Matlab 与科学计算*

易鹏

中山大学

内部版本号：V4.12.025 (内测版)

2022 年 4 月 27 日

*本笔记已经开源，可以免费下载，github 地址：https://github.com/Lovely-XPP/Notebook

gitee 分流地址：https://gitee.com/sysu_xpp/Notebook

目录

第 1 章　Matlab 数据及运算

1.1　矩阵的表示

1.1.1　矩阵的建立

1. 直接输入：将矩阵的元素用中括号括起来，按矩阵行的顺序输入各元素，同一行各元素之间用逗号或空格隔开，不同行的元素之间用分号分隔。例如：

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

2. 利用已建好的矩阵建立更大的模型：大矩阵可由已建好的小矩阵拼接而成。例如：

```
>> A = [1, 2, 3; 4, 5, 6; 7, 8, 9];
>> B = [-1, -2, -3; -4, -5, -6; -7, -8, -9];
>> C = [A, B; B, A]
C =
      1     2     3    -1    -2    -3
      4     5     6    -4    -5    -6
      7     8     9    -7    -8    -9
     -1    -2    -3     1     2     3
     -4    -5    -6     4     5     6
     -7    -8    -9     7     8     9
```

1.1.2　冒号表达式

在 Matlab 中，利用冒号表达式可以产生行向量，一般格式如下：

```
e1:e2:e3
```

其中，e1为初始值，e2为步长，e3为终止值。冒号表达式可以产生一个由e1开始到e3结束。以步长e2自增的行向量。例如：

```
>> t = 0:1:5
t =
     0   1   2   3   4   5
```

注意

在冒号表达式中如果省略e2不写，则默认步长为 1.

在 Matlab 中，还可以用linspace函数产生行向量，其调用格式如下：

```
linspace(a,b,n)
```

其中，a 和 b 是生成向量的第一个和最后一个元素，n 是元素总数。当 n 省略时，i 懂产生 100 个元素。例如：

```
>> x = linspace(0, pi, 6)
x =
    0    0.6283    1.2566    1.8850    2.5133    3.1416
```

1.1.3 矩阵元素的引用

1. 矩阵元素的引用方式

   (a) 下标引用：下标必须是正整数。例如，将矩阵 $\boldsymbol{A}$ 第 3 行第 2 列的元素赋值为 200（仅改变单个元素的值，其余元素值不改变）：

   ```
   >> A(3, 2) = 200
   ```

   如果给出的行下标或列下标大于原来矩阵的行数和列数，则 Matlab 自动扩展矩阵的行数和列数，并将扩展后未赋值的的矩阵元素置为 0. 例如：

   ```
   >> A[1, 2, 3; 4, 5, 6];
   >> A(4, 5) = 10
   A =
       1    2    3    0    0
       4    5    6    0    0
       0    0    0    0    0
       0    0    0    0   10
   ```

   (b) 序号引用：矩阵元素的序号就是相应元素在内存中的排列顺序，即线性索引号。在 Matlab 中，矩阵按列储存，即首先储存矩阵的第一列元素，然后储存第二列元素，……，一直到矩阵的最后一列矩阵。例如，矩阵 $\boldsymbol{A}$ 的第 3 号元素为 2，即 $A(1,2)=2$：

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

   (c) 序号与下标的转换：对于 $m\times n$ 的矩阵 $\boldsymbol{A}$ 来说，$A(i,j)$ 的序号为 $(j-1)*m+i$。sub2ind与ind2sub函数实现相互转换。

   sub2ind调用函数：

   ```
   D = sub2ind(S, I, J)
   ```

其中，S 表示要转换的矩阵的函数和列数，是行数和列数组成的向量，通常用函数size获取；I 是行下标，J 是列下标，D 为对应下标元素的序号，其行列数与 I、J 相同。例如：

```
>> A = [1:3; 4:6]
A =
    1   2   3
    4   5   6
>> D = sub2ind(size(A), [1, 2; 2, 2], [1, 1; 3, 2])
D =
    1   2
    6   4
```

对于上面的例子，可以解释如下：

$$\begin{bmatrix} d_1 & d_2 \\ d_3 & d_4 \end{bmatrix} \quad \Longrightarrow \quad \begin{bmatrix} (i_1, j_1) & (i_2, j_2) \\ (i_3, j_3) & (i_4, j_4) \end{bmatrix} \text{下标对应的元素}$$

即 $A(1,1)$ 的序号为 1，$A(2,1)$ 的序号为 2，$A(2,3)$ 的序号为 6，$A(2,2)$ 的序号为 4.

ind2sub调用函数

```
[I, J] = ind2sub(S, D)
```

其中，S 表示要转换的矩阵的行数和列数；D 是序号，返回值为序号所对应的行下标和列下标。例如：

```
>> [I, J] = ind2sub([3, 3], [1, 3, 5])
I =
    1   3   5
J =
    1   1   2
```

2. 利用冒号表达式获得子矩阵

子矩阵是指由矩阵中的一部分元素构成的矩阵。见下表：

符号	说明
A(i,j)	$\boldsymbol{A}$ 矩阵第 i 行，第 j 列的元素
A(i,:)	$\boldsymbol{A}$ 矩阵第 i 行的全部元素
A(:,j)	$\boldsymbol{A}$ 矩阵第 j 列的全部元素
A(i:i+n,:)	$\boldsymbol{A}$ 矩阵第 $i \sim i+n$ 行的全部元素
A(:,k:k+m)	$\boldsymbol{A}$ 矩阵第 $k \sim k+m$ 列的全部元素
A(i:i+n,k:k+m)	$\boldsymbol{A}$ 矩阵第 $i \sim i+n$ 行，且在第 $k \sim k+m$ 列的全部元素

例如：

```
>> A = [1:5, 6:10, 11:15, 16:20]
A =
     1     2     3     4     5
     6     7     8     9    10
    11    12    13    14    15
    16    17    18    19    20
>> A(1,:)   % 取A的第一行
ans =
     1   2   3   4   5
>> A(:,2:4)   % 取A的第2,3,4列
ans =
     2     3     4
     7     8     9
    12    13    14
    17    18    19
>> A(2:3,4:5)   % 取A的第2,3行, 第4,5列
ans =
     9    10
    14    15
>> A(2:3,1:2:5)   % 取A的第2,3行, 第1,3,5列
ans =
     6     8    10
    11    13    15
```

此外，还可以利用一般向量和end运算符来表示矩阵下标，从而获得子矩阵。end表示矩阵某一维的末尾元素的下标。例如：

```
>> A = [1:5, 6:10, 11:15, 16:20];
>> A = (end,:)    % 取A的最后一行
ans =
    16    17    18    19    20
>> A([1,4],3:end)   % 取A的第2,3两行中第3列到最后一列的元素
ans =
     3     4     5
    18    19    20
```

3. 利用空矩阵删除矩阵的元素

空矩阵是指没有任何元素的矩阵，即在建立矩阵时，中括号中为空。

调用格式为：

```
X = [ ]
```

将某些元素从矩阵中删除，采用将其置为空矩阵的方法是一种有效的方法，例如：

```
>> A = [1, 2, 3, 0, 0; 7, 0, 9, 2, 6; 1, 4, -1, 1, 8]
A =
    1   2   3   0   0
    7   0   9   2   6
    1   4   -1    1   8
```

```
>> A(:, [2, 4]) = [ ]
A =
     1    3    0
     7    9    6
     1   -1    8
```

4. 改变矩阵的形状

reshape(A, m, n)函数在矩阵总元素保持不变的前提下，将矩阵 $\boldsymbol{A}$ 重新排成 $m \times n$ 的二维矩阵。例如：

```
>> x = [23, 45, 65, 34, 65, 34, 98, 45, 78, 65, 43, 76];
>> y = reshape(x, 3, 4)
y =
     23    34    98    65
     45    65    45    43
     65    34    78    76
>> z = reshape(y, 2, 6)
z =
     23    65    65    98    78    43
     45    34    34    45    65    76
```

注意

reshape函数只是改变原矩阵的行数和列数，即改变其逻辑结构，但并不改变原矩阵的元素个数和其储存顺序。

A(:)将矩阵 A 的每一列元素堆叠起来，成为一个列向量，从而改变了矩阵的形状。例如：

```
>> A = [-45, 65, 71; 27, 35, 91]
A =
    -45    65    71
     27    35    91
>> B = A(:)
B =
    -45
     27
     65
     35
     71
     91
```

1.2 变量及其操作

1.2.1 变量的命名

- 在 Matlab 中，变量名是以字母开头、后接字母、数字或下画线的字符序列，最多 63 个字符。
- 在 Matlab 中，变量名区分字母的大小写。例如：example，Exmple，EXAMPLE是 3 个不同的变量。

注意

Matlab 提供的标准函数名以及命令名一般都是小写字母。例如求矩阵 A 的逆用`inv(A)`，不能写成`Inv(A)`，`INV(A)`，否则会出错。

1.2.2 变量的赋值

变量的赋值有以下下两种方式：

- 变量 = 表达式
  这种方式是把表达式的值直接赋值给变量。
- 表达式
  这种方式是把表达式的值赋值给 Matlab 预定义的变量`ans`中。

当赋值命令的结果不需要在命令行窗口输出的时候，通常在赋值语句后面加上;。

例 1.1 计算表达式 $\frac{5+\cos 47^\circ}{1+|x-y|}$ 的值，并将结果赋值给变量 z，然后显示计算结果。其中 $x=\sqrt{7}-2\mathrm{i}, y=\mathrm{e}^{\frac{\pi}{2}}$.

```
>> x = sqrt(7) - 2i;
>> y = exp(pi / 2);
>> z = (5 + cos(47 * pi / 180) / (1 + abs(x - y)))
z =
    1.4395
```

1.2.3 预定义变量

Matlab 常用的自定义变量如下表：

符号	说明	符号	说明
`ans`	计算结果的默认赋值变量	`nargin`	函数输入参数的个数
`eps`	机器零阈值	`nargout`	函数输出参数的个数
`pi`	圆周率 π 的近似值	`realmax`	最大正实数
`i, j`	虚数单位	`realmin`	最小正实数
`inf, Inf`	无穷大，如 $1/0$ 的结果	`lasterr`	存放最新的错误消息
`NaN, nan`	非数，如 $0/0$,inf/inf 的结果	`lastwarn`	存放最新的警告消息

注意

对于 Matlab 预定义的变量，在使用时尽量避免对这些变量重新赋值。对于我们循环常用的 i, j 来说，进行复数运算我们要注意：对复数赋值使用`3 + 4i`这样的记法，而不是`3 + 4 * i`的记法，下面举例说明不同之处。

```
>> i = 10;
>> 3 + 4i
ans =
    3.0000 + 4.0000i
>> 3 + 4 * i
ans =
    43
```

1.2.4 变量的管理

1. 内存变量的删除和修改

MATLAB 工作区窗口专门用于内存变量的管理。在工作区窗口中可以显示所有内存变量的属性。当选中某些变量后，选择右键快捷菜单中的“删除”命令，就能清除这些变量。当选中某个变量后，双击该变量或选择右键快捷菜单中的“打开所选内容”命令，将进入变量编辑器。通过变量编辑器可以直接观察变量中的具体元素，也可以修改变量中的具体元素。

通常，对于较大矩阵的输入，可以采用变量编辑器，操作方法如下：

(1) 在工作区窗口的右键快捷键中选择“新建”命令，并给变量命名。

(2) 在工作区中双击该变量，打开变量编辑器。

(3) 在变量编辑器的空白表格中填写元素值，表格的每一个方格对应矩阵的一个元素。

命令行窗口删除、查看变量的函数如下表：

命令	说明
`clear`	删除工作区的所有变量
`who`	输出所有变量的变量名
`whos`	输出所有变量的变量名、大小、所占字节数及数据类型等信息

```
>> who
Your variables are:
a x y z
>> whos
Name      Size            Bytes  Class     Attributes
a         2*6                96    double
x         1*1                 8    double
y         1*1                 8    double
z         1*1                 8    double
```

2. 内存变量文件

利用 Mat 文件可以把当前 Matlab 工作区中的一些有用变量长久的储存下来。Mat 文件是 Matlab 保存数据的一种标准的二进制格式文件，扩展名一定是.mat。Mat 文件的生成和载入由save和load来完成。

常用格式如下：

```
save [file name] [variable names] [-append] [-ascii]

load [file name] [variable names] [-ascii]
```

其中，

- file name是储存 Mat 文件的文件名（可以带路径，但不需要带后缀名.mat）。
- variable names是指变量名表，个数不限，只要工作区内存或文件中存在即可，变量名之间以空格分隔。**当变量名省略时，保存或载入全部变量。**
- -ascii选项使文件以 ASCII 格式处理，省略该选项时文件将以二进制格式储存。
- -append选项在save命令中控制将变量追加到 Mat 文件中。

```
>> save d:\example\mydata a x
>> load d:\example\mydata
```

1.3 Matlab 常用的内部函数

1.3.1 常用的数学函数

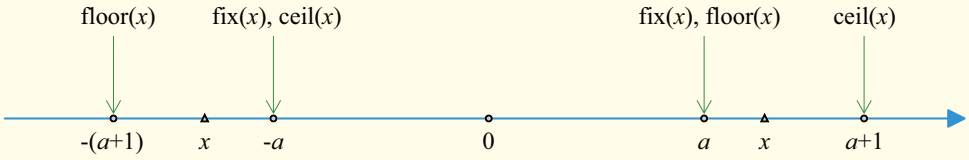
函数名	功能	函数名	功能
sin (sind)	正弦函数，输入值为弧度（角度）	abs	绝对值函数
cos (cosd)	余弦函数，输入值为弧度（角度）	rem	求余函数
tan (tand)	正切函数，输入值为弧度（角度）	mod	求模函数
asin (asind)	反正弦函数，输入值为弧度（角度）	fix	向零方向取整
acos (acosd)	反余弦函数，输入值为弧度（角度）	floor	不大于自变量的最大函数
atan (atand)	反正切函数，输入值为弧度（角度）	ceil	不小于自变量的最小整数
sinh (asinh)	双曲正弦函数（反双曲正弦函数）	round	四舍五入到最邻近的整数
cosh (acosh)	双曲余弦函数（反双曲余弦函数）	sign	符号函数
tanh (atanh)	双曲正切函数（反双曲正切函数）	gcd	最大公约数
sqrt	平方根函数	lcm	最小公倍数
log	自然对数函数	factorial	阶乘
log10	常用对数函数	isprime	判断是否为素数
log2	以 2 为底的对数函数	primes	生成素数列表
exp	自然指数幂数	perms	生成所有排列
pow2	2 的幂数	randperms	生成任意排列

注意

1. 取整函数的区别

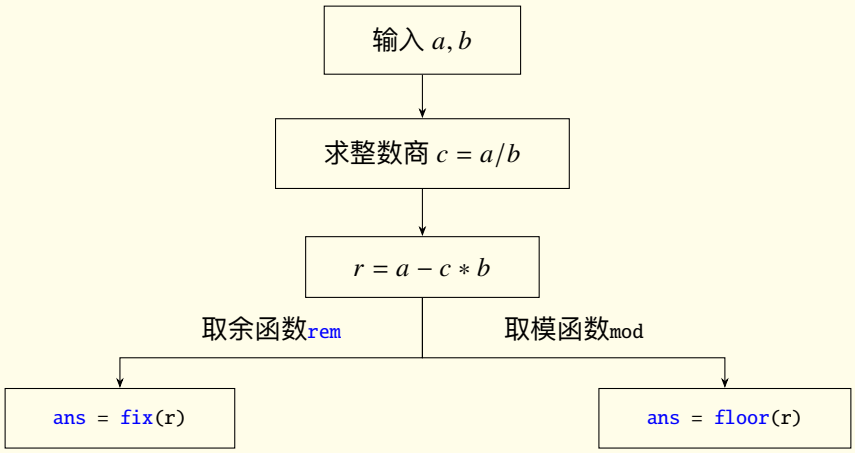
常用的取整函数有：`fix`，`floor`，`ceil`，`round`。其中，`round`等价于四舍五入。设 a 为最靠近 x 的正整数（$|x| \geq a$），则其余 3 个函数的区别如图1.1.

图 1.1: `fix`，`floor`，`ceil`取整函数对区别示意图



2. 求余运算和求模运算的区别如图1.2.

图 1.2: 求余运算和求模运算的算法示意图



1.3.2 矩阵的超越函数

Matlab 还提供了一些直接作用于矩阵的超越函数，**这些函数名都在上述内部函数名之后缀以 m，并规定输入参数A必须是方阵。**常见的超越函数如下表：

函数	说明
`sqrtm(A)`	计算矩阵 $\boldsymbol{A}$ 的平方根，即 $\sqrt{\boldsymbol{A}} \cdot \sqrt{\boldsymbol{A}} = \boldsymbol{A}$
`logm(A)`	计算矩阵 $\boldsymbol{A}$ 的自然对数
`expm(A)`	计算矩阵指数 $\mathrm{e}^{\boldsymbol{A}}$

普通矩阵函数

`funm(A, @fun)`对方阵 $\boldsymbol{A}$ 计算由`fun`定义的矩阵函数值。例如，当`fun`取`exp`时，`funm(A, @exp)`可以计算矩阵 $\boldsymbol{A}$ 的指数，与`expm(A)`的计算结果一致：

```
>> A = [2, -1; 1, 0];
>> funm(A, @exp)
ans =
    5.4366    -2.7183
```

```
    2.7183    0.0000
>> expm(A)
ans =
    5.4366   -2.7183
    2.7183         0
```

注意

funm(A, @fun)可以使用的函数有：exp, log, sin, cos, sinh, cosh等，**但求矩阵的的平方根只能用sqrtm(A)**。

1.4 Matlab 运算

1.4.1 算术运算

Matlab 运算是在矩阵意义下进行的，单个数据的算术运算只是一种特例。运算符号如下表：

运算符	说明	适用条件
A + B	矩阵的加法运算	矩阵 A, B 同型
A - B	矩阵的减法运算	矩阵 A, B 同型
A * B	矩阵的乘法运算	矩阵 A 的列数等于矩阵 B 的行数
A \ B	矩阵 A 左除矩阵 B，即 A 的逆左乘 B 矩阵	矩阵 A, B 同型
B / A	矩阵 B 右除矩阵 A，即 B 的逆右乘 A 矩阵	矩阵 A, B 同型
A ^ x	矩阵 A 的 x 次幂	矩阵 A 是方阵
A .* B	矩阵 A 和矩阵 B 单个元素之间对应相乘	矩阵 A, B 同型
A .\ B	矩阵 A 除以矩阵 B 的对应元素	矩阵 A, B 同型
B ./ A	与A .\ B等价	矩阵 A, B 同型
A .^ B	矩阵 A 和矩阵 B 单个元素之间对应取幂	矩阵 A, B 同型

注意

点运算和乘除法的区别

点运算是指两个矩阵对应元素进行基本的算术运算，而乘除法运算是矩阵特有的乘除法运算。尤其是乘法，点乘和乘法对矩阵的要求不一样，容易出错。下面举个例子。

```
>> x = 0.1:0.3:1;

>> y = sin(x) * cos(x)
Error using  *
Incorrect dimensions for matrix multiplication. Check that the number of columns in the first
matrix matches the number of rows in the second matrix. To perform elementwise multiplication,
use '.*'.

```

```
>> y = sin(x) .* cos(x)
y =
    0.0993    0.3587    0.4927    0.4546
```

1.4.2 关系运算

Matlab 提供了多种关系运算符及函数如下：

运算符	函数	说明
<	lt	小于
<=	le	小于等于
>	gt	大于
>=	ge	大于等于
==	eq	等于
~=	ne	不等于
——	all	若向量的所有元素非零，则结果为 1，否则为 0
——	any	若向量中任何一个元素非零，则结果为 1，否则为 0
——	exsit	若变量在工作区是否存在，如果存在，结果为 1，否则为 0
——	find	找出向量或矩阵中非零元素的位置
——	isempty	若被查矩阵是空矩阵，则结果为 1，否则为 0
——	isinf	若元素是inf，则结果矩阵相应位置元素取 1，否则取 0
——	isnan	若元素是nan，则结果矩阵相应位置元素取 1，否则取 0
——	isfinite	若元素大小有限，则结果矩阵相应位置元素取 1，否则取 0
——	isinteger	若元素变量是整型，则结果取 1，否则取 0
——	isnumeric	若元素变量是数值型，则结果取 1，否则取 0
——	isreal	若元素变量是实数，则结果取 1，否则取 0
——	isfloat	若元素变量是浮点型，则结果取 1，否则取 0

关系运算符的运算法则如下：

- 结果显示：成立为1，不成立为0.
- 标量比标量：两数直接按关系符运算。
- 矩阵比矩阵（要求同型）：相同位置的元素一一按关系符运算，每个元素的位置得到一个结果，最后输出一个结果矩阵（与比较矩阵同型）。
- 标量比矩阵：标量与矩阵的每一个元素一一按关系符运算，每个元素的位置得到一个结果，最后输出一个结果矩阵（与比较矩阵同型）。

例 1.2 产生 5 阶随机方阵 $\boldsymbol{A}$，其元素为 $[10, 90]$ 区间的随机整数，然后判断 $\boldsymbol{A}$ 元素是否能被 3 整除。

```
>> A = fix((90-10+1) * rand(5) + 10)
A =
    75    17    22    21    63
    83    32    88    44    12
```

```
    20    54    87    84    78
    83    87    49    74    85
    61    88    74    87    64
P = rem(A, 3) == 0      % 等价于 P = eq(rem(A, 3), 0)
P =
    1    0    0    0    1
    0    0    0    0    1
    0    1    1    1    1
    0    1    0    0    0
    0    0    0    1    0
```

1.4.3 逻辑运算

Matlab 提供了多种关系运算符及函数如下：

运算符	函数	说明
a & b	and(a, b)	与：当 a, b 全为非零时，结果为 1，否则为 0
a | b	or(a, b)	或：当 a, b 中只要有一个非零，结果为 1，否则为 0
~a	not(a, b)	非：当 a 是零时，结果为 1，否则为 0
——	xor(a,b)	异或：当 a, b 当值不同时，结果为 1，否则为 0

逻辑运算的运算法则

- 标量和标量：两数直接按逻辑运算符运算。
- 矩阵和矩阵（要求同型）：相同位置的元素一一按逻辑运算符运算，每个元素的位置得到一个结果，最后输出一个结果矩阵（与比较矩阵同型）。
- 标量和矩阵：标量与矩阵的每一个元素一一按逻辑运算符运算，每个元素的位置得到一个结果，最后输出一个结果矩阵（与比较矩阵同型）。

例如：

```
>> A = [4, 65, -54, 0 ,6];
>> B = [0, 5, 3, 2, -6];
>> A & B
ans =
    0    1    1    0    1
>> xor(A > 10 , B < 10)
ans =
    1    0    1    1    1
```

注意

运算的优先级

算术运算 > 关系运算 > 逻辑运算

1.5 字符串

1.5.1 字符串的表示

- **在 Matlab 中，字符串是用单引号括起来的字符序列，特别的，单引号字符是用两个的单引号来表示。**例如：

```
>> str = ' I ''m a teacher. '
str =
    I 'm a teacher.
```

- **Matlab 将一个字符串当作一个行向量，每个元素对应一个字符，其引用方法和数值矩阵相同。**例如：

```
>> A = 'abcdefg';
>> A(1:3)
ans =
    abc
```

- 在各行字符数相等的情况下（不相等可以用空格补齐），可以建立字符串矩阵。例如：

```
>> ch = ['abcdef'; '12345 ']
ch =
    'abcdef'
    '12345 '
>> ch(2, 3)
ans =
    3
```

例 1.3 建立一个字符串向量，然后对该向量做如下处理。

(1) 取 1～5 个字符组成对子字符串。

(2) 将字符串倒序。

(3) 将字符串中的小写字母变成相应的大写字母，其余字符不变。

(4) 统计字符串中小写字母的个数。

```
>> ch = 'Sun Yat-sen University';
>> subch = ch(1:5)      % 取子字符串
subch =
    'Sun Y'
>> revch = ch(end:-1:1)     % 字符串倒序
revch =
    'ytisrevinU nes-taY nuS'
>> k = find(ch >= 'a' & ch <= 'z');     % 找小写字母的位置
>> ch(k) = ch(k) - ('a' - 'A')      % 将小写字母换成大写字母
ch =
    'SUN YAT-SEN UNIVERSITY'
>> length(k)      % 小写字母的个数
```

```
ans =
    16
```

1.5.2 字符串的操作

常见的字符串操作见下表：

功能	函数名	说明
字符串的执行	eval	把字符串的内容作为对应的 Matlab 命令来执行
字符串的转换	abs, double	获取字符串矩阵对应的 ASCII 码数值矩阵
	char	把 ASCII 码矩阵转换成字符串矩阵
	setstr	把单个 ASCII 码值转换成对应的字符
	str2num, str2double	把数字字符转换成数值
	num2str	把数值转换成字符串
	int2str	把整数转换成字符串
字符串的连接	['str1', str2]	将若干个字符串连接起来
	strcat	将若干个字符串连接起来
字符串的比较	关系运算符	**当两个字符串拥有相同的长度时，**关系运算符对每个字符对 ASCII 码进行比较，最后输出一个数值向量，其元素为对应对比较结果
	strcmp(s1, s2)	比较字符串s1，s2是否相等，相等为 1，否则为 0
	strcmp(s1, s2, n)	比较前 n 个字符是否相等，相等为 1，否则为 0
	strcmpi(s1, s2)	忽略字符串大小写的情况下，比较字符串s1，s2是否相等，相等为 1，否则为 0
	strcmp(s1, s2, n)	忽略字符串大小写的情况下，比较前 n 个字符是否相等，相等为 1，否则为 0
字符串的查找	findstr(s1, s2)	返回短字符串在长字符串中的开始位置。
字符串的替换	strrep(s1, s2, s3)	将字符串的s1中的所有子字符串s2替换成s3

```
% 字符串的执行
>> t = pi;
>> m = ' [t, sin(t), cos(t)] '
>> y = eval(t)
y =
    3.1416    0.0000    -1.0000

% 字符串的转换
>> s1 = 'MATLAB'
>> a = abs(s1)
a =
    77    65    84    76    65    66
>> char(a + 32)
ans =
    matlab

% 字符串的连接
>> f = 70;
>> c = (f - 32) / 1.8;
>> [ 'Room temperature is ', num2str(c), ' degrees C.']
ans =
    Room temperature is 21.1111 degrees C.
>> strcat('ss', 'ff', 'DD', '1234')
ans =
    ssffDD1234

% 字符串的比较
>> 'www0' >= 'W123'
ans =
    1   1   1   0
>> strcmp('www0', 'W123')
ans =
    0
>> strncmpi('www0', 'W123')
ans =
    1

% 字符串的查找
>> p = findstr('This is a test!', 'is')
p =
    3   6

% 字符串的替换
>> result = strrep('This is a test!', 'test', 'class')
result =
    This is a class!
```

1.6 结构数据和单元数据

1.6.1 结构数据

1. 结构矩阵的建立与引用

- 结构矩阵的元素可以是不同的数据类型，它能将**一组具有不同属性的数据**纳入到一个统一的变量名下进行管理。
- 建立一个结构矩阵的表达式：

  结构矩阵元素.变量名 = 表达式

  其中，表达式应理解为矩阵表达式，而且，结构矩阵元素的成员也可以是结构数据。
- 引用结构矩阵
  - 引用结构矩阵的非矩阵成员：显示其值
  - 引用结构矩阵的矩阵成员：仅显示其大小参数而不显示其具体内容
  - 引用结构矩阵的元素：显示成员名和它的值
  - 引用结构矩阵：只显示结构矩阵的大小参数和成员名
- 示例代码

```
>> a(1).x1 = 10; a(1).x2 = 'liu'; a (1).x3 = [11, 21; 34, 78];
>> a(2).x1.x11 = 90; a(2).x1.x12 = 12; a(2).x1.x13 = 30; a(2).x2 = 'wang'; a(2).x3 =
    [34, 191; 27, 578];
>> a(3).x1 = 14; a(3).x2 = 'cai'; a(3).x3 = [13, 890; 67, 231];
>> a(2).x3      % 引用结构矩阵元素a(2)的成员x3
ans =
      34   191
    27   578
>> a(2)        % 引用结构矩阵元素a(2)
ans =
  包含以下字段的 struct:
    x1: [1×1 struct]
    x2: 'wang'
    x3: [2×2 double]
>> a           % 引用结构矩阵a
a =
  包含以下字段的 1×3 struct 数组:
    x1
    x2
    x3
```

2. 结构成员的修改

- 增加成员：

```
>> a(1).x4 = '410075'
```

- 删除成员：

```
>> a = rmfield(a, 'x4')
```

第 2 章　Matlab 矩阵处理

2.1　特殊矩阵

2.1.1　常用的特殊矩阵

常用的产生通用的特殊矩阵的函数如下表。

函数	说明
zeros	产生全 0 矩阵，即零矩阵
ones	产生全 1 矩阵，即幺矩阵
eye	产生单位矩阵
rand	产生 $(0,1)$ 区间均匀分布的随机矩阵
randn	产生均值为 0，方差为 1 的标准正态分布随机矩阵

这些函数的调用格式相似，以zeros函数为例子：

函数	说明
zeros(m)	产生 $m \times m$ 零矩阵
zeros(m,n)	产生 $m \times n$ 零矩阵
zeros(size(A))	产生与矩阵 $\boldsymbol{A}$ 同样大小的零矩阵

注意

建立随机矩阵

已知满足在 $(0,1)$ 区间上均匀分布的随机数 x_i，则任意 (a,b) 区间上均匀分布的随机数为

$$y_i = a + (b-a)x_i \tag{2.1}$$

已知满足均值为 0，方差为 1 的标准正态分布的随机数 x_i，则均值为 μ，方差为 σ^2 的随机数为

$$y_i = \mu + \sigma x_i \tag{2.2}$$

例 2.1 建立随机矩阵。

1. 在区间 $(20, 50)$ 内均匀分布的 5 阶随机矩阵。

2. 均值为 0.6，方差为 0.1 的 5 阶正态分布随机矩阵。

```
>> x = 20 + (50 - 20)*rand(5)
x =
    21.6185   37.0647   43.8285   27.8891   22.5146
    35.9239   34.0817   29.3365   39.6224   26.8693
    43.3750   20.3571   35.8560   40.6764   47.4001
    48.0203   30.1137   24.9695   42.4445   24.5713
    23.8972   24.8655   38.0595   33.5162   44.7745
>> y = 0.6 + sqrt(0.1)*randn(5)
y =
    0.6392    1.5196    0.5138    0.4881    0.6106
    1.0543    0.8610    0.9474    0.3396    0.1783
   -0.0201    1.0361    0.5121    0.1013    0.9565
    0.5375    0.2654    0.8218    0.7606    0.7107
    0.2180    0.4518   -0.0488    0.6892    0.5054
```

2.1.2 用于专门学科的特殊矩阵

1. 魔方矩阵

   - 性质：
     - n 阶魔方矩阵，其元素由 $1, 2, 3, \cdots, n^2$ 共 n^2 个整数组成。
     - 每行、每列及两条对角线上的元素和都等于 $\dfrac{n(n^2+1)}{2}$.
   - 函数：`magic(n)`，注意：$n > 3$.

2. 范德蒙矩阵

   - 性质：范德蒙矩阵的最后一列为 1，倒数第二列为一个指定的向量，其他各列是最后一列与倒数第二列对应元素的乘积。即

$$\boldsymbol{A} = \begin{bmatrix} a_1^{n-1} & \cdots & a_1^2 & a_1 & 1 \\ a_2^{n-1} & \cdots & a_2^2 & a_2 & 1 \\ a_3^{n-1} & \cdots & a_3^2 & a_3 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ a_n^{n-1} & \cdots & a_n^2 & a_n & 1 \end{bmatrix} \tag{2.3}$$

   - 函数：`vander(V)`，生成的矩阵与式(2.3)完全相同，其中输入的列向量 $\boldsymbol{V}$ 为：

$$\boldsymbol{V} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

3. 希尔伯特矩阵

- 性质：每个元素为

$$h_{ij} = \frac{1}{i+j-1} \tag{2.4}$$

这个矩阵是一个高度病态的矩阵，即任何一个元素法身微笑变动，整个矩阵的值和你矩阵都会发生很大变化，病态程度和阶数相关。

- 函数：`hilb(n)`，逆矩阵`invhlib(n)`。

4. 托普利兹矩阵

- 性质：除第一行第一列外，其他每个元素都与左上角的元素相同。
- 函数：`toeplitz(x,y)`，其中 $\boldsymbol{x}, \boldsymbol{y}$ 分别为第一行和第一列，均为向量。`toeplitz(x)`用向量 $\boldsymbol{x}$ 生成一个对称的托普利兹矩阵。
- 例子：

```
>> T = toeplitz(1:6)
T =
     1     2     3     4     5     6
     2     1     2     3     4     5
     3     2     1     2     3     4
     4     3     2     1     2     3
     5     4     3     2     1     2
     6     5     4     3     2     1
```

5. 伴随矩阵

- 定义：设多项式 $p(x)$ 为

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \tag{2.5}$$

称矩阵

$$\boldsymbol{A} = \begin{bmatrix} -\frac{a_{n_1}}{a_n} & -\frac{a_{n-2}}{a_n} & -\frac{a_{n-3}}{a_n} & \cdots & -\frac{a_2}{a_n} & -\frac{a_1}{a_n} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{2.6}$$

为多项式 $p(x)$ 的伴随矩阵，$p(x)$ 称为矩阵 $\boldsymbol{A}$ 的特征多项式，方程 $p(x) = 0$ 的根称为矩阵 $\boldsymbol{A}$ 的特征值。

- 函数：`compan(p)`，其中 $\boldsymbol{p}$ 是一个多项式的系数向量，即

$$\boldsymbol{p} = \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_1 \\ a_0 \end{bmatrix}$$

6. 帕斯卡矩阵

- 定义：二次项 $(x+y)^n$ 展开后的系数随 n 的增大组成一个三角形表，称为杨辉三角形。由杨辉三角形表组成的矩阵称为帕斯卡) 矩阵。
- 函数：`pascal(n)`生成一个 n 阶帕斯卡矩阵（$(x+y)^{n-1}$ 的展开式）。

2.2 矩阵变换

2.2.1 对角阵与三角阵

1. 对角阵

- 定义：只有对角线上有非 0 元素的矩阵称为对角矩阵，对角线上的元素相等的对角矩阵称为数量矩阵，对角线上的元素都为 1 的对角矩阵称为单位矩阵。
- 函数
  - 提取矩阵的对角阵元素：`diag(A)`, `diag(A, k)`.
    用于提取主对角线和第 k 条对角线[1]上的元素。
  - 构造对角阵：`diag(V)`, `diag(V,k)`.
    用于生成以向量 $\boldsymbol{V}$ 为主对角线的方阵和以向量 $\boldsymbol{V}$ 为第 k 条对角线的矩阵。

例 2.2 先建立 5×5 矩阵 A，然后将 $\boldsymbol{A}$ 的第一行元素乘以 1，第二行乘以 2，…，第五行乘以 5。

```
>> A = [17, 0, 1, 0, 15; 23, 5, 7, 14, 16 ; 4, 0, 13, 0, 22; 10, 12, 19, 21, 3;...
  11, 18, 25, 2, 19];
>> D = diag(1:5);
>> D*A    % 用D左乘A，对A的每行乘以一个指定常数，左行右列
ans =
      17    0    1    0    15
      46    10    14    28    32
      12    0    39    0    66
      40    48    76    84    12
      55    90    125  10    95
```

2. 三角阵

- 上三角阵
  - 定义：矩阵的对角线以下的元素全为 0 的一种矩阵
  - 函数：`triu(A)`, `triu(A, k)`.
    用于求矩阵 $\boldsymbol{A}$ 的上三角阵和第 k 条对角线以上的元素。
- 下三角阵
  - 定义：矩阵的对角线以上的元素全为 0 的一种矩阵
  - 函数：`tril(A)`, `tril(A, k)`.
    用于求矩阵 $\boldsymbol{A}$ 的下三角阵和第 k 条对角线以下的元素。

[1] 主对角线 $k=0$，主对角线上方 $k>0$，主对角线下方 $k<0$，下同

2.2.2 矩阵的转置与旋转

矩阵的转置与旋转操作函数和操作说明见下表：

功能	函数名	说明
转置	A.'	矩阵转置
共轭转置	A'	转置的基础上还要取每个数的复共轭
矩阵旋转	rot90(A, k)	将矩阵 $\boldsymbol{A}$ 旋转 90° 的 k 倍，当 k 为 1 时可省略
矩阵的左右翻转	fliplr(A)	将原矩阵的第一列和最后一列调换， 第二列和倒数第二列调换，…，依次类推
矩阵的上下翻转	flipud(A)	将原矩阵的第一行和最后一行调换， 第二行和倒数第二行调换，…，依次类推

2.2.3 矩阵的逆和伪逆

矩阵的逆和伪逆函数和操作说明见下表：

功能	函数名	说明
逆矩阵	B = inv(A)	求矩阵的逆
伪逆矩阵 （广义逆矩阵）	B = pinv(A)	如果矩阵 $\boldsymbol{A}$ 不是一个方阵，或者 $\boldsymbol{A}$ 是一个非满秩的方阵时， 矩阵 $\boldsymbol{A}$ 没有逆矩阵，但可以找到一个与 $\boldsymbol{A}$ 的转置矩阵 $\boldsymbol{A}'$ 同型的矩阵 $\boldsymbol{B}$，使得：$\boldsymbol{A}\cdot\boldsymbol{B}\cdot\boldsymbol{A}=\boldsymbol{A}$，$\boldsymbol{B}\cdot\boldsymbol{A}\cdot\boldsymbol{B}=\boldsymbol{B}$

2.3 矩阵求值

2.3.1 方阵的行列式

把一个方阵看作一个行列式，并按行列式的规则来求值的函数：

```
det(A)
```

2.3.2 矩阵的迹和秩

1. 矩阵的秩

   - 定义：矩阵线性无关的行数与列数称为矩阵的秩。通常，对于一组向量 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_p$，若存在一组不全为零的数 k_i $(i=1,2,\cdots,p)$，使

   $$k_1\boldsymbol{x}_1+k_2\boldsymbol{x}_2+\cdots k_p\boldsymbol{x}_p=0 \tag{2.7}$$

   那么这 p 个向量线性相关，反之线性无关。

   - 函数：rank(A).

2. 矩阵的迹

- 定义：矩阵的迹等于矩阵的对角线元素之和，也等于矩阵的特征值之和。
- 函数：`trace(A)`，矩阵 $\boldsymbol{A}$ 必须是方阵。

2.3.3 向量和矩阵的范数

类型	范数类型	函数	数学表达式	说明
向量	1--范数	`norm(V,1)`	$\|\boldsymbol{V}\|_1 = \sum_{n}^{i=1} \|v_i\|$	向量元素的绝对值之和
	2--范数	`norm(V,2)`; `norm(V)`	$\|\boldsymbol{V}\|_2 = \sqrt{\sum_{n}^{i=1} \|v_i\|^2}$	向量元素绝对值的平方和的平方根
	∞--范数	`norm(V,inf)`;	$\|\boldsymbol{V}\|_\infty = \max_{1\le i\le n}\{\|v_i\|\}$	向量元素绝对值中的最大值
矩阵	1--范数	`norm(A,1)`	$\|\boldsymbol{V}\|_1 = \sum_{n}^{i=1} \|v_i\|$	向量元素的绝对值之和
	2--范数	`norm(A,2)`; `norm(A)`	$\|\boldsymbol{V}\|_2 = \sqrt{\sum_{n}^{i=1} \|v_i\|^2}$	向量元素绝对值的平方和的平方根
	∞--范数	`norm(A,inf)`;	$\|\boldsymbol{V}\|_\infty = \max_{1\le i\le n}\{\|v_i\|\}$	向量元素绝对值中的最大值

2.3.4 矩阵的条件数

矩阵 $\boldsymbol{A}$ 的条件数等于 $\boldsymbol{A}$ 的范数与 $\boldsymbol{A}$ 的逆矩阵的范数的乘积，即

$$\mathrm{cond}(\boldsymbol{A}) = \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\| \tag{2.8}$$

定义的条件数总是大于 1 的。条件数越接近 1，矩阵的性能越好。反之，矩阵的性能越差。

范数类型	函数	数学表达式
1-范数下的条件数	`cond(A,1)`	$\mathrm{cond}(\boldsymbol{A},1) = \|\boldsymbol{A}\|_1 \cdot \|\boldsymbol{A}^{-1}\|_1$
2-范数下的条件数	`cond(A,2)`; `cond(A)`	$\mathrm{cond}(\boldsymbol{A},2) = \|\boldsymbol{A}\|_2 \cdot \|\boldsymbol{A}^{-1}\|_2$
∞-范数下的条件数	`cond(A,inf)`;	$\mathrm{cond}(\boldsymbol{A},\infty) = \|\boldsymbol{A}\|_\infty \cdot \|\boldsymbol{A}^{-1}\|_\infty$

2.3.5 矩阵的特征值与特征向量

定义 2.1 特征值与特征向量

对于 n 阶方阵 $\boldsymbol{A}$，求数 λ 和向量 $\boldsymbol{\xi}$，使得

$$\boldsymbol{A}\boldsymbol{\xi} = \lambda\boldsymbol{\xi} \tag{2.9}$$

即

$$(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{\xi} = 0 \tag{2.10}$$

则满足式(2.10)的向量 $\boldsymbol{\xi}$ 称为特征向量，数 λ 称为特征值。

由于式(2.10)有非零解，所以必须使其系数行列式为 0，即

$$\left|\boldsymbol{A}-\lambda\boldsymbol{I}\right|=0 \tag{2.11}$$

Matlab 求特征值和特征向量的函数如下表。

函数名 & 用法	说明
`V = eig(A)`	求矩阵 $\boldsymbol{A}$ 的全部特征值，构成向量 $\boldsymbol{V}$
`[X, D] = eig(A)`	求矩阵 $\boldsymbol{A}$ 的全部特征值，构成对角阵 $\boldsymbol{D}$，并产生矩阵 $\boldsymbol{X}$，$\boldsymbol{X}$ 各列是相应的特征向量。
`[X, D] = eig(A, 'nobalance')`	同上

注意

`[X, D] = eig(A, 'nobalance')` 与 `[X, D] = eig(A)` 的区别

`[X, D] = eig(A, 'nobalance')`是直接计算矩阵 $\boldsymbol{A}$ 的特征值和特征向量，而`[X, D] = eig(A)`在计算矩阵 $\boldsymbol{A}$ 的特征值和特征向量之前先求矩阵 $\boldsymbol{A}$ 的相似变换。

第 3 章　Matlab 程序流程控制

3.1　M 文件

M 文件分为两种类型：脚本（script）文件和函数（function）文件。

类别	脚本	函数
定义	批量语句，又称命令文件	声明一个函数，方便以后调用
输入 & 输出	无输入，无输出	可输入，可输出
变量操作	对工作区变量直接操作	操作变量仅为局部变量，不影响工作区
运行方式	直接运行：工作区窗口输入脚本名字	不能直接运行，需要调用

3.2　程序控制结构

3.2.1　顺序结构

顺序结构是指按照程序中语句对排列顺序依次执行，直到程序对最后一个语句为止。常见对操作函数如下表。

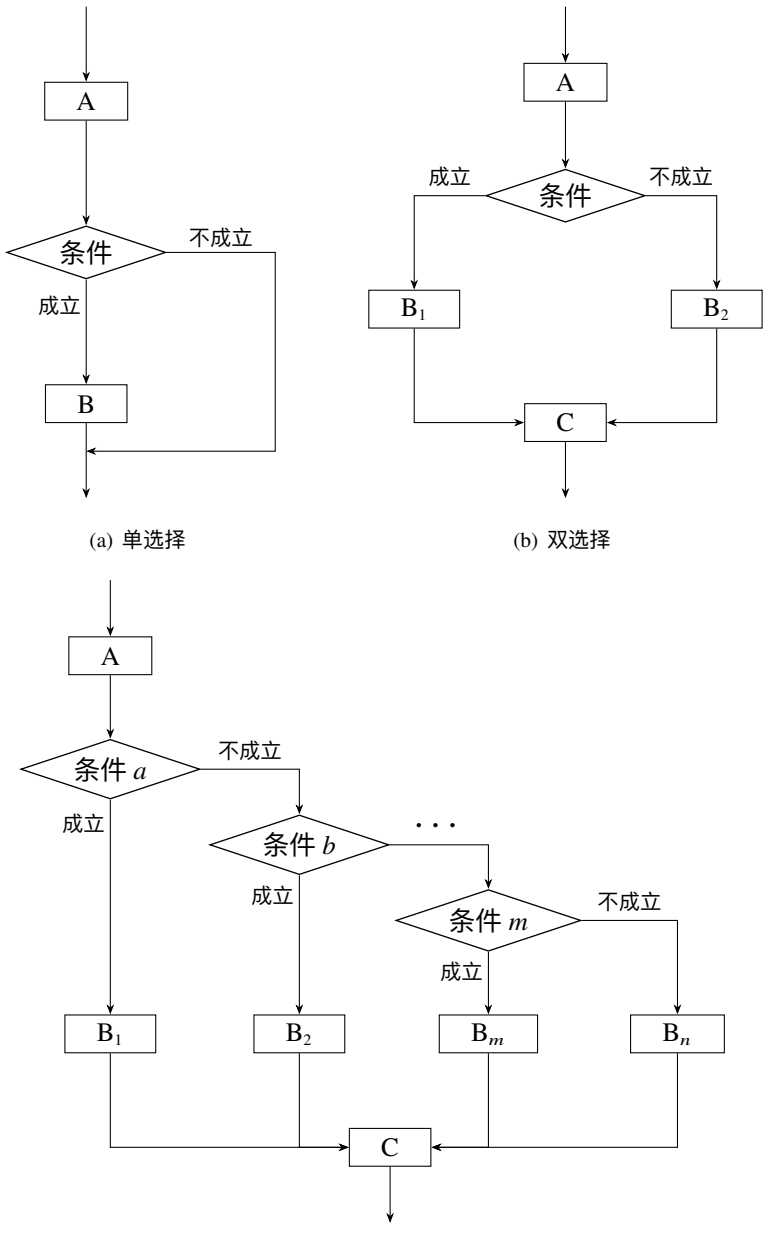
功能	函数	说明
输入数据	`A = input(提示信息)`	“提示信息”是一个字符串， 指输入数据时屏幕先显示的字符， 用于提示用户输入什么样的数据。
输出数据	`disp(输出项)`	输出项可以是任意的类型， 输出更紧凑，不留无意义的空行， 矩阵输出时不显示矩阵的变量名。
程序暂停	`pause(延迟秒数)`	直接使用`pause`， 则暂停程序直到按任意键继续。 程序执行中可按`Ctrl+C`强行暂停。

3.2.2 选择结构

选择结构又称为分支结构，它根据给定的条件是否成立，决定程序的运行路线，在不同条件下执行不同的操作。

1. if 语句

if 语句的常见三种结构如图3.1.



(a) 单选择 (b) 双选择

(c) 多重选择

图 3.1: if 选择结构流程图

对应的代码如下：

- 单选择 if 语句

```
1 if 条件
2   语句组
3 end
```

- 双选择 if 语句

```
if 条件
  语句组
else
  语句组
end
```

- 多重选择 if 语句

```
if 条件1
  语句组1
elseif 条件2
  语句组2
  ……
elseif 条件m
  语句组m
else
  语句组n
end
```

2. switch语句

switch语句的结构图如图3.2.

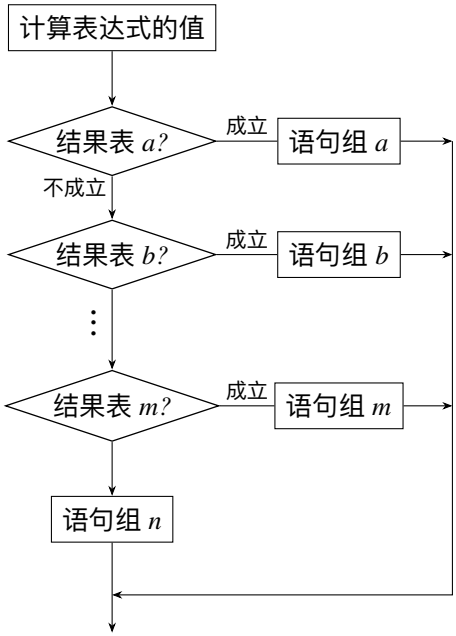


图 3.2: switch语句的结构图

代码如下:

```
switch  表达式
  case  结果表1
    语句组1
  case  结果表2
    语句组2
    ……
```

```
  case 结果表m
    语句组m
  otherwise
    语句组n
end
```

3.2.3 循环结构

循环结构的基本思想是重复，每次循环执行的语句相同，但语句中一些变量但值是变化但，而且当循环达到一定次数或满足一定条件时结束循环。循环结构的语句：for语句和while语句。

1. for语句

一般情况下，对于事先能确定循环次数的循环结构，使用for语句是比较方便的。一般格式如下

```
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:表达式2:表达式3将产生一个行向量，3 个表达式分别代表初值、步长和终值。其结构图如图3.3.

注意

for语句使用的注意事项

> for 语句针对向量的每一个元素执行一次循环体，循环的次数就是向量中元素的个数。
>
> for 语句中的行向量也可以用矩阵的形式定义。
>
> 可以在 for 循环体中修改循环变量的值，但回到循环开始时，就会自动设定为向量的下一个元素。
>
> for 语句中但 3 个表达式只在循环开始时计算一次，即循环向量一旦确定不再改变。
>
> 退出循环之后，循环变量的值就是向量最后的元素值。
>
> 当向量为空时，循环体一次也不执行。

例 3.1 一个 3 位整数各位数字的立方和等于该数本身则称该数为水仙花数。输出全部水仙花数。

```
shu = [ ];
for m = 100:999
  m1 = fix(m / 100);              % 求m的百位数字
  m2 = rem(fix(m / 10), 10);      % 求m的十位数字
  m3 = rem(m, 10)                 % 求m的个位数字
  if m == (m1*m1*m1 + m2*m2*m2 + m3*m3*m3)
    shu = [shu, m];               % 存入结果
  end
end
disp(shu)
```

结果如下

```
153   370   371   407
```

while语句

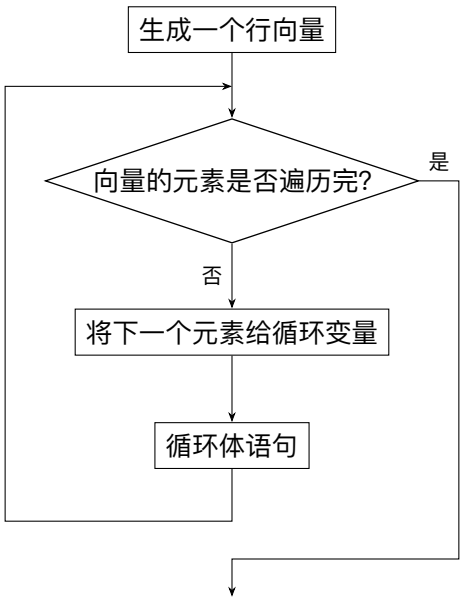


图 3.3: for语句的执行过程

while语句就是通过判断循环条件是否满足来决定是否继续的一种循环控制语句，也称为条件循环语句。它的特点是先判断循环条件，条件满足时执行循环。其一般格式如下：

```
while 条件
  循环体语句
end
```

结构图如图3.4.

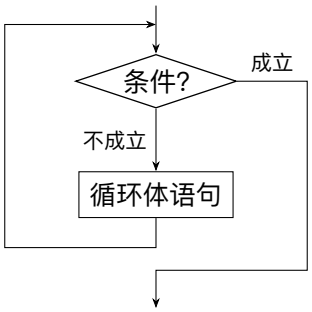


图 3.4: while语句的执行过程

3.3 函数文件

3.3.1 函数文件的基本结构

```
function 输出形参表 = 函数名(输入形参表)
% 此处显示有关此函数的摘要
% 此处显示详细说明
% 作者、修改日期、版本等
函数体语句
end
```

其中，

开头：以 function 开头的一行为引导行，表示定义一个函数。

函数名：函数名一般和文件名统一调用，方便 Matlab 调用。

return语句：若函数中插入了return语句，则执行到该语句就结束函数的执行，通常函数文件不写return语句，自动返回。

end语句：与return语句类似，一般函数文件中存在子函数需要用到。

3.3.2 函数调用

函数文件建立好以后，就可以调用函数，格式如下：

```
[输出实参表] = 函数名(输入实参表)
```

注意

在调用函数时，函数输入输出参数称为实际参数，简称实参。要注意的是，函数调用时各实参出现的顺序、个数，应与函数定义时的形参的顺序、个数相一致，否则会出错。

在 Matlab 中，函数可以嵌套调用，即一个函数可以调用其他函数，也可以调用自身。一个函数调用自身称为递归调用。

3.3.3 函数参数的可调性

在调用函数时，Matlab 用两个预定义的变量nargin和nargout分别记录调用该函数时的输入实参和输出实参的个数。只要在函数文件中包含这两个变量，就可以准确地知道该函数文件被调用时的输入/输出参数个数，从而决定函数如何进行处理。

3.3.4 全局变量与局部变量

在 Matlab 中，函数文件中的变量是局部的，与其他函数文件及 Matlab 工作空间相互隔离，即在一个函数文件中定义的变量不能被另一个函数文件引用。如果在若干函数中，都把某一变量定义为全局变量，那么这些函数将共用这个变量。全局变量的作用域是整个 Matlab 工作空间，即全程有效，所有的函数都可以对它进行存取和修改，因此，定义全局变量是函数间传递信息的一种手段。全局变量用global定义，格式如下：

```
global 变量名
```

注意

在函数文件中，全局变量的定义语句放在文件的前部。为了在工作空间中也使用全局变量，也要定义全局变量。

全局变量虽然带来了某些方便，但是也会破坏函数的封装性，降低了程序的可读性，因此，在结构程序化设计中，全局变量是不受欢迎的，尽量避免。

3.3.5 特殊形式的函数

1. 子函数

在 Matlab 中，可以在一个函数文件中同时定义多个函数，其中函数文件中出现的第一个函数称为**主函数**，其他函数称为**子函数**。

注意

> 子函数只能由同一函数文件中的函数调用，不能跨文件调用。
>
> 保存函数时，函数文件名和主函数名字相同，且主函数必须放在文件开头。
>
> 同一函数文件中主函数和子函数的工作区是彼此独立的，各个函数的信息传递可以通过输入输出参数和全局变量实现。

2. 内联函数

以字符串形式存在的函数表达式可以通过`inline`函数转化成内联函数。例如：

```
>> a = '(x + y)^2';
>> f = inline(a)
f =
     内联函数:
     f(x,y) = (x+y)^2

>> f(3,4)
ans =
     49
```

第 4 章　Matlab 绘图

4.1　二维图形

4.1.1　绘制二维曲线的基本函数

绘图函数格式如下

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

其中，

1. 数据的输入 x_i, y_i

   - 当输入的参数都为向量时，$x1, y1$，$x2, y2$，$\cdots$，xn, yn 分别组成一组向量对。
     - 每一组向量对的长度可以不同
     - 每一对向量绘制一条曲线
   - 当输入的参数有矩阵形式时，x, y 组成的矩阵对按对应列元素为横、纵坐标分别绘制曲线条数等于矩阵的列数。

2. 绘图选型

   选项的格式为

   ```
   '<颜色选项> <线型选项> <标记符号选项>'
   ```

   当选项省略时，线性一律用实线，自动循环使用当前坐标轴的属性，无数据点标记符号。

   - 线性选项

选项	线型	选项	线型
-	实线（默认值）	-.	点画线
:	虚线	--	双画线

   - 颜色选项

选项	颜色	选项	颜色
b	蓝色	m	品红色
g	绿色	y	黄色
r	红色	k	黑色
c	青色	w	白色

- 标记符号选项

选项	标记符号	选项	标记符号
.	点	v	朝下三角符号
o	圆圈	^	朝上三角符号
x	叉号	<	朝左三角符号
+	加号	>	朝右三角符号
*	星号	p	五角星
s	方块	h	六角星
d	菱形		

在 Matlab 中，如果需要绘制出具有不同纵坐标标度的两个图形，可以使用`plotyy`函数，其调用格式如下：

```
plotyy(x1,y1, x2,y2)
```

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 $x1, y1$ 曲线，右纵坐标用于 $x2, y2$ 曲线。

例 4.1 用不同线型和颜色在同一坐标系内绘制曲线 $y = 2e^{-0.5x}\sin(2\pi x)$ 及其包络线。

```
x = (0: pi/100 :3*pi)';
y1 = 2*exp(-0.5*x) * [1, -1];
y2 = 2*exp(-0.5*x) .* sin(2*pi*x);
x1 = (0:18)/2;
y3 = 2*exp(-0.5*x1) .* sin(2*pi*x1);
plot(x,y1,'k', x,y2,'b--', x1,y3,'rp');
axis([0,10,-2,2]);
xlabel('x');
ylabel('y');
```

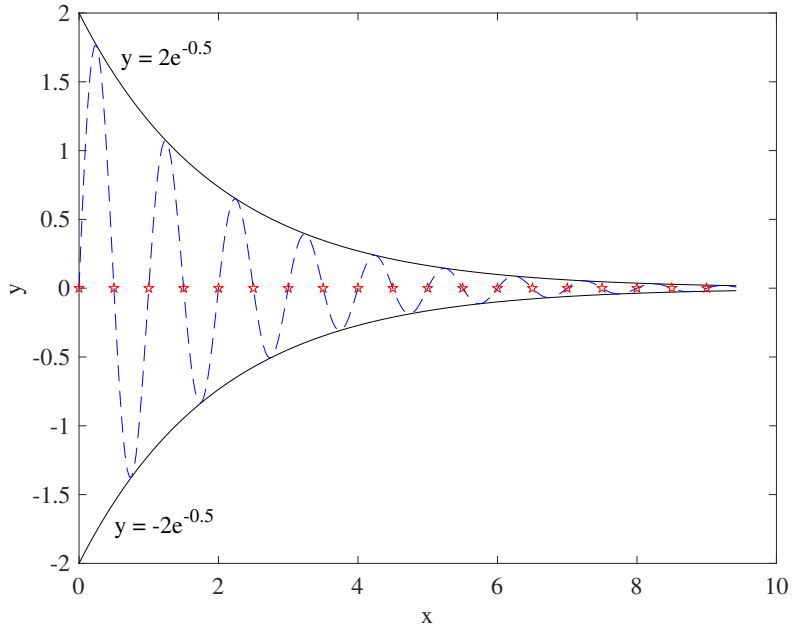


图 4.1: 例 **4.1**输出结果图

4.1.2 图形的辅助操作

常见的辅助操作如下表

功能	函数	说明
图形标注	title(图形名称)	图形的标题
	xlabel(x轴说明)	x 轴的标签
	ylabel(y轴说明)	y 轴的标签
	zlabel(z轴说明)	z 轴的标签
	legend(图例1, 图例2, ……)	曲线的图例
坐标轴控制	axis([xmin, xmax, ymin, ymax, zmin, zmax])	有几个特殊用法
	axis equal	坐标采用等长刻度
	axis square	产生正方形坐标系
	axis auto	默认设置
	axis on/off	显示/取消坐标轴
	grid on/off	打开/关闭网格线
	box on/off	加/不加边框线
图像窗口的控制	subplot(m, n, p)	划分 $m \times n$ 个绘图区，选定第 p 个区块绘制

例 4.2

```
x = 0 : 2*pi/60 : 2*pi;
y = sin(x); z = cos(x);
t = sin(x) ./ (cos(x) + eps); ct = cos(x) ./ (sin(x) + eps);
subplot(2, 2, 1);
plot(x, y-1);
title('sin(x) - 1');  axis([0, 2*pi, -2, 0]);
subplot(2, 1, 2);
plot(x, z-1);
title('cos(x) - 1');  axis([0, 2*pi, -2, 0]);
subplot(4, 4, 3);
plot(x, y);
title('sin(x)');  axis([0, 2*pi, -1, 1]);
subplot(4, 4, 4);
plot(x, z);
title('cos(x)');  axis([0, 2*pi, -1, 1]);
subplot(4, 4, 7);
plot(x, t);
title('tan(x)');  axis([0, 2*pi, -40, 40]);
subplot(4, 4, 8);
plot(x, ct);
title('cot(x)');  axis([0, 2*pi, -40, 40]);
```

输出结果如图4.2.

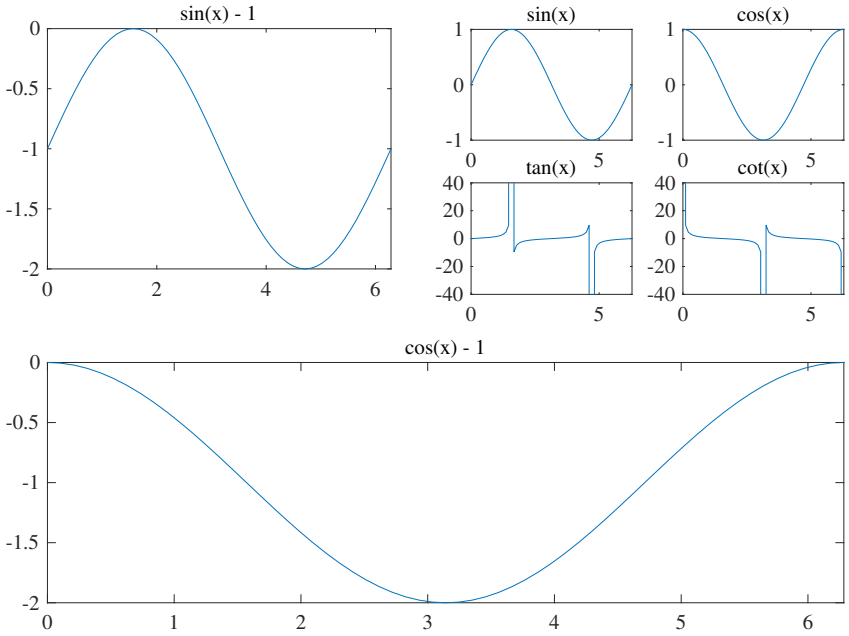


图 4.2: 例 **4.2**输出结果图

4.2 其他形式的二维图形

4.2.1 其他坐标系下的二维曲线图

功能	函数	说明
对数坐标系	`semilogx(x1,y1,选项1, x2,y2,选项2, ...)`	x 轴对数刻度，y 轴线性刻度
	`semilogy(x1,y1,选项1, x2,y2,选项2, ...)`	x 轴线性刻度，y 轴对数刻度
	`loglog(x1,y1,选项1, x2,y2,选项2, ...)`	x 轴对数刻度，y 轴对数刻度
极坐标系	`polar(theta, rho, 选项)`	theta 为极坐标极角，rho 为极坐标极径

注：选项与`plot`的选项用法一致。

例 4.3 绘制蝴蝶曲线 $\rho = \mathrm{e}^{\cos\theta} - 2\cos(4\theta) + \sin^5\dfrac{\theta}{12}$.

```
t = 0:pi/50: 20*pi;
r1 = exp(cos(t)) - 2*cos(4*t) + sin(t/12).^5;
r2 = exp(cos(t - pi/2)) - 2*cos(4*(t - pi/2)) + sin((t - pi/2)/12).^5;
subplot(1, 2, 1)
polar(t, r1)
subplot(1, 2, 2)
polar(t, r2)
```

输出结果如图4.3.

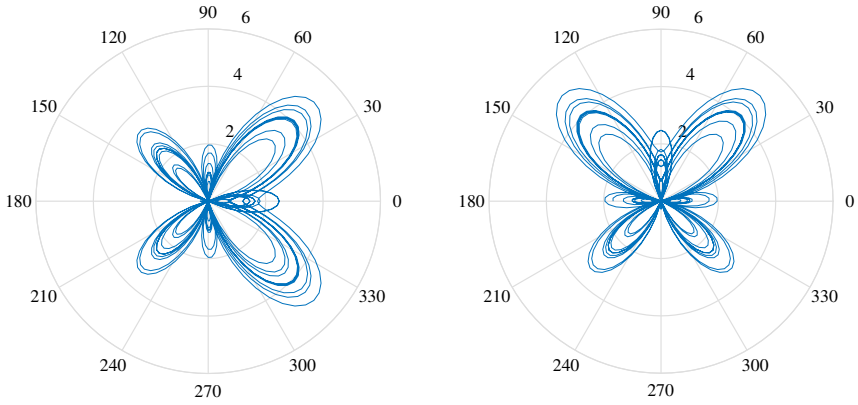


图 4.3: 例 **4.3**输出结果图

4.2.2 条形类图形

1. 条形图

   绘制二维条形图的函数为`bar`（垂直条形图）和`barh`（水平条形图），它们的调用格式是一致的。以`bar`为例

   - `bar(y)`
     - 若 y 为向量，则分别显示每个分量的高度，横坐标为 y 的下标
     - 若 y 为矩阵，则分别比较每一行的大小，横坐标为矩阵的行数
   - `bar(x, y, style)`
     - 当 y 是 $m \times n$ 的矩阵时，矩阵中每一行元素绘制在一组中，每组的横坐标为对应 x 的行元素
     - `style`指定条形的排列模式，可选的有`grouped`（簇状分组）和`stacked`（堆积分组），默认采用`grouped`
   - 例子

```
x = -1:1;
y = [1,2,3,4,5; 1,2,1,2,1; 5,4,3,2,1];
subplot(1,2,1); bar(x,y,'grouped');
title('Group'); axis([-3,3, 0,6]);
subplot(1,2,2); barh(x,y,'stacked')
title('Stack');
```

2. 直方图

   - `hist(y, x)`
     - 若 y 为向量，则将其最小值与最大值的区间等分，并统计每个区间元素的个数，然后以元素个数为高度绘制条形图
     - 若 y 为矩阵，将 y 的每一列作为一个向量，绘制每一列元素的直方图
     - 若 x 为标量，则统计区间均分成 x 个小区间
     - 若 x 为向量，则区间数为向量的长度，向量中的每一个数为各个区间的中心点
   - `rose(theta, x)`

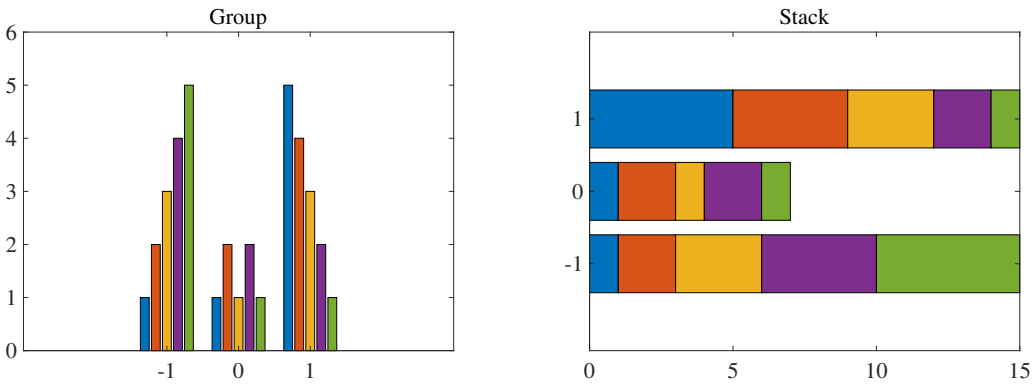


图 4.4: 条形图

- 向量theta用于确定每一区间的长度，每一区间的长度反映出在该区间的theta元素个数
- 若 x 为标量，则在 $[0, 2\pi]$ 区间内画出 x 个等角度的小扇形，默认值为 20
- 若 x 为向量，则 x 指定分组中心值，x 元素个数为数据分组数。

- 例子

```
y = randn(500,1);
subplot(2,2,1);
hist(y);  title('高斯分布直方图')
x = -4:0.1:4;
subplot(2,2,2);
hist(y, x); title('指定范围的高斯分布直方图')
subplot(2,1,2);
theta = y * pi;
rose(theta);  title('在极坐标下的直方图')
```

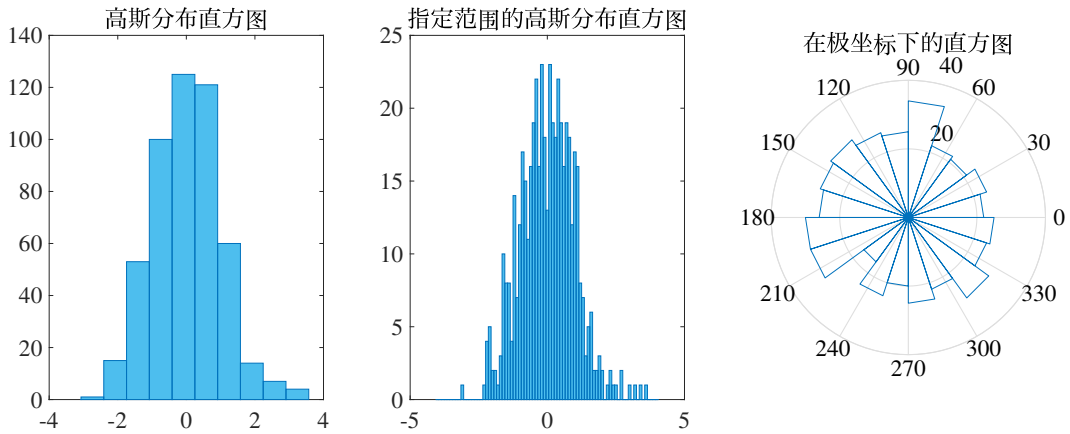


图 4.5: 直方图

4.2.3 面积类图形

1. 扇形统计图

- `pie(x, explode)`
  - x 为向量，x 的每个元素占有一个扇形，从饼图的正上方开始，按逆时针顺序绘制。
  - x 为矩阵，则将矩阵的所有元素按序号一个个绘制扇形
  - 若 x 的元素之和小于 1，则绘制的图形不是一个完整的圆
  - `explode`是与 x 同等大小的向量或矩阵，与`explode`的非零值对应的部分将从饼图中心分离出来
- 例子

```
pie([7,17,23,19,5], [0,0,0,0,1]);
title('pie map');
legend('A', 'A-', 'B', 'C', 'D');
```

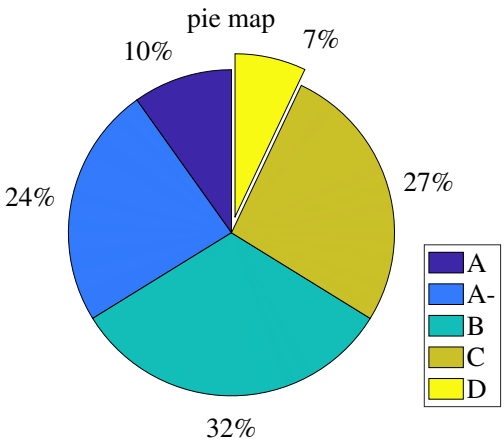


图 4.6: 饼图

2. 面积统计图

- `area(x, y)`
  - x, y 都是向量，与`plot(x, y)`一样，但将所得曲线下方填色
  - x 为向量，y 为矩阵，则矩阵但每一列与向量 x 成对绘图
- 例子

```
x = 1:2:9;
y = [1,3,5,2,6; 2,4,5,6,2; 5,4,7,2,2]
area(x, y);
grid on;
title('Area map');
```

3. 实心图

- `fill(x, y, color)`

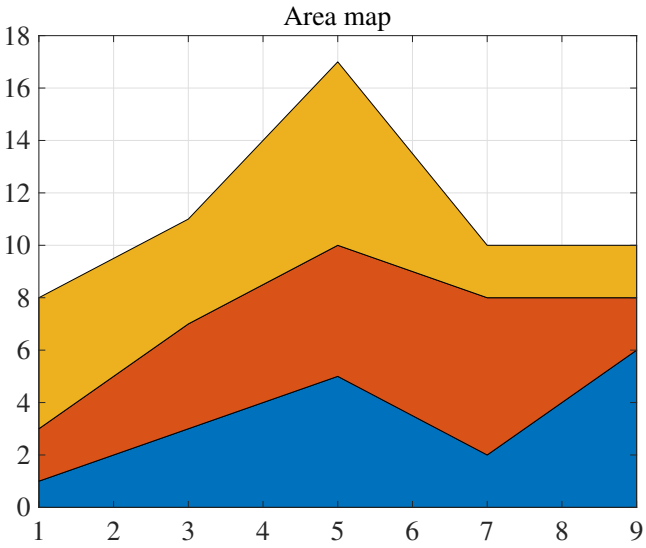


图 4.7: 面积图

– 按向量 x, y 下标增加的方向依次连接点 (x_i, y_i)，若最后首尾不封闭，那么将自动首尾相连，构成封闭多边形，然后内部涂色。

- 例子

```
t = 0: 2*pi/1000:2*pi;
x = sin(t);
y = cos(t);
fill(x,y,'k');
axis equal; axis([-1.5,1.5, -1.5,1.5]);
```

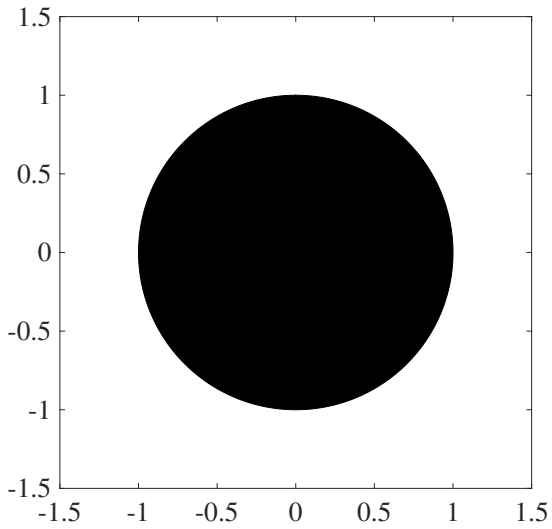


图 4.8: 实心图

4.2.4 散点类图形

- 散点图：`scatter(x, y, 'filled', color)`

- 阶梯图`stairs(x, y, 选项)`
- 杆图：`stem(x, y, 选项)`

其中，x, y 为大小相同的向量，用于绘制点。例子

```
x = 0:0.35:7;
y = 2*exp(-0.5*x);
subplot(1,3,1); scatter(x, y, 'g');
title('scatter'); axis([0,7, 0,2]);
subplot(1,3,2); stairs(x,y,'b');
title('stairs');  axis([0,7, 0,2]);
subplot(1,3,3); stem(x,y,'k');
title('stem');  axis([0,7, 0,2]);
```

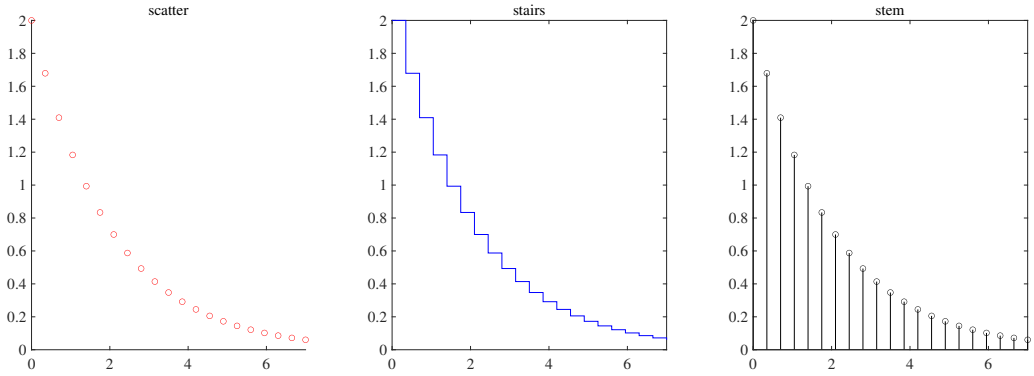


图 4.9: 散点图

4.2.5 矢量类图形

1. 罗盘图
   - `compass(x,y)`
     - x, y 为 n 个向量，显示 n 个箭头
     - 箭头的起点为原点，终点为 (x_i, y_i).
   - `compass(z)`
     - z 为 n 个复数向量，显示 n 个箭头
     - 箭头的起点为原点，终点为 $(\text{real}(z), \text{image}(z))$
2. 羽毛图
   - `feather(x,y)`
     - 绘制由 x, y 所确定的向量
   - `feather(z)`
     - 绘制由复数 z 所确定的向量，等价于`feather(real(z), imag(z))`.
3. 箭头图

- `quiver(x, y, u, v)`
  - (x, y) 为矢量起点，若省略，则在平面上均匀选取若干个点作为起点
  - (u, v) 为矢量终点

例子：

```
x= -pi:pi/8:pi;
y = sin(x);
subplot(2, 2, 1); compass(x, y);
title('罗盘图');
subplot(2, 2, 2); feather(x, y)
title('羽毛图');
subplot(2, 1, 2); quiver(x, y);
title('箭头图');
```

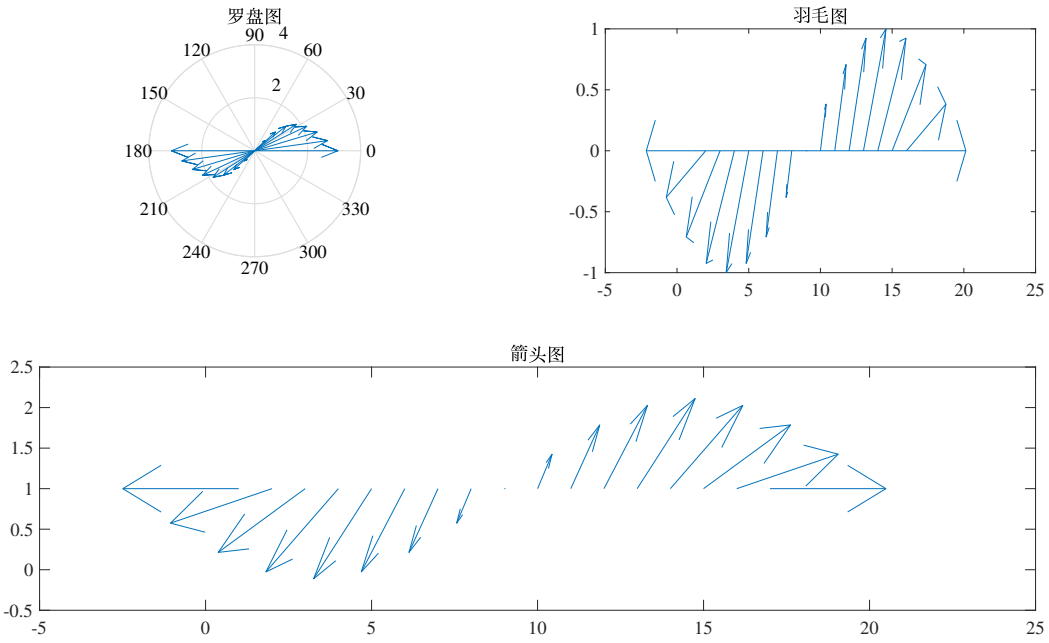


图 4.10: 3 种二维矢量类图形

4.3 三维图形

4.3.1 绘制三维曲线的基本函数

绘制三维图形的函数为

```
plot3(x1,y1,z1,选项1, x2,y2,z2,选项2, ..., xn,yn,选项n)
```

其调用格式与`plot`函数基本一致。

- 当 x, y, z 是同长度的向量时，则 x, y, z 构成一条三维曲线
- 当 x, y, z 是同型矩阵时，则以 x, y, z 对应的列向量绘制三维曲线，曲线条数等于矩阵列数

例 4.4 绘制空间曲线

$$\begin{cases} x^2+y^2+z^2=64 \\ y+z=0 \end{cases} \Longrightarrow \begin{cases} x=8\cos t \\ y=4\sqrt{2}\sin t \\ z=-4\sqrt{2}\sin t \end{cases}, \quad 0 \le t \le 2\pi$$

```
t = 0:pi/50:2*pi;
x = 8*cos(t);
y = 4*sqrt(2)*sin(t);
z = -4*sqrt(2)*sin(t);
plot3(x,y,z,'p');
title('Line in 3-D Space');
text(0,0,0,'origin');
xlabel('X'); ylabel('Y'); zlabel('Z'); grid;
```

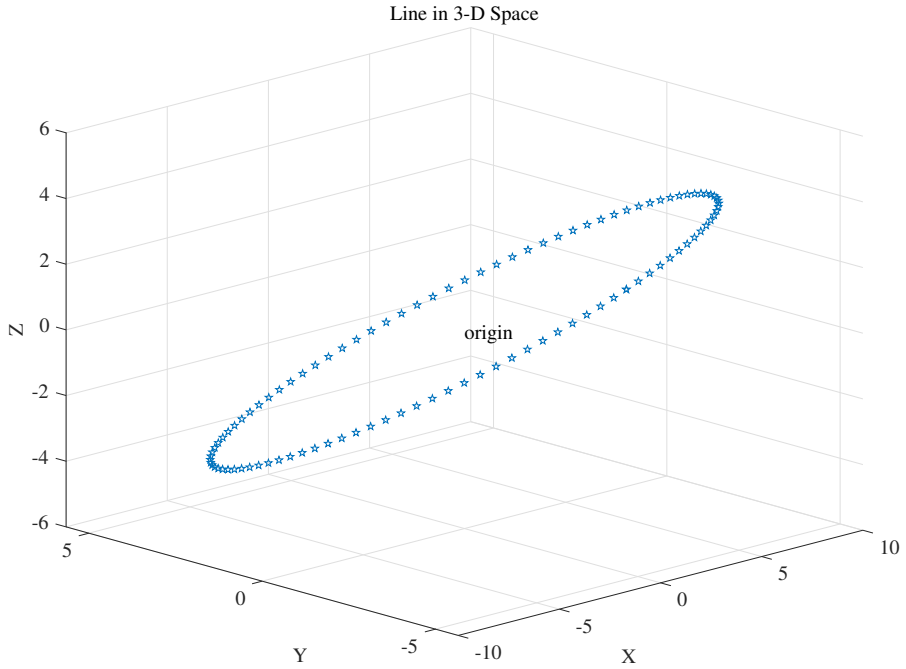


图 4.11: 三维曲线

4.3.2 三维曲面

1. **平面网格坐标矩阵的生成**

   利用`meshgrid`函数生成，调用格式如下

```
x = a:dx:b;
y = c:dy:d;
[X,Y] = meshgrid(x, y);
```

   当 $x = y$ 时，可写作`meshgrid(x)`.

2. **绘制三维曲面的函数**

   - `mesh(x, y, c)`

- `surf(x, y, c)`

一般情况下，x, y, z 是同型矩阵。

- x, y 是网格坐标矩阵
- z 是网格点上的高度矩阵
- c 称为色标矩阵，用于指定曲面的颜色。当 c 省略时，默认 c = z，即颜色的设定与高度成正比，这样就可以得到层次分明的三维图形
- 当 x, y 省略时，矩阵以 z 的行下标为 x 坐标，列下标为 y 坐标绘制图形

```
x = 0:0.1:2*pi;
[x, y] = meshgrid(x);
z = sin(y).*cos(x);
mesh(x,y,z);
xlabel('x'),ylabel('y'),zlabel('z');
title('mesh');
surf(x,y,z);
xlabel('x'),ylabel('y'),zlabel('z');
title('surf');
plot3(x,y,z);
xlabel('x'),ylabel('y'),zlabel('z');
title('plot3');
```

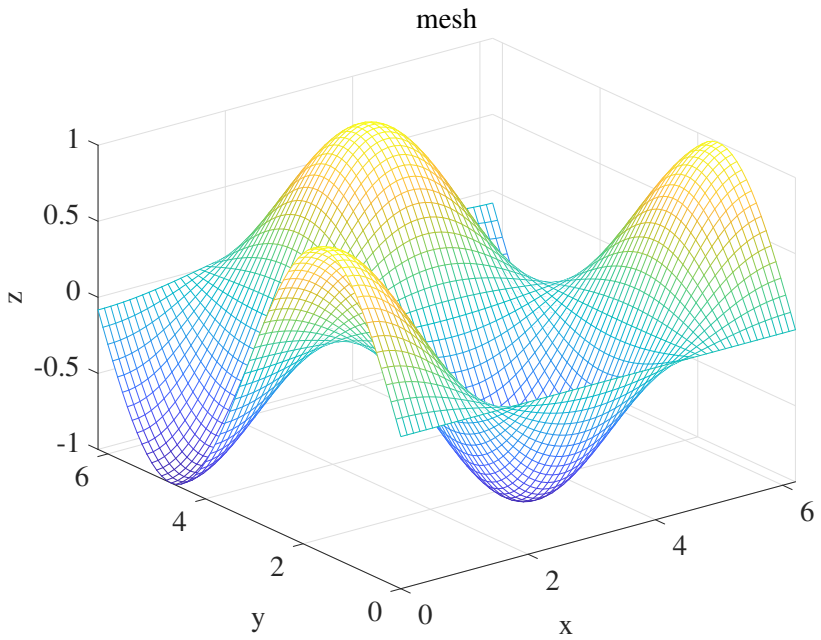


图 4.12: 三维网格图

4.3.3 其他三维图形

特殊的三维图形与特殊二维图形的使用类似，具体使用说明见4.2.2 – 4.2.5，常见的特殊三维图形见下表4.1. 例如：

```
subplot(2, 2, 1); bar3(magic(4)); title('bar3');
subplot(2, 2, 2); pie3([2347, 1827, 2043, 3025]); title('pie 3');
```

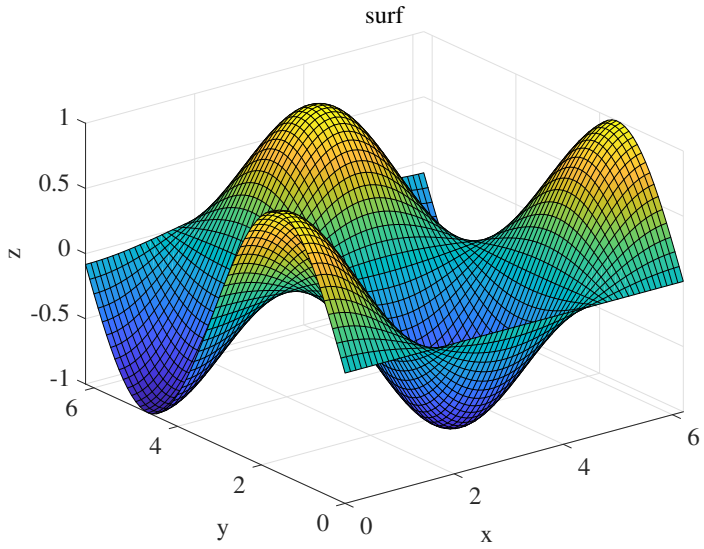


图 4.13: 三维曲面图

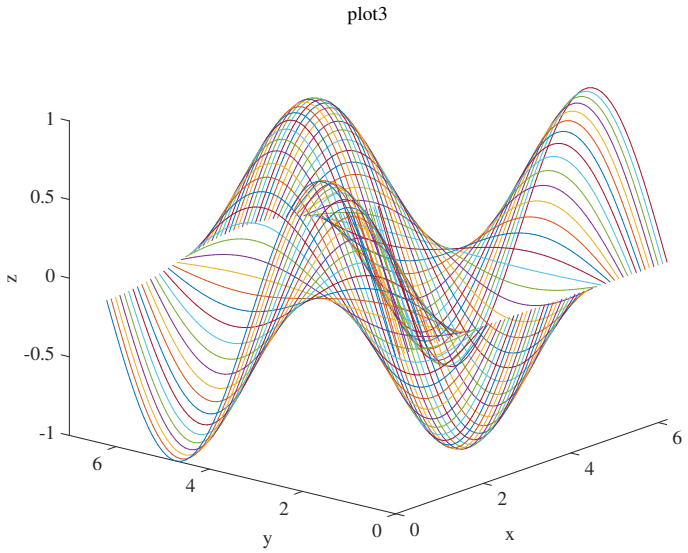


图 4.14: plot3三维图

类型	函数
三维条形图	bar3(y), bar3h(y) bar3(x, y), bar3h(x, y)
三维饼图	pie3(x, explode)
三维实心图	fill3(x, y ,z, c)
三维散点图	stem3(z), steam3(x, y, z)
三维箭头图	quiver3(x, y, z, u, v, w)
瀑布图	waterfall(x, y, z)
二维等高线图	contour(x, y, 等级数, 选项)
三维等高线图	contour3(x, y, z, 等级数, 选项)

表 4.1: 常见的特殊三维图函数

```
3 a = rand(3, 5); b = rand(3, 5); c = rand(3, 5);
4 subplot(2, 2, 3); fill3(a, b, c, 'y'); title('fill3');
5 y = 2*sin(0:pi/10:2*pi);
6 subplot(2, 2, 4); stem3(y); title('stem3');
```

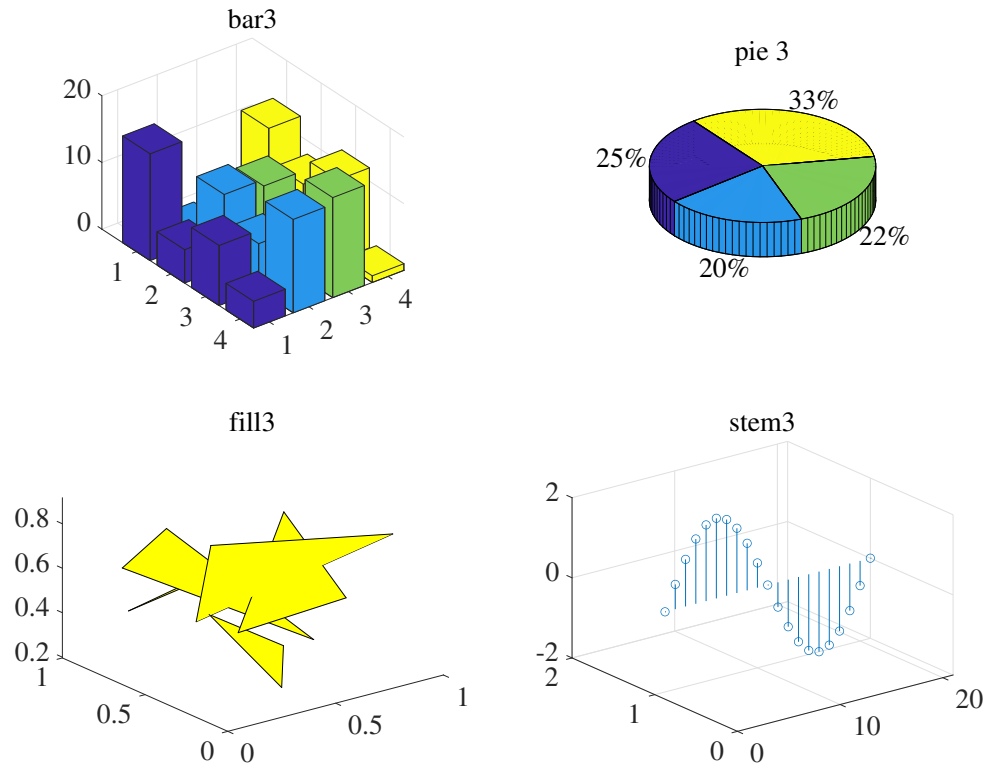


图 4.15: 特殊三维图

4.4 隐函数绘图

4.4.1 隐函数二维绘图

隐函数用ezplot进行绘图，它有各种变形如下

- 隐函数 $f(x) = 0$
  - `ezplot(f)`
    - 在默认区间 $-2\pi < x < 2\pi$ 绘制 $y = f(x)$ 的图形
    - f 可以是函数文件名或函数表达式组成的字符串，也可以是匿名函数表达式或函数名
  - `ezplot(f, [a,b])`
    - 在区间 $a < x < b\pi$ 绘制 $y = f(x)$ 的图形
- 隐函数 $f(x, y) = 0$
  - `ezplot(f)`
    - 在默认区间 $-2\pi < x < 2\pi, -2\pi < y < 2\pi$ 绘制 $f(x, y) = 0$ 的图形

  - `ezplot(f, [a, b])`
    - 在区间 $a < x < b, a < y < b$ 绘制 $f(x, y) = 0$ 的图形
  - `explot(f, [xmin,xmax, ymin, ymax])`
    - 在区间 $x\min < x < x\max$ 和 $y\min < y < y\max$ 绘制 $f(x, y) = 0$ 的图形

- 参数方程 $x = x(t), y = y(t)$
  - `ezplot(x, y)`
    - 在默认区间 $-2\pi < t < 2\pi$ 绘制 $x = x(t), y = y(t)$ 的图形
  - `ezplot(x, y, [xmin ,ymin])`
    - 在区间 $t\min < t < t\max$ 绘制 $x = x(t), y = y(t)$ 的图形

例子

```
subplot(2,2,1);   ezplot('x^2+y^2-9'); axis equal;
subplot(2,2,2);   ezplot(@(x, y) x^3+y^3-5.*x.*y + 1/5);
subplot(2,2,3);   ezplot('cos(tan(pi*x))', [0, 1]);
subplot(2,2,4);   ezplot('8*cos(t)', '4*sqrt(2)*sin(t)', [0, 2*pi]);
```

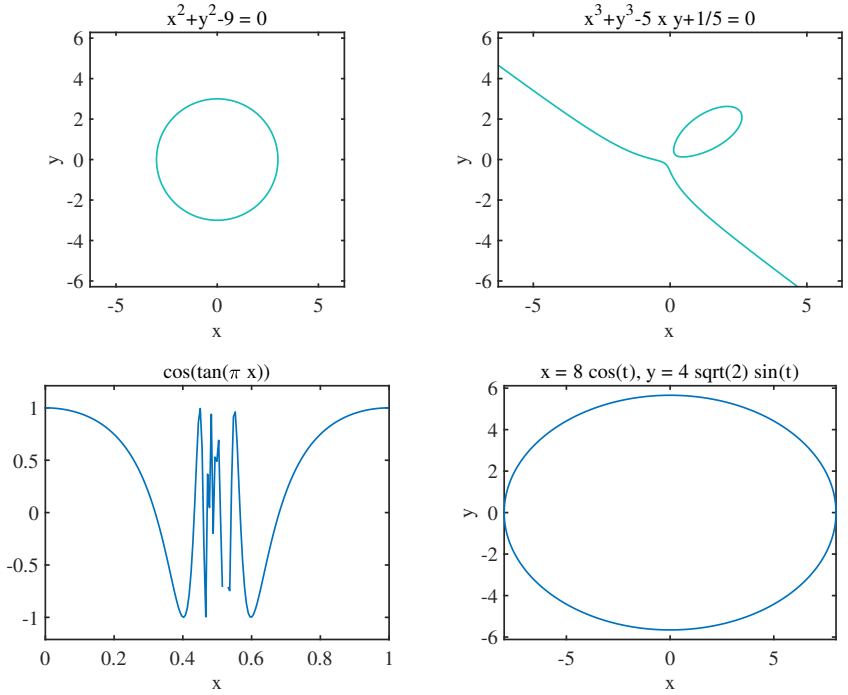


图 4.16: 二维隐函数

4.4.2 三维隐函数绘图

隐函数三维绘图的函数有`ezcontour`, `ezcontourf`, `ezmesh`, `ezmeshc`, `ezplot3`, `ezpolar`, `ezsurf`, `ezsurfc`, 它们的调用格式基本相同，需要时直接查阅帮助信息。下面对`ezsurf`进行简介

- `ezsurf(f)` 绘制曲线 $z = f(x, y)$, f 的表达与`ezplot`函数相同，默认区间 $-2\pi < x < 2\pi, -2\pi < y < 2\pi$
- `ezsurf(f, [xmin,xmax, ymin,ymax])`, `ezsurf(f, [min,max])` 在指定的区间上绘制曲线 $z = f(x, y)$

- `ezsurf(x, y, z)` 在默认区域 $-2\pi < s < 2\pi, -2\pi < s < 2\pi$ 上绘制参数方程 $x = x(s,t), y = y(s,t), z = z(s,t)$ 所确定的曲面
- `ezsurf(x, y, z, [smin,smax, tmin,tmax])`, `ezsurf(x, y, z, [min, max])` 在指定的区间上绘制参数方程曲面

例 4.5 绘制下列曲面

$$\begin{cases} x = \mathrm{e}^{-s}\cos t \\ y = \mathrm{e}^{-s}\sin t \\ z = t \end{cases}, \quad 0 \le s \le 8, 0 \le t \le 5\pi$$

```
1 >> ezsurf('exp(-s).*cos(t)', 'exp(-s).*sin(t)', 't', [0,8, 0,5*pi])
```

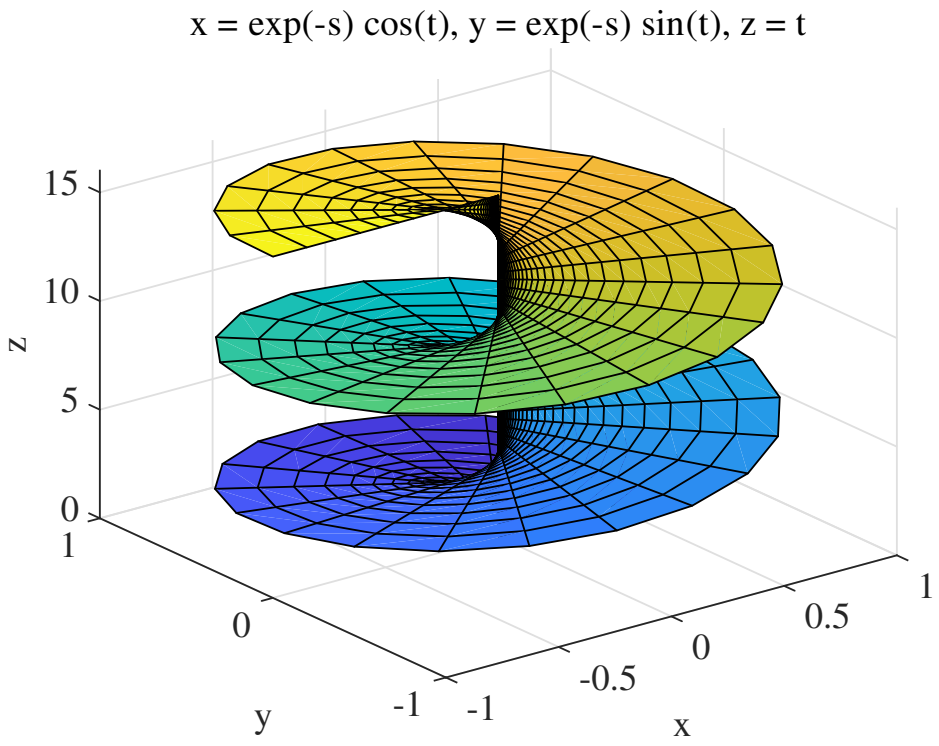


图 4.17: 三维参数曲面图

第 5 章　Matlab 数据分析与多项式计算

5.1　数据统计分析

5.1.1　最大值与最小值

以最大值函数为例子，介绍函数的用法，最小值函数的调用完全一致。

函数	说明
`max(A)`	返回一个行向量，向量的第 i 个元素是矩阵 $\boldsymbol{A}$ 第 i 列的最大值
`max(A,[ ],2)`	返回一个列向量，向量的第 i 个元素是矩阵 $\boldsymbol{A}$ 第 i 行的最大值
`[Y,U] = max(A)`	$\boldsymbol{Y}$ 行向量记录矩阵 $\boldsymbol{A}$ 每列的最大值 $\boldsymbol{U}$ 矩阵记录每列最大值的行号
`U = max(A,B)`	$\boldsymbol{A},\boldsymbol{B}$ 是同型的向量或矩阵 $\boldsymbol{U}$ 的每个元素等于 $\boldsymbol{A},\boldsymbol{B}$ 对应元素的最大值
`U = max(A,n)`	n 是一个标量，$\boldsymbol{U}$ 的每个元素等于 $\boldsymbol{A}$ 对应元素与 n 中的最大值

5.1.2　求和与求积

求和函数用`sum`，求积函数用`prod`. 两者调用方式完全一致，以`sum`为例：

函数	说明
`sum(X)`	返回向量 $\boldsymbol{X}$ 的元素之和
`sum(A)`	返回一个行向量，向量的第 i 个元素是矩阵 $\boldsymbol{A}$ 第 i 列元素之和
`sum(A,2)`	返回一个列向量，向量的第 i 个元素是矩阵 $\boldsymbol{A}$ 第 i 行元素之和

5.1.3　平均值和中值

平均值函数用`mean`，中值函数用`median`. 两者调用方式完全一致，以`mean`为例：

函数	说明
`mean(X)`	返回向量 $\boldsymbol{X}$ 的平均值
`mean(A)`	返回一个行向量，向量的第 i 个元素是矩阵 $\boldsymbol{A}$ 第 i 列元素的平均值
`mean(A,2)`	返回一个列向量，向量的第 i 个元素是矩阵 $\boldsymbol{A}$ 第 i 行元素的平均值

5.1.4 累加和与累乘积

设 $\boldsymbol{U}=(u_1,u_2,\cdots,u_n)$ 是一个向量，$\boldsymbol{V},\boldsymbol{W}$ 是与 $\boldsymbol{U}$ 等长的另外两个向量，并且

$$\boldsymbol{V}=\left(\sum_{i=1}^{1}u_i,\sum_{i=1}^{2}u_i,\cdots,\sum_{i=1}^{n}u_i\right) \tag{5.1}$$

$$\boldsymbol{W}=\left(\prod_{i=1}^{1}u_i,\prod_{i=1}^{2}u_i,\cdots,\prod_{i=1}^{n}u_i\right) \tag{5.2}$$

称 $\boldsymbol{V}$ 为 $\boldsymbol{U}$ 的累加和向量，$\boldsymbol{W}$ 为 $\boldsymbol{U}$ 的累乘积向量。求矩阵元素的累加和的函数为`cumsum`，累乘积的函数为`cumprod`. 两者调用方式完全一致，以`cumsum`为例：

函数	说明
`cumsum(X)`	返回向量 $\boldsymbol{X}$ 的累加和向量
`cumsum(A)`	返回一个矩阵，其第 i 列是矩阵 $\boldsymbol{A}$ 第 i 列元素的累加和
`cumsum(A,2)`	返回一个矩阵，其第 i 行是矩阵 $\boldsymbol{A}$ 第 i 行元素的累加和

5.1.5 标准差与相关系数

标准差的计算公式

$$S_1=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\overline{x})^2} \tag{5.3}$$

$$S_2=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\overline{x})^2} \tag{5.4}$$

其中

$$\overline{x}=\frac{1}{n}\sum_{i=1}^{n}x_i$$

相关系数的计算公式为

$$r=\frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2\sum_{i=1}^{n}(y_i-\overline{y})^2}} \tag{5.5}$$

功能	函数	说明
标准差	`std(A,flag,dim)`	dim = 1　求各列元素的标准差 dim = 2　求各行元素的标准差 flag = 0　按 S_1 计算标准差 flag = 1　按 S_2 计算标准差
相关系数	`corrcoef(X, Y)`	返回一个 2×2 矩阵：$\begin{bmatrix} r(\boldsymbol{X}) & r(\boldsymbol{X},\boldsymbol{Y}) \\ r(\boldsymbol{Y},\boldsymbol{X}) & r(\boldsymbol{Y}) \end{bmatrix}$

相关系数是反映两组数据序列之间的相互关系的指标，类似的指标还有协方差，计算公式为

$$c = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}) \tag{5.6}$$

Matlab 的函数为cov，其调用格式与corrcoef函数类似。

5.1.6 排序

sort函数对矩阵 ***A*** 对各列或各行重新排序，其调用格式为

```
[Y, I] = sort(A, dim, mode)
```

其中

- ***Y*** 为排序后对矩阵
- ***I*** 记录 ***Y*** 中对元素在 A 中对位置
- dim指明对 ***A*** 进行列/行排序
  - `dim = 1` 列排序
  - `dim = 2` 行排序
- mode表示升序/降序
  - `mode = 'ascend'` 升序（默认）
  - `mode = 'descend'` 降序

5.2 多项式计算

n 次多项式

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots a_1 x + a_0 \tag{5.7}$$

在 Matlab 中，$P(x)$ 表达为向量形式：

$$[a_n, a_{n-1}, a_{n-2}, \cdots, a_1, a_0] \tag{5.8}$$

5.2.1 多项式的四则运算

多项式的四则运算等价于多项式系数P1，P2之间的运算，详细见下表

功能	函数	说明
多项式加法	`P = P1 + P2`	直接相加即可
多项式减法	`P = P1 - P2`	直接相减即可
多项式乘法	`P = conv(P1, P2)`	——
多项式除法	`[Q, r] = deconv(P1, P2)`	Q 返回商值，r 返回余式

5.2.2 多项式的导函数

求多项式的导函数用polyder函数，其调用格式为

(1) `p = polyder(P)` 求多项式 P 的导函数

(2) `p = polyder(P, Q)` 求 P·Q 的导函数

(3) `[p, q] = polyder(P, Q)` 求 P/Q 的导函数，导函数分子存入 p，分母存入 q

5.2.3 多项式求值

1. **代数多项式求值**

```
polval(P, x)
```

若 x 为一数值，则求多项式在该点的值；若 x 为向量或矩阵，则对向量或矩阵中的每个元素求其多项式的值。

2. **矩阵多项式求值**

```
polyvalm(P, A)
```

其中 A 为方阵。

5.2.4 多项式求根

求根函数和已知根 x 构造多项式的函数：

```
x = roots(P)
P = poly(x)
```

5.3 数据插值

功能	函数	说明
一维插值	`Y1 = interp1(X, Y, X1, method)`	$\boldsymbol{X},\boldsymbol{Y}$ 为等长向量，分别为采样点和采样值
二维插值	`Y2 = interp2(X, Y, Z, X1, X2, method)`	X, Y 是采样点，Z 是采样值

method 有四种，见下表

方法	说明
`'linear'`	线性插值（默认），两点连线，不外插
`'nearest'`	最近点插值，插值点优先选择较近的数据点插值，不外插
`'pchip'`	分段 3 次埃米尔插值，不适用于二维插值
`'spline'`	3 次样条插值，每个分段内构造一个 3 次多项式，可外插

5.4 曲线拟合

最小二乘实现曲线拟合，使用polyfit函数：

```
P = polyfit(X, Y, m)
[P, S] = polyfit(X, Y, m)
[P, S, mu] = polyfit(X, Y, m)
```

函数根据采样点 $\boldsymbol{X}$ 和采样点函数 $\boldsymbol{Y}$，产生一个 m 次多项式 $\boldsymbol{P}$ 及其在采样点的误差向量 $\boldsymbol{S}$。其中

- $\boldsymbol{X},\boldsymbol{Y}$ 等长向量
- $\boldsymbol{P}$ 长度为 $m+1$ 的向量，其为多项式系数
- mu 二元向量，mu(1) 是 mean(X)，而 mu(2) 是 std(X)