

第3讲 MATLAB 的基本使用方法

司守奎

烟台市，海军航空大学

Email: sishoukui@163.com

3.1 MATLAB 简介

3.1.1 MATLAB 发展历史

MATLAB 每年更新两次，春季发布每年度的 A 版本，秋季发布 B 版本。

3.1.2 MATLAB 的安装

MATLAB 的安装分如下 4 步：

- (1) 输入产品序列号
- (2) 安装完毕后，要激活。
- (3) 把安装目录下的一些动态链接库文件用源文件目录下的对应文件替换。
- (4) 把 MATLAB 快捷方式发送到桌面。

注 3.1 序列号的纯文本文件和链接库文件在目录\serial 或 crack 下。

3.1.3 MATLAB 的帮助

通常能够起到帮助作用、获取帮助信息的命令有 help、lookfor、which、doc、get、type 等。

(1) help 命令

help 命令是 MATLAB 中最有用的命令之一。下面介绍 help 的几种常见使用情况。

i) 直接使用 help 命令

在命令窗口直接输入命令 help，并回车，则在命令窗口显示 MATLAB 的所有工具箱信息。下面我们用

help ↵ %显示所有工具箱信息。

表示上面的内容，这里 ↵ 表示回车，% 是 MATLAB 的注释引导符，% 后面的是注释内容。

ii) 使用 help 工具箱名

使用命令 help 工具箱名，可以获取该工具箱的所有函数的信息。例如

help optim ↵ %获得优化工具箱的基本信息和函数列表。

iii) 使用 help 函数名

使用命令 help 函数名，就可以获得该函数的帮助信息。例如

help linprog ↵ %获得优化工具箱中线性规划命令 linprog 的帮助信息。

(2) lookfor 命令

Lookfor 命令在 MATLAB 默认路径下搜索所有 M 文件第一个注释行中的关键字。通常在不确定某个函数时，仅知道该函数的功能，lookfor 命令可以根据用户提供的完整或不完整的关键字，去搜索出一组与之相关的命令。

lookfor integral ↵ %查找所有有关积分的命令。

(3) which 命令

which 命令可以用来定位函数的位置。通过这个位置信息，可以获取函数所属的类别。通常，在创建一个 MATLAB 文件时，为了避免与系统函数等同名，就应该先用“which 文件名”搜索查找是否存在你想要保存的文件名。

另外，利用得到的位置信息可以查找一些相关联的文件的帮助信息。例如，在编程过程中，需要计算一个微分方程在指定点的数值解，但想不起该函数名，但记得求微分方程数值解的命令 ode45，因此，采用 which ode45 定位 ode45。

which ode45 ↵

D:\Program Files\MATLAB\R2015a\toolbox\MATLAB\funfun\ode45.m

从给出的地址可以看出，ode45 命令属于 funfun 类。于是用 help funfun 查找该类别信息，在该类别的 Input and Output functions 子类别中找到：

deval - Evaluates the solution of a differential equation problem.

然后，再通过用 help deval 获得该函数的详细帮助信息和使用方法。

(4) 超文本格式的帮助文件

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 eig` 获取求矩阵最大特征值及对应特征向量的超文本帮助。

(5) pdf 帮助文件

可从 MathWorks 网站上下载有关的 pdf 帮助文件。

3.1.4 数据类型

常用的数据有数值矩阵, 细胞数值, 结构数据。

下面简单介绍一下数值矩阵。

3.1.4.1 数值矩阵的建立

在 MATLAB 中, 矩阵的建立方式多种多样。比较常用的建立方式有直接输入、通过语句和函数建立矩阵和从外部数据文件中导入矩阵 3 种。

1. 直接输入

直接输入是最简单的矩阵构建方式。直接输入矩阵, 应遵循如下几条规则:

- (1) 矩阵元素应当在方括号内;
- (2) 行内的元素, 用逗号或者空格隔开;
- (3) 行与行之间, 用分号或者回车分割;
- (4) 元素可以是数值或表达式。

2. 通过语句和函数建立矩阵

`t=[0:0.1:5]` %产生从 0 到 5 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)` %在 10^{n1} 和 10^{n2} 之间按照对数距离等间距产生 n 个数 (缺省 n 时, 产生 50 个数)。

3. 从外部数据文件中导入矩阵

例如从外部纯文本文件 `data1.txt` 导入矩阵的命令为

`a=load('data1.txt')`

或

`a=textread('data1.txt')`

3.1.4.2 特殊矩阵

1. 单位矩阵

`eye(m)` %生成 m 阶单位阵。

`eye(m,n)` %生成 $m \times n$ 矩阵, 其中得到一个可允许的最大单位矩阵而其余处补 0。

`eye(size(a))` %生成与矩阵 `a` 同样大小的单位矩阵。

2. 全部元素为 1 的矩阵

`ones(n)` %生成全部元素为 1 的 n 阶方阵。

`ones(size(a))` %生成与矩阵 `a` 同样维数的全部元素为 1 的矩阵。

`ones(m,n)` %生成全部元素为 1 的 $m \times n$ 矩阵。

3. 全部元素为 0 的矩阵

`zeros(n)` %生成全部元素为 0 的 n 阶方阵。

`zeros(m,n)` %生成全部元素为 0 的 $m \times n$ 矩阵。

4. 空矩阵

空矩阵是一个特殊矩阵, 这在线性代数中是不存在的。例如

`a=[]`

矩阵 `a` 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除矩阵的行与列。

例如

`b(:,3)=[]`

表示删除矩阵 `b` 的第 3 列。

5. 随机数矩阵

`rand(m,n)` %产生 $m \times n$ 矩阵, 其中的元素是服从 $[0,1]$ 区间上均匀分布的随机数。

`randi([imin,imax],m,n)` %生成 $m \times n$ 矩阵, 其中的元素为 $[imin,imax]$ 区间上的随机整数。

`normrnd(mu,sigma,m,n)` %产生 $m \times n$ 矩阵, 其中的元素是服从均值为 `mu`, 标准差为 `sigma` 的正态分布的随机数。

`randn(m,n)` %产生均值为 0，方差为 1 的正态分布的 $m \times n$ 随机数矩阵。
`exprnd(mu,m,n)` %产生 $m \times n$ 矩阵，其中的元素是服从均值为 `mu` 的指数分布的随机数。
`poissrnd(mu,m,n)` %产生 $m \times n$ 矩阵，其中的元素是服从均值为 `mu` 的泊松（Poisson）分布的随机数。
`unifrnd(a,b,m,n)` %产生 $m \times n$ 矩阵，其中的元素是服从区间[a,b]上均匀分布的随机数。
`mvnrnd(mu,sigma,n)` %产生 n 对均值向量为 `mu`，协方差矩阵为 `sigma` 的多维正态分布的随机数。
`randperm(n)`产生 1 到 n 的一个随机全排列。
`perms([1:n])`产生 1 到 n 的所有全排列。

3.1.5 运算符

MATLAB 提供了丰富的运算符，主要包括算术运算、关系运算和逻辑运算。算术运算用于数值计算，关系运算和逻辑运算的返回值为逻辑型变量，其中 1 代表逻辑真，0 代表逻辑假。

1.算法运算符

MATLAB 提供的基本算术运算有+（加）、-（减）、*（乘）、/或\（除）和^（乘方）。

这里我们强调一些点运算，点运算符有.`*`（点乘）、./、.\（点除）和.^（点乘方），这里的点运算是向量的逐个元素做对应的运算。

2.关系运算符

关系运算符用于比较两个操作数的大小，返回值为逻辑型变量。在 MATLAB 中，关系运算符如表 3.1 所示。

表 3.1 关系运算符

关系运算符	说明	函数
<	小于	lt
<=	小于等于	le
>	大于	gt
>=	大于等于	ge
==	恒等于	eq
~=	不等于	ne

3.逻辑运算符

在 MATLAB 中，逻辑运算分为三类，分别为基本逻辑运算、快捷逻辑运算和逐位逻辑运算。基本逻辑运算有四种，分别为逻辑与（&）、逻辑或（|）、逻辑非（~）和逻辑异或，如表 3.2 所示。逻辑与和逻辑或为双目运算符，逻辑非为单目运算符。

表 3.2 基本的逻辑运算

运算符	函数	说明	运算符	函数	说明
&	and	逻辑与	~	not	逻辑非
	or	逻辑或		xor	逻辑异或

在 MATLAB 中，可以对二进制数进行逐位逻辑运算，并将运算的结果转换为十进制数。MATLAB 中逐位逻辑运算函数如表 1-8 所示。

表 3.3 逐位逻辑运算函数

函数	说明	函数	说明
bitand(a,b)	逐位逻辑与	bitcmp(a,b)	逐位逻辑非
bitor(a,b)	逐位逻辑或	bitxor(a,b)	逐位逻辑异或

例 3.1 逐位逻辑运算实例。

```

clc, clear
a=imread('data2.bmp'); %读取一副 bmp 图像
b=bitand(a,240); %原图像与 11110000（二进制）=240（十进制）逐位与运算，提出原图像的高 4 位数据
c=bitand(a,15); %原图像与 00001111（二进制）=15（十进制）逐位与运算，提出原图像的低 4 位数据

```

```
imshow(a) %显示原图像
figure,imshow(b) %显示原图像的高 4 位数据
figure, imshow(c) %显示原图像的低 4 位数据。
```

3.2 MATLAB 程序设计

3.2.1 M 文件

M 文件是用 MATLAB 语言编写的、可以完成某些操作或者实现某种算法的程序文本文件。实际上, MATLAB 提供的内部函数以及各种工具箱都是利用 MATLAB 语言开发的 M 文件。

通常, M 文件根据调用方式的不同分为两类: 脚本文件和函数文件。脚本文件是许多 MATLAB 代码按顺序组成的命令序列集合的 M 文件。与在命令窗口逐行执行文件中的所有指令, 其结果是一样的。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。

1. 脚本文件

脚本文件的格式特征如下:

(1) 前面的若干行通常是程序的注释, 注释以 “%” 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, 整个程序应按 MATLAB 标识符的要求命名文件名。

2. 函数文件

函数文件是用来定义子程序的。它与脚本文件的主要区别有 3 点:

(1) 由 function 起头, 后跟的函数名要与文件名相同。

(2) 一般有输入输出变量, 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 3.2 函数举例。

编写表示 $y = x^2 + 2x + 3$ 的 MATLAB 函数, 并计算在 $x = -6, -5, \dots, 6$ 处的函数值。

```
function y=myfun1(x);
```

```
y=x.^2+2*x+3;
```

把上述函数的文件名命名为 myfun1.m, 并编写调用函数的 M 文件 ex32_1.m 如下:

```
clc, clear %清屏并清理内存空间
```

```
x=-6:6; y=myfun1(x)
```

上述函数虽然只有两行, 但也要放在一个文件中, 调用函数 myfun1 必须在命令窗口中或其他文件中, 使用起来不太方便。在这种情形下, 我们可以定义匿名函数如下

```
y=@(x) x.^2+2*x+3;
```

可以把匿名函数的定义和它的调用语句放在同一个文件 ex32_2.m 中, 程序如下:

```
clc, clear %清屏并清理内存空间
```

```
y=@(x) x.^2+2*x+3;
```

```
x0=-6:6; y0=y(x0)
```

匿名函数的定义, 和它的调用语句可以放在同一个文件中, 使用起来很方便。除非函数要有两个返回值, 无法使用匿名函数外, 其他情形下, 我们都是使用匿名函数。

3.2.2 MATLAB 流程控制结构

1. if 条件结构

if 条件结构是实现分支结构程序最常用的一种语句, 其使用方式如下:

if 表达式

语句组 1

elseif

语句组 2

else

语句组 3

end

2. switch-case 开关结构

switch-case 开关结构的使用格式为:

```
switch 开关表达式
    case 表达式 1
        语句组 1
    case 表达式 2
        语句组 2
    .....
    case 表达式 n
        语句组 n
    otherwise
        语句组 n+1
end
```

3.for 循环结构

for 语句是一种基本的实现循环结构的语句, 能够以确定的次数执行某一段程序。for 语句的格式如下:

```
for 指标=数组值
    语句组
end
```

for 语句的使用和其他程序语言是类似的。

4.while 循环结构

while 循环结构的使用格式如下:

```
while 表达式
    语句组
end
```

5.程序设计举例

(1) 二分法

例 3.3 用二分法求方程

$$x^3 + 1.1x^2 + 0.9x - 1.4 = 0$$

的实根的近似值, 使误差不超过 10^{-3} 。

计算的 MATLAB 程序如下:

```
clc, clear %清屏和清理内存空间
fx=@(x)x.^3+1.1*x.^2+0.9*x-1.4; %定义匿名函数
a=0; b=1;
c=(a+b)/2
while abs(b-a)>0.001;
    if fx(a)*fx(c)>0
        a=c;
    else
        b=c;
    end
    c=(a+b)/2
end
```

直接调用 MATLAB 工具箱函数的程序如下:

```
clc, clear %清屏和清理内存空间
fx=@(x)x.^3+1.1*x.^2+0.9*x-1.4; %定义匿名函数
x1=fsolve(fx,0) %调用工具箱的命令
x2=fzero(fx,0) %调用工具箱的另一个命令
```

(2) 切线法

在端点 $(x_0, f(x_0))$ 做曲线 $y = f(x)$ 的切线, 切线的方程为

$$y - f(x_0) = f'(x_0)(x - x_0),$$

令 $y = 0$, 从上式中解出 x , 就得到切线与 x 轴交点的横坐标为

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

切线法的迭代公式为

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}.$$

例 3.4 用切线法求方程

$$x^3 + 1.1x^2 + 0.9x - 1.4 = 0$$

的实根的近似值，使误差不超过 10^{-4} 。

计算的 MATLAB 程序如下：

```
clc, clear %清屏和清理内存空间
format long g %长小数的显示格式
fx=@(x)x.^3+1.1*x.^2+0.9*x-1.4; %定义匿名函数
df=@(x)3*x.^2+2.2*x+0.9;
x0=1; x1=x0-fx(x0)/df(x0)
while abs(x1-x0)>0.0001;
    x0=x1;
    x1=x0-fx(x0)/df(x0)
end
x2=fzero(fx,0) %调用工具箱的命令
format %等价于 format short, 恢复到短小数的显示格式
```

3.3 MATLAB 绘图命令

3.3.1 二维绘图命令

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。

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')`

其中 x,y 是同维数的向量和矩阵，当 x,y 是 n 个元素的向量时， x 表示 n 个点的横坐标， y 表示这 n 个点的纵坐标，上述 `plot` 命令顺次用线段把这 n 个点连接起来，即所画的图形是 $n-1$ 条线段的折线；当 x,y 是 m 列的矩阵时，每一列对应一条折线的坐标，`plot` 命令画出 m 条折线。

字符串 `'color_point_linestyle'` 完成对线色、数据点和线型 3 个参数的设置，这 3 个参数的控制没有先后次序。线色 (`r-red`, `g-green`, `b-blue`, `w-white`, `k-black`, `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。

`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)`

表示把屏幕分成 r 行、 c 列个子窗口，即 $r \times c$ 个子窗口，激活 p 号子窗口，窗口的排列次序是逐行排列的。

`subplot(r,c,p)` 使用时可以省略参数中间的两个逗号“`,`”，例如 `subplot(131)` 表示把屏幕分成 3 个子窗口（排成一行），激活 1 号子窗口。

画二维曲线图时，可以使用 3 种方式画图：（1）用描点画图命令 `plot`，（2）用函数画图命令 `fplot`，（3）用“Easy-to-use”函数画图命令 `ezplot`，该命令既可以执行符号函数画图，也可以执行匿名函数画图。下面通过例子说明这 3 种画图命令。

例 3.5 分别用 `plot`, `fplot`, `ezplot` 画出抛物线 $y = x^2$ 。

```
clc, clear
x=-6:0.6:6; y=x.^2;
subplot(141), plot(x,y)
```

```
yy=@(x)x.^2; %定义抛物线的匿名函数
subplot(142), fplot(yy,[-6,6])
subplot(143), ezplot(yy,[-6,6])
subplot(144), ezplot('y=x^2',[-6,6]) %符号函数画图
```

例 3.6 为了测量刀具的磨损速度，我们做这样的实验：经过一定时间（如每隔一小时），测量一次刀具的厚度，得到一组实验数据 $(t_i, y_i)(i=1,2, \dots, 8)$ 如表 3.4 所示。试画出观测数据的散点图。

表 3.4 实验观测数据

t_i	0	1	2	3	4	5	6	7
y_i	27.0	26.8	26.5	26.3	26.1	25.7	25.3	24.8

```
clc, clear
t=0:7; y=[27.0 26.8 26.5 26.3 26.1 25.7 25.3 24.8];
plot(t,y,'P')
```

例 3.7 已知随机变量 X 服从泊松分布

$$P\{X = k\} = \frac{\lambda^k}{k!} e^{-\lambda}, k = 0, 1, 2, \dots,$$

其中， $\lambda = 2$ ，画出其分布律当 $k = 0, 1, 2, \dots, 6$ 的散点图。

```
clc, clear, close all
k=0:6;
y=2.^k./factorial(k)*exp(-2)
plot(k,y,'*')
或者直接调用 MATLAB 工具箱的分布函数命令，程序如下：
clc, clear, close all
k=0:6; y=poisspdf(k,2);
plot(k,y,'*')
```

例 3.8（显函数） 画出箕舌线 $y = \frac{8}{x^2 + 4}$ 。

```
clc, clear, close all %关闭所有的图形窗口
f2=@(x)8./(x.^2+4); fplot(f2,[-6,6])
figure, ezplot('y=8/(x^2+4)',[-6,6,0,2]), title('')
```

例 3.9（隐函数） 画出笛卡儿叶形线 $x^3 + y^3 - 3xy = 0$ 。

```
ezplot('x^3+y^3-3*x*y=0',[-3,3])
title('') %不显示图题
```

例 3.10（参数方程） 画出星形线 $\begin{cases} x = 2\cos^3 \theta, \\ y = 2\sin^3 \theta. \end{cases}$

```
clc, clear, close all
x=@(t)2*cos(t).^3; y=@(t)2*sin(t).^3;
fplot(x,y)
```

3.3.2 图形标识

MATLAB 允许对图形对象进行文字标注，即图形标识。常用的图形标识命令有：

```
title('s') %添加图形标题；
xlabel('s') %横坐标命名；
ylabel('s') %纵坐标命名；
text(x,y,'s') %在位置 (x,y) 处添加说明文字。
```

命令

`legend('s1', 's2',..., 'Location', 'Location 参数值')`

在图形窗中开启一个注释小窗口，依据绘图的先后次序，依次输出字符串对各条曲线进行注释说明. Location 确定注释窗口的位置，Location 部分参数值含义如表 3.5 所示。

表 3.5 Location 参数值

参数值	位置	参数值	位置
best	自动放置在图形内部的最佳位置	northoutside	图形外部的上方
northeast	图形内部的右上方	southeast	图形内部的右下方
northwest	图形内部的左上方	southwest	图形内部的左下方

例 3.11 在同一个图形界面上分别画出 $y = 2\cos x$ ， $y = \sin(2x)$ 的图形。

```
clc, clear, close all
x=-6:0.1:6;
y1=2*cos(x); y2=sin(2*x);
plot(x,y1,'-r'), hold on %图形保持
plot(x,y2,'P-')
legend('余弦曲线','正弦曲线','Location','best')
```

3.3.3 三维画图命令

MATLAB 也提供了一些三维基本绘图命令，如三维曲线命令 `plot3`，三维网格图命令 `mesh` 和三维表面图命令 `surf`。

`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。

命令 `mesh(x,y,z)`画网格曲面。这里 x,y,z 分别表示数据点的横坐标、纵坐标、竖坐标，如果 x 和 y 是向量， x 是 m 维的向量， y 是 n 维的向量，则 z 是 $n \times m$ 的矩阵。 x,y,z 也可以都是同维数的矩阵。命令 `mesh(x,y,z)`将该数据点在空间中描出，并连成网格。

命令 `surf(x,y,z)`画三维表面图，这里 x,y,z 分别表示数据点的横坐标、纵坐标、竖坐标。

已知曲线或曲面的函数关系，提倡使用 `ezplot3`，`ezmesh`，`ezsurf` 等命令画图。

例 3.12 画出三维曲线 $x = t \cos t$ ， $y = t \sin t$ ， $z = t$ 的图形。

```
clc, clear, close all
t=0:0.01:100; x=t.*cos(t); y=t.*sin(t);
plot3(x,y,t)
```

例 3.13 二次曲面

(1) (隐函数) 椭圆锥面 $\frac{x^2}{4} + \frac{y^2}{2} = z^2$;

它的参数方程为
$$\begin{cases} x = 2z \cos t, \\ y = \sqrt{2}z \sin t, \\ z = z. \end{cases}$$

```
x=@(t,z)2*z.*cos(t); y=@(t,z)sqrt(2)*z.*sin(t);
z=@(t,z)z; ezsurf(x,y,z,[0,2*pi,-6,6])
```

(2) (隐函数) 椭球面 $\frac{x^2}{4} + \frac{y^2}{3} + \frac{z^2}{2} = 1$;

它的参数方程
$$\begin{cases} x = 2 \cos t \cos s \\ y = \sqrt{3} \sin t \cos s \\ z = \sqrt{2} \sin s \end{cases}$$

```
x=@(t,s)2*cos(t).*cos(s); y=@(t,s)sqrt(3)*sin(t).*cos(s);
z=@(t,s)sqrt(2)*sin(s); ezsurf(x,y,z,[0,2*pi,-pi/2,pi/2])
```

或

```
t=0:0.1:2*pi; s=-pi/2:0.1:pi/2;
```



```
[t,s]=meshgrid(t,s);
x=2*cos(t).*cos(s); y=sqrt(3)*sin(t).*cos(s);
z=sqrt(2)*sin(s); surf(x,y,z)
(3) (显函数) 双曲抛物面或马鞍面  $z = xy$ 。
ezmesh(@ (x,y)x.*y,[-6,6,-5,5])
标注字符是 LaTeX 格式时, 程序如下:
ezmesh(@ (x,y)x.*y,[-6,6,-5,5])
xlabel('$x$', 'Interpreter', 'Latex')
ylabel('$y$', 'Interpreter', 'Latex')
zlabel('$z$', 'Interpreter', 'Latex', 'Rotation', 0)
title('$xy$', 'Interpreter', 'Latex')
```

例 3.14 莫比乌斯带是一种拓扑学结构, 它只有一个面和一个边界, 是 1858 年由德国数学家、天文学家莫比乌斯和约翰·李斯丁独立发现的。其参数方程为

$$\begin{cases} x = \left(2 + \frac{s}{2} \cos \frac{t}{2} \right) \cos t, \\ y = \left(2 + \frac{s}{2} \cos \frac{t}{2} \right) \sin t, \\ z = \frac{s}{2} \sin \frac{t}{2}, \end{cases}$$

其中, $0 \leq t \leq 2\pi$, $-1 \leq s \leq 1$ 。绘制莫比乌斯带。

```
clc, clear
x=@(s,t)(2+s/2.*cos(t/2)).*cos(t);
y=@(s,t)(2+s/2.*cos(t/2)).*sin(t);
z=@(s,t)s/2.*sin(t/2);
ezmesh(x,y,z,[-1,1,0,2*pi]), title('')
view(-40,60)
```

3.3.4 四维空间画图命令

MATLAB 表现复变函数 (四维) 的方法是用三维空间坐标再加上颜色, 类似于地球仪用颜色表示海洋与高山。MATLAB 画复变函数的图形的命令主要有以下 3 个。

1. cplxgrid 构建一个极坐标的复数数据网格

$z = \text{cplxgrid}(m)$; %产生 $(m+1) * (2*m+1)$ 的极坐标下的复数数据网格, 最大半径为 1 的圆面。

在命令窗口输入 `type cplxgrid`, 屏幕将显示它的源程序, 可以参考它来编写自己的专用程序。

2. cplxmap 对复变函数做图

`cplxmap(z,f(z),[optional bound])` %画复变函数的图形, 可选项用以选择函数的做图范围。

`cplxmap` 做图时, 以 xy 平面表示自变量所在的复平面, 以 z 轴表示复变函数的实部, 颜色表示复变函数的虚部。

3. cplxroot 画复数的 n 次根函数曲面

`cplxroot(n,m)` %`cplxroot(n,m)` 是使用 $m \times m$ 数据网格画复数 n 次根的函数曲面。如果不指定 m 值, 则使用默认值 $m=20$ 。

例 3.15 画复变函数 $f(z) = z^3$ 的图形。记

$$f(z) = u + iv = (x + iy)^3 = x^3 - 3xy^2 + (3x^2y - y^3)i,$$

$f(z)$ 的实部 $u = x^3 - 3x^2y$, 虚部 $v = 3x^2y - y^3$ 。

下面分别使用复变函数画图命令和实变函数画图命令进行了比较, 画图的 MATLAB 程序如下:

```
clc, clear
z=cplxgrid(30); %生成网格数据
```

```
subplot(121), cplxmap(z,z.^3), colorbar %直接用工具箱的画图命令
title('工具箱画图效果')
%以下使用实函数 surf 命令画图，和上面做一下对比
r=linspace(0,1,50); t=linspace(0,2*pi,50);
[t,r]=meshgrid(t,r); %生成极坐标的网络数据
[x,y]=pol2cart(t,r); %极坐标转化为直角坐标
u=x.^3-3*x.*y.^2; v=3*x.^2.*y-y.^3; %计算复函数的实部和虚部
subplot(122), surf(x,y,u,v), colorbar %使用实函数画图
title('surf 命令画图效果')
```

例 3.16 绘制下列函数的图形

(1) $\sin z$;
 clc, clear, z=2*cplxgrid(20); %生成网格数据，最大半径为 2
 cplxmap(z,sin(z)), colorbar
 (2) $\ln z$ 。
 clc, clear
 z=cplxgrid(30); %生成网格数据
 cplxmap(z,log(z)), colorbar %直接用工具箱的画图命令

3.3.5 其他画图命令

1.极坐标下画图命令 polarplot, ezpolar

polarplot 的调用格式为

polarplot(theta,rho)

其中 theta 是极角的离散值，rho 是极径的离散值。

例 3.17 画出心形线

$$\rho = 2(1 - \cos \theta).$$

```
t=0:0.01:2*pi;
r=2*(1-cos(t)); polarplot(t,r)
```

ezpolar 的调用格式为

ezpolar(fun,[a,b])

其中 fun 可以是定义曲线的（匿名）函数或符号函数，[a,b]是极角的取值范围。

例 3.18 画出四叶玫瑰线

$$\rho = 3\sin 2\theta.$$

```
ezpolar(@(t)3*sin(2*t),[0,2*pi])
```

2.画等高线命令

MATLAB 画等高线的命令有 contour, contourf, contour3, contourc, 标注等高线的命令是 clabel. 其中 contour 的调用格式如下:

contour(x,y,z) %x,y 是平面 $z=0$ 上点的 x 坐标和 y 坐标, z 为相应点的高度值矩阵, 画出等高线;

contour(x,y,z,n) %画 n 条等高线;

contour(x,y,z,v) %在指定的高度 v 上画等高线。

其他画等高线命令的使用方式是类似的。

例 3.19 已知平面区域 $0 \leq x \leq 1400$, $0 \leq y \leq 1200$ 步长间隔为 100 的网格节点高程数据见表 3.6 (单位: m).

表 3.6 高程数据表

y / x	0	100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
1200	1350	1370	1390	1400	1410	960	940	880	800	690	570	430	290	210	150
1100	1370	1390	1410	1430	1440	1140	1110	1050	950	820	690	540	380	300	210
1000	1380	1410	1430	1450	1470	1320	1280	1200	1080	940	780	620	460	370	350
900	1420	1430	1450	1480	1500	1550	1510	1430	1300	1200	980	850	750	550	500

800	1430	1450	1460	1500	1550	1600	1550	1600	1600	1600	1550	1500	1500	1550	1550
700	950	1190	1370	1500	1200	1100	1550	1600	1550	1380	1070	900	1050	1150	1200
600	910	1090	1270	1500	1200	1100	1350	1450	1200	1150	1010	880	1000	1050	1100
500	880	1060	1230	1390	1500	1500	1400	900	1100	1060	950	870	900	936	950
400	830	980	1180	1320	1450	1420	400	1300	700	900	850	810	380	780	750
300	740	880	1080	1130	1250	1280	1230	1040	900	500	700	780	750	650	550
200	650	760	880	970	1020	1050	1020	830	800	700	300	500	550	480	350
100	510	620	730	800	850	870	850	780	720	650	500	200	300	350	320
0	370	470	550	600	670	690	670	620	580	450	400	300	100	150	250

(1) 画出该区域的等高线.

(2) 画出该区域的三维表面图.

画等高线及三维表面图的 MATLAB 程序如下:

```
clc, clear
```

```
a=load('data3.txt'); %把表 2-8 中的高程数据保存在纯文本文件 data28.txt 中
```

```
x0=0:100:1400; y0=1200:-100:0;
```

```
subplot(1,2,1), c=contour(x0,y0,a,7); clabel(c) %画 7 条等高线, 并标注等高线  
title('等高线图')
```

```
subplot(1,2,2), surf(x0,y0,a), title('三维表面图')
```

所画出的图形见图 3.1。

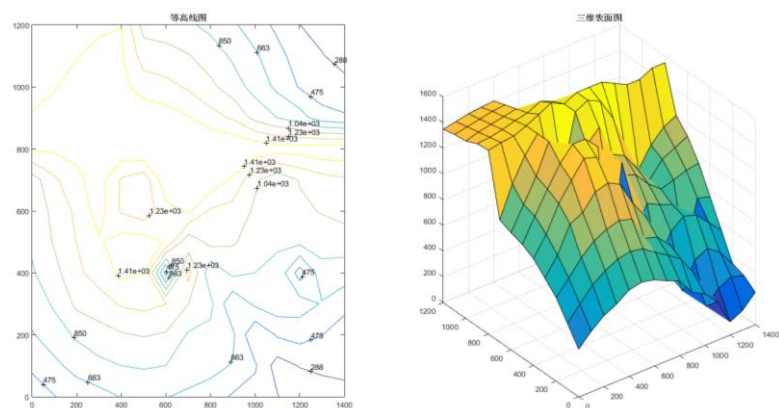


图 3.1 等高线图及三维表面图

3.4 求代数方程的符号解和数值解

符号运算又称计算机代数, 通俗地讲就是用计算机推导数学公式, 如对表达式进行因式分解、化简、微分、积分、解代数方程、求解常微分方程等。与数值运算相比, 符号计算存在以下的特点: (1) 运算以推理方式进行, 因此不受截断误差和累积误差问题的影响; (2) 符号计算的速度比较慢。

3.4.1 符号对象的创建

在 MATLAB 中, 提供了两个创建符号对象的函数, 即 `sym` 和 `syms` 函数。

1. `sym` 函数

`sym` 函数用来建立单个符号量, 函数的调用格式为:

`var=sym('var')` %创建一个符号变量 `var`.

`symexpr=sym(h)` %通过匿名函数的函数句柄创建符号表达式.

`A=sym('a', [m,n])` %创建一个 $m \times n$ 矩阵的符号变量.

2. `syms` 函数

`sym` 函数一次只能定义一个符号变量，使用不方便. MATLAB 提供了另一个函数 `syms`，一次可以定义多个符号变量. `syms` 函数的调用格式为：

`syms var1 ... varN` %创建符号变量 `var1 ... varN`。

`syms var1 ... varN set` %创建符号变量 `var1 ... varN`，并指定符号对象的格式。

例 3.20 利用 `syms` 创建符号对象。

```
clc, clear
```

```
syms a %定义符号变量 a
```

```
syms x y integer %定义整型符号变量 x 和 y
```

```
syms s(t) f(x,y) %定义两个符号函数
```

3.4.2 代数方程（组）的符号解

MATLAB 符号运算工具箱能够解简单的线性方程、非线性方程及简单的代数方程组。

在 MATLAB 中，提供了 `solve` 函数求解符号代数方程或方程组，其调用格式如下：

`S=solve(eqn,var)` %求表达式 `eqn` 的代数方程，求解变量为 `var`。

`S=solve(eqn,var,Name,Value)` %求表达式 `eqn` 的代数方程，求解变量为 `var`，其中指定一个或多个属性值名及其对应的属性值。

`[y1,...,yN]=solve(eqns,vars)` %求表达式 `eqns` 的代数方程组，求解变量组为 `vars`。

例 3.21 利用 `solve` 求符号代数方程的解.

(1) $ax^2 + bx = c$ ，其中 x 为未知数，(2) $ax^2 + bx = c$ ，其中 a 为未知数.

```
clc, clear, syms x a b c
```

```
eq=a*x^2+b*x==c %定义方程
```

```
xx=solve(eq) %求方程符号解，这里没有指明变量，使用 symvar(eq,1)确定的变量 x
```

```
a1=solve(eq,a)
```

```
a2=solve(a*x^2+b*x==c,a)
```

```
a3=solve(a*x^2+b*x-c,a) %这里我们给出了方程的另一种写法
```

例 3.22 求方程组

$$\begin{cases} x_1^2 + x_2^2 = 1, \\ x_1 = x_2, \end{cases}$$

的符号解。

```
clc, clear
```

```
syms x1 x2
```

```
[x1,x2]=solve(x1^2+x2^2==1,x1==x2)
```

3.4.3 代数方程（组）的数值解

1. 线性方程组的数值解

线性方程组 $Ax = b$ 的求解命令为 $x = \text{pinv}(A)*b$ ，当 A 列满秩时，可以使用命令 $x = A \setminus B$ 。

- (1) 数学上唯一解，MATLAB 也给出唯一解；
- (2) 数学上无解，MATLAB 给出最小二乘解；
- (3) 数学上无穷多解，MATLAB 给出最小范数解。

例 3.23 求线性方程组

$$\begin{cases} x_1 + x_2 = 1, \\ 2x_1 + 2x_2 = 2. \end{cases}$$

的最小范数解。

```
clc, clear
a=[1,1;2,2]; b=[1;2];
x=pinv(a)*b
```

例 3.24 求线性方程组

$$\begin{cases} x_1 + x_2 = 1, \\ 2x_1 + 2x_2 = 3. \end{cases}$$

的最小二乘解。

```
clc, clear
a=[1,1;2,2]; b=[1;3];
x=pinv(a)*b %求矛盾方程组的最小二乘解
```

2. 非线性方程的数值解

- (1) 求多项式的所有根命令 **roots**

roots 的调用格式为

$r = \text{roots}(p)$ %求多项式的所有根，这里 p 是一个向量，是由多项式从高次幂系数到常数项所组成的向量，返回值 r 是多项式的所有根。

例 3.25 求方程

$$x^{980} - 5.01x^{979} + 7.398x^{978} - 3.388x^{977} - x^3 + 5.01x^2 - 7.398x + 3.388 = 0$$

的所有实根。

解 求得的实数根有 4 个，分别为 1，1，1.21，2.8。

计算的 MATLAB 程序如下：

```
clc, clear
a=zeros(1,980); %生成全 0 的行向量
a(1)=-7.398; a(2)=5.01; a(3)=-1; a(977)=-3.388;
a(978)=7.398; a(979)=-5.01; a(980)=1;
b=[3.388,a]; %加入常数项
c=b([end:-1:1]); %向量进行逆序变换
r=roots(c) %求所有的复数根
ind=find(imag(r)==0) %找实根的下标
rr=r(ind) %显示所有的实根
```

我们可以换一种方式生成多项式的系数向量，MATLAB 程序如下：

```
clc, clear, syms x
y=x^980 - 5.01*x^979 + 7.398*x^978 - 3.388*x^977 - 1.0*x^3 + 5.01*x^2 - 7.398*x + 3.388;
p=sym2poly(y); %把符号多项式转换为多项式的系数向量
r=roots(p) %求所有的根
ind=find(imag(r)==0), rr=r(ind) %显示实数根
```

(2) 求非线性函数的零点或非线性方程的解

i) fzero 函数

MATLAB 求非线性方程的根，即求一元非线性函数零点的命令为 fzero，调用格式为：

$x=fzero(fun,x0)$ %当 $x0$ 为标量时，求该点附近的函数零点；当 $x0$ 为区间时，求该区间内的函数零点。

ii) fsolve 函数

fsolve 不仅可以求解非线性方程的解，而且也可以求解非线性方程组的解。其调用格式与 fzero 是类似的，这里就不赘述了。

注 3.2 fzero 和 fsolve 只能求给定初值附近的一个解。

例 3.26 (续例 3.25) 求方程

$$x^{980} - 5.01x^{979} + 7.398x^{978} - 3.388x^{977} - x^3 + 5.01x^2 - 7.398x + 3.388 = 0$$

在给定初值 0.5 附近的一个实根。

```
clc, clear
fx=@(x)x^980-5.01*x^979+7.398*x^978-3.388*x^977-x^3+5.01*x^2-7.398*x+3.388;
x1=fzero(fx,0.5) %求初值 0.5 附近的实根
x2=fsolve(fx,0.5) %未求得一个数值解
```

3.非线性方程组的数值解

fsolve 求解非线性方程组时，首先把非线性方程组写成

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

的形式，其中 \mathbf{x} 为向量， $\mathbf{F}(\mathbf{x})$ 为向量函数。

fsolve 的 MATLAB 调用格式为

$\mathbf{x} = \text{fsolve}(\text{fun}, \mathbf{x0}, \text{options})$

其中，fun 是定义向量函数 $\mathbf{F}(\mathbf{x})$ 的 MATLAB 函数或匿名函数， $\mathbf{x0}$ 是 \mathbf{x} 的初始向量，options 是 MATLAB 求解过程的一些参数设置。

例 3.27 (续例 3.22) 求方程组

$$\begin{cases} x_1^2 + x_2^2 = 1, \\ x_1 = x_2, \end{cases}$$

的一组数值解。

```
clc, clear
fx=@(x)[x(1)^2+x(2)^2-1; x(1)-x(2)]; %定义方程组的匿名向量函数
x=fsolve(fx,rand(1,2)) %初始值是一个随机向量
```

3.非线性方程组的最小二乘解

对于非线性方程组

$$\begin{cases} f_1(\mathbf{x}) = 0, \\ f_2(\mathbf{x}) = 0, \\ \dots \\ f_n(\mathbf{x}) = 0. \end{cases} \quad (1)$$

其中 \mathbf{x} 为 m 维向量，一般地， $n > m$ ，且方程组(1)是矛盾方程组，有时要求方程组(1)的最小二乘解，即求下面的多元函数

$$\delta(\mathbf{x}) = \sum_{i=1}^n f_i^2(\mathbf{x}) \quad (2)$$

的最小值。MATLAB 求非线性方程组最小二乘解的命令为

$\text{lsqnonlin}(\text{fun}, \mathbf{x0})$,

其中 fun 是定义向量函数

$$[f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ \cdots \ f_n(\mathbf{x})]^T$$

的匿名函数的返回值, $\mathbf{x0}$ 为 \mathbf{x} 的初始值。

例 3.28 已知 4 个观测站的位置坐标 $(x_i, y_i) (i=1, 2, 3, 4)$, 每个观测站都探测到距未知信号的距离 $d_i (i=1, 2, 3, 4)$, 已知数据见表 3.7, 试定位未知信号的位置坐标 (x, y) 。

表 3.7 观测站的位置坐标及探测到的距离

	1	2	3	4
x_i	245	164	192	232
y_i	442	480	281	300
d_i	126.2204	120.7509	90.1854	101.4021

解 位置信号的位置坐标 (x, y) 满足非线性方程组:

$$\begin{cases} \sqrt{(x_1 - x)^2 + (y_1 - y)^2} - d_1 = 0, \\ \sqrt{(x_2 - x)^2 + (y_2 - y)^2} - d_2 = 0, \\ \sqrt{(x_3 - x)^2 + (y_3 - y)^2} - d_3 = 0, \\ \sqrt{(x_4 - x)^2 + (y_4 - y)^2} - d_4 = 0. \end{cases} \quad (3)$$

显然方程组(3)是一个矛盾方程组, 必须求方程组(3)的最小二乘解。可以把问题转化为求如下多元函数

$$\delta(x, y) = \sum_{i=1}^4 \left(\sqrt{(x_i - x)^2 + (y_i - y)^2} - d_i \right)^2 \quad (4)$$

的最小点问题。

利用 MATLAB 的 lsqnonlin 命令求得 $x=149.5089$, $y=359.9848$ 。

计算的 MATLAB 程序如下:

```
clc, clear
xi=[245 164 192 232]';
yi=[442 480 281 300]';
d=[126.2204 120.7509 90.1854 101.4021]';
fx=@(x)sqrt((xi-x(1)).^2+(yi-x(2)).^2)-d; %定义误差向量函数的匿名函数
x1=lsqnonlin(fx,rand(1,2))
gx=@(x)sum(fx(x).^2); %定义误差平方和函数的匿名函数
x2=fminsearch(gx,rand(1,2)) %求多元函数的极小点
x3=fminunc(gx,rand(1,2)) %求多元函数的极小点
```

习题 3

3.1 在同一个图形界面上画出如下 3 个函数的图形并进行标注。

$$y = \cosh x, \quad y = \sinh x, \quad y = \frac{1}{2} e^x.$$

3.2 画出 Γ 函数 $\Gamma(x) = \int_0^{+\infty} e^{-t} t^{x-1} dt$ 的图形。

3.3 在同一个图形界面中分别画出 6 条曲线

$$y = kx^2 + 2k, k = 1, 2, \dots, 6.$$

3.4 把屏幕开成 2 行 3 列 6 个子窗口, 每个子窗口画一条曲线, 画出曲线

$$y = kx^2 + 2k, k = 1, 2, \dots, 6.$$

3.5 分别画出下列二次曲面。

(1) 单叶双曲面 $\frac{x^2}{4} + \frac{y^2}{10} - \frac{z^2}{8} = 1$;

(2) 双叶双曲面 $\frac{x^2}{4} - \frac{y^2}{12} - \frac{z^2}{8} = 1$;

(3) 椭圆抛物面 $\frac{x^2}{4} + \frac{y^2}{6} = z$.

3.6 附件 1: 区域高程数据.xlsx 给出了某区域 43.65×58.2 (km) 的高程数据, 画出该区域的三维网格图和等高线图, 在 A (30, 0) 和 B (43, 30) (单位: km) 点处建立了两个基地, 在等高线图上标注出这两个点。

3.7 先判断下列线性方程组解的情况, 然后求对应的唯一解、最小二乘解或最小范数解。

(1)
$$\begin{cases} 4x_1 + 2x_2 - x_3 = 2, \\ 3x_1 - x_2 + 2x_3 = 10, \\ 11x_1 + 3x_2 = 8; \end{cases}$$

(2)
$$\begin{cases} 2x + 3y + z = 4, \\ x - 2y + 4z = -5, \\ 3x + 8y - 2z = 13, \\ 4x - y + 9z = -6. \end{cases}$$

3.8 求方程组的符号解和数值解

$$\begin{cases} x^2 - y - x = 3, \\ x + 3y = 2. \end{cases}$$

3.9 已知 $f(x) = (|x+1| - |x-1|)/2 + \sin x$, $g(x) = (|x+3| - |x-3|)/2 + \cos x$, 求下列方程组的数值解。

$$\begin{cases} 2x_1 = 3f(y_1) + 4g(y_2) - 1, \\ 3x_2 = 2f(y_1) + 6g(y_2) - 2, \\ y_1 = f(x_1) + 3g(x_2) - 3, \\ 5y_2 = 4f(x_1) + 6g(x_2) - 1. \end{cases}$$

3.10 已知 $f(x) = (|x+1| - |x-1|)/2 + \sin x$, $g(x) = (|x+3| - |x-3|)/2 + \cos x$, 求下列超定(矛盾)方程组的最小二乘解。

$$\begin{cases} 2x_1 = 3f(y_1) + 4g(y_2) - 1, \\ 3x_2 = 2f(y_1) + 6g(y_2) - 2, \\ y_1 = f(x_1) + 3g(x_2) - 3, \\ 5y_2 = 4f(x_1) + 6g(x_2) - 1, \\ x_1 + y_1 = f(y_2) + g(x_2) - 2, \\ x_2 - 3y_2 = 2f(x_1) - 10g(y_1) - 5. \end{cases}$$