，所以它能帮助 MATLAB 程序和其他程序之间共享数据。

好的编程习惯

如果数据需要在 matlab 和其他程序之间交换使用，那么以 ASCII 格式存储数据。如果只在 matlab 中使用那么，应以 mat 文件的形式存储数据。

MATLAB 并不关心 ASCII 码的扩展名是什么？但是，用户最好用它的传统扩展名“dat”。

好的编程习惯

以“dat”的扩展名保存 ASCII 数据文件，以区别于以“mat”为扩展名的 mat 文件。

Load 命令与 save 命令相反。它从硬盘文件加载数据到 MATLAB 当前工作区。这个命令的基本格式为

load filename

filename 代表所加载文件的文件名。如果这个文件是 mat 文件，那么所有被子加载的变量的变量名的变量类型将和原来一样。如果一个变量包含在工作区间窗口，那么这些数据将会被修复。

MATLAB 能够加载由其他程序创建的 ascii 格式的数据文件。它首先检查所要加载的文件是 mat 文件还是 ascii 文件。如果在 load 语句中加入- ascii 中，则强制 MATLAB 把这个文件看作 ASCII 文件。这个文件的内容将会被转化为一个 MATLAB 的数组，这个数组名就所要加载的文件名。例如，假设一个名为 x.dat 的 ascii 文件包括下列数据：

```
1.23 3.14 6.28
-5.1  7.00 0
```

那么“load x.dat”将会在当前工作区创建一个 2×3 数组 x,包含数据值。

测试 2.3

本测试提供了一个快速的检查方式，看你是否掌握了 2.6 和 2.7 的基本内容。如果你对本测试有疑问，你可以重读 2.6 和 2.7，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.如何让 MATLAB 显示一个实数，带有十五个有效的数字，并有指数形式？

2.下列语句的作用是什么？它的输出是什么？

(a)
```
radius = input('Enter circle radius:\n');
area = pi * radius^2;
str = ['The area is ' num2str(area)];
disp(str);
```
(b)
```
value = int2str(pi);
disp(['The value is ' value '!']);
```
3.下列语句有作用是什么？它的输出是什么？
```
value = 123.4567e2;
fprintf('value = %e\n',value);
fprintf('value = %f\n',value);
fprintf('value = %g\n',value);
fprintf('value = %12.4f\n',value);
```

2.8 标量运算和数组运算

在 MATLAB 赋值语句中的计算，它的一般形式如下

variable_name = expression;

赋值语句计算出等号右边表达式的值，然后赋值于等号左边的变量名。注意这个等号并不是传统意义上的等号，它的意义是：存储表达式的值到左边的变量，由于这个原因，等号在这里应叫做赋值号。像

ii = ii + 1;

这样的语句在数学上是毫无意义的，但在 MATLAB 语言中，它有其固有的意义。

它的意义是：把变量 ii 加上 1 之后，再把值存储到变量 ii 中。

2.8.1 标量运算符

位于赋值号右边的表达式，可以包含标量，数组，括号和数学符号的任一个有效联合运算。两标量间的标准运算符号如表 2.5 所示。

当我们需要的时侯，我们可以运用括号来控制运算顺序。括号内的表达式优先于括号外的表达式来计算。例如表达式 2^((8+2)/5)的计算顺序如下

```
2 ^ ( ( 8 + 2 ) / 5 )  = 2 ^ ( 10 / 5 )
                       = 2 ^ 2
                       = 4
```

2.8.2 数组运算和矩阵运算

MATLAB 在数组运算中提供了两种不同类型的运算,一种是数组运算(array operations),一种是矩阵运算(matrix).数组运算是一种用于元素对元素的运算.也就是说,这个运算是针对两数组相对应的运算使用的.例如,$a=\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix}$ $b=\begin{bmatrix}-1 & 3\\ -2 & 1\end{bmatrix}$,那么 $a+b=\begin{bmatrix}0 & 6\\ 0 & 5\end{bmatrix}$.注意两数组的行与列必须相同.否则,MATLAB 将产生错误.

数组运算可以用于数组与标量的运算.当一个数组和一个标量进行运算时,标量将会和数组中的每一元素进行运算.例如

$a=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$,则 $a+4=\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}$

相对地,矩阵运算则遵守线性代数的一般规则,像矩阵的乘法.在线性代数中,c=a×b 的定义如下:

表 2.5 两标量间的数学运算符

运算符	代数形式	MATLAB 形式
加号	A+B	A+B

运算符	代数形式	MATLAB 形式
减号	A-B	A-B
乘号	A×B	A*B
除号	$\frac{A}{B}$	A/B
指数	A^B	A^B

$$c(i,j)=\sum_{k=1}^{n}a(i,k)b(k,j)$$

例如 $a=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}, a=\begin{bmatrix}-1 & 3\\ -2 & 1\end{bmatrix}$,那么 $a\times b=\begin{bmatrix}-7 & -6\\ -10 & 10\end{bmatrix}$。注意，在矩阵相乘中，a 阵的列数必须等于 b 阵的行数。

MATLAB 用一个特殊的符号来区分矩阵运算和数组运算。在需要区分两者不同的时侯，把点置于符号前来指示这是一个数组运算（例如，.*）。表 2。6 给出的是一些常见的数组和矩阵运算。

表 2.6 常见的数组和矩阵运算

运算	MATLAB 形式	注释
数组加法	A+B	数组加法和矩阵加法相同
数组减法	A-B	数组减法和矩阵减法相同
数组乘法	A.*B	A 和 B 的元素逐个对应相乘.两数组之间必须有相同的形,或其中一个是标量.
矩阵乘法	A*B	A 和 B 的矩阵乘法.A 的列数必须和 B 的行数相同.
数组右除法	A./B	A 和 B 的元素逐个对应相除: A(i,j)/B(i,j)两数组之间必须有相同的形,或其中一个是标量.
数组左除法	A.\B	A 和 B 的元素逐个对应相除: B(i,j)/A(i,j)两数组之间必须有相同的形,或其中一个是标量.
矩阵右除法	A/B	矩阵除法,等价于 A*inv(B), inv(B)是 B 的逆阵
矩阵左除法	A\B	矩阵除法,等价于 inv(B)*A, inv(A)是 A 的逆阵
数组指数运算	A.^B	AB 中的元素逐个进行如下运算 A(i,j)^B(i,j), A(i,j)/B(i,j)两数组之间必须有相同的形,或其中一个是标量.

初学者往往混淆数组运算和矩阵运算.在一些情况下,两者相互替换会导致非法操作,MATLAB 将会报告产生了错误。在另一些情况下，两种运算都是合法的，那么这时 MATLAB 进行错误的运算，并产生错误的结果。当我们进行方阵运算时，极易产生这样的错误。两个方阵具有相同的大小，两者之间的数组运算和矩阵运算都是合法的，但产生的结果完全不同。在这种情况下，你要万分的小心。

常见编程错误

在你的 MATLAB 代码中，仔细区分数组运算和矩阵运算。数组乘法和矩阵乘法极易混淆。

例 2.1

假设 a,b,c 和 d 的定义如下

$a=\begin{bmatrix}1 & 0\\ 2 & 1\end{bmatrix}$　　$b=\begin{bmatrix}-1 & 2\\ 0 & 1\end{bmatrix}$　　$c=\begin{bmatrix}3\\ 2\end{bmatrix}$　　$d=5$

分别指出下列表达式的运算结果

(a) a + b　　(b) a .* c　　(c) a * b　　(d) a * c

(e) a + c　　(f) a + d　　(g) a .* d　　(h) a * d

答案：

（a）　这是一个数组或矩阵加法：$a+b=\begin{bmatrix}0 & 2\\ 2 & 2\end{bmatrix}$。

（b）　这是一个数组乘法：$a.*b=\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$。

（c）　这是一个矩阵乘法：$a*b=\begin{bmatrix}-1 & 2\\ -2 & 5\end{bmatrix}$。

（d）　这是一个矩阵乘法：$a*c=\begin{bmatrix}3\\ 8\end{bmatrix}$。

（e）　操作非法，两数组形不同

（f）　数组与标量的加法：$a+d=\begin{bmatrix}6 & 5\\ 7 & 6\end{bmatrix}$。

（g）　数组乘法：$a.*d=\begin{bmatrix}5 & 0\\ 10 & 5\end{bmatrix}$。

（h）　矩阵乘法：$a*d=\begin{bmatrix}5 & 0\\ 10 & 5\end{bmatrix}$。

矩阵的左除运算有着十分重要意义，我们必须理解它。一个 3×3 的线性方程组的形式如下

$$\begin{aligned}a_{11}x_1+a_{12}x_2+a_{13}x_3&=b_1\\ a_{21}x_1+a_{22}x_2+a_{23}x_3&=b_2\\ a_{31}x_1+a_{32}x_2+a_{33}x_3&=b_3\end{aligned}\qquad(2.1)$$

可以写成如下形式

$$Ax=B\qquad(2.2)$$

$A=\begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}$，$A=\begin{bmatrix}b_1\\ b_2\\ b_3\end{bmatrix}$ 和 $x=\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}$

2.2 式的结果可以用线性代数的知识来解决。结果是

$$x=A^{-1}B\qquad(2.3)$$

因为左除运算 A\B 等价于 inv(A)*B，所以左除运算是解线性方程的好方法。

2.9 运算的优先级

许多的数学运算写入一个表达式是非常平常的事。例如，考虑初速度为 0 的匀加速运动的位移表达式

distance = 0.5 * accel * time ^ 2

这个表达式有二个乘法运算和一个幂运算。在这样的表达式中，知道运算的先后顺序是十分重要的。如果幂运算先于乘法运算执行，这个表达式等价于

distance = 0.5 * accel * (time ^ 2)

如果乘法运算先于幂运算执行，这个表达式等价于

distance = (0.5 * accel * time) ^ 2

这两个式子将产生不同的结果，所以我们必须清楚它们中那个是正确的。

为了使表达的值精确，MATLAB 建立了一系列的规则控制运算的层次或顺序。这些规则一般情况下遵循代数的运算法则。数学运算的顺序如表 2.7。

表 2.7 运算的优先级

优先级	运算
1	括号里的内容先运算，从最里面的括号去运算
2	幂运算，从左向右
3	乘除法，从左向右
4	加减法，从左向右

例 2.2

变量 a,b,c,d 初始化如下

a = 3; b = 2; c = 5; d = 3;

计算如下的 MATLAB 的赋值语句

(a) output = a*b+c*d;
(b) output = a*(b+c)*d;
(c) output = (a*b)+(c*d);
(d) output = a^b^d;
(e) output = a^(b^d);

正如我们看到的，运算的顺序对一个代数表达式的最终值产生重大的影响。

将程序中的每个表达式尽量写清楚，这是十分重要的。编写的程序不仅要能够计算出所要求的值的来，在需要的时侯，还要考虑它的可维护性。你应当经常问自己“六个月后我能看得懂我现在编得程序吗？其他的程序员看到我的代码，他能迅速的理解吗？”。如果在你的心中有所疑虑，那就用更多的括号使之更加清晰。

好的编程习惯

在需要的时侯用括号使用表达式更加清晰和易于理解。

如果在一个表达式中用到括号，那么括号必须平衡。也就是说，左括号数与右括号数相等。如果两者数目不相同，那么将会导致错误的产生。这种错误经常在输入过程中发生，当 MATLAB 编译器在执行这个命令时被发现。例如

（2+4) /2)

在执行时将会出现一个错误。

测试 2.4

本测试提供了一个快速的检查方式，看你是否掌握了 2.8 和 2.9 的基本内容。如果你对本测试有疑问，你可以重读 2.8 和 2.9，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.假设 abcd 的定义如下，计算下面合法运算的结果，如果不合法，指出原因

$$a=\begin{bmatrix}2 & 1\\ -1 & 2\end{bmatrix} \qquad b=\begin{bmatrix}0 & -1\\ 3 & 1\end{bmatrix} \qquad c=\begin{bmatrix}1\\ 2\end{bmatrix} \qquad d=-3$$

(a) result = a .* c;
(b) result = a * [c c];
(b) result = a .* [c c];
(d) result = a + b * c;
(e) result = a + b .* c;

2.求矩阵 x，已知 Ax=B,

$$A=\begin{bmatrix}1 & 2 & 1\\ 2 & 3 & 2\\ -1 & 0 & 1\end{bmatrix} \qquad B=\begin{bmatrix}1\\ 1\\ 0\end{bmatrix}$$

2.10 MATLAB 的内建函数

2.10.1 选择性结果

与数学的函数不同，MATLAB 函数返回一个或多个值给调用函数。max 函数就是这样的一个例子。这个函数一般情况下返回输入向量中的最大值，但是它返回的第二个参数是输入向量中的最大值在向量中的位置。例如，语句

```
maxval = max ([1 -5 6 -3])
```

返回的结果为 maxval=6,但是要有两个返回值，那么这个函数包括最大所处的位置。

```
[maxval index] = max ([1 -5 6 -3])
```

将会产生结果 maxval=6,和 index=3.

2.10.2 带数组输入的 MATLAB 函数的应用

许多 MATLAB 函数定义了一个或多个标量输入，产生一个输出。例如，语句 y=sin(x) 计算了 x 的正弦，并将结果存储到 y 变量中。如果这些函数接受了输入值构成的数组，那么 MATLAB 将一一计算出每个元素所对应的值。例子，假设

```
x=[0 pi/2 3*pi/2 2*pi]
```

那么语句

```
y=sin(x)
```

将会产生 y=[0 1 -1 -0].

2.10.3 常见的 MATLAB 函数

一些极其常用的 MATLAB 函数列入了表 2.8 中.这些函数将会用在以后的例子的作业中.如果你要加载不在下表中的函数,那么你通过前面介绍的方法,搜索适当的函数.

注意与大多数的计算语言不同,许多的 MATLAB 函数能够正确计算出复数结果.matlab 自动计算出正确的结果,尽管其结果可能是虚数和复数.例如,在 C 和 Fortan 语言中运行函数 sqrt(-2)时将会出现运行时错误.相反地,MATLAB 将会产生虚部答案.

```
>> sqrt(-2)
ans =
          0 + 1.4142i
```

2.11 画图入门

MATLAB 的扩展性和机制独立的画图功能是一个极其重要的功能.这个功能使数据画图变得十分简单.画一个数据图,首先要创建两个向量，由 x, y 构成,然后使用 plot 函数.

例如,假设我们要画出函数 $y=x^2-10x+10$ 的图象,定义域为［0,10］.只需要 3 个语句就可以画出此图.第二句用于计算 y 值(注意我们用的是数组运算符,所以可以对 x 的元素一一运算.).最后打印出此图.

```
x=0:1:10;
y=x.^2-10*x+15;
plot(x,y);
```

当执行到 plot 函时,MATLAB 调用图象窗口,并显示图象.如图图 2.4.

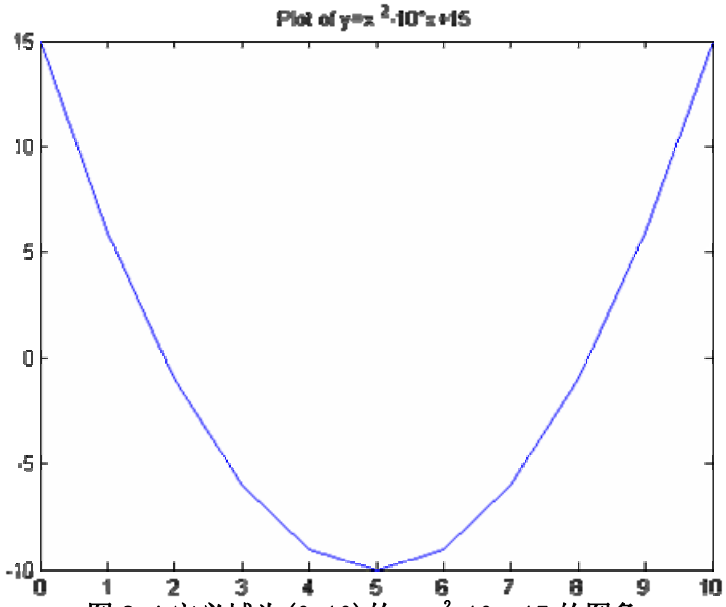


图 2.4 定义域为(0,10)的 $y=x^2-10x+15$ 的图象.

表 2.8 常见的 MATLAB 函数

函数	描述
数学函数	
abs(x)	计算 x 的绝对值
acos(x)	计算 x 的反余弦函数
angle(x)	计算复数 x 的幅角
asin(x)	计算 x 的反正弦函数值
atan(x)	计算 x 的反正切函数值
atan2(y,x)	$\tan^{-2}(y/x)$
cos(x)	cosx
exp(x)	e^x
log(x)	$\log_e x$
[value,index]=max(x)	返回 x 中的最大值，和它所处的位置
[value,index]=min(x)	返回 x 中的最小值，和它所处的位置
mod(x,y)	余数
sin(x)	sinx
sqrt(x)	x 的平方根
tan(x)	tanx
rounding(取整)函数	
ceil(x)	
fix(x)	
round(x)	
字符转换函数	
char(x)	将矩阵中的数转化为字符,矩阵中的元素就不大于 127
double(x)	将子符串转化为矩阵
int2str(x)	将整数 x 转化为字符串形式
num2str(x)	将带小数点的数转化为一个字符型数组
str2num(x)	将字符串转化为数

2.11.1 简单的 xy 画图的应用

正如我们所看到的,在 MATLAB 中画图是十分容易的.只要任何一对向量的长度相同,那

么它就可以就能可视化地画出来.但是这还不是最后的结果,因为它还没有标题,坐标轴标签,网格线.

给图增加标题和坐标轴标签将会用到 title, xlabel, ylable 函数。调用每个函数时将会有一个字符串，这个字符串包含了图象标题和坐标轴标签的信息。用 grid 命令可使网格线出现或消失在图象中，grid on 代表在图象中出现网格线，grid off 代表去除网格线。例如下面的语句将会产生带有标题，标签和网格线的函数图象。结果如图 2.5 所示。

```
x=0:1:10;
y=x.^2-10*x+15;
plot(x,y);
title ('Plot of y=x.^2-10*x+15');
xlabel ('x');
ylabel ('y');
grid on;
```

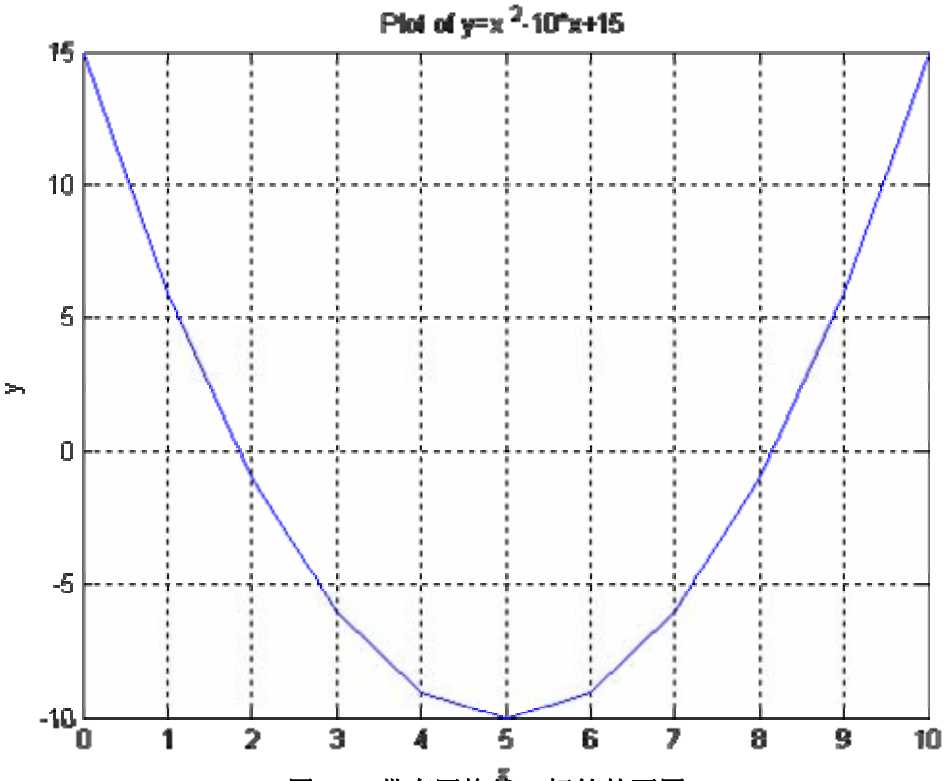


图 2.5 带有网格线，标签的画图

2.11.2 打印图象

一个图象一旦建立，我们就可以用 print 命令在打印机上打印出这幅图，也可以单击图象窗口的打印图标或者在文件菜单中选择打印项打印。

print 命令的一般形式如下：

print <选项> <文件名>

如果没有文件名，这个命令就会命令打印机打印当前图片。如果带有文件名，那么这个命令就会打印这个图片到指定的文件。有许多的选项指定输出到文件或打印机的格式。一个最重要的选项是-dtiff.这个选项指定输出图片的格式是标签影像档案格式（TIFF）。因为在 PC，Mac 和 UNIX 平台上的文字处理软件都支持这种格式。这就使得在文档中插入 MATLAB 图象变得十分的简单。下面这个命令将会创建一个 TIFF 格式的当前图象的图片，并保存在一个叫 my_image.tif 的文件中

print –dtiff my_image.tif

你也可以选择图象窗口中的“file/export”选项来创建 tiff 图片。

2.11.3 联合作图

在同一坐标内作出多个函数的图象的情况是十分常见的。假如，你要在同一坐标轴内作出 f(x)=sin2x 和他的微分函数的图象。它的微分式为

$$\frac{d}{dt}\sin 2x = 2\cos 2x \qquad (2.4)$$

在同一坐标系内打印两个函数，我们必须产生一系列的 x 值和每一个函数分别对应的 y 值。然后利用这些值画出图象，plot 函数的格式如下所示：

```
x=0:pi/100:2*pi;
y1=sin(2*x);
y2=2*cos(2*x);
plot (x,y1,x,y2);
```

所得图像如图图 2.6 所示。

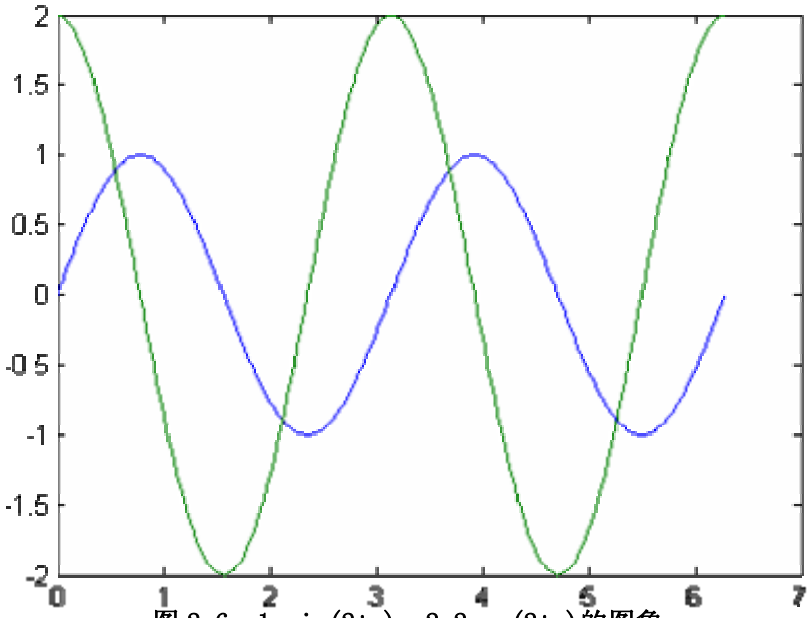


图 2.6 y1=sin(2*x) y2=2cos(2*x)的图象。

2.11.4 线的颜色,线的形式,符号形式和图例

MATLAB 允许程序员选择轨迹的颜色,轨迹的形式,和符号的类型.在 X,Y 向量参数后带有这些属性的字符串的 plot 函数,可以选择这些细节.

这些属性字符串包括三个方面,

第一方面指定轨迹的颜色,

第二方面指定符号的类型,

第三方面指定线的类型.

各种颜色,符号和线的类型将在表 2.9 中显示.

表 2.9 图象的颜色，标记（符号）类型，线型

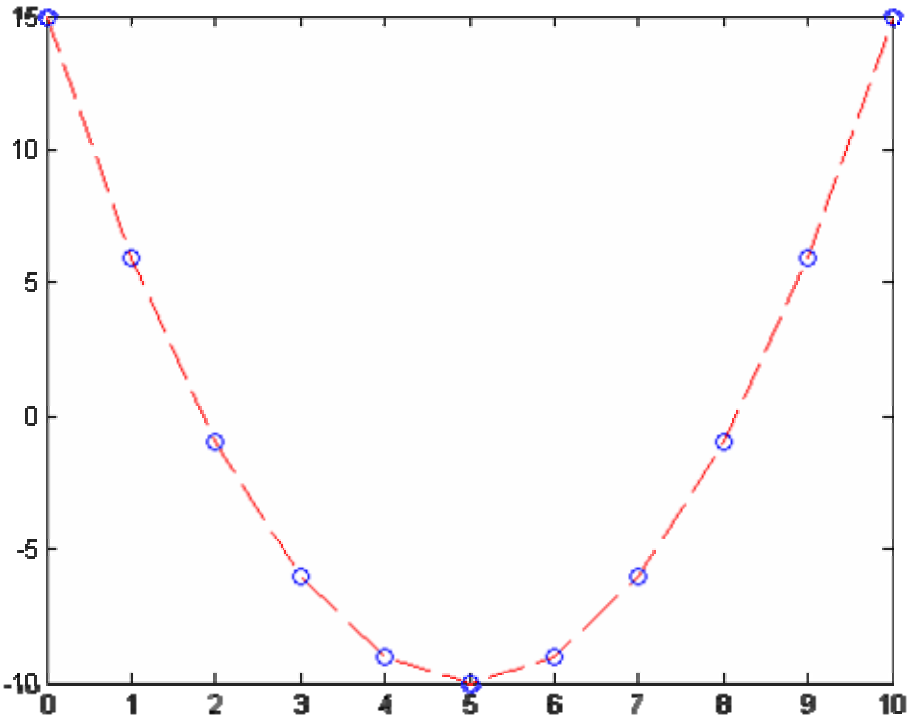
颜色		标记类型		线型	
y	黄色	.	点	-	实线
m	品红色	o	圈	:	点线
c	青绿色	x	×号	-.	画点线
r	红色	s	正方形	--	虚线
g	绿色	d	菱形	<none>	无

颜色		标记类型		线型	
b	蓝色	v	倒三角		
w	白色	^	正三角		
k	黑色	>	三角(向右)		
		<	三角(向左)		
		p	五角星		
		h	六线形		
		<none>	无		

这些属性字符串可以任意的混合使用.如果有多个函数,每个函数都有它自己的属性字符串.

例如,函数 $y=x^2-10x+15$ 的图象,曲线为红色的虚线,重要的数值用蓝色的小圆圈表示.

```
x=0:1:10;
y=x.^2 -10.*x +15;
plot(x,y,'r--',x,y,'bo');
```



我们可以用 legend 来制作图例。它的基本的形式如下

legend('string1','string2',...,pos)

其中 string1,string2 等等是与轨迹标签名，而 pos 是一个整数，用来指定图例的位置。这些整数所代表的意义在表 2.10 中的列出。用 legend off 命令将能去除多余的图例。一个完整的图象例子将会显示图 2.7 中，产生这个图象的语句如下所示。图 2.7 在同一坐标系内，显示了 f(x)=sin2x 和它的微分函数的图象，用黑实线代表 f(x),用红虚线代表它的微分函数。图中有标题，坐标轴标签和网格线。

```
x=0:pi/100:2*pi;
y1=sin(2*x);
y2=2*cos(2*x);
plot(x,y1,'k-',x,y2,'b--');
title(' Plot of f(x)=sin(2x) and its derivative');
xlabel('x');
ylabel('y');
```

```
legend('f(x)','d/dx f(x)')
grid on;
```

2.11.5 对数尺度

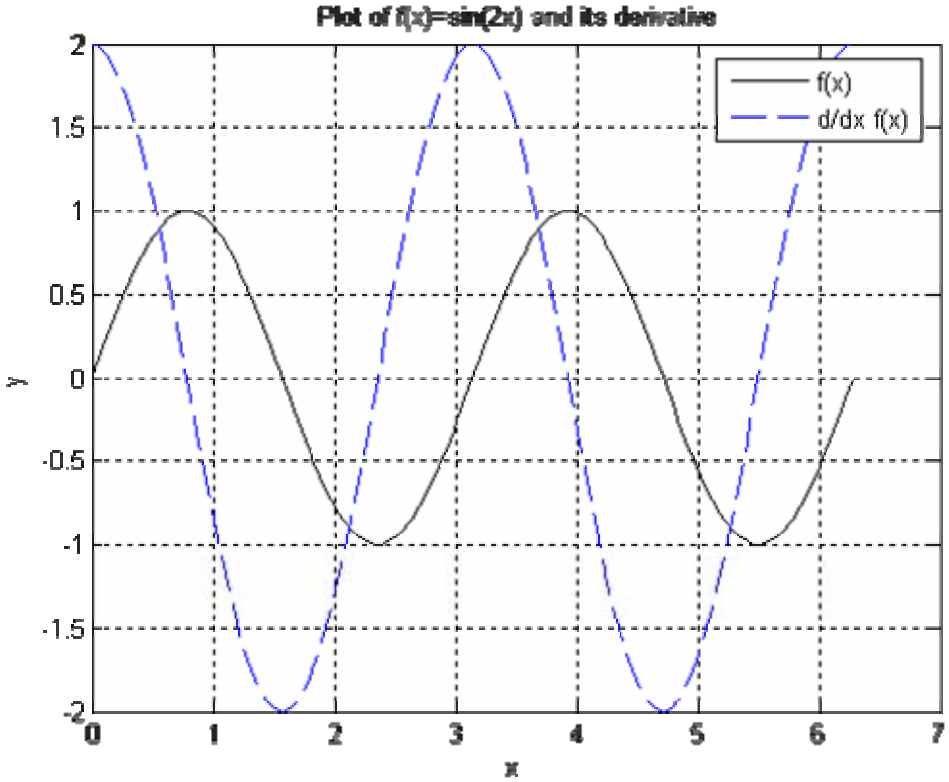


图 2.7 在同一坐标系内，显示了 f(x)=sin2x 和它的微分函数的图象

打印数据既可以用对数尺度，也可以用线性尺度。在 x,y 轴上使用这两种尺度的一种或两种可以组合形成 4 种不同的坐标系。每一种组合者有一个特定的函数。

1.plot 函数的 x,y 均用线性尺度

2.semilog 函数 x 轴用对数尺度，y 轴将用线性尺度

3.semiloge 函数 x 轴用线性尺度，y 轴用对数尺度

4.loglog 函数两坐标轴将会都用对数尺度。

这四个函数在意义上是等价的，只是坐标轴的类型不同。每一个图象的例子如图 2.8 所示。

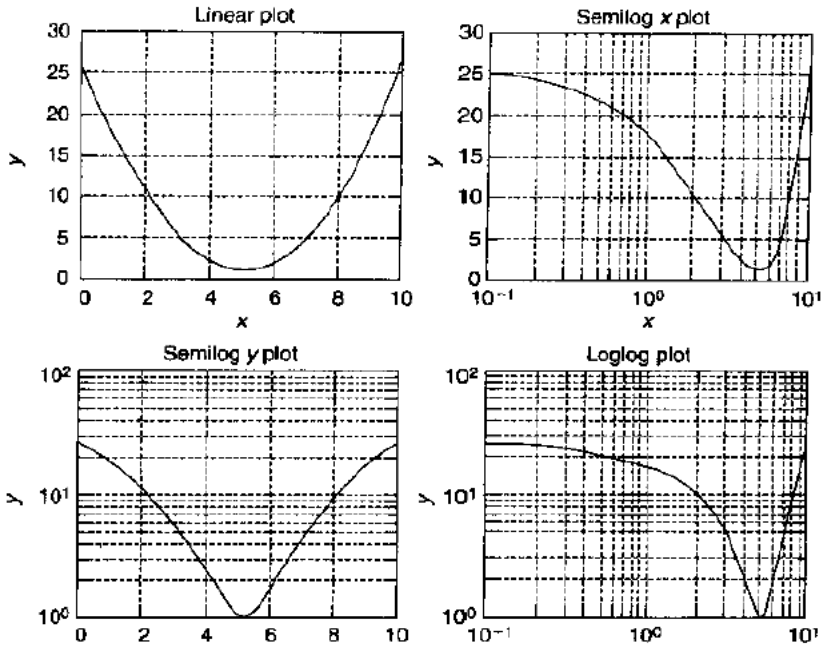


图 2.8 四种不同画图函数的对比

表 2.10 在 legend 命令中 pos 的值

值	意义
0	自动寻找最佳位置，至少不与数据冲突
1	在图象的右上角
2	在图象的左上角
3	在图象的左下角
4	在图象的右下角
-1	在图象的右边

2.12 例子

下面的例子将向大家介绍如何用 MATLAB 解决问题。

例 2.3

（温度转换）设计一个 MATLAB 程序，读取一个华氏温度的输入，输出开尔文温度。

答案

华氏温度和开尔文温度的转换关系式可在物理学课本中找到。其关系式为：

$$T(\text{开尔文}) = \left(\frac{5}{9}T(\text{摄氏度}) - 32.0\right) + 273.15 \qquad (2.5)$$

在物理学参考书中举了一些例子，我们可以用来检验我们程序是否正确。例如

	华氏度（°C）	开尔文(K)
沸水的温度	212	373.15
冰水混合物的温度	-110	194.26

我们设计程序的步骤如下

1.提示用户键入华氏温度值

2.读取输入值

3.通过关系式转换为开氏温度

4.输出结果，结束

我们将会用 input 函数输入华氏温度，用 fprintf 函数输出结果。

```
%    Script file:temp_conversion.m
```

```
%
%    Purpose:
%    To convert an input temperature from degrees Fahrenheit to
%    an output temperature in kelvins.
%
%    Record of revisions:
%    Date       Programmer    Description of change
%    ====       ==========    =====================
%    12/01/97   S.J.Chapman   Original code
%
%Define variables:
%    temp_f    --Temperature in degrees Fahrenheit
%    temp_k    --Temperature in kelvins

%Prompt the user for the input temperature.
temp_f=input('Enter the temperature in degrees Fahrenheit:');
%Converttokelvins.
temp_k=(5/9)*(temp_f-32)+273.15;
%Writeouttheresult.
fprintf('%6.2f degrees Fahrenheit = %6.2f kelvins.\n',...
temp_f,temp_k);
```

我们输入上面的例子中的华氏温度值，以检测程序的正确性。注意用户的输入值已用黑体字标出。

```
>> temp_conversion
Enter the temperature in degrees Fahrenheit:212
212.00 degrees Fahrenheit = 373.15 kelvins.
>> temp_conversion
Enter the temperature in degrees Fahrenheit:-110
-110.00 degrees Fahrenheit = 194.26 kelvins.
```

这个结果和物理教科书的结果相同。在本程序中,我们重复出带单位的输入值和输出值.只有带上单们神经质输出才有意义.

按照惯例,任何输入变量和输出变量的单位都应打印出来.

好的编程习惯

当你读取和写入数据时,使用适当的单位

例 2.4

电子工程：负载的最大输出功率

一个内阻 R_s＝50Ω，电动势 V＝120V 的电源驱动一个负载 R_L。当 R_L 为多少时，R_L 的功率最大？在这种情况下，功率为多少？画以 R_L 为自变量的 R_L 功率图。

答案：

在本程序中，我们需要改变 R_L 的值，然后计算出每一个 R_L 的功率。R_L 功率的表达式为

$$P_L = I^2 R_L \qquad (2.6)$$

I 代表流经负载的电流。电流可由欧姆定律计算得到。

$$I = \frac{V}{R_{总}} = \frac{V}{R_s + R_L} \qquad (2.7)$$

这个问题解决的步骤如下

1. 创建一个数组。这个数组是以 1 为起始项，以 1 步长的等差数组，共 100 项，这是 R_L 的取值。
2. 计算 R_L 的电流，
3. 计算每个 R_L 的功率。

4. 画出 R_L 的功率图，以确定 R_L 为多少时其功率最大。

整个程序的代码如下：

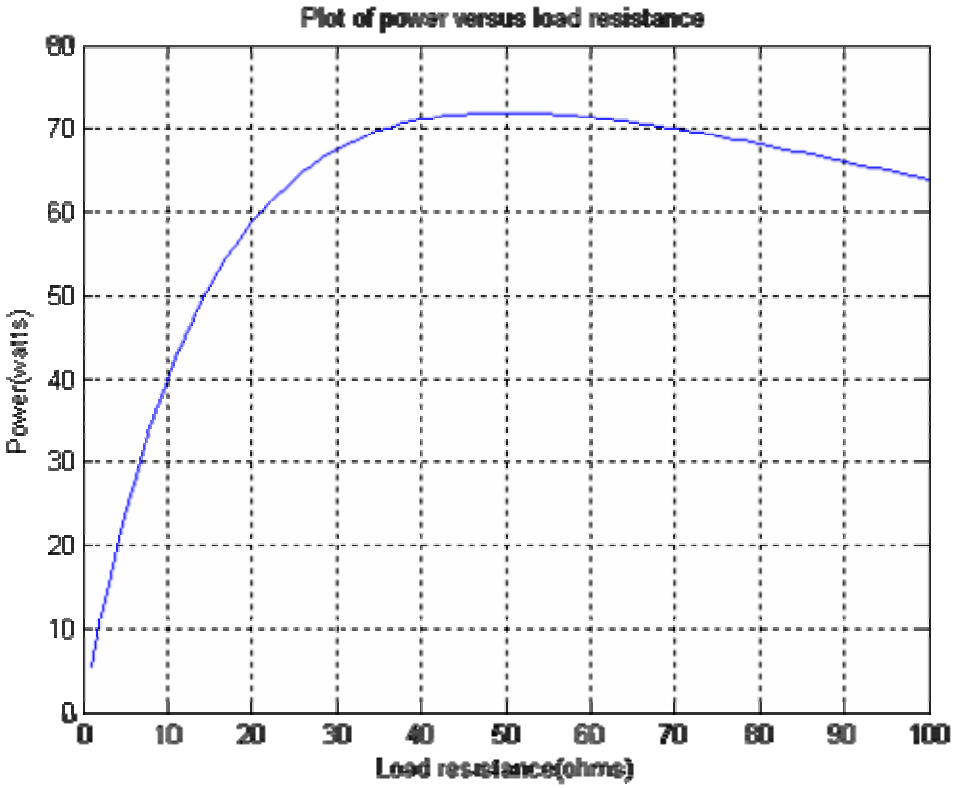


图 2.10 供给负载电阻的功率图象

```
%    Script file:calc_power.m
%
%    Purpose:
%    To calculate and plot the power supplied to a load as
%    a function of the load resistance.
%
%    Record of revisions:
%    Date  Programmer    Descriptionofchange
%    ====        ==========    ====================
%    12/01/98 S.J.Chapman Original code
%
%Define variables:
%    amps       --Current flow to load(amps)
%    pl     --Power supplied to load(watts)
%    rl     --Resistance of the load(ohms)
%    rs     --Internal resistance of the power source(ohms)
%    volts--Voltage of the power source(volts)
%Set the values of source voltage and internal resistance
volts=120;
rs=50;
%Create an array of load resistances
rl=1:1:100;
%Calculate the current flow for each resistance
amps=volts./(rs+rl);
%Calculate the power supplied to the load
pl=(amps.^2).*rl;
%Plot the power versus load resistance
plot(rl,pl);
```

```
title('Plot of power versus load resistance');
xlabel('Load resistance(ohms)');
ylabel('Power(watts)');
grid on;
```

当这个程序运行时，产生的图象如图 2.10。从这个图我们可知当负载电阻为 50Ω时，功率最大。最大功率为 72W。

注意在本例中，用的是数组运算符.*,.^和./.这些运算符将会使数组 amps 和 pl 按元素一一对应计算.

例 2.5

用 C-14 确定年代

一个元素的放射性同位素是不稳定元素的一种特殊形态.在一段时间内,它会自然的衰变为另一种元素.衰变一种呈指数下降的过程.如果 Q_0 是放射性物质在 t=0 时的初始量,那么它的质量与变量 t 的关系式为

$$Q(t) = Q_0 e^{-\lambda t} \qquad (2.8)$$

其中 λ 代表衰变率.

因为放射性元素的衰变是以一定的速率发生的，我们可以把它当作一个时钟来测定的衰变开始的时间。如果我们知道衰变开始时物质的质量和现在放射性元素剩余的质量,我们可以根据公式(2.8)换算出衰变时间 t,即

$$t_{\text{decay}} = -\frac{1}{\lambda}\log_e \frac{Q}{Q_0} \qquad (2.9)$$

公式 2.9 在科学的许多领域有着广泛的应用.例如,考古学家可以根据 C14 的衰变周期,来确定古生物距今生活的年代.现在活着的生物 C14 的含量是不变的,所以可以根据古生物 C14 的现存量来确定古生物的生存年代.已知 C14 的衰变率 λ 为 0.00012097/年,所以如果 C14 的剩余量可以通过测量得到,那么我们就可以根据公式 2.9 算出这个生物活在多少年之前.图 2.1 向大家展示了以时间为自变量的 C14 的剩余量函数.

编定一个程序,读取样品中 C14 剩余量的百分比,计算样品距今的年代,并打印出结果.

这个问题解决的步骤如下

1. 提示用户输入样品中 C14 的剩余量
2. 读取百分比
3. 将百分比转化成分数 $\frac{Q}{Q_0}$.
4. 利用公式(2.9)计算出距今的年数
5. 输出结果,结束.

代码如下

```
%    Script file:c14_date.m
%
%    Purpose:
%    To calculate the age of an organic sample from the percentage
%    of the original carbon 14 remaining in the sample.
%
%    Record of revisions:
%    Date Programmer    Description of change
%    ====      ========== ====================
%    12/02/97  S.J.Chapman   Original code
%
%Define variables:
%     age              --The age of the sample in years
%     lamda            --the radioactive decay constant for carbon-14,in units of 1/years.
```

```
%     percent          --The age of carbon 14 remaining at the time of the measurement
%     ratio            --The ratio of the carbon 14 remaining at the time of the measurement to
the original amount of carbon 14.
%Set decay constant for carbon-14
lamda=0.00012097;
%Prompt the user for the percentage of C-14 remaining.
percent=input('Enter the percentage of carbon 14 remaining:\n');
%Perform calculations
ratio=percent/100;    %Convert to fractional ratio
age=(-1.0/lamda)*log(ratio);%Get age in years
%Tell the user about the age of the sample.
string=['The age of the sample is ' num2str(age) 'years.'];
disp(string);
```

我们通过计算 C14 的半周期来来测试这个程序.

测试结果如下

```
>> c14_date
Enter the percentage of carbon 14 remaining:
50
The age of the sample is 5729.9097years.
```

在化学物理 CRC 手册(The CRC Handbook of Chemistry and Physics)中,C14 的半衰期为 5730 年.我们计算的结果和参考书相符.

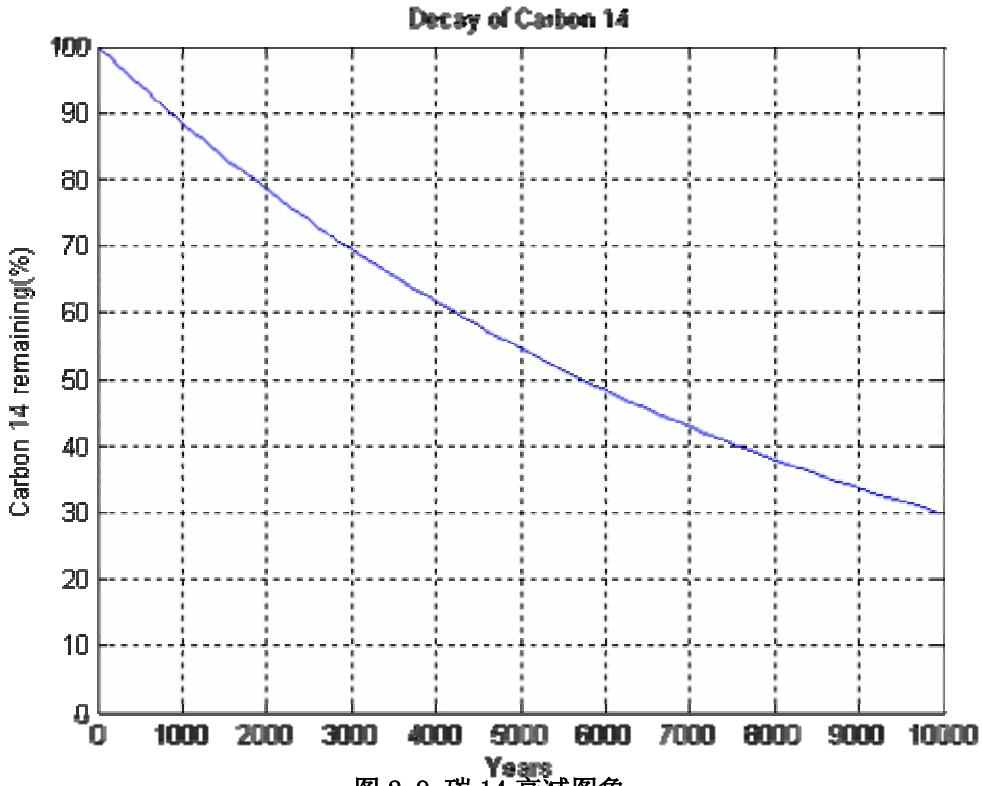


图 2.9 碳 14 衰减图象

2.13 调试 MATLAB 程序

有一个古老的说法,人这一生唯一能够确定的东西是死亡和税收.我们在这里再增加一项,无论你编定多大的程序,你第一次运行时,肯定通不过!程序中的错误我们称之为 BUGS,找出并排出它们,我们称之为调试(debugging).已知一个程序,而无法运行,我们怎样调试它呢?

在 MATLAB 中有三种类型的错误.第一种错误是**语法错误**.语法错误是 MATLAB 自身的

错误,例如拼写错误和标点错误.当编译 M 文件时,maltab 编译器将会找出这些错误.例如,语句

```
x = ( y + 3 ) / 2 );
```

有一个语法错误,因为其括号不平衡.

如果这句存储在 M 文件 test.m 中，当 test 编译进，将会出现下面的信息。

```
>> test
??? Error: File: d:\MATLAB7\work\test.m Line: 1 Column: 10
Unbalanced or misused parentheses or brackets.
```

第二种类型的错误是一种运行时错误。当一个非法的数学运算出现在程序的过程(例如，除以 0)，将会出现运行时错误。这些错误将会使程序返回 Inf 或 NaN,用来参与下一步的运算。导致这个程序的结果变无效。

错误的第三种形式是逻辑错误(logical error).逻辑错误是指编译和运行都能通过，而产生了错误的结果。

在编程过程中出现的最普遍的错误是书写错误。一些书写错误可能产生无效的 MATLAB 语句。这些错误产生的语法错误可能会被编译器发现。另一个书写错误发生在变量名的书写上。例如，变量中的字符可能被调换，漏写或错写。这样就会创建一个新的变量，在前面我们已经提到，MATLAB 能够很容易地创造一个新的变量，它不会发现这个错误。书写错误也能导致逻辑错误。例如，如果变量 vel1 和 vel2 都在程序中代表速度，如果一时疏忽用其中一个替代了另一个，那么你就只能用人工检查代码找出此类错误。有的时侯程序开始时能够执行，但是运行时错误和逻辑错误可能在执行中发生。在这种情况下，可能是输入错误，也可能是逻辑结构错误。找出这类错误的第一步是检查程序的输入数据。既可以去掉输入语句后的分号，也可以加入一个多余的输出语句以证明这个输入值是不是你想要的。如果你已经排除了变量名错误和输入值错误，接着你要处理的是逻辑错误。你应该检测是否有逻辑错误，应当检查每一个赋值语句。

1.如果一个赋值语句非常的长，把他分成许多小的赋值语句。小的语句易证明。

2.检查你的赋值语句中括号的放置。在赋值语句中，由于括号导致运算顺序错误是极其常见的错误。如果你对运算顺序仍有疑问，应该多加括号，使之更加清晰。

3.保证每个变量正确的初始化。

4.保证函数中用到的单位统一。例如，在三角函数中输入必须是弧度值，而不是角度值。如果你仍然得到的是错误的语句，在更多的位加上输出语句，以检查中间计算。如果你能确定错误的位置，那么你就知道在那里找到问题所在，百分九十五地在这片区域内。如果问题依然存在，那么这时你就应当把你遇到的问题解释给你的同学或老师，让他们给你检查错误。一个人看自己编写的代码找不到错误是非常常见的，而其他的人则可以迅速地找出错误的地方，而这个地方你可能已经看了一次又一次。

好的编程习惯

确保你在编程设计过程：

1.初始化所有变量

2.适当应用括号使运算顺序清晰以减少调试的工作量

在 MATLAB 中有一个专门的调试器，叫做 symbolic debugger. symbolic debugger 允许用户一句一句地执行语句，检测出所有的变量值，它能让你看到所有的中间值，而不用在其中加入输出语句。我们将会在第三章中介绍 symbolic debugger。

2.14 总结

在本章中，我们将向大家介绍了两种数据类型：double 和 char.我们还向大家介绍了赋值语句，数学计算，常用函数,输入输出语句和数据文件.

MATLAB 表达的运算顺序遵守一定的规则,即优先级高的先执行,优先级低的后执行.运算的优先级总结在表 2.11 中.

表 2.11 运算的优先级

优先级	运算
1	括号里的内容先运算，从最里面的括号去运算
2	幂运算，从左向右
3	乘除法，从左向右
4	加减法，从左向右

MATLAB 语言包括许许多多地的内建函数,帮助我们解决问题.它的函数比 C 和 Fortan 语言中的函数要多得多,包括机制独立的画图功能.一些常见的固有函数在表 2.8 中列出,其他函数将会在以后的章节中介绍.所有 MATLAB 的函数列表可在在线帮助浏览窗口中得到.

2.14.1 好的编程习惯

每一个 MATLAB 程序都应让其他熟悉 MATLAB 编程的人容易理解.所以有一个好的编程习惯十分重要,因为它能使一个程序使用很好时间.过了一个段时间,条件可能改变,程序也可能要改变以适应这些变化.修改这个程序的人可能是其他人而不是这个程序的原作者.这个程序员在修改程序之前必须先理解原程序.

编写清晰,易理解,可维护强的程序要比编写简单的程序要难得多.一个程序员必须发展这方面的能力以证明自己的工作,还有程序必须避免一些常见的错误.下面的指导意见,将有助于你养成好的编程习惯.

1. 尽可能的使用有意义的变量名,一眼就可以看懂,像 day,month,year.
2. 给每一个程序创建一个数据字典,以提高程序的可维护性.
3. 变量名一律用小写字母,这样可以不会因大小写不同而造成变量混淆。
4. 在所有的 MATLAB 赋值语句的后面加上一个分号,用来禁止赋值的重复.在程序调试期间,如果你检验检某个语句的值,可去掉语句后的分号.
5. 如果要在 MATLAB 和其他程序之间交换数据,那么就要以 ASCII 格式存储数据.如果数据只应用在此 MATLAB 中那么,应以 mat- file 格式存储数据.
6. 以”dat”为扩展名保存 ASCII 数据以区分 MAT 文件,MAT 文件的扩展名为 mat.
7. 用适当的括号使你的表达式清晰,易理解.
8. 当你读取和写入数据时,使用适当的单位

2.14.2 MATLAB 总结

下面的总结列举了本章出现的所有特殊符号,命令和函数,后面跟的是简短的描述.

特殊符号

符号	说明
[ ]	数组构造器
( )	用来装载下标
' '	用来限制一个字符串
,	分开下标,或分开元素
;	1.防止在命令窗口的重复
	2.分开矩阵的行
	3.在一行内分开几个赋值语句
%	标志注释的开始
:	克隆运算符
+	数组和矩阵的加法
-	数组和矩阵的减法
.*	数组乘法
*	矩阵乘法
./	数组右除法
.\	数组左除法
/	矩阵右除法

符号	说明
\	矩阵左除法
.^	数组幂运算
'	转义运算符命令和函数
…	且来表示语句太长,转到第二行写
abs(x)	计算 x 的绝对值
acos(x)	计算 x 的反余弦函数
angle（x）	计算复数 x 的幅角
asin（x）	计算 x 的反正弦函数值
atan（x）	计算 x 的反正切函数值
atan2（y，x）	
cos（x）cosx	
exp（x）	
log（x）	
[value,index]=max（x）	返回 x 中的最大值，和他所处的位置
[value,index]=min（x）	返回 x 中的最小值，和他所处的位置
mod（x,y）余数，	
sin（x）sinx	
sqrt（x）	x 的平方根
tan（x）tanx	
rounding(取整)函数	
ceil(x)	
fix(x)	
round(x)	
字符转换函数	
char(x)	将矩阵中的数转化为字符,矩阵中的元素就不大于 127
double(x)	将子符串转化为矩阵
int2str(x)	将整数 x 转化为字符串形式
num2str(x)	将带小数点的数转化为一个字符型数组
str2num(x)	将字符串转化为数
format short	保留小数点后 4 位（默认格式）
format long	保留小数点后 14 位
format short e	带有 5 位有效数字科学记数法
format short g	总共有 5 个数字，可以用科学记数法，也可不用
format long e	带有 15 位有效数字科学记数法
format long g	总共有 5 个数字，可以用科学记数法，也可不用
format bank	美元格式
format hex	用 16 进制表示
format rat	两个小整数的比
format compact	隐藏多余的换行符
format loose	使用多余的换行符
format +	只显示这个数的正负
pi	有 15 个有效值的 π
i,j	代表虚数 i($\sqrt{-1}$)
Inf	这个符号代表无穷大，它一般情况下是除以 0 产生的
NaN	这个符号代表没有这个数。它一般由数学运算得到的。例如，0 除以 0。
clock	这个特殊变量包含了当前的年，月，日，时，分，秒，是一个 6 元素行向量

符号	说明
date	包含当前的日期，是用的字符形式
eps	变量名是 epsilon 的简写。它代表计算能机辨别的两数之间的最小数
ans	常用于存储表达式的结果，如果这个结果没有明确的赋值于某个变量
char	字符型
plot	函数的 x,y 均用线性尺度
semilog	函数 x 轴用对数尺度，y 轴将用线性尺度
Semiloge	函数 x 轴用线性尺度，y 轴用对数尺度
loglog	函数两坐标轴将会都用对数尺度。

2.15 练习

2.1

看下面的数组回答有关问题

$$array1 = \begin{bmatrix} 1.1 & 0.0 & 2.1 & -3.5 & 6.0 \\ 0.0 & 1.1 & -6.6 & 2.8 & 3.4 \\ 2.1 & 0.1 & 0.3 & -0.4 & 1.3 \\ -1.4 & 5.1 & 0.0 & 1.1 & 0.0 \end{bmatrix}$$

a. array1 的大小是多少?
b. array1(4,1)的值是多少?
c. array1(:,1:2)的大小和值为多少?
d. array1([1 3],end)的大小和值为多少?

2.2

下面的变量名那些合法那些不合法.为什么?

a. dog1　　b. 1dog　　c. Do_you_know_the_way_to_san_jose
d. _help　　e. What's_up?

2.3

写出下面的数组的大小和内容.注意后面的数组可能根据前面数组的定义.

a. a=1:2:5;
b. b=[a' a' a'];
c. c=b(1:2:3,1:2:3);
d. d=a+b(2,:);
e. w=[zeros(1,3) ones(3,1)' 3:5'];
f. b([1 3],2)=b([3 1],2);

2.4

数组定义如下,写下面的子数组的内容

$$array1 = \begin{bmatrix} 1.1 & 0.0 & 2.1 & -3.5 & 6.0 \\ 0.0 & 1.1 & -6.6 & 2.8 & 3.4 \\ 2.1 & 0.1 & 0.3 & -0.4 & 1.3 \\ -1.4 & 5.1 & 0.0 & 1.1 & 0.0 \end{bmatrix}$$

a. array1(3,:)
b. array1(:,3)

c. array1(1:2:3,[3 3 4])
d. array1([1 1],:)

2.5

已知 value 的初始化值是 10π,写出下列语句的输出

```
disp(['value = ' num2str(value)]);
disp(['value = ' int2str(value)]);
fprintf('value = %e\n',value);
fprintf('value = %f\n',value);
fprintf('value = %g\n',value);
fprintf('value = %12.4f\n',value);
```

2.6

a,b,c 的定义如下,如果下面运算是合法的,那么写出结果,如果不合法,说出原因.

$$a=\begin{bmatrix}2 & -2\\ -1 & 2\end{bmatrix} \qquad b=\begin{bmatrix}1 & -1\\ 0 & 2\end{bmatrix}$$

$$a=\begin{bmatrix}1\\ -2\end{bmatrix} \qquad d=eye(2)$$

a. result=a+b; b. result=a*d; c. result=a.*d; d. result=a*c;
e. result=a.*c; f. result=a\b; g. result=a.\b; h. result=a .^ b;

2.7

求下列表达式的值

a. 11/5+6 b. (11/5)+b c. 11/(5+b)
d. 3^2^3 e. 3^(2^3) f. (3^2)^3
g. round(-11/5)+6 h. ceil(-11/5)+6 i. floor(-11/5)+6

2.8

用 MATLAB 计算下列表达式的值

a.(3-5i)(-4+6i) b. $\cos^{-1}(1.2)$

2.9

求下列联立方程组中的各 x 的值

$$\begin{aligned}
&-2.0x_1+5.0x_2+1.0x_3+3.0x_4+4.0x_5-1.0x_6=0.0\\
&2.0x_1-1.0x_2-5.0x_3-2.0x_4+6.0x_5+4.0x_6=1.0\\
&-1.0x_1+6.0x_2-4.0x_3-5.0x_4+3.0x_5-1.0x_6=-6.0\\
&4.0x_1+3.0x_2-6.0x_3-5.0x_4-2.0x_5-2.0x_6=10.0\\
&-3.0x_1+6.0x_2+4.0x_3+2.0x_4-6.0x_5+4.0x_6=-6.0\\
&2.0x_1+4.0x_2+4.0x_3+4.0x_4+5.0x_5-4.0x_6=-2.0
\end{aligned}$$

2.10

球的位置和速度.如果一静止小球在离地 h_0 的地方以初速度 v_0 做垂直运动,其等式为

$$h(t)=\frac{1}{2}gt^2+v_ot+h_o \qquad (2.10)$$

$$v(t)=gt+v_o \qquad (2.11)$$

其中 g 为重力加速度(-9.81m/(s*s)),$h(t)$代表在 t 时刻小球的高度.$v(t)$代表在时刻 t 小球的速度.编写一个 MATLAB 程序,计算出每一秒钟的速度和高度,并打印出 h,v 关于时间 t 的函数.

确保在你的图中有合适的标签.

2.11

编写一个程序,计算出坐标系中用户指定两点(X1,Y1)和(X2,Y2)之间的距离.在编个程序时,注意培养好的编程习惯.这两点之间距离的计算公式为

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \qquad (2.12)$$

2.12

工程师们经常用分贝或 dB 来描述两功率之比.1dB 的定义如下

$$dB = 10\log_{10}\frac{P_2}{P_1} \qquad (2.13)$$

P_2 是已测量的功率,P_1 代表参考功率.

a. 假设参考功率 P_1 为 1mw,编写一个程序,接受一个输入功率 P_2 并把转化成为以 1mw 为参考功率的 dB.(它在工程上有一个特殊单位 dBm).在编写程序时,注意培养好的编程习惯.

b. 写一个程序,创建一个以 W 为单位的功率的相对功率(单位为 dBm)的图象.第一个图象的 XY 轴都要用线性轴.而第二图象要用对数-线性 xy 轴.

2.13

双曲余弦.

双曲余弦的定义如下

$$\cosh x = \frac{e^x + e^{-x}}{2} \qquad (2.14)$$

编写一个程序,计算出用户指定的 x 的值对应的双曲余弦值.用这个程序计算 3.0 的双曲余弦值.和 MATLAB 中的内建函数 cosh(x)得到的值是否完全相同.用 MATLAB 打印出这个函数的图象.当 x 为何值时,这个函数有最小值?最小值为多少?

2.14

弹簧中的能量.压缩弹簧弹力的大小可由下面的公式计算出来

$$F = kx \qquad (2.15)$$

F 代表弹力,单位为 N.k 代表劲度系数单位为 N/m.存储在压力弹簧中的势能为

$$E = \frac{1}{2}kx^2 \qquad (2.16)$$

E 代表势能,单位为焦.下面是 4 个可用压缩弹簧的信息.

	弹簧 1	弹簧 2	弹簧 3	弹簧 4
力/(N)	20	24	22	20
弹簧劲度系数 k(N/m)	500	600	700	800

编写一个程序,计算出每一个弹簧的压缩量和弹力势能.哪一个弹簧的弹力最大?

2.15

收音接收机.

一个最简单的调幅收音接收机如图 2.13 所示.这个接收机由一个 *RLC* 振荡电路组成,这个振荡电路包括一个电阻,一个电容和一个电感,还有连接它们的导线.就像图中所描绘的,将这个 *RLC* 电路连接到天线和大地.

这个振荡电路保证了这个接收机在 AM 波段中的众多的电台中接收到特定的一个台.只有在共振频率下,接收的信号最强.*LC* 电路的共振频率公式为

$$f_o = \frac{1}{2\pi\sqrt{LC}} \qquad (2.17)$$

L 代表电感,单们为 H,*C* 代表电容,单们为 F.编写一个程序,输入 *L* 和 *C* 的值,计算出它的共振频率.用 *L*=0.1mH,*C*=0.5nF 来检测这个程序,计算共振频率.

2.16

收音接收机.电阻上的电压可通过频率计算出来,公式如下

$$V_R = \frac{R}{\sqrt{R^2 + (\omega L - \frac{1}{\omega C})^2}} V_o \qquad (2.18)$$

$\omega=2\pi f$,以 Hz 为单位的频率.假设 *L*=0.1mH，*C*=0.25nF，*R*=50Ω，V_0=10mV.

a. 画出以频率为自变量的电阻电压函数.在什么频率下,电阻上的电压最大?这时的电压为多少?这个频率叫做电路的固有频率.

b. 如果这个频率比固有频率大百分之十,此时电阻上的电压为多少?这个收音接收机是如何选台的?

c. 在什么频率下这个电阻上的电压会降到固有频率电压的一半?

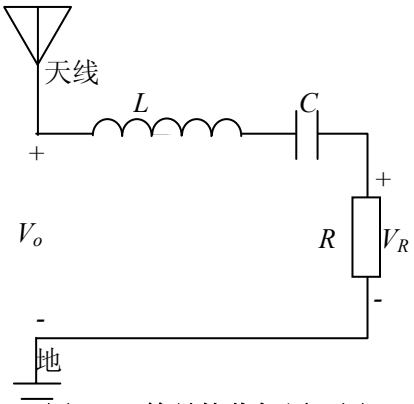


图 2.13 简易接收机原理图

2.17

假设两个信号同时被天线接收.其中一个信号的大小为 1V,频率为 1000kHz,而另一个信号的大小为 1V，950kHz.第一个信号给负载 *R* 的功率是多少?第二个信号给负载 *R* 的功率是多少?计算第二个信号相对第二个信号的增益或衰减.与第一个信号相比,第二个信号增益或衰减了多少?

2.18

飞船的运转半径.图 2.14 向大家显示的是一个做匀速圆周运动的物体.

其向心加速度公式为

$$a = \frac{v^2}{r} \qquad (2.19)$$

a 代表向心加速度,单位为 m/s^2.*v* 代表物体运动的速率,单位为 m/s,*r* 代表半径,单位为 m.假设这个物体是一个飞机,回答下列问题:

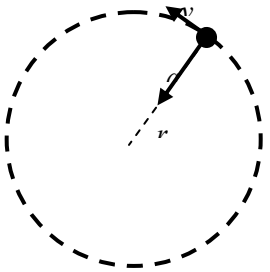


图 2.4 做匀速圆周运动的物体

a. 假设飞机的运动速度为 0.85 马赫,即声速的 85%.如果向心加速度为 2*g*,那么飞机的半径为多少?

b. 假设飞行速度增大到 1.5 马赫,那么飞机的半径为多少?

c. 运转半径是飞机飞行速度的一个函数,自变量的定义域为 0.5 马赫到 2.0 马赫,向心加速度仍为 2*g*,画出这个函数的图象.

d. 假设飞行员能忍耐的最大加速度为 7*g*.那么以 1.5 马赫飞行的最小半径为多少?

e. 画出以向心加速度为自变量的半径函数,向心加速度的取值为[2*g*,6*g*],假设运转速度为 0.85 马赫.