

附录一 Matlab 入门

§ 1 概论

常用的数学软件有 Maple, Mathematica, Matlab 等; 常用的大型统计软件有 SAS, SPSS 等。下面我们简要地介绍一些 Matlab 的功能, 应用范围及发展史。

Matlab 有五大通用功能: 数值计算功能(Numeric), 符号运算功能(Symbolic) (当要求 Matlab 进行符号运算时,它就请求 Maple 计算并将结果返回到 Matlab 命令窗口), 数据可视化功能(Graphic), 数据图形文字统一处理功能(Notebook)和建模仿真可视化功能(Simulink)。

Matlab 在线性代数, 矩阵分析, 数值及优化, 数理统计和随机信号分析, 电路与系统, 系统动力学, 信号和图像处理, 控制理论分析和系统设计, 过程控制, 建模和仿真, 通信系统, 财政金融的众多领域的理论研究和工程设计中得到了广泛应用。

MATLAB 是 1984 年由美国 Mathworks 公司推向市场。该软件有三大特点: 一是功能强大; 二是界面友善, 语言自然; 三是开放性强(仅 Mathworks 公司就推出了 30 多个应用工具箱)。Matlab 的版本目前已经发展到 Matlab7.4。

§ 2 Matlab 简介

1. 指令行的编辑

启动 Matlab 后, 就可以利用 Matlab 工作。由于 Matlab 是一种交互式语言, 随时输入指令, 即时给出运算结果是它的工作方式。

```
2*sin(0.3*pi)/(1+sqrt(5)) ↵
```

ans=0.5000 (ans 是一个保留的 Matlab 字符串, 它表示上面一个式子的返回结果, 用于结果的缺省变量名。)

2. 入门演示

```
intro ↵ demo ↵
```

3. 帮助

- ① help ↵ %帮助总揽
- help elfun ↵ %关于基本函数的帮助信息
- help exp ↵ %指数函数 exp 的详细信息

② lookfor 指令

当要查找具有某种功能但又不知道准确名字的指令时, help 的能力就不够了, lookfor 可以根据用户提供的完整或不完整的关键词, 去搜索出一组与之相关的指令。

```
lookfor integral ↵ %查找有关积分的指令
lookfor fourier ↵ %查找能进行傅里叶变换的指令
```

③ 超文本格式的帮助用户

在 Matlab 中, 关于一个函数的帮助信息可以用 doc 命令以超文本的方式给出, 如:

```
doc ↵
doc doc ↵
doc eig ↵ %eig 求矩阵的特征值和特征向量
```

④ pdf 帮助文件

4. 简单的矩阵输入

① 要直接输入矩阵时, 矩阵元素用空格或逗号分隔; 矩阵行用分号“;”隔离, 整个矩阵放在方括号“[]”里。

```
A=[1,2,3;4,5,6;7,8,9] ↵
```

说明：指令执行后，矩阵 **A** 被保存在 Matlab 的工作间（workspace）中，以备后用。如果用户不用 clear 指令清除它，或对它进行重新赋值，那么该矩阵会一直保存在工作间中，直到本 Matlab 指令窗关闭为止。

② 矩阵的分行输入

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

5. 语句与变量

① Matlab 语句有两种最常见的形式

i) 表达式

ii) 变量=表达式

例1 表达式的计算结果

```
1996/18 ↵
ans=110.8889
```

例2 计算结果的赋值

```
s=1-1/2+1/3-1/4+1/5-1/6+...
1/7-1/8; ↵
```

说明：三个小黑点是“连行号”，分号“;”作用是：指令执行结果将不显示在屏幕上，但变量 **s** 将驻留在内存中。若用户想看 **s** 的值，可键入以下命令：

```
s ↵
(s=
    0.6345)
```

② 特殊变量

ans 用于结果的缺省变量名

pi 圆周率

eps 计算机的最小数

flops 浮点运算次数

inf 无穷大 如 1/0

NaN 不定量 如 0/0

i (j) $i=j=\sqrt{-1}$

nargin 所用函数的输入变量数目

nargout 所用函数的输出变量数目

realmin 最小可用正实数

realmax 最大可用正实数

6. 数据结构：向量、矩阵、结构数组和细胞数组

① 向量的共轭转置

```
z=[1+j,2+pi*i,-sqrt(-1)]'
z =
```

```
1.0000 - 1.0000i
2.0000 - 3.1416i
0 + 1.0000i
```

得到其共轭转置向量。

z.'（非共轭转置向量）

②产生一个行向量

`t=[0:0.1:10]` %产生从 0 到 10 的行向量, 元素之间间隔为 0.1

`t=linspace(n1,n2,n)`

%产生 n1 和 n2 之间线性均匀分布的 n 个数 (缺省 n 时, 产生 100 个数)

`t=logspace(n1,n2,n)` (缺省 n 时, 产生 50 个点)

③ `who`, `whos`, `size` 和 `length` 是对提供工作空间变量信息很有用处的四个命令。

`who` 执行该命令可列出工作空间的所有变量;

`whos` 显示所有的变量, 变量的元素个数和所占的字节数等;

`size(a)` 执行该命令可以得到矩阵 a 的行数与列数;

`length(a)` 执行该命令后, 屏幕上显示出向量 a 的长度。如果 a 是矩阵, 则显示参数为行数列数中的最大者。

④ 矩阵的标号

`A(m,n)` 表示矩阵 A 的第 m 行, 第 n 列的元素;

`A(1:2,1:3)` 表示矩阵 A 的从第一行到第二行, 从第一列到第三列的所有元素;

`A(:)` 可以得到一个长向量, 该向量的元素是按列一一叠加在一起的。例如

`a=[1 2;3 4]; a(:) ↴`

`ans=`

1

3

2

4

矩阵的下标也可以是向量。例如

`b=a(x,y) ↴`; 可以得到一个矩阵 b。a 的行标来自向量 x, 而列标来自向量 y。

例如, 矩阵 a 有 n 列, 那末

`b=a(:,n:-1:1)`

将得到矩阵 b, 它等于矩阵 a 按列的逆序排列。

⑤ 特殊矩阵

i) 单位矩阵

`eye(m)`, `eye(size(a))` 可以得到与矩阵 a 同样大小的单位矩阵, `eye(m,n)` 可得到一个可允许的最大单位矩阵而其余处补 0。

ii) 所有元素为 1 的矩阵

`ones(n)`, `ones(size(a))`, `ones(m,n)`

iii) 所有元素为 0 的矩阵

`zeros(n)`, `zeros(m,n)`

iv) 空矩阵是一个特殊矩阵, 这在线性代数中是不存在的。

例如: `q=[]`

矩阵 q 在工作空间之中, 但它的大小为零。通过空矩阵的办法可以删去行与列。例如

`a=rand(5); a(:,1:3)=[]`

第一条指令执行后得到一个 5×5 的矩阵; 第二条指令将矩阵 a 的前三列删除。

v) 对角矩阵

当 v 是向量时, `diag(v)` 得到以 v 的元素为对角线上元素的对角矩阵;

当 v 是矩阵时, `diag(v)` 得到一个列向量, 其元素为矩阵 v 对角线上的元素, `diag(v,1)` 得到矩阵 v 对角线上移一行的元素组成的列向量, `diag(v,-1)` 得到矩阵 v 对角线下移一行的元素组成的列向量。

⑥ 字符串要用单引号。例如：

```
disp('text string') % disp 显示命令
```

还有几个字符串命令可以作为文字说明和绘图标题说明等，如 num2str, int2str, fprintf 和 sprintf。同样，可以借助于 help 命令了解它们的具体用法。

⑦ 结构数组

有时需要将不同的数据类型组合成一个整体，以便于引用。这些组合在一个整体中的数据是相互联系的。例如，一个学生的学号、姓名、性别、年龄、成绩、家庭地址等项都是和该学生有联系的。

下面简单介绍结构体的定义与引用。

i) 结构数组的定义

定义结构数组可以采用两种方法：用赋值语句定义和用函数 struct 定义。

用赋值语句定义结构时，只要给出结构的属性赋值，Matlab 就会自动把该属性增加到结构中，赋值时，结构名和属性名用“.”分开。例如，下面三条语句将定义一个 1×1 的结构数组，结构名为 student，有三个属性：name、num、test。该结构数组只有一个元素，在命令窗口中键入结构名 student，将显示该元素所有属性的属性值的特性。

```
student.name='John Doe';
student.num=123456;
student.test=[79 75 73;80 78 79;90 85 80];
再键入以下三行可给该结构数组增加一个元素。
student(2).name='Ann Lane';
student(2).num=123422;
student(2).test=[70 76 73;80 99 79;90 85 80;80 85 86];
```

现在结构数组 student 的维数为 1×2。当结构数组的元素超过 1 个时，MATLAB 的显示信息中，不再显示不同属性的值，而只显示数组名、属性名和维数大小。

函数 struct 也可用来定义结构数组，其调用格式为：

结构数组名=struct('属性 1','属性值 1', '属性 2','属性值 2', ...)

ii) 结构数组属性值的修改、设置和获取

结构数组一旦形成，就可取出数组中的某个元素并修改该元素的某个属性的值。以上面建立的 student 数组为例，命令

```
str=student(2).name
```

可取出第二个元素的 name 属性的值。

命令

```
n=student(2).test(4,2)
```

取出第二个元素 test 的值中第四行第二列上的数。

同理，可用命令

```
student(2).test(4,2)=0
```

修改第二个元素 test 的值中第四行第二列上的数的值。

关于结构数组有表 1 中的函数。

表 1 结构数组的有关函数

函数名	作用
struct	生成和转换为结构数组
fieldnames	查询结构数组的属性名
getfield	查询结构数组的属性值
setfield	设置结构数组的属性值

rmfield	删除属性
isfield	检查是否为数组的属性
isstruct	检查数组是否为结构型

⑧ 细胞数组

细胞数组也是 MATLAB 里的一类特殊的数组。在 MATLAB 里，由于有细胞数组这个数据类型，才能把不同类型、不同维数的数组组成为一个数组。

细胞数组的每一个元素可为类型不同、维数不同的矩阵、向量、标量或多维数组，**所有元素用大括号括起来**。如矩阵 $A = [1 \ 2 \ 3 \ 4; 2 \ 3 \ 4 \ 5; 3 \ 4 \ 5 \ 6]$ ，则命令

```
c = {A, sum(A), sum(sum(A))}
```

得到一个 1×3 的细胞数组。

关于细胞数组有表 2 中的函数。

表 2 细胞数组的有关函数

函数名	作用
celldisp	显示细胞数组的内容
cell	生成细胞数组
cellplot	用图形方式显示细胞数组
num2cell	把数值型转换为细胞型
deal	输入和输出的匹配
cell2struct	把细胞数组转换为结构数组
struct2cell	把结构数组转换为细胞数组
iscell	检验数组是否为细胞型

i) 细胞数组的生成

有两种方法可以生成细胞数组：用赋值语句直接生成；先用 cell 函数预分配数组，然后再对每个元素赋值。

有两种方法可对元素赋值：一种方法采用数组元素的下标赋值。下面四句命令将建立一个 2×2 的细胞数组。

```
A(1,1) = {[1:5;6:10]};
```

```
A(1,2) = {'Anne cat'};
```

```
A(2,1) = {3+7i};
```

```
A(2,2) = {0:pi/10:pi};
```

在大括号中，逗号或者空格表示每行元素之间的分割，分号表示不同行之间的分割。

另一种方法则把细胞数组的下标用大括号括起来，而所赋的值采用普通数组的形式。例如下面四句生成的细胞数组和上面所生成的完全一样。

```
A{1,1} = [1:5;6:10];
```

```
A{1,2} = 'Anne cat';
```

```
A{2,1} = 3+7i;
```

```
A{2,2} = 0:pi/10:pi;
```

命令

```
B = cell(3,4)
```

创建一个 3×4 的细胞矩阵。

ii) 细胞数组内容的查看

对于上面建立的数组 A，在 Matlab 命令窗口键入变量名 A，将显示数组的简要信息。用大括号 { } 括起来的下标为细胞数组的第几个元素，用圆括号 () 括起来的下标为大

括号{ }对应的某个元素的分量。如

`A{2,2}(1), A{4}(1)`

对上面的细胞数组显示的结果是一样的。

函数 `celldisp` 用来显示细胞数组的每个元素的值。函数 `cellplot` 将画出细胞数组的每个元素的结构图。

当给已经定义的细胞数组下标范围外的元素赋值时, Matlab 自动扩维, 对于没有赋值的元素, 赋值为空矩阵。

7. 数学运算与函数

① 基本代数运算操作 `+, -, *, \, /, ^`,

② 矩阵运算函数: 求行列式(`det`), 矩阵求逆(`inv`), 求秩(`rank`), 求迹(`trace`), 求模(`norm`), `d=eig(A)` 求矩阵 A 的特征值, `[v,d]=eig(A)` 求矩阵 A 的特征向量和特征值, 这里 v 的列向量是对应的特征向量。

矩阵基本运算:

`A\B, B/A, A.*B, A./B, A.\B, A.^B`。

③ 基本数学函数

常用的数学函数有 `sin, cos, tan, abs, min, sqrt, log, log10, sign, asin, acos, atan, max, sum, exp, fix` 等。具体使用方法请参看帮助 `help`。

常用的矩阵函数有 `expm, logm, sqrtm` 和 `funm`, `funm` 函数可计算任何一个基本数学函数的矩阵函数。它可以表示为

`fa=funm(a,'fun')`

式中, `fun` 可以是任意一个基本函数, 如 `sin, cos, log10` 等。

④ 多项式

任意多项式都可以用一个行向量来表示, 即 n 维的向量 a 表示多项式 $y(x) = a(1)x^{n-1} + a(2)x^{n-2} + \cdots + a(n-1)x + a(n)$, 反过来, 任意一个向量就可以作为多项式。

例如:

`p=[1 -6 11 -6]; poly2sym(p,'x')`

`ans =`

`x^3-6*x^2+11*x-6`

求 $s^3 + 2s^2 + 3s + 4$ 的根可用如下命令。

`A=[1 2 3 4]; roots(A)`

i) **poly 函数**

`p=poly(A)`, A 是一个 $n \times n$ 的矩阵时, 此函数返回矩阵 A 的特征多项式 p , p 是 $n+1$ 维向量; A 是向量时, 此函数返回以向量中的元素为根的多项式。

ii) 多项式的数组运算

`y=polyval(p,x)` 计算多项式在 x 处的值, x 可以是矩阵或向量, 此时函数计算多项式在 x 的每个元素处的值。

iii) 多项式的矩阵运算

`y=polyvalm(p,x)` 相当于用矩阵 x 代替多项式的变量来对矩阵而不是对数组进行运算, x 必须是方阵。例如:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \quad p(A) = A^2 + 3A + 2I$$

可采用如下的命令进行计算

`p=[1 3 2];`

```
a=[1 2; 3 4];
```

```
polyvalm(p,a)
```

iv) 多项式的乘法和除法运算

$w=\text{conv}(u,v)$ 此函数求多项式 u 和 v 的乘积, 即求向量 u 和 v 的卷积。如果 $m=\text{length}(u)$, $n=\text{length}(v)$, 则 w 的长度为 $m+n-1$ 。

$[q,r]=\text{deconv}(u,v)$ 此函数表示多项式 u 除以多项式 v 得到商多项式 q 和余数多项式 r , 如果 r 的元素全部为零, 则表示多项式 v 可以整除多项式 u 。

8. 绘图命令

① 二维图形

二维绘图的基本命令有 `plot`, `loglog`, `semilogx`, `semilogy` 和 `polar`。它们的使用方法基本相同, 其不同特点是在不同的坐标中绘制图形。`plot` 命令使用线性坐标空间绘制图形; `loglog` 命令在两个对数坐标空间中绘制图形; 而 `semilogx`(或 `semilogy`) 命令使用 x 轴(或 y 轴)为对数刻度, 另外一个轴为线性刻度的坐标空间绘制图形; `polar` 使用极坐标空间绘制图形。

二维绘图命令 `plot` 为了适应各种绘图需要, 提供了用于控制线色、数据点和线型的 3 组基本参数。它的使用格式如下:

```
plot(x,y,'color_point_linestyle')
```

该命令是绘制 y 对应 x 的轨迹的命令。 y 与 x 均为向量, 且具有相同的元素个数。用字符串 'color_point_linestyle' 完成对上面 3 个参数的设置。线色(r-red, g-green, b-blue, w-white, k-black, i-invisible, y-yellow), 数据点(., o, x, +, *, S, H, D, V, ^, >, <, p) 与线型(—, —., —, :) 都可以根据需要适当选择。

当 `plot(x,y)` 中的 x 和 y 均为 $m \times n$ 矩阵时, `plot` 命令将绘制 n 条曲线。

`plot(t, [x1, x2, x3])` 在同一坐标轴内同时绘制三条曲线。

如果多重曲线对应不同的向量绘制, 可使用命令

```
plot(t1,x1,t2,x2,t3,x3)
```

式中 $x1$ 对应 $t1$, $x2$ 对应 $t2$ 等等。在这种情况下, $t1$, $t2$ 和 $t3$ 可以具有不同的元素个数, 但要求 $x1$, $x2$ 和 $x3$ 必须分别与 $t1$, $t2$ 和 $t3$ 具有相同的元素数量。

`subplot` 命令使得在一个屏幕上可以分开显示 n 个不同坐标系, 且可分别在每一个坐标系中绘制曲线。其命令格式如下:

```
subplot(r,c,p)
```

该命令将屏幕分成 $r \times c$ 个子窗口, 而 p 表示激活第 p 个子窗口。窗口的排号是从左到右, 自上而下。

在图形绘制完毕后, 执行如下命令可以再在图中加入标题、标号、说明和分格线等。这些命令有 `title`, `xlabel`, `ylabel`, `text`, `gtext` 等。它们的命令格式如下:

```
title('My Title'), xlabel('My X-axis Label'), ylabel('My Y-axis Label'),
```

```
text(x,y,'Text for annotation'), gtext('Text for annotation'),  
grid
```

`gtext` 命令是使用鼠标器定位的文字注释命令。当输入命令后, 可以在屏幕上得到一个光标, 然后使用鼠标器控制它的位置。按鼠标器的左键, 即可确定文字设定的位置。

`hold on` 是图形保持命令, 可以把当前图形保持在屏幕上不变, 同时在这个坐标系内绘制另外一个图形。`hold` 命令是一个交替转换命令, 即执行一次, 转变一个状态(相当于 `hold on`、`hold off`)。

Matlab 可以自动选择坐标轴的定标尺度, 也可以使用 `axis` 命令定义坐标轴的特殊定标尺度。其命令格式如下:

```
axis([x-min, x-max, y-min, y-max])
```

axis 命令的另一个作用是控制纵横尺度的比例。例如，输入 axis('square') 后，可得到一个显示方框，此时再在该框内绘制一个圆形时（如：plot(sin(x),cos(x))），在屏幕上可以看到一个圆（一般情况下，由于屏幕的不规则原因，只能看到一个椭圆）。再次输入 axis('normal') 命令，屏幕返回到一般状态。

例 3

```
x=0:0.25:5;
y1=x.^0.1;
y2=x.^0.5;
y3=x.^0.8;
y4=x.^1.5;
t=0:0.001:2*pi;
hold on
plot(3*cos(t),3*sin(t))
plot(x,y1,'bo',x,y2,'rH-')
plot(x,y3,'gp--')
plot(x,y4,'mx-.')
title('My Title'),xlabel('My X-axis Label'),ylabel('My Y-axis Label')
text(2,8,'Text for annotation')
gtext('Text for annotation'),grid
```

结果如图1所示。

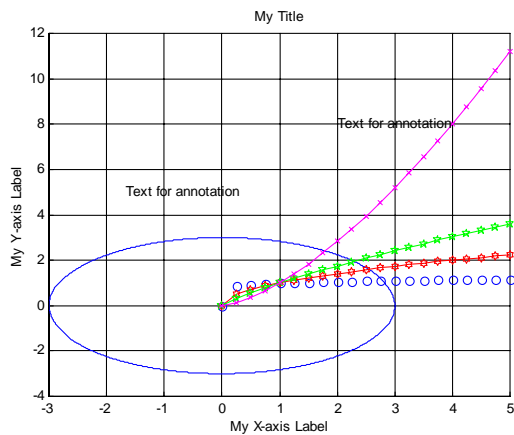


图 1 plot 等命令示意图

例 4 用极坐标绘图命令绘制方程 $r = \cos 2\theta$ （四叶玫瑰线）的图形。

```
x=0:0.1:2*pi;
r=cos(2*x);
polar(x,r)
```

例 5 随机地产生 20 个数据，再根据这些数据画统计直方图。

```
x=rand(1,20);
y=round(20*x);
subplot(1,2,1)
hist(x)
```



```
subplot(1,2,2)
hist(y)
```

② 三维图形

在实际工程计算中，最常用的三维绘图是三维曲线图、三维网格图和三维曲面图 3 种基本类型。与此对应，Matlab 也提供了 3 个三维基本绘图命令（三维曲线命令 plot3、三维网格命令 mesh 和三维表面命令 surf）。下面初步介绍其中的两个。

i) 三维曲线

plot3(x, y, z) 通过描点连线画出曲线，这里 x, y, z 都是 n 维向量，分别表示该曲线上点集的横坐标、纵坐标、竖坐标。

例 6 在区间 $[0, 10 * \pi]$ 画出参数曲线 $x = \sin(t)$ ， $y = \cos(t)$ ， $z = t$ ，并分别标注。

```
t=0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
xlabel('sin(t)'),ylabel('cos(t)'),zlabel('t')
```

ii) 网格

命令 mesh(x, y, z) 画网格曲面。这里 x, y, z 是三个同维数的数据矩阵，分别表示数据点的横坐标、纵坐标、竖坐标，命令 mesh(x, y, z) 将该数据点在空间中描出，并连成网格。

例 7 绘制二元函数

$$z = \frac{\sin(xy)}{xy}$$

的三维网格图。

```
x=-3:0.1:3;y=-5:0.1:5;
x1=ones(size(y'))*x;y1=y'*ones(size(x));
[x2,y2]=meshgrid(x,y);
z1=(sin(x1.*y1)+eps)./(x1.*y1+eps);
z2=(sin(x2.*y2)+eps)./(x2.*y2+eps);
subplot(1,2,1),mesh(x1,y1,z1)
subplot(1,2,2),mesh(x2,y2,z2)
```

③ 符号函数的简易绘图函数 ezplot

ezplot(f) 绘制 f(x) 的函数图，这里 f 为代表数学表达式的包含单个符号变量 x 的字符串或符号表达式。x 轴的近似范围为 $[-2 * \pi, 2 * \pi]$ 。

ezplot(f,[xmin,xmax]) 使用输入参数来代替默认横坐标范围 $[-2 * \pi, 2 * \pi]$ 。

例 8 画出函数 $y = \tan x$ 的图形

解 ezplot('tan(x)')

④ 绘制函数图函数 fplot

fplot(fun,lims) 绘制由字符串 fun 指定函数名的函数在 x 轴区间为 lims=[xmin, xmax] 的函数图。若 lims=[xmin,xmax,ymin,ymax]，则 y 轴也被限制。fun 必须为 M 文件的函数名或对变量 x 的可执行字符串，此字符串被送入函数 eval 后被执行。函数 fun(x) 必须要返回针对向量 x 的每一元素结果的向量。

例 9 画 $f(x) = \begin{cases} x+1, & x < 1 \\ 1+\frac{1}{x}, & x \geq 1 \end{cases}$ 的图形。

解：(1) 首先用 M 文件 fun1.m 定义函数 $f(x)$ 如下：

```
function y=fun1(x);
if x<1
    y=x+1;
else
    y=1+1./x;
end
```

(2) 在 matlab 命令窗口输入

```
fplot('fun1',[-3,3])
```

就可画出函数 $f(x)$ 的图形。

这里也可以使用匿名函数，编写程序如下：

```
fun1=@(x) (x+1)*(x<1)+(1+1/x)*(x>=1);
fplot(fun1,[-3,3])
```

⑤ 画等高线的函数 contour

在 Matlab 中，可以用 contour 命令绘制平面等高线，用 contour3 命令绘制空间等高线。

绘制平面等高线 contour 命令的具体使用格式为：

```
contour(x,y,z,n)
contour(x,y,z,v)
contourf(...)
```

其中 x, y 为其横纵坐标值向量，如果 x 为 s 维，y 为 m 维，则 z 为 m×s 维矩阵，为对应于坐标 (x,y) 的高度。参数 n 为整数，指定了绘出等高线的条数。参数 v 为向量，指定了在哪些高度绘出等高线，如只想在一个高度 z 绘出等高线，则 v=[z, z]。

contourf(...)命令的参数与 contour 命令完全相同，只是其绘出的等高线图将被自动填上颜色。

```
c=contour(x,y,z,n)
```

```
c=contour(x,y,z,v)
```

上面 2 个命令用来计算所画等高线的 x, y 坐标值。

```
clabel(c)
```

```
clabel(c,v)
```

上面 2 个命令用来标注计算的 c 阵处的高度值。clabel(c)将把所绘等高线全部自动标注，clabel(c,v)将自动标注由向量 v 确定的若干条等高线的高度值。

例 10

```
clc;clf;
```

```
x=0:400:5600;
```

```
y=0:400:4800;
```

```
z=[370,470,550,600,670,690,670,620,580,450,400,300,100,150,250;
```

```
510,620,730,800,850,870,850,780,720,650,500,200,300,350,320;
```

```
650,760,880,970,1020,1050,1020,830,800,700,300,500,550,480,350;
```

```
740,880,1080,1130,1250,1280,1230,1040,900,500,700,780,750,650,550;
```

```
830,980,1180,1320,1450,1420,1400,1300,700,900,850,810,380,780,750;
```

```
880,1060,1230,1390,1500,1500,1400,900,1100,1060,950,870,900,930,950;
```

```
910,1090,1270,1500,1200,1100,1350,1450,1200,1150,1010,880,1000,1050,1100;
```

```

950,1190,1370,1500,1200,1100,1550,1600,1550,1380,1070,900,1050,1150,1200;
1430,1450,1460,1500,1550,1600,1550,1600,1600,1600,1550,1500,1500,1550,1550;
1420,1430,1450,1480,1500,1550,1510,1430,1300,1200,980,850,750,550,500;
1380,1410,1430,1450,1470,1320,1280,1200,1080,940,780,620,460,370,350;
1370,1390,1410,1430,1440,1140,1110,1050,950,820,690,540,380,300,210;
1350,1370,1390,1400,1410,960,940,880,800,690,570,430,290,210,150];
hold on
c=contourf(x,y,z,10);clabel(c)
text(0,800,'▼\leftarrow 起点','fontsize',15,'color','r')
text(4000,2000,'▼\leftarrow 居民点','fontsize',15,'color','r')
text(2000,4000,'▼\leftarrow 矿区','fontsize',15,'color','b')
line([2400,4800],[2400,0])
figure
surf(x,y,z),view(50,30),hold on
text(0,800,680,'▼','fontsize',15,'color','r')
text(0,600,880,'起点','fontsize',15,'color','r')
text(4000,2000,950,'▼','fontsize',15,'color','r')
text(4200,2000,1150,'居民点','fontsize',15,'color','w')
text(2000,4000,1320,'▼\leftarrow 矿区','fontsize',15,'color','b')

```

§3 程序设计

1. 关系和逻辑运算

关系运算符有以下 6 种

<, <=, >, >=, ==, ~=,

关系成立时结果为 1，否则为 0。

逻辑运算符有

&, |, ~, xor

分别代表逻辑运算中的与、或、非、异或。0 的逻辑量为“假”，而任意非零数的逻辑量为“真”。

例 11

```
a=[1, 2, 3, 4];
```

```
b=[0, 1, 0, 2];
```

```
s1=a&b, s2=a|b, s3=~a, s4=xor(a, b)
```

2. 关系和逻辑函数

除了关系和逻辑运算符，Matlab 提供了关系和逻辑函数。

any(x)

如果在向量 x 中，至少有一个非零元素，则 any(x) 返回 1；矩阵 x 的每一列有非零元素，返回 1。

all(x)

如果在一个向量 x 中，所有元素非零，则 all(x) 返回 1；矩阵 x 中的每一列所有元素非零，则返回 1。

find: 找出向量或矩阵中非零元素的位置标识

find 函数在对数组元素进行查找、替换和修改等操作中占有非常重要的地位，熟练运用可以方便而灵活地对数组进行操作。

find(a) 返回由矩阵 a 的所有非零元素的位置标识组成的向量（元素的标识是按列进行的），如果没有非零元素则会返回空值。

例 12

```
a=[0,1;0,2]
b=zeros(1,5)
s1=find(a),s2=find(b)
```

```
[i,j,v]=find(a)
```

此函数返回矩阵 a 的非零元素的行和列的标识,其中 i 代表行标,而 j 代表列标,同时,将相应的非零元素的值放于列向量 v 中。

例 13

```
a=[0,5;0,7]
[i,j,v]=find(a)
```

例 14 找出 a 中不等于 7 的元素的位置。

```
a=[0,5;0,7]
s=find(a~=7)
```

例 15 将矩阵 a 中等于 7 的元素的值换成矩阵 c 中相应位置上的元素。

```
a=[0,5;0,7]
c=rand(2,2)
a(find(a==7))=c(find(a==7))
```

例 16 将矩阵 a 中等于 0 的元素删除。

```
a=[1,0,5;0,2,7]
a(find(a==0))=[]
b=reshape(a,[2,2])
b(:,2)=[]
```

或者使用命令 nonzeros 实现,程序如下

```
a=[1,0,5;0,2,7]
a=nonzeros(a)
```

3. 流程控制语句

计算机程序通常都是从前到后逐条执行的。但往往也需要根据实际情况,中途改变执行的次序,称为流程控制。Matlab 设有 4 种流程控制的语句结构,即 if 语句、while 语句、for 语句和 switch 语句。

i) if 语句

根据复杂程度,if 语句有 3 种形式

```
if 表达式 语句组 a, end
if 表达式 1 语句组 a, else 语句组 b, end
if 表达式 1 语句组 a, elseif 表达式 2 语句组 b, else 语句组 c, end
```

ii) while 语句

while 语句的结构形式为

```
while 表达式 语句组, end
```

例 17 求 Matlab 中的一个充分大的实数。

设定一个数 x,让它不断增大,直到 Matlab 无法表示它的值,只能表示为 Inf 为止。

```
x=rand;
while x~=Inf
    x1=x;x=2*x;
end
```

x1

iii) for 语句

for 语句的结构形式为

for k=初值: 增量: 终值 语句组 a, end

iv) switch 语句

switch-case-otherwise 语句可用来实现均衡的多分支语句, 其基本语句结构可表示为

switch 表达式 (标量或字符串)

case 值 1

语句组 1

case 值 2

语句组 2

.....

otherwise

语句组 n

end

例 18 判断输入数 n 的奇偶性。

```
n=input('n=')
```

```
switch mod(n,2)
```

```
case 1,a='奇'
```

```
case 0,a='偶'
```

```
otherwise,a='非整数'
```

```
end
```

4. M 文件与 M 函数

由 Matlab 语句构成的程序文件称作 M 文件, 它将 m 作为文件的扩展名。M 文件可分为程序文件和函数文件两种。

程序文件一般是由用户为解决特定的问题而编制的程序, 函数文件也称为子程序, 它必须由 Matlab 程序来调用。函数文件往往具有一定的通用性, 并且可以进行递归调用。

i) 程序文件

程序文件的格式特征如下:

(1) 前面的若干行通常是程序的注释, 每行以 “%” 开始, 当然注释可以放在程序的任何部分。注释可以是汉字, 注释是对程序的说明, 它增加了程序的可读性。在执行程序时, Matlab 将不理睬 “%” 后直到行末的全部文字。

(2) 然后是程序的主体。如果文件中有全局变量, 即子程序和主程序共用的变量, 应在程序的起始部分注明。其语句是

```
global 变量名 1 变量名 2 .....
```

(3) 整个程序应按 Matlab 标识符的要求起文件名, 文件名不能以数字开始, 不允许用汉字。

ii) 函数文件

函数文件是用来定义子程序的。它与程序文件的主要区别有 3 点:

(1) 由 function 起头, 后跟的函数名要与文件名相同。

(2) 有输入输出变量, 可进行变量传递。

(3) 除非用 global 声明, 函数中的变量均为局部变量, 不保存在工作空间中。

例 19 编写求阶乘的函数。

```

function y=fac(n);
if n<0
    error('n is smaller than 0,error input. ');
    return;
end
if n==0|n==1
    y=1;
else
    y=n*fac(n-1);
end

```

并把上述文件命名为 fac.m，调用函数时实际上是调用文件名。

5. 利用字符串模拟运算式

利用字符串建立表达式后，再用 eval 命令执行它，可以使程序设计更加灵活。

但是注意表达式一定要是字符串。其命令格式为：

```
eval('字符串')
```

例 20 先定义字符串 t 为平方根运算，再用 eval 求出 1 到 10 的平方根。

```

clear,clc
t='sqrt(i)';
for i=1:10
    s(i)={char(['The square root of ',int2str(i), ' is ',...
               num2str(eval(t))])};
    %上面大括号 {} 代表建立细胞数组
end
s(:)

```

例 21 如果要输入几十个甚至上百个文件，用手工操作十分繁琐，然而灵活运用 eval 函数可以自动完成这一工作。假设数据文件名从 data1.dat ~ data10.dat，放在 D:\matlab\chp 目录下，操作如下：

```

for i=1:10
    eval(['load d:\matlab\chp\data',int2str(i),'.dat'])
end

```

函数 feval 用于执行字符串代表的文件或函数。

例 22

```

fun=['sin';'cos';'log'];
k=input('选择第几个函数:');
x=input('输入数值:');
feval(fun(k,:),x)

```

例 23 当前 matlab\work 目录下有三个图形文件 hlpstep1.gif ~ hlpstep3.gif，分别打开这三个文件。

```

clear,clc
fun='imread';
for i=1:3
    str=char(['hlpstep',int2str(i),'.gif']);
    x=feval(fun,str)
end

```

§4 文件

根据数据的组织形式, Matlab 中的文件可分为 ASCII 文件和二进制文件。ASCII 文件又称文本文件, 它的每一个字节存放一个 ASCII 代码, 代表一个字符。二进制文件是把内存中的数据按其内存中的存储形式原样输出到磁盘上存放。

Matlab 中的关于文件方面的函数和 C 语言相似, 见表 3。

表 3 Matlab 的文件操作命令

函数分类	函数名	作用
打开和关闭文件	fopen	打开文件
	fclose	关闭文件
读写二进制文件	fread	读二进制文件
	fwrite	写二进制文件
格式 I/O	fscanf	从文件中读格式数据
	fprintf	写格式数据
	fgetl	从文件中读行, 不返回行结束符
	fgets	从文件中读行, 返回行结束符
读写字符串	sprintf	把格式数据写入字符串
	sscanf	格式读入字符串
文件定位	feof	检验是否为文件结尾
	fseek	设置文件定位器
	ftell	获取文件定位器
	frewind	返回到文件的开头

1. 文件的打开和关闭

对文件读写之前应该“打开”该文件, 在使用结束之后应“关闭”该文件。

函数 fopen 用于打开文件, 其调用格式为:

```
fid=fopen('filename','permission')
```

fid 是文件标识符 (file identifier), fopen 指令执行成功后就会返回一个正的 fid 值, 如果 fopen 指令执行失败, fid 就返回-1。

filename 是文件名。

permission 是文件允许操作的类型, 可设为以下几个值:

‘r’ 只读

‘w’ 只写

‘a’ 只能追加 (append)

‘r+’ 可读可写

与 fopen 对应的指令为 fclose, 它用于关闭文件, 其指令格式为:

```
status=fclose(fid)
```

如果成功关闭文件, status 返回的值就是 0。

2. 读写操作

i) fwrite 的指令格式

```
fid=fopen('filename','permission')
```

```
fwrite(fid, 要保存的数据矩阵, '精度格式')
```

执行 help fread 即可查到精度格式的设置。

ii) fprintf 的指令格式

```
fprintf(fid, '数据格式', 需要保存的数据矩阵)
```

例 24 产生 10 个随机数, 并保存到一个纯文本文件 data1.txt 中。

```
clear,clc
```

```

a=rand(1,10)
fid=fopen('data1.txt','w');
fprintf(fid,'%8.4f',a);
fclose(fid);
load data1.txt
data1
iii) dlmwrite 的指令格式
dlmwrite('文件名',矩阵,'delimiter','分隔符','precision','精度格式')
例 25 产生一个随机的 4×5 矩阵，并保存在纯文本文件中。
clear,clc
a=rand(4,5)
dlmwrite('data2.txt',a,'delimiter','\t','precision','%6f','newline','pc')
iv) save 的指令格式

```

```

save filename 变量1 变量2 ...

```

执行该命令把变量 1、变量 2、…保存在文件 filename 中。

使用 load filename 即可把变量 1、变量 2、…调出来。

如果要保存为 ASCII 码，就要在后面加上 -ascii

```

save filename 变量1 变量2 ... -ascii

```

对于 save 指令，处理大量数据存取有一个技巧非常有用，即：

```

save('filename','变量1','变量2',...)

```

由于 filename 是用字符串表示的，所以可以使用程序进行控制，使其每处理完一次就存一个不同的文件名称。

例 26

```

clear,clc
m=1:10;
for i=1:length(m)
    n=m.^2;
    nf=[m',n'];
    t=char(['nf',int2str(i),'=nf'])
    eval(t)
    save(['data',int2str(i)],['nf',int2str(i)])
end

```

iv) load 纯文本文件

```

load filename.txt

```

就建立了 filename 命名的变量。

例 27 现有一纯文本数据文件 caipao.txt，保存了山东省 65 期的福利彩票中奖号码，试对中奖号码给出一些统计，并按一定的规则产生两组彩票号码。

```

clc,clear
load caipiao.txt;
cp=caipiao;
for i=1:30
    b(i)=length(find(cp==i));
end
[b,id]=sort(b);
mai1=sort(id(1:7)), mai2=sort(id(24:30))

```


%LINGO 软件对纯文本文件的格式识别能力比较差，下面我们给出两种写纯文本文件的格式，供将来向 LINGO 软件传递数据使用

```
dlmwrite('cpsj1.txt',caipiao); %把 caipiao 矩阵重新写到纯文本文件中
fid1=fopen('cpsj2.txt','w');
fprintf(fid1,'%6d\n',caipiao'); %把 caipiao 矩阵重新写到纯文本文件中
fclose(fid1);
```

§5 动画制作

动画制作通常有两种方法：一种是预先将图形制作好，并放到图形缓冲区内，然后一帧一帧地播放。另一种方法是保持整个背景图案不变，只更新运动部分的图案，以便加快每幅图的实时生成速度。

1. 简单动画制作

在 `getframe` 命令下，可以完成动态数据到动态画面的制作。其使用格式及制作步骤一般为：

```
for j=1:n
    plot_command
    M(j)=getframe;
end
movie(M,n)
```

其中 `M(j)=getframe` 将当前图形窗口中的画面作为第 j 帧存入矩阵 `M`，`movie(M,n)` 将按顺序放映矩阵 `M` 中存储的画面，并重复 n 次。

例 28

```
x=-3:0.1:3;
[x,y]=meshgrid(x);
z=sin(x.*y).*exp(x.*y/5);
for j=1:30
    mesh(cos(4*pi*j/30)*z,z)
    m(j)=getframe
end
movie(m,10)
```

例 29 模拟 6 个移动物体

```
x0=[150 85 150 145 130 0];
y0=[140 85 155 50 150 0];
q=[243 236 220.5 159 230 52]*pi/180;
t=0:0.05:2*pi;
for i=0:280
    pause(0.01);
    for j=1:5
        axis([0 160 0 160]);
        fill(x0(j)+0.8*i*cos(q(j))+4*cos(t),y0(j)+0.8*i*sin(q(j))+4*sin(t),'b')
        hold on;
    end
    fill(x0(6)+0.8*i*cos(q(6))+4*cos(t),y0(6)+0.8*i*sin(q(6))+4*sin(t),'r')
    hold off;
end
```

2. 其它例子

在 Matlab 里利用图形的“EraseMode”属性可以实现显示新对象，擦除旧对象，而又不破坏背景图案。图形的“EraseMode”属性有以下四种：

normal 方式

重绘整个显示区，这种方式产生的图形最准确，但较慢。

none 方式

不做任何擦除，直接在原来图形上绘制。

xor 方式

对象的绘制和擦除由背景颜色和屏幕颜色的异或而定。只擦除和屏幕颜色不一致的旧对象的点，只绘制和屏幕颜色不一致的新对象的点。

background 方式

把旧对象的颜色变为背景色，这种方式影响被擦除对象下面的对象。

当新对象的属性修改后，应刷新屏幕，使新的对象显示出来。Matlab 里屏幕刷新的命令是 **drawnow**。**drawnow** 命令使 Matlab 暂停目前的任务而去刷新屏幕。若不使用 **drawnow** 命令，Matlab 则等到任务序列执行完后才刷新。

例 30 布朗运动

(1) 先确定布朗运动的点数 n 和一个温度 s (或速度)。比如 $n=20$, $s=0.002$ 。达到最好的动画效果的 n 、 s 与使用的计算机的性能有关。在以原点为中心、边长为 1 的正方形内产生 n 个位置随机分布的点。

(2) 在单位正方形里绘制所有的点。设置图形的“EraseMode”属性为异或 (**xor**)，即当图形的某个点的位置变化时，不用全部重绘图形，而只绘制不相同的部分，这将大大提高绘图速度。

(3) 用 **while** 循环实现动画效果，在每一次循环中给点的坐标加上一些正态分布的噪声，这样整个图形中只有点的坐标发生变化，而不用全部重绘整个图形。

程序如下：

```
clc,clear
n=20;s=0.002;
axis square
grid off
x=rand(n,1)-0.5;y=rand(n,1)-0.5;
h=plot(x,y,'.');
set(h,'EraseMode','xor','MarkerSize',18)
while 1
    x=x+s*rand(n,1);y=y+s*rand(n,1)
    set(h,'Xdata',x,'Ydata',y)
    if all(x>1) & all(y>1)
        break
    end
    drawnow
end
```

例 31 制作一幅钻石沿着圆周运动 2 周的动画。

程序如下：

```
t=0:pi/200:pi*2;
x=sin(t);y=cos(t);
axis square
plot(x,y,'b')
n=length(t);
```

```

h=line('color','red','marker','diamond','erasemode','xor');
i=1;j=1;
while 1
    set(h,'xdata',x(i),'ydata',y(i))
    pause(0.01)
    i=i+1;j=j+1;
    if i>n
        i=1;
    end
    if j>2*n
        break
    end
end
end

```

习题

1. 求方程 $x^3 + x^2 + x + 1 = 0$ 的根。
2. 产生一个随机的 3 阶矩阵，并求该矩阵的逆矩阵，特征值与特征向量和行列式的值。
3. 存储工作间变量 x, y, z 到文件 data1.mat，再在程序中调用变量 x, y, z。
4. 产生 20 个随机数，再把这 20 个随机数存放在纯文本文件中。