

附录 B MATLAB 简介

MATLAB(矩阵实验室, Matrix Laboratory 的简称)是一款著名的数学软件,由美国 MathWorks 公司出品。MATLAB 可以进行矩阵运算、绘制函数和数据、实现算法、创建用户界面、连接其他编程语言的程序等,主要应用于工程计算、控制设计、信号处理与通信、图像处理、信号检测、金融建模设计与分析等领域。MATLAB 对许多专门的领域都开发了功

能强大的模块集和工具箱(toolbox),诸如信号处理、图像处理、系统辨识、金融分析、地图工具、电力系统仿真等。事实上,MathWorks 公司还有另一个重要产品叫 Simulink,它原本是 MATLAB 中的一部分,目前已独立出来,成为实现动态系统建模与仿真、功能更丰富的一个独立软件。图 B-1 是 MATLAB 软件的启动画面。

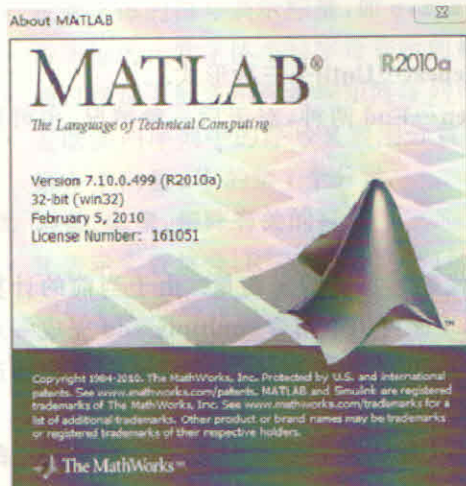


图 B-1 MATLAB 软件的版权页

除了 MATLAB,还有其他一些著名的数学软件,例如 Wolfram Research 公司的 Mathematica。Mathematica 的特点是符号计算能力非常突出,便于进行解析计算和公式的推导。相比这些数学软件,MATLAB 在数值计算,尤其是矩阵计算方法方面性能优越,因此了解、使用它的一些功能对学习数值分析等有关课程很有帮助。

MATLAB 是跨平台的软件,适用于 Windows、UNIX/Linux 以及 MAC 操作系统。MathWorks 公司每年进行两次产品发布,时间分别在 3 月和 9 月,以 a 和 b 区分版本号,比如 2010a 和 2010b。每个版本都包含 32 位和 64 位两种格式,安装时会根据计算机的情况自动选择。本附录以在 32 位 PC、Windows 7 环境下运行的 MATLAB 2010a 为例介绍 MATLAB 的基本功能。

B.1 用户界面

MATLAB 软件(可执行程序为 matlab.exe)启动后,用户界面如图 B-2 所示,其中包含了多个不同功能的子窗口。在命令窗口中输入 edit 命令,将弹出程序编辑窗口,可在其中编写程序。在命令窗口中按 ↑ 键会出现以往输入过的命令,而输入命令的前几个字符再按 ↑ 键将从历史命令中找到匹配的命令。通过这个快捷键,可方便地调用以前输入过的命令,或在其基础上进行修改。在主程序界面中,通过菜单 Help|Product Help 可调出帮助文档,其中包含了丰富的信息。程序编辑窗口和帮助窗口示于图 B-3 中。

MATLAB 的命令窗口和程序编辑窗口中的字体可以通过菜单项 File|Preferences|Fonts 设置,用户可将字体调整为合适的大小。另外,通过菜单项 File|Set Paths,可设置命

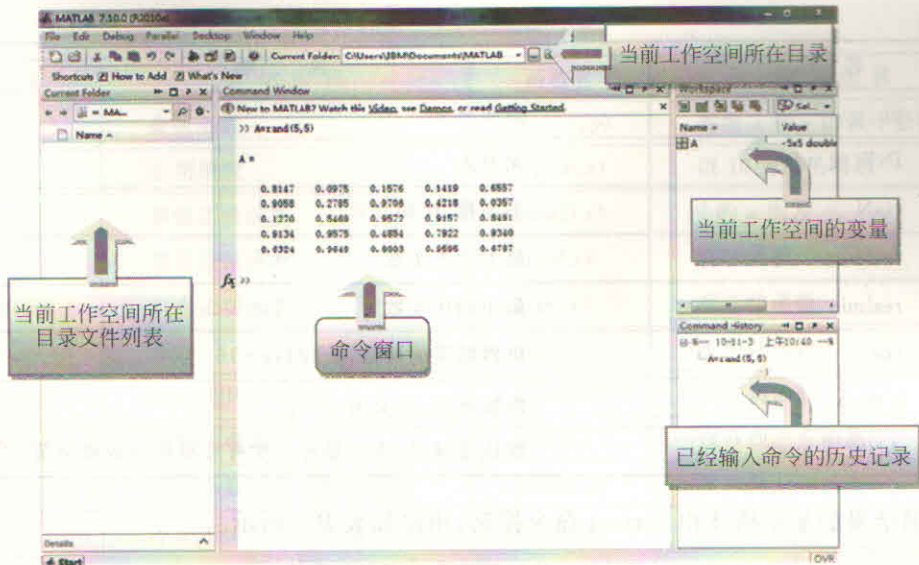


图 B-2 Windows 下的 MATLAB 使用界面

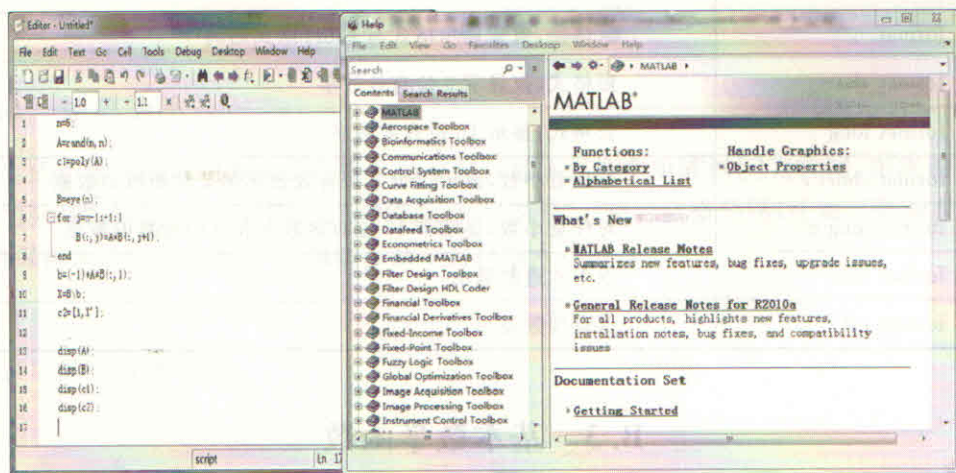


图 B-3 程序编辑窗口(左)和帮助窗口(右)

令路径,便于 MATLAB 自动搜索到不在当前目录下的文件和程序。

B.2 基本数据格式

默认情况下, MATLAB 采用 IEEE 双精度浮点数进行数据的存储与计算. MATLAB 中还有一些预先定义的保留字,用来表示常数(见表 B-1)。

MATLAB 变量名是以字母开头,由字母、数字和下划线组成的字符串,并且字母区别大小写. 表 B-1 中的特殊变量也可以当作一般的变量使用,比如 i 和 j . MATLAB 中的变量的基本数据格式是矩阵,单个变量也默认为单个元素的矩阵,并且不需要事先定义,一个变量在给它赋值时自动定义(但运算表达式中不允许出现未定义的变量). `clear` 命令可以清空所有已经合法定义了的变量。

表 B-1 MATLAB 中预定义的常数

名 称	说 明
pi	圆周率 π
INF(或 inf)	无穷大
NaN	不定量, 比如 0/0
realmax	最大的正实数
realmin	最小的正实数
eps	机器精度的两倍, $2.2204\text{e}-16$
i(或 j)	虚数单位, 定义为 $\sqrt{-1}$
ANS(或 ans)	默认变量名, 表示最近一次有效操作的运算结果

数据结果的显示格式由 format 命令控制, 用法如表 B-2 所示。

表 B-2 控制数据显示格式的命令

命 令	显 示 格 式
format	默认值, 数据显示格式与 short 格式相同
format short	短格式, 仅显示 5 位有效数字
format long	长格式, 显示 15 位有效数字
format short e	对任意小数, 显示仅带 5 位有效数字和 3 位指数的数据
format long e	对任意小数, 显示带 15 位有效数字和 3 位指数的数据
format hex	以十六进制显示
format rat	以有理数显示

B.3 基本数学函数

MATLAB 支持的部分常用数学函数如表 B-3 所示。这些函数的自变量可以是标量, 有可能是矩阵(包括向量)。

表 B-3 常用的数学函数

函数名	数 学 运 算	函数名	数 学 运 算
abs(x)	数或向量的模	gcd(x, y)	整数 x 和 y 的最大公约数
angle(z)	复数的辐角(弧度)	log(x)	以 e 为底的对数
real(z)	复数的实部	log2(x)	以 2 为底的对数
round(x)	四舍五入至最近整数	exp(x)	e 的 x 次方
floor(x)	不超过 x 的最大整数	sin(x)	正弦函数

函数名	数 学 运 算	函数名	数 学 运 算
$\cos(x)$	余弦函数	$\text{lcm}(x,y)$	整数 x 和 y 的最小公倍数
$\tan(x)$	正切函数	$\log_{10}(x)$	以 10 为底的对数
$\sinh(x)$	双曲正弦函数	$\text{pow2}(x)$	2 的 x 次方
$\cosh(x)$	双曲余弦函数	$\text{sign}(x)$	符号函数
$\tanh(x)$	双曲正切函数	$\text{asin}(x)$	反正弦函数
$\text{sqrt}(x)$	开方	$\text{acos}(x)$	反余弦函数
$\text{conj}(z)$	复数的共轭	$\text{atan}(x)$	反正切函数
$\text{imag}(z)$	复数的虚部	$\text{asinh}(x)$	反双曲正弦函数
$\text{fix}(x)$	舍去小数部分至最近整数	$\text{acosh}(x)$	反双曲余弦函数
$\text{ceil}(x)$	不小于 x 的最小整数	$\text{atanh}(x)$	反双曲正切函数

B.4 矩阵运算

B.4.1 矩阵的输入

生成小矩阵的常用方法是直接从命令窗口输入。整个矩阵以[]为首尾,矩阵的行与行之间用分号“;”或者回车相隔,每行中的元素用逗号“,”或者空格分隔(注意不要输入中文标点,确保输入法在英文状态)。例如在命令窗口输入:

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

命令窗口显示的结果为

```
a=
```

```
8 1 6
```

```
3 5 7
```

```
4 9 2
```

如果在上述命令后面加上分号“;”则同样生成矩阵,但并不在命令窗口显示结果。

另外,MATLAB 中矩阵的数据存储顺序是列优先的。例如在上述命令之后再输入:

```
>>a(4)
```

命令窗口显示的结果为

```
ans=
```

```
1
```

较大规模的矩阵可以使用 load 命令从文件读入,例如在当前工作空间的文件夹下创建文本文件 input.txt,包括如下内容:

```
1 2 3
```

```
4 5 6
7 8 9
```

然后在命令窗口输入:

```
>>b=load('input.txt')
```

命令窗口显示的结果为

```
b=
    1    2    3
    4    5    6
    7    8    9
```

类似地,Windows 下的 MATLAB 也可以从 Excel 数据表格中读取数据(只有较高版本的 MATLAB 才支持从 2007 版本的 Excel 创建的 `xlsx` 文档中读取数据,所以一般请将数据保存为 `xls` 格式),使用 `xlsread` 命令,例如在命令窗口输入:

```
>>a=xlsread('test.xls')
```

则读出 `test.xls` 文件中的数据表格,命令窗口显示的结果是:

```
a=
    1    2    3
    4    5    6
    7    8    9
```

还有一些命令可生成特殊矩阵,列于表 B-4 中.

表 B-4 生成特殊矩阵的命令

命 令	功 能
<code>zeros(m,n)</code>	生成 m 行 n 列全零矩阵
<code>ones(m,n)</code>	生成 m 行 n 列全 1 矩阵
<code>eye(m,n)</code>	生成 m 行 n 列对角线元素(行标等于列标)为 1 的矩阵
<code>rand(m,n)</code>	生成 m 行 n 列随机矩阵
<code>diag(x,d)</code>	生成一个以向量 x 的元素为距主对角线距离 d 的对角线元素的矩阵(主对角线以上 $d>0$ 反之则 $d<0$)

注意: `diag` 这个命令也可以反过来使用,即如果下面的命令中 A 是矩阵

```
>>x=diag(A,d)
```

那么结果是将矩阵 A 距主对角线距离 d 的对角线的元素存储在向量 x 中,这里的 x 默认为列向量.

向量(数组)是一种特殊的矩阵,可以用输入矩阵的方式建立. 常用的是数值为等差数列的数组,它方便地通过“:”来创建,例如:

```
>>x=-5:1:5
```

其中两个“:”之间的是等差间隔(若省略它表示间隔为 1). 执行该命令后,命令窗口显示的

结果为(得到一个行向量):

```
x=
-5 -4 -3 -2 -1 0 1 2 3 4 5
```

B.4.2 矩阵的代数运算

矩阵的加、减、乘等运算与标量的没什么区别,表 B-5 中列出一些常见的矩阵运算.

表 B-5 常见的矩阵运算

命 令	功 能	命 令	功 能
$[m,n]=\text{size}(A)$	m 为行数, n 为列数	$\text{rank}(A)$	矩阵的秩
$\det(A)$	方阵的行列式	$\text{eig}(A)$	求出矩阵 A 所有特征值
$\text{inv}(A)$	矩阵的逆	$\text{cond}(A)$	矩阵的条件数
A'	矩阵的共轭转置	$A.'$	元素转置不求共轭

关于稀疏矩阵和线性方程组求解,以及相关函数的使用,参见本书第 3 章有关内容. 下面介绍通过“:”符号取矩阵的部分元素的功能. 例如,命令

```
>>B=A(m:n,p:q)
```

得到的结果将是 A 的第 m 行到第 n 行,第 p 列到第 q 列的子矩阵. 而 $A(:,n)$ 和 $A(m,:)$ 分别表示矩阵 A 的第 n 列和第 m 行向量. 例如,对于前面定义的矩阵 a ,输入:

```
>>b=a(2,:)
```

得到结果为:

```
b=
4 5 6
```

除了这些基本的运算,表 B-6 列出了一些较复杂的运算.

表 B-6 较复杂的矩阵算法的命令

命 令	功 能
$[V,J]=\text{jordan}(A)$	J 是 A 的约当标准型, V 是相似变换矩阵, $A=V * J * \text{inv}(V)$,默认复数域
$[L,U,P]=\text{lu}(A)$	求矩阵 A 的 LU 分解, $P * A=L * U$
$R=\text{chol}(A)$	Cholesky 分解, $A=R' * R$, R 为上三角矩阵
$[Q,R]=\text{qr}(A)$	QR 分解, $A=Q * R$
$[V,D]=\text{eig}(A)$	特征值分解, D 对角线元素为 A 的特征值, $A * V=V * D$
$[U,S,V]=\text{svd}(A)$	奇异值分解, $A=U * S * V'$

B.4.3 逐项运算符

MATLAB 中有个特殊的逐项运算符“.”,它的功能通过表 B-7 说明,其中 A 和 B 是同阶的矩阵.

表 B-7 逐项运算符

命 令	说 明
$P=A.*B$	得到和 A 同阶矩阵 P, $P(i,j)=A(i,j)*B(i,j)$
$Q=A.^B$	得到和 A 同阶矩阵 Q, $Q(i,j)=A(i,j)^{B(i,j)}$
$R=A./B$	得到和 A 同阶矩阵 R, $R(i,j)=A(i,j)/B(i,j)$
$S=A.\backslash B$	得到和 A 同阶矩阵 S, $S(i,j)=B(i,j)/A(i,j)$

值得一提的是,线性代数中还有一些方阵函数,例如指数函数

$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k,$$

在 MATLAB 中它们对应于 expm、logm、sqrtm 等命令,其名字是对应的标量函数后面加上 m,例如输入:

```
>>A=[1,0;0,1];
```

则

```
>>expm(A)
```

结果为

```
ans=
    2.7183    0
    0        2.7183
```

而输入:

```
>>exp(A)
```

的结果为(注意这个命令的不同含义)

```
ans=
    2.7183    1.0000
    1.0000    2.7183
```

B.4.4 向量的函数

向量是特殊的矩阵,表 B-8 列出了一些经常用于向量的函数.

表 B-8 常用于向量的函数

命 令	功 能	命 令	功 能
min(x)	向量 x 元素的最小值	sort(x)	对向量 x 的元素升序排序
max(x)	向量 x 元素的最大值	norm(x)	向量 x 的模
mean(x)	向量 x 元素的平均值	sum(x)	向量 x 的元素总和
median(x)	向量 x 元素的中位数	prod(x)	向量 x 元素总乘积
std(x)	向量 x 元素的标准差	diff(x)	向量 x 相邻元素的差
dot(x,y)	向量 x 和 y 的内积	cross(x,y)	向量 x 和 y 的外积

B.5 其他常用的数学计算

B.5.1 逻辑运算

在 MATLAB 中任何数值都可以参与逻辑运算,并且在逻辑运算中,把所有非零值看作逻辑真,零值看作逻辑假.一般的数值可以用 `logic` 命令转换为逻辑变量.常用的逻辑运算、关系运算符列于表 B-9 中.

表 B-9 常用逻辑运算符

运 算 符	功 能	运 算 符	功 能
<code>&</code>	逻辑与运算	<code>==</code>	等于
<code> </code>	逻辑或运算	<code>~=</code>	不等于
<code>~</code>	逻辑非运算	<code><, <=</code>	小于、小于等于
<code>xor</code>	逻辑异或运算	<code>>, >=</code>	大于、大于等于

B.5.2 多项式

在 MATLAB 中,可用行向量表示多项式,其元素为按照降幂排列的多项式系数,例如:

$$p(x) = x^3 - x + 1$$

在 MATLAB 中可表示为

```
p=[1,0,-1,1]
```

MATLAB 中的 `roots` 命令用于求多项式的零点,其输入就是上面表示的多项式.

```
>>roots(p)
```

结果为

```
ans =  
    -1.3247  
    0.6624+0.5623i  
    0.6624-0.5623i
```

还有一个相关的命令是 `poly`,它的输入是多项式零点组成的向量,而输出是多项式对应的向量,如果输入是一个矩阵,则得到矩阵的特征多项式.

B.5.3 函数极值和最值

在实际应用中,经常会遇到求函数极小值的情况,常用的命令是 `fminunc`. 有两种使用方式:

```
fminunc(f,x1,x2)
```

功能是求函数 f 在 $x1$ 和 $x2$ 之间的最小值.

还有一种是:

```
fminunc(f,x0)
```

功能是求函数 f 从初始解 x_0 出发的最小值. 这个命令中的函数可以是下面要讲到的自定义的函数. MATLAB 没有求最大值的函数, 求最大值时, 转化为求这个函数的相反数的最小值.

B.6 程序设计

MATLAB 中的程序都保存为扩展名为 m 的文件, 称为 M 文件. M 文件有脚本文件和函数文件两种. 脚本文件是简单执行一系列的 MATLAB 语句, 按照文件中指定的顺序执行命令序列. 函数文件中有函数定义语句——function, 用户可以用自己编写的函数来扩充函数库. 在编程语句中, 字符 $\%$ 之后的内容是注释语句.

B.6.1 结构控制语句

1. 选择结构

选择结构常使用 if 语句, 例如:

```
if rem(a,2)==0      %判断 a 是否是偶数, 其中 rem 为取余数的函数
    a=a/2;
end
```

if 可以和 else 一起使用, 成为 if...else...end 语句, 例如:

```
if a>b
    y=a;
else
    y=b;
end
```

另一种选择结构使用 if...elseif...elseif...else...end 语句, 稍有不同. 当然, 选择语句可嵌套使用, 但应注意关键字 end 的匹配.

和 C 语言类似, MATLAB 也实现了多路分支选择的结构, 即 switch 语句, 下面是一个例子:

```
switch var          %对 var 分情况讨论
case 1
    y=a;
case 2
    y=b;
case 3
    y=c;
otherwise
    y=0;
end
```

2. 循环结构

最基础的循环是 while 循环,例如:

```
while i<n
    i=i+1;
end
```

也有 for 循环,例如:

```
for i=1:n
    sum=sum+1/i;
end
```

对于 while 循环和 for 循环,都可以使用 continue 语句,含义是结束本次循环,跳过循环体中下面未执行的语句,直接进入下一次循环。

MATLAB 中虽然有循环结构,但是实际编程中要尽可能利用 MATLAB 高效的矩阵运算,减少不必要的循环。可以说,一个会写 MATLAB 程序的人,是尽量少地用 for 循环的。

比如表达式

$$\sum_{i=1}^n [(x_i - u_i)^2 + (y_i - v_i)^2]$$

涉及向量 x, u, y, v 。用前面逐项运算符,它对应命令为:

```
sum((x-u).^2+(y-v).^2)
```

又比如 u 是列向量, $u * \text{ones}(1, n)$ 就得到一个有 n 列的矩阵,矩阵每一列都是 u 。类似地, $\text{ones}(m, 1) * v$ 把行向量 v 扩展成一个有 m 行的矩阵。

再举一个例子,说明利用 Kronecker 张量积命令 kron 简化计算。kron(A, B) 就是把 B 放到 A 的每个元素处,比如 X 是一个 2×3 的矩阵,那么 kron(X, Y) 表示:

```
[X(1,1)*Y X(1,2)*Y X(1,3)*Y
 X(2,1)*Y X(2,2)*Y X(2,3)*Y]
```

利用这个可以简化很多操作,例如(x, y 为两个列向量):

```
P=zeros(121,2);
```

```
for i=0:10
```

```
    for j=1:11
```

```
        P(11*i+j,1)=x(i+1);
```

```
        P(11*i+j,2)=y(j);
```

```
    end
```

```
end
```

这里有两重循环,如果用直积来写的话就是:

```
P(:,1)=kron(x,ones(11,1));
```

```
P(:,2)=kron(ones(11,1),y);
```

这样就省去了循环。

B.6.2 字符串

1. 字符串的基本操作

和其他高级程序设计语言一样, MATLAB 中也有字符和字符串. 字符串的命名和变量相同, 建立字符串命令的一个例子是:

```
>>a='a b c'
a=
a b c
```

注意空格是一个字符, 而且字符串中的空格是算字符串长度的, 比如对于上例:

```
>>length(a)
ans=
5
```

可以用方括号把多个字符串合并, 可以合并成单行或者多行的字符串(多行字符串为字符型矩阵). 例如:

```
>>a='a b c';
>>b='c d e';
>>c=[a b]
c=
a b c c d e
```

下面是合并成为多行字符串的例子(此时要求两个字符串的长度必须相同):

```
>>a='a b c';
>>b='d e f';
>>c=[a; b]
c=
a b c
d e f
```

2. 字符串的操作函数

常用的字符串操作函数及其功能如表 B-10 所示, 实际上和 C/C++ 语言非常类似.

表 B-10 字符串操作的函数

命 令	功 能
strcmp(a,b)	比较 a 和 b 是否相同, 若相同返回逻辑真, 否则返回逻辑假
strncmp(a,b,n)	比较 a 和 b 的前 n 个字符是否相同
strcat(a,b)	合并 a 和 b (合并成为新的字符串, 并不是像 C 语言中连接到 a 后面)

B.6.3 人机交互和输入输出

1. 命令窗口函数

常用的命令窗口输出函数是 disp 函数. disp 使用的例子如下:


```
disp(x); %在命令窗口显示 x 的值
disp('Hello World!'); %在命令窗口显示“Hello World!”
```

disp 命令也可以同时显示字符和数字,注意要把数字转化为字符显示,例如:

```
x=1.5
disp(['The value of x is ',num2str(x)]);
```

其中 num2str(x)的功能是把 x 转化为字符,输出结果是:

```
The value of x is 1.5
```

input 命令可以用来提示用户从键盘输入数据,利用 input 命令输入数据的例子如下:

```
x=input('input data x:');
```

这个代码运行的结果是命令窗口显示:

```
input data x:
```

提示用户输入数据 x .

2. 文件操作函数

除了前面讲到的 load 函数以外,MATLAB 还有类似 C 语言的文件输入函数 fscanf 和文件输出函数 fprintf. 使用的时候有配套的先用 fopen 函数打开文件,fclose 函数关闭文件. 这些函数如何使用,结合下面的例子具体说明.

```
fp1=fopen('input1.txt','r'); %指向文件 input1.txt,以'r'只读方式打开
fp2=fopen('output1.txt','w'); %指向文件 output1.txt,以'w'只写方式打开
x=fscanf(fp1,'%f'); %利用 fp1 从 input1.txt 读入数据到 x
fprintf(fp2,'%9.8f',x); %利用 fp2 输出数据到 output1.txt
fclose(fp1); %关闭文件 input1.txt
fclose(fp2); %关闭文件 output1.txt
```

输入函数 fscanf 括号中的引号中的 f 代表数据是实数.

输出函数 fprintf 括号中的引号中的符号代表了输出格式,有多种形式,下面是两个例子.

(1) %p.qf: 小数形式输出,总字长 p 位,其中小数点后 q 位.

(2) %p.qe: 指数形式输出,总字长 p 位,其中指数部分 q 位.

这里只介绍了最基本的文件操作,MATLAB 也支持 Excel 等格式文件的数据读写,详见帮助文档.

B.6.4 用户自定义函数

用户可以自定义函数写成函数式的 M 文件,第一行以 function 关键词开始,说明此文件是一个函数. 默认情况下,函数式 M 文件中的变量都是局部变量,仅在函数运行期间有效,函数运行结束,这些变量将从工作空间中清除. 下面是一个简单的自定义函数的例子:

```
function f=myfun(x)
f=2*x+512/x;
```

用户在命令窗口输入:

```
>> fminunc(@myfun,1)
```

结果为

```
ans =  
15.9991
```

除了写成 M 文件,还可以用匿名函数的方式定义函数.例如,上面的函数可以这样定义: $\text{myfun} = @(x) 2 * x + 512/x$;再使用它时不需要加@符号.例如: $\text{fminunc}(\text{myfun}, 1)$.

和 C/C++ 等高级语言一样, MATLAB 中也有结构体类型数据,可以实现更复杂的数据结构.不仅如此, MATLAB 中还有类似 C/C++ 中指针的机制用以传递数据,有类和对象,可以进行面向对象的程序设计(OOP),可以进行 MATLAB 和 C++ 的混合编程,还可以开发图形界面的程序(GUI),感兴趣的读者可以参考进一步的资料.

B.7 符号计算

MATLAB 也具有相当不错的符号计算能力,这里做简要介绍.符号计算最常用的两个命令是 `sym` 和 `syms`.

`sym` 常用的是两种功能.

(1) $S = \text{sym}(A)$,把表达式 A 转化为符号对象 S .

(2) $x = \text{sym}('x')$,以 'x' 为名创建符号变量,并将结果存储到 x .

下面是一个用 `sym` 函数定义符号表达式的例子:

```
>> a = sym('a');  
>> b = sym('b');  
>> c = sym('c');  
>> x = sym('x');  
>> f = a * x^2 + b * x + c  
f =  
a * x^2 + b * x + c  
>> h = sym('a * x^2 + b * x + c')  
h =  
a * x^2 + b * x + c
```

MATLAB 还可以用 `syms` 函数创建符号对象,调用语法如下:

```
syms arg1 arg2; % 等同于 arg1=sym('arg1'); arg2=sym('arg2');
```

符号多项式运算的常用命令如表 B-11 所示.

表 B-11 符号多项式运算的常用命令

命 令	功 能	命 令	功 能
<code>collect(表达式, v)</code>	指定 v 为独立变量,合并同类项	<code>factor(表达式)</code>	将表达式进行因式分解
<code>expand(表达式)</code>	将表达式展开为多项式	<code>simply(表达式)</code>	将表达式简化

用符号运算求解方程和方程组的函数是 solve,使用格式如下:

```
solve('f') %解方程 f
solve('f1',..., 'fn', 'x1',..., 'xn') %对指定的自变量 x1,...,xn 解方程组 f1,...,fn
```

下面通过两个例子简单说明. 输入:

```
>>syms a b c x
>>S=a*x^2+b*x+c;
>>solve(S,'x')
```

结果是:

```
ans=
-1/2*(b-(b^2-4*a*c)^(1/2))/a
-1/2*(b+(b^2-4*a*c)^(1/2))/a
```

正好是求根公式.

输入:

```
>>syms u v a
>>S=solve('u^2-v^2=a^2','u+v=1','u','v')
```

结果是:

```
S=
u: [1x1 sym]
v: [1x1 sym]
```

需要具体查看的话,输入:

```
>>S.u
```

读者从这里可以看出,S 其实是一个“结构体”,结果是:

```
ans=
1/2+1/2*a^2
```

B.8 绘图功能

MATLAB 具有强大的图形编辑功能,拥有大量灵活、易用的二维和三维图形命令,这里介绍最基本的应用.

B.8.1 二维图形

对于离散实函数 $y_i = f(x_i) (i=1, \dots, n)$, 对于自变量组成的向量, 求出对应函数值的向量, 将对应的点列在直角坐标系中画出, 就实现了离散函数的可视化. 对于连续的函数, 也是先计算出在一组离散自变量上的函数值, 并把这些点列在图中表示, 为表现出函数的连续性, 可对区间做出很细的分割.

plot 是最基本的绘图命令, 例如输入:


```
>>t=-1:0.05:1;
>>y=2*t.^2-1;
>>plot(t,y,'b')
```

则显示出图形窗口,如图 B-4 所示。

这里使用了“.”操作符,表示对向量的各个元素逐个进行计算,这是绘图中常用到的操作符之一;单引号中的 b 代表 blue,表示用蓝色绘制曲线。plot 命令还有多种使用格式,详见表 B-12。

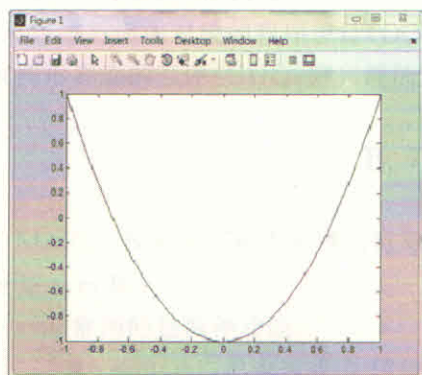


图 B-4 抛物线 $y=2t^2-1$ 在 $[-1,1]$ 上的图像

表 B-12 plot 命令的使用格式

命 令	功 能
plot(x,y)	以 x 为横坐标, y 为纵坐标,按照坐标 (x_i, y_i) 的有序排列绘制曲线
plot(y)	以 i 为横坐标, y_i 为纵坐标,按照坐标 (i, y_i) 的有序排列绘制曲线
plot(z)	以横轴为实轴,纵轴为虚轴,在复平面上绘制复数序列 z_i
plot(x,y,'s')	功能基本同 plot(x,y),但按 s 指定的样式绘制曲线
plot(x,y1,'s1',x,y2,'s2',...)	将多条曲线画在同一个坐标系中,分别绘成 $s1, s2, \dots$ 的颜色

表 B-12 中的 $s, s1, s2$ 等样式字符串可以是颜色和线型标识符的组合,具体的这两类标识符列于表 B-13 中。

表 B-13 plot 命令中的颜色、线型设置参数

线型	功 能	线型	功 能	颜色	功 能	颜色	功 能
—	实线	o	小圆圈	r	红色	k	黑色
--	虚线	*	星号	b	蓝色	w	白色
:	点线	×	叉号	g	绿色	m	品红色
-·	点划线	+	加号	y	黄色	c	青色

MATLAB 中可以对图形和坐标进行操作和修改,常用的函数如表 B-14 所示。

表 B-14 对绘图区的一些属性修改的命令

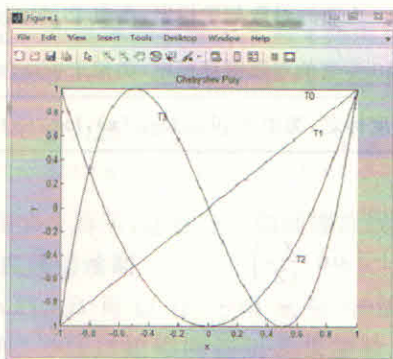
命 令	功 能
subplot(m,n,k)	在一个窗口中绘制 $m \times n$ 个图形,其后 plot 的图位于位置 k
title('s')	在图形上方显示标题 s
xlabel('s')	用 s 标记 X 轴
ylabel('s')	用 s 标记 Y 轴
text(x,y,'s')	将 s 显示在 (x,y) 决定的位置上

命 令	功 能
gtext('s')	将 s 显示在鼠标确定的位置上
hold on	保留现有图形,随后的图形叠在此图上
hold off	解除 hold on,随后的图形覆盖现有图形
grid on	显示坐标网格
axis auto	坐标使用默认设置
axis(xmin,xmax,ymin,ymax)	设定 x 和 y 坐标的刻度显示范围
axis equal	横纵坐标采用等长刻度

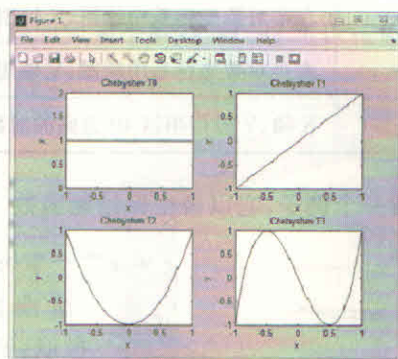
下面看一个综合性的例子,由于命令较多,这里写成脚本式的 m 文件,写好后,在命令窗口运行. 设文件名为 Che.m,内容如下:

```
x=-1:0.05:1;
t0=1.0+0*x;
t1=x;
t2=2*x.*t1-t0;
t3=2*x.*t2-t1;
title('Chebyshev Poly');
xlabel('x');ylabel('y');
hold on;
plot(x,t0);gtext('T0');
plot(x,t1);gtext('T1');
plot(x,t2);gtext('T2');
plot(x,t3);gtext('T3');
```

在命令窗口输入 Che,输出图形如图 B-5(a)所示.



(a) 在同一坐标系



(b) 在不同坐标系

图 B-5 0 次至 3 次第一类切比雪夫多项式在区间 $[-1, 1]$ 上的曲线

如果程序改写成 subplot 的形式,即:

```
x=-1:0.05:1;
t0=1.0+0*x;
```

```

t1=x;
t2=2*x.*t1-t0;
t3=2*x.*t2-t1;
subplot(2,2,1);plot(x,t0);title('Chebyshev T0');xlabel('x');ylabel('y');
subplot(2,2,2);plot(x,t1);title('Chebyshev T1');xlabel('x');ylabel('y');
subplot(2,2,3);plot(x,t2);title('Chebyshev T2');xlabel('x');ylabel('y');
subplot(2,2,4);plot(x,t3);title('Chebyshev T3');xlabel('x');ylabel('y');

```

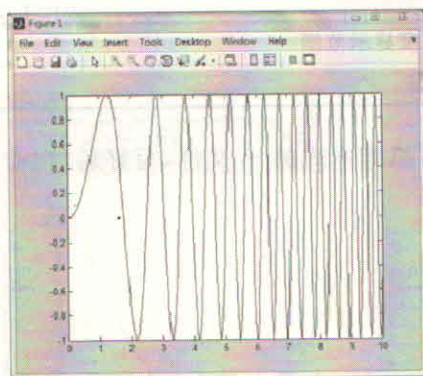


图 B-6 函数 $f(x) = \sin(x^2)$ 在区间 $[0, 10]$ 上的图像

输出的图形如图 B-5(b)所示。

fplot 命令可以方便绘出自定义函数的图形, 看一个简单的例子, 输入:

```
>> fplot('sin(x.^2)', [0, 10])
```

得到函数 $f(x) = \sin(x^2)$ 在区间 $[0, 10]$ 上的图像(如图 B-6 所示)。

和 plot 命令的区别是 fplot 并不需要给出取值点的选取, 实际上, 它是根据函数值自适应地选取取值点的。另一个常用绘制函数曲线的命令是 ezplot, 读者可以自己试试它。

plot 命令使用的是笛卡儿坐标系, MATLAB 中也可以绘制其他坐标系的图形, 比如极坐标、半对数坐标、双对数坐标。命令及对应功能如表 B-15 所示。

表 B-15 针对不同坐标系的绘图命令

命 令	功 能
polar(x,r)	在极坐标中绘图, x 代表极角, 单位弧度, r 代表极径
semilogx(x,y)	在半对数坐标系中绘图, X 轴用以 10 为底的对数刻度标定, 类似于 $\text{plot}(\log_{10}(x), y)$
semilogy(x,y)	在半对数坐标系中绘图, Y 轴用以 10 为底的对数刻度标定, 类似于 $\text{plot}(x, \log_{10}(y))$
loglog(x,y)	X 轴、Y 轴均用以 10 为底的对数刻度标定, 类似于 $\text{plot}(\log_{10}(x), \log_{10}(y))$

下面看一个例子, 它以极坐标绘制函数

$$r = e^{\cos t} - 2\cos 4t + \left(\sin \frac{t}{12}\right)^5.$$

输入:

```

>> t=linspace(0, 22*pi, 1100);
>> r=exp(cos(t))-2*cos(4*t)+(sin(t./12)).^5;
>> p=polar(t,r)

```

图形如图 B-7(a)所示。其中 linspace(x1,x2,N)的功能是在 x_1 和 x_2 之间均匀地产生 N 个取值点。在上述命令之后输入:

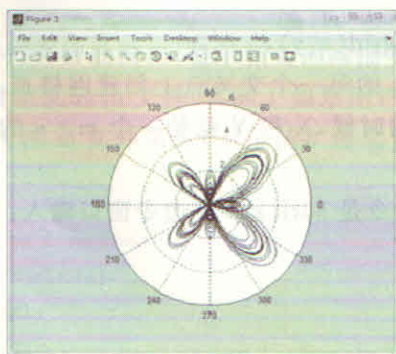
```
>> [x,y]=pol2cart(t,r);
```

%算出对应的笛卡儿坐标

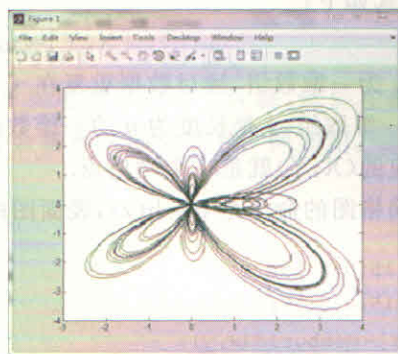
>>plot(x,y)

%在笛卡儿坐标系中绘图

得到笛卡儿坐标系中的图形如图 B-7(b)所示。



(a)



(b)

图 B-7 极坐标绘制函数图像示例

B.8.2 三维图形

1. 曲线的绘制

三维空间里的曲线绘制通过 plot3 命令实现,命令格式和对应功能如表 B-16 所示。

表 B-16 三维空间里的曲线绘制

命 令	功 能
plot3(x,y,z)	按照坐标 (x_i, y_i, z_i) 的有序排列绘制曲线
plot3(x,y,z,'s')	功能基本同上,把曲线绘制成 s 指定的样式
plot3(x1,y1,z1,'s1',x2,y2,z2,'s2'...)	在同一坐标系内,用 s1 的样式绘制曲线 1,用 s2 的样式绘制曲线 2

这里三个坐标的数组 x、y 和 z 必须有相同的元素个数,s、s1、s2 可以按照表 B-13 进行选择。下面看一个三维空间的曲线的例子,输入:

```
>>t=0:pi/100:5*pi;
```

```
>>plot3(3*sin(t),2*cos(t),t)
```

图形如图 B-8 所示,这是一个椭圆螺旋线。

2. 曲面的绘制

MATLAB 可以在三维坐标中绘制曲面。MATLAB 中通过矩形网格组合来描绘曲面,即将 (x,y) 定义的区域划分为一个个矩形区域,然后计算在这些矩形区域的顶点处的 z 值,在显示的时候,把邻接的顶点互相连接起来,组合出曲面。组合这些网格显示整个曲面的时候,MATLAB 可以采用

两种方式:一种是只用线条将各个临近顶点连接,网格区域内部显示为空白,通过网格边框来显示整个曲面,这种曲面图称为网线图;另一种不但显示网格线边框,而且将其内部填充

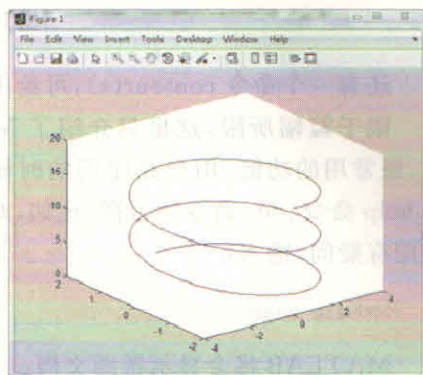


图 B-8 椭圆螺旋线

着色,从而通过一个个矩形平面来组合显示整个曲面,这种曲面图称为表面图。

在 MATLAB 中绘制三维曲面图,先使用 meshgrid 函数在 (x,y) 的矩形区域上创建网格,调用格式如下:

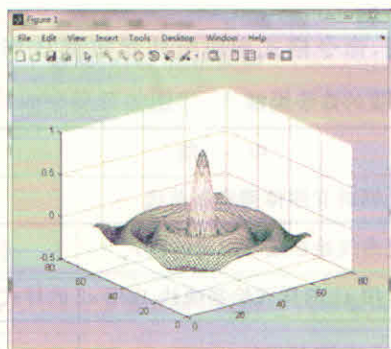
$$[X,Y]=\text{meshgrid}(x,y),$$

其中 x 和 y 为一维数组,通过数据重复在 x 和 y 的每一个交叉点上创建网格点。当 x 是长度为 m 的一维数组, y 是长度为 n 的一维数组的时候, X 和 Y 就是一个 $m \times n$ 的二维数组,每一个对应的 (X,Y) 就是一个网格点。

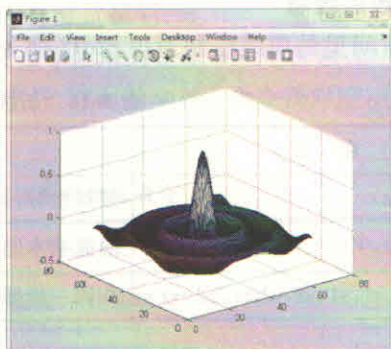
绘制网格图的命令是 mesh(z),表面图的命令是 surf(z)。在命令窗口输入:

```
>>x=-15:0.5:15;  
>>y=-15:0.5:15;  
>>[X,Y]=meshgrid(x,y);  
>>r=sqrt(X.^2+Y.^2)+eps; % 为什么要加上 eps,原因请读者思考  
>>z=sin(r)./r;  
>>mesh(z)
```

可以看出这里生成的 z 其实是一个和网格结点矩阵维数相同的矩阵,得到的图形如图 B-9(a)所示,如果把 mesh(z)换成 surf(z),得到图 B-9(b)。



(a)



(b)

图 B-9 三维曲面绘图示例

还有一个命令 contour(z),可绘出矩阵 z 的等高线,如图 B-10 所示。

限于篇幅所限,这里只介绍了 MATLAB 最基本、最常用的功能,用户在使用的时候,最好养成使用 help 命令、doc 命令的习惯,比如,对 mesh 函数的使用有疑问,输入:

```
>>help mesh
```

MATLAB 将会显示帮助文档:

MESH 3-D mesh surface.

MESH(X,Y,Z,C) plots the colored parametric mesh defined by four matrix arguments.

The view point is specified by VIEW. The

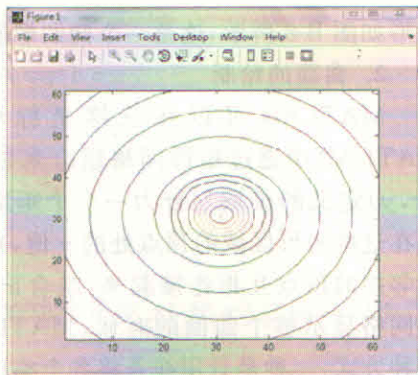


图 B-10 等高线图

axis labels are determined by the range of X,Y and Z,
or by the current setting of AXIS.

...

doc 命令将打开帮助窗口,显示相关的帮助文档,内容更全面、丰富。

对于自定义的 m 文件,实际上如果用户在开头用 % 标记写上注释,然后在命令窗口输入 help 对应的 m 文件名称(不包括扩展名),将会显示这部分注释。例如用户自定义函数如下:

```
function f=myfun(x)
%return sin(x)
f=sin(x);
```

在命令窗口输入:

```
>>help myfun
```

结果是:

```
return sin(x)
```

除了上面介绍的,读者应多使用 doc、help 阅读帮助文档,并通过实际应用熟练掌握 MATLAB 的其他高级功能。