

# MATLAB 快速入门电子版

<b>第一章 MATLAB 简介及操作指南</b>	<b>1</b>
1.1 MATLAB 语言的特点	1
1.2 MATLAB 的基本操作入门	2
1.3 MATLAB 的命令和窗口环境	4
1.4 MATLAB 的指令行的操作	12
1.5 MATLAB 的在线帮助系统	14
1.6 演示程序	15
<b>第二章 MATLAB 的基础语法</b>	<b>16</b>
2.1 矩阵的输入与运算	16
2.2 数组及其运算	19
2.3 语句、变量和表达式	20
2.4 函数	21
2.5 程序设计	22
<b>第三章 MATLAB 图形制作</b>	<b>26</b>
3.1 函数 plot	26
3.2 函数 ezplot	27
3.3 函数 ezpolar	29
3.4 函数 polar	30
3.5 函数 ezcontour	31
3.6 函数 fplot 和其它画图函数	32
3.7 对曲线的进一步处理	33
3.8 画多重线的方法	35
3.9 图形窗口的分割	39
3.10 空间图形制作	40
3.11 综合作图	43
<b>第四章 MATLAB 的符号运算</b>	<b>46</b>
4.1 符号变量与符号表达式	46
4.2 微积分运算	49
4.3 解方程	55
4.4 线性代数	61
4.5 化简与代换	64
4.6 其它	69

# 第一章 MATLAB 简介及操作指南

MATLAB 是 MATrix LABoratory 的缩写，它是一种数值计算和图形图像处理工具软件，它的特点是语法结构简明、数值计算高效、图形功能完备、易学易用。它在矩阵代数、数值计算、数字信号处理、振动理论、神经网络控制、动态仿真等领域都有广泛的应用。

最初的 MATLAB 软件包是 1967 年由 Clere Maler 用 FORTRAN 语言编写的，新版的 MATLAB 是由 Mathorks 用 C 语言完成的，它自 1984 年推向市场以来，历经十几年的发展和竞争，现已成为国际认可(IEEE)的最优化的科技应用软件。在它的发展过程中，许多优秀的编者为了它的完善作出了卓越的贡献，使它从一个简单的矩阵分析软件逐渐发展成为一个具有极高通用性的、带有众多实用工具的运算操作平台。

正是由于 MATLAB 与其他高级语言相比，MATLAB 提供了一个人机交互的数学系统环境，并以矩阵作为基本的数据结构，可以大大节省编程时间。在美国大学中，MATLAB 语言受到了教授与学生的欢迎和重视。由于它将使用者从繁重重复的计算中解放出来，把更多的精力投入到对数学的基本含义的理解上，因此，它已逐步成为许多大学生和研究生课程中的标准和重要的工具。不论在教学还是学生解题时，它都表现出高效、简单和直观的性能，是计算机辅助设计强有力的工具。因此，在国外的高等院校里，熟练运用 MATLAB 已成为大学生、硕士生、博士生必须掌握的基本技能；在设计研究单位和工业部门，MATLAB 已成为研究必备软件 and 标准软件。

## 1.1 MATLAB 语言的特点

**MATLAB 语言的特点如下：**

### 特点 1：起点高

- (1) 每个变量代表一个矩阵，它可以有  $n \times m$  个元素。
- (2) 每个元素都看作复数，这个特点在其它语言中也是不多见的。
- (3) 所有的运算，包括加、减、乘、除、函数运算都对矩阵和复数有效。

### 特点 2：人机界面适合科技人员

- (1) 语言规则与笔算式相似：MATLAB 的程序与科技人员的书写习惯相近，因此易写易读，易于在科技人员之间交流。
- (2) 矩阵行数、列数无需定义：若要输入一个矩阵，在用其它语言编程时必须先定义矩阵的阶数，而用 MATLAB 语言则不必有阶数定义语句，输入数据的行列数就决定了它的阶数。
- (3) 键入算式立即得结果，无需编译：MATLAB 是以解释方式工作的，即它对每条语句解释后立即执行。若有错误也立即作出反应，便于编程者马上改正。这些都大大减轻了编程和调试的工作量。

### 特点 3：强大而简易的作图功能

- (1) 能根据输入数据自动确定坐标绘图。
- (2) 能以多种坐标系（极坐标、对数坐标等）绘图。
- (3) 能绘制三维坐标中的曲线和曲面。
- (4) 可设置不同颜色、线型、视角等。

如果数据齐全，通常只需一条命令即可出图。

### 特点 4：智能化程度高

- (1) 绘图时自动选择最佳坐标以及按输入或输出变元数自动选择算法等。
- (2) 做数值积分时自动按精度选择步长。
- (3) 自动检测和显示程序错误的能力强，易于调试。

### 特点 5：功能丰富，可扩展性强

MATLAB 软件包括基本部分和专业扩展部分。基本部分包括：矩阵的运算和各种变换，代数和超越方程的求解，数据处理和傅里叶变换，数值积分等等，可以满足大学理工科本科的计算需要。本书将介绍这部分的矩阵的运算和计算方法等方面的主要内容。

扩展部分称为工具箱。它实际上是用 MATLAB 的基本语句编成的各种子程序集，用于解决某一方面的专门问题，或实现某一类的新算法。现在已经有控制系统、信号处理、图像处理、系统辨识、模糊集合、神经网络、小波分析等 20 余个工具箱，并且还在继续发展中。

MATLAB 的核心内容在它的基本部分，所有的工具箱子程序都是用它的基本语句编写的。在大学本科学习阶段，不仅要会算，更重要的是知其所以然。因此我们原则上不提倡学生在低年级使用工具箱。为了避免学生在 4 年学习过程中重复编程，学生可以用基本语句来编自己的子程序，保存起来，以巩固已掌握的知识，并用于后续课程。

## 1.2 MATLAB 的基本操作入门

本节介绍如何以不同方式进入和退出 MATLAB、MATLAB 的命令和窗口环境、MATLAB 的指令行的操作和演示程序等。为了能够更快的理解和掌握 MATLAB 执行命令的方式，还将介绍一些简单的例子，通过这些例子可以很快体会到 MATLAB 对计算和图形的操作确实方便快捷。

### 1.2.1 MATLAB 的几种启动方法

在 Win95/98 环境下，我们常用如下两种方法启动 MATLAB：

方法一：以快捷方式启动

(1) 启动 Windows；

(2) 双击 MATLAB 图标。见图 1-1。进入 MATLAB 的命令窗口，也称作工作窗口（见图 1-2 MATLAB 的命令窗口）。



图 1-1 Windows 窗口

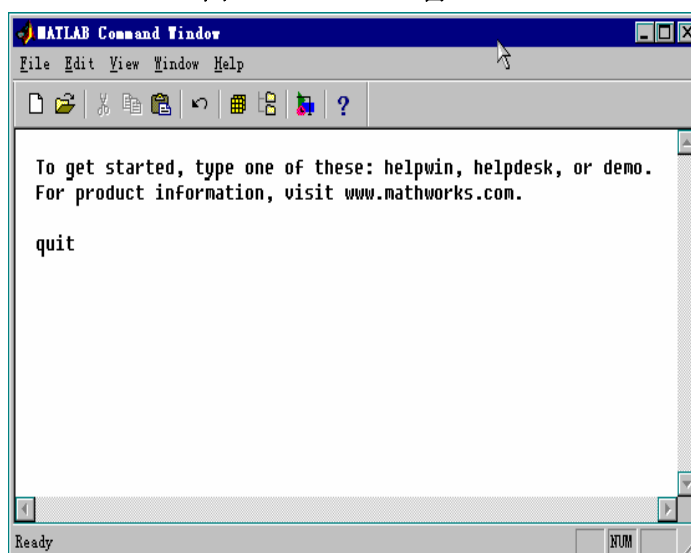


图 1-2 MATLAB 的命令窗口

方法二：以菜单方式启动 MATLAB：

(1) 启动 Windows；

(2) 单击[开始]；

(3) 依次选择[程序].Matlab 和 Matlab 命令，见图 1-3，进入 MATLAB 的命令窗口。

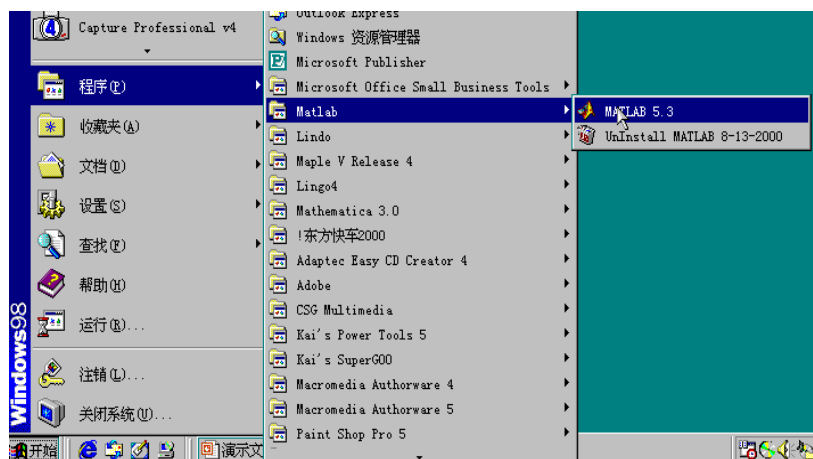


图 1-3 从开始菜单进入 MATLAB 的命令窗口

说明：在 MATLAB 命令窗口的上方两行文字是初始提示信息。您可以在第三行键入命令。

## 1.2.2 MATLAB 的几种退出方法

有多种退出 MATLAB 的方式，我们常用如下五种方法退出 MATLAB：

方法一：在 MATLAB 命令窗口的“File”菜单下选择“Exit MATLAB”；

方法二：快捷键“Ctrl+q”；

方法三：在 MATLAB 的命令窗口输入“Quit”或“exit”命令；

方法四：用鼠标单击 MATLAB 命令窗口角的 ；

方法五：用鼠标单击 MATLAB 命令窗口左上角的 。

## 1.2.3 MATLAB 的几个应用实例

为了能够更快的理解和掌握 MATLAB 执行命令的方式，我们下面介绍一些简单的例子，通过这些例子可以很快体会到 MATLAB 对进行计算和图形的操作确实方便快捷。

我们可以试着在 MATLAB 的命令窗口中分别输入下面几个例子的程序，在每个程序输入后，敲回车键，就会给您们一种惊喜。试试看吧！！

**[例 1]** 输入命令：

```
>> v=eye(3,4) %3×4 对角线为 1 的矩阵。
```

运行后输出结果如下：

```
v =  
    1    0    0    0  
    0    1    0    0  
    0    0    1    0
```

**[例 2]** 输入命令：

```
>> s1='Hello';s2='every';s3='student';  
s=[s1','s2','s3],ss=s(1:5)
```

运行后输出结果如下：

```
s =  
Hello,every student
```

```
ss =
```

```
Hello
```

```
>> subplot(2,1,1),  
Y=peaks; plot(Y),title(' peaks '), gtext(' peaks ')  
subplot(2,1,2),  
fplot('tan(x),sin(x)/x,cos(x)',2*pi*[-1 1 -1 1])  
title(' tan(x),sin(x)/x,cos(x)')
```

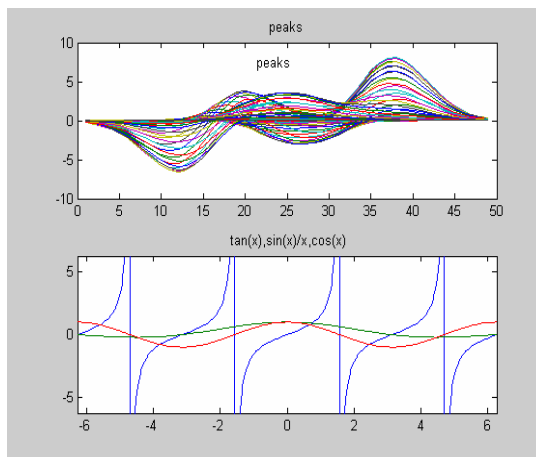


图 1-4 例 3 的图形

运行后输出图 1-4。

**[例 4]** 输入命令:

```
>> x=linspace(0,16,100);y=sin(x);z=cos(x);  
plot(x,y,x,z)  
grid  
xlabel('Independent Variable X')  
ylabel('Dependent Variables Y and Z')  
title('sin(x)和 cos(x)的曲线')
```

运行后输出图 1-5。

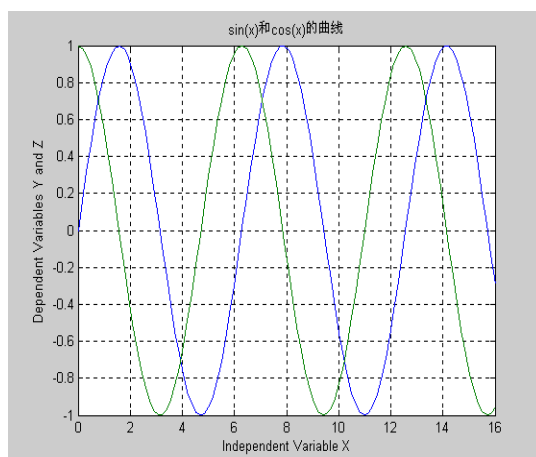


图 1-5 例 4 的图

**[例 5]** 作曲面  $z = f(x, y)$  的图形。

$$z = \frac{\sin \sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}}, \quad -7.5 \leq x \leq 7.5, \quad -7.5 \leq y \leq 7.5$$

用以下程序实现:

```
>> x=-7.5:0.5:7.5; y=x;  
[X,Y]=meshgrid(x,y);  
R=sqrt(X.^2+Y.^2)+eps; Z=sin(R)./R;  
mesh(X,Y,Z)
```

运行后输出图 1-6

**[例 6]** 作螺旋线  $x = \sin t, \quad y = \cos t, \quad z = t$

**解** 用以下程序实现:

```
>> t=0:pi/50:10*pi; plot3(sin(t),cos(t),t)
```

运行后输出图 1-7。

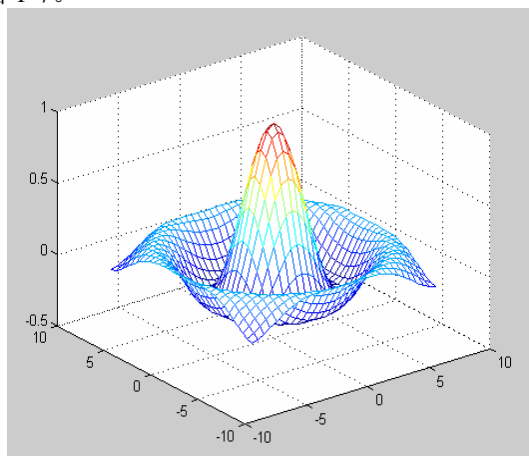


图 1-6 例 5 的图

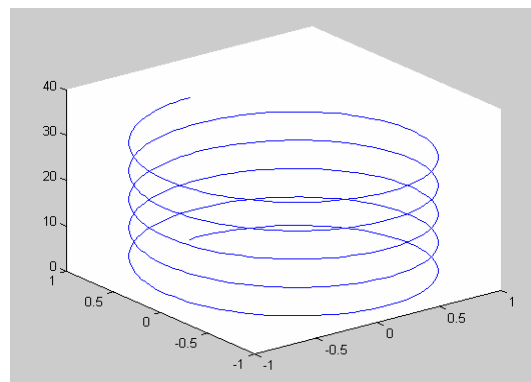


图 1-7 例 6 的图

## 1.3 MATLAB 的命令和窗口环境

MATLAB 是一个标准的 Windows 界面, 可以利用菜单中的命令完成对命令窗口的操作。它的使用方法与 Windows 的一般应用程序相同, 参看图 1-8。下面将对菜单进行介绍。

### 1.3.1 M-文件

**M-文件**有两种类型: 文本 M-文件和函数 M-文件。

#### (一)、文本 M-文件

一个比较复杂的程序常常要作反复的调试, 这时你不妨建立一个文本文件并把它储存起来, 可以随时调用进行计算。建立文本文件可以在 File 菜单中选择 New, 再选择 M-file, 这时 MATLAB 将打开一个文本编辑窗口, 在这里输入命令和数据。储存时文件名遵循 MATLAB 变量命名的原则, 但必须以 **m** 为扩展名, 其一般形式为

<M-文件名>.m

如 juzhen.m, pp.m 等。

值得注意的是，文本 M-文件中的变量都是全局变量，在执行过程中，文本 M-文件中的命令可以使用目前工作区中的变量，它所产生的变量也将成为工作区的一部分。比如我们把下面的程序保存为名为 budijifen1.m 的文本 M-文件：

```
function budijifen1
x=-4*pi:0.014*pi;a=1;
for C=-3:3
y=1/2*(-sin(a*x)+a*x)/a+C
plot(x,y)
hold on
end
grid
hold off
xlabel('自变量 X')
ylabel('因变量 Y')
title('不定积分 的积分曲线族')
legend('不定积分 的积分曲线族')
```

然后在 MATLAB 工作窗口中输入文件名：

```
>> budijifen1
```

运行后屏幕显示计算结果(略)和图形(图 1-9)。

再如，将下面的一个生成矩阵的程序保存为名为 juzhen.m 的文本 M-文件：

```
for i=1:k
for j=1:n
b(i,j)=7/(2*i+3*j-6);
end
end
a=rats(b)
```

然后在 MATLAB 工作窗口中输入：

```
>> k=3;n=4; juzhen,b
```

运行后屏幕显示运行结果如下：

```
for i=1:k
for j=1:n
b(i,j)=7/(2*i+3*j-6);
b(i,j)=7/(2*i+3*j-6);
b(i,j)=7/(2*i+3*j-6);
b(i,j)=7/(2*i+3*j-6);
for j=1:n
b(i,j)=7/(2*i+3*j-6);
b(i,j)=7/(2*i+3*j-6);
b(i,j)=7/(2*i+3*j-6);
b(i,j)=7/(2*i+3*j-6);
for j=1:n
b(i,j)=7/(2*i+3*j-6);
b(i,j)=7/(2*i+3*j-6);
b(i,j)=7/(2*i+3*j-6);
b(i,j)=7/(2*i+3*j-6);
end
end
a=rats(b)
a =
    -7          7/2          7/5          7/8
     7          7/4          1          7/10
    7/3          7/6          7/9          7/12

b =
-7.000000000000000 3.500000000000000 1.400000000000000 0.875000000000000
 7.000000000000000 1.750000000000000 1.000000000000000 0.700000000000000
 2.333333333333333 1.166666666666667 0.777777777777778 0.583333333333333
```

## (二)、函数 M -文件

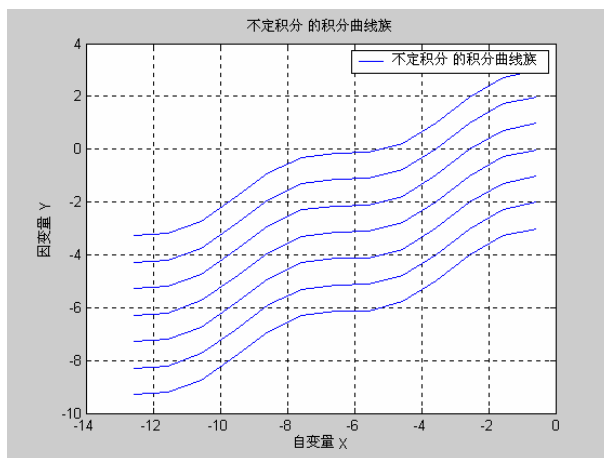


图 1-9 名为 budijifen1.m 的文本 M-文件



函数 M-文件是另一类 M-文件，我们可以根据自己需要建立自己的函数文件，它们能够像库函数一样方便地调用，从而极大地扩展 MATLAB 的能力。如果对于一类特殊的问题，建立起许多函数 M-文件，就能最终形成独立的工具箱。

函数 M-文件的第一行有特殊的要求，其形式必须为

**function** <因变量>= <函数名> (<自变量>)

其他各行均为从自变量计算因变量的语句，并最终将结果赋予因变量。而这个 M-文件的文件名必须是<函数名>.m。下面给出函数文件的一个简单例子。

如果要在某个程序中要调用函数  $y = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ ，就必须建立并保存下面的名为 fun.m 的函数 M-文件。

```
function y = fun(x)
y = exp((-x.^2)/2)./(sqrt(2*pi));
```

然后在 MATLAB 工作窗口中输入程序：

```
>> [Q1,FCNT14] = quad(@fun,0,pi/2,1.e-14,3) % quad 是一个计算数值积分的程序名
```

运行后屏幕显示计算的部分结果如下：

```
.....
551      1.5441340226      1.33311521e-002      0.0015978928
553      1.5574651747      1.33311521e-002      0.0015650585
Q1 =
    0.44188501721659
FCNT14 =
    553
```

符号 % 后面是注释部分，给出了这个程序的说明，MATLAB 执行时忽略这些内容。实际上 MATLAB 的所有函数都有这样一段注释。

函数 M-文件有多个因变量时，要用[]将它们括起来，请看下面数值积分程序 quad.m 的部分内容：

```
function [Q,fcnt] = quad(funfcn,a,b,tol,trace,varargin)
%QUAD    Numerically evaluate integral, adaptive Simpson quadrature.
% .....
%    [Q,FCNT] = QUAD(...) returns the number of function evaluations.
%
%    QUAD(FUN,A,B,TOL,TRACE) with non-zero TRACE shows the values
%    of [fcnt a b-a Q] during the recursion.
%
%    QUAD(FUN,A,B,TOL,TRACE,P1,P2,...) provides for additional
%    arguments P1, P2, ... to be passed directly to function FUN,
%    FUN(X,P1,P2,...).  Pass empty matrices for TOL or TRACE to
%    use the default values.
%.....
f = fcnchk(funfcn);
if nargin < 4 | isempty(tol), tol = 1.e-6; end;
if nargin < 5 | isempty(trace), trace = 0; end;
% Initialize with three unequal subintervals.
h = 0.13579*(b-a);
x = [a a+h a+2*h (a+b)/2 b-2*h b-h b];
y = feval(f, x, varargin{:});
fcnt = 7;
% Fudge endpoints to avoid infinities.
if ~isfinite(y(1))
    y(1) = feval(f,a+eps*(b-a),varargin{:});
    fcnt = fcnt+1;
end
if ~isfinite(y(7))
    y(7) = feval(f,b-eps*(b-a),varargin{:});
    fcnt = fcnt+1;
end
% Call the recursive core integrator.
.....
```

其用途不难从注释行知道。

函数 M-文件中的变量一般是局部变量，它们的变量名独立于目前的工作区和其它的函数。对于 5.0 以上的版本，在工作区和函数的定义中可以用 `global` 命令把某些变量说明为全局变量。

当 MATLAB 执行到 M-文件名的语句时，它首先搜索当前工作区中的变量和内建的命令，然后搜索有无内部函数以此命名，最后在搜索路径的目录中寻找以此命名的 M-文件。

一般情况下 MATLAB 不显示 M-文件中的内容，不过命令 `echo on` 可以让 MATLAB 显示 M-文件中的命令，并且用命令 `echo off` 关闭显示。在 M-文件中还可以引用其它 M-文件，包括递归地引用自己。

### 1.3.2 File （文件）菜单

File （文件）菜单的内容如下表 1-2 所示。

表 1-2 File （文件）菜单的使用

菜单命令	功 能
New	建立 M 文件、建立图形 、建立 Simulink 模块
Open	打开已知文件
Open Selection	打开指定文件
Run Script	运行已有的 M 文件
Load Workspace	将文件中的内容放入 MATLAB 的工作区中
Show Workspace As	将 MATLAB 工作区中的内容放入文件
Show Workspace	显示 MATLAB 工作区
Show Graphics Property Editor	显示图形属性编辑器
Show GUI Layout Tool	显示 GUI 界面布局管理器
Set Path	设置工作路径
Preferences	定义工作环境
Print Setup	打印设置
Print	打印
Print Selection	打印 指定的文件
Exit MATLAB	退 出 MATLAB

#### （一）、New 选项

File 菜单下的子菜单 New 有三个选项，我们分别介绍他们的功能。

**1、“M-File” 选项：**该选项是子菜单 New 的三个选项之一。我们用“M-File”新建一个 M 文件，该命令将打开 MATLAB 的 M 文件编辑/ 调试器，参看图 1-10。通过它们,我们可以创建和编辑 M--文件,调试 MATLAB 程序。所谓的 M--文件就是用 MATLAB 语言编写的程序，保存为一个以.m 为后缀的文件名的文件中，可以在 MATLAB 工作窗口运行其文件名调用此程序。

在 File 菜单 New 选项中，除了 M-file 选项外，还有如下两个选项：**2、 New ---“Figure” 选项：**在 MATLAB 命令窗口执行 New ---“Figure”命令可以产生一个图形窗口，参看图 1-11。执行一次 `close` 命令，关闭一个当前的图形窗口 ；要同时关闭所有的窗口，使用 `close all`。

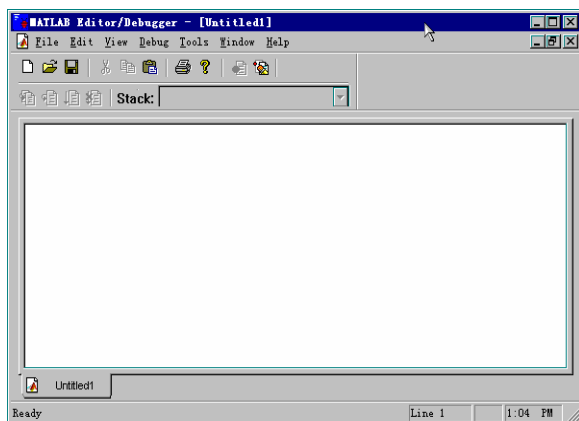


图 1-10 M 文件编辑/ 调试












图 1-11 图形窗口



图形窗口中每个工具条的功能如下表 1-3 所示：

表 1-3 图形窗口中每个工具条的功能

图 标	功 能
	允许对图形进行编辑
	在图形窗口中添加
	在图形窗口中添加
	在图形窗口中添加
	允许对图形进行放缩操作，按下该按钮后，在图形中单击鼠标左（右）键，放大（缩小）图形
	 与  的作用和操作完全相反
	允许把图形旋转为三维图形

3、New(续)---“Model” 选项：该选项新建一个 Simulink 模型窗口（参看图 1-12），并且显示 Simulink 模块库浏览器（参看图 1-13）。Simulink 是对动态系统进行建模、仿真和分析的一个软件包。

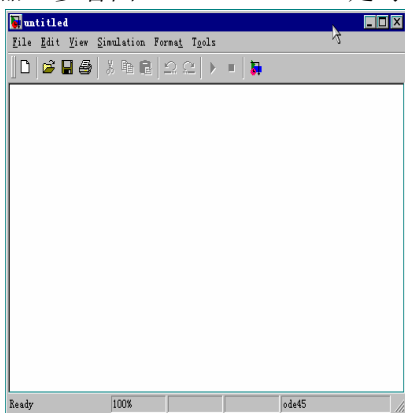


图 1-12 新建 Simulink 模型窗口

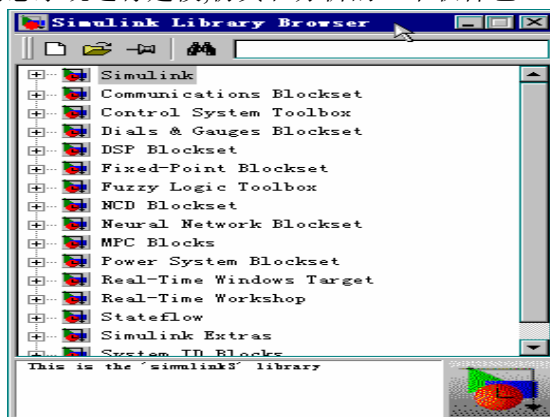


图 1-13 Simulink 模块库浏览器

## （二）、 Open 选项

单击 Open 菜单弹出打开文件对话框（参看图 1-14）。我们可以搜寻并打开 MATLAB 的 M 文件所在的目录，选中该文件，再单击“打开”按钮，将打开 MATLAB 文件（参看图 1-15）。



图 1-14 单击 Open 菜单



图 1-15 打开 MATLAB 文件

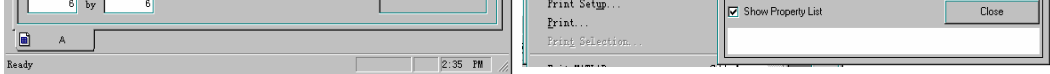
## （三）、 Open Selection 选项

我们可以在 MATLAB 命令窗口编辑区用鼠标选中 M 文件的文件名，再单击“Open Selection”选项（参看图 1-16），将用编辑器打开该文件。

## （四）、 Run Script 选项

Run Script 选项是用来运行脚本文件的。我们可以在弹出的“Run Script”对话框中（参看图 1-17）直接输入脚本文件名，再单击“OK”运行脚本文件（参看图 1-18）。

## （五）、 Load Workspace 选项



Load Workspace 选项是用来载入 MATLAB（数据）文件中的内容到工作空间。单击该选项打开一个 Load.mat file 对话框（参看图 1-19），在目录框中列出所选目录的后缀为 Mat 的文件。选中某个文件后，将把该文件中保存的变量载入到当前工作空间。

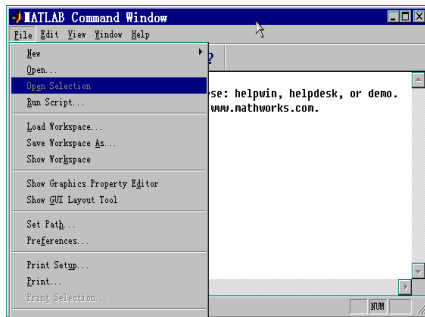


图 1-16 单击“Open Selection”选项

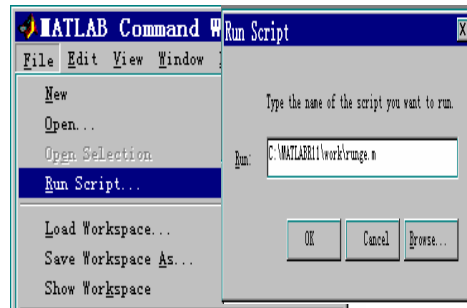


图 1-17 Run Script 对话框

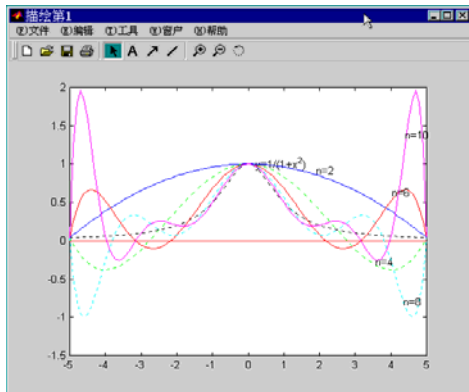


图 1-18 脚本文件

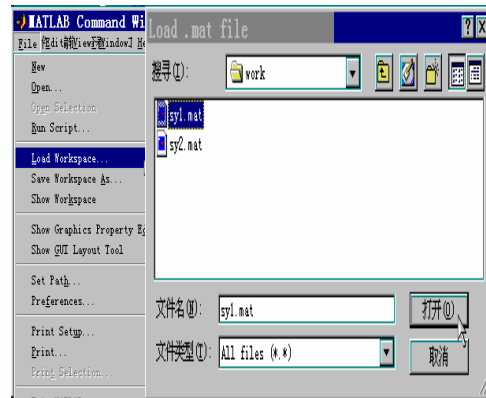


图 1-19 Load.mat file 对话框

#### （六）、Save Workspace 选项

Save Workspace 选项的功能是使用二进制的 .mat 文件保存 MATLAB 工作区的内容，把当前工作空间的所有变量用后缀为 \*.mat 的文件保存起来。单击该选项将弹出一个目录框（参看图 1-20），用户通过该目录框选择文件的存储目录和名字。

#### （七）、Show Workspace 选项

单击 Show Workspace 选项将打开变量浏览器（参看图 1-21）。变量浏览器中显示当前工作空间中所有变量的类型，大小及占用的存储空间。单击变量浏览器下方的 Open 按钮，打开下面的对话框（参看图 1-22）。

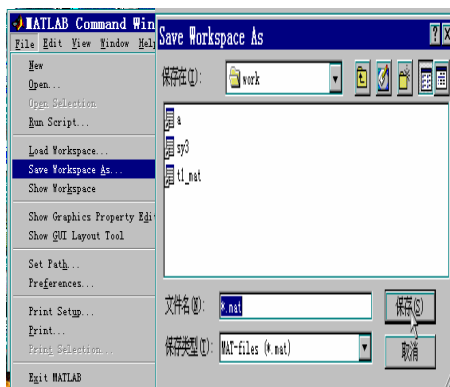


图 1-20 目录框

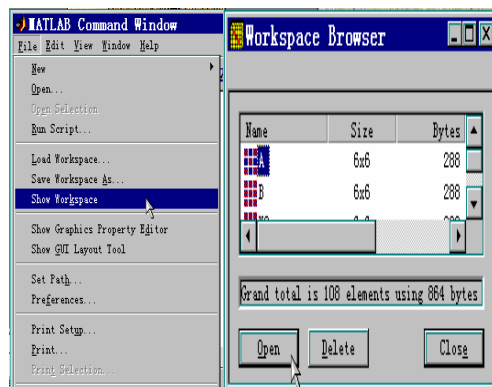


图 1-21 变量浏览器

#### （八）、Show Graphics Property Editor 选项

单击 Show Graphics Property Editor 选项将打开图形属性编辑器（参看图 1-23）。在该图形属性编辑器中，可以选择 MATLAB 已经打开的图形窗口，对每个图形对象的属性值进行修改。

#### （九）、Show GUI Layout Tool 选项

单击 Show GUI Layout Tool 选项, MATLAB 将会打开图形界面控制面板 (参看图 1-24)。我们可以通过选中面板里的图形模板, 再在打开的图形窗口中绘制就可以生成精美的图形界面。

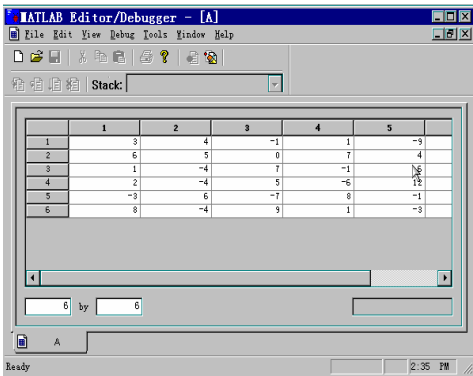


图 1-22

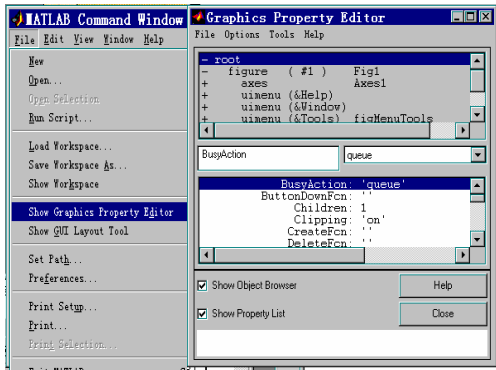


图 1-23 图形属性编辑器

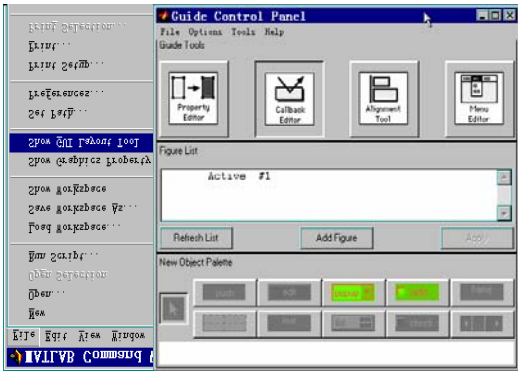


图 1-24 图形界面控制面板

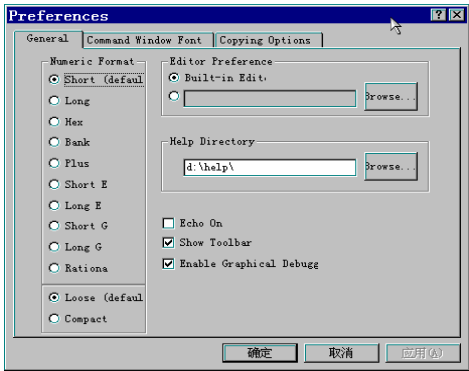


图 1-25 “Preferences”选项对话框

### (十)、 Preferences 选项

单击 菜单栏上“File”下的“Preferences”选项时,弹出如下图所示的对话框 (参看图 1-25),它包含三个选项卡“General”,“Command Window Font”和 “Copying Options”,我们可以 用来调整命令编辑区的显示格式。

1、General 选项卡“Preferences”选项中的 General 选项卡 (参看图 1-26) 的功能是选择程序编辑器,制定帮助目录,命令执行时是否回显,命令与命令之间以及变量名与变量的值之间是否留空行等。

“General”选项卡上左边的 “Numeric Format” 框中,从“Short (def-aule)”到 “Rational” 项 是互斥的,通过它们可以控制命令窗口中计算结果的显示格式,并不影响数据的存储和计算精度。MATLAB 存储和运算的数据只有一种格式,即十六位十进制 (二进制双精度),但屏幕显示有八种格式。对同一个数  $\pi$  的八种格式列表 1-4 如下:

表 1-4

MATLAB 命令	显示	说明
format short	3.1416	小数点后 4 位 (默认)
format long	3.14159265358979	15 位十进制数字
format bank	3.14	小数点后 2 位
format +	+	显示正, 负或零
format short e	3.1416e+000	5 位科学计数法
format long e	3.141592653589793e+000	15 位科学计数法
format rat	355/113	最接近的有理数
Format hex	400921fb54442d18	16 位十六进制数
Loos(default)	选择 <b>稀疏 (Loos)</b> 在命令窗口中的显示结果中加入一些空行, <b>默认 (default) 值为稀疏。</b>	
Compact	<b>紧凑 (Compact)</b> 格式, 压缩掉输出到命令窗口中的许多空行, 一屏上可以显示更多的内容	

## 2、Command Window Font 选项卡

“Preferences”选项中的 Command Window Font 选项卡（参看图 1-26）中的选项控制着命令窗口字体的类型大小，风格（ Bold/ Regular/ Light）以及命令窗口的背景和字体颜色。在图中“Sample”框显示选择的样例。

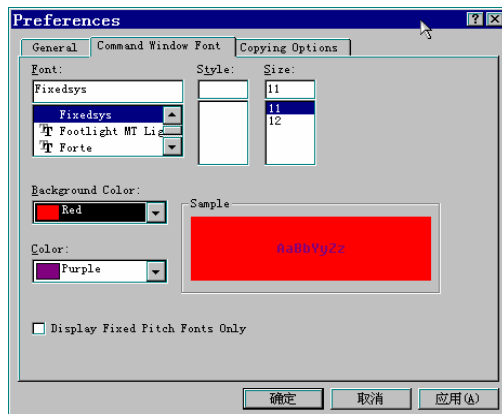


图 1-26 Command Window Font 选项卡

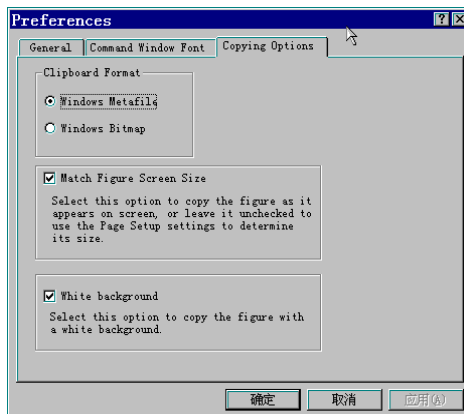


图 1-27 Copying Options 选项卡

## 3、 Copying Options 选项卡

“Preferences”选项中的 Copying Options 选项卡（参看图 1-27）的功能是拷贝 MATLAB 所绘制的图形的形式。选中“Clipboard Format”框中的“Windows Metafile”选项，图形拷贝时采用矢量图形格式；选中“Windows Bitmap”选项，图形将以 Windows 的位图格式拷贝。选中“Honor figure size properties”，会根据需要粘贴纸张大小，自动调整图形；否则，以图形实际大小粘贴。

### 1.3.3 Edit 菜单

Edit 菜单中的命令与 Windows 界面中的 Edit 菜单中的命令的使用方法大部分相同（参看图 1-28）。

下面将 Edit 菜单的各项功能列入表 1-5 中进行介绍。

表 1-5 Edit 菜单的各项功能

菜单命令	功 能
Undo	撤消上一步的操作
Cut	将 选中内容删除，放入剪贴板
Copy	将 选中内容放剪贴板，但不删除所选内容
Paste	将剪贴板的内容放入 MMATLAB 工作窗口
Clear	清除工作空间中的变量
Select All	选中命令窗口中的所有内容
Clear Session	清除命令窗口里的所有显示的内容

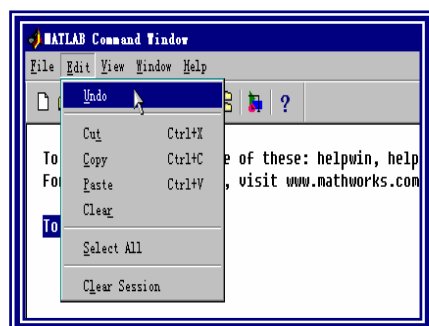


图 1-28 Edit 下拉菜单



图 1-29 View 下拉菜单

[例 7] 在 MATLAB 工作区中输入命令：

```
>> x=pi,y=2^(1/2),z=sqrt(2),t=1/3,
```

运行后输出结果如下：

```
x =
    3.1416
y =
    1.4142
```

```

z =
    1.4142
t =
    0.3333

```

再试用 Edit 菜单的各项功能。

### 1.3.4 其它菜单

MATLAB 除了可以利用 File 和 Edit 菜单中的命令完成对命令窗口的操作以外，还可以利用 View、Window 和 Help 菜单中的命令完成对命令窗口的操作。

#### (一)、View 菜单

View 下拉菜单中的选项 Toolbar,它用于控制是否显示工具栏（参看图 1-29）。

#### (二)、Window 菜单

Window 菜单用于打开多个 MATLAB 窗口，用它在各个窗口之间切换（参看图 1-30）。

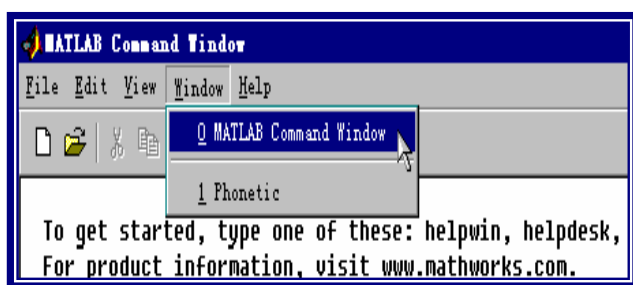


图 1-30 Window 菜单

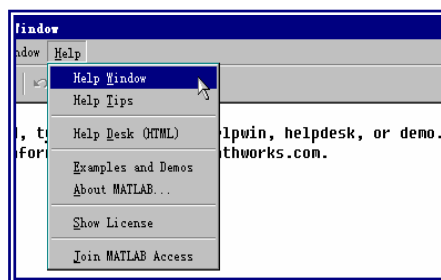


图 1-31 Help 菜单

#### (三)、Help 菜单

Help 菜单参看图 1-31。

下面将对 Help 菜单（参看图 1-31）的各项功能列入表 1-6 中进行介绍。

表 1-6 Help 菜单的各项功能

菜单命令	功 能
Help Window	显示帮助窗口
Help Tips	关于帮助的目录
Help Desk	显示 HTML 格式的帮助用户文件
Examples and Demos	进入 MATLAB 演示窗口
About MATLAB	显示版权信息
Show License	显示用户授权协议
Join MATLAB Access	加入 MATLAB 用户协会

## 1.4 MATLAB 的指令行的操作

### 1.4.1 MATLAB 工作区的功能

MATLAB 工作区是用来接受 MATLAB 命令的内存区域，可以在工作区中用命令实现表 1-7 中的功能。

表 1-7 MATLAB 工作区的功能

指令名称	指令功能
Who 或 whos	显示在当前工作区中的所有变量名，前者显示变量名，后者还显示变量的大小、字节数和类型。
disp(x)	显示 x 的内容，它可以是矩阵或字符串。
which	显示 M 文件 test.m 的目录
type test	在命令窗口下显示 test.m 的内容
cd,chdir,pwd	显示目前的工作目录
load （文件名）	调出 MAT-文件中的数据。也可以调出文本文件，但是文本文件只能是由数字组成的矩阵形式。

diary	建立一个文本文件,记录在 MATLAB 中输入的所有命令和它们的输出,但是不能包括图形。如果想把你的输入存入一个特定的文件中,可使用 <code>diary filename</code> 建立文件。使用 <code>diary off</code> 命令可以停止记录。
What	返回目前目录下 M,MAT,MEX 文件的列表
Echo	控制运行文字指令是否显示
Clc	擦除 MATLAB 工作区中所有显示的内容
Clf	擦除 MATLAB 工作区中的图形
hole	控制当前图形窗口对象是否被刷新
dir,ls	列出指定目录下的文件和子目录清单 <code>path</code> 显示目前的搜索路径,你可以用 File 菜单中的 Set Path 观察和修改路径。
Pack	搜集内存碎块以扩大内存空间
quit	退出工作区可以用,也可选择 File 菜单中 Exit 命令

### 1.4.2 指令行的编辑及应用举例

在 MATLAB 工作区可以用键盘操作,其功能表 1-8 所示:

表 1-8 在 MATLAB 工作区用键盘操作的功能

键盘操作		功 能
↑	Ctrl+p	重调前一行 (用于调出前面的命令进行修改,重新计算)
↓	Ctrl+n	重调下一行
→	Ctrl+b	向前移一个字符
←	Ctrl+f	向后移一个字符
Ctrl+→	Ctrl+r	右移一个字
Ctrl+←	Ctrl+l	左移一个字
Home	Ctrl+a	移动到行首
End	Ctrl+e	移动到行尾
Esc	Ctrl+u	清除一行
Del	Ctrl+d	删除光标处字符
Backspace	Ctrl+h	删除光标前的一个字符
	Ctrl+k	删除到行尾

[例 8] 作曲面  $z = x^2 - y^2$  在  $(-500 \leq x \leq 500, -500 \leq y \leq 500)$  的图形。

解 作曲面  $z = x^2 - y^2$  的图形在 MATLAB 工作区输入一下程序实现:

```
>> x=-500:20:500;y=x;
[X,Y]=meshgrid(x,y);
Z=X.^2-Y.^2;
mesh(X,Y,Z)
```

运行后屏幕显示曲面  $z = x^2 - y^2$  的图形(参看图 1-32)。

[例 9] 绘制 MATLAB 的图标

解 在 MATLAB 工作区输入一下程序实现:

```
>> load logo
surf(L,R,colormap(M),n=length(L(:,1)));
axis([1,n,1,n,-2.8]),view(-37.5,30),
title('MATLAB.The Language of Techical
```

Computing.')

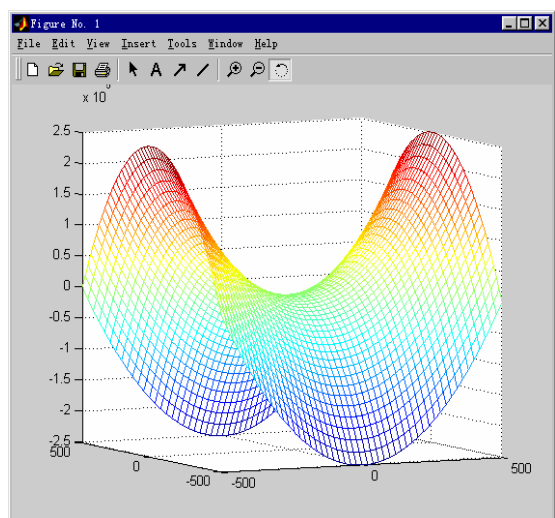


图 1-32



运行后屏幕显示 MATLAB 的图标（参看图 1-33）。

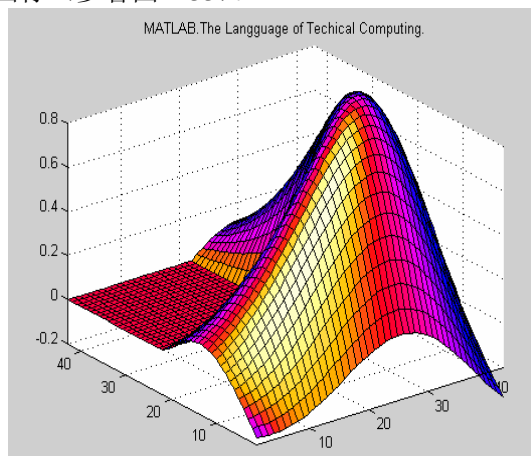


图 1-33

[例 10] （1）在 MATLAB 命令窗口中输入

```
>> 5*sin(0.2*pi)/(1+sqrt(17)).
```

（2）计算出第一个表达式的结果后，再计算第二个表达式的结果。按↑键（或用 Ctrl+p),调出上次的输入。用 ↑ 或→ 移动光标，将 2 改 4，7 改 3。按 Enter 键，MATLAB 给出计算结果。

## 1.5 MATLAB 的在线帮助系统

MATLAB 提供了大量的函数和命令，如果想记住所有的函数及其调用格式几乎是不可能的。为此，MATLAB 提供了非常方便的在线帮助的功能，用户可以容易地获得对想查询的各个函数的信息。在线查询可以由命令 **help** 来获得。

### 1.5.1 查询某一命令或函数

如果你要对某一命令或函数进行查询，直接在 **help** 后跟上该命令或函数名即可，即用下列程序

```
help 函数（命令）名
```

可得到帮助，例如

```
》 help feval
```

### 1.5.2 查询 MATLAB 所有的主题

单独使用 **help** 命令，MATLAB 将列出所有的主题。即

```
》 help
```

### 1.5.3 查询 MATLAB 所有的演示主题

使用命令

```
》 help demos
```

将给出所有的演示主题。

### 1.5.4 查询 MATLAB 的关键词

MATLAB 还提供了关键词查询命令 **lookfor**。例如，如果你想查询与 **complex** 有关的命令和函数,则可以在 MATLAB 的工作空间中键入命令

```
》 lookfor complex
```

### 1.5.5 使用 **help** 菜单查询帮助信息

MATLAB 提供了 Windows 下的查询方法，这和一般 Windows 程序的联机帮助系统是一致的。你可以使用 MATLAB 的工作空间中的 **help** 菜单（见图 1-34）查询帮助信息。

例如，你用鼠标单击 **help** 菜单中的子菜单 **Help Tips** 选项，则将出现如图 1-35 所示的帮助界面。你在此界面中能查询感兴趣的问题。例如，如果你想查询 MATLAB 所有的演示主题,请单击按钮 **Home**,则弹出

如图 1-36 所示的帮助主题界面。

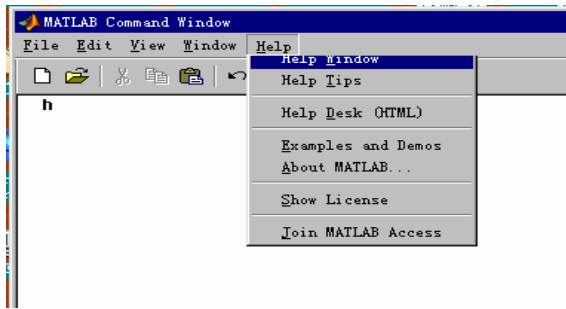


图 1-34

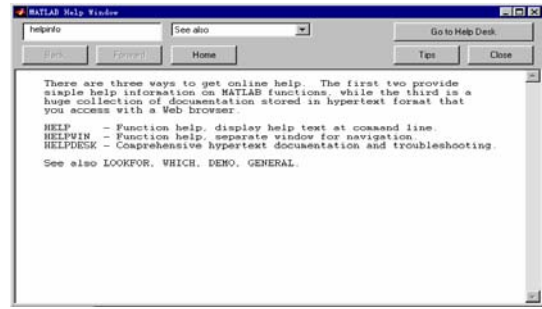


图 1-35 帮助界面

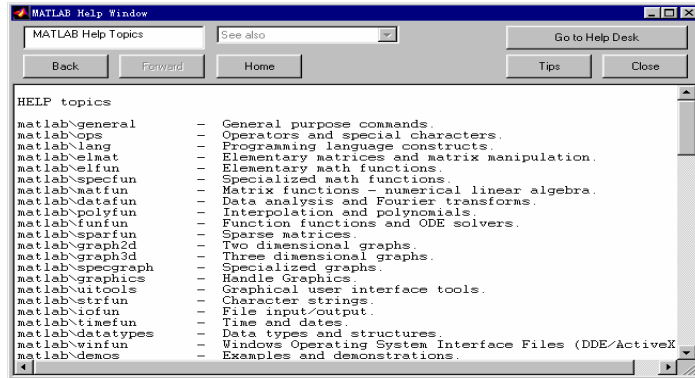


图 1-36 帮助主题界面

## 1.6 演示与漫游

### 1.6.1 演示 (demo)

直接在命令窗口输入命令 `demo`,就打开 MATLAB 演示窗口,如图 1-37 所示。图中有三个文本框,左边的文本框中是总的结构,可选择需要演示的类别。右上方的文本框是对选中的类别的介绍。右下方的文本框是供选择的示例名称,在这里选择需要演示的例子,再单击按钮 `Run...`,即打开该例子的演示窗口。

我们在左边的文本框中选择 `MATLAB` 下的“`Matrices`”,在右下方的文本框中选择的“`Basic matrix operations`”,再单击按钮 “`Run...`”,这时会打开如图 1-38 所示的演示画面。在右图中,如果要逐页观察演示,可以按“`Start`”按钮,再逐次按下 “`Next`”; 也可以只按下“`AutoPlay`” 按钮,会自动播放各页的内容。按钮“`Info`”给出该演示程序的相关信息,”`Close`”关闭这个演示窗口。

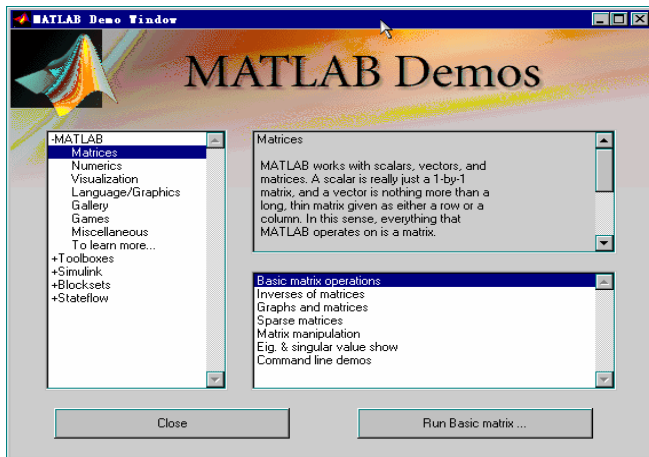


图 1-37 MATLAB 演示窗口

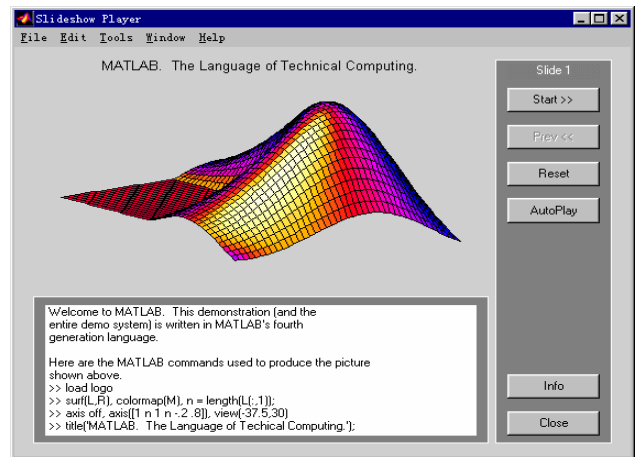


图 1-38 演示窗口

### 1.6.2 漫游(tour)

该程序的用法和 demo 类似，在命令窗口直接键入 `tour`，就会打开图 1-39 所示的窗口。用户可以单击该窗口中的响应主题按钮来打开相应的内容。



图 1-39



图 1-40

例如用鼠标单击 Stateflow 按钮，将会出现 Stateflow 窗口（参看图 1-40）。我们在该窗口左边的文本框中选择“Stateflow Examples”，在右下方的文本框中选择“Automatic Transmission Control”，再单击按钮“Run: Automatic...”，这时会打开如图 1-41 所示的演示画面，用鼠标单击在右上角 sf-car\*模型窗口中的 Simulation 菜单下的子菜单 Start，将会开始模拟演示。

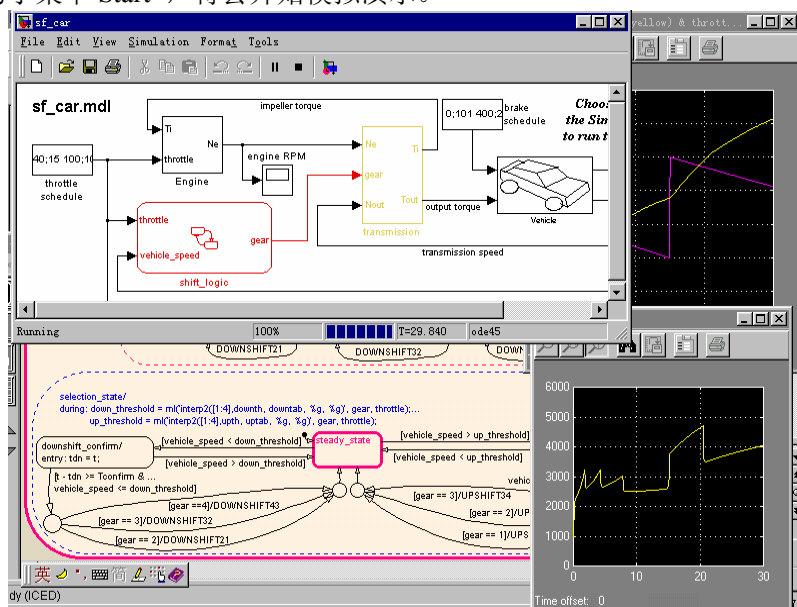


图 1-41

## 第二章 MATLAB 的基础语法

### 2.1 矩阵的输入与运算

MATLAB 的主要数据对象是矩阵，标量、数组、行向量、列向量都是它的特例，最基本的功能是进行矩阵运算，MATLAB 对于矩阵和数组的操作有一些特殊的规定。

矩阵输入有多种办法，如直接输入每个元素；由语句或函数生成；在 M-文件（以后介绍）中生成等。

#### 2.1.1 矩阵的直接输入

MATLAB 中不用描述矩阵的类型和维数，它们由输入的格式和内容决定。小矩阵可以用排列各个元素的方法输入，同一行元素用逗号或空格分开，不同行的元素用分号或回车分开。如，在 MATLAB 工作窗口输入程序：

```
» A=[1,2,3;4,5,6]
```

或

```
» A=[1 2 3;4 5 6]
```

或

```
» A=[1 2 3
      4 5 6]
```

都输入了一个  $2 \times 3$  矩阵 A，屏幕上显示输出变量为

```
A=
    1    2    3
    4    5    6
```

矩阵中的元素可以用它的行、列数表示，例如，在 MATLAB 工作窗口输入程序：

```
» a=A(2,1)%MATLAB 区分大小写字母，a 和 A 是不同的变量.
```

运行后输出结果如下：

```
a=
    4
```

或者不指定输出变量，MATLAB 将回应 ans（answer 的缩写），如，在 MATLAB 工作窗口输入程序：

```
» A(2,1)
```

运行后输出结果如下：

```
ans=
    4
```

A 输入后一直保存在工作空间中，可随时调用，除非被清除或替代。

可以直接修改矩阵的元素，如，在 MATLAB 工作窗口输入程序：

```
» A(2,1)=7,A(3,4)=1,
```

运行后输出结果如下：

```
A=
    1    2    3
    7    5    6

A=
    1    2    3    0
    7    5    6    0
    0    0    0    1
```

原来的 A 没有 3 行 4 列，MATLAB 自动增加行列数，对未输入的元素赋值 0。

### 2.1.2 函数生成的矩阵

MATLAB 提供了一些函数来构造特殊矩阵，如，在 MATLAB 工作窗口输入程序：

```
» w=zeros(2,3)% 2×3 零矩阵.
```

运行后输出结果如下：

```
w=
    0    0    0
    0    0    0
```

在 MATLAB 工作窗口输入程序：

```
» u=ones(3)%3×3 全 1 矩阵，方阵只需输入行数，这几个矩阵均如此
```

运行后输出结果如下：

```
u=
    1    1    1
    1    1    1
    1    1    1
```

在 MATLAB 工作窗口输入程序：

```
» v=eye(3,4)%3×4 对角线为 1 的矩阵.
```

运行后输出结果如下：

```
v=
    1    0    0    0
    0    1    0    0
    0    0    1    0
```

在 MATLAB 工作窗口输入程序：

```
» x=rand(1,3)%1×3(0,1)均匀分布随机矩阵.
```

运行后输出结果如下：

```
x=
    0.9501    0.2311    0.6068
```

还有  $m \times n$  标准正态分布矩阵 `randn(m,n)`,  $n$  阶 Hilbert 矩阵 `hilb(n)`,  $n$  阶幻方矩阵 `magic(n)`等。

### 2.1.3 矩阵的裁剪与拼接

从一个矩阵中取出若干行(列)构成新矩阵称为**裁剪**, MATLAB 中“`:`”是非常重要的裁剪工具, 如, 在 MATLAB 工作窗口输入程序:

》 `A(3,:) % A 的第 3 行.`

运行后输出结果如下:

```
ans =  
      0  0  0  1
```

在 MATLAB 工作窗口输入程序:

》 `A(:,2) % A 的第 2 列.`

运行后输出结果如下:

```
ans =  
      2  
      5  
      0
```

在 MATLAB 工作窗口输入程序:

》 `B=A(1:2,:) % A 的第 1,2 行.`

运行后输出结果如下:

```
B =  
      1  2  3  0  
      7  5  6  0
```

在 MATLAB 工作窗口输入程序:

》 `C=B(:,2:4) % B 的第 2~4 列.`

运行后输出结果如下:

```
C =  
      2  3  0  
      5  6  0
```

在 MATLAB 工作窗口输入程序:

》 `D=A(2:3,2:2:4) % A 的第 2,3 行, 第 2,4 列.`

运行后输出结果如下:

```
D =  
      5  0  
      0  1
```

请特别注意“`:`”的用法, 想想在 MATLAB 工作窗口输入程序:

》 `A(1:2:3,4:-1:2)`

应该有什么结果。在 MATLAB 工作窗口输入程序:

》 `D(:,1)=[] % 删除 D 的第 1 列, [] 为空集符号.`

运行后输出结果如下:

```
D =  
      3  2  
      6  5
```

将几个矩阵接在一起称为**拼接**, 左右拼接时行数要相同, 上下拼接时列数要相同, 如, 在 MATLAB 工作窗口输入程序:

》 `E=[D,zeros(2,1)]`

运行后输出结果如下:

```
E =  
      0  3  2  0  
      0  6  5  0
```

在 MATLAB 工作窗口输入程序:

》 `F=[A(1:2,:);eye(1,4)]`

运行后输出结果如下:

```
F =  
      1  2  3  0  
      7  5  6  0  
      1  1  1  1
```

在 MATLAB 工作窗口输入程序：

```
》 G=[C,ones(2);9,F(1,:)]
```

运行后输出结果如下：

```
G=
    2    3    0    1    0
    5    6    0    0    1
    9    1    2    3    0
```

### 2.1.4 矩阵的运算

MATLAB 中提供了下列矩阵运算符：

+ 加法；- 减法；' 转置；\* 乘法；^ 乘幂；\ 左除；/ 右除。

它们要符合矩阵运算的规律，如果矩阵的行列数不符合运算符的要求，将产生错误信息。这里只将左除和右除的用法叙述如下：

设 A 是可逆矩阵， $AX=B$  的解是 A 左除 B，即  $X=A\backslash B$ （当 B 为列向量时，得到方程组的解）； $XA=B$  的解是 A 右除 B，即  $X=B/A$ 。

还应注意标量与矩阵进行上述运算的含义，请看在 MATLAB 工作窗口输入程序：

```
》 E+3
```

运行后输出结果如下：

```
ans=
    3    6    5    3
    3    9    8    3
```

## 2.2 数组及其运算数组及其运算

一行或一列的矩阵是向量或数组，在 MATLAB 中向量和数组虽然形式上一致，却是两个不同的概念，遵循不同的运算规则。

### 2.2.1 数组的输入

除了像  $1 \times n$  矩阵（行向量）一样地输入外，数组常采用“:”和函数 linspace 两种输入方式，它们的用法可以从下面的例子知道。在 MATLAB 工作窗口输入程序：

```
》 a=1:5 %从 1 到 5 公差为 1 的等差数组。
```

运行后输出结果如下：

```
a=
    1    2    3    4    5
```

在 MATLAB 工作窗口输入程序：

```
》 b=1:2:7 %从 1 到 7 公差为 2 的等差数组。
```

运行后输出结果如下：

```
b=
    1    3    5    7
```

如果输入  $b=1:2:8$ ，得到同样结果。在 MATLAB 工作窗口输入程序：

```
》 c=6:-3:-6 %从 6 到 -6 公差为 -3 的等差数组。
```

运行后输出结果如下：

```
c=
    6    3    0   -3   -6
```

在 MATLAB 工作窗口输入程序：

```
》 b=[0:2:8,ones(1,3)] %等差数组和行向量拼接。
```

运行后输出结果如下：

```
b=
    0    2    4    6    8    1    1    1
```

在 MATLAB 工作窗口输入程序：

```
》 linspace(0,1,9) %从 0 到 1 共 9 个数值的等差数组。
```

运行后输出结果如下：

```
ans=
    0    0.1250    0.2500    0.3750    0.5000    0.6250    0.7500    0.8750    1.0000
```

**linspace(a,b,n)**生成从 a 到 b 共 n 个数值的等差数组，公差不必给出。与它相仿的是



logspace(a,b,n)

生成从  $10^a$  到  $10^b$  共  $n$  个数值的等比数组。

4 等分  $\pi$  (MATLAB 中  $\pi$  的符号是 pi) 的数组可以用下面两种方式输入。在 MATLAB 工作窗口输入程序：

```
» x=0:pi/4:pi
```

运行后输出结果如下：

```
x=
    0    0.7854    1.5708    2.3562    3.1416
```

在 MATLAB 工作窗口输入程序：

```
» x=linspace(0,pi,5)
```

运行后输出结果如下：

```
x=
    0    0.7854    1.5708    2.3562    3.1416
```

## 2.2.2 数组的运算

数组除了作为  $1 \times n$  矩阵 (行向量) 遵循矩阵运算规则外, MATLAB 还为数组提供了一些特殊的运算:

. \* 乘法; . ^ 乘幂; . \ 左除; ./ 右除。

通过下面的例子说明这些运算。在 MATLAB 工作窗口输入程序：

```
» a=[1 2 3 4 5],b=[1 3 5 7 9],c=a.*b
```

运行后输出结果如下：

```
a=
    1     2     3     4     5
b=
    1     3     5     7     9
c=
    1     6    15    28    45
```

在 MATLAB 工作窗口输入程序：

```
» d=a.^2,e=b.^a
```

运行后输出结果如下：

```
d=
    1     4     9    16    25
e=
    1     9    125   2401  59049
```

在 MATLAB 工作窗口输入程序：

```
» f=a./b,g=a.\b
```

运行后输出结果如下：

```
f=
    1.0000    0.6667    0.6000    0.5714    0.5556
g=
    1.0000    1.5000    1.6667    1.7500    1.8000
```

可以看出, 数组运算是对应元素的运算。

## 2.3 语句、变量和表达式

### 2.3.1 语句形式

MATLAB 语句的一般形式为：

**变量=表达式**

若变量和“=”省略, 则名为 ans 的变量自动建立。如果你输入一个语句并以回车结束, 则在工作区中显示计算的结果; 如果语句以分号“;”结束, MATLAB 只进行计算, 不显示计算的结果。如果一个表达式太长, 可以用续行号 ... 将其延续到下一行。此外, 一行中可以写几个语句, 它们之间要用逗号或分号分开。如, 在 MATLAB 工作窗口输入程序：

```
» a=[1 2 3 4 5];b=[1 3 5 7 9];
   c=a.*b, d=a*b', e=a'*b
```

运行后输出结果如下：

```
c =
```

```

      1      6     15     28     45
d =
    95
e =
      1      3      5      7      9
      2      6     10     14     18
      3      9     15     21     27
      4     12     20     28     36
      5     15     25     35     45

```

### 2.3.2 变量

MATLAB 的变量由字母、数字和下划线组成，最多 31 个字符，区分大小写字母，第一个字符必须是字母。对于变量，MATLAB 不需要任何类型的说明或维数语句。当输入一个新变量名时 MATLAB 自动建立变量并为其分配内存空间。

MATLAB 有几个特殊的量：

pi 圆周率  $\pi$ ；eps 最小浮点数；Inf 正无穷大，特指  $1/0$ ；NaN 不定值，特指  $0/0$ ；i,j 虚数单位。

在 MATLAB 工作窗口输入程序：

```

    》 a=[0 1 0],b=[1 0 0],c=a./b

```

运行后输出结果如下：

```

a =
      0      1      0
b =
      1      0      0
Warning: Divide by zero.
c =
      0    Inf    NaN

```

### 2.3.3 字符串

字符串是用单引号括起来的字符集合，可以像向量一样进行拼接和裁剪，如，在 MATLAB 工作窗口输入程序：

```

    》 s1='Hello';s2='every';s3='body';
    s=[s1',' ,s2',' ,s3],ss=s(1:5)

```

运行后输出结果如下：

```

s=
    Hello,every body
ss=
    Hello

```

## 2.4 函数

### 2.4.1 标量函数

将常用的列出如下，只作必要的注释。

三角函数： sin cos tan cot sec csc asin acos  
atan acot asec acsc sinh cosh tanh asinh acosh  
atanh

其它基本函数： sqrt exp log log10 abs（绝对值或复数模） round（四舍五入取整） floor（向  $-\infty$  方向取整） ceil（向  $+\infty$  方向取整） fix（向 0 方向取整） sign（符号函数） real（取实部） imag（取虚部） angle（取幅角） rats（有理逼近）

这些函数本质上是作用于标量的，当它们作用于矩阵（或数组）时，是作用于矩阵（或数组）的每一个元素。在 MATLAB 工作窗口输入程序：

```

    》 x=(0:0.2:1)*pi;
    y=sin(x)

```

运行后输出结果如下：

```

y=
    0    0.5878    0.9511    0.9511    0.5878    0.0000

```

在 MATLAB 工作窗口输入程序：

```
» a=[-3.5 4.6];  
b=round(a),c=floor(a),  
d=ceil(a),e=fix(a),f=rats(a)
```

运行后输出结果如下：

```
b =  
    -4     5  
c =  
    -4     4  
d =  
    -3     5  
e =  
    -3     4  
f =  
   -7/2   23/5
```

另一个计算函数值的命令是 `feval(F,x)`，F 是表示函数名的字符串，如在 MATLAB 工作窗口输入程序：

```
» x=(0:0.2:1)*pi;y=feval('sin',x)✓
```

得到与上面同样的结果。

## 2.4.2 向量函数

有些函数只有当它们作用于（行或列）向量时才有意义，称为向量函数，这些函数也可以作用于矩阵，此时它产生一个行向量，行向量的每个元素是函数作用于矩阵相应列向量的结果。常用的有：

`max`   `min`   `sum`（和）   `length`（长度）   `mean`（平均值）  
`median`（中值）   `prod`（乘积）   `sort`（从小到大排列）

在 MATLAB 工作窗口输入程序：

```
» a=[4 3.1 -1.2 0 6];  
b=min(a),c=sum(a),d=median(a),e=sort(a)
```

运行后输出结果如下：

```
b =  
   -1.2000  
c =  
   11.9000  
d =  
    3.1000  
e =  
   -1.2000    0    3.1000    4.0000    6.0000
```

## 2.4.3 矩阵函数

MATLAB 有大量的处理矩阵的函数，从其作用来看可分为两类：构造矩阵的函数；进行矩阵计算的函数。对于前者，我们已经介绍了

`zeros`（0 阵）   `ones`（1 阵）   `eye`（单位阵）   `rand`（随机阵）   `randn`（随机阵）

还有

`diag`（生成或提取对角阵）   `triu`（生成或提取上三角阵）   `tril`（生成或提取下三角阵）

等，将在实验 5 中介绍。

对于后者，常见的有

`size`（大小）`det`（行列式）`rank`（秩）`inv`（逆矩阵）`eig`（特征值）`trace`（迹）`expm`（矩阵指数）`poly`（特征多项式）等，还有

`norm`（范数）   `cond`（条件数）   `lu`（LU 分解）   `qr`（正交分解）   `svd`（奇异值分解）等。

## 2.5 程序设计

MATLAB 提供了一个完善的程序设计语言环境，使我们能方便地编制复杂的程序，完成各种计算。本节先简要介绍关系、逻辑运算和条件、循环语句等许多高级语言都具备的、编程的重要手段，再着重介绍 MATLAB 所特有的 M-文件。

### 2.5.1 关系和逻辑运算

MATLAB 的关系运算符有:

< 小于; > 大于; <= 小于或等于; >= 大于或等于; == 等于; ~= 不等于。

关系运算比较两个数值, 当指出的关系成立时结果为 1 (表示真), 否则为 0 (表示假)。

关系运算可以作用于两个同样大小的矩阵或数组, 结果是一个 0,1 矩阵或数组, 每个分量代表相应的矩阵或数组分量的关系运算结果, 例如, 在 MATLAB 工作窗口输入程序:

```
》 A=1:5,B=5:-1:1
```

运行后输出结果如下:

```
A =  
    1    2    3    4    5  
B =  
    5    4    3    2    1
```

在 MATLAB 工作窗口输入程序:

```
》 C=A>=4
```

运行后输出结果如下:

```
C =  
    0    0    0    1    1
```

在 MATLAB 工作窗口输入程序:

```
》 D=A==B
```

运行后输出结果如下:

```
D =  
    0    0    1    0    0
```

下面举一个例子说明如何将关系运算和数值运算结合起来。在 MATLAB 工作窗口输入程序:

```
》 x=(-3:3)/3
```

运行后输出结果如下:

```
x =  
-1.0000 -0.6667 -0.3333  0  0.3333  0.6667  1.0000
```

在 MATLAB 工作窗口输入程序:

```
》 sin(x)./x
```

运行后输出结果如下:

Warning: Divide by zero

```
ans =  
    0.8415    0.9276    0.9816   NaN    0.9816    0.9276    0.8415
```

在计算  $\sin x/x$  时给出了警告信息, 是因为第 4 个数据  $\sin 0/0$  没有定义, MATLAB 返回 NaN。为了避免这种情况出现可以用最小浮点数  $\text{eps}$  来代替 0, 在 MATLAB 工作窗口输入程序:

```
》 x=(-3:3)/3; x=x+(x==0)*eps; sin(x)./x
```

运行后输出结果如下:

```
ans =  
    0.8415    0.9276    0.9816    1.0000    0.9816    0.9276    0.8415
```

给出了  $\sin x/x$  在  $x=0$  时正确的极限值。

MATLAB 的逻辑运算符有:

& 与运算; | 或运算; ~ 非运算。

它们满足熟知的运算规则:

a	b	a&b	a b	~a
0	0	0	0	1
1	0	0	1	0
1	1	1	1	0

逻辑运算将任何非零元素视为 1 (真)。逻辑运算也可以作用于矩阵或数组。在 MATLAB 工作窗口输入程序:

```
》 a=1:9,b=9-a,c=~(a>4),d=(a>=3)&(b<6)
```

运行后输出结果如下:

```

a=
    1     2     3     4     5     6     7     8     9
b=
    8     7     6     5     4     3     2     1     0
c=
    1     1     1     1     0     0     0     0     0
d=
    0     0     0     1     1     1     1     1     1

```

MATLAB 还提供了一些关系和逻辑函数，常见的有 `all` 和 `any`，用法是：

`y=all(x)` 若 `x` 为向量，当所有元素非零时 `y=1`，否则 `y=0`；若 `x` 为矩阵，`all` 作用于列元素，`y` 为行向量。

`y=any(x)` 若 `x` 为向量，当有一元素非零时 `y=1`，否则 `y=0`；若 `x` 为矩阵，`all` 作用于列元素，`y` 为行向量。

例如，在 MATLAB 工作窗口输入程序：

```

» a=[1 0 -5 0;-3 0 8 2],
  b=all(a),c=any(a),d=all(b),e=any(c)

```

运行后输出结果如下：

```

a=
     1     0    -5     0
    -3     0     8     2
b=
     1     0     1     0
c=
     1     0     1     1
d=
     0
e=
     1

```

## 2.5.2 条件和循环语句

条件和循环语句属于流控制语句，MATLAB 的流控制语句主要有四个：`if`、`while`、`for`、`switch`，它们都用 `end` 结束。

### （一）、if 语句

条件语句 `if` 最简单的用法是：

```

if <关系表达式>
    <语句 1>
end

```

如果关系表达式的值为 1，则语句 1 执行；否则，执行 `end` 的后续命令。

`if` 语句的另外一种用法是：

```

if <关系表达式>
    <语句 1>
else
    <语句 2>
end

```

如果关系表达式的值为 1，则语句 1 执行；否则（关系表达式的值为 0），语句 2 执行，然后执行 `end` 的后续命令。

当我们有多个选择时还可以用下列结构：

```

if <关系表达式 1>
    <语句 1>
elseif <关系表达式 2>
    <语句 2>
...
elseif <关系表达式 n>
    <语句 n>
else

```

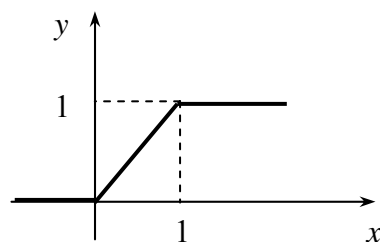


图 2-1

<语句 n+1>

end

如果关系表达式  $j(j=1,2,\dots,n)$  的值为 1, 则语句  $j$  执行, 然后执行 end 的后续命令; 否则, 语句 2 执行, 然后执行 end 的后续命令。例如, 可用以下程序得到图 2-1 的分段函数。

```
function y=fun(x)
if      x<=0
    y=0;
elseif  x<=pi/2
    y=x;
else
    y=1;
end
```

在 MATLAB 工作窗口输入程序:

```
>>x=31; y=fun(x)
```

运行后输出结果如下:

```
y =
    1
```

## (二)、for 语句

循环语句 for 的一般形式为

```
for    <循环参数>=<初值>:<步长>:<终值>
    <语句>
```

end

步长为 1 时可以省略。对于每一参数值, 语句都重复执行。

当作多重循环时 for 语句可以嵌套使用, 如用以下程序可以生成 Hilbert 矩阵。

```
function A=f(n,m)
A=zeros(n,m);
for i=1:n
    for j=1:m
        A(i,j)=1/(i+j-1);
    end
end
```

format rat

在 MATLAB 工作窗口输入程序:

```
>> n=4,m=5,A=f(n,m)
```

运行后输出结果如下:

```
A =
    1    1/2    1/3    1/4    1/5
    1/2    1/3    1/4    1/5    1/6
    1/3    1/4    1/5    1/6    1/7
    1/4    1/5    1/6    1/7    1/8
```

for 语句的循环参数可以是任意的数组或矩阵, 循环参数依次取数组元素的值, 或按矩阵的列依次取值。

## (三)、while 语句

for 循环主要应用于已知循环次数的情况, 如果不知道循环次数, 可以使用 while 循环来完成, 其表达方式为

```
while    <关系表达式>
    <语句>
```

end

当关系表达式的值为 1 (真) 时, 语句被反复执行, 直至关系表达式为 0 (假) 时终止, 如, 在 MATLAB 工作窗口输入程序:

```
» n=0;EPS=1;
while (1+EPS)>1
    EPS=EPS/2;
    n=n+1;
end
EPS=2*EPS;
```



```
n,format short E,EPS
```

运行后输出结果如下：

```
n=
53
EPS =
2.2204e-016
```

这个例子给出了计算 MATLAB 中特殊常量 `eps` 的过程（我们用大写 `EPS` 以便与 `eps` 相区别）。`EPS` 不断地被 2 除直到  $(1+EPS)>1$  为假时终止。这里需要注意的是，MATLAB 用 16 位数来表示数据，因此当 `EPS` 接近  $10^{-16}$  时，它会认为  $(1+EPS)>1$  不成立。

MATLAB 还提供了跳出循环的 `break` 语句，遇到此语句立即执行此循环 `end` 的后续语句，如上面计算 `eps` 的程序可以改为：

```
》 EPS=1;
for n=1:100
    EPS=EPS/2;
    if (1+EPS)<=1
        EPS=2*EPS;
        break
    end
end
n,format short E,EPS
```

运行后输出同样的结果。

#### （四）、switch 语句

switch 语句根据表达式的值来执行相应的语句，用法如下：

```
switch    <表达式>
case      value1
    <语句 1>
case      value2
    <语句 2>
otherwise
    <语句 n>
end
```

## 第三章 MATLAB 图形制作

### 3.1 函数 plot

`plot` 是 MATLAB 中最常用的画平面曲线的函数，它的主要功能是用于绘制显式函数  $y = f(x)$  和参数式函数  $x = x(t)$ ,  $y = y(t)$  的平面曲线。Plot 函数的调用格式如下：

**plot(x,y,'可选项 s')**

其中  $x$  是曲线上的横坐标， $y$  是曲线上的纵坐标，'可选项  $s$ ' 中通常包含确定曲线颜色、线型、两坐标轴上的比例等等的参数。用户在作图时可以根据需要选择可选项。如果用户在绘图时不用可选项，那么 Plot 函数将制动选择一组默认值，画出曲线。下面我们将通过例题逐步介绍。

**【例 1】** 作出通过表 3-1 中数据点  $(x_i, y_i)$  的折线。

表 3-1 例 1 的数据点  $(x_i, y_i)$

x	1	2	3	4	5	6	7	8	9
y	0	0.68	0.70	0.99	0.83	0.20	0.80	0.12	0.98

**解** 因为横坐标为序号 1, 2, ..., 9，所以作图时，只需输入下列程序：

```
>> y=[0 0.68 0.70 0.99 0.83 0.20 0.80 0.12 0.98];
plot(y) ✓
```

运行后屏幕显示数据点  $(x_i, y_i)$  的折线（见图 3-1）。

**【例 2】** 作出函数  $y=2e^{-0.5x} \cos(3x-4)$  在区间  $[0, 6\pi]$  上的图形。

**解** 输入下列程序：

```
>> x=linspace(0,6*pi,500);% 在[0,6π]上取 500 个点
y=2*exp(-0.5*x).*cos(3*x-4);
plot(x,y)✓
```

运行后屏幕显示函数  $y=2e^{-0.5x}\cos(3x-4)$  在区间  $[0,6\pi]$  上 500 个点连成的光滑的余弦曲线(见图 3-2)。

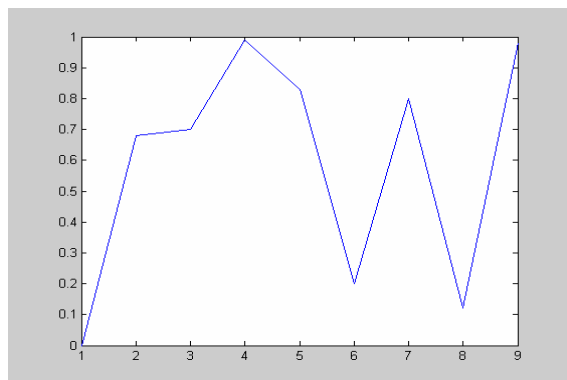


图 3-1 表 3-1 中数据点  $(x_i, y_i)$  的折线

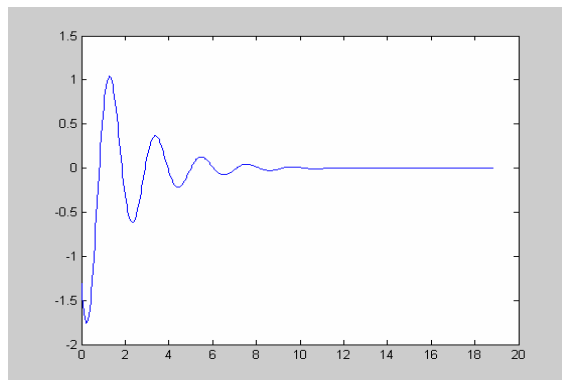


图 3-2  $y=2e^{-0.5x}\cos(3x-4)$  在区间  $[0,6\pi]$  上的曲线

[例 3] 作出参数函数  $x=\sin(mt)$ ,  $y=\cos(nt)$  在区间  $[0,2\pi]$  上的图形, 其中  $a=7, m=3, n=5$ 。

解 输入下列程序:

```
>> a=7;m=3;n=5;
t=0:pi/60:2*pi;
x=a*sin(m*t); y=a*cos(n*t);
plot(x,y,'b-.')
```

运行后屏幕显示参数函数  $x=\sin(mt)$ ,  $y=\cos(nt)$  在区间  $[0,2\pi]$  上 121 个点连成的光滑的曲线(见图 3-3)。

读者可以改变此例中  $m$  与  $n$  的值, 例如取  $(m,n) = (2,3), (2,5), (2,7), (3,5), (3,7), (4,5), (4,7), \dots, (14,15)$  等等, 将会画出许多有趣的曲线。

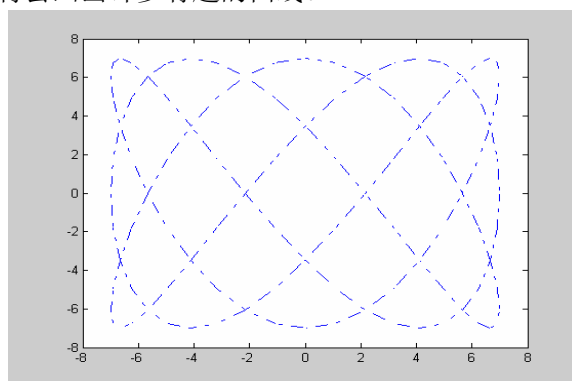


图 3-3 参数函数  $x=\sin(mt)$ ,  $y=\cos(nt)$  在区间  $[0,2\pi]$  上的曲线

## 3.2 函数 ezplot

ezplot 是 MATLAB 中另外一种画平面曲线的函数, 它的主要功能是用于绘制隐函数  $F(x, y) = 0$ 、参数式函数  $x = x(t)$ ,  $y = y(t)$  和显式函数  $y = f(x)$  的平面曲线。下面分别给以介绍。

### 3.2.1 绘制隐函数的平面曲线

(一)、绘制隐函数  $F(x, y) = 0$  在  $a \leq x \leq b$  和  $c \leq y \leq d$  的平面曲线

1、绘制隐函数  $F(x, y) = 0$  在  $a \leq x \leq b$  和  $c \leq y \leq d$  上的图形的调用格式为  
ezplot (F, [a,b,c,d])

2、当  $a=c, b=d$  时, 即绘制隐函数  $F(x, y) = 0$  在  $a \leq x \leq b$  和  $a \leq y \leq b$  的图形的调用格式为

ezplot (F, [a,b])

$$\frac{\sin \sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}} = 1$$

[例 4] 绘制隐函数  $\frac{\sin \sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}} = 1$  在  $-6\pi \leq x \leq 6\pi, -5.6\pi \leq y \leq 6.6\pi$  上的图形。

解 输入命令:

```
>> ezplot('sin(sqrt(x^2+y^2))/sqrt(x^2+y^2)',[-6*pi,6*pi],[-5.6*pi,6.6*pi])
```

运行后画出图形, 见图 3-4。

[例 5] 绘制隐函数  $x^3 + y^3 - 5xy = -1/5$  在  $-3 < x < 3, -3 < y < 3$  上的图形。

解 输入命令:

```
>> ezplot('x^3 + y^3 - 5*x*y + 1/5',[-3,3])
```

运行后画出  $x^3 + y^3 - 5xy = -1/5$  在  $-3 < x < 3, -3 < y < 3$  上的图形, 见图 3-5。

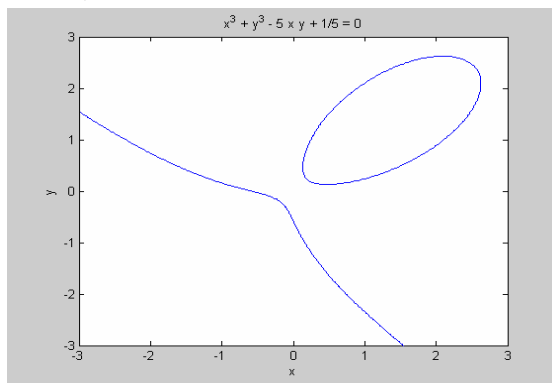
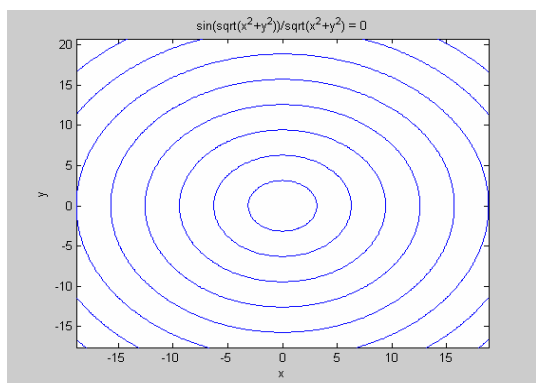


图 3-4  $u^2 - v^2 = 1$  在  $-3 < u < 4, -2 < v < 3$  上的图形 图 3-5  $x^3 + y^3 - 5xy = -1/5$  在  $-3 < x < 3, -3 < y < 3$  上的图形

(二)、不指定  $x, y$  的范围绘制隐函数  $F(x, y) = 0$  的平面曲线

绘制隐函数  $F(x, y) = 0$  的图形时, 如果不易估计  $x, y$  的范围, 可以调用格式

ezplot (F)

缺省值域为  $-2\pi \leq x \leq 2\pi$  和  $-2\pi \leq y \leq 2\pi$ 。

[例 6] 绘制隐函数  $x^4 + y^4 - 8x^2 - 10y^2 = a, a = -16$  的图形。

解 输入命令: >>syms x y

```
F=x^4+y^4-8*x^2-10*y^2+16; ezplot (F)
```

运行后画出隐函数  $x^4 + y^4 - 8x^2 - 10y^2 = -16$  的图形, 见图 3-6。如果改变参数  $a$ , 图形的形状也会随之改变。

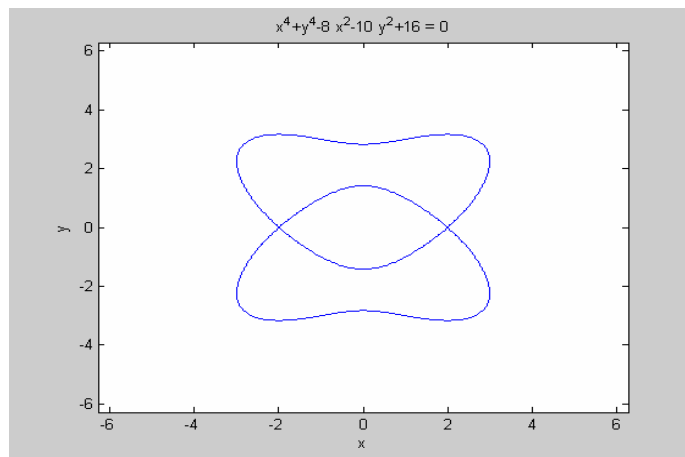


图 3-6 隐函数  $x^4 + y^4 - 8x^2 - 10y^2 = -16$  的图形

(三)、绘制隐函数  $F(x, y) = 0$  在  $xmin \leq x \leq xmax$  和  $ymin \leq y \leq ymax$  上的平面曲线, 可调用格式

ezplot (F, [xmin,xmax,ymin,ymax])

### 3.2.2 绘制参数函数

(一)、绘制参数函数  $x = x(t)$  和  $y = y(t)$  在  $t_{\min} \leq t \leq t_{\max}$  上的平面曲线的调用格式  
ezplot (x,y, [tmin,tmax])

[例 7] 绘制参数函数  $x = \sin(at) \cos t$ ,  $y = \sin(at) \sin t$ ,  $a = 3$  在  $[0, b\pi]$ ,  $b = 1$  上的图形。

解 输入命令: `>> ezplot('sin(3*t)*cos(t)','sin(3*t)*sin(t)',[0,pi])`

运行后可得图 3-7。如果改变参数  $a$  和  $b$ , 将会得到很多有趣的图形。

(二)、绘制参数函数  $x = x(t)$  和  $y = y(t)$  在  $0 \leq t \leq 2\pi$  上的平面曲线的调用格式  
ezplot (x,y)

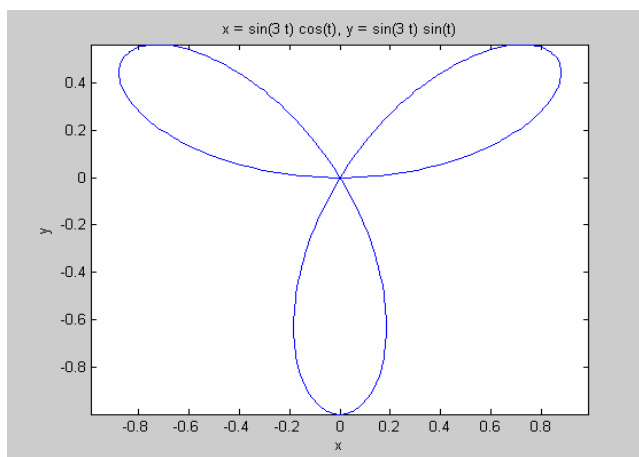


图 3-7 参数函数  $x = \sin(3t) \cos t$ ,  $y = \sin(3t) \sin t$  在  $[0, \pi]$  上的图形

[例 8] 绘制参数函数  $x = (1 - \sqrt{t})(3 + \cos t) \cos(4\pi t)$ ,  $y = (1 - t)(3 + \cos t) \sin(4\pi t)$  在  $[0, \pi]$  上的图形。

解 输入命令: `>>ezplot('(1-t^(1/2))*(3+cos(t))*cos(4*pi*t)','(1-t)*(3+cos(t))*sin(4*pi*t)')`

运行后画出参数函数的图形, 见图 3-8。如果  $t$  的指数, 将会得到很多有趣的图形。

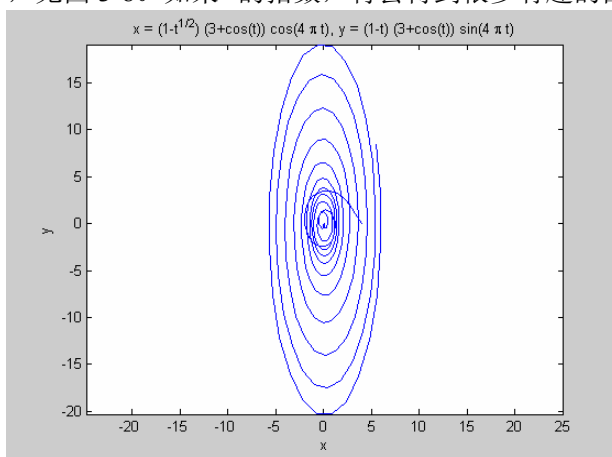


图 3-8 参数函数  $x = (1 - \sqrt{t})(3 + \cos t) \cos(4\pi t)$ ,  $y = (1 - t)(3 + \cos t) \sin(4\pi t)$  的图形

## 3.3 函数 ezpolar

ezpolar 的主要功能是用于绘制极坐标函数  $\rho = \varphi(\theta)$  的平面曲线, 它的调用格式有两种, 下面分别给以介绍。

3.3.1 绘制极坐标函数  $\rho = \varphi(\theta)$  在  $0 < \theta < 2\pi$  上的平面曲线

调用格式 `rho = ezpolar('phi(theta)')`

[例 9] 绘制极坐标函数  $\rho = \sin(\tan t)$  在  $[0, 2\pi]$  上的图形。

解 输入命令:

```
>> ezpolar('sin(tan(t))')
```

运行后画出极坐标函数  $\rho = \sin(\tan t)$  在  $[0, 2\pi]$  上的图形，见图 3-9。

#### 4.3.2 绘制极坐标函数 $\rho = \varphi(\theta)$ 在 $a < \theta < b$ 上的平面曲线

调用格式 `ezpolar (f,[a,b])`

**[例 10]** 绘制极坐标函数  $\rho = 1 - 2 \sin(at)$ ,  $a = 5$  在  $[0, 4\pi]$  上的图形。

**解** 输入命令: `>>ezpolar('1 - 2*sin(5*t)',[0,4*pi])`

运行后画出极坐标函数  $\rho = 1 - 2 \sin(5t)$  在  $[0, 4\pi]$  上的图形，见图 3-10。如果改变参数  $a$ ，将会得到很多有趣的图形，例如， $a = 15$ ，画出图 3-11。

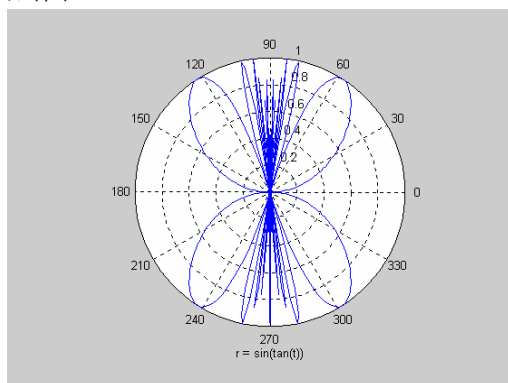


图 3-9 极坐标函数  $\rho = \sin(\tan t)$  在  $[0, 2\pi]$  上的图形

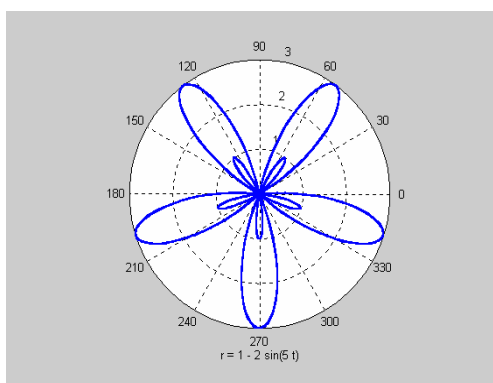


图 3-10  $\rho = 1 - 2 \sin(5t)$  的图形

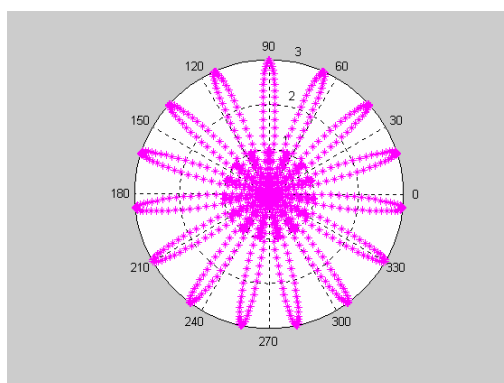


图 3-11  $\rho = 1 - 2 \sin(15t)$  的图形

### 3.4 函数 polar

polar 的主要功能是用于绘制极坐标函数  $\rho = \varphi(\theta)$  的平面曲线,它的调用格式如下:

`polar(THETA,RHO,S)`

其中 THETA 是极角 (弧度值), RHO 是极径, S 是可选项, S 的内容和用法与前面 plot 函数的全相同。

**[例 11]** 绘制极坐标枫叶函数  $\rho = \frac{\sin(at)}{t}$  在  $[-6.5\pi, 6.5\pi]$  上的图形,  $a = 1$ 。

**解** 输入命令:

```
>> t=-6.5*pi:pi/50:6.5*pi;
r= sin(t)./t;
polar(t,r,'r-')
```

运行后画出极坐标函数  $\rho = \frac{\sin(at)}{t}$  在  $[-6.5\pi, 6.5\pi]$  上的图形，见图 3-12。如果改变参数  $a$ ，将会得到很多有趣的图形。

**[例 12]** 绘制极坐标函数  $\rho = \frac{100 [2 - \sin(at) - 0.5 \cos(bt)]}{100 + (t - 0.5\pi)^8}$  在  $[-\frac{\pi}{2}, \frac{3\pi}{2}]$  上的图形,  $a = 7, b = 30$ 。

**解** 输入命令:

```
>> t=-0.5*pi:pi/500:3.5*pi;
r = 100*(2-sin(7*t)-1/2*cos(30*t))./(100+(t-1/2*pi).^8);
```

polar(t,r,'gp')

运行后画出图 3-13。如果改变参数  $a, b$ ，将会得到很多有趣的图形。

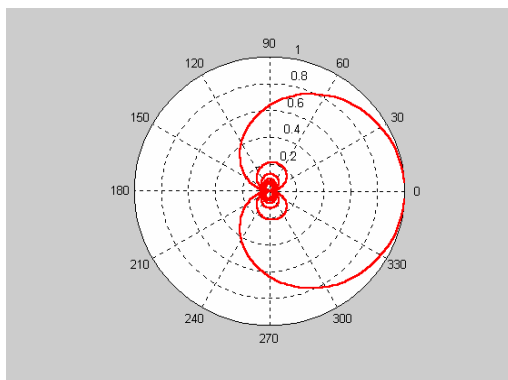


图 3-12  $\rho = \frac{\sin(at)}{t}$  的图形

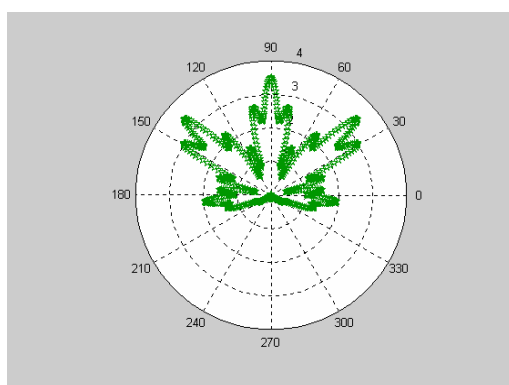


图 3-13  $\rho = \frac{100 [2 - \sin(at) - 0.5 \cos(bt)]}{100 + (t - 0.5\pi)^8}$  的图形

### 3.5 函数 ezcontour

ezcontour 的主要功能是用于绘制  $z = f(x, y)$  在  $a < x < b, c < y < d$  的等高线,它的调用格式如下:

ezcontour(f,[a,b,c,d],N)

其中 f 是二元函数  $z = f(x, y)$ 。

**[例 13]** 绘制函数  $u = -\frac{3z}{1+t^2-z^2}$  在  $-4 \leq x, y, z \leq 4$  的等高线。

**解** 输入命令:

```
>> ezcontour(' -3*z/(1+t^2-z^2)',[-4,4],420)
```

运行后画出这个函数在  $-4 \leq x, y, z \leq 4$  上的等高线图形, 见图 3-14。

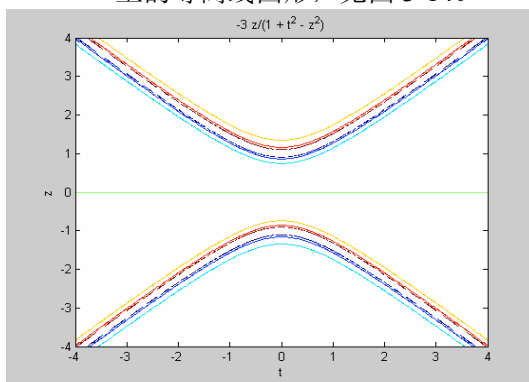


图 3-14 函数  $u = -\frac{3z}{1+t^2-z^2}$  在  $-4 \leq x, y, z \leq 4$  的等高线



### 3.6 函数 fplot 和其它画图函数

fplot 是 MATLAB 中常用的画平面曲线的函数，它的调用格式如下：

fplot('fun',[xmin xmax ymin ymax])

在[xmin xmax]内画出以字符串 fun 表示的函数的图形，[ymin ymax]给出了 y 的限制。还有一些画 2 维图形的函数，如

semilogx(x,y) 半对数坐标，x 轴为常用对数坐标

semilogy(x,y) 半对数坐标，y 轴为常用对数坐标

loglog(x,y) 全对数坐标

其它一些画特殊 2 维图形（如条形图等）的函数请查阅在线帮助系统。

用下面的例题是用函数 fplot 作图，可以看出它的方便之处。

[例 14] 绘制函数  $y = \frac{\sin x}{x}$  在  $-50 \leq x \leq 50$ ， $-0.3 \leq y \leq 1.1$  的曲线。

解 输入命令：

```
>> fplot('sin(x)./x',[-50 50 -0.3 1.1]),  
      gtext('sinx/x')
```

运行后屏幕显示图 3-15。

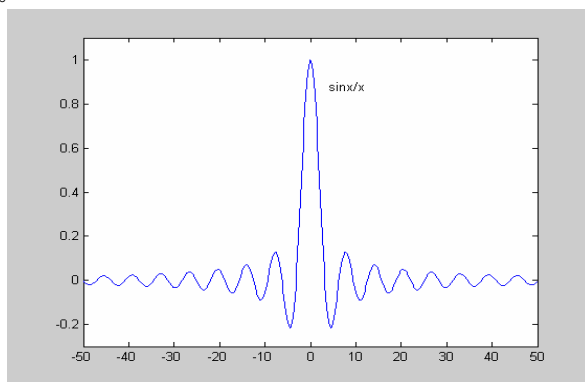


图 3-15 函数  $y = \frac{\sin x}{x}$  的图形

[例 15] 绘制足球图形。

解 在工作空间中输入下程序：

```
>> [B,V] = bucky;  
H = sparse(60,60);  
k = 31:60;  
H(k,k) = B(k,k);  
gplot(B-H,V,'b-');  
hold on  
gplot(H,V,'r-');  
for j = 31:60  
    text(V(j,1),V(j,2),int2str(j),'FontSize',10,'HorizontalAlignment','center');  
end  
hold off  
axis off equal
```

运行后屏幕显示图 3-16。

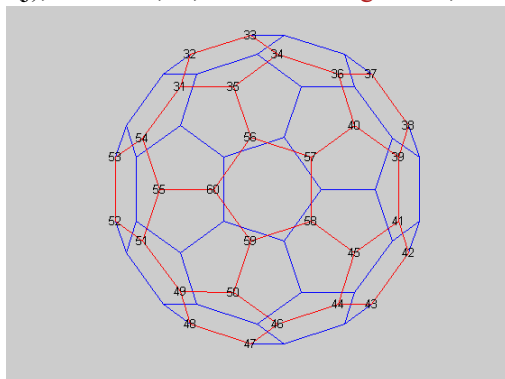


图 3-16 例 15 绘制足球图形

### 3.7 对曲线的进一步处理

在前面所介绍的绘制图形的方法是最基本的方法。为了使图形更加丰富，功能更加完善，满足用户的需要，还需要对图形作进一步的处理。比如给曲线的线型，颜色，给图形加标注，对坐标轴和边框作某些变动等。

#### 3.7.1 线型和颜色

在例 14 和例 15 中，我们对 `polar(t,r,'gp')` 中加入了 `g` 和 `p`，其中 `g` 表示绿色，`p` 表示线型是五角型，这样更能体现枫叶函数代表绿色的枫叶。**线型和颜色**

MATLAB 对曲线的线型和颜色有许多选择，标注的方法是在每一对数组后加一个字符串参数，关于平面曲线的颜色和线型可选项的内容请参看表 3-2。

表 3-2 平面曲线的颜色和线型

线 型	线方式	- 实线	: 点线	-· 点画线	-- 虚线	空出 小黑点
	点方式	· 圆点	+ 加号	* 星号	X 斜叉线	O 小圆
		^ 上尖线	∨ 字母 V	< 左尖线	> 右尖线	
		d 菱形	p 五角形	h 六角形	S 正方形	
颜 色		r 红	g 绿	b 蓝	w 白	k 黑
		c 青	y 黄	m 洋红		

**[例 16]** 作参数函数  $x=8\sin(2x)$ ,  $y=8\cos(3x)$  和  $x=4\sin(x)$ ,  $y=4\cos(x)$  在区间  $[-2.6\pi, 2.6\pi]$  上的图形。

**解** 输入程序:

```
>> t=-2.6*pi:pi/90:2.6*pi;  
x1=8*sin(2*t);y1=8*cos(3*t);  
x2=4*sin(t);y2=4*cos(t);  
plot(x1,y1,'g-',x2,y2,'mo')
```

运行后屏幕显示图 3-17。请读者将 `plot` 的内容写成 `plot(x1,y1,'rp',x2,y2,'bd')` 运行后观察结果有何变化。

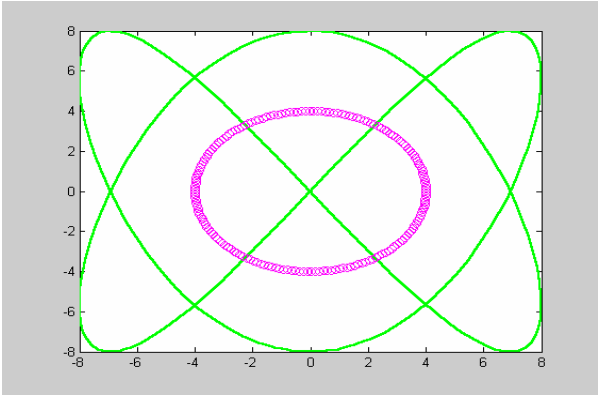


图 3-17 参数函数  $x=8\sin(2x)$ ,  $y=8\cos(3x)$  和  $x=4\sin(x)$ ,  $y=4\cos(x)$  的图形

#### 3.7.2 网格和标记

在一个图形上可以加网格、标题、X 轴标记、Y 轴标记，有两种常用的方法：法一是用手工完成这些工作，法二是用 MATLAB 函数完成。

(一)、用手工在图形上加数轴、标题、标记

**[例 17]** 作函数  $y_1=x\sin(x/2)$  和  $y_2=4.5-x^2/5$  的图形，并给标题、X 轴、Y 轴和交点加标记。

**解** 输入程序:

```
>> x=-3*pi:pi/30:3*pi;  
y1=x.*sin(x./2);  
y2=4.5-x.^2/5;  
plot(x,y1,'b',x,y2,'r-')
```

运行后屏幕显示图 4-18。然后，用图形窗口的工具条或菜单可以完成其余的工作。

(二)、用 MATLAB 函数在图形上加数轴、标题、标记

在例 17 中，我们对图形添加了标记，使得曲线代表什么函数和图形代表什么意思一目了然。软件 MATLAB 对在曲线的图形上加数轴、标题、标记有许多选择函数，标注的方法是在每一对数组后加一个字符串，有关详细的内容请参看表 4-3。用下列函数完成这些工作。

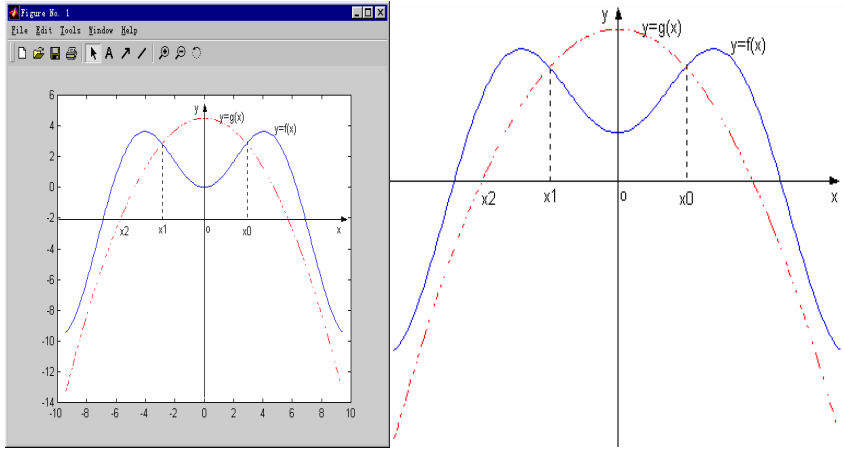


图 3-18 函数  $y_1=x*\sin(x/2)$ 和  $y_2=4.5-x^2/5$  的图形

表 3-3 在图形上加数轴、标题、标记的 MATLAB 函数

加标注的 MATLAB 函数	功 能
grid on	在图形窗口中自动画出网格虚线（包括极坐标网格线）
grid off	在当前图形窗口中自动去掉网格虚线
text( <i>a</i> , <i>b</i> , '字符串')	在图形窗口的点 ( <i>a</i> , <i>b</i> ) 处加上一个字符串作为图形的说明
gtext('字符串')	用鼠标在图形的任何位置加字符串作为图形的说明，即在窗口十字线的交点是字符串的位置，用鼠标点一下就将字符串放在那里。
xlabel('字符串')	在 <i>x</i> 轴下方中部位置添加字符串作为标注
ylabel('字符串')	在 <i>y</i> 轴左边中部位置添加字符串作为标注
title('字符串')	在图形窗口顶端的位置加上一个字符串作为标题
legend('字符串')	将字符串写在图形窗口某个位置的标签里，作为图形的说明

[例 18] 作函数  $y=\sin(x)$ 和  $z=\cos(x)$ 的图形，并给标题、X 轴、Y 轴和交点加标记。

解 输入程序：

```
>> x=linspace(0,4*pi,50);
y=sin(x);z=cos(x);
plot(x,y,'g-',x,z,'r-.')
grid
xlabel('自变量 X')
ylabel('因变量 Y 和 Z')
title('正弦函数 sinx 和余弦函数 cosx 的曲线')
legend('正弦函数 sinx','余弦函数 cosx')
gtext('y=sinx'),gtext('z=cosx')
```

运行后屏幕显示图 4-19。

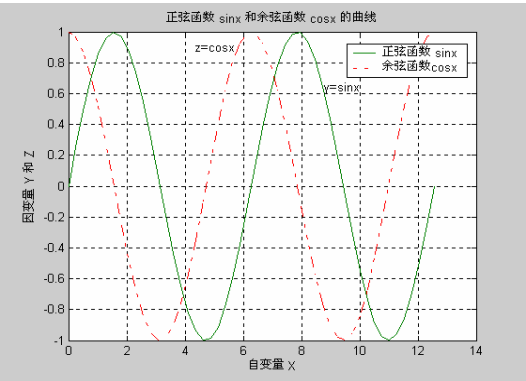


图 3-19 函数  $y=\sin(x)$ 和  $z=\cos(x)$ 的图形

可以在图形的任何位置加上一个字符串，如用  
text(2.7,1.7,'sinx')

表示在坐标  $x=2.5, y=0.7$  处加上字符串 `sinx`。

更方便的是用鼠标来确定字符串的位置，方法是输入函数：

`gtext('sinx'), gtext('cosx')`

在图形窗口十字线的交点是字符串的位置，用鼠标点一下就可以将字符串放在那里，如图 3-19。

### 3.7.3 坐标轴与边框的控制

在缺省情况下 MATLAB 自动选择图形的横、纵坐标的比例，如果你对这个比例不满意，可以用 MATLAB 函数控制坐标轴和边框，常用的函数如表 3-4：

表 3-4 控制坐标轴和边框的 MATLAB 函数

控制坐标轴和边框的 MATLAB 函数	功 能
<code>axis([xmin xmax ymin ymax])</code>	<code>[]</code> 中分别给出 $x$ 轴和 $y$ 轴的最小、最大值
<code>axis equal</code> 或 <code>axis('equal')</code>	$x$ 轴和 $y$ 轴的单位长度相同
<code>axis square</code> 或 <code>axis('square')</code>	图框呈方形
<code>axis off</code> 或 <code>axis('off')</code>	清除坐标刻度（含直角坐标和极坐标）
<code>axis on</code> 或 <code>axis('on')</code>	显示坐标刻度（含直角坐标和极坐标）
<code>axis manual</code> 或 <code>axis('manual')</code>	固定坐标轴刻度。如果当前图形窗口 <code>hold</code> 打开状态，则后面的图形将采取同样的刻度。
<code>box on</code>	给图形加上边框
<code>box off</code>	去掉图形边框

还有 `axis auto`，`axis image`，`axis xy`，`axis ij`，`axis normal`，`axis(axis)`。用法可参考在线帮助系统。请读者运行下面的程序，并观察结果。

`x=0:pi/20:2*pi; plot(sin(x), cos(x)), axis equal, axis([-1 1 0 1])` ✓

## 3.8 画多重线的方法

我们用迭代法求解非线性方程组的解时，需要寻找解的初始值，这就需要在同一个坐标系中画出每个方程对应的图形，找这些图形的交点（线）。在同一个坐标平面中画出许多条曲线的常用方法有三类。第一类方法是利用循环 `for` 语句作多条曲线；第二类方法是用 `hold on` 和 `hold off` 函数在原有的图形上增加曲线；第三类方法是在 `plot` 函数中，填写几条曲线。我们可以用这些方法在同一个直角坐标系或极坐标系中，画出多条隐函数的图形，显函数的图形，参数函数的图形或极坐标的函数的图形等等。下面分别给以介绍。

### 3.8.1 利用循环 for 语句作多条曲线

循环语句 `for` 的一边办形式为

`for` <循环参数>=<初值>:<步长>:<终值>  
<语句>

`end`

步长为 1 时可以省略。对于每一个参数，语句都重复执行。当作多重循环时，循环语句 `for` 可以嵌套使用。用此方法作概率与数理统计中的散点图和拟合曲线的效果特别的好。下面的例题展示了如何用循环语句 `for` 作多条曲线和点。

**[例 19]** 如果我们要画出图 3-20 中《项链》的图形，只需在 MATLAB 工作窗口中输入下面的循环语句 `for` 的程序：

```
>> t = 0:0.02:2*pi;
y = zeros(10,length(t));
x = zeros(size(t));
for k=1:2:19
    x = x + cos(2.*k*t)/k;
    y((k+1)/2,:) = x;
end
plot(y(1:2:9,:))
运行后可绘制出图 4-20。
```

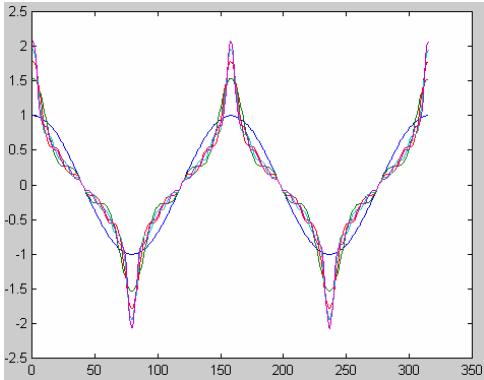


图 3-20 《项链》的图形

### 3.8.2 利用 hold on 和 hold off 画多重曲线的三种方法

在同一个坐标系中画多条曲线的另一种方法是利用 hold on 和 hold off 函数在已经画好的图形上再增加曲线。如果设置 hold on，则 MATLAB 软件将把新的 plot 或者 ezplot 函数产生的图形画在原来的图形上。而函数 hold off 结束这个过程。下面通过例题分别说明这些方法的具体应用。

#### (一)、利用 hold on, ezplot 和 hold off 函数在原有的图形上增加曲线

这种方法适用于在同一个坐标系中画多条隐函数  $F_1(x, y) = 0$ ,  $F_2(x, y) = 0, \dots$ ,  $F_n(x, y) = 0$  的曲线。具体用法如下面的例题。

**[例 20]** 在同一个坐标系中画出双曲线  $x^2 - y^2 = -1$  和椭圆  $8x^2 + 4y^2 = 16$  的图形。

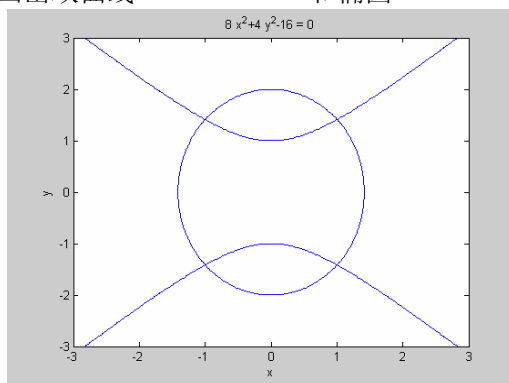


图 3-21 双曲线  $x^2 - y^2 = -1$  和椭圆  $8x^2 + 4y^2 = 16$  的图形

解 输入程序：

```
>> syms x y
F1=x^2-y^2+1;F2=8*x^2+4*y^2-16;
ezplot(F1, [-3, 3]),
hold on
ezplot(F2, [-3,3])
hold off
```

运行后屏幕显示图 3-21。

#### (二)、利用 hold on, plot 和 hold off 函数在原有的图形上增加曲线

这种方法适用于在同一个坐标系中画多条显函数  $F_1(x, y) = 0$ ,  $F_2(x, y) = 0, \dots$ ,  $F_n(x, y) = 0$  的曲线。具体用法如下面的例题。

**[例 21]** 给出一组数据点  $(x_i, y_i)$  列入表 3-5 中，试在同一个坐标系中画出数据点和拟合曲线  $f(x) = 5.0911x^3 - 14.1905x^2 + 6.4102x - 8.2574$  及其  $g(x) = x^4 - 14x^2 - 180$  的图形。

表 3-5 例 21 的一组数据点

$x_i$	-2.5	-1.7	-1.1	-0.8	0	0.1	1.5	2.7	3.6
$y_i$	-192.9	-85.50	-36.15	-26.52	-9.10	-8.43	-13.12	6.50	68.04

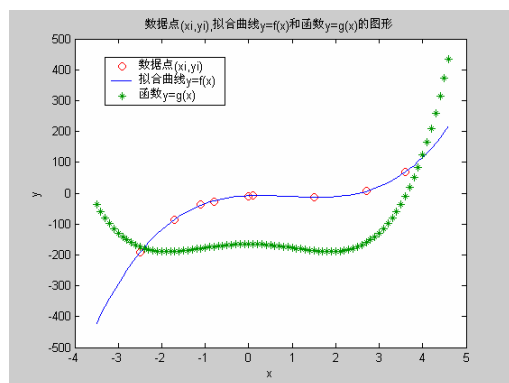


图 3-22 例 21 的数据散点图和拟合曲线的图形

解 输入程序:

```
>> xi=[-2.5 -1.7 -1.1 -0.8 0 0.1 1.5 2.7 3.6];  
y=[-192.9 -85.50 -36.15 -26.52 -9.10 -8.43 -13.12 6.50 68.04];  
x=-3.5:0.1:4.6;  
F=5.0911.*x.^3-14.1905.*x.^2+6.4102.*x-8.2574;  
G=2.*x.^4-14.*x.^2-165;  
plot(xi,y,'ro'),  
hold on  
plot(x,F,'b-')  
hold off  
hold on  
plot(x,G,'g*')  
hold off  
legend('数据点(xi,yi)', '拟合曲线 y=f(x)', '函数 y=g(x)')  
xlabel('x'), ylabel('y'),  
title('数据点(xi,yi),拟合曲线 y=f(x)和函数 y=g(x)的图形')
```

运行后屏幕显示数据点(xi,yi),拟合曲线 y=f(x)和函数 y=g(x)的图形, 见图 3-22。

最后, 如果你在一段程序中画了几个图形, 需要逐个观察, 那么应该在每两个 plot 函数之间加一个 pause 函数, 它暂停函数的执行, 直到你击下任何一个键。

### (三)、利用 hold on, polar 和 hold off 函数在原有的图形上增加曲线

这种方法适用于在同一个极坐标系中画多条极坐标函数  $\rho_1 = \rho_1(\theta)$ ,  $\rho_2 = \rho_2(\theta), \dots, \rho_n = \rho_n(\theta)$  的曲线。具体用法如下面的例题。

**[例 22]** 在同一个极坐标系中, 绘制极坐标函数  $\rho = \frac{100[2 - \sin(at) - 0.5 \cos(bt)]}{100 + (t - 0.5\pi)^8}$  和  $\rho = \frac{\sin(ct - d)}{t}$  在  $[-6.5\pi, 6.5\pi]$  上的图形,  $a = 7, b = 30, c = 4, d = 1.5$ 。

解 输入命令:

```
>> t=-6.5*pi:pi/500:6.5*pi;  
r = 100*(2-sin(7*t)-1/2*cos(30*t))./(100+(t-1/2*pi).^8);  
polar(t,r,'gp')  
hold on  
r1=sin(4*t-1.5)./t;  
polar(t,r1,'r-')  
hold off  
title('绿枫叶的粉领结')
```

运行后画出图 3-23。如果改变参数  $a, b, c, d$ , 将会得到很多有趣的图形。

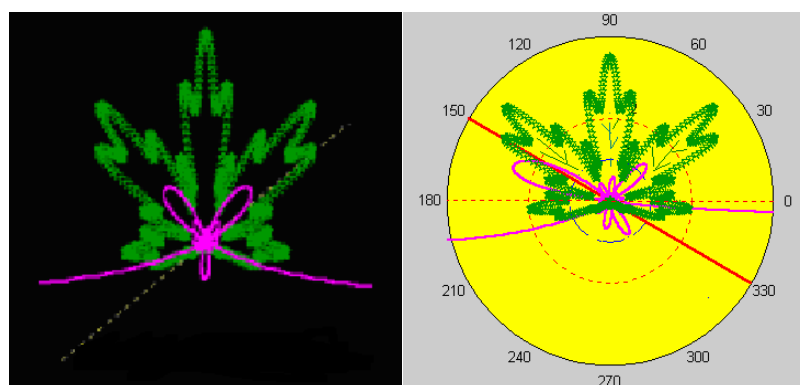


图 3-23  $\rho = \frac{100[2 - \sin(at) - 0.5 \cos(bt)]}{100 + (t - 0.5\pi)^8}$  和  $\rho = \frac{\sin(ct - d)}{t}$  的图形

以此类推, hold on 和 hold off 函数还可以与其它画图函数配合, 在原有的图形上增加曲线。读者不妨试试。

### 3.8.3 利用几个数组作多条曲线画多重曲线

在绘制平面曲线的 MATLAB 函数 plot, ezplot, polar, ezpolar, ezcontour 等中, 输入几个数组可以作

多条曲线，如下面分别给出的例题所示。

**[例 23]** 在同一个坐标系中，绘制函数  $y_1 = \frac{1}{1+x^2}$ ,  $y_2 = 3(1-x)^2 e^{-x^2}$ ,  $y_3 = -10(\frac{x}{5} - x^2) e^{-x^2}$ ,  $y_4 = -\frac{1}{3} e^{-(x+1)^2}$  在  $-1.5\pi \leq x \leq 1.5\pi$  的曲线。

**解** 输入命令：

```
>> x=-1.5*pi:0.01:1.5*pi;
y1=1./(1+x.^2);
y2=3*(1-x).^2.*exp(-(x.^2));
y3=-10*(x/5 - x.^2).*exp(-x.^2);
y4=-1/3*exp(-(x+1).^2);
plot(x,y1,'r-',x,y2,'b--',x,y3,'g:',x,y4,'m-')
```

运行后屏幕显示图 3-24。

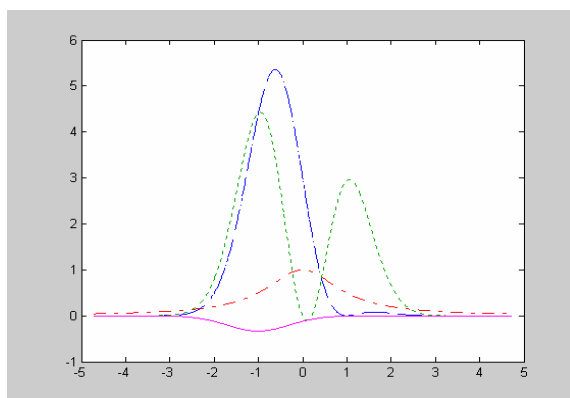


图 3-24  $y_1 = \frac{1}{1+x^2}$ ,  $y_2 = 3(1-x)^2 e^{-x^2}$ ,  $y_3 = -10(\frac{x}{5} - x^2) e^{-x^2}$ ,  $y_4 = -\frac{1}{3} e^{-(x+1)^2}$  的图形

**[例 24]** 在同一个坐标平面中，绘制三个函数  $z_1 = 3(1-x)^2 e^{-x^2-y^2}$ ,  $z_2 = -10(\frac{x}{5} - x^2 - y^5) e^{-x^2-y^2}$ ,  $z_3 = -\frac{1}{3} e^{-(x+1)^2-y^2}$  在  $-\pi \leq x, y, z \leq \pi$  的等高线。

**解** 输入命令：

```
>> f=['3*(1-x)^2*exp(-(x^2) - (y+1)^2)'; '-10*(x/5 - x^3 - y^5)*exp(-x^2-y^2)'; '-1/3*exp(-(x+1)^2 - y^2)'];
ezcontour(f,[-pi,pi])
```

运行后画出这三个函数在  $-\pi \leq x, y, z \leq \pi$  上的等高线图形，见图 3-25。

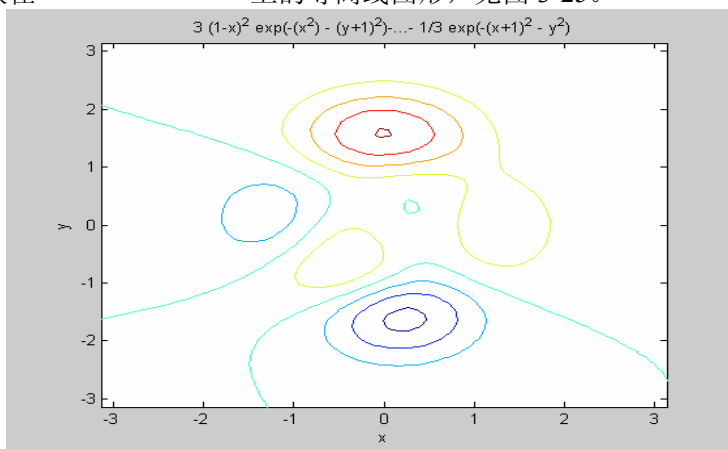


图 3-25 三个函数在  $-\pi \leq x, y, z \leq \pi$  上的等高线图形

### 3.9 图形窗口的分割

有时需要在同一个图形窗口中绘制多个图形，以便于对比和观察。这是可以用 MATLAB 函数 `subplot` 将原窗口分割，划为多个子窗口来实现。它的调用格式如下：

**subplot(m,n,p)**

其功能是把一个图形窗口分成  $m \times n$  个图形区域，并按行从左到右，按列从上到下的顺序进行编号， $p$  代表当前的区域号，在每个区域中分别画一个图，如下面例题所示。

**[例 25]** 将一个图形窗口分成四个图形区域，在每个子区域中分别画  $y=\sin(x)$ ， $z=\cos(x)$ ， $u=\sin(1/x)$  和  $v=\sin(x)/\cos(x)$  的图形，并比较差异。

**解** 输入程序：

```
>> x=linspace(0,4*pi,60);
y=sin(x);z=cos(x);
v=sin(x)/cos(x);
subplot(2,2,1),
plot(x,y),axis([0 4*pi -1 1]),
title('sin(x)')
subplot(2,2,2),
plot(x,z),axis([0 4*pi -1 1]),
title('cos(x)')
subplot(2,2,3),
fplot('sin(1./ x)', [0.01 0.1],1e-3)
title(' sin(1 / x)')
subplot(2,2,4),
plot(x,v),axis([0 4*pi -20 20]),
title('sin(x)/cos(x)')
```

运行后得到  $2 \times 2$  共 4 幅图形，见图 3-26。请读者比较差异。

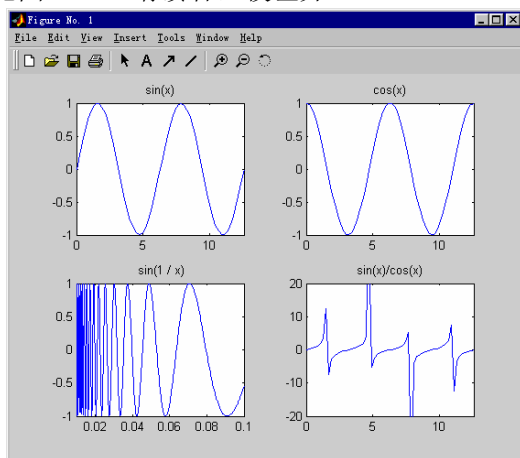


图 3-26

**[例 26]** 将一个图形窗口分成三个图形区域，在每个子区域中分别画@humps（可以用@指定 MATLAB 库函数）， $z = \text{abs}(\exp(-j*x*(0:9))*\text{ones}(10,1))$ ， $u = \sin(1/x)$  和函数组  $[\tan(x), \sin(x), \cos(x)]$  的图形，并比较差异。

**解** 输入程序：

```
>> subplot(2,2,1),
fplot(@humps,[0 1]) % FUN can be specified using @
title(' @humps ')
f = inline('abs(exp(-j*x*(0:9))*ones(10,1))');% an inline object, or an expression
subplot(2,2,2),
fplot(f,[0 2*pi])
title(' abs(exp(-j*x*(0:9))*ones(10,1))')
subplot(2,1,2),
fplot('[\tan(x),sin(x),cos(x)]',2*pi*[-1 1 -1 1])
title(' \tan(x),sin(x),cos(x)')
```



运行后得到共 3 幅图形，见图 3-27。请读者比较差异。

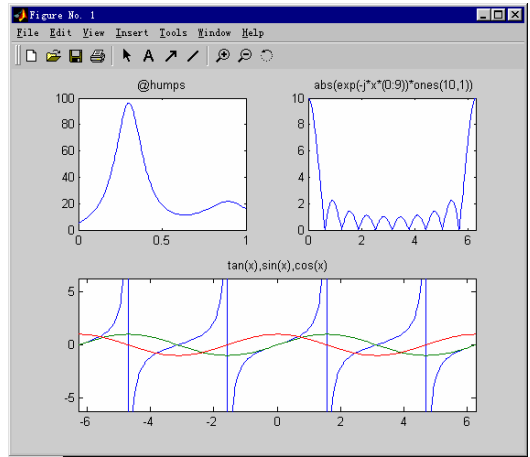


图 3-27

### 3.10 空间图形制作

三重积分和曲面积分的积分区域是空间图形。MATLAB 软件为我们提供了众多的功能强大的绘制空间图形的函数，它们分别是 plot3, mesh, meshc, meshz, meshgrid, sphere, cylinder, hesurf 等等。

#### 3.10.1 函数 plot3

plot3 是 MATLAB 中最常用的画空间曲线的函数，它的主要功能是用于绘制显式函数  $z = f(x, y)$  和参数式函数  $x = x(t)$ ,  $y = y(t)$ ,  $z = z(t)$  的空间曲线，也可以用空间中一组平行平面上的截线的方式表示曲面。Plot3 函数的调用格式如下：

**plot3(x,y,z,'可选项 s')**

其中 x,y,z 分别是曲线上的横坐标、纵坐标和竖坐标，曲线上的'可选项 s'中通常包含确定曲线颜色、线型、两坐标轴上的比例等等的参数。用户在作图时可以根据需要选择可选项。如果用户在绘图时不用可选项，那么 Plot3 函数将制动选择一组默认值，画出空间曲线。下面我们将通过例题逐步介绍。

**[例 27]** 作出参数函数  $x=x$ ,  $y=\sin(x)$ ,  $z=\cos(x)$  在区间  $[0, 12\pi]$  上的图形。

**解** 输入下列程序：

```
>> x=linspace(0,12*pi,5000);% 在[0,12π]上取 5000 个点
y=sin(x);z=cos(x);
plot3(x,y,z)✓
```

运行后屏幕显示函数  $x=x$ ,  $y=\sin(x)$ ,  $z=\cos(x)$  在区间  $[0, 12\pi]$  上 5000 个点连成的光滑的图形（见图 3-28）。

**[例 28]** 作出函数  $z = axe^{-b(x^2+y^2)}$  在矩形区域  $-c1 \leq x \leq c2$ ,  $-d1 \leq y \leq d2$  上的图形。其中  $a = b = 0.1$ ,  $c1 = c2 = 5$ ,  $d1 = d2 = 6$ 。

**解** 输入下列程序：

```
>> [x,y]=meshgrid(-5:0.1:5,-6:0.1:6);
z=0.1*x.*exp(-0.1*(x.^2+y.^2));
plot3(x,y,z)
```

运行后屏幕显示所求作的图形（见图 3-29）。

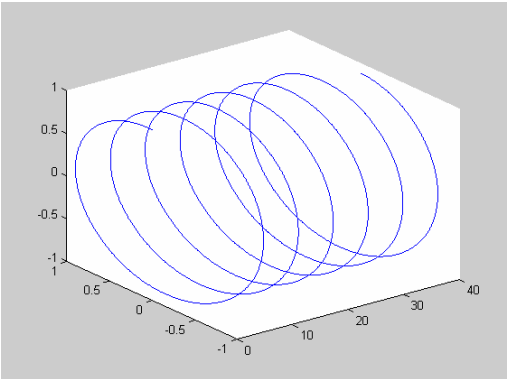


图 3-28  $x=x$ ,  $y=\sin(x)$ ,  $z=\cos(x)$  在  $[0, 12\pi]$  上的图形

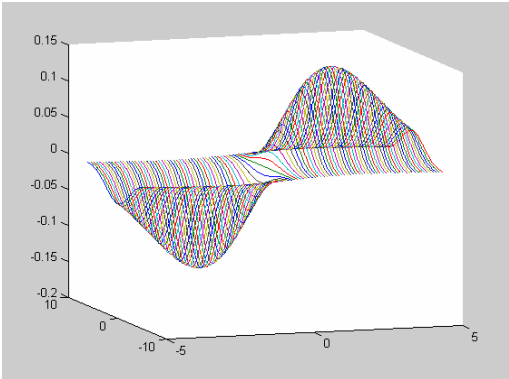


图 3-29  $z = 0.1xe^{-0.1(x^2+y^2)}$  在  $-5 \leq x \leq 5$ ,  $-6 \leq y \leq 6$  上的图形

读者可以改变此例中  $a, b, c1, c2, d1, d2$  的值，将会画出许多有趣的图形。

3.10.2 绘制曲面的网图函数

将 MATLAB 中绘制曲面的常用网图函数列入表 3-6。

表 3-6 MATLAB 中绘制曲面的常用网图函数

序号	MATLAB 函数名	功 能
(1)	mesh(x,y,z)	用空间中的两组相交的平行平面上的网状线的方式表示曲面。
(2)	meshc(x,y,z)	用（1）的方式表示曲面，并附带有等高线。
(3)	meshz(x,y,z)	屏蔽的网格图
(4)	surf(x,y,z)	用空间中网状线并网格中填充色彩的方式表示曲面。
(5)	surfc(x,y,z)	用（4）的方式表示曲面，并附带有等高线。
(6)	surfl(x,y,z)	用（4）的方式表示曲面，并附带有阴影。
(7)	hidden on	消除掉被遮住部分的网状线。
(8)	hidden off	将被遮住部分的网状线显示出来。

用表 3-6 中的 MATLAB 函数绘制曲面时，首先根据网格坐标命令

meshgrid

把节点坐标的常数向量  $x$  和  $y$  或  $z$  转化为矩阵  $X$  和  $Y$  或  $Z$ ，这些所有的绘制曲面方法都要求利用名为 meshgrid 的程序产生矩阵，然后选用对应的调用格式直接用表 3-6 中的 MATLAB 函数绘制曲面即可。 $X$  和  $Y$  或  $Z$  可以是不等距分布。所以，首先介绍 meshgrid 的 M-函数文件的功能、详细的调用方法。

常用的 MESHGRID 命令的调用格式有三种，分别介绍如下：

1、调用格式一： $[X,Y]=\text{meshgrid}(x,y)$

$[X,Y]=\text{MESHGRID}(x,y)$  将向量  $x$  和  $y$  转换成矩阵  $X$  和  $Y$ ，其中矩阵  $X$  的每行是向量  $x$ ，矩阵  $Y$  的每列是向量  $y$ ，此命令可以被用于计算二元函数或作三维曲面的图形。向量  $x$  和  $y$  的输入形式有两种：

2、调用格式二：

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

是  $[X,Y]=\text{MESHGRID}(x,x)$  的一种缩写式。

3、调用格式三：

$[X,Y,Z]=\text{MESHGRID}(x,y,z)$

此命令将向量  $x$ 、 $y$  和  $z$  转换成矩阵  $X$ 、 $Y$  和  $Z$ ，经常被用于计算三元函数插值或作三维立体的图形。

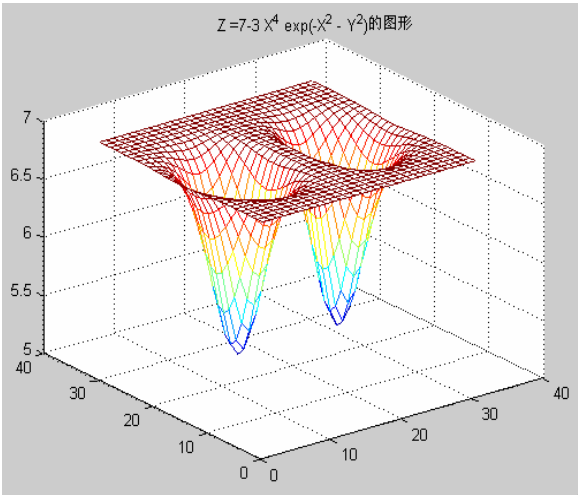


图 3-30 函数  $z = 7 - 3x^4 e^{-x^2 - y^2}$  的图形

[例 29] 已知  $x=-3:0.2:3$ ； $y=x$ ，计算函数

$z = 7 - 3x^4 e^{-x^2 - y^2}$  的值，并作出函数的图形。

解 输入程序：

```
>> [X,Y]=meshgrid(-3:.2:3,-3:.2:3);
Z=7-3*X.^4.*exp(-X.^2-Y.^2);
mesh(Z)
```

运行后输出函数值（略）和图形如图 3-30。

[例 30] 作出函数  $z = 2 + x e^{-x^2 - y^2}$  在区域  $-2 \leq x \leq 2, -2 \leq y \leq 2$  上的图形。

解 输入程序：

```
>> [X,Y]=meshgrid(-2:.2:2,-2:.2:2);
Z=2+X.*exp(-X.^2-Y.^2);
meshc(Z)
```

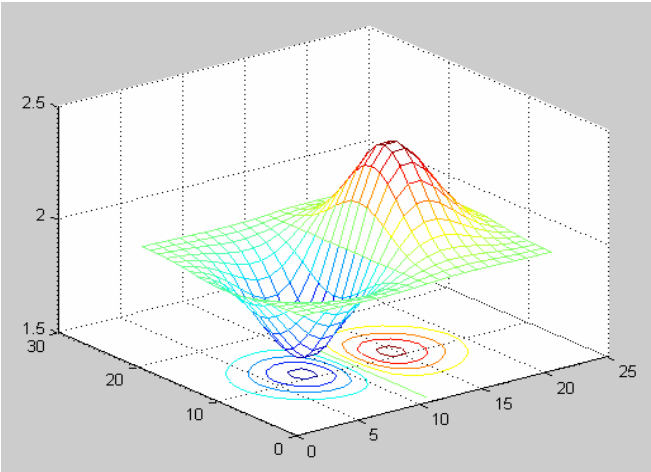


图 3-31 函数  $z = 2 + x e^{-x^2 - y^2}$  在区域  $-2 \leq x \leq 2, -2 \leq y \leq 2$  上的图形

运行后输出函数值（略）和图形如图 3-31。

**[例 31]** 设节点  $(x, y, z)$  中的  $x = -3:0.5:3$ ,  $y = x$  和函数  $Z = 7 - 3x^3 e^{-x^2-y^2}$  值, 作  $Z$  在插值点  $X = -3.9:0.5:5$ ,  $Y = -4.9:0.5:4.5$  处的拟合曲面  $z = 7 - 3x^3 e^{-x^2-y^2}$  和节点的图形。

**解** 输入程序:

```
>> x=rand(50,1);
y=rand(50,1); %生成 50 个 1 元均匀分布随机数 x 和 y, x, y。
X=-3+(3-(-3))*x;%利用 x 生成随机变量。
Y=-2.5+(3.5-(-3.5))*y;%利用 y 生成 上的随机变量。
Z=7-3* X.^3 .* exp(-X.^2 - Y.^2); %在每个随机点 (X,Y) 处计算 Z 的值。
X1=-3.2:0.1:3.2;
Y1=-2.9:0.1:3.9;
[X1,Y1] = meshgrid(X1,Y1); %将坐标 (X1,Y1) 网格化。
ZI=7-3* X1.^3 .* exp(-X1.^2 - Y1.^2);
mesh(X1,Y1, ZI) %作二元拟合图形。
xlabel('x'), ylabel('y'), zlabel('z'),
title('被拟合函数 z=7-3 x^3 exp(-x^2 - y^2) 的曲面和节点的图形')
%legend('被拟合函数曲面','节点(xi,yi,zi)')
hold on %在当前图形上添加新图形。
plot3(X,Y,Z, 'bo')
%用蓝色小圆圈画出每个节点(X,Y,Z)。
hold off %结束在当前图形上添加新图形。
```

运行后屏幕显示被拟合函数  $Z = 7 - 3x^3 e^{-x^2-y^2}$  的曲面和节点的图形（见图 3-32）。

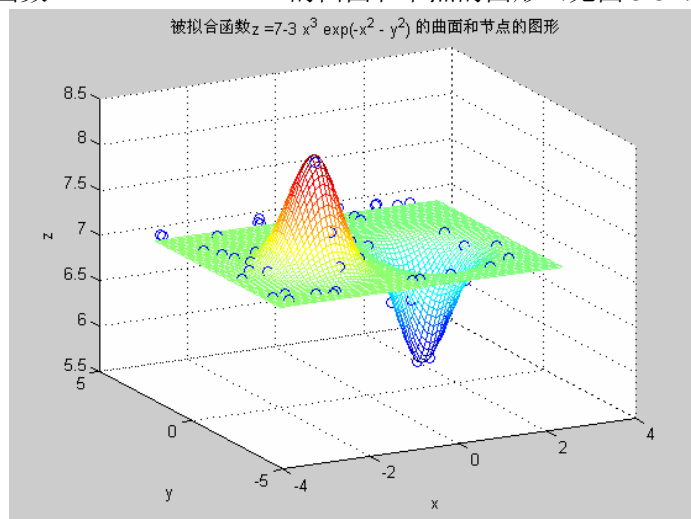


图 3-32 被拟合函数  $Z = 7 - 3x^3 e^{-x^2-y^2}$  的曲面和节点的图形

### 3.10.3 绘制旋转曲面和球面的函数

#### （一）、绘制球面的函数

sphere 是绘制球面的 MATLAB 函数, 其调用格式有三种:

**调用格式一:**  $[X,Y,Z] = \text{sphere}(N)$

此函数生成三个  $(N+1) \times (N+1)$  阶矩阵, 利用 SURF(X,Y,Z) 可以产生一个单位球面。

**调用格式二:**  $[X,Y,Z] = \text{sphere}$

此形式使用默认值  $N = 20$ 。

**调用格式三:**  $\text{sphere}(N)$

只绘制球面图, 不返回任何值。

**[例 32]** 绘制如图 3-33 所示的地球表面的气温分布示意图。

**解** 输入下列程序:

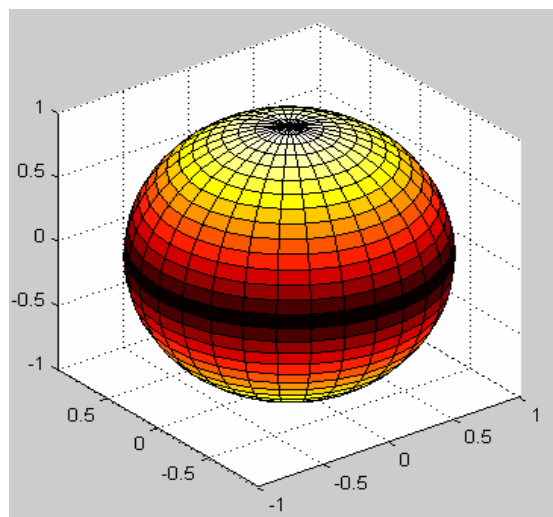


图 3-33 球面

```
>> [a,b,c] = sphere(30);
t=abs(c);
surf(a,b,c,t);
axis('equal'),axis('square'),%将坐标轴的刻度控制为相同
colormap('hot')
```

运行后输出球面如图 3-33。

## (二)、绘制旋转曲面的函数

cylinder 是绘制球面的 MATLAB 函数，其调用格式有两种：

**调用格式一：**  $[X,Y,Z]=\text{cylinder}(R,N)$

此函数以母线向量  $R$  生成单位柱面，母线向量  $R$  是在单位高度里等分刻度上定义的半径向量。 $N$  为旋转圆周上的分格线的条数。利用  $\text{surf}(X,Y,Z)$  可以产生一个此柱面。

**调用格式二：**  $[X,Y,Z]=\text{cylinder}(R)$ ，或  $[X,Y,Z]=\text{cylinder}$   
此形式使用默认值  $N=20$  和  $R=[1\ 1]$ 。

**[例 33]** 绘制由连续函数  $y = 6 + \sin x$ ， $x = 0$ ， $x = 6\pi$  和  $y = 0$  所围成的平面图形，绕  $x$  轴旋一周所得到的旋转曲面的图形。

**解** 输入下列程序：

```
>>x=0:pi/20:6*pi;
R=6+sin(x);
[a,b,c]=cylinder(R,20);
surf(a,b,c),
```

运行后输出旋转曲面的图形如图 3-34。

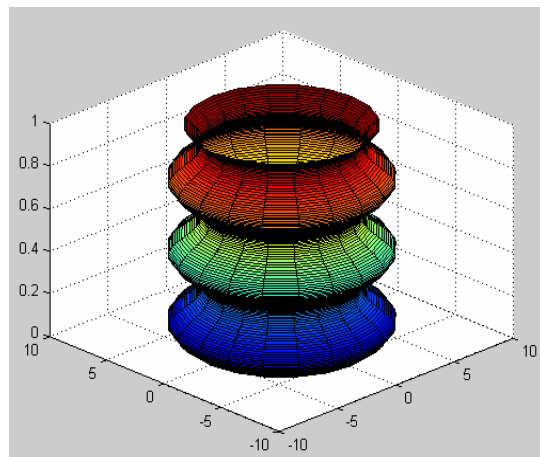


图 3-34 旋转曲面的图形

## 3.11 综合作图

我们可以用上面介绍的方法，作平面区域和空间区域的图形。下面通过具体例题介绍方法。

**[例 34]** 画出由曲线  $2xy=1$ ， $y=\sqrt{2x}$ ， $x=2.5$  所围成的平面区域  $D_{xy}$  的图形。

**解** 输入下列程序：

```
>>x=0.001:0.001:3;
y1=1./(2*x);y2=sqrt(2*x);
plot(x,y1,'b-',x,y2,'m-',2.5,y,'r-'),
axis([-0.5 3 -0.5 3])
```

$\text{title('由 } y1=1/2x, y2=\sqrt{2x} \text{ 和 } x=2.5 \text{ 所围成的积分区域 } D_{xy}')$

运行后屏幕显示图 3-35。

**[例 35]** 画出由曲线  $x=1$ ， $y=x$ ， $y=0$  所围成的平面区域  $D_{xy}$  的和其内部单元条的填充图。

**解** 输入下列程序：

```
>> clear,format compact
fill([0,1,1,0],[0,0,1,0],'y')% 画出区域 的图形
hold on
fill([0.55,0.6,0.6,0.55,0.55],[0,0,0.6,0.55,0],'r') %
画出单元条的图形
hold off
title('由 x=1,y=x 和 y=0 所围成的积分区域
```

$D_{xy}$ )

运行后屏幕显示图 3-36。

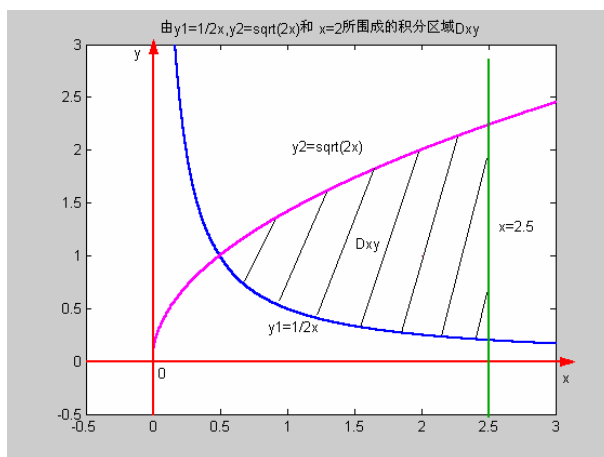


图 3-35 由  $y1=1/2x, y2=\sqrt{2x}$  和  $x=2.5$  所围成的积分区域  $D_{xy}$

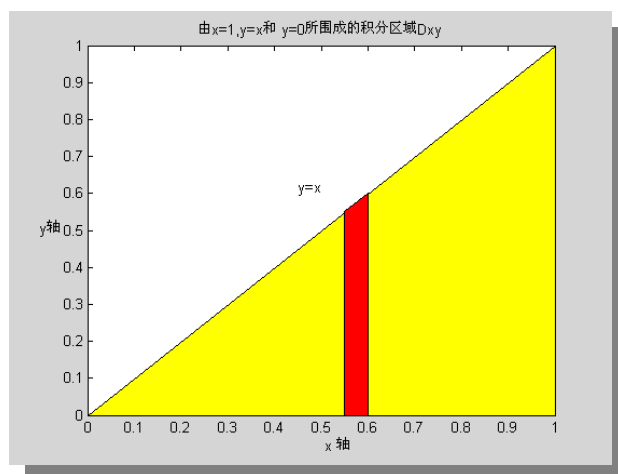


图 3-36 曲线  $x=1$ ,  $y=x$ ,  $y=0$  所围成的平面区域  $D_{xy}$  的图

**[例 36]** 画出由旋转抛物面  $z=8-x^2-y^2$ , 圆柱  $x^2+y^2=4$  和  $z=0$  所围成的空间闭区域及其在  $xOy$  面上的投影。

**解** (1) 画出积分区域  $V$  的草图。输入程序：

```
>> [x,y]=meshgrid(-2:0.01:2);
z1=8-x.^2-y.^2;
figure(1)
meshc(x,y,z1)
hold on
x=-2:0.01:2;
r=2;
[x,y,z]=cylinder(r,30)
mesh(x,y,z)
hold off
title('由旋转抛物面  $z=8-x^2-y^2$ , 圆柱面  $x^2+y^2=4$  和  $z=0$  所围成的积分区域  $V$ ')
figure(2)
contour(x,y,z,10)
title('由  $z=8-x^2-y^2$ , 圆柱面  $x^2+y^2=4$  和  $z=0$  所围成区域  $V$  在  $xOy$  面上的投影区域  $D_{xy}$ ')
```

运行后屏幕显示图 3-37。

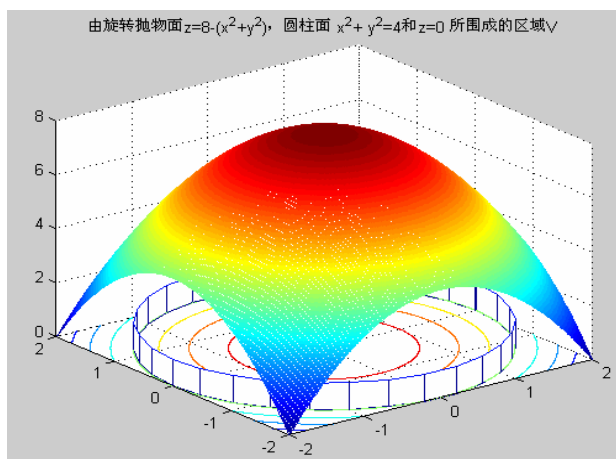


图 3-37 由  $z=8-x^2-y^2$ ,  $x^2+y^2=4$  和  $z=0$  围成闭区域  $V$  及其在  $xOy$  面上的投影区域  $D_{xy}$

**[例 37]** 画出马鞍面  $z=x^2-2y^2$  和平面  $z=2x-3y$  的交线。

**解** 输入程序：

```
>> [x,y]=meshgrid(-52:2:52);
z1=x.^2-2*y.^2;
z2=2*x-3*y;
```

```

mesh(x,y,z1)
hold on
mesh(x,y,z2)
hold off
title('由马鞍面  $z=x^2-2y^2$  和平面  $z=2x-3y$  的交线')

```

运行后屏幕显示图 3-38。

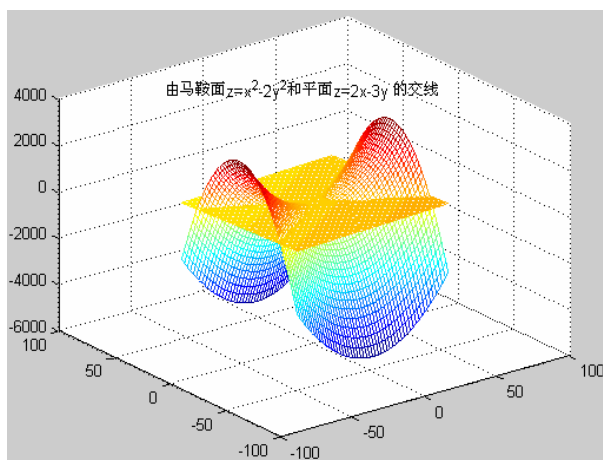


图 3-38 马鞍面  $z = x^2 - 2y^2$  和平面  $z = 2x - 3y$  的交线

[例 38] 画出由  $x = 1$ ,  $y = x$ ,  $z = xy$  及  $z = 0$  所围成的闭区域。

解 输入计算程序：

```

[x,y]=meshgrid(0:0.04:1); %确定计算和绘制的定义域网格
z1=x.*y;z2=zeros(size(z1)); %马鞍面  $z=xy$  和坐标面  $z=0$  的方程
mesh(x,y,z1); %画出马鞍面  $z=xy$ 
hold on
mesh(x,y,z2); %画出坐标面  $z=0$ 
x1=[0:0.02:1];y1=x1; %sx1=length(x1);平面  $y=x$  的方程
zd=[zeros(1,length(x1));x1.*y1];
% 平面  $y=x$  的上, 与  $z$  轴平行的线族的端点  $zd=xy$ 
line([x1;x1],[y1;y1],zd); % 画出此平行线族
plot3([x1;x1],[y1;y1],zd,'*'); %端点画 '*'
plot3(ones(2,length(x1)),[y1;y1],[zeros(1,length(x1));y1 ],'ro'); %画出平面  $x=1$  上的交线
xlabel('x'), ylabel('y'), zlabel('z'),
hold off
pause,rotate3d

```

运行后屏幕显示图 3-39。

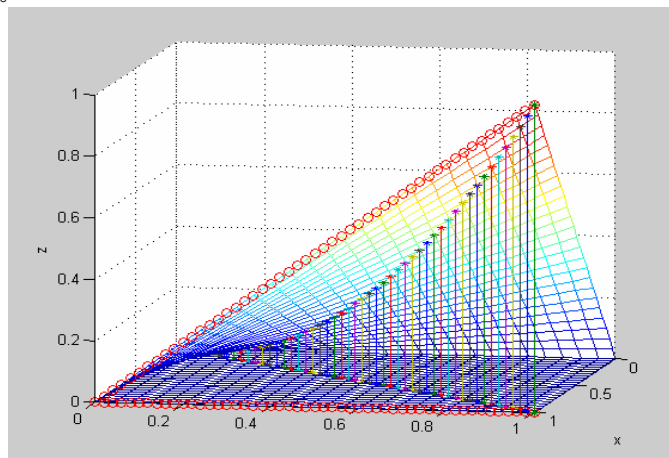


图 3-39 由  $x = 1$ ,  $y = x$ ,  $z = xy$  及  $z = 0$  所围成的闭区域

在 MATLAB 软件为我们提供了众多的功能强大的图形绘制函数，用于绘制平面曲线的 MATLAB 函数有 plot, ezplot, polar, ezpolar, fplot, ezcontour 等，绘制空间图形的函数等，下面分别给以介绍。



## 第四章 MATLAB 符号计算

MATLAB 系统本来只能做数值计算,并没有符号运算的功能,符号运算工具箱(Symbolic Math Toolbox)则扩充了 MATLAB 这方面的功能,它是由 Maple 软件的核心来完成的。

这个工具箱在 MATLAB 中的符号是 Toolbox/Symbolic,下面对它的常用功能作一简单介绍,详细用法请参阅帮助系统。

### 4.1 符号变量与符号表达式

符号运算是指运算的主要对象是符号、文字,或者是变量所进行的运算,自然是精确公式解中的运算。符号运算和符号表达式总称为符号对象。MATLAB 系统本来只能做数值计算,并没有符号运算的功能,符号运算工具箱(Symbolic Math Toolbox)则扩充了 MATLAB 这方面的功能,它是由 Maple 软件的核心来完成的。本节主要介绍与极限的符号运算相关的问题。

#### 4.1.1 符号变量和符号表达式的生成

符号运算工具箱处理的主要对象是符号和符号表达式,为此要使用一种新的数据类型——符号变量。在 MATLAB 工具箱用来生成符号对象的函数有 **sym** 和 **syms**,其中用 **sym** 来定义一个符号或符号表达式,而 **syms** 可定义多个符号。

##### (一)、符号变量的生成

要使符号变量有意义,必须首先生成符号变量,生成符号变量可用 **sym** 或 **syms**。

##### 1、用 sym 生成符号变量

用 **sym** 生成符号变量的常用形式为

**x=sym('x')** %符号变量的值为单引号内的字符或字符串

或 **x=sym('x','Variable Format')** %同上,但设定了变量 **x** 的类型

[例 1] 试将  $y = a + b\sqrt{x}$  中的系数  $a, b$ , 自变量  $x$  和数学表达式  $a + b\sqrt{x}$  分别生成符号变量,要求自变量  $x$  为实型变量。

解 输入: >> **x=sym('x','real')**%设定了变量  $x$  为实型变量

**a=sym('a')**

**b=sym('b')**

**r=sym('a+bsqrt(x)')**

运行后屏幕显示结果(略)。

##### 2、用 syms 生成符号变量

**syms** 可定义多个符号,用 **syms** 生成符号变量的常用形式请看下例。

[例 2] 试将  $y = 2a - 5e^{\sqrt{x-b}}$  中的系数  $a, b$  和自变量  $x$  生成符号变量。

解 输入:

>> **syms a b x**

**y=2\*a-5\*exp(sqrt(x-b))**

运行后屏幕显示:

**y =**

**2\*a-5\*exp((x-b)^(1/2))**

## (二)、符号表达式的生成

当符号变量生成以后，就可以进一步生成含有这些变量符号的表达式，常用的形式有两种。

方法一 `f=sym('数学表达式')`  
或 `f='数学表达式'`

方法二 `syms a b c x`  
`f=含有 a , b, c, x 的数学表达式`

[例 3] 试将  $z = a(2x-t)^3 + b\sin(4y)$  生成符号表达式。

解 方法一 输入：

```
>> f=sym(a*(2*x-t)^3+b*sin(4*y)),  
或 f='a*(2*x-t)^3+b*sin(4*y)'
```

运行后屏幕显示：

```
f =  
a*(2*x-t)^3+b*sin(4*y)  
f =  
a*(2*x-t)^3+b*sin(4*y)
```

方法二 输入：

```
>> syms a b c k t y  
f=a*(2*x-t)^3+b*sin(4*y)
```

运行后屏幕显示：

```
f =  
a*(2*x-t)^3+b*sin(4*y)
```

如果要生成方程的符号表达式，则不能用方法二，只能用方法一。

[例 4] `solve` 是 MATLAB 软件解方程的函数，调用格式为：`x=solve(f)`。试用函数 `solve` 解方程

$$8x^9 + 17x^3 - 3x = -1。$$

解 在 MATLAB 命令窗口输入命令：

```
>> f=sym('8*x^9+17*x^3-3*x=-1'),  
x=solve(f) % solve 是 MATLAB 软件解方程的函数
```

或输入命令：`>> f='8*x^9+17*x^3-3*x=-1',x=solve(f)`

或输入命令：`>> x=solve('8*x^9+17*x^3-3*x=-1')`

则运行后输出相同的结果：

```
x =  
[-.95778266908844934303923445046258-.59070709534579563538068568150441*i]  
[-.95778266908844934303923445046258+.59070709534579563538068568150441*i]  
[  
-.53275697173102083274354955617700]  
[-.62165529622636773563391338820961e-2-1.1576656136411973167991239807601*i]  
[-.62165529622636773563391338820961e-2+1.1576656136411973167991239807601*i]  
[.26762926822201693804563287259316-.19580612317589362415618068133297*i]  
[.26762926822201693804563287259316+.19580612317589362415618068133297*i]  
[.96274843969420649872171548984002-.57475793354361098651731421962321*i]  
[.96274843969420649872171548984002+.57475793354361098651731421962321*i]
```

[例 5] 生成一个符号矩阵。



**解** 在 MATLAB 命令窗口输入命令:

```
>> syms a b c d e f g h i j k l
```

```
A=[ a b c;d e f;g h i;j k l]
```

则运行后输出结果:

```
A =  
[ a, b, c]  
[ d, e, f]  
[ g, h, i]  
[ j, k, l]
```

#### 4.1.2 符号形式与数值形式的相互转换

在用 MATLAB 软件处理数学问题时, 经常需要符号形式与数值形式的相互转换。下面分别介绍它们的转换方法。

##### (一)、将符号形式转换为数值形式

将符号形式转换为数值形式常用的方法有两类, 一类是将数的符号形式转换为数值形式; 另一类是将多项式的符号形式转换为对应的系数的数值形式的向量。

##### 1、将数的符号形式转换为数值形式

将数的符号形式转换为数值形式主要用 MATLAB 函数 `eval` 和 `numeric` 来实现。

**[例 6]** 首先将  $a = \frac{1+\sqrt{5}}{2}$  生成为符号表达式, 然后再转换为数值形式。

**解** 首先输入:

```
>> a='(1+sqrt(5))/2'
```

运行后屏幕显示  $a = \frac{1+\sqrt{5}}{2}$  的符号表达式为

```
a =  
(1+sqrt(5))/2
```

然后再输入: `>> eval(a)`, 或 `>> numeric(a)`

运行后屏幕显示  $a = \frac{1+\sqrt{5}}{2}$  的相同数值形式为

```
ans =  
1.6180
```

##### 2、将多项式的符号形式转换为对应的系数的数值形式的向量

将多项式的符号形式转换为对应的系数的数值形式的向量主要用 MATLAB 函数 `sym2poly` 来实现。

**[例 7]** 将  $y = 5x^2 - 6$  转换为对应的系数的数值形式的向量。

**解** 输入: `>> syms x y`

```
y=5*x^2-6;
```

```
yc=sym2poly(y)
```

运行后屏幕显示  $y = 5x^2 - 6$  的系数的数值形式的向量为

$$y_c = \begin{bmatrix} 5 & 0 & -6 \end{bmatrix}$$

## (二)、将符号形式转换为数值形式

将数值形式转换为符号形式常用的方法有两类，一类是将数的数值形式转换为符号形式；另一类是将多项式的系数的数值形式的向量转换为对应的符号形式。

### 1、将数的数值形式转换为符号形式

将数的数值形式转换为符号形式主要用 MATLAB 函数 `sym` 来实现。

**[例 8]** 将  $\pi$  的近似值  $pj=3.141593$  转化为符号形式。

解 输入:

```
>> pj=3.141593;
```

$p_j = \text{sym}(p_j)$

运行后屏幕显示:

$$p_j =$$
$$7074238532074879 \cdot 2^{(-51)}$$

## 2、将多项式的系数的数值形式的向量转换为对应的符号形式

将多项式的符号形式转换为对应的系数的数值形式的向量主要用 MATLAB 函数 `poly2sym` 来实现。

**[例 9]** 将  $y = 5x^2 - 6$  的系数的向量  $\mathbf{vc} = [5 \quad 0 \quad -6]$  转换为对应的符号形式。

解 输入:

```
>> yc = [ 5 0 -6];
```

$$y = \text{poly2sym}(yc)$$

运行后屏幕显示为

$$y =$$
$$5x^2 - 6$$

## 4.2. 微积分运算

### 5.2.1 导数

在 MATLAB 系统中为用户提供了一元显函数求导的符号计算函数 `diff`，可以调用此函数求符号导数，不但使用方便，而且计算准确，迅速，尤其是求结构复杂的高阶导数更显示出其优越性。用 `diff` 可以求一元显函数的各阶导函数和在某点处的各阶导数。在这里将一元显函数  $y = f(x)$  的各阶导函数  $y', y'', y''', \dots$  分别记作  $yx, yxx, yxxx, \dots$ 。用 `diff` 作符号求导函数和在一点处的导数的调用格式和功能如表 4-1。

**【例 1】** 设一元函数  $y = \arctan(3x^2 + 2\sin(5x^3 - 2))$ , 求  $y$  的一阶导函数  $y'(x)$  和四阶导函数  $y^{(4)}(x)$ 。

解 下面用函数 `diff` 的两种调用格式求解。

(1) 输入计算  $y$  对  $x$  的一阶导函数和四阶导函数的程序:

```
>> syms x y
```

$$y = \arctan(3x^2 + 2\sin(5x^3 - 2));$$

```
yx=diff (y,x);
```

```
yx4=diff (y,x,4);
```

```
y1=simple(yx), pretty(y1)
```

y4=simple(yx4), pretty(y4)

表 4-1 求一元显函数  $y = f(x)$  的各阶导函数的 MATLAB 方法

符号求导的命令	功 能
$yx=\text{diff}(f(x),x)$	求函数 $y = f(x)$ 对 $x$ 的一阶导函数 $y' = f'(x)$
$yxx=\text{diff}(f(x),x,2)$ 或 $yxx=\text{diff}(yx,x)$	求函数 $y = f(x)$ 对 $x$ 的二阶导函数 $y'' = f''(x)$
$yxxx=\text{diff}(f(x),x,3)$ 或 $yxxx=\text{diff}(yxx,x)$	求函数 $y = f(x)$ 对 $x$ 的三阶导函数 $y''' = f'''(x)$
$yxn=\text{diff}(f(x),x,n)$	求函数 $y = f(x)$ 对 $x$ 的 $n$ 阶导函数 $y^{(n)} = f^{(n)}(x)$
$yn=\text{simple}(yxn)$	将 $n$ 阶导函数 $y^{(n)} = f^{(n)}(x)$ 化简，并记作 $yn$ 。
$\text{pretty}(\text{diff}(f(x),x))$	输出一个符合日常书写习惯的表达式

运行后屏幕显示  $y$  对  $x$  的一阶导数和四阶导数如下：

```
y1 =
(6*x+30*cos(5*x^3-2)*x^2)/(1+(3*x^2+2*sin(5*x^3-2))^2)
y4 =
(101250*sin(5*x^3-2)*x^8-81000*cos(5*x^3-2)*x^5-9000*sin(5*x^3-2)*x^2)/(1+(3*x^2+2*sin(5*x^3-2))^2)-8*(-6750*cos(5*x^3-2)*x^6-2700*sin(5*x^3-2)*x^3+60*cos(5*x^3-2))/(1+(3*x^2+2*sin(5*x^3-2))^2)^2*(3*x^2+2*sin(5*x^3-2))*(6*x+30*cos(5*x^3-2)*x^2)+48*(6-450*sin(5*x^3-2)*x^4+60*cos(5*x^3-2)*x)/(1+(3*x^2+2*sin(5*x^3-2))^2)^3*(3*x^2+2*sin(5*x^3-2))^2*(6*x+30*cos(5*x^3-2)*x^2)^2-12*(6-450*sin(5*x^3-2)*x^4+60*cos(5*x^3-2)*x)/(1+(3*x^2+2*sin(5*x^3-2))^2)^2*(6*x+30*cos(5*x^3-2)*x^2)^2-6*(6-450*sin(5*x^3-2)*x^4+60*cos(5*x^3-2)*x)^2/(1+(3*x^2+2*sin(5*x^3-2))^2)^2*(3*x^2+2*sin(5*x^3-2))-48*(6*x+30*cos(5*x^3-2)*x^2)^4/(1+(3*x^2+2*sin(5*x^3-2))^2)^4*(3*x^2+2*sin(5*x^3-2))^3+24*(6*x+30*cos(5*x^3-2)*x^2)^4/(1+(3*x^2+2*sin(5*x^3-2))^2)^3*(3*x^2+2*sin(5*x^3-2))
```

(2) 另外还可以用下面的 MATLAB 程序分别计算  $y$  对  $x$  的一阶导数和四阶导数：

```
>> syms x y
yx=diff((6*x+30*cos(5*x^3-2)*x^2)/(1+(3*x^2+2*sin(5*x^3-2))^2),x)
yx4=diff((6*x+30*cos(5*x^3-2)*x^2)/(1+(3*x^2+2*sin(5*x^3-2))^2),x,4)
```

所得到的结果与 (1) 相同。

[例 2] 设一元函数  $y = (3x^2 + 2\cos(5x^3 - 2))\ln(13x^5 + 6) + \frac{2x - 1}{5x^2 + 6}$ 。

(1) 求  $y$  的一阶导函数  $y'(x)$  和五阶导函数  $y^{(5)}(x)$ ；

(2) 求  $y$  在  $x = 0$  处的一阶导数  $y'(0)$  和五阶导数  $y^{(5)}(0)$ 。

解 (1) 输入计算  $y$  对  $x$  的一阶导函数  $y'(x)$  和五阶导函数  $y^{(5)}(x)$  的程序：

```
>> syms x y
y=(3*x^2+2*cos(5*x^3-2))*log(13*x^5+6)+(2*x-1)/(5*x^2+6);
yx=diff(y,x,1);
yx5=diff(y,x,5);
y1=simple(yx)
y5=simple(yx5)
```

运行后屏幕显示化简后的  $y$  对  $x$  的一阶导函数  $y1$  和五阶导函数  $y5$  如下:

```
y1 =
(6*x-30*sin(5*x^3-2)*x^2)*log(13*x^5+6)+65*(3*x^2+2*cos(5*x^3-2))*x^4/(13*x^5+6)+2/(5*x^2+
6)-10*(2*x-1)/(5*x^2+6)^2*x
y5 =
2400000*(2*x-1)/(5*x^2+6)^5*x^3-507000*(6-450*cos(5*x^3-2)*x^4-60*sin(5*x^3-2)*x)*x^7/(13*x
^5+6)^2-2028000*(6*x-30*sin(5*x^3-2)*x^2)*x^6/(13*x^5+6)^2+325*(101250*cos(5*x^3-2)*x^8+81000*sin(
5*x^3-2)*x^5-9000*cos(5*x^3-2)*x^2)*x^4/(13*x^5+6)+7800*(6-450*cos(5*x^3-2)*x^4-60*sin(5*x^3-2)*x)*x
^2/(13*x^5+6)+7800*(6*x-30*sin(5*x^3-2)*x^2)*x/(13*x^5+6)-90000*(2*x-1)/(5*x^2+6)^4*x-360000/(5*x^2+
6)^4*x^2+(-1518750*sin(5*x^3-2)*x^10+2025000*cos(5*x^3-2)*x^7+540000*sin(5*x^3-2)*x^4-18000*cos(5*
x^3-2)*x)*log(13*x^5+6)+2400000/(5*x^2+6)^5*x^4+6000/(5*x^2+6)^3+197730000*(3*x^2+2*cos(5*x^3-2))
*x^10/(13*x^5+6)^3+2600*(6750*sin(5*x^3-2)*x^6-2700*cos(5*x^3-2)*x^3-60*sin(5*x^3-2))*x^3/(13*x^5+6)
+1560*(3*x^2+2*cos(5*x^3-2))/(13*x^5+6)-42250*(6750*sin(5*x^3-2)*x^6-2700*cos(5*x^3-2)*x^3-60*sin(5*
x^3-2))*x^8/(13*x^5+6)^2+5492500*(6-450*cos(5*x^3-2)*x^4-60*sin(5*x^3-2)*x)*x^12/(13*x^5+6)^3+65910
000*(6*x-30*sin(5*x^3-2)*x^2)*x^11/(13*x^5+6)^3-535518750*(6*x-30*sin(5*x^3-2)*x^2)*x^16/(13*x^5+6)^
4-4284150000*(3*x^2+2*cos(5*x^3-2))*x^15/(13*x^5+6)^4+27846975000*(3*x^2+2*cos(5*x^3-2))*x^20/(13*
x^5+6)^5-12000000*(2*x-1)/(5*x^2+6)^6*x^5-2535000*(3*x^2+2*cos(5*x^3-2))*x^5/(13*x^5+6)^2
```

(2) 输入计算  $y$  在  $x = 0$  处的一阶导数  $y'(0)$  和五阶导数  $y^{(5)}(0)$  的程序:

```
>> x=0;
y1=(6*x-30*sin(5*x^3-2)*x^2)*log(13*x^5+6)+65*(3*x^2+2*cos(5*x^3.....
-2))*x^4/(13*x^5+6)+2/(5*x^2+6)-10*(2*x-1)/(5*x^2+6)^2*x,
y5=2400000*(2*x-1)/(5*x^2+6)^5*x^3-507000*(6-450*cos(5*x^3-2)*x^4.....
-60*sin(5*x^3-2)*x)*x^7/(13*x^5+6)^2-2028000*(6*x-30*sin(5*x^3-2)*x^2)*x^6/(13*x^5+6)^2+32
5*(101250*cos(5*x^3-2)*x^8+81000*sin(5*x^3-2)*x^5-9000*cos(5*x^3-2)*x^2)*x^4/(13*x^5+6)+7800*(6-450
*cos(5*x^3-2)*x^4-60*sin(5*x^3-2)*x)*x^2/(13*x^5+6)+7800*(6*x-30*sin(5*x^3-2)*x^2)*x/(13*x^5+6)-9000
0*(2*x-1)/(5*x^2+6)^4*x-360000/(5*x^2+6)^4*x^2+(-1518750*sin(5*x^3-2)*x^10+2025000*cos(5*x^3-2)*x^7
+540000*sin(5*x^3-2)*x^4-18000*cos(5*x^3-2)*x)*log(13*x^5+6)+2400000/(5*x^2+6)^5*x^4+6000/(5*x^2+6
)^3+197730000*(3*x^2+2*cos(5*x^3-2))*x^10/(13*x^5+6)^3+2600*(6750*sin(5*x^3-2)*x^6-2700*cos(5*x^3-
2)*x^3-60*sin(5*x^3-2))*x^3/(13*x^5+6)+1560*(3*x^2+2*cos(5*x^3-2))/(13*x^5+6)-42250*(6750*sin(5*x^3-
2)*x^6-2700*cos(5*x^3-2)*x^3-60*sin(5*x^3-2))*x^8/(13*x^5+6)^2+5492500*(6-450*cos(5*x^3-2)*x^4-60*si
n(5*x^3-2)*x)*x^12/(13*x^5+6)^3+65910000*(6*x-30*sin(5*x^3-2)*x^2)*x^11/(13*x^5+6)^3-535518750*(6*
x-30*sin(5*x^3-2)*x^2)*x^16/(13*x^5+6)^4-4284150000*(3*x^2+2*cos(5*x^3-2))*x^15/(13*x^5+6)^4+27846
975000*(3*x^2+2*cos(5*x^3-2))*x^20/(13*x^5+6)^5-12000000*(2*x-1)/(5*x^2+6)^6*x^5-2535000*(3*x^2+2*
cos(5*x^3-2))*x^5/(13*x^5+6)^2
```

运行后屏幕显示  $y$  在  $x = 0$  处的一阶导数  $y1$  和五阶导数  $y5$  如下:

```
y1 =
0.3333
```

$$y_5 = -188.6186$$

**[例 3]** 求下列函数的导数

$$(1) \quad y = \frac{3x^3}{1-2x} \cdot \sqrt{\frac{23-6x^5}{(6+7x)^2}}; \quad (2) \quad y = (\ln 5x)^{x+2}.$$

**解** 输入程序：

```
>> syms x
y=(log(5*x))^(x+2);
f=3*x^3*(sqrt((23-6*x^5)/(6+7*x^2)))/(1-2*x);
yx=diff(y,x);
fx=diff(f,x);
y1=simple(yx)
f1=simple(fx)
```

运行后屏幕显示（1）和（2）的导数分别为 f1 和 y1 如下：

```
y1 =
log(5*x)^(x+2)*(log(log(5*x))+(x+2)/x/log(5*x))
f1 =
3*(294*x^7+135*x^6-198*x^5-322*x^2-230*x+414)*x^2/(6+7*x)^2/(-1+2*x)^2/(23-6*x^5)^(1/2)
```

## 4.2.2 积分

如果函数  $f(x)$  在区间  $[a, b]$  上连续，且  $F(x)$  是  $f(x)$  的一个原函数，则函数  $f(x)$  在区间  $[a, b]$  上的

定积分为  $\int_a^b f(x)dx = F(b) - F(a)$ 。我们可以用 MATLAB 软件中的函数 `int` 对定积分进行符号计算的调用格式和功能如表 4-2 所示。

**表 4-2 函数 int 的求定积分的调用格式和功能**

函数 int 的求定积分的命令	功 能
$F=\text{int}(S,a,b)$	<p>输入量：S 可以是被积函数 <math>f(x)</math>，也可以是由几个被积函数 <math>f_1(x), f_2(x), \dots, f_n(x)</math> 构成的矩阵，积分上、下限 a 和 b 可以是数，也可以是符号标量，积分变量是 <math>x</math>。</p> <p>输出量：F 是 S 关于积分变量 <math>x</math> 的从 a 到 b 的定积分。</p>
$F=\text{int}(S,t,a,b)$	<p>输入量：S、a 和 b 同上，t 是积分变量。</p> <p>输出量：F 是 S 关于积分变量 t 的从 a 到 b 的定积分。</p>

**[例 1]** 求  $\int_4^5 \frac{5}{(t-1)(t-2)(t-3)} dt$

**解** 输入程序：

```
>> syms t
f=5/((t-1)*(t-2)*(t-3));
```

```
F=int(f,t,4,5)
```

```
y= numeric (F)
```

运行后屏幕显示计算的定积分值 F 和近似值 y 如下:

```
F =
```

```
25/2*log(2)-15/2*log(3)
```

```
y =
```

```
0.4247
```

即  $\int_4^5 \frac{5}{(t-1)(t-2)(t-3)} dt \approx 0.4247$ 。

**[例 2]** 求  $\int_0^1 \frac{ax+b}{\sqrt[n]{dx+e}} dx$

**解** 输入程序:

```
>> syms x a b d e n
```

```
f=(a*x+b)/((d*x+e)^(1/n));
```

```
F=int(f,0,1)
```

```
y=simple(F)
```

运行后屏幕显示计算的定积分 F 和化简后的定积分 y 如下:

```
F =
```

```
n*(-2*e*(d+e)^(1/n)*n*b*d+e^2*(d+e)^(1/n)*n*a+2*e^(1/n)*n*b*e*d+e^(1/n)*a*d^2*n+2*e^(1/n)*n*  
b*d^2-e^(1/n)*n*a*e^2-e^(1/n)*b*d^2-e^(1/n)*d*a*e-e^(1/n)*a*d^2+e*(d+e)^(1/n)*b*d-e^(1/n)*e*b*d)/(e^(1/n)  
)/((d+e)^(1/n))/d^2/(-3*n+1+2*n^2)
```

```
y =
```

```
n*((-2*e*(d+e)^(1/n)*b*d+e^2*(d+e)^(1/n)*a+e^(1/n)*a*d^2+2*e^(1/n)*b*d^2-e^(1/n)*a*e^2+2*e^(1  
/n)*e*b*d)*n-e^(1/n)*a*d^2+e*(d+e)^(1/n)*b*d-e^(1/n)*b*d^2-e^(1/n)*d*a*e-e^(1/n)*e*b*d)/(e^(1/n))/((d+e)^(  
1/n))/d^2/(-3*n+1+2*n^2)
```

**[例 3]** 求下列不定积分

$$(1) \int_1^2 \frac{2x^2}{\sqrt{9-x^2}} dx; (2) \int_1^2 \frac{\sqrt{x^2-1}}{2x} dx; (3) \int_1^2 \frac{dx}{\sqrt{(x^2+4)^3}}.$$

**解** 下面将三个不定积分的被积函数 f1, f2, f3 构成一个矩阵 f, 用一个积分命令求不定积分, 即输入程序:

```
>> syms x
```

```
f1=(2*x^2)/(sqrt(9-x^2));
```

```
f2=sqrt(x^2-1)/(2*x);
```

```
f3=1/(sqrt((4+x^2)^3));
```

```
f=[ f1; f2; f3];
```

```
F=int(f,1,2)
```

```
y= numeric(F)
```

运行后屏幕显示三个定积分值的矩阵 F 和近似值矩阵 y 如下:

```
F =
```

```
[-2*5^(1/2)+9*asin(2/3)+2*2^(1/2)-9*asin(1/3)]
```

```
[
```

```
1/2*3^(1/2)-1/6*pi]
```

```
[
```

```
1/8*2^(1/2)-1/20*5^(1/2)]
```

y =  
1.8653  
0.3424  
0.0650

即

$$\int_1^2 \frac{2x^2}{\sqrt{9-x^2}} dx \approx 1.8653, \quad \int_1^2 \frac{\sqrt{x^2-1}}{2x} dx \approx 0.3424, \quad \int_1^2 \frac{dx}{\sqrt{(x^2+4)^3}} \approx 0.0650。$$

**【例 4】** 求下列定积分

$$(1) \int_{\frac{\pi}{3}}^{\frac{2\pi}{5}} \frac{t^2 + 2t - 1}{(t-1)(t^2 - t + 1)} dt; \quad (2) \int_{\frac{\pi}{3}}^{\frac{2\pi}{5}} \frac{\cos^2 t + 1}{\cos 2t + 1} dt。$$

**解** 下面将两个不定积分的被积函数 f1, f2 构成一个矩阵 f, 用一个积分命令求不定积分, 即输入程序:

```
>> syms t
f1=(t^2+2*t-1)/((t-1)*(t^2-t+1));
f2=((cos(t))^2+1)/(cos(2*t)+1);
f=[ f1; f2];
F=int(f,pi/3,2*pi/5)
y= numeric(F)
```

运行后屏幕显示两个定积分的矩阵 F 及其近似值的矩阵 y 如下:

```
F =
[ -log(5)+2*log(2*pi-5)-1/2*log(25+4*pi^2-10*pi)+5/3*3^(1/2)*atan((4/15*pi-1/3)*3^(1/2))+log(3)-2*log(pi-3)+1/2*log(9+pi^2-3*pi)-5/3*3^(1/2)*atan((2/9*pi-1/3)*3^(1/2))]
[ 1/30*(pi*5^(1/2)-pi+15*2^(1/2)*(5+5^(1/2))^(1/2)-15*3^(1/2)*5^(1/2)+15*3^(1/2))/(5^(1/2)-1)]
y =
3.7172
0.7775
```

$$\text{即 } \int_{\frac{\pi}{3}}^{\frac{2\pi}{5}} \frac{t^2 + 2t - 1}{(t-1)(t^2 - t + 1)} dt \approx 3.7172, \quad \int_{\frac{\pi}{3}}^{\frac{2\pi}{5}} \frac{\cos^2 t + 1}{\cos 2t + 1} dt \approx 0.7775。$$

**【例 5】** 计算下列定积分:

$$(1) \int_0^1 e^{\sqrt{2}t} dt; \quad (2) \int_0^1 (\cos 3t)^2 dt; \quad (3) \int_0^1 \frac{t^2}{(1+t^2)^2} dt$$

**解** 因为题设给定的三个定积分的上、下限分别相同, 所以可以用向量的形式输入程序:

```
>> syms t
A=[exp(sqrt(2*t)),(cos(3*t))^2,(t^20/((1+t^2)^2))];
g=int(A,t,0,1)
S=numeric(g)
```

运行后屏幕显示计算三个定积分的值的向量 g 及其近似值的向量 S 依次为

```
g =
[exp(2^(1/2))*2^(1/2)-exp(2^(1/2))+1, 1/6*cos(3)*sin(3)+1/2, 22894441/3063060-19/8*pi]
S =
2.7038 0.4767 0.0131
```

即  $\int_0^1 e^{\sqrt{2}t} dt \approx 2.7038$ ,  $\int_0^1 (\cos 3t)^2 dt \approx 0.4767$  和  $\int_0^1 \frac{t^2}{(1+t^2)^2} dt \approx 0.0131$ 。

[例 6] 已知 
$$f(x) = \begin{cases} 1 + \sin x, & x \leq 1 \\ \frac{1}{2}x^2 + 5x - 7, & x > 1 \end{cases}$$
, 求  $\int_0^2 f(x) dx$ 。

解 输入程序:

```
>> syms x
f1=sin(x)+1; f2=(x^2)/2+5*x-7;
F=int(f1,x,0,1)+ int(f2,x,1,2)
y= numeric (F)
```

运行后屏幕显示计算的定积分值 F 和近似值 y 如下:

```
F =
-cos(1)+11/3
y =
3.1264
```

即  $\int_0^2 f(x) dx \approx 3.1264$ 。

[例 7] 计算  $\int_0^\pi (2 + |3 \cos x|) dx$

$$|3 \cos x| = \begin{cases} 3 \cos x, & 0 \leq x \leq \frac{\pi}{2} \\ -3 \cos x, & \frac{\pi}{2} < x \leq \pi \end{cases}$$

解 由于

根据积分的可加性, 有

$$\int_0^\pi (2 + |3 \cos x|) dx = \int_0^{\frac{\pi}{2}} (2 + 3 \cos x) dx + \int_{\frac{\pi}{2}}^\pi (2 - 3 \cos x) dx$$

输入程序:

```
>> syms x
f1=3*cos(x)+2; f2=-3*cos(x)+2;
F=int(f1,x,0,pi/2)+ int(f2,x,pi/2,pi)
y= numeric (F)
```

运行后屏幕显示计算的定积分值 F 和近似值 y 如下:

```
F =
6+2*pi
y =
12.2832
```

## 4.3 解方程

### 4.3.1 代数方程

求方程  $f(x)=q(x)$  的根可以用 MATLAB 命令:



```
>> x=solve('方程 f(x)=q(x)', '待求符号变量 x')
求方程组  $f_i(x_1, \dots, x_n) = q_i(x_1, \dots, x_n), (i=1, 2, \dots, n)$  的根可以用 MATLAB 命令:
>> E1=sym('方程 f1(x1,...,xn)=q1(x1,...,xn)');
.....
En=sym('方程 fn(x1,...,xn)=qn(x1,...,xn)');
[x1,x2,...,xn]=solve(E1,E2,...,En, x1,...,xn)
```

**[例 1]** 解方程  $8x^9 + 17x^3 - 3x = -1$ .

**解** 在 MATLAB 命令窗口输入命令:

```
>> x=solve('8*x^9+17*x^3-3*x=-1')
```

则运行后输出此方程的根为

```
x=
[ -0.95778266908844934303923445046258-.59070709534579563538068568150441*i]
[ -0.95778266908844934303923445046258+.59070709534579563538068568150441*i]
[ -0.53275697173102083274354955617700]
[ -0.62165529622636773563391338820961e-2-1.1576656136411973167991239807601*i]
[ -0.62165529622636773563391338820961e-2+1.1576656136411973167991239807601*i]
[ 0.26762926822201693804563287259316-.19580612317589362415618068133297*i]
[ 0.26762926822201693804563287259316+.19580612317589362415618068133297*i]
[ 0.96274843969420649872171548984002-.57475793354361098651731421962321*i]
[ 0.96274843969420649872171548984002+.57475793354361098651731421962321*i]
```

**[例 2]** 解下列方程，并且求根的近似值，要求精度为  $10^{-14}$ .

(1)  $\sin(\cos(2x^3)) = 0$ ;      (2)  $\sin(\cos(4)) * a + \tan(a+3) * 5 = 0$ ;

(3)  $\sin(\cos(x)) * a + \tan(a) * b = 0$ .

**解** 只需在 MATLAB 工作窗口输入命令:

```
>> x1=solve('sin(cos(2*x^3))=0','x')
a1=solve('sin(cos(4))*a+tan(a+3)*5=0','a')
x2=solve('sin(cos(x))*a+tan(a)*b=0','x')
```

则运行后输出三个方程的根如下:

```
x1 =
[ 1/2*2^(1/3)*pi^(1/3)]
[ -1/4*2^(1/3)*pi^(1/3)+1/4*i*3^(1/2)*2^(1/3)*pi^(1/3)]
[ -1/4*2^(1/3)*pi^(1/3)-1/4*i*3^(1/2)*2^(1/3)*pi^(1/3)]
a1 =
-3.3911693397183601737264718371364
x2 =
pi-acos(asin(tan(a)*b/a))
```

若想求精度为  $10^{-14}$  的根的近似值，首先需要选中 MATLAB 工作窗口中的下拉菜单 File 的子菜单 Preferences，单击鼠标左键，在 General 对话框的 Double Formal 中选 long；然后，在 MATLAB 工作窗口依次输入程序：

```
>>x1= 1/2*2^(1/3)*pi^(1/3),
x2=-1/4*2^(1/3)*pi^(1/3)+1/4*i*3^(1/2)*2^(1/3)*pi^(1/3),
```

$$x3 = -1/4*2^{(1/3)}*pi^{(1/3)} - 1/4*i*3^{(1/2)}*2^{(1/3)}*pi^{(1/3)}$$

运行后即可得到近似值:

$$x1 = 0.92263507432201$$

$$x2 = -0.46131753716101 + 0.79902541278541i$$

$$x3 = -0.46131753716101 - 0.79902541278541i$$

MATLAB 系统本来只能做数值计算, 并没有符号运算的功能, 符号运算工具箱(Symbolic Math Toolbox)则扩充了 MATLAB 这方面的功能, 它是由 Maple 软件的核心来完成的. 用 solve 命令求方程的根的方法有一个缺点, 它不能求出周期函数  $f(x)$  对应的方程  $f(x)=0$  的全部根 (如  $\sin x=0$ , 用命令

`>>x=solve('sin(x)=0')`, 运行后输出一个根  $x=0$ ).

**[例 3]** 解超越方程组 
$$\begin{cases} x^x - 4 = 0, \\ 2xy + x = 1. \end{cases}$$

**解** 在 MATLAB 命令窗口输入命令:

```
>> E1=sym('x^x-4=0'); E2=sym('2*x*y+x=1');
```

```
[x,y]=solve(E1,E2)
```

```
x1= double (x),y1= double (y)
```

则运行后输出超越方程组精确解  $x,y$  和近似解  $x1,y1$  如下:

$$x = \log(4)/\text{lambertw}(\log(4))$$

$$y = -1/2*(\log(4)-\text{lambertw}(\log(4)))/\log(4)$$

$$x1 = 2$$

$$y1 = -0.2500.$$

**[例 4]** 解非线性方程组 
$$\begin{cases} x + y + z = u, \\ 2xz - yu = -1, \\ (x + y)^2 - z = u, \\ xu + yz = 4. \end{cases}$$

**解** 在 MATLAB 命令窗口输入命令:

```
>> E1=sym('x+y+z=u'); E2=sym('2*x*z-y*u=-1'); E3=sym('(x+y)^2-z=u');
```

```
E4=sym('x*u+y*z=4'); [x,y,z,u]=solve(E1,E2,E3,E4), x1= double (x)
```

```
y1= double (y), z1= double (z), u1= double (u)
```

则运行后输出方程组精确解  $x,y,z,u$ (略)和近似解  $x1,y1,z1,u1$  如下:

$x1 =$	$y1 =$
3.000000000000000	1.000000000000000
0.09943200266874	23.60367144267526
-1.77193822355659 + 0.46109615509798i	0.25371983421793 - 0.42474610911316i
-1.77193822355659 - 0.46109615509798i	0.25371983421793 + 0.42474610911316i
$z1 =$	$u1 =$
1.000000000000000	1.000000000000000
-23.43369830490221	-0.07054113510432
-1.00537306977112 - 1.40752182928383i	-1.02028498800340 + 2.29336409349498i
-1.00537306977112 + 1.40752182928383i	-1.02028498800340 - 2.29336409349498i

### 4.3.2 微分方程

在 MATLAB 系统中提供了常微分方程(组)符号解(Symbolic solution of ordinary differential equations)



```
x4=dsolve('D2x+a*Dx/m+b*x/m = F(t)/m'),
```

```
y5= dsolve('D2y = sin(y)'), pretty(y5)
```

运行后屏幕显示常微分方程 (1)、(2)、(3)、(4) 和 (5) 的通解  $x1$ 、 $y2$ 、 $y3$ 、 $x4$  和  $y5$  依次如下:

```
x1 =
```

```
C1*exp(-a*t)
```

```
y2 =
```

```
-b*(a*cos(x)-a*sin(x)+sin(x)+cos(x))/(a^2+1)+C1+C2*exp(-a*x)
```

```
Warning: Explicit solution could not be found; implicit solution returned.
```

```
> In C:\MATLAB6p5\toolbox\symbolic\dsolve.m at line 292
```

```
y3 =
```

```
t+1/y-log(y)+log(-1+y)+C1=0
```

```
x4 =
```

```
(-int(F(t)*exp(1/2*(a+(a^2-4*m*b)^(1/2))*t/m),t)*exp(1/2*(a-(a^2-4*m*b)^(1/2))*t/m)+int(F(t)*exp(1/2*(a-(a^2-4*m*b)^(1/2))*t/m),t)*exp(1/2*(a+(a^2-4*m*b)^(1/2))*t/m))*exp(-a*t/m)/(a^2-4*m*b)^(1/2)+C1*exp(-1/2*(a+(a^2-4*m*b)^(1/2))*t/m)+C2*exp(-1/2*(a-(a^2-4*m*b)^(1/2))*t/m)
```

```
Warning: Explicit solution could not be found; implicit solution returned.
```

```
> In C:\MATLAB6p5\toolbox\symbolic\dsolve.m at line 292
```

```
y5 =
```

```
[Int(1/(-2*cos(a)+C1)^(1/2),a='..y)-t-C2=0, -Int(1/(-2*cos(a)+C1)^(1/2),a='..y)-t-C2=0]
```

其中  $C1$  和  $C2$  是任意常数.因为常微分方程 (3) 和 (5) 的显式解没有被找到, 所以返回隐式解.

**[例 2]** 求下列常微分方程组的通解:

$$(1) \begin{cases} \frac{df(t)}{dt} = af(t) + bg(t), \\ \frac{dg(t)}{dt} = -af(t) + bg(t); \end{cases} \quad (2) \begin{cases} \frac{dy}{dx} + 4z = \sin x, \\ \frac{dy}{dx} + 3\frac{dz}{dx} = \cos x. \end{cases}$$

**解** 输入程序:

```
>> S1=dsolve('Df = a*f + b*g','Dg = -a*f + b*g'),
```

```
f=S1.f,g=S1.g,de1='Dy + 4*z=sin(x)';
```

```
de2='Dy + 3*Dz=cos(x)';
```

```
S2=dsolve(de1, de2, 'x'), Y=S2.y,z=S2.z
```

运行后屏幕显示微分方程组 (1) 和 (2) 的通解如下:

```
S1 =
```

```
f: [1x1 sym]
```

```
g: [1x1 sym]
```

```
f =
```

```
C1*exp(1/2*a*t+1/2*t*b-1/2*t*(a^2-6*a*b+b^2)^(1/2))+C2*exp(1/2*a*t+1/2*t*b+1/2*t*(a^2-6*a*b+b^2)^(1/2))
```

```
g =
```

```
-1/2*(C1*a*exp(1/2*a*t+1/2*t*b-1/2*t*(a^2-6*a*b+b^2)^(1/2))-C1*exp(1/2*a*t+1/2*t*b-1/2*t*(a^2-6*a*b+b^2)^(1/2))*b+C1*(a^2-6*a*b+b^2)^(1/2)*exp(1/2*a*t+1/2*t*b-1/2*t*(a^2-6*a*b+b^2)^(1/2))+C2*a*exp(1/2*a*t+1/2*t*b+1/2*t*(a^2-6*a*b+b^2)^(1/2))-C2*exp(1/2*a*t+1/2*t*b+1/2*t*(a^2-6*a*b+b^2)^(1/2))*b-C2*(a^2-6*a*b+b^2)^(1/2)*exp(1/2*a*t+1/2*t*b+1/2*t*(a^2-6*a*b+b^2)^(1/2)))/b
```

```
S2 =
```

```
y: [1x1 sym]
```

```

z: [1x1 sym]
Y =
C1-3*exp(4/3*x)*C2+3*C2+4/25*sin(x)+3/25*cos(x)
z =
exp(4/3*x)*C2-1/25*cos(x)+7/25*sin(x)

```

## (二)、求常微分方程（组）的特解

如果给定常微分方程（4.1）的初始条件

$$y(x_0) = a_0, y'(x_0) = a_1, \dots, y^{(n)}(x_0) = a_n, \quad (4.3)$$

则求方程（4.1）的特解的主要调用格式如下：

**调用格式三：** `S=dsolve('eqn', 'condition1', ..., 'conditionn', 'var')`

输入的量：eqn 是常微分方程（4.1）改用符号方程表示的常微分方程  $F(x, y, Dy, D^2y, \dots, D^ny) = 0$ ;

condition1, ..., conditionn 是初始条件（4.3）；Var 表示自变量，默认自变量为  $t$ 。

输出的量： $S$  是常微分方程（4.1）的特解。

同理，如果给定常微分方程组（4.2）的初始条件，则求方程（4.2）的特解的主要调用格式如下：

**调用格式四：** `S=dsolve('eqn1','eqn2', ..., 'eqnm', 'condition1', 'condition2', ..., 'var')`

输入的量：eqn1, eqn2, ..., eqnm 分别是常微分方程组（4.2）中用符号方程表示的  $m$  个常微分方程；condition1, condition2, ... 是常微分方程组（4.2）的初始条件；默认自变量为  $t$ ，也可以将自变量  $t$  变为其它的符号变量 var。

输出的量： $S$  是常微分方程组（4.2）的特解。

**[例 3]** 求下列常微分方程在给定初始条件下的特解：

$$(1) \left( \frac{dy}{dx} \right)^2 + y^2 = 1, y(0) = 0; \quad (2) \frac{d^3 w}{dt^3} = -w, w(0) = 1, \frac{dw}{dt} \Big|_{t=0} = 0, \frac{d^2 w}{dt^2} \Big|_{t=0} = 0;$$

$$(3) \frac{d^2 f(x)}{dx^2} = -a^2 \frac{df(x)}{dx}, f(0) = 1, \frac{df(x)}{dx} \Big|_{x=\frac{\pi}{a}} = 0.$$

**解** 输入程序：

```

>> y = dsolve('(Dy)^2 + y^2 = 1','y(0) = 0')
w = dsolve('D3w = -w','w(0)=1, Dw(0)=0, D2w(0)=0')
f = dsolve('D2f = -a^2*f','f(0) = 1, Df(pi/a) = 0')

```

运行后屏幕显示常微分方程（1）、（2）和（3）在给定初始条件下的特解  $y, w, f$  依次如下：

```

y =
[ sin(t)]
[-sin(t)]
w =
1/3*exp(-t)+2/3*exp(1/2*t)*cos(1/2*3^(1/2)*t)
f =
cos(a*t)

```

**[例 4]** 求下列微分方程组满足所给初始条件的特解：

$$(1) \begin{cases} \frac{df(t)}{dt} = af(t) + bg(t), \\ \frac{dg(t)}{dt} = -af(t) + bg(t), \end{cases} f(0) = 1, g(0) = 2;$$

$$(2) \begin{cases} \frac{d^2y}{dx^2} + 2\frac{dy}{dx} + y + \frac{dz}{dx} + z = 0, \\ \frac{dy}{dx} + y + \frac{d^2z}{dx^2} + 2\frac{dz}{dx} + z = e^x, \end{cases} y(0) = 0, \left. \frac{dy}{dx} \right|_{x=3} = 2, z(5) = 1, \left. \frac{dz}{dx} \right|_{x=0} = 0$$

解 输入程序：

```
>> S1=dsolve('Df=a*f+b*g','Dg=-a*f+b*g','f(0)=1','g(0)=2')
```

```
f=S1.f; fs=simple(f), g=S1.g;
```

```
gs=simple(g), de1='D2y+2*Dy+y+Dz+z=0';
```

```
de2='Dy+y+D2z+2*Dz+z=exp(x)';
```

```
S2=dsolve(de1,de2,'y(0)=0, Dy(3)=2, z(5)=1, Dz(0)=0')
```

```
y=S2.y; ys=simple(y), z=S2.z; zs=simple(z),
```

运行后屏幕显示常微分方程组 (1) 和 (2) 在给定初始条件下的特解  $f, g, y$  和  $z$  依次如下：

```
S1 =
```

```
f: [1x1 sym]
```

```
g: [1x1 sym]
```

```
fs =
```

```
-1/2*(a+3*b-(a^2-6*a*b+b^2)^(1/2))/(a^2-6*a*b+b^2)^(1/2)*exp(1/2*a*t+1/2*t*b-1/2*t*(a^2-6*a*b+b^2)^(1/2))+1/2*(a+3*b+(a^2-6*a*b+b^2)^(1/2))/(a^2-6*a*b+b^2)^(1/2)*exp(1/2*a*t+1/2*t*b+1/2*t*(a^2-6*a*b+b^2)^(1/2))
```

```
gs =
```

```
(2*exp(1/2*(a+b-(a^2-6*a*b+b^2)^(1/2))*t)*a-exp(1/2*(a+b-(a^2-6*a*b+b^2)^(1/2))*t)*b+exp(1/2*(a+b-(a^2-6*a*b+b^2)^(1/2))*t*(a^2-6*a*b+b^2)^(1/2))-2*exp(1/2*(a+b+(a^2-6*a*b+b^2)^(1/2))*t)*a+exp(1/2*(a+b+(a^2-6*a*b+b^2)^(1/2))*t)*b+exp(1/2*(a+b+(a^2-6*a*b+b^2)^(1/2))*t*(a^2-6*a*b+b^2)^(1/2))/(a^2-6*a*b+b^2)^(1/2)
```

```
S2 =
```

```
y: [1x1 sym]
```

```
z: [1x1 sym]
```

```
ys =
```

```
-1/2*exp(-3)*(-2*exp(x-2)-exp(-1-t+x)+exp(1+x)-4*exp(-1-t)+exp(-1+x)+4*exp(-1)+2*exp(4-t+x)-2*exp(x+4)+8*exp(4-t)-8*exp(4)+2*exp(-2*t+6)+4*exp(-2*t+9)+2*exp(1+x)*t+2*exp(x+3)*t-exp(-4+x)*t-12*exp(x+3)-2*exp(6)-6*exp(x-2*t+6)+2*exp(-2-t+x)-exp(x-t+9)+12*exp(x-t+3)+4*exp(3)+6*exp(x+6)-exp(x-2*t+1)-exp(x+6)*t-4*exp(3-t)-4*exp(9-t)+exp(x-2*t+9))/(2*exp(-2)+2-exp(-7)-exp(3))
```

```
zs =
```

```
-1/2*(4+3*exp(x-2)+4*exp(-4)-2*exp(1+x)-8*exp(1)-exp(x-7)+8*exp(6-t)-4*exp(-2*t+6)+2*exp(x-t+6)+exp(-4+x)+exp(x-2*t-2)-exp(x-7)*t+5*exp(x+3)+2*exp(x-2)*t-exp(x-t-7)-10*exp(x)-exp(x-2*t+6)-13*exp(x-t+3)-exp(x+3)*t-2*exp(3)+2*exp(x-t)+6*exp(x-2*t+3)-2*exp(x-5)+2*exp(x)*t-2*exp(-2*t+3)+4*exp(3-t))/(-2*exp(-2)-2+exp(-7)+exp(3))
```

## 4.4 线性代数

### 4.4.1 逆矩阵、行列式及其 MATLAB 命令

当矩阵  $A$  为方阵时,  $A$  的行列式通常表示为  $|A|$  或  $\det(A)$ ; 当  $\det(A) \neq 0$  时,  $A$  可逆, 且可以用表 4-3

中的 MATLAB 命令求  $A$  的逆矩阵和行列式.

表 4-3

命 令	功 能
NA=inv(A)	输入矩阵 $A$ ，运行后输出 $A$ 的逆矩阵 $A^{-1}$ ；
HA=det(A)	输入矩阵 $A$ ，运行后输出 $A$ 的行列式 $ A $ 的值.

[例 1] 求下列矩阵的逆矩阵和行列式

$$(1) A = \begin{pmatrix} 1 & -5 \\ -1 & 4 \end{pmatrix}; \quad (2) B = \begin{pmatrix} 1 & 2 & -3 \\ 2 & 0 & 4 \\ 0 & -1 & 5 \end{pmatrix}; \quad (3) C = \begin{pmatrix} 4 & 2 & 3 \\ 0 & -1 & 5 \\ 2 & 1 & d \end{pmatrix}.$$

解 在 MATLAB 工作窗口输入命令:

```
>> A=[1 -5;-1 4]; NA=inv(A), B=[1 2 -3;2 0 4; 0 -1 5]; NB=inv(B),
syms d
C=[4 2 3; 0 -1 5; 2 1 d]; NC=inv(C),
HA=det(A), HB =det(B), HC =det(C)
```

运行后分别输出  $A$ 、 $B$ 、 $C$  的逆矩阵和行列式如下:

```
NA =    -4    -5
      -1    -1
NB = -0.400000000000000    0.700000000000000 -0.800000000000000
      1.000000000000000 -0.500000000000000    1.000000000000000
      0.200000000000000 -0.100000000000000    0.400000000000000
NC = [ 1/2*(d+5)/(2*d-3),    1/2,    -13/2/(2*d-3)]
      [    -5/(2*d-3),    -1,    10/(2*d-3)]
      [    -1/(2*d-3),    0,    2/(2*d-3)]
HA=-1, HB=-10, HC=-4*d+6
```

#### 4.4.2 方程组的逆矩阵解法及其 MATLAB 程序

在 MATLAB 中引进了矩阵除法的概念，它包括左除和右除.设两矩阵  $A, B$  为  $n$  阶方阵，且  $A, B$  皆可逆，则矩阵方程  $AX = C$ ， $YB = D$ ， $AZB = F$  的解都可以用表 4-4 中的命令求出.

表 4-4

命 令	相同功能的命令	功 能
$X=A \backslash C$	$X = \text{inv}(A)*C$	输入矩阵 $A$ 和 $C$ ，运行后输出矩阵方程 $AX = C$ 的解 $X = A^{-1}C$ ；
$Y=D/B$	$Y=D*\text{inv}(B)$	输入矩阵 $B$ 和 $D$ ，运行后输出矩阵方程 $YB = D$ 的解 $Y = DB^{-1}$ ；
$Z=A \backslash F/B$	$Z = \text{inv}(A)*F*\text{inv}(B)$	输入矩阵 $A$ 、 $B$ 和 $F$ ，运行后输出矩阵方程 $AZB = F$ 的解 $Z = A^{-1}FB^{-1}$ 。

[例 2] 解下列矩阵方程

$$(1) \begin{pmatrix} 1 & -5 \\ -1 & 4 \end{pmatrix} X = \begin{pmatrix} 3 & 2 \\ 1 & 4 \end{pmatrix}; \quad (2) Y \begin{pmatrix} 1 & -1 & 1 \\ 1 & 1 & 0 \\ 2 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & -3 \\ 2 & 0 & 4 \\ 0 & -1 & 5 \end{pmatrix};$$

$$(3) \begin{pmatrix} 1 & -1 & 1 \\ 1 & 1 & 0 \\ 2 & 1 & 1 \end{pmatrix} Z \begin{pmatrix} 1 & -1 & 1 \\ 1 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 4 & 2 & 3 \\ 0 & -1 & 5 \\ 2 & 1 & 1 \end{pmatrix}.$$

**解** 下面用两种方法求矩阵方程的解.

**方法一** 在 MATLAB 工作窗口输入命令:

```
>> A1=[1 -5;-1 4];C=[3 2; 1 4]; X=A1\C,
B1=[1 -1 1;1 1 0;2 1 1]; D=[1 2 -3;2 0 4; 0 -1 5];
Y=D/B1, A2=[1 -1 1;1 1 0;2 1 1];
B2=[1 -1 1;1 1 0;3 2 1];F=[4 2 3; 0 -1 5; 2 1 1];Z=A2\F/B2
```

运行后输出矩阵方程 (1)、(2)、(3) 的解分别为:

```
X =   -17   -28
      -4    -6

Y =
      2      9     -5
     -2     -8      6
     -4    -14      9

Z =
-8.999999999999999 -51.999999999999999  20.999999999999999
 4.999999999999999  28.999999999999999 -12.000000000000000
12.999999999999999  73.999999999999999 -28.999999999999999
```

**方法二** 在 MATLAB 工作窗口输入命令:

```
>> A1=[1 -5;-1 4];C=[3 2; 1 4]; X=inv(A1)*C,
B1=[1 -1 1;1 1 0;2 1 1]; D=[1 2 -3;2 0 4; 0 -1 5];
Y=D*inv(B1), A2=[1 -1 1;1 1 0;2 1 1];
B2=[1 -1 1;1 1 0;3 2 1]; F=[4 2 3; 0 -1 5; 2 1 1];
Z=inv(A2)*F*inv(B2)
```

运行后输出矩阵方程 (1)、(2) 的解与法一相同, 矩阵方程 (3) 的解如下:

```
Z =   -9.000000000000000 -52.000000000000000  21.000000000000000
      5.000000000000000  29.000000000000000 -12.000000000000000
     13.000000000000000  74.000000000000001 -29.000000000000001
```

在 MATLAB 工作窗口输入命令:

```
>> (A2\F)/B2-inv(A2)*F*inv(B2)
```

运行后输出矩阵方程 (3) 的两种解法的解的差为

```
ans =   1.0e-013 *
      0.07105427357601    0.14210854715202   -0.10658141036402
     -0.07105427357601   -0.07105427357601    0.03552713678801
     -0.17763568394003   -0.28421709430404    0.14210854715202
```

**[例 3]** 利用逆矩阵解下列线性方程组



$$(1) \begin{cases} x_1 + 2x_2 + 3x_3 = 1, \\ 2x_1 + 2x_2 + 5x_3 = 2, \\ 3x_1 + 5x_2 + x_3 = 3; \end{cases} \quad (2) \begin{cases} x_1 - x_2 - x_3 = 2, \\ 2x_1 - x_2 - 3x_3 = 1, \\ 3x_1 + 2x_2 - 5x_3 = 0. \end{cases}$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 5 \\ 3 & 5 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

解 (1) 方程组可表示为

在 MATLAB 工作窗口输入命令:

```
>> A=[1 2 3;2 2 5;3 5 1];C=[1;2;3]; X=A\C
```

或 >> A=[1 2 3;2 2 5;3 5 1];C=[1;2;3]; X=inv(A)\*C

运行后输出线性方程组 (1) 的解为  $x_1 = 1$ ,  $x_2 = 0$ ,  $x_3 = 0$ .

(2) 同理, 在 MATLAB 工作窗口输入命令:

```
>> A=[1 -1 -1;2 -1 -3;3 2 -5];C=[2;1;0]; X=A\C
```

运行后输出线性方程组 (1) 的解为  $x_1 = 5$ ,  $x_2 = 0$ ,  $x_3 = 3$ .

## 4.5 化简和代换

符号表达式的初等运算是符号表达式的变换和化简、符号的代换、复合函数的运算和反函数的运算、加、减、乘、除运算、乘方和开方运算。有关加、减、乘、除运算、乘方和开方运算在第一章实验的内容中我们已经作了介绍。下面主要介绍符号表达式的变换和化简、符号的代换、复合函数和反函数的运算。在 MATLAB 中, 这些运算的常用函数如表 4-5 所示。

表 4-5 符号表达式的变换和化简、复合函数和反函数的运算的 MATLAB 函数

MATLAB 函数名		函数功能
变 换 和 化 简	expand(表达式)	将表达式中的括号进行展开,即将乘积展开为和式
	factor(表达式)	将表达式进行分解,即把多项式转换为乘积形式
	horner(表达式)	将一般的表达式变为嵌套的形式
	collect(表达式)	合并同类项
	simplify(表达式)	利用各种恒等式关系、函数关系将表达式进行化简
	simple(表达式)	(1) f=simple(S) 对表达式 S 进行化简,输出长度最短的表达式 f; (2) simple(S) 对表达式 S 进行化简,输出用各种函数化简的结果,及长度最短的表达式; (3) [f,how]=simple(S) 对表达式 S 进行化简,输出长度最短的表达式 f,及 f 是哪一个函数作用的结果 how。
代 换	findsym(表达式)	确认符号表达式中的符号
	subs(f,old,new)	用符号 new 代替表达式 f 中的符号 old
复合 函数 运算	subexpr(f)	将表达式 f 中的公共部分用 sigma 表示
	y=compose(f,g)	输出 y=f(u)和 u=g(x)的复合函数 y=f(g(x))
复合 函数 运算	y=compose(f,g,t)	输出 y=f(u)和 u=g(t)的复合函数 y=f(g(t))

反函数运算	$g=\text{finverse}(f)$	输入函数 $y=f(x)$ ，其中 $y=f(x)$ 和 $y=g(x)$ 互为反函数，输出的 $y=g(x)$ 是 $y=f(x)$ 的反函数
	$g=\text{finverse}(f,t)$	输入函数 $y=f(t)$ ，其中 $y=f(t)$ 和 $y=g(t)$ 互为反函数，输出的 $y=g(t)$ 是 $y=f(t)$ 的反函数

**[例 1]** 设函数  $y = 5x^2 - 6$  和  $z = \sqrt{3y}$ ，试求  $z$  关于  $x$  的复合函数。

**解** 输入

```
>> syms x y
z=sqrt(3*y);
y=5*x^2-6;
z=compose(z,y)
```

运行后屏幕显示为

```
z =
3^(1/2)*(5*x^2-6)^(1/2)
```

**[例 2]** 分别求函数  $y_1 = e^{-x}$  和  $y_2 = \sin t$  的反函数。

**解** 输入

```
>> syms x t y1 y2
y1=exp(-x);
y2=sin(t);
g1=finverse(y1)
g2=finverse(y2,t)
```

运行后屏幕显示为

```
g1 =
-log(x)
g2 =
asin(t)
```

**[例 3]** 求函数  $y = \sqrt[5]{\frac{ax+b}{cx+d}}$  ( $ac \neq 0$ ) 的反函数。

**解** 输入

```
>> syms x a b c d
y=((a*x+b)/(c*x+d))^(1/5)';
g=finverse(y)
```

运行后屏幕显示为

```
g =
-(b-conj(x)^5*d)/(a-conj(x)^5*c)
```

为了用符号  $x$  代替表达式  $g$  中的符号  $\text{conj}(x)$ ，再输入：

```
>> g=subs(-(b-conj(x)^5*d)/(a-conj(x)^5*c), conj(x), x)
```

运行后屏幕显示为

```
g =
(-b+x^5*d)/(a-x^5*c)
```

**[例 4]** 在 MATLAB 工作窗口输入下面程序，运行后，观察其结果，说明所使用 MATLAB 函数 `collect`、`expand`、`horner`、`factor` 和 `simplify` 的意义和用法。

**解** 在 MATLAB 工作窗口输入下面程序：

```
>> syms x y t a
f1=collect(x^3+2*x^2-5*x^2+4*x-3*x+12-3),
f2=collect((1+x)*t+t*x),
f3=expand((x-1)*(x-2)*(x-3)),
f4=expand(cos(x+y)),
f5=expand(cos(3*acos(x))),
f6=horner(x^3-6*x^2+11*x-6),
f7=horner(1.1+2.2*x+3.3*x^2),
f8=factor(x^3-6*x^2+11*x-6),
f9=simplify((1-x^2)/(1-x)),
f10=simplify(sin(x)^2+cos(x)^2),
f11=simplify((1/a^3+6/a^2+12/a+8)^(1/3))
```

运行后屏幕显示如下：

```
f1 =
x^3-3*x^2+x+9
f2 =
2*t*x+t
f3 =
x^3-6*x^2+11*x-6
f4 =
cos(x)*cos(y)-sin(x)*sin(y)
f5 =
4*x^3-3*x
f6 =
-6+(11+(-6+x)*x)*x
f7 =
11/10+(11/5+33/10*x)*x
f8 =
(x-1)*(x-2)*(x-3)
f9 =
x+1
f10 =
1
f11 =
((2*a+1)^3/a^3)^(1/3)
```

请读者观察其结果，说明所使用 MATLAB 函数 **collect** 等的意义和用法。

**[例 5]** 用 **findsym** 来确认例 10 中的符号表达式 f11 中的符号。

**解** 输入

```
>> findsym(f11)
```

运行后屏幕显示如下：

```
ans =
a
```

**[例 6]** 工具箱为我们提供了一个强有力的函数 **simple**，它综合运用上面的函数进行化简，并找出长度

最短的表达式。

(1) 请问 **simple** 命令常用的调用格式有几种？

(2) 对于 **simple** 命令这些调用格式，选择下面的函数在计算机上演示，并说明它们的功能。

(i)  $f(x)=\sin(x)^2+\cos(x)^2$ ;

(ii)  $g(x)=(1/a^3+6/a^2+12/a+8)^{(1/3)}$ ;

**解** (1) **simple** 命令常用调用格式有三种，即

**f=simple(S)**, **simple(S)**和**[f,how]=simple(S)**

(1) (i) **f=simple(S)**的功能是对表达式 S 进行化简，输出长度最短的表达式 f。

例如，输入

```
>> syms a x
f=simple(sin(x)^2+cos(x)^2)
g=simple((1/a^3+6/a^2+12/a+8)^(1/3))
```

运行后屏幕显示：

```
f =
1
g =
(2*a+1)/a
```

(ii) **simple(S)** 的功能是对表达式 S 进行化简，输出用各种函数化简的结果，及长度最短的表达式。

例如，输入

```
>> simple(sin(x)^2+cos(x)^2)
```

运行后屏幕显示：

```
simplify:
1
radsimp:
sin(x)^2+cos(x)^2
combine(trig):
1
factor:
sin(x)^2+cos(x)^2
expand:
sin(x)^2+cos(x)^2
combine:
1
convert(exp):
-1/4*(exp(i*x)-1/exp(i*x))^2+(1/2*exp(i*x)+1/2/exp(i*x))^2
convert(sincos):
sin(x)^2+cos(x)^2
convert(tan):
4*tan(1/2*x)^2/(1+tan(1/2*x)^2)^2+(1-tan(1/2*x)^2)^2/(1+tan(1/2*x)^2)^2
collect(x):
sin(x)^2+cos(x)^2
ans =
1
```

请读者在计算机上演示 g(x)。

(3) **[f,how]=simple(S)** 的功能是对表达式 S 进行化简，输出长度最短的表达式 f 及 f 是哪一个函数

作用的结果 how

例如, 输入

```
>> [f,how]=simple((1/a^3+6/a^2+12/a+8)^(1/3)),
```

```
[g,how]=simple(sin(x)^2+cos(x)^2)
```

运行后屏幕显示:

```
f =  
(2*a+1)/a  
how =  
radsimp  
g =  
1  
how =  
combine
```

**[例 7]** 工具箱中提供了两种**代换函数 subs 和 subexpr**。请分别写出它们的一种调用格式, 说明它们的功能, 并选择下面的函数在计算机上演示。

(1) 用 5 代替表达式  $a+b$  中的  $a$ ; (2) 分别用  $\alpha$  和 5 代替表达式中的  $a, b$ ;

(3) 首先用命令  $t = \text{solve}('a*x^3+b*x^2+c*x+d = 0')$  求方程的根  $t$ , 然后将求解的结果中的公共部分用  $s$  代换。

**解 (1) 代换函数 subs 的一种调用格式为**

**subs(S,old,new)**

它的功能是用符号  $new$  代替表达式  $S$  中的符号  $old$ 。

例如, 输入

```
>> syms a b,subs(a+b,a,5)
```

```
f=subs(cos(a)+sin(b),[a,b],[sym('alpha'),5])
```

运行后屏幕显示:

```
ans =  
5+b  
f =  
cos(alpha)+sin(5)
```

**(2) 代换函数 subexpr 的一种调用格式为**

**[Y,SIGMA] = subexpr (X,SIGMA)**

或**[Y,SIGMA] = subexpr (X,'SIGMA')**

它的功能是将表达式  $X$  中的公共部分用  $SIGMA$  表示。

例如, 输入

```
>> t = solve('a*x^3+b*x^2+c*x+d = 0'),
```

```
[r,s] = subexpr(t,'s')
```

运行后屏幕显示:

```
t =  
[1/6/a*(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a)^(1/3)-2/3*(3*c*a-b^2)/a/(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a)^(1/3)-1/3*b/a]  
[ -1/12/a*(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a)^(1/3)+1/3*(3*c*a-b^2)/a/(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a)^(1/3)-1/3*b/a+1/2*i*3^(1/2)*(1/6/a*(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a)^(1/3)-2/3*(3*c*a-b^2)/a/(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a)^(1/3)-1/3*b/a)]
```

$$2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a^(1/3)+2/3*(3*c*a-b^2)/a/(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a^(1/3))]$$

$$[-1/12/a*(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a^(1/3)+1/3*(3*c*a-b^2)/a/(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a^(1/3))-1/3*b/a-1/2*i*3^(1/2)*(1/6/a*(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a^(1/3)+2/3*(3*c*a-b^2)/a/(36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a^(1/3)))]$$

$$r =$$

$$[1/6/a*s^(1/3)-2/3*(3*c*a-b^2)/a/s^(1/3)-1/3*b/a]$$

$$[-1/12/a*s^(1/3)+1/3*(3*c*a-b^2)/a/s^(1/3)-1/3*b/a+1/2*i*3^(1/2)*(1/6/a*s^(1/3)+2/3*(3*c*a-b^2)/a/s^(1/3))]$$

$$[-1/12/a*s^(1/3)+1/3*(3*c*a-b^2)/a/s^(1/3)-1/3*b/a-1/2*i*3^(1/2)*(1/6/a*s^(1/3)+2/3*(3*c*a-b^2)/a/s^(1/3))]$$

$$s =$$

$$36*c*b*a-108*d*a^2-8*b^3+12*3^(1/2)*(4*c^3*a-c^2*b^2-18*c*b*a*d+27*d^2*a^2+4*d*b^3)^(1/2)*a$$

## 4.6 其它

工具箱中提供了 50 多个特殊函数, 如贝塞尔(Bessel)函数、椭圆积分、误差函数及切比雪夫(Chebyshev)正交多项式、拉格朗日正交多项式等。用 **mfunlist** 命令可以看到这些函数的列表, 用

**mhhelp <函数名>**

可以了解那个函数的细节。

工具箱对于数值计算提供了有理数计算方式和可变位数的浮点计算方式, 用法可从下例看出:  
输入

```
>> 《a=1/2+1/3,a1=sym(a),a2=vpa(a,10)
```

运行后屏幕显示:

```
a =
    0.8333
a1 =
    5/6
a2 =
    .8333333333
```