

# 第 1 章 基础准备及入门

本章有三个目的：一是讲述 MATLAB 正常运行所必须具备的基础条件；二是简明地介绍 MATLAB 及其操作桌面 Desktop 的基本使用方法；三是全面介绍 MATLAB 的帮助系统。

本章的前两节讲述：MATLAB 的正确安装方法和 MATLAB 环境的启动。因为指令窗是 MATLAB 最重要的操作界面，所以本章用第 1.3、1.4 两节以最简单通俗的叙述、算例讲述指令窗的基本操作方法和规则。这部分内容几乎对 MATLAB 各种版本都适用。第 1.5 到第 1.8 节专门介绍 MATLAB 最常用的另五个交互界面：历史指令窗、当前目录浏览器、工作空间浏览器、数组编辑器、M 文件编辑器。鉴于实际应用中，帮助信息和求助技能的重要性。本章专设第 1.9 节专门叙述 MATLAB 的帮助体系和求助方法。

作者建议：不管读者此前是否使用过 MATLAB，都不要忽略本章。

## 1.1 MATLAB 的安装和工具包选择

MATLAB 只有在适当的外部环境中才能正常运行。因此，恰当地配置外部系统是保证 MATLAB 运行良好的先决条件。MATLAB 本身可适应于许多机种和系统，如 PC 机和 Unix 工作站等。但本节只针对我国使用最广的 PC 机系统给予介绍。

对 PC 机用户来说，常常需要自己安装 MATLAB。MATLAB R2006a（即旧编号 MATLAB7.2）版要求 Win2000 或 WinXP 平台。下面介绍从光盘上安装 MATLAB 的方法。

- (1) 一般说来，当 MATLAB 光盘插入光驱后，会自启动“安装向导”。假如自启动没有实现，那么可以在<我的电脑>或<资源管理器>中双击 setup.exe 应用程序，使“安装向导”启动。安装过程中出现的所有界面都是标准的，用户只要按照屏幕提示操作，如输入用户名、单位名、口令等就行。
- (2) 在安装 MATLAB.R2006a 时，会出现一个界面，该界面上有两个选项：Typical 和 Custom。假如你不熟悉 MATLAB，或假如你机器的硬盘的自由空间远大于 3G，或假如你需要用到光盘上 MATLAB 的所有功能及工具包，那么你就点选“Typical”。否则，点选“Custom”。
- (3) 在点选“Custom”后，会引出如图 1.1-1 的界面。你可以根据需要，在“Select products to install”栏中勾选相应的组件。注意：MATLAB 软件光盘总包含很多工具包，它们有的是通用的，有的则专业性很强。对一般用户来说，完全不必采取全部安装，而应根据需要有所选择。否则将占据很多硬盘空间。表 1.1-1 对各组件的描述供用户选择时参考。

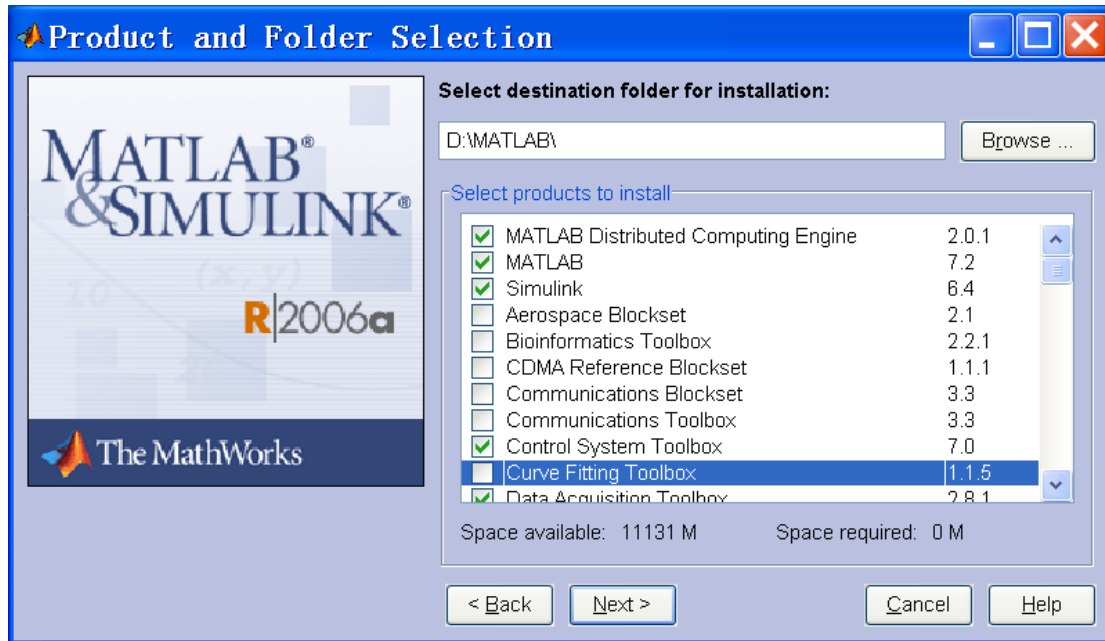


图 1.1-1 MATLAB 组件选择界面

表 1.1-1 MATLAB 各组件的功用

组 件 名 称	功 用
<b>1. 必须选择的本原性组件</b>	
MATLAB	这是最核心的部分。没有它就没有 MATLAB 环境。有了它，就可以对各类数据（除符号类数据以外）进行操作、运算和可视化。
<b>2. 最常选的通用性工具包组件</b>	
Symbolic Math Toolbox	符号类数据的操作和计算。
Extended Symbolic Math Toolbox	更丰富的符号计算函数和指令
<b>3. 其他通用性工具包组件</b>	
Simulink	不用编写程序，利用方块图实现建模和仿真；主要用于研究微分和差分方程描写的非线性动态系统。
Optimization Toolbox	包含求函数零点、极值、规划等优化程序。
Matlab Compiler	把 MATLAB 的 M 文件编译成独立应用程序。
Matlab builder for Excel	与 MATLAB Compiler 配合使用，生成 Excel 插件。
<b>4. 常用专业性工具包组件</b>	
Control System Toolbox	MATLAB 涉及控制的工具包有 10 多个，而本栏工具包是最基本的，是解决控制问题所必选的。至于其他控制工具包则是用户根据需要选择的。
Signal Processing Toolbox	是 MATLAB 信号处理中的基本工具包。
Spline Toolbox	内含样条和插值函数。
Statistics Toolbox	包含进行复杂统计分析所需的程序。
<b>5. 其他专业性工具包组件（举例）</b>	
Stateflow	与 Simulink 配合使用，主要用于较大型、复杂的（离散事件）动态系统的建模、分析、仿真。

System Identification Toolbox	动态系统辨识。
.....	


## 1.2 Desktop 操作桌面的启动

### 1.2.1 MATLAB的启动


#### (1) 方法一

当 MATLAB 安装到硬盘上以后，一般会在 Windows 桌面上自动生成 MATLAB 程序图标。在这种情况下，只要直接点击那图标即可启动 MATLAB，打开如图 1.2-1 的 MATLAB 操作桌面 (Desktop)。注意：本书作者建议用户优先采用启动“方法一”。

#### (2) 方法二

假如 Windows 桌面上没有 MATLAB 图标，那么点击 matlab\ 文件夹下的快捷方式图标  MATLAB。

#### (3) 方法三

点击 matlab\bin\win32 文件夹中的 matlab.exe (它的图标是 )，也会自动创建类似图 1.2-1 所示的 Desktop 操作桌面。唯一的区别是：采用这种方式创建的 MATLAB 工作环境以 MATLAB 所在的根目录为当前目录。

### 1.2.2 Desktop操作桌面简介

MATLAB R2006a 版的 Desktop 操作桌面，是一个高度集成的 MATLAB 工作界面。其默认形式，如图 1.2-1 所示。该桌面的上层铺放着三个最常用的界面：指令窗 (Command Window)、当前目录 (Current Directory) 浏览器、历史指令 (Command History) 窗。在当前目录窗的下面还铺放一个 MATLAB 内存工作空间 (Workspace) 浏览器。

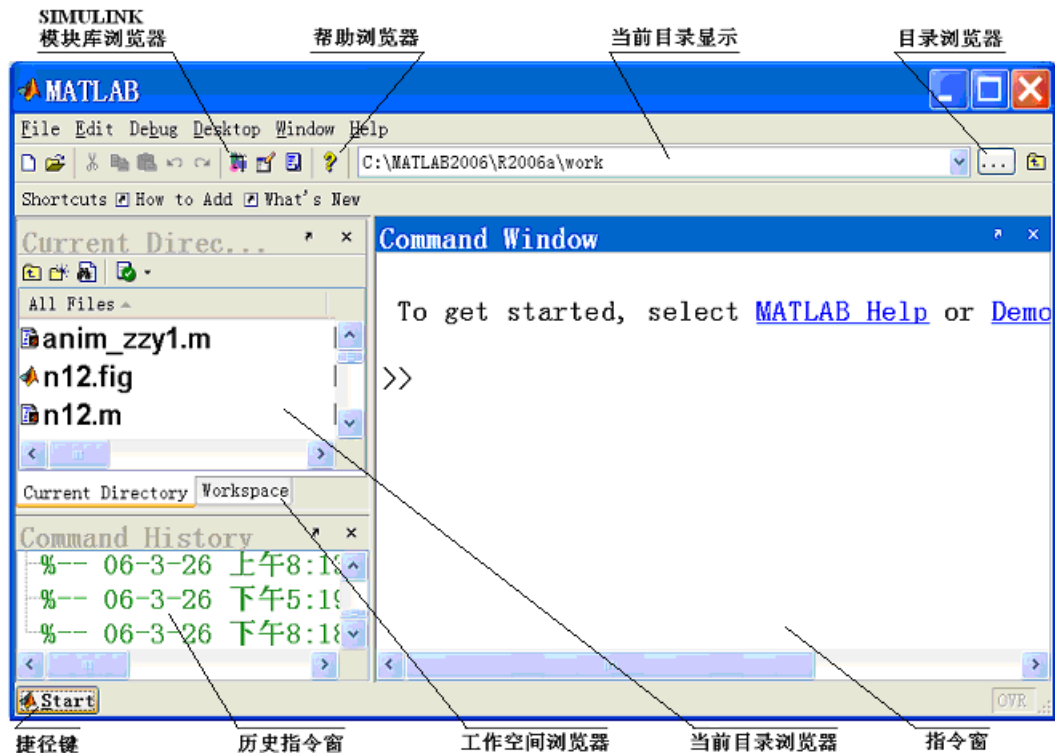


图 1.2-1 Desktop 操作桌面的默认外貌


- 指令窗  
该窗是进行各种 MATLAB 操作的最主要窗口。在该窗内，可键入各种送给 MATLAB 运作的指令、函数、表达式；显示除图形外的所有运算结果；运行错误时，给出相关的出错提示。
- 历史指令窗  
该窗记录已经运作过的指令、函数、表达式，及它们运行的日期、时间。该窗中的所有指令、文字都允许复制、重运行及用于产生 M 文件。
- 当前目录浏览器  
在该浏览器中，展示着子目录、M 文件、MAT 文件和 MDL 文件等。对该界面上的 M 文件，可直接进行复制、编辑和运行；界面上的 MAT 数据文件，可直接送入 MATLAB 工作内存。此外，对该界面上的子目录，可进行 Windows 平台的各种标准操作。
- 工作空间浏览器  
该浏览器默认地处于当前目录浏览器的后台。该窗口罗列出 MATLAB 工作空间中所有的变量名、大小、字节数；在该窗中，可对变量进行观察、图示、编辑、提取和保存。
- 捷径 (Start) 键  
引出通往本 MATLAB 所包含的各种组件、模块库、图形用户界面、帮助分类目录、演示算例等的捷径，以及向用户提供自建快捷操作的环境。

## 1.3 Command Window 运行入门

MATLAB 的使用方法和界面有多种形式。但最基本的，也是入门时首先要掌握的是：MATLAB 指令窗 (Command Window) 的基本表现形态和操作方式。本书作者相信，通过

本节的文字解释，读者将对 MATLAB 使用方法有一个良好的初始感受。

### 1.3.1 Command Window指令窗简介

MATLAB 指令窗默认地位于 MATLAB 桌面的右方（见图 1.2-1）。假如，用户希望得到脱离操作桌面的几何独立指令窗，只要点击该指令窗右上角的  键，就可获得如图 1.3-1 所示的指令窗。

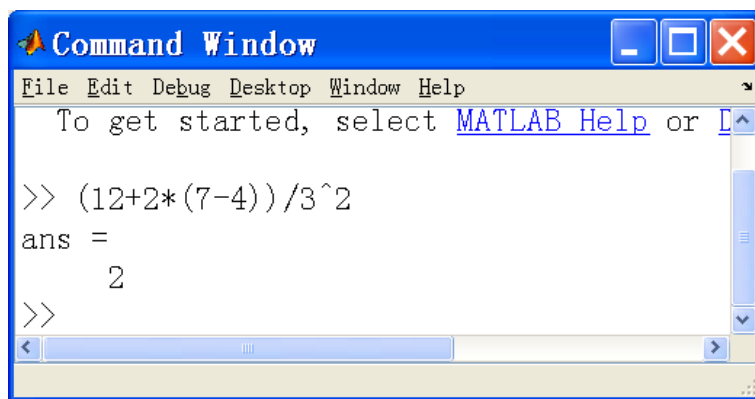



图 1.3-1 几何独立的指令窗

【说明】

- 图 1.3-1 指令窗表现了例 1.3-1 运行的情况。
- 若用户希望让独立指令窗嵌放回桌面，则只要点击 Command Window 右上角的  按钮，或选中指令窗菜单 {Desktop: Dock Command Window} 便可。

### 1.3.2 最简单的计算器使用法

为易于学习，本节以算例方式叙述，并通过算例归纳一些 MATLAB 最基本的规则和语法结构。建议读者，在深入学习之前，先读一读本节。

【例 1.3-1】求  $[12 + 2 \times (7 - 4)] \div 3^2$  的算术运算结果。本例演示：最初步的指令输入形式和必需的操作步骤。

（1）用键盘在 MATLAB 指令窗中输入以下内容

```
>> (12+2*(7-4))/3^2
```

（2）在上述表达式输入完成后，按 [Enter] 键，该指令被执行，并显示如下结果。

```
ans =  
2
```

【说明】

- 本例在指令窗中实际运行的情况参见图 1.3-1。
- 指令行“头首”的“>>”是“指令输入提示符”，它是自动生成的。本书在此后的输入指令前将不再带提示符“>>”。理由是：（A）为使本书简洁；（B）本书用 MATLAB 的 M-book 写成，而在 M-book 中运行的指令前是没有提示符的。

- MATLAB 的运算符（如+、- 等）都是各种计算程序中常见的习惯符号。
- 一条指令输入结束后，必须按 [Enter] 键，那指令才被执行。
- 由于本例输入指令是“不含赋值号的表达式”，所以计算结果被赋给 MATLAB 的一个默认变量“ans”。它是英文“answer”的缩写。

【例 1.3-2】“续行输入”法。本例演示：或由于指令太长，或出于某种需要，输入指令行必须多行书写时，该如何处理。

```
S=1-1/2+1/3-1/4+ ...
1/5-1/6+1/7-1/8
```

```
S =
    0.6345
```

【说明】

- MATLAB 用 3 个或 3 个以上的连续黑点表示“续行”，即表示下一行是上一行的继续。
- 本例指令中包含“赋值号”，因此表达式的计算结果被赋给了变量 S。
- 指令执行后，变量 S 被保存在 MATLAB 的工作空间（Workspace）中，以备后用。如果用户不用 clear 指令清除它，或对它重新赋值，那么该变量会一直保存在工作空间中，直到本 MATLAB 指令窗被关闭为止。

### 1.3.3 数值、变量和表达式

前节算例只是表演了“计算器”功能，那仅是 MATLAB 全部功能中小小一角。为深入学习 MATLAB，有必要系统介绍一些基本规定。本节先介绍关于变量的若干规定。

#### 一 数值的记述

MATLAB 的数值采用习惯的十进制表示，可以带小数点或负号。以下记述都合法。

```
3      -99      0.001      9.456      1.3e-3      4.5e33
```

在采用 IEEE 浮点算法的计算机上，数值通常采用“占用 64 位内存的双精度”表示。其相对精度是 eps（MATLAB 的一个预定义变量），大约保持有效数字 16 位。数值范围大致从  $10^{-308}$  到  $10^{308}$ 。

#### 二 变量命名规则

- **变量名、函数名是对字母大小写敏感的。**如变量 myvar 和 MyVar 表示两个不同的变量。sin 是 MATLAB 定义的正弦函数名，但 SIN,Sin 等都不是。
- **变量名的第一个字符必须是英文字母，最多可包含 63 个字符（英文、数字和下连符）。**如 myvar201 是合法的变量名。
- **变量名中不得包含空格、标点、运算符，但可以包含下连符。**如变量名 my\_var\_201 是合法的，且读起来更方便。而 my,var201 由于逗号的分隔，表示的就不是一个变量名。

### 三 MATLAB 默认的预定义变量

在 MATLAB 中有一些所谓的预定义变量（Predefined Variable），见表 1.3-1。每当 MATLAB 启动，这些变量就被产生。这些变量都有特殊含义和用途。建议：用户在编写指令和程序时，应尽可能不对表 1.3-1 所列预定义变量名重新赋值，以免产生混淆。

表 1.3-1 MATLAB 中最常用的预定义变量

预定义变量	含 义	预定义变量	含 义
<b>ans</b>	计算结果的默认变量名	<b>NaN 或 nan</b>	不是一个数（Not a Number），如 0/0， $\infty/\infty$
<b>eps</b>	机器零阈值	<b>nargin</b>	函数输入宗量数目
<b>Inf 或 inf</b>	无穷大，如 1/0	<b>nargout</b>	函数输出宗量数目
<b>i 或 j</b>	虚单元 $i = j = \sqrt{-1}$	<b>realmax</b>	最大正实数
<b>pi</b>	圆周率 $\pi$	<b>realmin</b>	最小正实数

【说明】

- 假如用户对表中任何一个预定义变量中进行赋值，则那个变量的默认值将被用户新赋的值“临时”覆盖。所谓“临时”是指：假如使用 **clear** 指令清除 MATLAB 内存中的变量，或 MATLAB 指令窗被关闭后重新启动，那么所有的预定义变量将被重置为默认值，不管这些预定义变量曾被用户赋过什么值。
- 在遵循 IEEE 算法规则的机器上，被 0 除是允许的。它不会导致程序执行的中断，只是在给出警告信息的同时，用一个特殊名称（如 **Inf**，**NaN**）记述。这个特殊名称将在以后的计算中以合理的形式发挥作用。

【例 1.3-3】运用以下指令，以便初步了解预定义变量。本例演示：预定义变量已经存在的事实；若干预定义变量的数量级概念。

```
format long
realmax
ans =
    1.797693134862316e+308

realmin
ans =
    2.225073858507201e-308

eps
ans =
    2.220446049250313e-016

pi
ans =
    3.14159265358979
```

## 四 运算符和表达式

(1) 经典教科书上的算术运算符在 MATLAB 中的表达方式，见表 1.3-2。

表 1.3-2 MATLAB 表达式的基本运算符

	数学表达式	矩阵运算符	数组运算符
加	$a + b$	$a + b$	$a + b$
减	$a - b$	$a - b$	$a - b$
乘	$a \times b$	$a * b$	$a .* b$
除	$a \div b$	$a / b$ 或 $b \setminus a$	$a ./ b$ 或 $b .\setminus a$
幂	$a^b$	$a ^ b$	$a .^ b$
圆括号	( )	( )	( )

【说明】

- 因为 MATLAB 面向复数设计，其所有运算定义在复数域上。所以对于方根问题，运算只返还一个“主解”。要得复数的全部方根，必须专门编写程序（见例 1.3-6）。
- 因为 MATLAB 面向矩阵/数组设计，标量被看作  $(1 \times 1)$  的矩阵/数组。
- 数组运算的“乘、除、幂”规则与相应矩阵运算根本不同。前者的算符比后者多一个“小黑点”。（参见例 1.3-9，例 1.3-10。更详细说明请看第 3 章）
- MATLAB 用左斜杠或右斜杠分别表示“左除”或“右除”运算。对标量而言，“左除”和“右除”的作用结果相同。但对矩阵来说，“左除”和“右除”将产生不同的结果。

(2) MATLAB 书写表达式的规则与“手写算式”几乎完全相同。

- 表达式由变量名、运算符和函数名组成。
- 表达式将按与常规相同的优先级自左至右执行运算。
- 优先级的规定是：指数运算级别最高，乘除运算次之，加减运算级别最低。
- 括号可以改变运算的次序。
- 书写表达式时，赋值符“=”和运算符两侧允许有空格，以增加可读性。

## 五 面向复数设计的运算——MATLAB 特点之一

MATLAB 的所有运算都是定义在复数域上的。这样设计的好处是：在进行运算时，不必像其他程序语言那样把实部、虚部分开处理。为描述复数，虚数单位用预定义变量  $i$  或  $j$  表示。

复数  $z = a + bi = re^{i\theta}$  直角坐标表示和极坐标表示之间转换的 MATLAB 指令如下。

**real(z)** 给出复数  $z$  的实部  $a = r \cos \theta$ 。

**imag(z)** 给出复数  $z$  的虚部  $b = r \sin \theta$ 。

**abs(z)** 给出复数  $z$  的模  $\sqrt{a^2 + b^2}$ 。

**angle(z)** 以弧度为单位给出复数  $z$  的幅角  $\arctan \frac{b}{a}$ 。



【例 1.3-4】复数  $z_1 = 3 + 4i$ ,  $z_2 = 1 + 2i$ ,  $z_3 = 2e^{\frac{\pi}{6}i}$  表达, 及计算  $z = \frac{z_1 z_2}{z_3}$ 。本例演示: 正

确的复数输入法; 涉及复数表示方式的基本指令。

(1) 经典教科书的直角坐标表示法

```
z1= 4 + 3i           %合法, 但建议少用或不用
z1 =
    4.0000 + 3.0000i
```

〔说明〕

- 本书建议读者不要使用这种输入格式。因为这种书写格式, 只适用于“数值标量”复数, 而不适用于“数值矩阵”。
- 在这种书写格式中,  $4i$  是一个完整的虚数, 在 4 和  $i$  之间不许“空格”存在。

(2) 采用运算符构成的直角坐标表示法和极坐标表示法

```
z2 = 1 + 2 * i       %运算符构成的直角坐标表示法
z3=2*exp(i*pi/6)     %运算符构成的极坐标表示法
z=z1*z2/z3
z2 =
    1.0000 + 2.0000i
z3 =
    1.7321 + 1.0000i
z =
    1.8840 + 5.2631i
```

(3) 复数的实虚部、模和幅角计算

```
real_z=real(z)
image_z=imag(z)
magnitude_z=abs(z)
angle_z_radian=angle(z)      %弧度单位
angle_z_degree=angle(z)*180/pi %度数单位
real_z =
    1.8840
image_z =
    5.2631
magnitude_z =
    5.5902
angle_z_radian =
    1.2271
angle_z_degree =
    70.3048
```

【例 1.3-5】图示复数  $z_1 = 4 + 3i$ ,  $z_2 = 1 + 2i$  的和 (配图 1.3-2)。本例演示: MATLAB 的运

算在复数域上进行；指令后“分号”的作用；复数加法的几何意义；展示 MATLAB 的可视化能力（让读者感受，但不要求理解）。

```
z1=4+3*i;z2=1+2*i; %在一个物理行中，允许输入多条指令。  
%但各指令间要用“分号”或“逗号”分开。  
%指令后采用“分号”，使运算结果不显示。
```

```
z12=z1+z2  
%以下用于绘图  
clf,hold on %clf 清空图形窗。逗号用来分隔两个指令。  
plot([0,z1,z12],'-b','LineWidth',3)  
plot([0,z12],'-r','LineWidth',3)  
plot([z1,z12],'ob','MarkerSize',8)  
hold off,grid on,  
axis equal  
axis([0,6,0,6])  
text(3.5,2.3,'z1')  
text(5,4.5,'z2')  
text(2.5,3.5,'z12')  
xlabel('real')  
ylabel('image')  
shg
```

```
z12 =  
5.0000 + 5.0000i
```

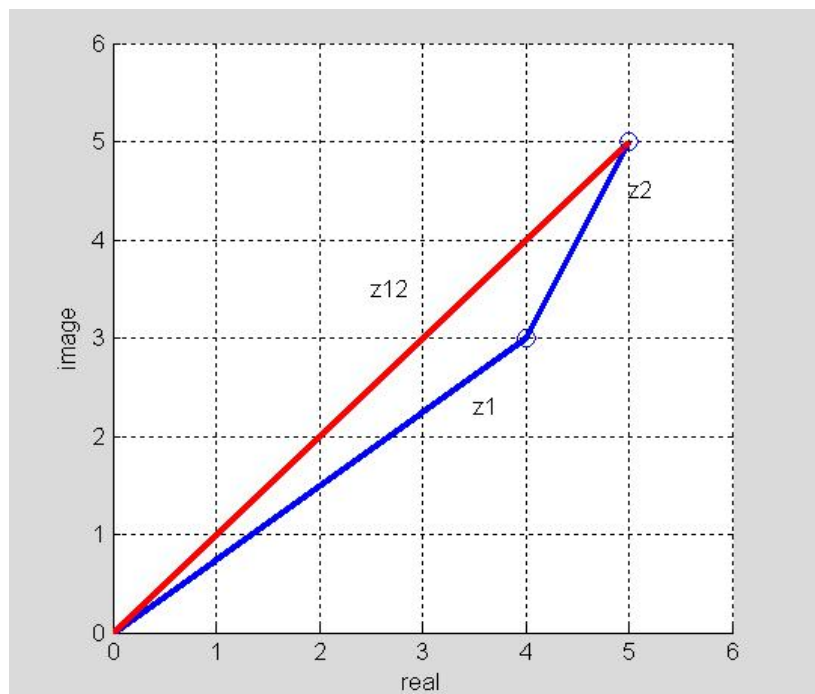


图 1.3-2 两个复数相加

【例 1.3-6】用 MATLAB 计算  $\sqrt[3]{-8}$  能得到  $-2$  吗（配图 1.3-3）？本例演示：MATLAB 运

算定义在复数域的实质；指令后“分号”抑制运算结果的显示；MATLAB 的方根运算规则；更复杂指令的表示方式；展现 MATLAB 的图形表现力。（对于本例指令，读者能有体验就可，不必强求理解。）

(1) 直接计算时，得到处于第一象限的方根。

```
a=-8;
r_a=a^(1/3)    %求 3 次根
r_a =
    1.0000 + 1.7321i
```

(2)  $\sqrt[3]{-8}$  的全部方根计算如下

% 先构造一个多项式  $p(r) = r^3 - a$

```
p=[1,0,0,-a]; %p 是多项式  $p(r)$  的系数向量
```

%指令末尾的“英文状态分号”使该指令运行后，不显示结果。

```
R=roots(p)    %求多项式的根
R =
   -2.0000
    1.0000 + 1.7321i
    1.0000 - 1.7321i
```

(3) 图形表示

```
MR=abs(R(1));    %计算复根的模
t=0:pi/20:2*pi;  %产生参变量在 0 到 2*pi 间的一组采样点
x=MR*sin(t);
y=MR*cos(t);
plot(x,y,'b:'),grid on    %画一个半径为 R 的圆
                           %注意“英文状态逗号”在不同位置的作用

hold on
plot(R(2),'.','MarkerSize',30,'Color','r')    %画第一象限的方根
plot(R([1,3]),'o','MarkerSize',15,'Color','b')%画另两个方根
axis([-3,3,-3,3]),axis square    %保证屏幕显示呈真圆
hold off
```

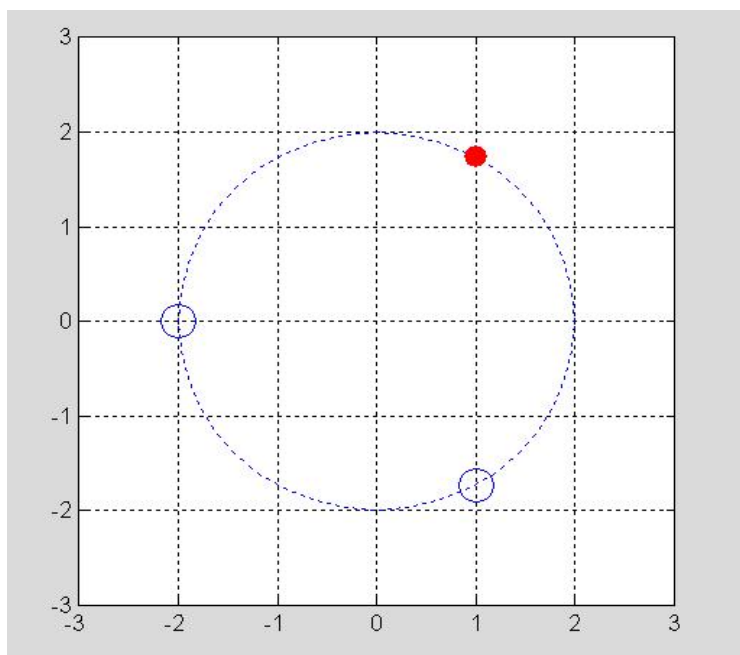


图 1.3-3  $(-8)$  的全部三次方根分布

【说明】

- 本例有助于理解 MATLAB 的计算特点。
- 对复数进行方根运算时，MATLAB 只给出处于“第一象限”的那个根。

## 六 面向数组设计的运算——MATLAB 特点之二

在 MATLAB 中，标量数据被看作  $(1 \times 1)$  的数组（Array）数据。所有的数据都被存放在适当大小的数组中。为加快计算速度（运算的向量化处理），MATLAB 对以数组形式存储的数据设计了两种基本运算：一种是所谓的数组运算；另一种是所谓的矩阵运算。在此仅以算例展示 MATLAB 的计算特点，更详细的叙述请见第 3 章。

【例 1.3-7】实数数组  $\mathbf{AR} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$  的“一行”输入法。本例演示：二维数组的最基本、最

常用输入法；二维数组输入的三大要素。

（1）在键盘上输入下列内容

**AR= [1,3;2,4]**

（2）按 [Enter] 键，指令被执行。

（3）在指令执行后，MATLAB 指令窗中将显示以下结果：

```
AR =
     1     3
     2     4
```

【说明】

- 在 MATLAB 中，不必事先对数组维数及大小做任何说明，内存将自动配置。
- 二维数组输入的三大要素：数组标识符 “[ ]”；元素分隔符空格或逗号“,”；数组行间分隔符分号“;”或“回车键”。注意：所有标点符号都是“英文状态的符号”。

- MATLAB 对字母大小写是敏感的。比如本例中的数组赋给了变量 AR，而不是 Ar, aR, 或 ar。
- 在全部键入一个指令行内容后，必须按下 [Enter] 键，该指令才会被执行。请读者务必记住此点。出于叙述简明的考虑，本书此后将不再重复提及此操作。

【例 1.3-8】实数数组  $\mathbf{AI} = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}$  的“分行”输入法。

```
AI=[5,7
    6,8]
AI =
     5     7
     6     8
```

【说明】

- 本例采用这种输入法是为了视觉习惯。当然，对于较大的数组也可采用此法。
- 在这种输入方法中，“回车”符用来分隔数组中的行。

【例 1.3-9】对复数数组  $\mathbf{A} = \begin{bmatrix} 1-5i & 3-7i \\ 2-6i & 4-8i \end{bmatrix}$  进行求实部、虚部、模和幅角的运算。本例演

示：复数数组的生成；MATLAB 指令对数组元素“并行操作”的实质。

(1) 创建复数数组

```
AR=[1,3;2,4];AI=[5,7;6,8];
A=AR-AI*i                      %形成复数矩阵
A =
    1.0000 - 5.0000i    3.0000 - 7.0000i
    2.0000 - 6.0000i    4.0000 - 8.0000i
```

(2) 求复数数组的实部和虚部

```
A_real=real(A)
A_image=imag(A)
A_real =
     1     3
     2     4
A_image =
    -5    -7
    -6    -8
```

(3) 求复数数组中各元素的模和幅角——循环法（笨拙!）

```
for m=1:2
    for n=1:2
        Am1(m,n)=abs(A(m,n));
```

```

        Aa1(m,n)=angle(A(m,n))*180/pi; %以度为单位计算幅角
    end
end
Am1,Aa1
Am1 =
    5.0990    7.6158
    6.3246    8.9443
Aa1 =
   -78.6901  -66.8014
   -71.5651  -63.4349

```

(4) 求复数数组中各元素的模和幅角——直接法

```

Am2=abs(A)
Aa2=angle(A)*180/pi
Am2 =
    5.0990    7.6158
    6.3246    8.9443
Aa2 =
   -78.6901  -66.8014
   -71.5651  -63.4349

```

【说明】

- 函数 `real`, `imag`, `abs`, `angle` 是同时、并行地作用于数组的每个元素。对 4 个元素运算所需的时间大致与对单个元素所需时间相同。这有利于运算速度的提高。这是“向量化”运算的一种形式。
- 本例给出了循环法求各元素模和幅角的指令。这是很不有效的计算方法。对于 MATLAB 以外的许多编程语言来说，可能不得不采用“循环”处理方式来解决本例。记住：对于 MATLAB 来说，应该尽量摒弃“循环”处理，而采用“向量化”处理方式。

【例 1.3-10】画出衰减振荡曲线  $y = e^{-\frac{t}{3}} \sin 3t$ ， $t$  的取值范围是  $[0, 4\pi]$ （配图 1.3-4）。本例演示：展示数组运算的优点；展示 MATLAB 的可视化能力。

```

t=0:pi/50:4*pi;           %定义自变量 t 的取值数组
y=exp(-t/3).*sin(3*t);     %计算与自变量相应的 y 数组。注意：乘法符前面的小黑点。
plot(t,y,'-r','LineWidth',2) %绘制曲线
axis([0,4*pi,-1,1])
xlabel('t'),ylabel('y')

```

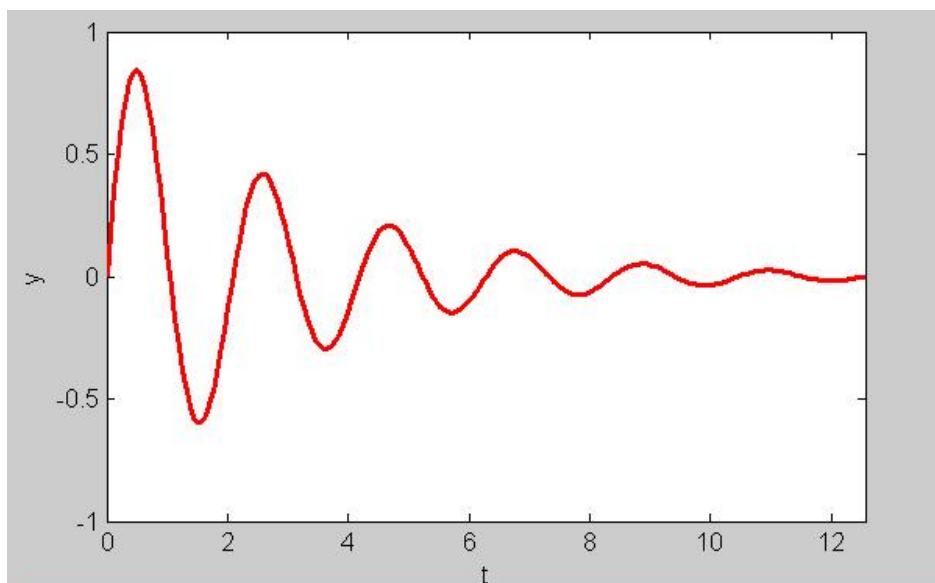


图 1.3-4 衰减振荡曲线

【说明】

- 本例第二条指令中的“.”符号表示乘法是在两个数组相同位置上的元素间进行的。本书把这种乘法称为“数组乘”。数组乘的引入，不但使得程序简洁自然，而且避免了耗费机时的“循环计算”。关于数组运算的详细叙述请见第3章。
- 本例第二条指令是典型的“向量化”处理形式。本书作者建议读者，只要可能，应尽量采用“向量化”运算形式。

【例 1.3-11】复数矩阵  $\mathbf{B} = \begin{bmatrix} 3+2i & 2+6i \\ 5+3i & 4-2i \end{bmatrix}$  的生成，及计算  $\mathbf{A} \cdot \mathbf{B}$  矩阵乘积（A 取自算例

1.3-9）。本例演示：MATLAB 矩阵运算指令的简捷性。

```
B=[3+2i,2+6i;5+3*i,4-2*i] %复数数组的又一种输入方式
%注意标点符号的作用
C=A*B %矩阵乘法
B =
    3.0000 + 2.0000i    2.0000 + 6.0000i
    5.0000 + 3.0000i    4.0000 - 2.0000i
C =
    49.0000 -39.0000i    30.0000 -38.0000i
    62.0000 -42.0000i    40.0000 -40.0000i
```

【说明】

- 当数组被赋予“变换”属性时，二维数组就被称为矩阵。只有当两个矩阵的“内维大小相等”时，矩阵乘法才能进行。本例中，矩阵 A 的列数与矩阵 B 的行数相等，所以可以进行 A 乘 B。
- 从表达方式看，“矩阵相乘”的指令格式与“标量相乘”指令格式一样。在其他编程语言中，矩阵乘法不得不依赖“循环”进行。
- MATLAB 之所以能把矩阵运算表达得像标准“线性代数”那样简洁易读、自然流畅，那是由于 MATLAB 的设计者采用了“面向对象”编程技术。

## 1.4 Command Window操作要旨

前一节借助算例，使读者对 MATLAB 指令窗的使用方法有了一个直观的感受。本节将在上节的基础上对控制指令窗的指令和操作进行较系统的归纳，以便读者更全面地了解 MATLAB，更方便地使用 MATLAB。

### 1.4.1 指令窗的显示方式

#### 一 默认的输出显示方式

从 MATLAB7.0 起，指令窗中的字符、数值等采用更为醒目的分类显示：

- (1) 对于输入指令中的 if, for, end 等控制数据流的 MATLAB 关键词自动地采用蓝色字体显示。
- (2) 对于输入指令中的非控制指令、数码，都自动地采用黑色字体显示。
- (3) 输入的字符串自动呈现为紫色字体。

#### 二 运算结果的显示

在指令窗中显示的输出有：指令执行后，数值结果采用黑色字体输出；而运行过程中的警告信息和出错信息用红色字体显示。

运行中，屏幕上最常见到的数字输出结果由 5 位数字构成。这是“双精度”数据的默认输出格式。用户不要误认为，运算结果的精度只有 5 位有效数字。实际上，MATLAB 的数值数据通常占用 64 位 (Bit) 内存，以 16 位有效数字的“双精度”进行运算和输出。MATLAB 为了比较简洁、紧凑地显示数值输出，才默认地采用 **format short g** 格式显示出 5 位有效数字。用户根据需要，可以在 MATLAB 指令窗中，直接输入相应的指令，或者在菜单弹出框中进行选择，都可获得所需的数值计算结果显示格式。MATLAB 数值计算结果显示格式的类型见表 1.4-1。

表 1.4-1 数据显示格式的控制指令

指 令	含 义	举 例 说 明
<b>format</b> <b>format short</b>	通常保证小数点后四位有效，最多不超过 7 位；对于大于 1000 的实数，用 5 位有效数字的科学记数形式显示。	314.159 被显示为 314.1590； 3141.59 被显示为 3.1416e+003
<b>format long</b>	15 位数字表示	3.14159265358979
<b>format short e</b>	5 位科学记数表示	3.1416e+00
<b>format long e</b>	15 位科学记数表示	3.14159265358979e+00
<b>format short g</b>	从 format short 和 format short e 中自动选择最佳记数方式	3.1416
<b>format long g</b>	从 format long 和 format long e 中自动选择最佳记数方式	3.14159265358979
<b>format rat</b>	近似有理数表示	355/113
<b>format hex</b>	十六进制表示	400921fb54442d18
<b>format +</b>	显示大矩阵用。正数、负数、零 分别用 +	+



	+, -, 空格表示。	
<b>format bank</b>	(金融) 元、角、分表示	3.14
<b>format compact</b>	显示变量之间没有空行	
<b>format loose</b>	在显示变量之间有空行	
【说明】 <ul style="list-style-type: none"> <li>● format short 显示格式是默认显示格式。</li> <li>● 该表中实现的所有格式设置仅在 MATLAB 的当前执行过程中有效。</li> </ul>		

### 三 显示方式的永久设置

用户根据需要，可以对指令窗的字体风格、大小、颜色和数值计算结果显示格式进行设置。设置方法是：选中 {File: Preferences} 下拉菜单项，引出一个参数设置对话框；在此弹出对话框的左栏选中“Font & Colors”，对话框的右边就出现相应的选择内容；用户根据需要和对话框提示对数据显示格式，或字体等进行选择；最后，通过点击 [OK] 键，便完成了设置。注意：该设置立即生效，并且这种设置将被永久保留，即这种设置不因 MATLAB 关闭和开启而改变，除非用户进行重新设置。

在此还要指出，对于数值显示格式的设置，也可以直接在指令窗中，通过指令的运作进行。但这样的设置仅对当前的 MATLAB 指令窗起作用，一旦 MATLAB 关闭，这种设置也就随之失效。

#### 1.4.2 指令行中的标点符号

通过前面算例，读者可能已对标点符号的作用有所体会。在此要强调指出：标点符号在 MATLAB 中的地位极其重要。为此，把各标点的作用归纳成表 1.4-2。

表 1.4-2 MATLAB 常用标点符号的功能

名 称	标 点	作 用
空格		(为机器辨认) 用作输入量与输入量之间的分隔符； 数组元素分隔符。
逗号	,	用作要显示计算结果的指令与其后指令之间的分隔； 用作输入量与输入量之间的分隔符； 用作数组元素分隔符号。
黑点	.	数值表示中，用作小数点； 用于运算符前，构成“数组”运算符。
分号	;	用于指令的“结尾”，抑制计算结果的显示； 用作不显示计算结果指令与其后指令的分隔； 用作数组的行间分隔符。
冒号	:	用以生成一维数值数组； 用做单下标索引时，表示全部元素构成的长列； 用做多下标索引时，表示那维上的全部元素。
注释号	%	由它“启首”的所有物理行部分被看作非执行的注释
单引号对	' '	字符串记述符
圆括号	( )	改变运算次序； 在数组索引时用；



		函数指令输入宗量列表时用。
方括号	[ ]	输入数组时用； 函数指令输出宗量列表时用。
花括号	{ }	胞元数组记述符； 图形中被控特殊字符括号。
下连符	-	（为使人易读）用作一个变量、函数或文件名中的连字符； 图形中被控下脚标前导符。
续行号	...	由三个以上连续黑点构成。它把其下的物理行看作该行的“逻辑”继续，以构成一个“较长”的完整指令。
“@”号	@	放在函数名前，形成函数句柄； 匿名函数前导符； 放在目录名前，形成“用户对象”类目录。
【说明】为确保指令正确执行，以上符号一定要在英文状态下输入。因为 MATLAB 不能识别含有中文标点的指令。		

### 1.4.3 指令窗的常用控制指令

表 1.4-3 常见的通用操作指令

指令	含 义	指 令	含 义
<b>cd</b>	设置当前工作目录。	<b>exit</b>	关闭/退出 MATLAB
<b>clf</b>	清除图形窗	<b>quit</b>	关闭/退出 MATLAB
<b>clc</b>	清除指令窗中显示内容	<b>more</b>	使其后的显示内容分页进行
<b>clear</b>	清除 MATLAB 工作空间中保存的变量	<b>return</b>	返回到上层调用程序；结束键盘模式
<b>dir</b>	列出指定目录下的文件和子目录清单	<b>type</b>	显示指定 M 文件的内容
<b>edit</b>	打开 M 文件编辑器	<b>which</b>	指出其后文件所在的目录

【说明】

- 表 1.4-3 所列的指令是基本的，它们对 MATLAB 各版都适用。
- 尽管随版本的升级，不断增添着列表中指令的“等价”菜单选项操作或工具条图标操作，但这种“等价”仅对“人机交互”过程而言。至于这些指令在 M 文件中的作用仍是不可替代的。
- **cd** 及 **dir** 指令的操作响应，可以用 MATLAB 操作桌面上或当前目录浏览器中的“浏览键（Browser）”替代。关于当前目录浏览器的使用，请看第 1.6 节。
- **clear** 清除内存变量的操作，可以等价地在“工作空间浏览器”交互界面中实现。关于工作空间浏览器的使用，请看第 1.7 节。
- **edit** 指令的等价操作是：选择 MATLAB 操作桌面或指令窗的下拉菜单项 {File: New: M-file}，或点击相应工具条上的  和  图标。关于 M 文件编辑器的使用，请看第 1.9 节。

### 1.4.4 指令窗中指令行的编辑

为了操作方便，MATLAB 不但允许用户在指令窗中对输入的指令行进行各种编辑和运行，而且允许用户对过去已经输入的指令行进行回调、编辑和重运行。具体的操作方式见表 1.4-4。

表 1.4-4 MATLAB 指令窗中实施指令行编辑的常用操作键

键 名	作 用	键 名	作 用
↑	前寻式调回已输入过的指令行	Home	使光标移到当前行的首端
↓	后寻式调回已输入过的指令行	End	使光标移到当前行的尾端
←	在当前行中左移光标	Delete	删去光标右边的字符
→	在当前行中右移光标	Backspace	删去光标左边的字符
PageUp	前寻式翻阅当前窗中的内容	Esc	清除当前行的全部内容
PageDown	后寻式翻阅当前窗中的内容		

【说明】

- 表 1.4-4 所列的操作对 MATLAB 各版均适用。
- 事实上，MATLAB 把指令窗中输入的所有指令都记录在内存中专门开辟的“指令历史空间（Command History）”中，只要用户对它们不进行专门的删除操作，它们既不会因为用户对指令窗进行“清屏”操作（即运行 `clc` 指令）而消失，也不会因用户对“工作空间”进行“清除内存变量”（即运行 `clear` 指令）而消失。
- 指令窗中输入过的所有指令都被显示在“历史指令浏览器”交互界面中，以供随时观察和调用。关于“历史指令浏览器”的使用请看第 1.5 节。

【例 1.4-1】指令行操作过程示例。

(1) 若用户想计算  $y_1 = \frac{2\sin(0.3\pi)}{1+\sqrt{5}}$  的值，那么用户应依次键入以下字符

```
y1=2*sin(0.3*pi)/(1+sqrt(5))
```

(2) 按 [Enter] 键，该指令便被执行，并给出以下结果

```
y1 =  
0.5000
```

(3) 通过反复按键盘的箭头键，可实现指令回调和编辑，进行新的计算。

若又想计算  $y_2 = \frac{2\cos(0.3\pi)}{1+\sqrt{5}}$ ，用户当然可以像前一个算例那样，通过键盘把相应字符一个一个“敲入”。

但也可以较方便地用操作键获得该指令，具体办法是：先用 [↑] 键调回已输入过的指令 `y1=2*sin(0.3*pi)/(1+sqrt(5))`；然后移动光标，把 `y1` 改成 `y2`；把 `sin` 改成 `cos`；再按 [Enter] 键，就可得到结果。即

```
y2=2*cos(0.3*pi)/(1+sqrt(5))  
y2 =  
0.3633
```

【说明】

- 可以借助“历史指令窗”进行历史指令的再运行，相关内容请看第 1.5.1 节。

## 1.5 Command History历史指令窗

MATLAB 所拥有的丰富资源和友善灵活的环境特别适于用来验证一些思想，思考一些问题，和帮助进行创造性思维。用户可以在 MATLAB 环境中，边想边做，做做想想，对随时蹦出的思想“火花”可即刻通过计算加以验证。历史指令窗（Command History）就是为这种应用方式设计的。

### 1.5.1 Command History历史指令窗简介

历史指令窗记录着：每次开启 MATLAB 的时间，及开启 MATLAB 后在指令窗中运行过的所有指令行。该窗不但能清楚地显示指令窗中运行过的所有指令行，而且所有这些被记录的指令行都能被复制，或再运行。关于历史指令窗的功能详见表 1.5-1。

表 1.5-1 历史指令窗主要应用功能的操作方法

应用功能	操作方法	简捷操作方法
单行或多行指令的复制	点亮单行或多行指令；按鼠标右键引出现场菜单；选中 {Copy} 菜单项，即可用复合键 [Ctrl + V] 把它“粘贴”到任何地方（包括指令窗）。	
单行指令的运行	点亮单行指令；按鼠标右键引出现场菜单；选中 {Evaluate Selection} 菜单项，即可在指令窗中运行，并见到相应结果。	鼠标左键双击单行指令。
多行指令的运行	点亮多行指令；按鼠标右键引出现场菜单；选中 {Evaluate Selection} 菜单项，即可在指令窗中运行，并见到相应结果。（详见例 1.5-1）	
把多行指令写成 M 文件	点亮多行指令；按鼠标右键引出现场菜单；选中 {Create M-File} 菜单项，就引出书写着这些指令的 M 文件编辑调试器；再进行相应操作，即可得所需 M 文件。	

### 1.5.2 历史指令的再运行

历史指令的重新调用，既可以采用第 1.4.4 节所介绍的方法实现，也可以借助历史指令窗进行。在许多场合，后者显得更为方便、直观。

【例 1.5-1】演示如何再运行算例 1.3-10 中的全部绘图指令。

具体操作过程：先利用组合操作 [Ctrl + 鼠标左键] 点亮如图 1.5-1 所示历史指令窗中的那五行指令；当鼠标光标在点亮区时，点击鼠标右键，引出现场菜单；选中现场菜单项 {Evaluate Selection}，计算结果就出现在指令窗中。

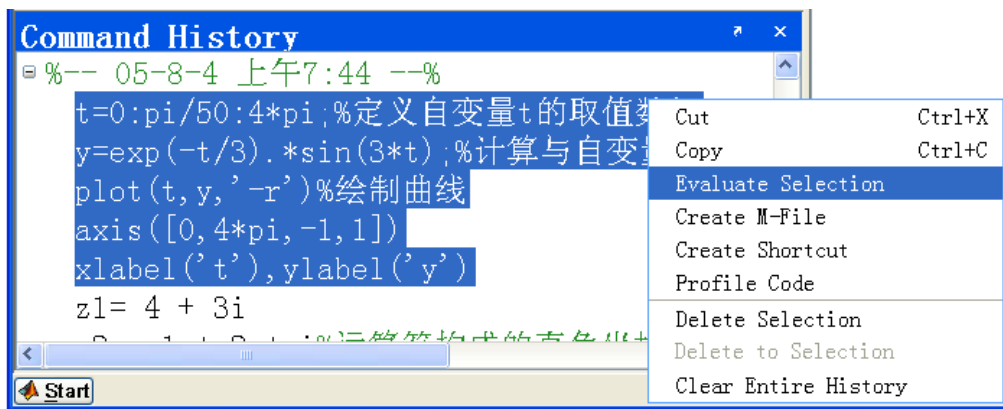


图 1.5-1 再运行历史指令的演示

【说明】

- 历史指令的复制操作步骤大抵相同。但在现场菜单中，应选 {Copy} 项。
- 单行历史指令的再运行操作更简单，只要用鼠标左键双击所需的那行指令即可。

## 1.6 Current Directory、路径设置器和文件管理

当在指令窗中运行一条指令时，MATLAB 是怎样从庞大的函数和数据库中，找到所需的函数和数据的呢？用户怎样才能保证自己所创建的文件能得到 MATLAB 的良好管理，又怎样能与 MATLAB 原有环境融为一体呢？这就是本节要介绍的内容。

### 1.6.1 Current Directory当前目录浏览器简介

如图 1.6-1 所示的当前目录浏览器界面上，自上而下分别是：当前目录名，工具条，文件、文件夹列表及文件描述区等。此外，MATLAB 还为当前目录窗设计了一个专门的操作菜单。借助该菜单可方便地打开或运行 M 文件、装载 MAT 文件数据等。详见表 1.6-1。

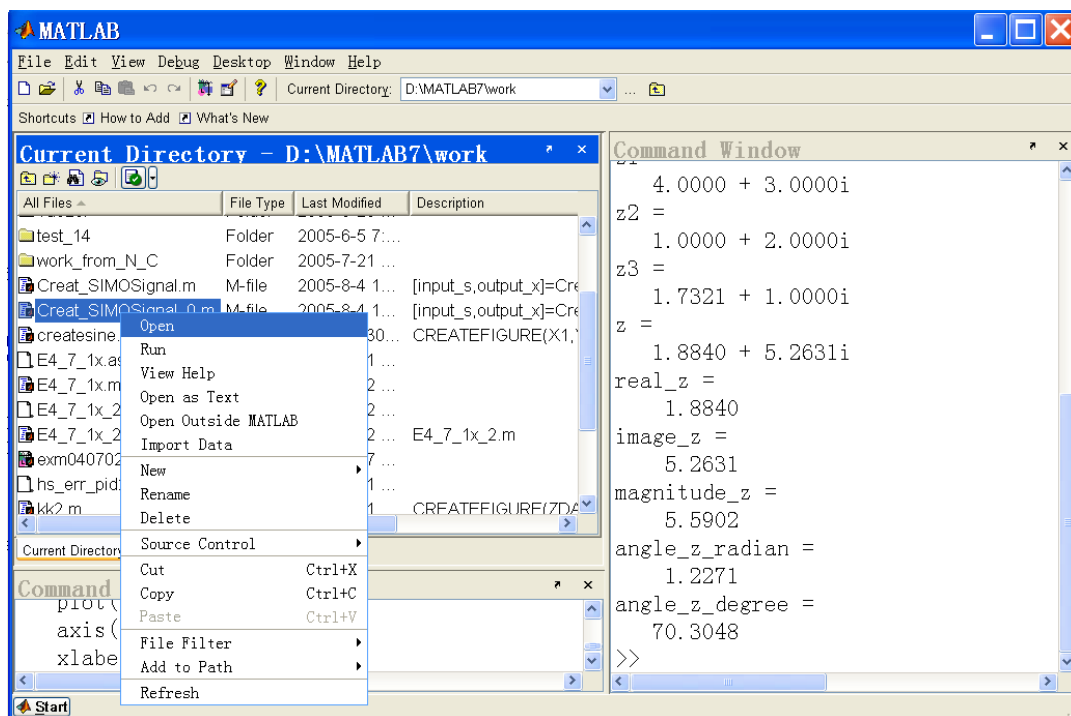


图 1.6-1 当前目录浏览器和适配的弹出菜单

表 1.6-1 当前目录适配菜单的应用

应用功能	操作方法	简捷操作方法
运行 M 文件	点亮待运行文件；按鼠标右键引出现场菜单；选中 {Run} 菜单项，即可使该 M 文件运行。	
编辑 M 文件	点亮待运行文件；按鼠标右键引出现场菜单；选中 {Open} 菜单项，此 M 文件就出现在编辑/调试器中。	鼠标左键双击 M 文件。
把 MAT 文件全部数据输入内存	点亮待装数据文件；按鼠标右键引出现场菜单；选中 {Open} 菜单项，此文件的数据就全部装入工作内存。	鼠标左键双击 MAT 文件。
把 MAT 文件部分数据输入内存	点亮待装载数据文件；按鼠标右键引出现场菜单；选中 {Import Data} 菜单项，引出数据预览选择对话框 “Import Wizard”；在此框中“勾选”待装数据变量名，点击 [Finish] 键，就完成操作。	

【说明】

- MATLAB 启动后的默认当前目录通常是：MATLAB\R2006a\work。应当指出：在该默认当前目录上存放用户文件是允许的、完全的、可靠的。MathWorks 公司之所以设计这样一个目录，就是供用户使用的。
- 若 MATLAB 的启动由点击 MATLAB\R2006a\bin\win32 目录下的 matlab.exe 触发，那么当前目录将是 MATLAB 所在的根目录。提醒读者：千万不要把 MATLAB\R2006a\bin\win32 设成当前目录，也尽量不要把 MATLAB 所在根目录设成当前目录。对此，用户应该通过重新设置，把当前目录设置在适当的目录上。

## 1.6.2 用户目录和当前目录设置

### (1) 建立用户目录

在使用 MATLAB 的过程中，为管理方便，本书作者建议：用户应尽量为自己建立一个专门的工作目录，即“用户目录”，用来存放自己创建的应用文件。尽管，matlab\work 允许用户存放文件，用户最好仅把它作为临时工作目录使用。

在 Windows 环境下，创建目录是十分简单的规范操作。因此，本文不再赘述。

### (2) 应把用户目录设置成当前目录

在 MATLAB 环境中，如果不特别指明存放数据和文件的目录，那么 MATLAB 总默认地将它们存放在当前目录上。因此，出于 MATLAB 运行可靠和用户方便的考虑，本书作者建议：在 MATLAB 开始工作的时候，就应把用户自己的“用户目录”设置成当前目录。

### (3) 把用户目录设置成当前目录的方法

方法一：交互界面设置法

在 MATLAB 操作桌面右上方，或当前目录浏览器左上方，都有一个当前目录设置区。它包括：“目录设置栏”和“浏览键”。用户或在“设置栏”中直接填写待设置的目录名，或借助“浏览键”和鼠标选择待设置目录。

方法二：指令设置法

通过指令设置当前目录是各种 MATLAB 版本都适用的基本方法。这种指令设置法的适

用范围比交互界面设置法大。它不仅能在指令窗中执行，而且可以使用在 M 文件中。假设待设置的用户目录是 `c:\mydir`，那么把它设置为当前目录的指令是 `cd c:\mydir`。

注意：以上方法设置的当前目录，只是在当前开启的 MATLAB 环境中有效。一旦 MATLAB 重新启动，以上设置操作必须重新进行。

### 1.6.3 MATLAB 的搜索路径

MATLAB 的所有(M、MAT、MEX)文件都被存放在一组结构严整的目录树上。MATLAB 把这些目录按优先次序设计为“搜索路径”上的各个节点。此后，MATLAB 工作时，就沿着此搜索路径，从各目录上寻找所需的文件、函数、数据。

当用户从指令窗送入一个名为 `cont` 的指令后，MATLAB 的基本搜索过程大抵如下。

- 检查 MATLAB 内存，看 `cont` 是不是变量；假如不是变量，则进行下一步。
- 检查 `cont` 是不是内建函数 (Built-in Function)；假如不是，再往下执行。
- 在当前目录上，检查是否有名为 `cont` 的 M 文件存在；假如不是，再往下执行。
- 在 MATLAB 搜索路径的其他目录中，检查是否有名为 `cont` 的 M 文件存在。

应当指出：(A) 实际搜索过程远比前面描述的基本过程复杂。但又有一点可以肯定，凡不在搜索路径上的内容，不可能被搜索。(B) 指令 `exist`、`which`、`load` 执行时，也都遵循搜索路径定义的先后次序。

### 1.6.4 MATLAB 搜索路径的扩展

#### 一 何时需要修改搜索路径

假如用户有多个目录需要同时与 MATLAB 交换信息，那么就应把这些目录放置在 MATLAB 的搜索路径，使得这些目录上的文件或数据能被调用。又假如其中某个目录需要用来存放运行中产生的文件和数据，那么还应该把这个目录设置为当前目录。

#### 二 利用设置路径对话框修改搜索路径

采用以下任何一种方法都可以引出设置路径对话框 (图 1.6-2)。

- 在指令窗里，运行指令 `pathtool`。
- 在 MATLAB 桌面、指令窗等的菜单条中，选择 {File:Set Path} 下拉菜单项。

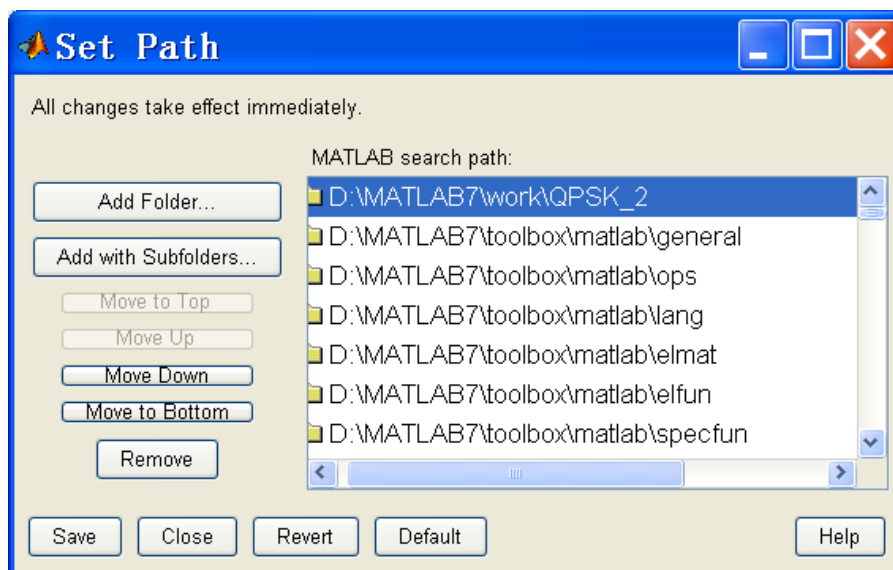


图 1.6-2 路径设置对话框

【说明】

- 该对话框设置搜索路径有两种修改状态：  
 当前有效修改            假如在路径设置过程中，仅使用了该对话框的左侧按键。  
 永久有效修改            假如在设置后，点击了对话框下方 [Save] 按键。
- 所谓永久有效修改是指：所进行的修改不因 MATLAB 的关闭而消失。

### 三 利用指令 `path` 设置路径

利用 `path` 指令设置路径的方法对任何版本的 MATLAB 都适用。假设待纳入搜索路径的目录为 `c:\my_dir`，那么以下任何一条指令均能实现：

`path(path,'c:\my_dir')`        把 `c:\my_dir` 设置在搜索路径的尾端  
`path('c:\my_dir',path)`       把 `c:\my_dir` 设置在搜索路径的首端

【说明】

- 用 `path` 指令扩展的搜索路径仅在当前 MATLAB 环境下有效。也就是说：若用户退出当前 MATLAB 后，再重新启动 MATLAB，那么在前一环境下用 `path` 所定义的扩展搜索路径无效。
- 用 `path` 指令扩展的搜索路径的方法可以编写在程序中。

## 1.7 工作空间浏览器和数组编辑器

### 1.7.1 Workspace 工作空间浏览器简介

工作空间浏览器（或称内存浏览器）默认地放置于 MATLAB 操作桌面的左上侧后台。点击桌面左上侧框下方外露的“Workspace”窗标，可使工作空间浏览器出现在桌面的前台。该浏览器的功用，详见表 1.7-1。



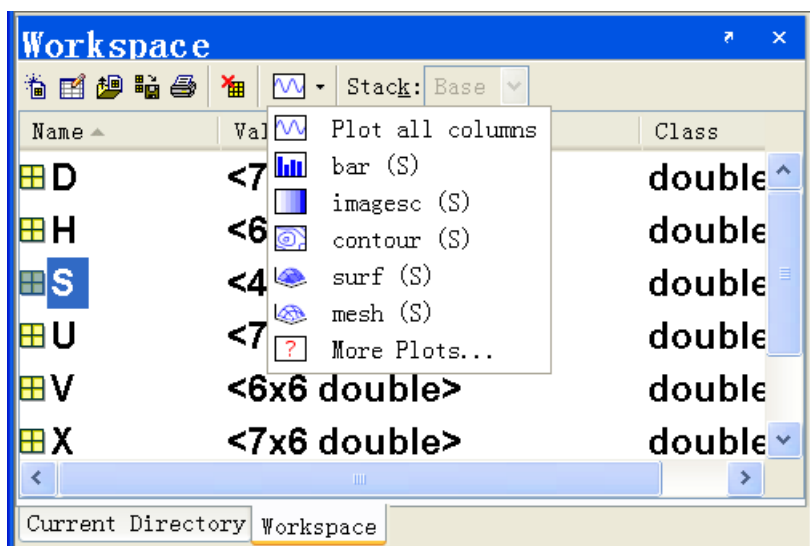


图 1.7-1 工作空间浏览器及“绘图工具”图标的展开

表 1.7-1 工作空间浏览器主要功能及其操作方法

功 能	操 作 方 法	简捷操作法
创建 新变量	点击图标，在工作空间中生成一个“unnamed”的新变量；双击该新变量图标，引出 Array Editor 数组编辑器(如图 1.7-3)；在数组编辑器中，向各元素输入数据；最后，对该变量进行重命名。	
显示 变量内容	点亮变量；或点击图标，或选中弹出菜单中的 {Open Selection} 项，则变量内含的数据就显示在“Array Editor”数组编辑器中。（数组编辑器详见第 1.7-3 节）	用鼠标左键 双击变量。
图示变量	点亮变量；或选中图标下拉菜单项，或选中弹出菜单中的绘图项，就可以适当地绘出选中变量的图形。（详见例 1.7-1）	
用文件 保存变量	选择待保存到文件的（一个或多个）变量，或点击图标，或选中弹出菜单中的 {Save Workspace As}项，便可把那些变量保存到 MAT 数据文件。（详见第 1.7-4 节之一）	
从文件向 内存装载 变量	点击图标；选择 MAT 数据文件；再单击那文件，引出“Import Wizard”界面，它展示文件所包含的变量列表；再从列表中，选择待装载变量，便可。	

【例 1.7-1】通过“工作空间浏览器”的运作，采用图形显示内存变量  $y$ 。（注意：本例是在例 1.3-10 运行后进行的，因此内存中保存着由例 1.3-10 产生的全部变量。）

（1）借助“图标菜单”图示变量

在“工作空间浏览器”中，用鼠标点亮所需图示的变量  $y$ ；再选中“图标菜单”的 {plot(y)}，就可得到图 1.7-2 所示的图形。

（2）借助“现场弹出菜单”图示变量

在鼠标点亮变量  $y$  后，按鼠标右键引出“现场菜单”，再选择现场菜单中的 {plot(y)}菜单项，也可以画出图 1.7-2 所示的曲线。

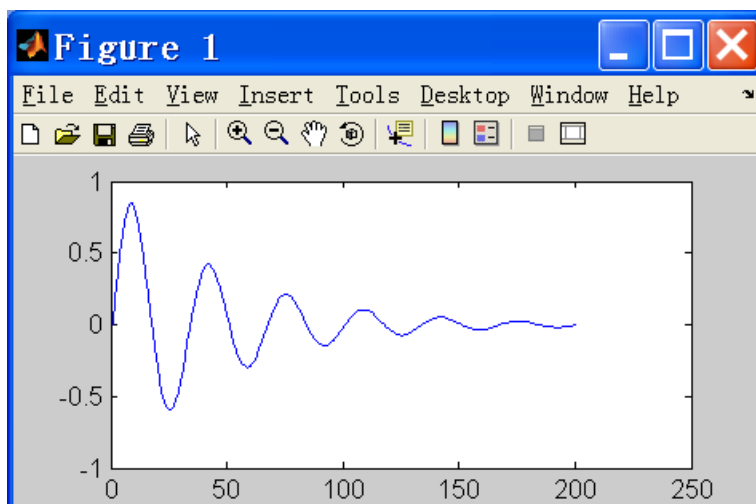


图 1.7-2 数组  $y$  表示的曲线

【说明】

- 例 1.3-10 的图 1.3-4 虽与本例图 1.7-2 形状相同，但含义却不同。这两张图的纵轴刻度相同，但横轴刻度不同。前者表现的是函数关系  $y = f(t)$ ；而后者表现的是数组  $y(k)$ ，在此  $k$  是数组元素的序号。
- 注意：借助图形表现数据是十分常用的手段。这是由于图形具有表现数据内在联系和宏观特征的卓越能力。

## 1.7.2 工作空间的管理指令

本节要介绍管理工作空间的三个常用指令：who, clear, pack。

### 一 查询指令who及whos

【例 1.7-2】在指令窗中运用 who, whos 查阅 MATLAB 内存变量。

who, whos 在指令窗中运行后的显示结果如下：

who

Your variables are:

ans t y

whos

Name	Size	Bytes	Class
ans	1x1	8	double array
t	1x201	1608	double array
y	1x201	1608	double array

Grand total is 403 elements using 3224 bytes

【说明】

- who, whos 指令操作对 MATLAB 的所有版本都适用。
- 本例两个指令的差别仅在于获取内存变量信息的简单和详细程度不同。
- 读者运行 who, whos 指令后的变量列表随具体情况而不同。本例的变量列表是在本书作者进行特定操作后产生的。

## 二 从工作空间中删除变量和函数的指令clear

最常用的几种格式：

clear	清除工作空间中的所有变量
clear var1 var2	清除工作空间中的 var1 和 var2 变量
clear all	清除工作空间中所有的变量、全局变量、编译过的 M 函数和 MEX 链接。
clear fun1 fun2	清除工作空间中名为 fun1 和 fun2 的函数

注意：在第 2、4 调用格式中，clear 后面的变量名和函数名之间一定要采用“空格”分隔，而不能采用其他符号。

## 三 整理工作空间内存碎片的指令pack

在 MATLAB 运行期间，它会自动地为产生的变量分配内存，也会为使用到的 M 函数分配内存。有时对于容量较大的变量，会出现“Out of memory”的错误。此时，可能使用 clear 指令清除若干内存中的变量也无济于事。产生这种问题的一个原因是：MATLAB 存放一个变量时，必须使用“连成一片”的内存空间。对于那些被碎片分割得“支离破碎”的内存空间，即便它们的总容量超过待生成变量，也无法使用。在这种情况下，借助 pack 指令也许能解决问题。

### 1.7.3 Array Editor数组编辑器

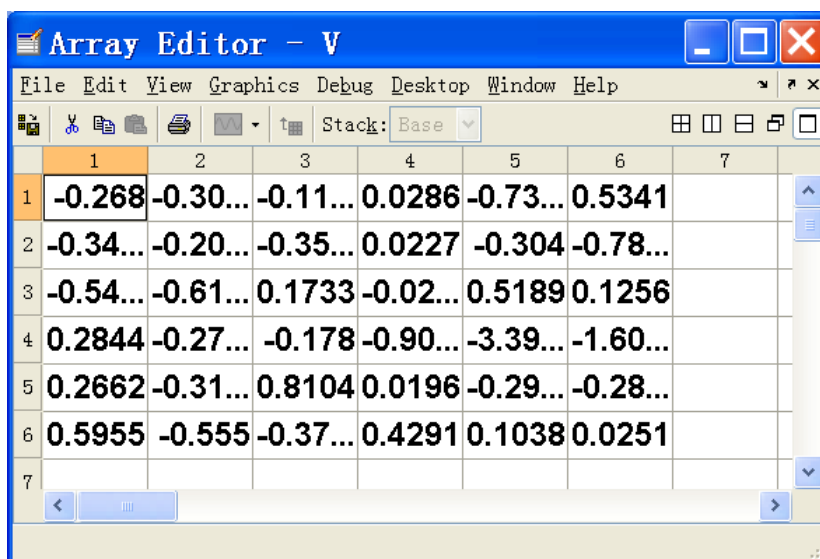



图 1.7-3 工作空间变量数组编辑器


双击工作空间浏览器中的变量图标，将引出如图 1.7-3 所示的数组编辑器 Array Editor。该编辑器可用来查看、编辑数组元素；对数组中指定的行或列进行图示。

点击图标，创建一个名为“unnamed”的变量；在双击该变量引出一个与图 1.7-3 类似的界面。但数组中，除第一个元素为 0 外，其余均为“空白”。利用这个界面，读者就可以比较自在地输入较大的数组。


从 MATLAB7.0 版起，数组编辑器不但能观察和编辑“双精度”数组，而且也能观察和编辑“字符串”数组、“胞元”数组、和“构架”数组。

## 1.7.4 数据文件和变量的存取

### 一 借助工作空间浏览器产生保存变量的MAT文件

从工作空间浏览器中选择待保存到文件的（一个或多个）变量；再点击图标，或选中弹出菜单中的 {Save As} 项；就可把那些变量保存到（由用户自己命名的）mydata.mat 的数据文件。该数据文件默认地存放在“当前目录”上。

### 二 借助输入向导Import Wizard向工作空间装载变量

点击工作空间浏览器上的图标，或 MATLAB Desktop 下拉菜单 {File: Import Data}；在用户希望的目录上，选中 MAT 数据文件（如当前目录上的 mydata.mat）；再双击那文件，引出如图 1.7-4 所示的“Import Wizard”界面，它展示出文件所包含的变量列表；再从列表中，通过“勾选”，选择待装载变量（如图中的 t 和 y）；再点击 [Finish] 按键，变量 t 和 y 就被装载到工作空间。

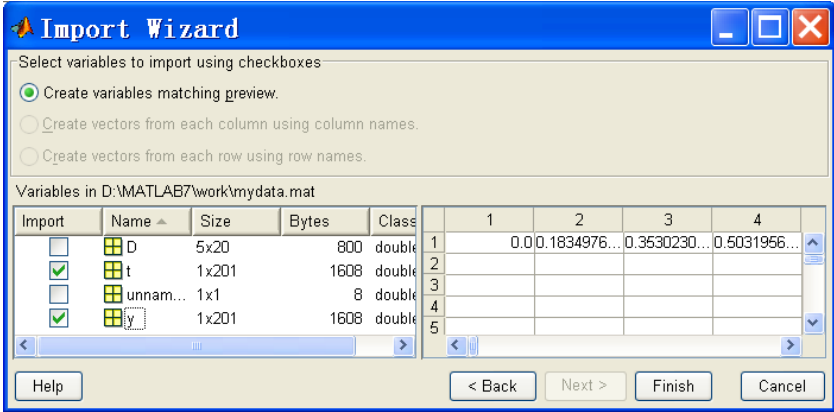


图 1.7-4 输入向导

### 三 存取数据的操作指令save 和load

利用 save, load 指令实现数据文件存取是 MATLAB 各版都采用的基本操作方法。它的具体使用格式如下。

<b>save FileName</b>	把全部内存变量保存为 FileName.mat 文件
<b>save FileName v1 v2</b>	把变量 v1, v2 保存为 FileName.mat 文件
<b>save FileName v1 v2 -append</b>	把变量 v1, v2 添加到 FileName.mat 文件中
<b>save FileName v1 v2 -ascii</b>	把变量 v1, v2 保存为 FileName 8 位 ASCII 文件
<b>save FileName v1 v2 -ascii -double</b>	把变量 v1, v2 保存为 FileName 16 位

ASCII 文件

<b>load FileName</b>	把 FileName.mat 文件中的全部变量装入内存
<b>load FileName v1 v2</b>	把 FileName.mat 文件中的 v1, v2 变量装入内存
<b>load FileName v1 v2 -ascii</b>	把 FileName ASCII 文件中的 v1, v2 变量装入内存

【说明】

- FileName 文件名可以带路径，也可以带扩展名。
- v1, v2 代表变量名；指定的变量个数不限，只要内存或文件中存在；变量名与变量名之间必须以空格相分隔。
- -ascii 选项使数据以 ASCII 格式处理。生成的（不带扩展名的）ASCII 文件可以在任何“文字处理器”中被修改。如果数据较多的变量需要进行修改，那么 ASCII 格式的数据文件很适用。
- 如果指令后没有 -ascii 选项，那么数据以二进制格式处理。生成的数据文件一定带 mat 扩展名。

【例 1.7-3】数据的存取。（假定内存中已经存在变量 X,Y,Z）

（1）建立用户目录，并使之成为当前目录，保存数据

<b>mkdir('c:\', 'my_dir');</b>	%在 C 盘上创建目录 my_dir
<b>cd c:\my_dir</b>	%使 c:\my_dir 成为当前目录
<b>save saf X Y Z</b>	%选择内存中的 X,Y,Z 变量保存为 saf.mat 文件
<b>dir</b>	%显示目录上的文件
<b>. .. saf.mat</b>	

（2）清空内存，从 saf.mat 向内存装载变量 Z

<b>clear</b>	%清除内存中的全部变量
<b>load saf Z</b>	%把 saf.mat 文件中的 Z 变量装入内存
<b>who</b>	%检查内存中有什么变量
<b>Your variables are:</b>	
<b>Z</b>	

【说明】

- 如果一组数据是经过长时间的复杂计算后获得的，那么为避免再次重复计算，常使用 save 加以保存。此后，每当需要，都可通过 load 重新获取这组数据。这种处理模式常在实际中被采用。

## 1.8 Editor/Debugger和脚本编写初步

对于比较简单的问题，对于那些一次性问题，通过指令窗中直接输入一组指令去求解，也许是比较简便、快捷的。但当待解决问题所需的指令较多和所用指令结构较复杂时，或当一组指令通过改变少量参数就可以被反复使用去解决不同问题时，直接在指令窗中输入指令的方法就显得烦琐和笨拙。M 脚本文件就是设计来解决这个矛盾的。

## 1.8.1 Editor/Debugger M文件编辑器简介

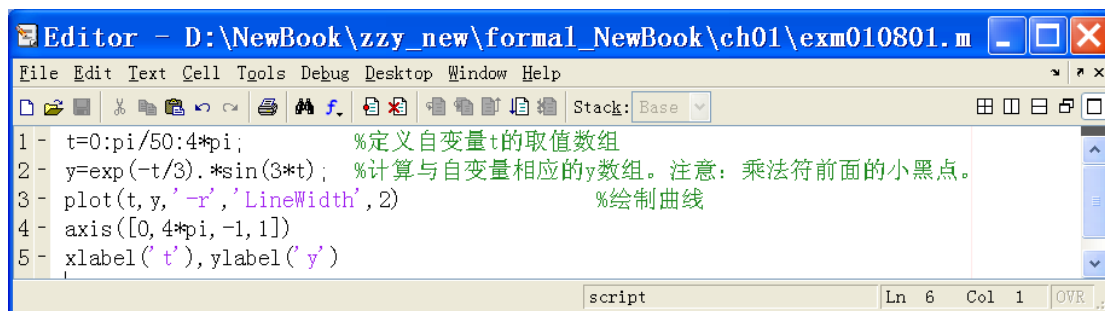




图 1.8-1 M 文件编辑器示图

默认情况下，M 文件编辑器（Editor/Debugger）不随 MATLAB 的启动而开启，而只有当编写 M 文件时才启动。M 编辑器不仅可以编辑 M 文件，而且可以对 M 文件进行交互式调试；M 文件编辑器不仅可处理带 .m 扩展名的文件，而且可以阅读和编辑其他 ASCII 码文件。

M 文件编辑器的启动方法有以下几种：

- 点击 MATLAB 桌面上的  图标，或选中菜单项 {File: New: M-File}，或直接在指令窗口输入指令 `edit`，都可以打开空白的 M 文件编辑器。
- 点击 MATLAB 桌面上的  图标，或选中菜单项 {File: Open}，可引出“Open”文件选择对话框，在填写所选文件名后，再点 [Open] 键，就可引出展示相应文件的 M 文件编辑器。在指令窗中，把待打开文件名（经一空格）写在 `edit` 后，指令运行后，文件编辑器就打开那文件。
- 用鼠标左键双击当前目录窗中的所需 M 文件，可直接引出展示相应文件的 M 文件编辑器。

## 1.8.2 M脚本文件编写初步

所谓 M 脚本文件是指：（1）该文件中的指令形式和前后位置，与解决同一个问题时在指令窗中输入的那组指令没有任何区别。（2）MATLAB 在运行这个脚本时，只是简单地从文件中读取那一条条指令，送到 MATLAB 中去执行。（3）与在指令窗中直接运行指令一样，脚本文件运行产生的变量都是驻留在 MATLAB 基本工作空间中。（4）文件扩展名是“.m”。

【例 1.8-1】编写解算例 1.3-10 题目的 M 脚本文件，并运行之。

操作步骤：

- （1）在历史指令窗中，找到算例 1.3-10 的运行指令，并把它们选中点亮。
- （2）点击鼠标右键，选中弹出现场菜单中的 {Create M-file}，便引出如图 1.8-1 的 M 文件编辑器。
- （3）以 `exm010801` 为文件名进行保存，于是就得到了 `exm010801.m` 文件。
- （4）保证这个文件在 MATLAB 的当前目录或搜索路径上，在指令窗中运行 `exm010801` 就可以得到如图 1.3-4 的曲线。

【说明】

- 当使用 M 文件编辑调试器保存文件时，或当在 MATLAB 指令窗中运行 M 文件时，不

必写出文件的扩展名。

- 在 M 文件编辑调试器中，可以采用汉字注释，并总可获得正确显示。

## 1.9 帮助系统及其使用

读者接触、学习 MATLAB 的起因可能不同，借助 MATLAB 所想解决的问题也可能不同，从而会产生不同的求助需求。如对于初学者，最急于知道的是：MATLAB 的基本用法。又如 MATLAB 老用户很想知道的是：MATLAB 新版本有什么新特点、新功能。再如对科研工作者来说，面对不断变化的实际问题，常常产生两类困惑：知道具体指令，但不知道该怎么用；或想解某个具体问题，不知道 MATLAB 有哪些指令可用。

MATLAB 作为一个优秀的科学计算软件，其帮助系统考虑了不同用户的不同需求，构成了一个比较完备的帮助体系。并且，这种帮助体系随 MATLAB 版本的重大升级，其完备性和友善性都会有较大的进步。

作者在此建议：不管以前是否使用过 MATLAB，任何用户都应尽快了解 MATLAB 的帮助系统，掌握各种获取帮助信息的方法。只有这样，用户才可能较好地运用 MATLAB 资源，快捷、可靠、有效地独立解决自己面临的各种问题。

### 1.9.1 六大子系统构成的帮助体系

表 1.9-1 MATLAB 的帮助体系

子系统名称	特 点	资 源
指令窗帮助子系统	文本形式；最可信、最原始；不适于系统阅读	所有 M 文件的帮助注释内容；包含在 M 文件之中
帮助导航系统	HTML 形式；系统叙述 MATLAB 规则，一般用法；适于系统阅读和交叉查阅	位于 matlab\help 目录下；HTML 和 XML 文件，物理上独立于 M 文件
典型算例演示系统	HTML 和 GUI 交互界面；以算例为载体分类介绍 MATLAB 使用方法	位于 matlab\toolbox 下各分类子目录的 demos 子目录；专门编写 HTML 和演示 M 文件
视频演示系统	Flash 形式；视听兼备，直观形象；内容限于版本新特点	位于 matlab\demos 目录下；Flash 文件
PDF 文件帮助系统	系统的标准打印文件；便于长时间系统阅读	下载网站 <a href="http://www.mathworks.com/access/helpdesk/help/helpdesk.html">http://www.mathworks.com/access/helpdesk/help/helpdesk.html</a>
Web 网上帮助系统	交互式讨论具体问题	制造商网站 <a href="http://www.mathworks.com">http://www.mathworks.com</a> 新闻组 comp.soft-sys.matlab 讨 论 站 <a href="http://www.mit.edu/~pwb/cssm/matlab-faq.html">www.mit.edu/~pwb/cssm/matlab-faq.html</a>

### 1.9.2 常用帮助指令



## 一 函数搜索指令

在“知道具体函数指令名称，但不知道该函数如何使用”的场合，运用函数搜索指令能很好地获得帮助信息。函数搜索指令的调用方法如下：

<b>help</b>	列出所有函数分组名 (Topic Name)
<b>help TopicName</b>	列出指定名称函数组中的所有函数
<b>help FunName</b>	给出指定名称函数的使用方法

<b>helpwin</b>	列出所有函数分组名 (Topic Name)
<b>helpwin TopicName</b>	列出指定名称函数组中的所有函数
<b>helpwin FunName</b>	给出指定名称函数的使用方法

<b>doc ToolboxName</b>	列出指定名称工具包中的所有函数名
<b>doc FunName</b>	给出指定名称函数的使用方法

【说明】

- 在此，TopicName, FunName, ToolboxName 分别用来表示待搜索的分组函数名、函数文件名、工具包名。
- help 搜索的资源是 M 文件帮助注释区的内容。这部分资源用纯文本形式写成。它简扼地叙述该函数的调用格式和输入输出量含义。该帮助内容最原始，但也最真切可靠。
- helpwin 搜索的资源还是 M 文件帮助注释区的内容。但它的显示形式已不再是“比较简陋的文本”，而被自动转换成“比较方便的超文本”。
- doc 搜索是在 HTML 文件构成的帮助子系统中进行的。HTML 文件是根据 M 文件资源编写的，内容比 M 文件帮助注释详细。该子系统，由于采用“超链接”机理，因此检索、查阅比较方便。
- 函数搜索指令的功能与帮助导航器中的“Index 搜索窗”相同。

## 二 词条搜索指令

在“想解某具体问题，但不知道有哪些函数指令可以使用”的场合，词条搜索指令也许比较有用。

<b>lookfor Keyword</b>	对 M 文件 H1 行进行单词条检索
<b>docsearch(' Keyword1 @ Keyword2 @ Keywordn' )</b>	对 HTML 子系统进行多词条检索

【说明】

- 在此，Keyword, Keyword1, Keyword2, Keyword3 分别是待检索的词条。而 @ 表示逻辑运算符。在实际使用中，该 @ 应写成 OR, AND, NOT 中的任意一个。待索词条、逻辑符之间用“空格”分隔。
- lookfor 搜索的资源是 M 文件帮助注释区中的第一行（简称 H1 行）。
- docsearch 搜索是在 HTML 文件构成的帮助子系统中进行的。它的搜索功能强、效率高，搜索到的内容也比较详细。
- 词条搜索指令的功能与帮助导航器中的“Search 搜索窗”相同。




## 1.9.3 Help Navigator/Browser帮助导航浏览器

帮助导航/浏览器搜索的资源是 Mathworks 专门写成的 HTML 帮助子系统。它的内容来源于所有 M 文件，但更详细。它的界面友善、方便。这是用户寻求帮助的最主要资源。

引出帮助导航/浏览器的方法有以下几种：

- (1) 在指令窗中运行 `helpbrowser` 或 `helpdesk`。
- (2) 在 MATLAB 的默认操作桌面，及各独立出现的交互窗口中

或点击工具条的  图标；

或选中下拉菜单项 {Windows: Help}；

或选中下拉菜单项 {Help: MATLAB Help} 。

### 一 帮助导航/浏览器简介

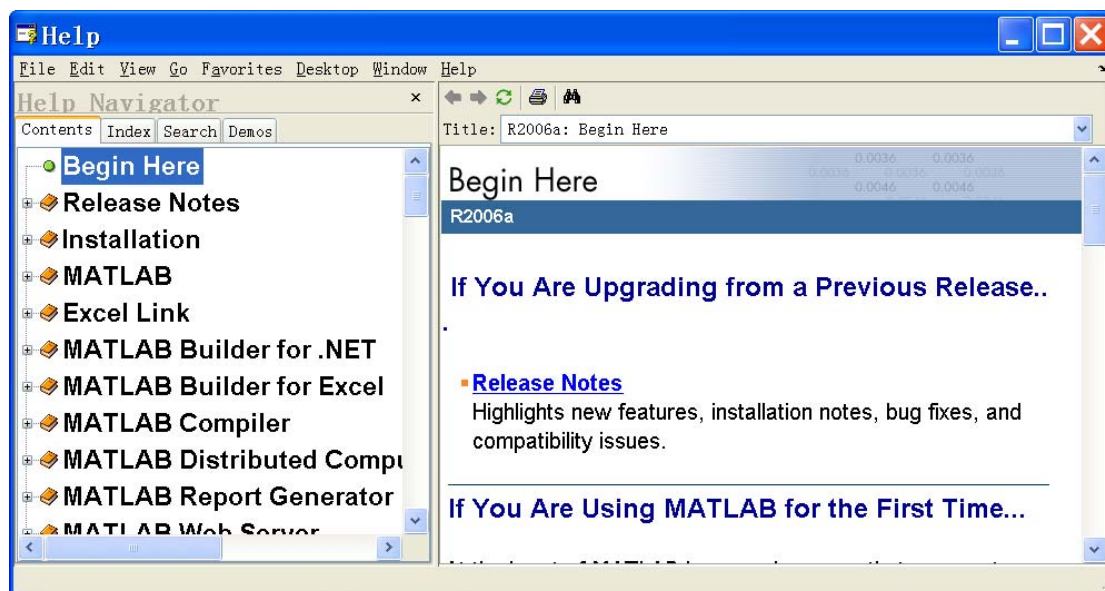


图 1.9-1 帮助导航/浏览器

整个帮助界面由分列于左右半侧的帮助导航器 Help Navigator 和帮助浏览器 Help Browser 部分组成。借助鼠标对这两部分的“中间分隔线”的左右调节，改变每部分所占版面。帮助导航器有四个“导航窗”：内容分类目录 Contents，指令检索 Index，词条检索 Search，演示实例 Demos。

### 二 Contents目录窗

该窗口列出“节点可展开的目录树”。用鼠标点击目录条，即可在 Help Browser 帮助浏览器中显示出相应标题的 HTML 帮助文件。

该窗口是向用户提供全方位系统帮助的“向导图”。该向导图的构造贯穿四个原则，即“深浅层次”、“功能划分”、“帮助方式”和“英文字母次序”。关于“深浅层次”原则的详细说明，请见表 1.9-2。

表 1.9-2 按深浅层次设置的条目

层次类型	功 用	条 目 举 例	阅读建议
概括	简介特点、内容、方法	Begin Here	粗读
版本特点	专门介绍版本升级变化	Release Notes	老用户必读
入门和使用初步	以简单操作、说明和算例引导用户	Getting Started	新用户必读 老用户浏览
详细指南	全面、系统、按功能叙述	Using MATLAB	或系统读 或选读
综合算例	强调指令间配合和使用要领演示	Examples	选读

### 三 Index检索窗

正如许多书籍后所附的“术语索引表”一样，MATLAB 也有一个事先制作的指令、函数列表文件。该文件专供“Index”窗（图 1.9-2）使用。当用户在“Index”窗上方“待查栏（Search index for）”中填入某词汇时，其下白板上就列出与之最匹配的词汇列表。

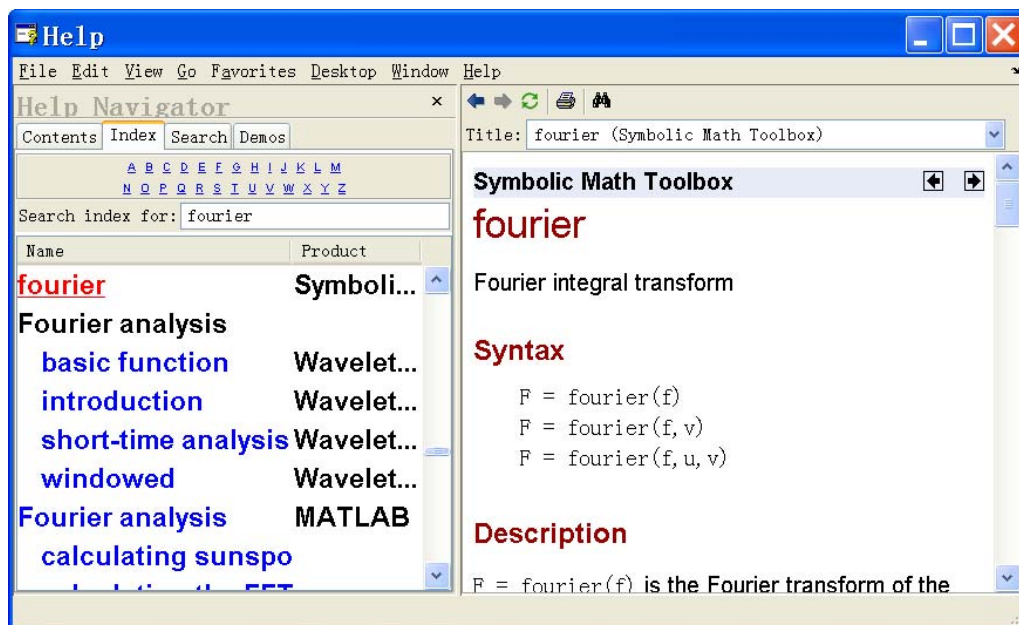


图 1.9-2 “Index”窗对 fourier 词汇的搜索结果片断

### 四 Search搜索窗

“Search”窗（图 1.9-3）是利用关键词查找全文中与之匹配章节条目的交互界面。这是电子读物所特有的最大优点之一。它与 Index 搜索有两点主要区别：

- （1）与 Index 搜索只在专用指令表中查找不同，Search 搜索是在整个 HTML 文件中进行的，因此其覆盖面更宽。
- （2）Index 搜索只能进行单指令名搜索，而 Search 搜索采用多词条的逻辑组合搜索，功能更强，搜索效率更高。

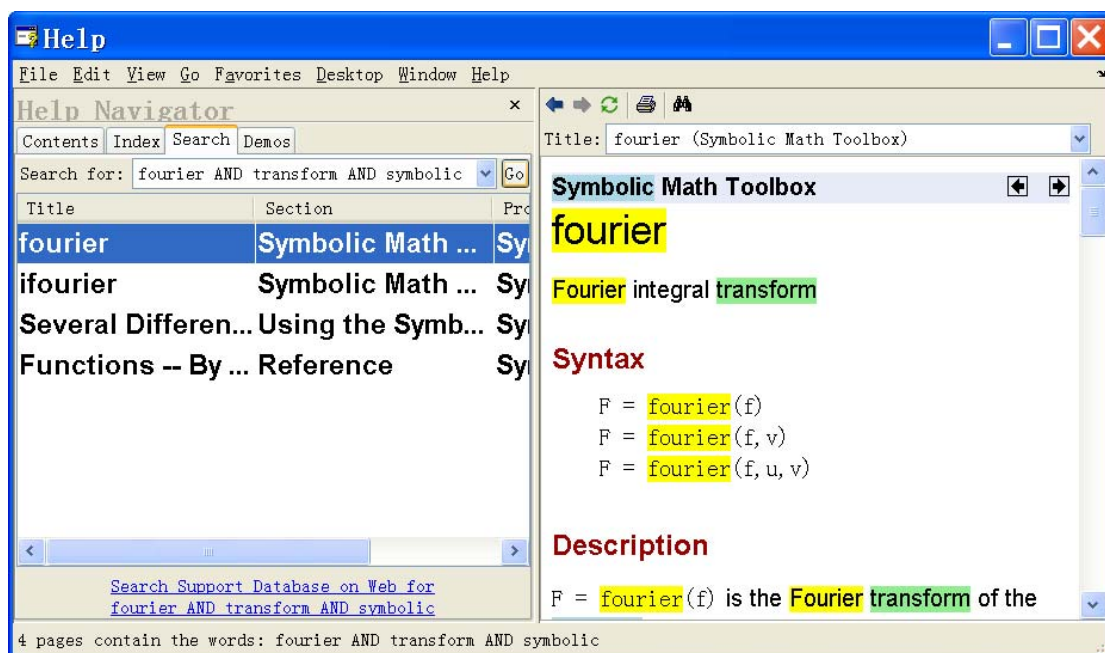


图 1.9-3 Search 窗逻辑组合搜索示意图

## 1.9.4 DEMO演示系统

与以前版本相比，MATLAB R2006.a 版的 DEMO 显得更为系统和完善。它以算例为载体，由 HTML、GUI 界面、显示 M 文件的编辑器组合而成。该演示系统，分布在 matlab\toolbox 的各个分类子目录中。该系统综合演示：为解决一个具体算例，各指令间如何配合使用。无论是对 MATLAB 新用户来说还是对老用户来说，都是十分有益的。该演示程序的示范作用是独特的，是包括 MATLAB 用户指南在内的有关书籍所不能代替的。用户若想学习和掌握 MATLAB，不可不看这组演示程序。但对初学者来说，不必急于求成去读那些太复杂的程序。

在指令窗中运行 demos 指令，或点击帮助导航器中的“Demos”窗，就可引出如图 1.9-4 的 DEMO 演示系统。该界面左侧采用标准的 Windows 目录树结构清晰地展示各分类内容的多层次演示结构。界面右侧是 HTML 写成的一组“蓝色”超链接。用鼠标点击超链接就可引出如图 1.9-5 的 HTML 演示浏览界面。

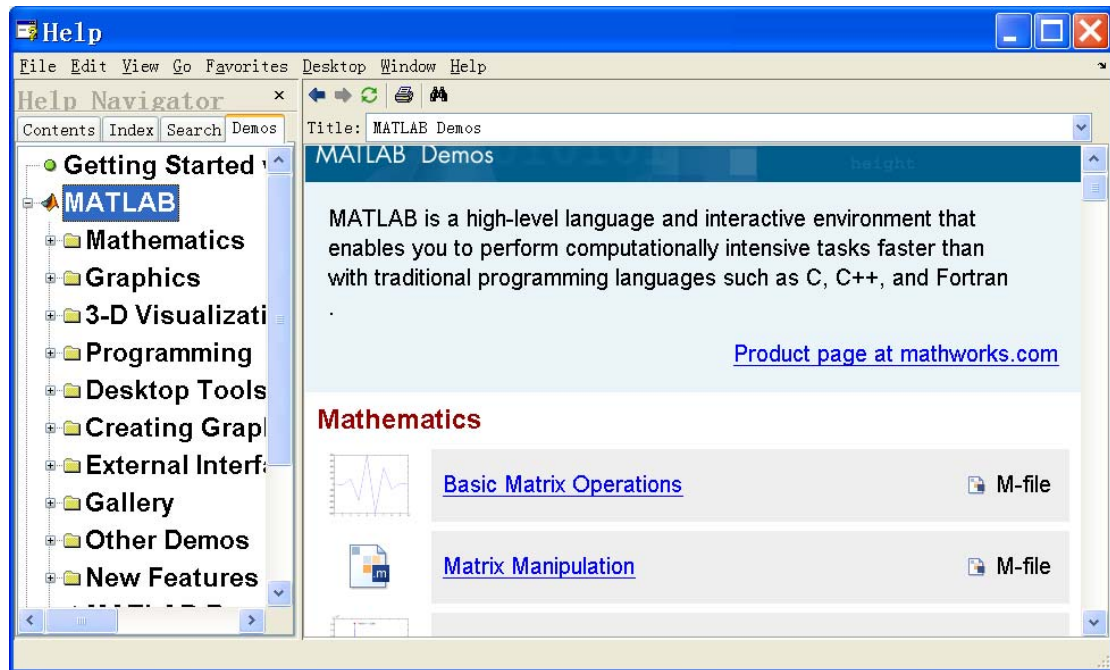


图 1.9-4 帮助导航其中的 Demos 窗

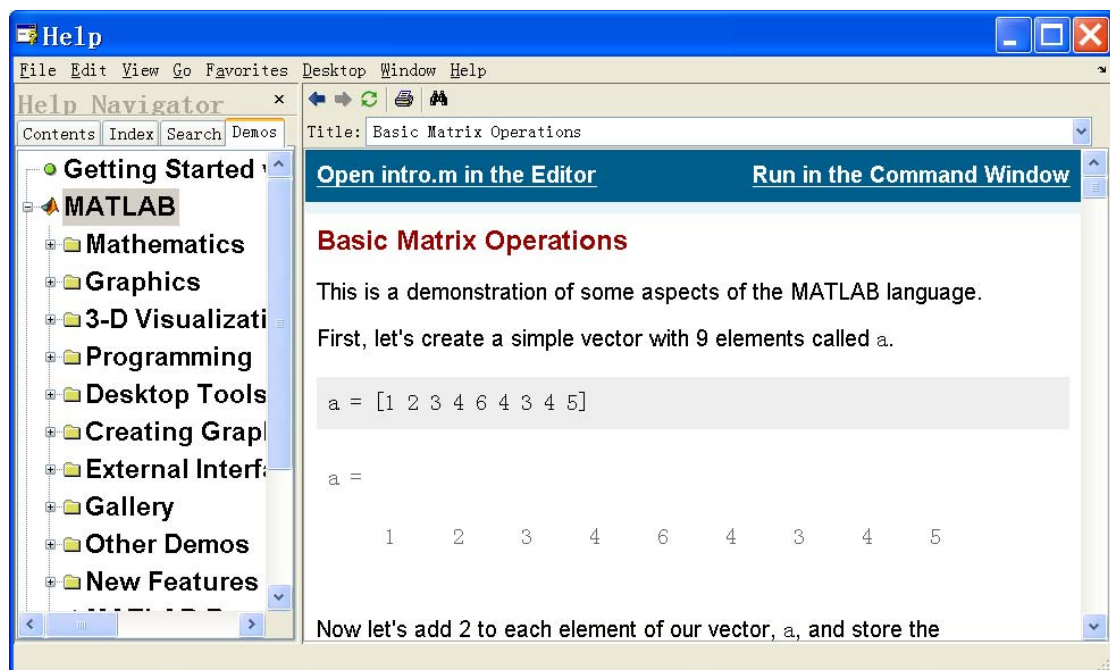


图 1.9-5 算例演示的 HTML 浏览器

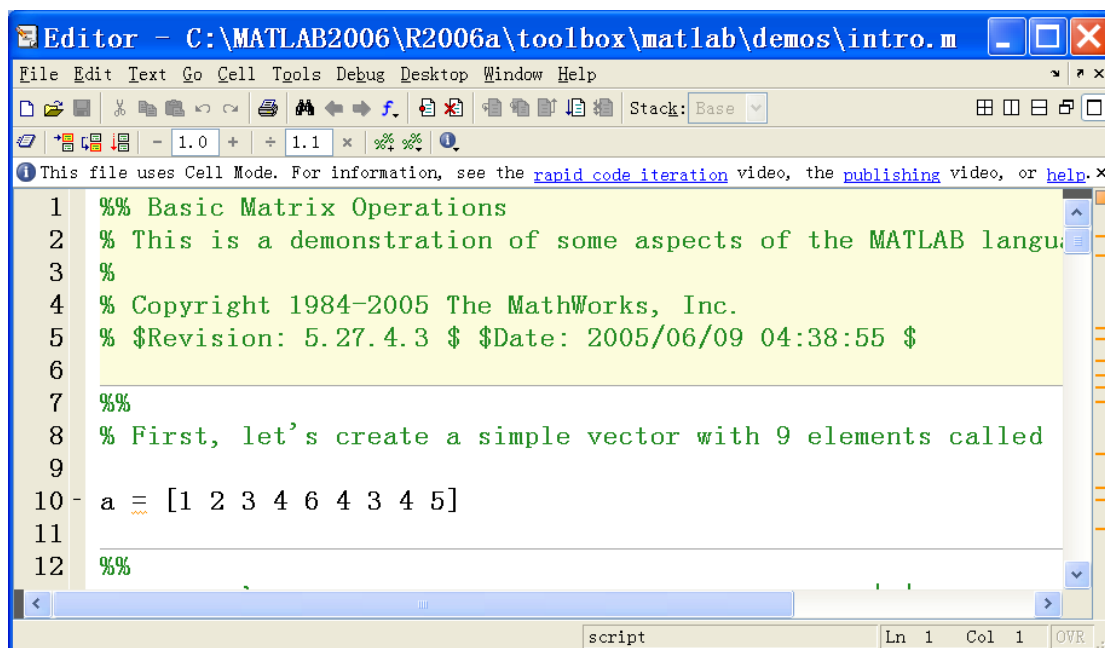


图 1.9-6 演示算例的 M 文件

图 1.9-5 算例演示浏览器的用法：

- (1) 直接阅读浏览器显示的内容。这种形式比较接近读者的阅读习惯，看起来比较舒畅。
- (2) 点击图 1.9-5 左上方的“Open intro.m in the Editor”超链接，引出如图 1.9-6 所示的采用 M 文件编辑器显示的演示 M 文件。读者从编辑器中看到是：直接面向 MATLAB 的 M 文件。它有助于读者习惯和理解 M 文件的编写。这些文件都可以在 MATLAB 中直接运行。
- (3) 点击图 1.9-5 右上方的“Run in the Command Window”超链接，就会出现如图 1.9-7 所示的 Command Window 界面。通过操作该界面左侧的“Next”超链接，一步一步地观察指令、运算结果和可视化图形之间的关系。

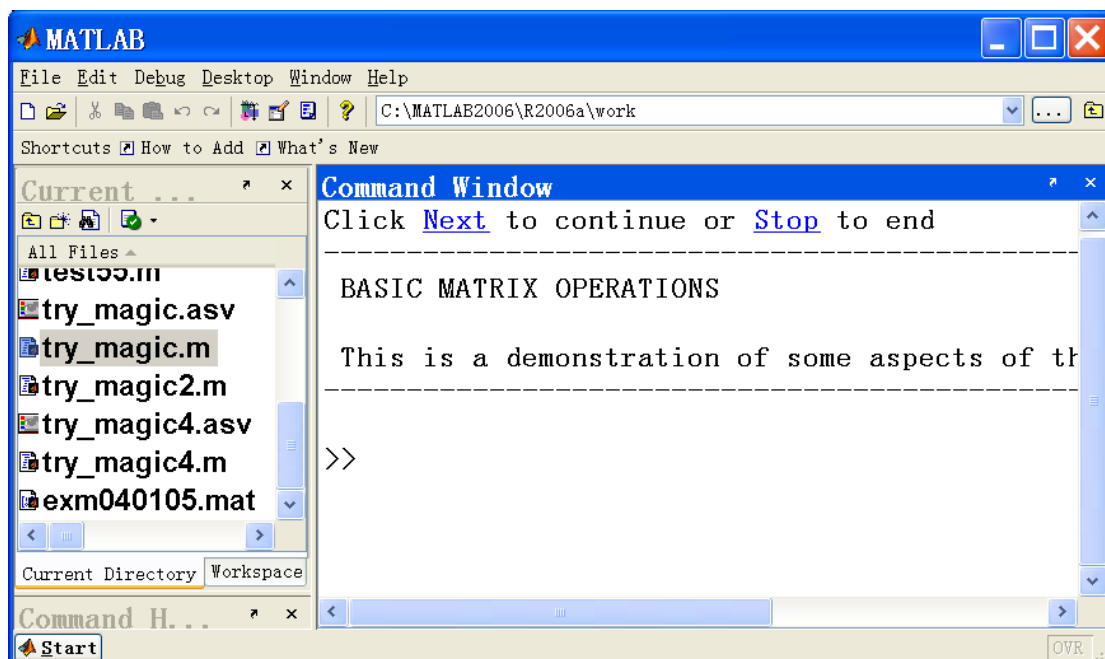




图 1.9-7 演示算例的 GUI 交互界面

## 1.9.5 VIDEO 视频演示帮助系统

在 MATLAB 的 Demos 里还提供一种新的视频帮助资料，专门介绍新特点。它形象、生动、直观，便于用户了解和掌握新版的功能。

进入 VIDEO 演示系统最方便的途径是：选择帮助导航器 Demos 窗中的“New Features in Version 7”，直接导出 VIDEO 视频演示“点播台”。

根据需要，点中“点播台”上所需栏目，就导出如图 1.9-9 的视频播放器。通过眼看鼠标操作和界面图像变化，耳听相配的英语解释，用户很容易学会各种操作之间的关系和 MATLAB 各种新功能的使用。

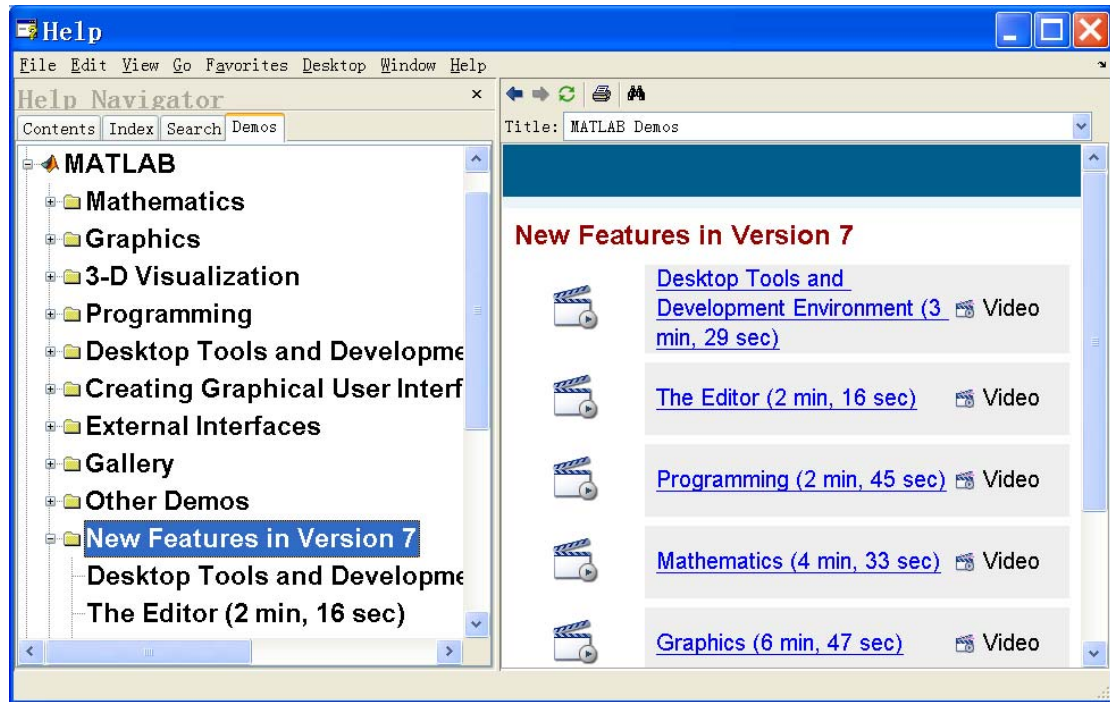


图 1.9-8 VIDEO 视频演示点播台

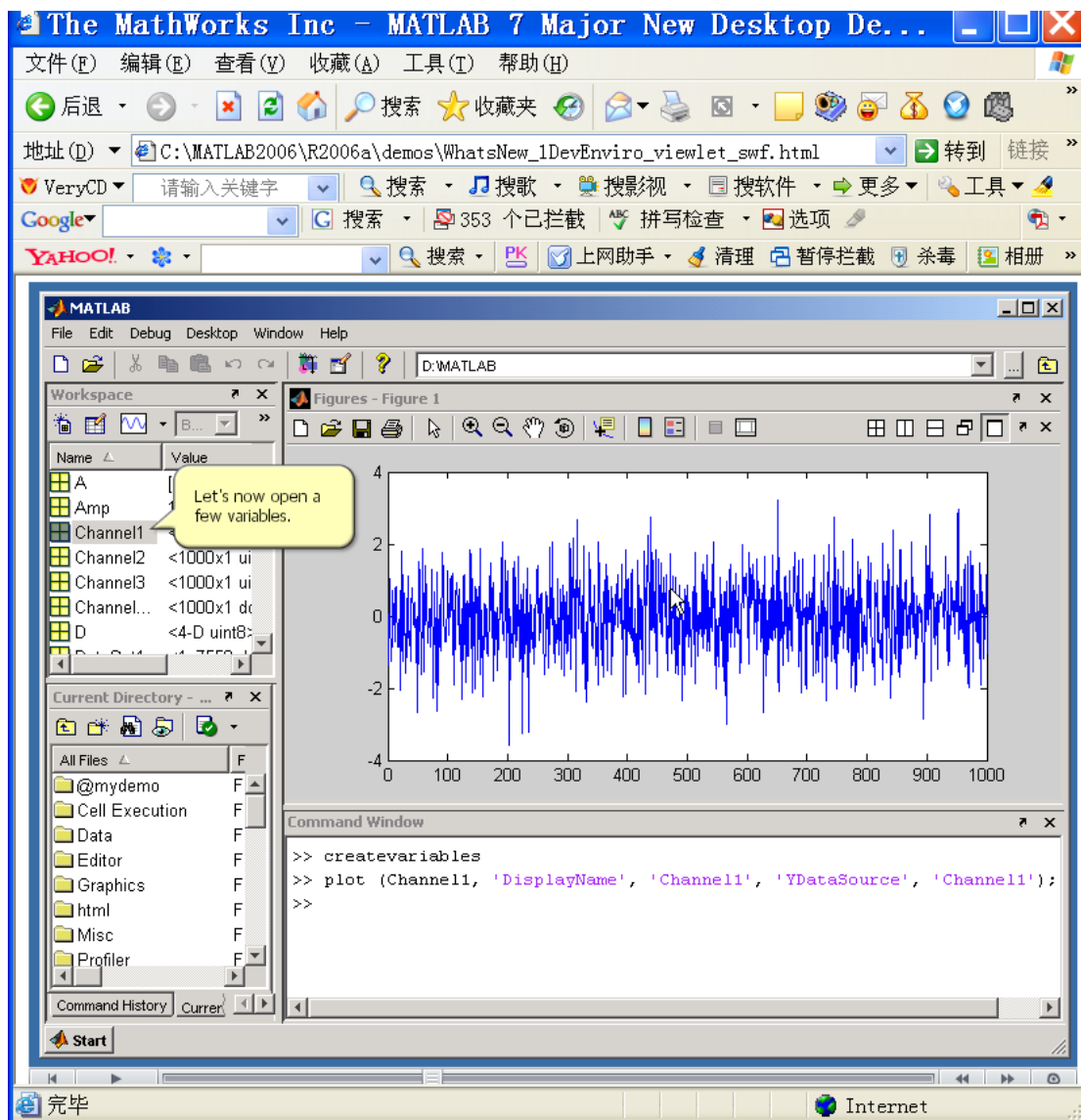


图 1.9-9 视频播放器

## 习题 1

1. 在安装 MATLAB 软件时，哪个组件是必须选择的？假如不“勾选”这个组件，那么就不可能建立 MATLAB 工作环境。
2. 数字 1.5e2, 1.5e3 中的哪个与 1500 相同？
3. 请指出如下 5 个变量名中，哪些是合法的？  
abcd-2 xyz\_3 3chan a 变量 ABCDefgh
4. 在 MATLAB 环境中，比 1 大的最小数是多少？
5. 设  $a = -8$ ，运行以下三条指令，问运行结果相同吗？为什么？  
 $w1 = a^{(2/3)}$

`w2=(a^2)^(1/3)`

`w3=(a^(1/3))^2`

6. 指令 `clear`, `clf`, `clc` 各有什么用处?
7. 以下两种说法对吗? (1) “MATLAB 的数值表达精度与其指令窗中的数据显示精度相同。” (2) MATLAB 指令窗中显示的数值有效位数不超过 7 位。”

8. 想要在 MATLAB 中产生二维数组  $S = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ , 下面哪些指令能实现目的?

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

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

`S=[1, 2, 3; 4, 5, 6; 7, 8, 9]` %整个指令在中文状态下输入

9. 试为例 1.3-5 编写一个解题用的 M 脚本文件?

## 第 1 章 符号计算

所谓符号计算是指: 解算数学表达式、方程不是在离散化的数值点上进行, 而是凭借一系列恒等式, 数学定理, 通过推理和演绎, 力求获得解析结果。这种计算建立在数值完全准确表达和推演严格解析的基础之上, 因此所得结果是完全准确的。

本书之所以把符号计算内容放在第 2 章, 是出于以下考虑: 一, 相对于 MATLAB 的数值计算“引擎”和“函数库”而言, 符号计算的“引擎”和“函数库”是独立的。二, 在相当一些场合, 符号计算解算问题的指令和过程, 显得比数值计算更自然、更简明。三, 大多数理工科的本科学生在学过高等数学和其他专业基础课以后, 比较习惯符号计算的解题理念和模式。

在编写本章时, 作者在充分考虑符号计算独立性的同时, 还考虑了章节的自完整性。为此, 本章不但全面地阐述符号计算, 而且在最后一节还详细叙述了符号计算结果的可视化。这样的安排, 将使读者在阅读完本章后, 就有可能运用 MATLAB 的符号计算能力去解决相当一些具体问题。

### 1.1 符号对象和符号表达式

#### 1.1.1 符号对象的创建和衍生

##### 一 生成符号对象的基本规则

##### 二 符号数字



【例 2.1-1】符号（类）数字与数值（类）数字之间的差异。

```
a=pi+sqrt(5)
sa=sym('pi+sqrt(5)')
Ca=class(a)
Csa=class(sa)
vpa(sa-a)
a =
    5.3777
sa =
    pi+sqrt(5)
Ca =
    double
Csa =
    sym
ans =
    .138223758410852e-16
```

### 三 符号参数

### 四 符号变量

【例 2.1-2】用符号计算研究方程  $uz^2 + vz + w = 0$  的解。

(1)

```
syms u v w z
Eq=u*z^2+v*z+w;
result_1=solve(Eq)

%

findsym(Eq,1)
result_1 =
    -u*z^2-v*z
ans =
    w
```

(2)

```
result_2=solve(Eq,z)
result_2 =
    1/2/u*(-v+(v^2-4*u*w)^(1/2))
    1/2/u*(-v-(v^2-4*u*w)^(1/2))
```

【例 2.1-3】对独立自由符号变量的自动辨认。

```
(1)
syms a b x X Y
k=sym('3');
z=sym('c*sqrt(delta)+y*sin(theta)');
EXPR=a*z*X+(b*x^2+k)*Y;
```

```
(2)
findsym(EXPR)
ans =
X, Y, a, b, c, delta, theta, x, y
```

```
(3)
findsym(EXPR,1)
ans =
x
```

```
(4)
findsym(EXPR,2),findsym(EXPR,3)
ans =
x,y
ans =
x,y,theta
```

【例 2.1-4】findsym 确定自由变量是对整个矩阵进行的。

```
syms a b t u v x y
A=[a+b*x,sin(t)+u;x*exp(-t),log(y)+v]
findsym(A,1)
A =
[ a+b*x, sin(t)+u]
[ x*exp(-t), log(y)+v]
ans =
x
```

## 1.1.2 符号计算中的算符

## 1.1.3 符号计算中的函数指令

## 1.1.4 符号对象的识别

【例 2.1-5】数据对象及其识别指令的使用。

(1)

```

clear
a=1;b=2;c=3;d=4;
Mn=[a,b;c,d]
Mc='[a,b;c,d] '
Ms=sym(Mc)
Mn =
     1     2
     3     4
Mc =
[a,b;c,d]
Ms =
[ a, b]
[ c, d]

```

(2)

```

SizeMn=size(Mn)
SizeMc=size(Mc)
SizeMs=size(Ms)
SizeMn =
     2     2
SizeMc =
     1     9
SizeMs =
     2     2

```

(3)

```

CMn=class(Mn)
CMc=class(Mc)
CMs=class(Ms)
CMn =
double
CMc =
char
CMs =
sym

```

(4)

```

isa(Mn,'double')
isa(Mc,'char')
isa(Ms,'sym')
ans =
     1
ans =
     1

```

```

ans =
    1

(5)
whos Mn Mc Ms
Name      Size      Bytes  Class

Mc        1x9        18    char array
Mn        2x2        32    double array
Ms        2x2        312   sym object

Grand total is 21 elements using 362 bytes

```

## 1.2 符号数字及表达式的操作

### 1.2.1 数值数字与符号数字之间的转换

- 一 数值数字向符号数字的转换
- 二 符号数字向双精度数字转换

### 1.2.2 符号数字的任意精度计算

【例 2.2-1】digits, vpa 指令的使用。

```

digits
p0=sym('(1+sqrt(5))/2')
pr=sym((1+sqrt(5))/2)
                                %
pd=sym((1+sqrt(5))/2,'d')
                                %

e32r=vpa(abs(p0-pr) )
e16=vpa(abs(p0-pd),16)
e32d=vpa(abs(p0-pd))

Digits = 32

p0 =
(1+sqrt(5))/2
pr =
7286977268806824*2^(-52)
pd =

```

```
1.6180339887498949025257388711907
e32r =
.543211520368251e-16
e16 =
0.
e32d =
.543211520368251e-16
```

## 1.2.3 符号表达式的基本操作

【例 2.2-2】简化  $f = \sqrt[3]{\frac{1}{x^3} + \frac{6}{x^2} + \frac{12}{x} + 8}$ 。

```
syms x
f=(1/x^3+6/x^2+12/x+8)^(1/3);
g1=simple(f)
g2=simple(g1)
g1 =
(2*x+1)/x
g2 =
2+1/x
```

## 1.2.4 表达式中的置换操作

### 一 子表达式置换操作

【例 2.2-3】对符号矩阵  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$  进行特征向量分解。

```
clear all
syms a b c d W
[V,D]=eig([a b;c d])
[RVD,W]=subexpr([V;D],W)
V =
[
-(1/2*d-1/2*a-1/2*(d^2-2*a*d+a^2+4*b*c)^(1/2))/c,
-(1/2*d-1/2*a+1/2*(d^2-2*a*d+a^2+4*b*c)^(1/2))/c]
[
1,
1]
D =
[
1/2*d+1/2*a+1/2*(d^2-2*a*d+a^2+4*b*c)^(1/2),
```

```

0]
[
1/2*d+1/2*a-1/2*(d^2-2*a*d+a^2+4*b*c)^(1/2)]
RVD =
[ -(1/2*d-1/2*a-1/2*W)/c, -(1/2*d-1/2*a+1/2*W)/c]
[
1, 1]
[ 1/2*d+1/2*a+1/2*W, 0]
[ 0, 1/2*d+1/2*a-1/2*W]
W =
(d^2-2*a*d+a^2+4*b*c)^(1/2)

```

## 二 通用置换指令

【例 2.2-4】用简单算例演示 subs 的置换规则。

(1)

```

syms a x;f=a*sin(x)+5
f =
a*sin(x)+5

```

(2)

```

f1=subs(f,'sin(x)',sym('y'))
%<2>
class(f1)
f1 =
a*y+5
ans =
sym

```

(3)

```

f2=subs(f,{a,x},{2,sym('pi/3')})
%<3>
class(f2)
f2 =
3^(1/2)+5
ans =
sym

```

(4)

```

f3=subs(f,{a,x},{2,pi/3})
%<4>

```

```
class(f3)
f3 =
    6.7321
ans =
double
```

```
(5)
f4=subs(subs(f,a,2),x,0:pi/6:pi)
    %<5>
class(f4)
f4 =
    5.0000    6.0000    6.7321    7.0000    6.7321    6.0000    5.0000
ans =
double
```

```
(6)
f5=subs(f,{a,x},{0:6,0:pi/6:pi})
    %<6>
class(f5)
f5 =
    5.0000    5.5000    6.7321    8.0000    8.4641    7.5000    5.0000
ans =
double
```

## 1.3 符号微积分

### 1.3.1 极限和导数的符号计算

【例 2.3-1】 试求  $\lim_{x \rightarrow \infty} \left(1 - \frac{1}{x}\right)^{kx}$ 。

```
syms x k
Lim_f=limit((1-1/x)^(k*x),x,inf)
Lim_f =
exp(-k)
```

【例 2.3-2】 求  $f = \begin{bmatrix} a & t^3 \\ t \cos x & \ln x \end{bmatrix}$  求  $\frac{df}{dx}$ ,  $\frac{d^2 f}{dt^2}$ ,  $\frac{d^2 f}{dt dx}$ 。

```
syms a t x;f=[a,t^3;t*cos(x), log(x)];
```

```

df=diff(f)
dfdt2=diff(f,t,2)
dfdxdt=diff(diff(f,x),t)
df =
[      0,      0]
[ -t*sin(x),  1/x]
dfdt2 =
[  0, 6*t]
[  0,  0]
dfdxdt =
[      0,      0]
[ -sin(x),  0]

```

【例 2.3-3】求  $\mathbf{f}(x_1, x_2) = \begin{bmatrix} x_1 e^{x_2} \\ x_2 \\ \cos(x_1) \sin(x_2) \end{bmatrix}$  的 Jacobian 矩阵  $\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} \end{bmatrix}$ 。

```

syms x1 x2;f=[x1*exp(x2);x2;cos(x1)*sin(x2)];
v=[x1 x2];fjac=jacobian(f,v)
fjac =
[      exp(x2),      x1*exp(x2)]
[      0,      1]
[ -sin(x1)*sin(x2),  cos(x1)*cos(x2)]

```

【例 2.3-4】 $f(x) = \sin|x|$ ，求  $f'_x(0)$ ， $f'_x(x)$ 。

(1)

```

clear
syms x
syms d positive
f_p=sin(x); %
df_p=limit((subs(f_p,x,x+d)-f_p)/d,d,0) % <4>
df_p0=limit((subs(f_p,x,d)-subs(f_p,x,0))/d,d,0) % <5>
df_p =
cos(x)
df_p0 =
1

```

(2)

```

f_n=sin(-x);
df_n=limit((f_n-subs(f_n,x,x-d))/d,d,0) % <7>

```



```

df_n0=limit((subs(f_n,x,0)-subs(f_n,x,-d))/d,d,0)           %<8>
df_n =
-cos(x)
df_n0 =
-1

(3)
f=sin(abs(x));
dfdx=diff(f,x)                                               %
    <10>
dfdx =
cos(abs(x))*abs(1,x)

(4)
xn=-3/2*pi:pi/50:0;xp=0:pi/50:3/2*pi;xnp=[xn,xp(2:end)];
hold on
plot(xnp,subs(f,x,xnp),'k','LineWidth',2.5)                 %   <13>
plot(xn,subs(df_n,x,xn),'--r','LineWidth',2.5)
plot(xp,subs(df_p,x,xp),' :r','LineWidth',2.5)
legend(char(f),char(df_n),char(df_p),'Location','NorthEast')% <16>
grid on
xlabel('x')
hold off

```

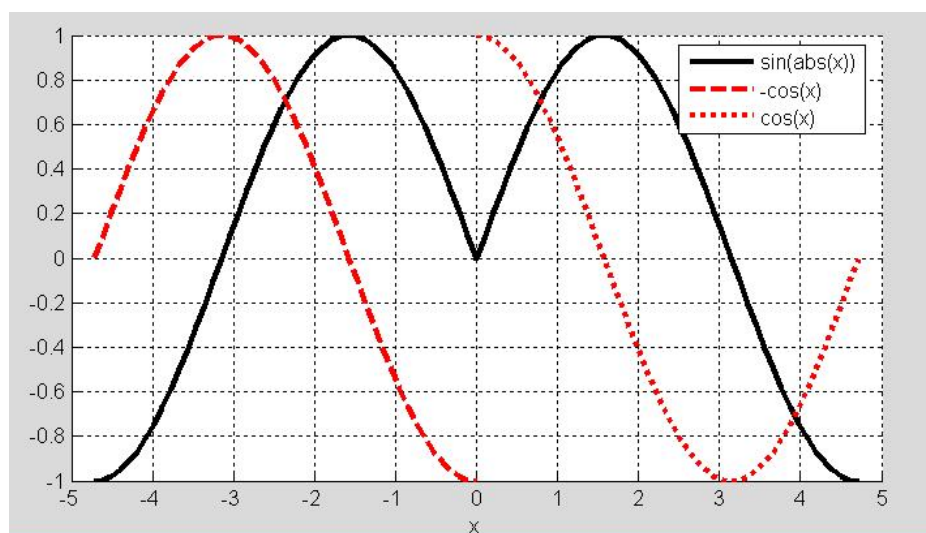


图 2.3-1 函数及其导函数

【例 2.3-5】设  $\cos(x + \sin y) = \sin y$ ，求  $\frac{dy}{dx}$ 。

```

(1)
clear

```

```

syms x
g=sym('cos(x+sin(y(x)))=sin(y(x))')           %           <3>
dgdxdx=diff(g,x)                                %           <4>
g =
cos(x+sin(y(x)))=sin(y(x))
dgdxdx =
-sin(x+sin(y(x)))*(1+cos(y(x))*diff(y(x),x)) = cos(y(x))*diff(y(x),x)

(2)
disp(maple('isolate',dgdxdx,diff('y(x)',x)))      %
<5>
diff(y(x),x)                                       =
sin(x+sin(y(x)))/(-sin(x+sin(y(x)))*cos(y(x))-cos(y(x)))

```

【例 2.3-6】求  $f(x) = xe^x$  在  $x = 0$  处展开的 8 阶 Maclaurin 级数。

```

syms x
r=taylor(x*exp(x),9,x,0)           %忽略 9 阶及 9 阶以上小量的展开
r =
x+x^2+1/2*x^3+1/6*x^4+1/24*x^5+1/120*x^6+1/720*x^7+1/5040*x^8

```

【例 2.3-7】求  $\sin(x^2 + y)$  在  $x = 0, y = 0$  处的截断 8 阶小量的 Taylor 展开近似。

```

TL1=maple('mtaylor(sin(x^2+y),[x=0,y=0],8)')
class(TL1)
TL1 =
y+x^2-1/6*y^3-1/2*x^2*y^2+1/120*y^5-1/2*x^4*y+1/24*x^2*y^4-1/6*x^6-1/
5040*y^7+1/12*x^4*y^3
ans =
char

```

### 1.3.2 序列/级数的符号求和

【例 2.3-8】求  $\sum_{t=0}^{t-1} [t, k^3], \sum_{k=1}^{\infty} \left[ \frac{1}{(2k-1)^2}, \frac{(-1)^k}{k} \right]$ 。

```

syms k t;f1=[t k^3];f2=[1/(2*k-1)^2,(-1)^k/k];
s1=simple(symsum(f1))
s2=simple(symsum(f2,1,inf))
s1 =
[ 1/2*t*(t-1),      k^3*t]
s2 =

```

```
[ 1/8*pi^2, -log(2)]
```

### 1.3.3 符号积分

【例 2.3-9】求  $\int \frac{1}{x} \sqrt{\frac{1+x}{x}} dx$ 。

```
clear
syms x
f=sqrt((1+x)/x)/x
s=int(f,x)
s=simple(simple(s))
f =
(1+x)/x^(1/2)/x
s =
((1+x)/x)^(1/2)/x*(-2*(x^2+x)^(3/2)+2*(x^2+x)^(1/2)*x^2+log(1/2+x+(x^
2+x)^(1/2))*x^2)/((1+x)*x)^(1/2)
s =
log(1/2+x+((1+x)*x)^(1/2))-2*((1+x)*x)^(1/2)/x
```

【例 2.3-10】求  $\int \begin{bmatrix} ax & bx^2 \\ \frac{1}{x} & \sin x \end{bmatrix} \cdot dx$ 。

```
syms a b x;f=[a*x,b*x^2;1/x,sin(x)];
disp('The integral of f is');pretty(int(f))
The integral of f is
```

```
[      2      3]
[1/2 a x    1/3 b x ]
[
[ log(x)    -cos(x) ]
```

【例 2.3-11】求积分  $\int_1^2 \int_{\sqrt{x}}^{x^2} \int_{\sqrt{xy}}^{x^2y} (x^2 + y^2 + z^2) dz dy dx$ 。

```
syms x y z
F2=int(int(int(x^2+y^2+z^2,z,sqrt(x*y),x^2*y),y,sqrt(x),x^2),x,1,2)
VF2=vpa(F2)
F2 =
-6072064/348075*2^(1/2)+64/225*2^(3/4)+14912/4641*2^(1/4)+1610027357/
6563700
VF2 =
224.92153573331143159790710032805
```

【例 2.3-12】求阿基米德（Archimedes）螺线  $r = a \cdot \theta$ , ( $a > 0$ ) 在  $\theta = 0$  到  $\varphi$  间的曲线长

度函数，并求出  $a = 1, \varphi = 2\pi$  时的曲线长度。

(1)

```
syms a r theta phi positive
x=r*cos(theta);x=subs(x,r,a*theta);
y=r*sin(theta);y=subs(y,r,a*theta);
dLdth=sqrt(diff(x,theta)^2+diff(y,theta)^2);
L=simple(int(dLdth,theta,0,phi))
L =
1/2*a*(phi*(1+phi^2)^(1/2)+asinh(phi))
```

(2)

```
L_2pi=subs(L,[a,phi],sym('[1,2*pi]'))
L_2pi_vpa=vpa(L_2pi)
L_2pi =
pi*(1+4*pi^2)^(1/2)+1/2*asinh(2*pi)
L_2pi_vpa =
21.256294148209098800702512272566
```

(3)

```
L1=subs(L,a,sym('1'));
ezplot(L1*cos(phi),L1*sin(phi),[0,2*pi])
grid on
hold on
x1=subs(x,a,sym('1'));
y1=subs(y,a,sym('1'));
h1=ezplot(x1,y1,[0,2*pi]);
set(h1,'Color','r','LineWidth',5)
title(' ')
legend('螺线长度-幅角曲线','阿基米德螺线')
hold off
```

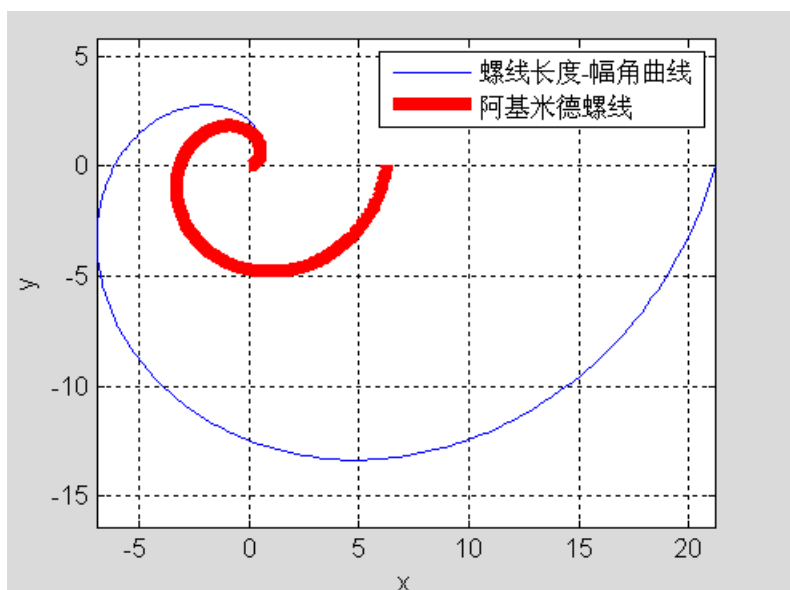


图 2.3-2 阿基米德螺线（粗红）和螺线长度函数（细蓝）

## 1.4 微分方程的符号解法

### 1.4.1 符号解法和数值解法的互补作用

### 1.4.2 求微分方程符号解的一般指令

### 1.4.3 微分方程符号解示例

【例 2.4-1】求  $\frac{dx}{dt} = y$ ,  $\frac{dy}{dt} = -x$  的解。

```
S=dsolve('Dx=y,Dy=-x')
disp([blanks(12),'x',blanks(21),'y'],disp([S.x,S.y])
S =
    x: [1x1 sym]
    y: [1x1 sym]
           x           y
[ -C1*cos(t)+C2*sin(t),  C1*sin(t)+C2*cos(t)]
```

【例 2.4-2】图示微分方程  $y = xy' - (y')^2$  的通解和奇解的关系。

```
y=dsolve('(Dy)^2-x*Dy+y=0','x') % <1>
```

```

clf,hold on,
ezplot(y(1),[-6,6,-4,8],1)
cc=get(gca,'Children');
set(cc,'Color','r','LineWidth',5)
for k=-2:0.5:2;ezplot(subs(y(2),'C1',k),[-6,6,-4,8],1);end, %
hold off,title('\fontname{隶书}\fontsize{16}通解和奇解')
y =
    1/4*x^2
    -C1^2+x*C1

```

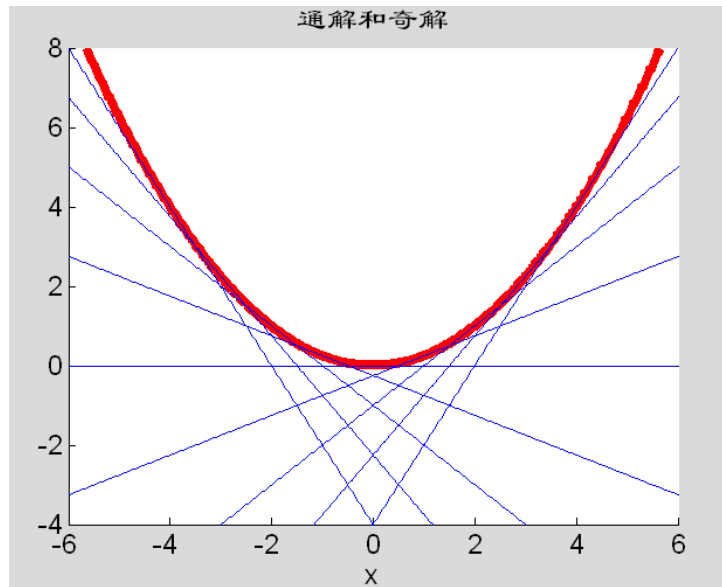


图 2.4-1 通解和奇解曲线

【例 2.4-3】求解两点边值问题：  $xy'' - 3y' = x^2$ ,  $y(1) = 0$ ,  $y(5) = 0$  。

(1)

```
y=dsolve('x*D2y-3*Dy=x^2','y(1)=0,y(5)=0','x')
```

y =

```
31/468*x^4-1/3*x^3+125/468
```

(2)

```

ezplot(y,[-1,6])
hold on
plot([1,5],[0,0],'.r','MarkerSize',20)
text(1,1,'y(1)=0')
text(4,1,'y(5)=0')
title(['x*D2y-3*Dy=x^2',' ', 'y(1)=0,y(5)=0'])
hold off

```

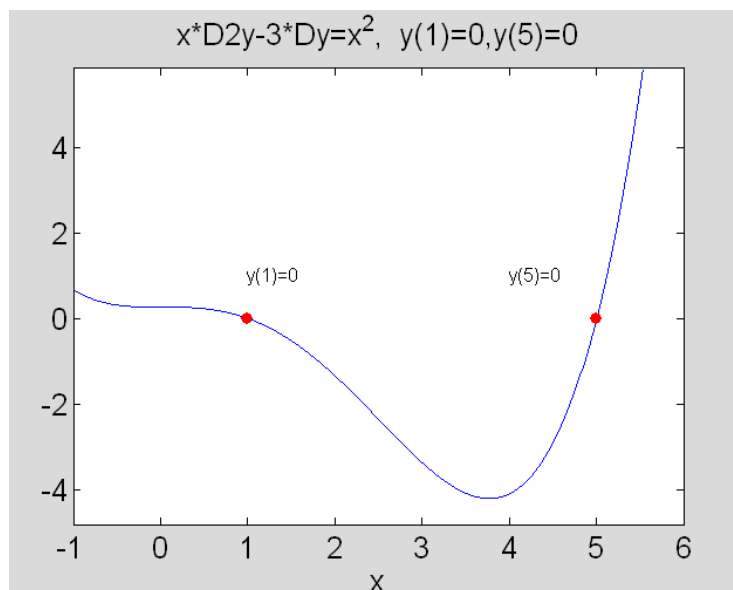


图 2.4-2 两点边值问题的解曲线

## 1.5 符号变换和符号卷积

### 1.5.1 Fourier变换及其反变换

【例 2.5-1】求  $f(t) = \begin{cases} 1 & t > 0 \\ 0 & t < 0 \end{cases}$  的 Fourier 变换。

(1)

```
syms t w
ut=heaviside(t);
UT=fourier(ut)
UT =
pi*dirac(w)-i/w
```

(2)

```
Ut=ifourier(UT,w,t)
Ut =
heaviside(t)
```

【例 2.5-2】根据 Fourier 变换定义，用积分指令求方波脉冲  $y = \begin{cases} A & -\tau/2 < t < \tau/2 \\ 0 & \text{else} \end{cases}$  的

Fourier 变换。

(1)

```

syms A t w
syms tao positive
yt=heaviside(t+tao/2)-heaviside(t-tao/2);
Yw=fourier(A*yt,t,w)
Yw =
2*A/w*sin(1/2*tao*w)

(2)
Yt=ifourier(Yw,w,t)
Yt =
A*(heaviside(t+1/2*tao)-heaviside(t-1/2*tao))

(3)
yt3=subs(yt,tao,3)
Yw3=subs(Yw,[A,tao],[1,3])
subplot(2,1,1)
Ht=ezplot(yt3,[-3,3]);
set(Ht,'Color','r','LineWidth',3)
subplot(2,1,2),ezplot(Yw3)
yt3 =
heaviside(t+3/2)-heaviside(t-3/2)
Yw3 =
2/w*sin(3/2*w)

```

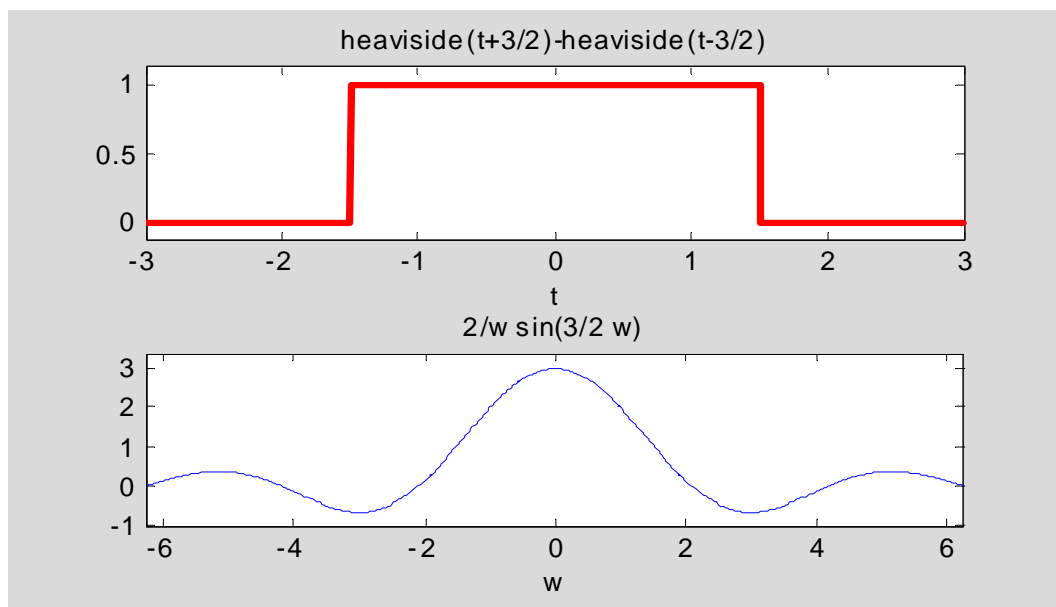


图 2.5-1 时域方波及其 Fourier 变换

【例 2.5-3】求  $f(t) = \begin{cases} e^{-(t-x)} & t \geq x \\ 0 & t < x \end{cases}$  的 Fourier 变换，在此  $x$  是参数， $t$  是时间变量。



```

syms t x w;
ft=exp(-(t-x))*heaviside(t-x);
F1=simple(fourier(ft,t,w))
F2=simple(fourier(ft))
F3=simple(fourier(ft,t))
F1 =
1/(1+i*w)/exp(i*x*w)
F2 =
i*exp(-i*t*w)/(i+w)
F3 =
-exp(-t*(2+i*t))/(-1+i*t)

```

## 1.5.2 Laplace变换及其反变换

【例 2.5-4】求  $\begin{bmatrix} \delta(t-a) & u(t-b) \\ e^{-at} \sin bt & t^2 e^{-t} \end{bmatrix}$  的 Laplace 变换。

```

syms t s;
syms a b positive % <2>
Dt=dirac(t-a);
Ut=heaviside(t-b);
Mt=[Dt,Ut;exp(-a*t)*sin(b*t),t^2*exp(-t)];MS=laplace(Mt,t,s)
MS =
[ exp(-s*a), exp(-s*b)/s]
[ 1/b/((s+a)^2/b^2+1), 2/(s+1)^3]

```

【例 2.5-5】验证 Laplace 时移性质： $L\{f(t-t_0)U(t-t_0)\} = e^{-st_0}L\{f(t)\} \quad t_0 > 0$ 。

```

syms t s
syms t0 positive
ft=sym('f(t-t0)')*heaviside(t-t0)
FS=laplace(ft,t,s)
FS_t=ilaplace(FS,s,t)
ft =
f(t-t0)*heaviside(t-t0)
FS =
exp(-s*t0)*laplace(f(t),t,s)
FS_t =
f(t-t0)*heaviside(t-t0)

```

## 1.5.3 Z变换及其反变换

【例 2.5-6】求序列  $f(n) = \begin{cases} 0 & n < 0 \\ 2 & n = 0 \\ 6(1-0.5^n) & n > 0 \end{cases}$  的 Z 变换，并用反变换验算。

(1)

```
syms n
Delta=sym('charfcn[0](n)');           %           <2>
D0=subs(Delta,n,0);                   %
D15=subs(Delta,n,15);                 %
disp(['D0,D15']);disp([D0,D15])
[D0,D15]
[ 1, 0]
```

(2)

```
syms z
fn=2*Delta+6*(1-(1/2)^n)
FZ=simple(ztrans(fn,n,z));
disp('FZ = ')
pretty(FZ)
FZ_n=iztrans(FZ,z,n)
fn =
2*charfcn[0](n)+6-6*(1/2)^n
FZ =
```

$$\frac{4z^2 + 2}{2z^2 - 3z + 1}$$

```
FZ_n =
2*charfcn[0](n)+6-6*(1/2)^n
```

## 1.5.4 符号卷积

【例 2.5-7】已知系统冲激响应，求  $u(t) = e^{-t}U(t)$  输入下的输出响应。

```
syms T t tao
ut=exp(-t);
ht=exp(-t/T)/T;
uh_tao=subs(ut,t,tao)*subs(ht,t,t-tao);
yt=simple(simple(int(uh_tao,tao,0,t)))
yt =
-(exp(-t)-exp(-t/T))/(T-1)
```

【例 2.5-8】采用 Laplace 变换和反变换求上例的输出响应。

```
syms s
yt=ilaplace(laplace(ut,t,s)*laplace(ht,t,s),s,t);
yt=simple(yt)
yt =
(-exp(-t)+exp(-t/T))/(T-1)
```

【例 2.5-9】求函数  $u(t) = U(t) - U(t-1)$  和的卷积。

```
syms tao
t=sym('t','positive');
ut=heaviside(t)-heaviside(t-1);
ht=t*exp(-t);
yt=int(subs(ut,t,tao)*subs(ht,t,t-tao),tao,0,t)
yt=collect(yt,'heaviside(t-1)')
yt =
1+t*heaviside(t-1)*exp(1-t)-heaviside(t-1)+(-t-1)/exp(t)
yt =
(exp(1-t)*t-1)*heaviside(t-1)+1+(-t-1)/exp(t)
```

## 1.6 符号矩阵分析和代数方程解

### 1.6.1 符号矩阵分析

【例 2.6-1】求矩阵的行列式、逆和特征根。

```
syms a11 a12 a21 a22
A=[a11,a12;a21,a22]
DA=det(A)
IA=inv(A)
EA=eig(A)
A =
[ a11, a12]
[ a21, a22]
DA =
a11*a22-a12*a21
IA =
[ a22/(a11*a22-a12*a21), -a12/(a11*a22-a12*a21)]
```

```
[ -a11/(a11*a22-a12*a21), a11/(a11*a22-a12*a21)]
EA =
1/2*a11+1/2*a22+1/2*(a11^2-2*a11*a22+a22^2+4*a12*a21)^(1/2)
1/2*a11+1/2*a22-1/2*(a11^2-2*a11*a22+a22^2+4*a12*a21)^(1/2)
```

【例 2.6-2】著名的 Givens 旋转（变换） $G = \begin{bmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{bmatrix}$  对矩阵  $A = \begin{bmatrix} \sqrt{3}/2 & 1/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix}$  的

旋转作用。

(1)

```
syms t
A=sym([sqrt(3)/2,1/2;1/2,sqrt(3)/2])
G=[cos(t),-sin(t);sin(t),cos(t)];
GA=G*A
A =
[ sqrt(3/4),      1/2]
[      1/2, sqrt(3/4)]
GA =
[ 1/2*cos(t)*3^(1/2)-1/2*sin(t), 1/2*cos(t)-1/2*sin(t)*3^(1/2)]
[ 1/2*sin(t)*3^(1/2)+1/2*cos(t), 1/2*sin(t)+1/2*cos(t)*3^(1/2)]
```

(2)

```
An=subs(GA,t,110/180*pi);
Op=[0;0];
Ad=double(A);
v1=[Op,Ad(:,1)]';v2=[Op,Ad(:,2)]';
u1=[Op,An(:,1)]';u2=[Op,An(:,2)]';
plot(v1(:,1),v1(:,2),'--k',v2(:,1),v2(:,2),'b')
axis([-1,1,-1,1]),axis square
hold on
hu=plot(u1(:,1),u1(:,2),'--k',u2(:,1),u2(:,2),'b');
set(hu,'LineWidth',4)
title('Givens Rotation')
legend(' v1',' v2',' u1',' u2','Location','South')
hold off
grid on
```

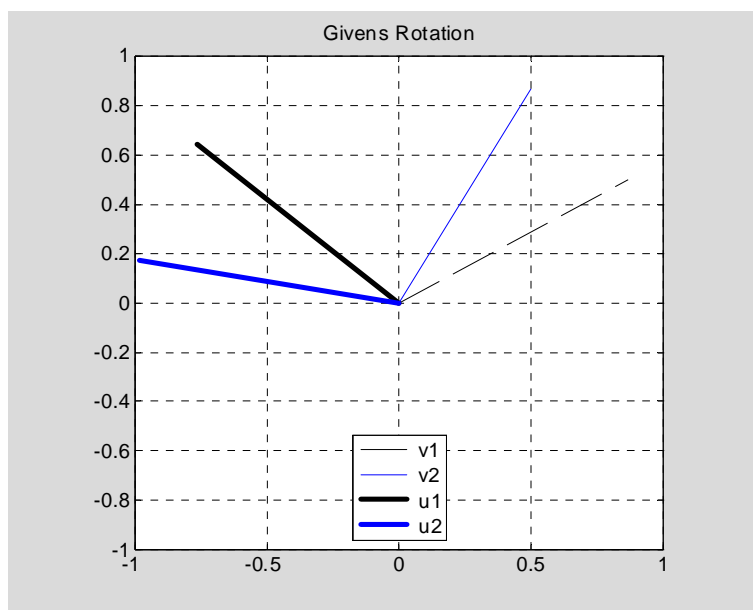


图 2.6-1 Givens 旋转的几何意义

【例 2.6-3】求  $d + \frac{n}{2} + \frac{p}{2} = q, n + d + q - p = 10, q + d - \frac{n}{4} = p, q + p - n - 8d = 1$  线性方程组的解。

(1)

```
A=sym([1 1/2 1/2 -1;1 1 -1 1;1 -1/4 -1 1;-8 -1 1 1]);
```

```
b=sym([0;10;0;1])
```

```
X1=A\b
```

```
b =
```

```
0
```

```
10
```

```
0
```

```
1
```

```
X1 =
```

```
1
```

```
8
```

```
8
```

```
9
```

(2)

```
S=solve('d+n/2+p/2-q','d+n-p+q-10','d-n/4-p+q','-8*d-n+p+q-1');
```

```
disp([' d',' n',' p',' q'])
```

```
disp([S.d,S.n,S.p,S.q])
```

```
 d n p q
```

```
[ 1, 8, 8, 9]
```

## 1.6.2 线性方程组的符号解

## 1.6.3 一般代数方程组的解

【例 2.6-4】求方程组  $uy^2 + vz + w = 0$ ,  $y + z + w = 0$  关于  $y, z$  的解。

```
S=solve('u*y^2+v*z+w=0','y+z+w=0','y','z')
disp('S.y'),disp(S.y),disp('S.z'),disp(S.z)
S =
    y: [2x1 sym]
    z: [2x1 sym]
S.y
-1/2/u*(-2*u*w-v+(4*u*w*v+v^2-4*u*w)^(1/2))-w
-1/2/u*(-2*u*w-v-(4*u*w*v+v^2-4*u*w)^(1/2))-w
S.z
1/2/u*(-2*u*w-v+(4*u*w*v+v^2-4*u*w)^(1/2))
1/2/u*(-2*u*w-v-(4*u*w*v+v^2-4*u*w)^(1/2))
```

【例 2.6-5】solve 指令求  $d + \frac{n}{2} + \frac{p}{2} = q$ ,  $n + d + q - p = 10$ ,  $q + d - \frac{n}{4} = p$  构成的“欠定”方程组解。

```
syms d n p q
eq1=d+n/2+p/2-q;eq2=n+d+q-p-10;eq3=q+d-n/4-p;
S=solve(eq1,eq2,eq3,d,n,p,q);
disp([' S.d',' S.n',' S.p',' S.q'])
disp([S.d,S.n,S.p,S.q])
Warning: 3 equations in 4 variables.
> In solve at 113
    In sym.solve at 49
    S.d    S.n    S.p    S.q
[      d,      8, 4*d+4, 3*d+6]
```

【例 2.6-6】求  $(x+2)^x = 2$  的解。

```
clear all,syms x;
s=solve('(x+2)^x=2','x')
s =
.69829942170241042826920133106081
```

## 1.7 利用MAPLE的深层符号计算资源

## 1.7.1 经典特殊函数的调用

【例 2.7-1】求  $\int_0^x \frac{1}{\ln t} dt$  ,  $x \in (0,1)$  。注意：被积函数在  $t=0$  无定义，在  $t=1^-$  处为负无穷。

(1)

```
syms t x
ft=1/log(t);
sx=int(ft,t,0,x)
sx =
-Ei(1,-log(x))
```

(2)

```
x=0.5:0.1:0.9
sx_n=-mfun('Ei',1,-log(x))
x =
    0.5000    0.6000    0.7000    0.8000    0.9000
sx_n =
   -0.3787   -0.5469   -0.7809   -1.1340   -1.7758
```

## 1.7.2 MAPLE库函数在线帮助的检索树

(1)

```
mhelp index
```

Index of help descriptions

Calling Sequence

```
?index[category] or help(index, category)
```

Description

- The following categories of topics are available in the help subsystem:

```
expression operators for forming expressions
function    list of Maple functions
misc        miscellaneous facilities
module      topics related to modules
packages    descriptions of library packages
procedure   topics related to procedures and programming
statement   list of Maple statements
```

To access these help pages, you must prefix the category with index, thus ?

index[category].

(2)

**mhhelp index[function]**

Index of descriptions for standard library functions

Description

- The following are the names of Maple's standard library functions. For more

information, see ?f where f is any of these functions.

about	abs	add	addcoords
additionally	addproperty	addressof	AFactor
AFactors	AIrreduc	AiryAi	AiryAiZeros
AiryBi	AiryBiZeros	algebraic	algsubs
alias	allvalues	anames	andmap
AngerJ	antihermitian	antisymm	apply
applyop	applyrule	arccos	arccosh
arccot	arccoth	arccsc	arccsch
arcsec	arcsech	arcsin	arcsinh
arctan	arctanh	argument	Array
array	ArrayDims	ArrayElems	ArrayIndFns
ArrayOptions	assemble	assign	assigned
asspar	assume	assuming	asympt
...	(注: 省略号是作者所加, 以省篇幅。)		
writeto	zero	Zeta	zip
ztrans	—		

See Also

libname, index[package]

(3)

## 1.7.3 发挥MAPLE的计算潜力

【例 2.7-2】求递推方程  $f(n) = -3f(n-1) - 2f(n-2)$  的通解。

(1)

**gs1:=maple('rsolve(f(n)=-3\*f(n-1)-2\*f(n-2),f(k));')**

**%<1>**

**gs1 =**



```
(-f(0)-f(1))*(-2)^k+(2*f(0)+f(1))*(-1)^k
```

(2)

```
gs2=maple('rsolve','f(n)=-3*f(n-1)-2*f(n-2)','f(k)') %<2>
```

```
gs2 =
```

```
(-f(0)-f(1))*(-2)^k+(2*f(0)+f(1))*(-1)^k
```

【例 2.7-3】求  $f = xyz$  的 Hessian 矩阵。

(1)

```
FH1=maple('hessian(x*y*z,[x,y,z]);')
```

```
FH1 =
```

```
matrix([[0, z, y], [z, 0, x], [y, x, 0]])
```

(2)

```
FH2=maple('hessian','x*y*z','[x,y,z]')
```

```
FH2 =
```

```
matrix([[0, z, y], [z, 0, x], [y, x, 0]])
```

(3)

```
FH=sym(FH2)
```

```
FH =
```

```
[ 0, z, y]
```

```
[ z, 0, x]
```

```
[ y, x, 0]
```

## 1.8 符号计算结果的可视化

### 1.8.1 直接可视化符号表达式

#### 一 单独立变量符号函数的可视化

【例 2.8-1】绘制  $y = \frac{2}{3}e^{-\frac{t}{2}}\cos\frac{\sqrt{3}}{2}t$  和它的积分  $s(t) = \int_0^t y(t)dt$  在  $[0, 4\pi]$  间的图形（图

2.8-1）。

```
syms t tao
```

```
y=2/3*exp(-t/2)*cos(sqrt(3)/2*t);
```

```
s=subs(int(y,t,0,tao),tao,t);
```

```
subplot(2,1,1)
```

```

%
ezplot(y,[0,4*pi]),ylim([-0.2,0.7])
grid on
subplot(2,1,2)
ezplot(s,[0,4*pi])
grid on
title('s = \int y(t)dt')

```

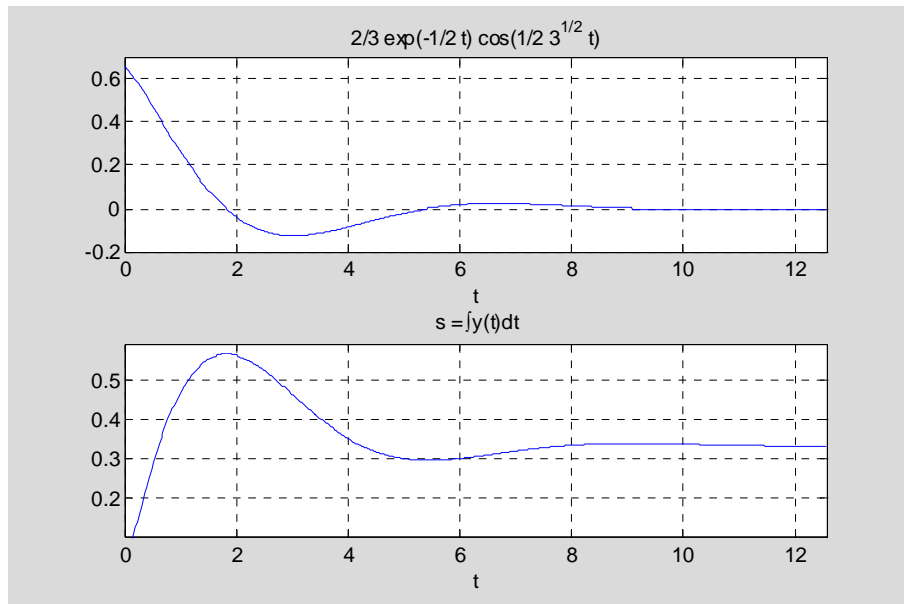


图 2.8-1 ezplot 使用示例

## 二 双独立变量符号函数的可视化

【例 2.8-2】借助可视化手段，加深 Taylor 展开的邻域近似概念。图形研究函数  $f(x, y) = \sin(x^2 + y)$  在  $x = 0, y = 0$  处的截断 8 阶小量的 Taylor 展开（例 2.3-7 讨论过）。

(1)

```

TL1=maple('mtaylor(sin(x^2+y),[x=0,y=0],8)')
TL1 =
y+x^2-1/6*y^3-1/2*x^2*y^2+1/120*y^5-1/2*x^4*y+1/24*x^2*y^4-1/6*x^6-1/
5040*y^7+1/12*x^4*y^3

```

(2)

```

Fxy='sin(x^2+y)'
Fxy_TL1=[Fxy, '-( ',TL1,')']
Fxy =
sin(x^2+y)
Fxy_TL1 =
sin(x^2+y)-(y+x^2-1/6*y^3-1/2*x^2*y^2+1/120*y^5-1/2*x^4*y+1/24*x^2*y^
4-1/6*x^6-1/5040*y^7+1/12*x^4*y^3)

```

(3)

```
clf  
ezsurf(Fxy,[-4,4,-6,6])  
shading interp  
view([-30,50])
```

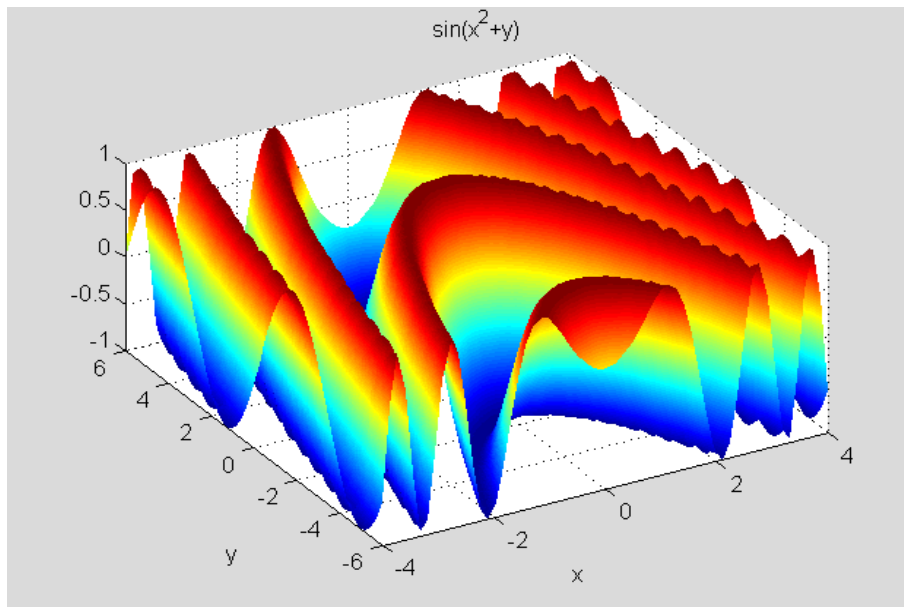


图 2.8-2 原函数在较大范围内的图形

(4)

```
clf  
ezsurf(TL1,[-4,4,-6,6])  
shading interp  
view([-30,50])
```

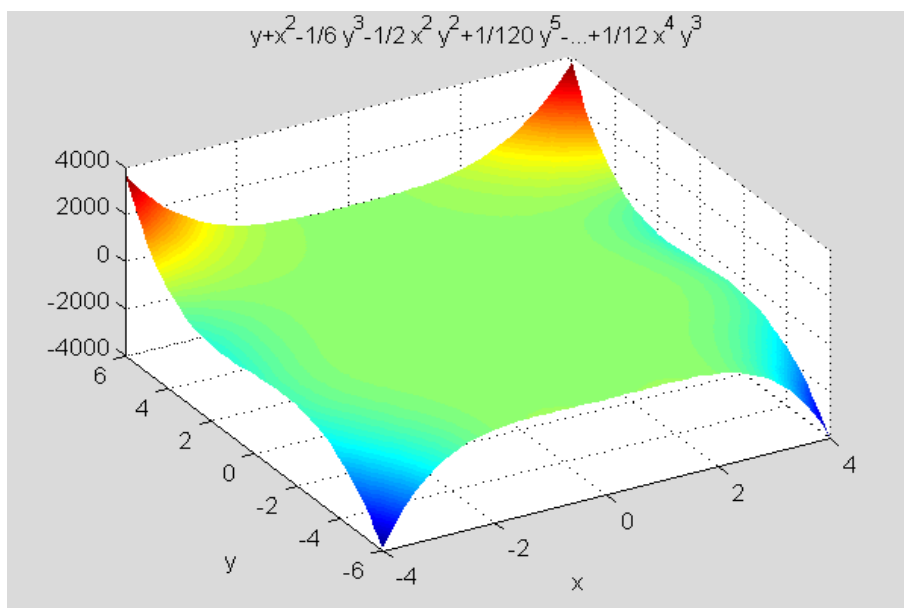


图 2.8-3 Taylor 展开在较大范围内的图形

```
(5)
clf
ezsurf(Fxy,[-0.5,0.5,-0.5,0.5],'circ')
shading interp
view([-75,15])
```

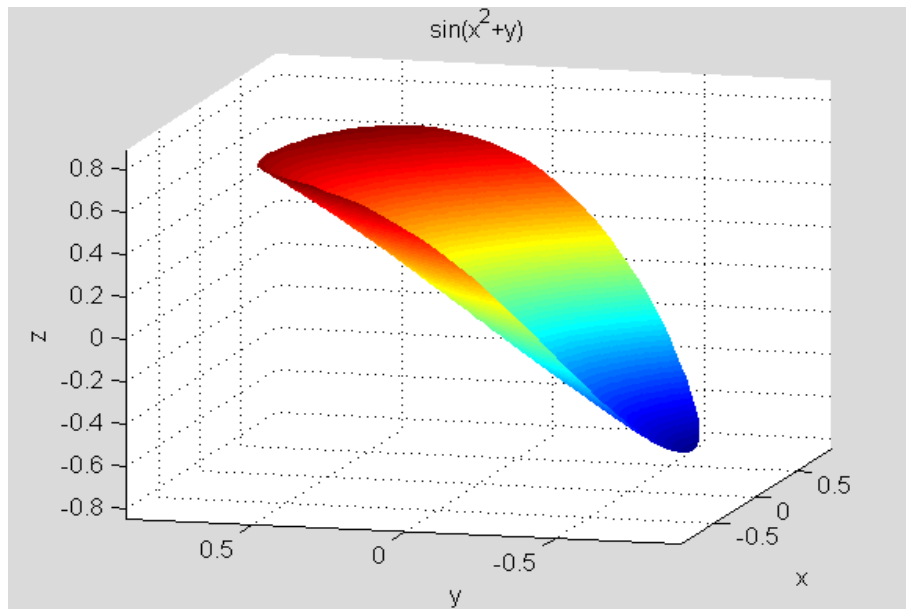


图 2.8-4 原函数在小范围内的图形

```
(6)
clf
ezsurf(TL1,[-0.5,0.5,-0.5,0.5],'circ')
shading interp
view([-75,15])
```

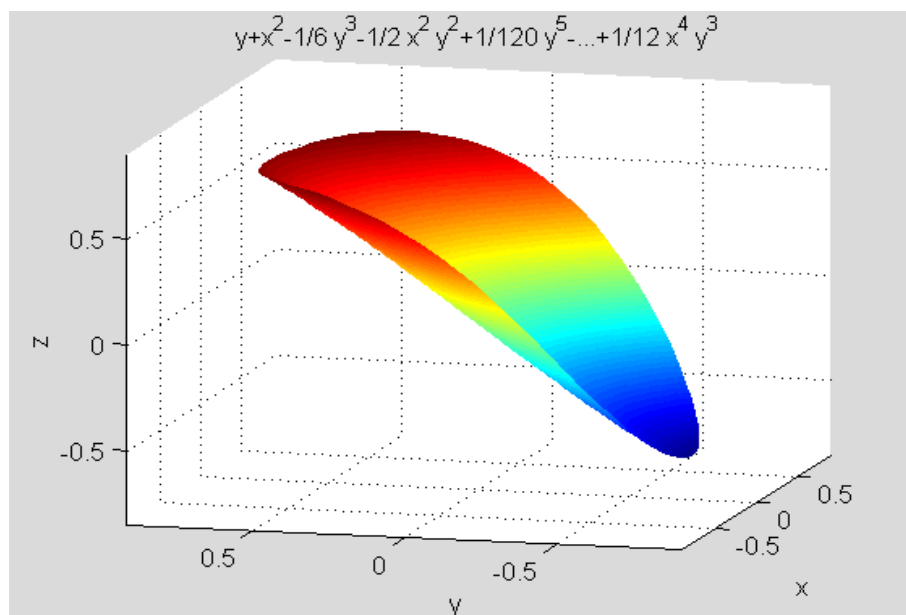


图 2.8-5 Taylor 展开在小范围内的图形

(7)

```
clf
ezsurf(Fxy_TL1,[-0.5,0.5],'circ')
shading interp
view([-50,10])
```

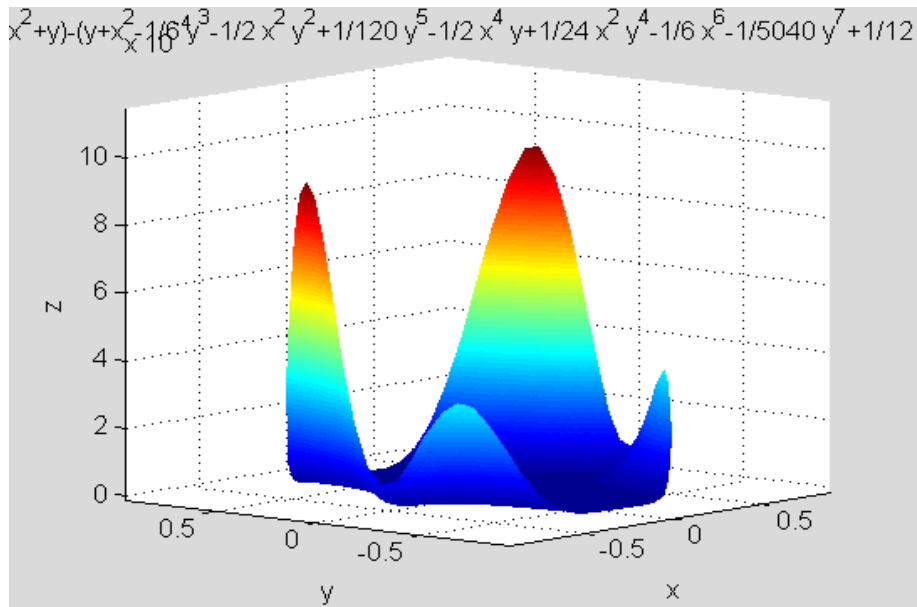


图 2.8-6 小范围内的误差曲面

【例 2.8-3】使用球坐标参量画部分球壳（图 2.8-7 ）。

```
x='cos(s)*cos(t)';
y='cos(s)*sin(t)';
z='sin(s)';
ezsurf(x,y,z,[0,pi/2,0,3*pi/2])
view(17,40)
shading interp
colormap(spring)
light('position',[0,0,-10],'style','local')
light('position',[-1,-0.5,2],'style','local')
material([0.5,0.5,0.5,10,0.3])
```

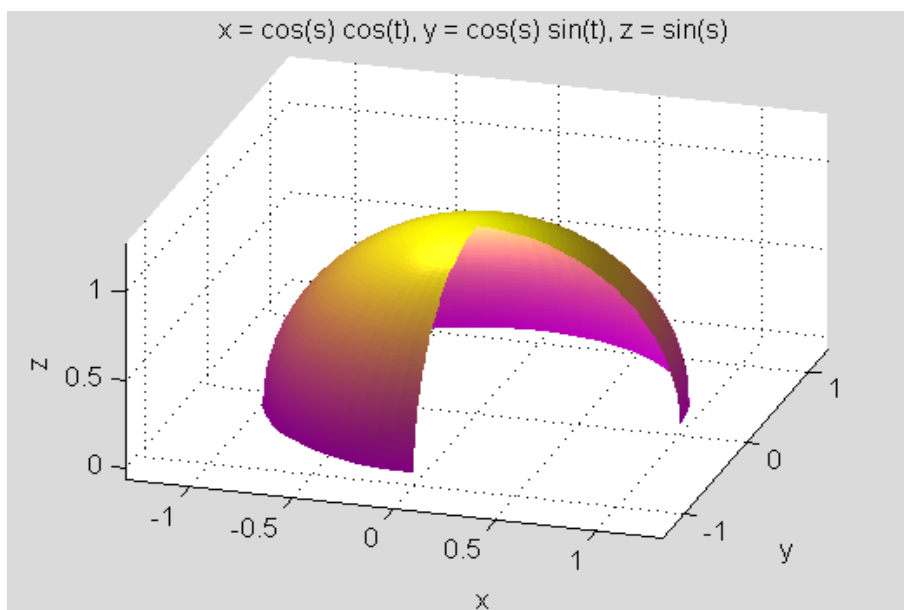


图 2.8-7 ezsurf 在参变量格式下绘制的图形

## 1.8.2 符号计算结果的数值化绘图

【例 2.8-4】在  $(0,1)$  区间，图示  $\frac{1}{\ln t}$  和  $\int_0^x \frac{1}{\ln t} dt$  （在例 2.7-1 中已计算过积分）。

(1)

```
syms t x
ft=1/log(t);
sx=int(ft,t,0,x)
sx =
-Ei(1,-log(x))
```

(2)

```
x=0.1:0.05:0.9;
sx_n=-mfun('Ei',1,-log(x));
```

(3) 图示被积函数和积分函数

```
clf
ezplot('1/log(t)',[0.1,0.9])
grid on
hold on
plot(x,sx_n,'LineWidth',3)
Char1='1/ln(t)';
Char2='{ \int_0^x 1/ln(t) dt';
title([Char1,' and ',Char2])
legend(Char1,Char2,'Location','SouthWest')
```



```

0.
v2 =
0.
v3 =
.1480297366166876e-15

```

```

(2) x=pi/3
a =
    1.0472
b =
pi/3
c =
1.0471975511965976313177861811710
d =
pi/3

```

```

v1 =
0.
v2 =
0.
v3 =
.1148364282799222e-15

```

```

(3) x=pi*3^(1/3)
a =
    4.5310
b =
5101408179057732*2^(-50)
c =
4.5309606547207899041040946030989
d =
pi*3^(1/3)

```

```

v1 =
.2660111416629094e-15
v2 =
.2660111416629094e-15
v3 =
.2660111416629094e-15

```

2. 说出以下三条指令产生的结果各属于哪种数据类型，是“双精度”对象，还是“符号”对象？

```

3/7+0.1, sym(3/7+0.1), vpa(sym(3/7+0.1))

```



〔答案〕

```
c1 =  
    0.5286  
c2 =  
37/70  
c3 =  
.52857142857142857142857142857143
```

3. 在不加专门指定的情况下，以下符号表达式中的哪一个变量被认为是独立自由变量。

`sym('sin(w*t)'), sym('a*exp(-X)'), sym('z*exp(j*theta)')`

〔答案〕

```
ans =  
w
```

```
ans =  
a
```

```
ans =  
z
```

4. 求符号矩阵  $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$  的行列式值和逆，所得结果应采用“子表达式置换”

简洁化。

〔答案〕

```
A =  
[ a11, a12, a13]  
[ a21, a22, a23]  
[ a31, a32, a33]
```

```
DA =  
a11*a22*a33-a11*a23*a32-a21*a12*a33+a21*a13*a32+a31*a12*a23-a31*a  
13*a22
```

```
IA =  
[(a22*a33-a23*a32)/d, -(a12*a33-a13*a32)/d,  
-(-a12*a23+a13*a22)/d]  
[-(a21*a33-a23*a31)/d, (a11*a33-a13*a31)/d,
```

```

-(a11*a23-a13*a21)/d]
[(-a21*a32+a22*a31)/d,                                -(a11*a32-a12*a31)/d,
(a11*a22-a12*a21)/d]

d =
a11*a22*a33-a11*a23*a32-a21*a12*a33+a21*a13*a32+a31*a12*a23-a31*a
13*a22

```

5. 对函数  $f(k) = \begin{cases} a^k & k \geq 0 \\ 0 & k < 0 \end{cases}$ , 当  $a$  为正实数时, 求  $\sum_{k=0}^{\infty} f(k)z^{-k}$ 。(实际上, 这就是根据定义求 Z 变换问题。)

【答案】

```

z1 =
-z/(-z+a)

```

6. 对于  $x > 0$ , 求  $\sum_{k=0}^{\infty} \frac{2}{2k+1} \left( \frac{x-1}{x+1} \right)^{2k+1}$ 。(提示: 理论结果为  $\ln x$ )

【答案】

```

s_ss =
log(x)
syms k x
f=2/(2*k+1)*((x-1)/(x+1))^(2*k+1)
s=simple(symsum(f,k,0,inf))
s =
log(x)

```

7. (1) 通过符号计算求  $y(t) = |\sin t|$  的导数  $\frac{dy}{dt}$ 。(2) 然后根据此结果, 求  $\frac{dy}{dt} \Big|_{t=0^-}$  和

$$\frac{dy}{dt} \Big|_{t=\frac{\pi}{2}}.$$

【答案】

```

d =
abs(1,sin(t))*cos(t)
d0_ =
-1
dpi_2 =

```

0

8. 求出  $\int_{-5\pi}^{1.7\pi} e^{-|x|} |\sin x| dx$  的具有 64 位有效数字的积分值。

〔答案〕

**si =**

**1.087849499412904913166671875948174520895458535212845987519414167**

9. 计算二重积分  $\int_1^2 \int_1^{x^2} (x^2 + y^2) dy dx$ 。

〔答案〕

**r =**

**1006/105**

10. 在  $[0, 2\pi]$  区间, 画出  $y(x) = \int_0^x \frac{\sin t}{t} dt$  曲线, 并计算  $y(4.5)$ 。

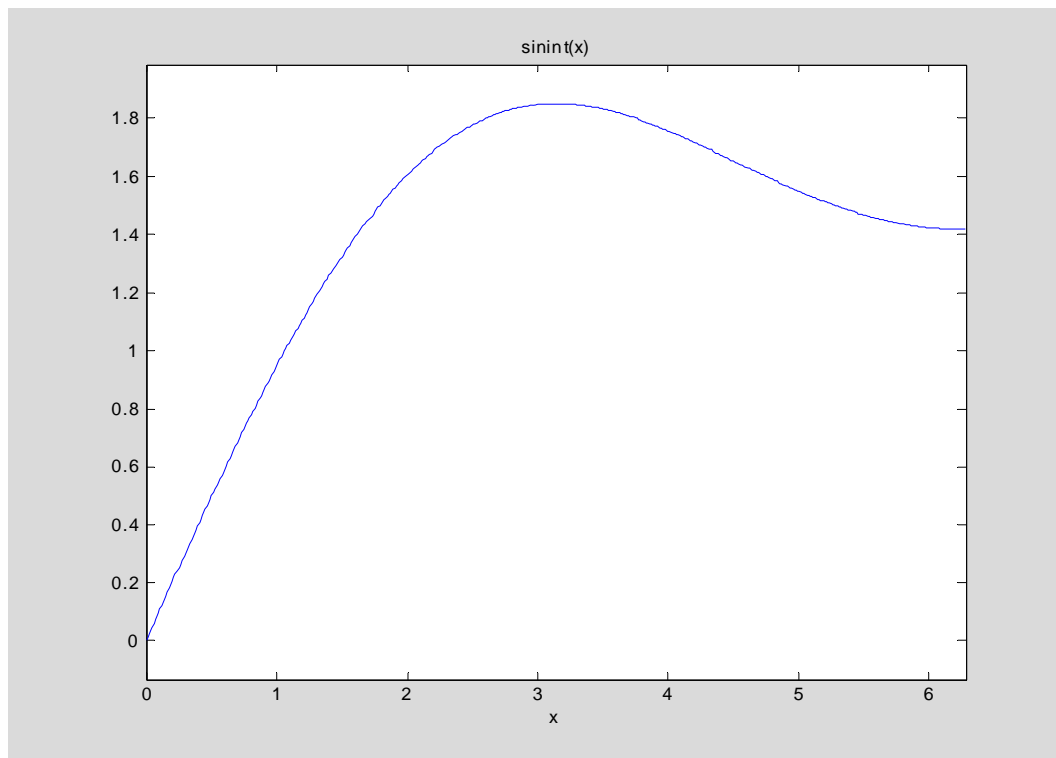
〔答案〕

**y =**

**sinint(x)**

**y5 =**

**1.6541404143792439835039224868515**



11. 计算  $y(n) = \int_0^{\pi/2} \sin^n x dx$  , 并求  $y(5)$  。

【答案】

```
yn =
1/2*pi^(1/2)*gamma(1/2+1/2*n)/gamma(1+1/2*n)
y5 =
8/15
```

12. 有序列  $x(k) = a^k$  ,  $h(k) = b^k$  , (在此  $k \geq 0$  ,  $a \neq b$  ) , 求这两个序列的卷积

$$y(k) = \sum_{n=0}^k h(n)x(k-n)。$$

【答案】

```
y1 =
(-(b/a)^(k+1)+1)*a^k*a/(-b+a)
或
y2 =
(-b*b^k+a^k*a)/(-b+a)
```

13. 设系统的冲激响应为  $h(t) = e^{-3t}$ ，求该系统在输入  $u(t) = \cos t$ ， $t \geq 0$  作用下的输出。

(提示：运用卷积进行计算)

〔答案〕

hut =  
-3/10/exp(t)^3+3/10\*cos(t)+1/10\*sin(t)

14. 求  $f(t) = Ae^{-\alpha|t|}$ ,  $\alpha > 0$  的 Fourier 变换。

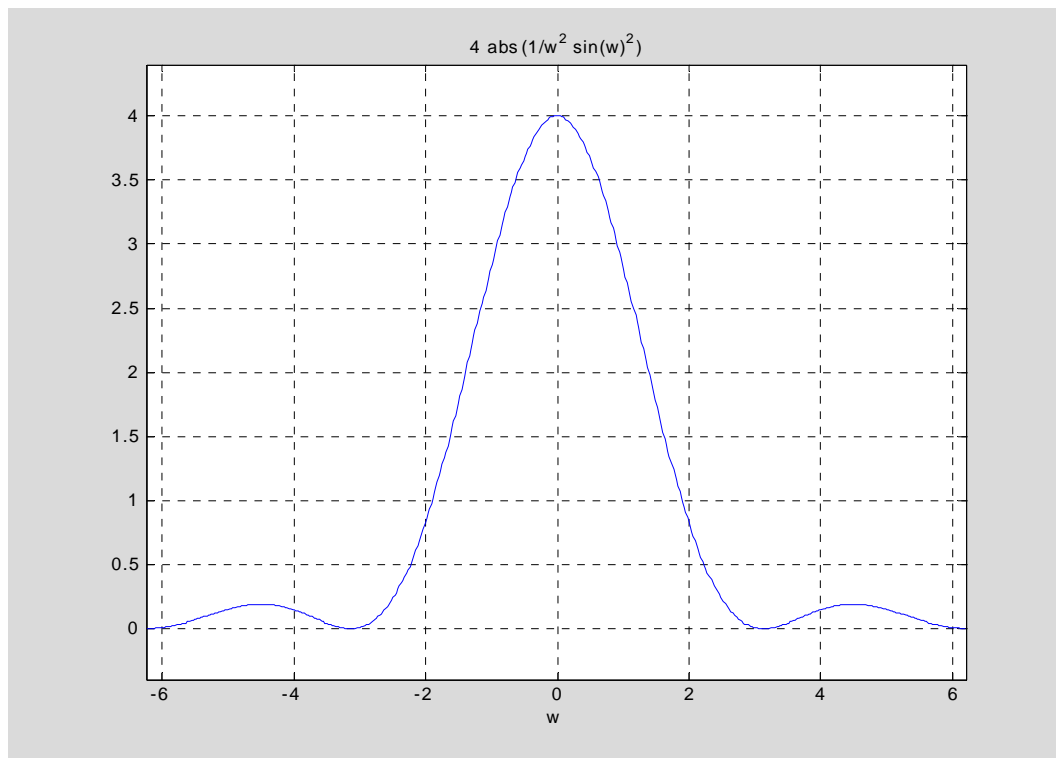
〔答案〕

F =  
2\*A\*a/(a^2+w^2)

15. 求  $f(t) = \begin{cases} A\left(1 - \frac{|t|}{\tau}\right) & |t| \leq \tau \\ 0 & |t| > \tau \end{cases}$  的 Fourier 变换，并画出  $A = 2, \tau = 2$  时的幅频谱。

〔答案〕

Fws =  
4\*A/tao/w^2\*sin(1/2\*tao\*w)^2  
Fw2 =  
4/w^2\*sin(w)^2



16. 求  $F(s) = \frac{s+3}{s^3+3s^2+6s+4}$  的 Laplace 反变换。

【答案】

**f =**

**1/3\*exp(-t)\*(-2\*cos(3^(1/2)\*t)+3^(1/2)\*sin(3^(1/2)\*t)+2)**

17. 利用符号运算证明 Laplace 变换的时域求导性质： $L\left[\frac{df(t)}{dt}\right] = s \cdot L[f(t)] - f(0)$ 。

【答案】

**Ldy =**

**s\*laplace(f(t),t,s)-f(0)**

18. 求  $f(k) = ke^{-\lambda kT}$  的 Z 变换表达式。

【答案】

**F\_z =**

$$z \cdot \exp(-\lambda T) / (z - \exp(-\lambda T))^2$$

19. 求方程  $x^2 + y^2 = 1, xy = 2$  的解。

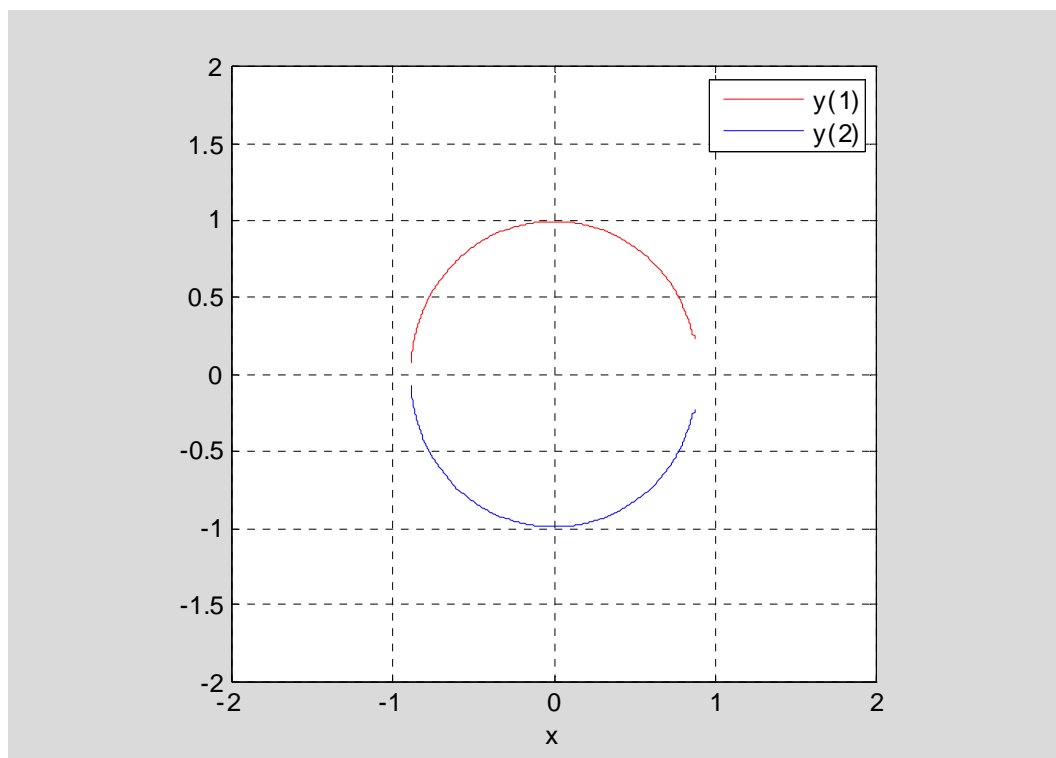
【答案】

$$\begin{aligned} x &= \\ &-1/2*(1/2*5^{(1/2)}+1/2*i*3^{(1/2)})^3+1/4*5^{(1/2)}+1/4*i*3^{(1/2)} \\ &-1/2*(1/2*5^{(1/2)}-1/2*i*3^{(1/2)})^3+1/4*5^{(1/2)}-1/4*i*3^{(1/2)} \\ &-1/2*(-1/2*5^{(1/2)}+1/2*i*3^{(1/2)})^3-1/4*5^{(1/2)}+1/4*i*3^{(1/2)} \\ &-1/2*(-1/2*5^{(1/2)}-1/2*i*3^{(1/2)})^3-1/4*5^{(1/2)}-1/4*i*3^{(1/2)} \\ y &= \\ &1/2*5^{(1/2)}+1/2*i*3^{(1/2)} \\ &1/2*5^{(1/2)}-1/2*i*3^{(1/2)} \\ &-1/2*5^{(1/2)}+1/2*i*3^{(1/2)} \\ &-1/2*5^{(1/2)}-1/2*i*3^{(1/2)} \end{aligned}$$

20. 求微分方程  $yy'/5 + x/4 = 0$  的通解，并绘制任意常数为 1 时解的图形。

【答案】

$$\begin{aligned} y &= \\ &1/2*(-5*x^2+4*C1)^{(1/2)} \\ &-1/2*(-5*x^2+4*C1)^{(1/2)} \\ yy &= \\ &1/2*(-5*x^2+4)^{(1/2)} \\ &-1/2*(-5*x^2+4)^{(1/2)} \end{aligned}$$



21. 求一阶微分方程  $x' = at^2 + bt$ ,  $x(0) = 2$  的解。

【答案】

$x =$

$$\frac{1}{3}at^3 + \frac{1}{2}bt^2 + 2$$

$$\frac{1}{3}at^3 + \frac{1}{2}bt^2 + 2$$

22. 求边值问题  $\frac{df}{dx} = 3f + 4g$ ,  $\frac{dg}{dx} = -4f + 3g$ ,  $f(0) = 0$ ,  $g(0) = 1$  的解。

【答案】

$f =$

$$\exp(3t) \sin(4t)$$

$g =$

$$\exp(3t) \cos(4t)$$

## 第 1 章 数值数组及向量化运算



数值数组（Numeric Array）和数组运算（Array Operations）始终是 MATLAB 的核心内容。本书从第 3 章起，全部注意力将集中于数值数组及其运算。

本章系统阐述：数组浮点算法的特点；一、二维数值数组的创建和寻访；数组运算和向量化编程；实现数组运算的基本函数；常用标准数组生成函数和数组构造技法；非数 NaN、“空”数组概念和应用；关系和逻辑操作。

## 1.1 数值计算的特点和地位

【例 3.1-1】已知  $f(t) = t^2 \cos t$ ，求  $s(x) = \int_0^x f(t) dt$ 。

（1）符号计算解法

```
syms t x
ft=t^2*cos(t)
sx=int(ft,t,0,x)
ft =
t^2*cos(t)
sx =
x^2*sin(x)-2*sin(x)+2*x*cos(x)
```

（2）数值计算解法

```
dt=0.05;
t=0:dt:5;
Ft=t.^2.*cos(t);
Sx=dt*cumtrapz(Ft);
t(end-4:end)
Sx(end-4:end)
plot(t,Sx,'.k','MarkerSize',12)
xlabel('x'),ylabel('Sx'),grid on
ans =
    4.8000    4.8500    4.9000    4.9500    5.0000
ans =
   -20.1144   -19.9833   -19.7907   -19.5345   -19.2131
```

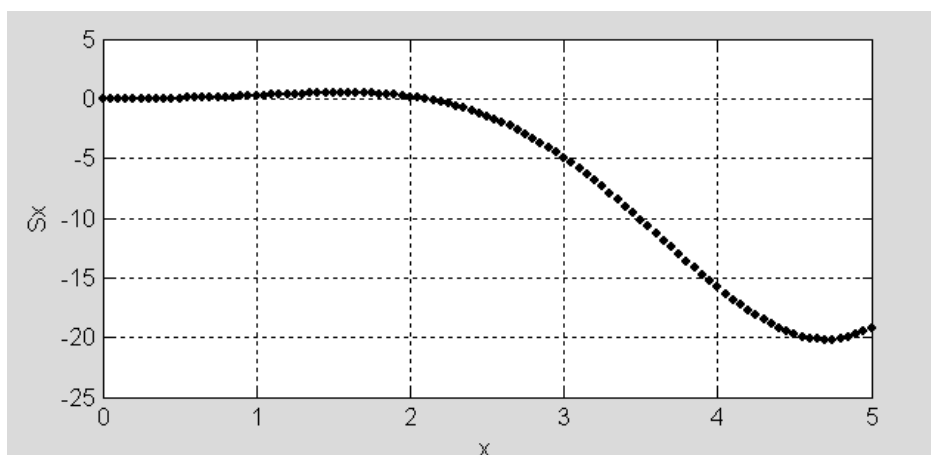


图 3.1-1 在区间[0,5]采样点上算得的定积分值

【例 3.1-2】已知  $f(t) = e^{-\sin(t)}$ ，求  $s(x) = \int_0^4 f(t) dt$ 。

(1) 符号计算解法

```
syms t x
ft=exp(-sin(t))
sx=int(ft,t,0,4)
ft =
exp(-sin(t))
Warning: Explicit integral could not be found.
> In sym.int at 58
sx =
int(exp(-sin(t)),t = 0 .. 4)
```

(2) 数值计算解法

```
dt=0.05;
t=0:dt:4;
Ft=exp(-sin(t));
Sx=dt*cumtrapz(Ft);
Sx(end)
plot(t,Ft,'*r','MarkerSize',4)
hold on
plot(t,Sx,'.k','MarkerSize',15)
hold off
xlabel('x')
legend('Ft','Sx')
ans =
3.0632
```

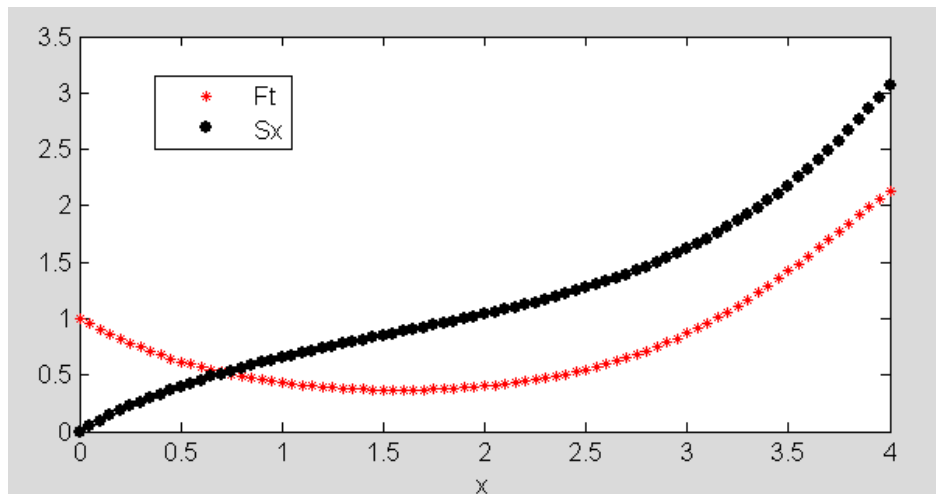


图 3.1-2 在区间[0, 4]中间的被积函数及其原函数的离散计算结果

## 1.2 数值数组的创建和寻访

### 1.2.1 一维数组的创建

#### 一 递增/减型一维数组的创建

#### 二 通用型一维数组的创建

【例 3.2-1】一维数组的常用创建方法举例。

```

a1=1:6
a2=0:pi/4:pi
a3=1:-0.1:0
a1 =
     1     2     3     4     5     6
a2 =
     0    0.7854    1.5708    2.3562    3.1416
a3 =
Columns 1 through 6
    1.0000    0.9000    0.8000    0.7000    0.6000    0.5000
Columns 7 through 11
    0.4000    0.3000    0.2000    0.1000         0

b1= linspace(0,pi,4)
b2=logspace(0,3,4)
b1 =
     0    1.0472    2.0944    3.1416
b2 =
     1     10    100   1000

```

```

c1=[2 pi/2 sqrt(3) 3+5i]
c1 =
    Columns 1 through 3
    2.0000    1.5708    1.7321
    Column 4
    3.0000 + 5.0000i

rand('state',0)
c2=rand(1,5)
c2 =
    0.9501    0.2311    0.6068    0.4860    0.8913

```

## 1.2.2 二维数组的创建

### 一 小规模数组的直接输入法

【例 3.2-2】在 MATLAB 环境下，用下面三条指令创建二维数组 C。

```

a=2.7358; b=33/79;
C=[1,2*a+i*b,b*sqrt(a);sin(pi/4),a+5*b,3.5+i]

C =
    1.0000    5.4716 + 0.4177i    0.6909
    0.7071    4.8244    3.5000 + 1.0000i

```

### 二 中规模数组的数组编辑器创建法

【例 3.2-3】根据现有数据创建一个 (3×8) 的数组。

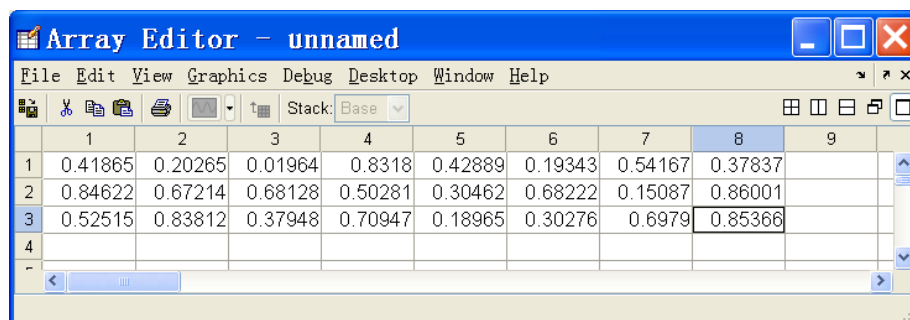


图 3.2-1 利用数组编辑器创建中规模数组

### 三 中规模数组的M文件创建法

【例 3.2-4】创建和保存数组 AM 的 MyMatrix.m 文件。

- (1) 打开文件编辑调试器，并在空白填写框中输入所需数组（见图 3.2-2）。
- (2) 最好，在文件的首行，编写文件名和简短说明，以便查阅（见图 3.2-2）。
- (3) 保存此文件，并且文件起名为 MyMatrix.m 。
- (4) 以后只要在 MATLAB 指令窗中，运行 MyMatrix.m 文件，数组 AM 就会自动生成于 MATLAB 内存中。

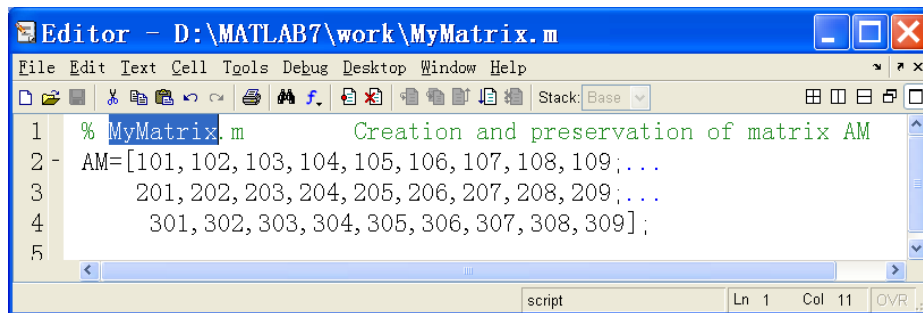


图 3.2-2 利用 M 文件创建数组

### 四 利用MATLAB函数创建数组

【例 3.2-5】标准数组产生的演示。

```
ones(2,4)
```

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

```
randn('state',0)
```

```
randn(2,3)
```

```
ans =  
   -0.4326    0.1253   -1.1465  
   -1.6656    0.2877    1.1909
```

```
D=eye(3)
```

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

```
diag(D)
```

```
ans =  
     1  
     1  
     1
```

```

diag(diag(D))
ans =
    1     0     0
    0     1     0
    0     0     1

randsrc(3,20,[-3,-1,1,3],1)
ans =
Columns 1 through 10
    3     1     3    -1    -3     3    -3    -3    -1     3
    1     3     1     3    -1     1     1     1     1     1
    3    -1    -3    -1     1    -1     3    -1    -1     1
Columns 11 through 20
   -1    -1    -3     1     1    -3     3    -1     1     3
    1     1     3    -1    -1    -3    -1    -1     1    -3
    1    -1    -3     3    -1    -1     3    -1     3     3

```

### 1.2.3 二维数组元素的标识和寻访

【例 3.2-6】本例演示：数组元素及子数组的各种标识和寻访格式；冒号的使用；end 的作用。

```

A=zeros(2,6)
A(:)=1:12
A =
    0     0     0     0     0     0
    0     0     0     0     0     0
A =
    1     3     5     7     9    11
    2     4     6     8    10    12

A(2,4)
A(8)
ans =
    8
ans =
    8

A(:,[1,3])
A([1,2,5,6]')
ans =
    1     5
    2     6
ans =
    1
    2

```

```

5
6

A(:,4:end)
ans =
    7     9    11
    8    10    12

A(2,1:2:5)=[-1,-3,-5]
A =
     1     3     5     7     9    11
    -1     4    -3     8    -5    12

B=A([1,2,2,2],[1,3,5] )
B =
     1     5     9
    -1    -3    -5
    -1    -3    -5
    -1    -3    -5

L=A<3
A(L)=NaN
L =
     1     0     0     0     0     0
     1     0     1     0     1     0
A =
    NaN     3     5     7     9    11
    NaN     4    NaN     8    NaN    12

```

## 1.2.4 数组构造技法综合

【例 3.2-7】数组操作函数 reshape, diag, repmat 的用法；空阵 [] 删除子数组的用法。

```

a=1:8
A=reshape(a,4,2)
A=reshape(A,2,4)
a =
     1     2     3     4     5     6     7     8
A =
     1     5
     2     6
     3     7
     4     8

```

```

A =
    1     3     5     7
    2     4     6     8

b=diag(A)
B=diag(b)
b =
    1
    4
B =
    1     0
    0     4

D1= repmat(B,2,4)
D1 =
    1     0     1     0     1     0     1     0
    0     4     0     4     0     4     0     4
    1     0     1     0     1     0     1     0
    0     4     0     4     0     4     0     4

D1([1,3],:)=[]
D1 =
    0     4     0     4     0     4     0     4
    0     4     0     4     0     4     0     4

```

【例 3.2-8】函数 flipud, fliplr, rot90 对数组的操作体现着“矩阵变换”。

```
A=reshape(1:9,3,3)
```

```

A =
    1     4     7
    2     5     8
    3     6     9

```

```
B=flipud(A)
```

```

B =
    3     6     9
    2     5     8
    1     4     7

```

```
C=fliplr(A)
```

```

C =
    7     4     1
    8     5     2
    9     6     3

```



```
D=rot90(A,2)
D =
     9     6     3
     8     5     2
     7     4     1
```

## 1.3 数组运算

### 1.3.1 数组运算的由来和规则

- 一 函数关系数值计算模型的分类
- 二 提高程序执行性能的三大措施
- 三 数组运算规则
- 四 数组运算符及数组运算函数

### 1.3.2 数组运算和向量化编程

【例 3.3-1】欧姆定律： $r = \frac{u}{i}$ ，其中  $r, u, i$  分别是电阻（欧姆）、电压（伏特）、电流（安培）。验证实验：据电阻两端施加的电压，测量电阻中流过的电流，然后据测得的电压、电流计算平均电阻值。（测得的电压电流具体数据见下列程序）。

（1）非向量化程序

```
clear
vr=[0.89, 1.20, 3.09, 4.27, 3.62, 7.71, 8.99, 7.92, 9.70, 10.41];
ir=[0.028, 0.040, 0.100, 0.145, 0.118, 0.258, 0.299, 0.257, 0.308, 0.345];

% -----
L=length(vr);
for k=1:L
    r(k)=vr(k)/ir(k);
end
% -----
sr=0;
for k=1:L
    sr=sr+r(k);
end
rm=sr/L
rm =
```

30.5247

(2) 向量化程序

```
clear
vr=[0.89, 1.20, 3.09, 4.27, 3.62, 7.71, 8.99, 7.92, 9.70, 10.41];
ir=[0.028, 0.040, 0.100, 0.145, 0.118, 0.258, 0.299, 0.257, 0.308, 0.345];

r=vr./ir
rm=mean(r)
r =
    31.7857    30.0000    30.9000    29.4483    30.6780    29.8837    30.0669
    30.8171    31.4935    30.1739
rm =
    30.5247
```

【例 3.3-2】用间距为 0.1 的水平线和垂直线均匀分割  $x \in [-5,5]$ ,  $y \in [-2.5,2.5]$  的矩形域,

在所有水平线和垂直线交点上计算函数  $z = \sin(xy)$  的值, 并图示。

(1)

```
clear
x=-5:0.1:5;
y=(-2.5:0.1:2.5)';
N=length(x);
M=length(y);
for ii=1:M
    for jj=1:N
        X0(ii,jj)=x(jj);
        Y0(ii,jj)=y(ii);
        Z0(ii,jj)=sin(abs(x(jj)*y(ii)));
    end
end
```

(2)

```
[X,Y]=meshgrid(x,y);
Z=sin(abs(X.*Y));
```

(3)

```
norm(Z-Z0)
ans =
    0
```

(4)

```
surf(X,Y,Z)
```

```

xlabel('x')
ylabel('y')
shading interp
view([190,70])

```

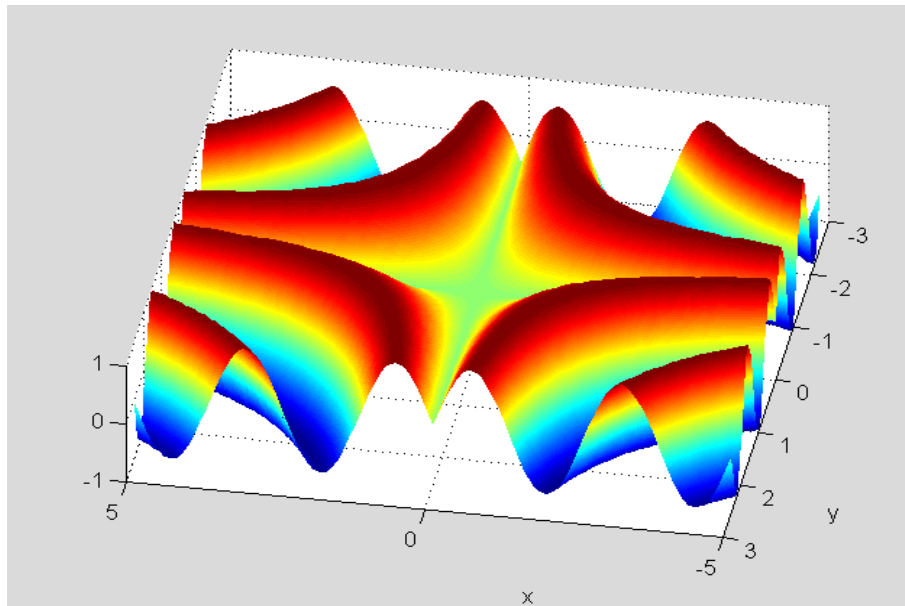


图 3.3-1 指定域上的二元函数图形

## 1.4 “非数”和“空”数组

### 1.4.1 非数NaN

【例 3.4-1】非数的产生和性质演示。

(1) 非数的产生

```
a=0/0,b=0*log(0),c=inf-inf
```

Warning: Divide by zero.

```
a =
```

```
NaN
```

Warning: Log of zero.

```
b =
```

```
NaN
```

```
c =
```

```
NaN
```

(2) 非数的传递性

```
0*a,sin(a)
```

```
ans =
```

```
NaN
```

```
ans =
```

```
NaN
```

### (3) 非数的属性判断

```
class(a)
isnan(a)
ans =
double
ans =
     1
```

【例 3.4-2】非数元素的寻访。

```
rand('state',0)
R=rand(2,5);R(1,5)=NaN;R(2,3)=NaN
R =
    0.9501    0.6068    0.8913    0.4565    NaN
    0.2311    0.4860     NaN    0.0185    0.4447

LR=isnan(R)
LR =
     0     0     0     0     1
     0     0     1     0     0

si=find(LR)
[rj,cj]=ind2sub(size(R),si)
[ri,ci]=find(LR)
disp('非数在二维数组 R 中的位置')
disp(['单下标时的第',int2str(si(1)),'和第',int2str(si(2)),'个元素'])
si =
     6
     9
ri =
     2
     1
ci =
     3
     5
rj =
     2
     1
cj =
     3
     5
非数在二维数组 R 中的位置
```

单下标时的第 6 和第 9 个元素

## 1.4.2 “空”数组

【例 3.4-3】关于“空”数组的算例。

(1) 创建“空”数组的几种方法

```
a=[]
b=ones(2,0),c=zeros(2,0),d=eye(2,0)
f=rand(2,3,0,4)
a =
    []
b =
    Empty matrix: 2-by-0
c =
    Empty matrix: 2-by-0
d =
    Empty matrix: 2-by-0
f =
    Empty array: 2-by-3-by-0-by-4
```

(2) “空”数组的属性

```
class(a)
isnumeric(a)
isempty(a)
ans =
double
ans =
    1
ans =
    1

which a
ndims(a)
size(a)
a is a variable.
ans =
    2
ans =
    0    0
```

(3) “空”数组用于子数组的删除和大数组的大小收缩

```
A=reshape(-4:5,2,5)
A =
    -4    -2     0     2     4
```

```

-3    -1    1    3    5

A(:,[2,4])=[]
A =
    -4     0     4
    -3     1     5

```

## 1.5 关系操作和逻辑操作

### 1.5.1 关系操作

【例 3.5-1】关系运算示例。

```

A=1:9,B=10-A
r0=(A<4)
r1=(A==B)
A =
     1     2     3     4     5     6     7     8     9
B =
     9     8     7     6     5     4     3     2     1
r0 =
     1     1     1     0     0     0     0     0     0
r1 =
     0     0     0     0     1     0     0     0     0

```

【例 3.5-2】关系运算应用。

```

t=-3*pi:pi/10:3*pi;
y=sin(t)./t;
tt=t+(t==0)*eps;
yy=sin(tt)./tt;
subplot(1,2,1),plot(t,y),axis([-9,9,-0.5,1.2]),
xlabel('t'),ylabel('y'),title('残缺图形')
subplot(1,2,2),plot(tt,yy),axis([-9,9,-0.5,1.2])
xlabel('tt'),ylabel('yy'),title('正确图形')
Warning: Divide by zero.

```

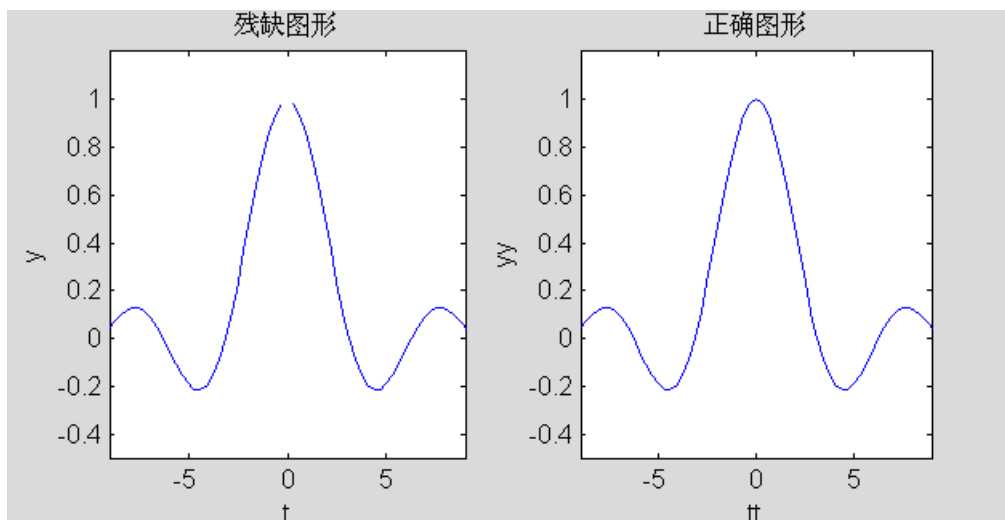


图 3.5-1 采用近似极限处理前后的图形对照

## 1.5.2 逻辑操作

【例 3.5-3】逻辑操作和关系操作。本例演示：逻辑、关系操作的组合；xor 的作用。

(1) 逻辑、关系操作的组合

**A=[-2,-1,0,0,1,2,3]**

**L1=~(A>1)**                      %判断 A 中, 哪些元素不大于 1

**L2=(A>0)&(A<2)**                %判断 A 中, 哪些元素大于 0 且小于 3

```
A =
    -2    -1     0     0     1     2     3
L1 =
     1     1     1     1     1     0     0
L2 =
     0     0     0     0     1     0     0
```

(2) xor 的作用

**A,B=[0,-1,1,0,1,-2,-3]**

**C=xor(A,B)** %当 A, B 数组中, 两个对应元素中仅一个为 0 时, 给出 1。否则为 0。

```
A =
    -2    -1     0     0     1     2     3
B =
     0    -1     1     0     1    -2    -3
C =
     1     0     1     0     0     0     0
```

【例 3.5-4】试绘制如图 3.5-2 最下那幅子图所示的“正弦波  $\sin t$  的削顶半波整流波形”，削顶发生在每个周期的  $[60^\circ, 120^\circ]$  之间。

```

clear,t=linspace(0,3*pi,500);y=sin(t);
z1=((t<pi)|(t>2*pi)).*y;
w=(t>pi/3&t<2*pi/3)+(t>7*pi/3&t<8*pi/3);
wn=~w;
z2=w*sin(pi/3)+wn.*z1;
subplot(4,1,1),plot(t,y,'r'),axis([0,10,-1.5,1.5])
ylabel('y'),grid on
subplot(4,1,2),plot(t,z1,'r'),axis([0,10,-0.2,1.5]),ylabel('z1')
subplot(4,1,3),plot(t,wn,'r'),axis([0,10,-0.2,1.5]),ylabel('wn')
subplot(4,1,4),plot(t,z2,'-b'),axis([0,10,-0.2,1.5]),ylabel('z2')
xlabel('t')

```

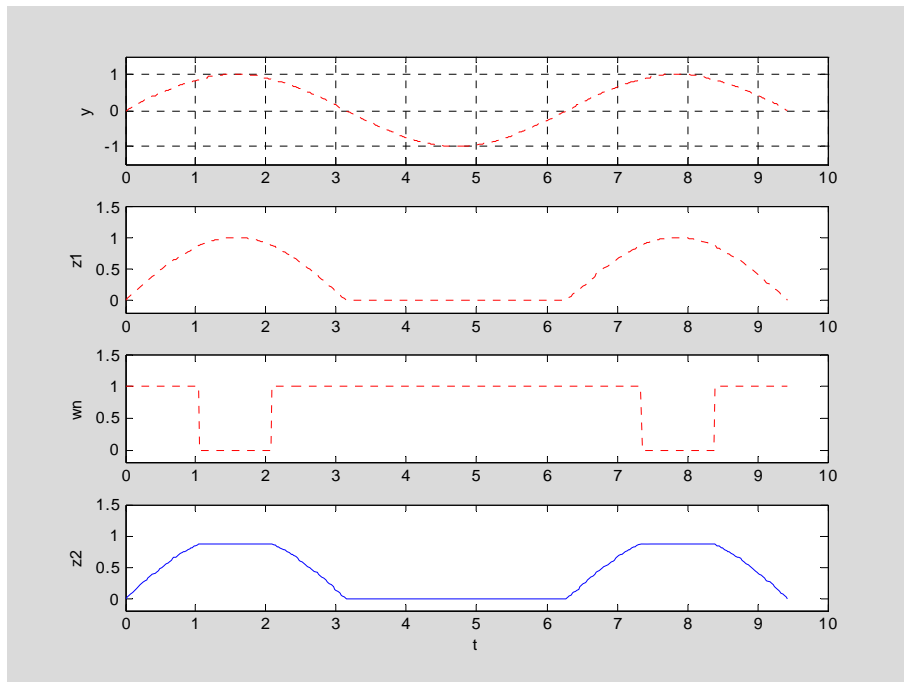


图 3.5-2 逐段解析函数的产生

### 1.5.3 常用逻辑函数

#### 习题 3

1. 要求在闭区间 $[0, 2\pi]$ 上产生具有 10 个等距采样点的一维数组。试用两种不同的指令实现。
2. 由指令 `rand('state',0),A=rand(3,5)` 生成二维数组 A，试求该数组中所有大于 0.5 的元素的位置，分别求出它们的“全下标”和“单下标”。

〔答案〕

大于 0.5 的元素的全下标



行号 1 3 2 3 3 2 3 1 2  
列号 1 1 2 2 3 4 4 5 5

大于 0.5 的元素的单下标

1 3 5 6 9 11 12 13 14

3. 在使用 123 作为 rand 随机数发生器的初始化状态的情况下, 写出产生长度为 1000 的“等概率双位 (即取 -1, +1) 取值的随机码”程序指令, 并给出 -1 码的数目。

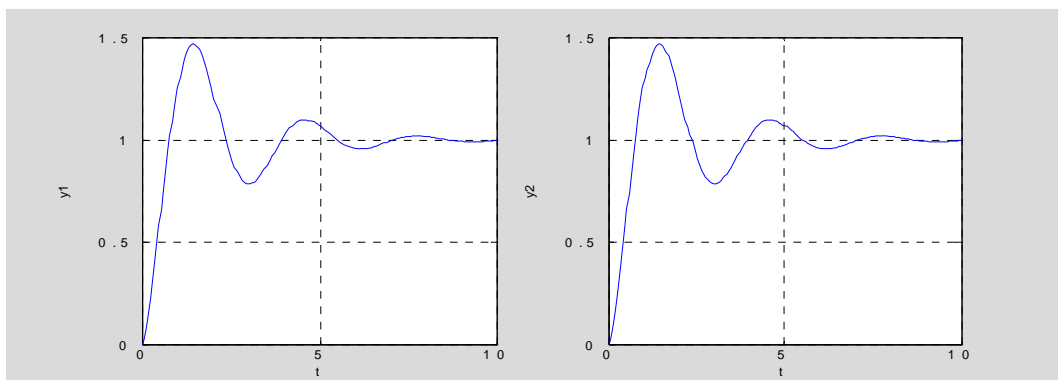
【答案】

Na =  
490

4. 已知矩阵  $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ , 运行指令  $B1=A.^{(0.5)}$ ,  $B2=A^{(0.5)}$ , 可以观察到不同运算方法所得结果不同。(1) 请分别写出根据 B1, B2 恢复原矩阵 A 的程序。(2) 用指令检验所得的两个恢复矩阵是否相等。

5. 在时间区间 [0,10]中, 绘制  $y = 1 - e^{-0.5t} \cos 2t$  曲线。要求分别采取“标量循环运算法”和“数组运算法”编写两段程序绘图。

【答案】



6. 先运行 clear,format long,rand('state',1),A=rand(3,3), 然后根据 A 写出两个矩阵: 一个对角阵 B, 其相应元素由 A 的对角元素构成; 另一个矩阵 C, 其对角元素全为 0, 而其余元素与对应的 A 阵元素相同。

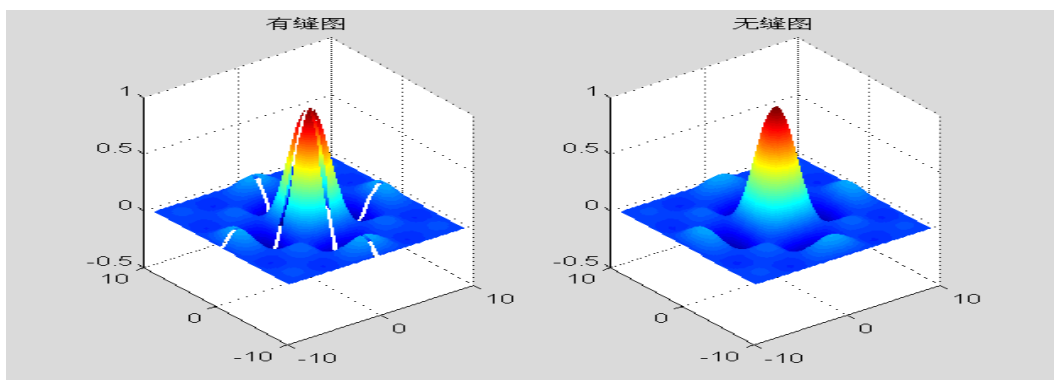
7. 先运行指令  $x=-3\pi: \pi/15: 3\pi$ ;  $y=x$ ;  $[X,Y]=\text{meshgrid}(x,y)$ ; warning off;  $Z=\sin(X).\sin(Y)./X./Y$ ; 产生矩阵 Z。(1) 请问矩阵 Z 中有多少个“非数”数据? (2)

用指令 `surf(X,Y,Z); shading interp` 观察所绘的图形。(3) 请写出绘制相应的“无缝缝”图形的全部指令。

〔答案〕

`NumOfNaN =`

`181`



8. 下面有一段程序，企图用来解决如下计算任务：有矩阵  $\mathbf{A}_k = \begin{bmatrix} 1 & k+1 & \cdots & 9k+1 \\ 2 & k+2 & \cdots & 9k+2 \\ \vdots & \vdots & \ddots & \vdots \\ k & 2k & \cdots & 10k \end{bmatrix}$ ,

当  $k$  依次取 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 时，计算矩阵  $\mathbf{A}_k$  “各列元素的和”，并把此求和结

果存放为矩阵  $\mathbf{Sa}$  的第  $k$  行。例如  $k=3$  时， $\mathbf{A}$  阵为  $\begin{bmatrix} 1 & 4 & \cdots & 28 \\ 2 & 5 & \cdots & 29 \\ 3 & 6 & \cdots & 30 \end{bmatrix}$ ，此时它各列元素

的和是一个  $(1 \times 10)$  行数组  $[6 \ 15 \ \cdots \ 87]$ ，并把它保存为  $\mathbf{Sa}$  的第 3 行。问题：该段程序的计算结果对吗？假如计算结果不正确，请指出错误发生的根源，并改正之。

```
for k=10:-1:1
    A=reshape(1:10*k,k,10);
    Sa(k,:)=sum(A);
end
Sa
```

## 第 1 章 数值计算

与符号计算相比，数值计算在科研和工程中的应用更为广泛。MATLAB 也正是凭借其卓越的数值计算能力而称雄世界。随着科研领域、工程实践的数字化进程的深入，具有数字化本质的数值计算就显得愈益重要。

较之十年、二十年前，在计算机软硬件的支持下当今人们所能拥有的计算能力已经得到了巨大的提升。这就自然地激发了人们从新的计算能力出发去学习、理解概念的欲望，鼓舞了人们用新计算能力试探解决实际问题的雄心。

鉴于当今高校本科教学偏重符号计算和便于手算简单示例的实际，也出于帮助读者克服对数值计算生疏感的考虑，本章在内容安排上仍从“微积分”开始。这一方面与第 2 章符号计算相呼应，另方面通过“微积分”说明数值计算离散本质的微观和宏观影响。

为便于读者学习，本章内容的展开脉络基本上沿循高校数学教程，而内容深度力求控制在高校本科水平。考虑到知识的跳跃和交叉，本书对重要概念、算式、指令进行了尽可能完整地说明。

## 1.1 数值微积分

### 1.1.1 近似数值极限及导数

【例 4.1-1】设  $f_1(x) = \frac{1 - \cos 2x}{x \sin x}$ ,  $f_2(x) = \frac{\sin x}{x}$ , 试用机器零阈值 eps 替代理论 0 计算极限  $L_1(0) = \lim_{x \rightarrow 0} f_1(x)$ ,  $L_2(0) = \lim_{x \rightarrow 0} f_2(x)$ 。

```
x=eps;
L1=(1-cos(2*x))/(x*sin(x)),
L2=sin(x)/x,
L1 =
    0
L2 =
    1
```

```
syms t
f1=(1-cos(2*t))/(t*sin(t));
f2=sin(t)/t;
Ls1=limit(f1,t,0)
Ls2=limit(f2,t,0)
Ls1 =
    2
Ls2 =
    1
```

【例 4.1-2】已知  $x = \sin(t)$ ，求该函数在区间  $[0, 2\pi]$  中的近似导函数。

```
d=pi/100;  
t=0:d:2*pi;  
x=sin(t);  
dt=5*eps;  
x_eps=sin(t+dt);  
dxdt_eps=(x_eps-x)/dt;  
plot(t,x,'LineWidth',5)  
hold on  
plot(t,dxdt_eps)  
hold off  
legend('x(t)','dx/dt')  
xlabel('t')
```

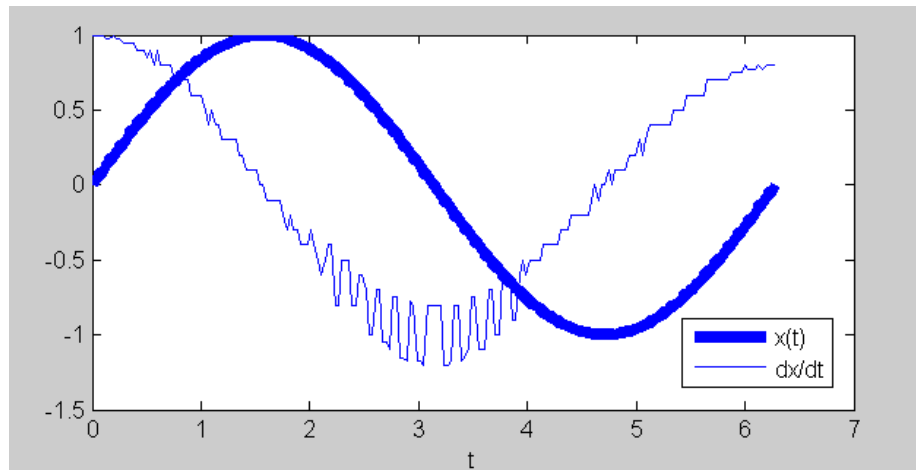


图 4.1-1 增量过小引起有效数字严重丢失后的毛刺曲线

```
x_d=sin(t+d);  
dxdt_d=(x_d-x)/d;  
plot(t,x,'LineWidth',5)  
hold on  
plot(t,dxdt_d)  
hold off  
legend('x(t)','dx/dt')  
xlabel('t')
```

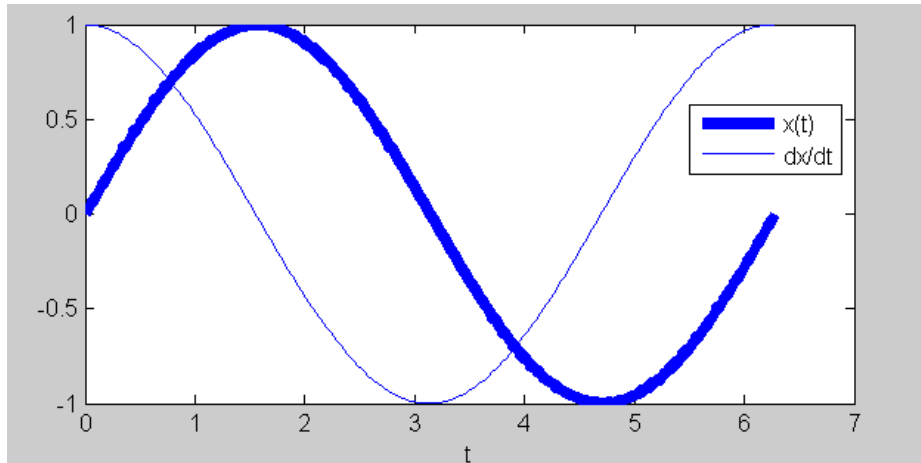


图 4.1-2 增量适当所得导函数比较光滑

【例 4.1-3】已知  $x = \sin(t)$ ，采用 diff 和 gradient 计算该函数在区间  $[0, 2\pi]$  中的近似导函数。。

```
clf
d=pi/100;
t=0:d:2*pi;
x=sin(t);
dxdt_diff=diff(x)/d;
dxdt_grad=gradient(x)/d;
subplot(1,2,1)
plot(t,x,'b')
hold on
plot(t,dxdt_grad,'m','LineWidth',8)
plot(t(1:end-1),dxdt_diff,'.k','MarkerSize',8)
axis([0,2*pi,-1.1,1.1])
title('[0, 2\pi]')
legend('x(t)','dxdt_{grad}','dxdt_{diff}','Location','North')
xlabel('t'),box off
hold off
subplot(1,2,2)
kk=(length(t)-10):length(t);
hold on
plot(t(kk),dxdt_grad(kk),'om','MarkerSize',8)
plot(t(kk-1),dxdt_diff(kk-1),'k','MarkerSize',8)
title('[end-10, end]')
legend('dxdt_{grad}','dxdt_{diff}','Location','SouthEast')
xlabel('t'),box off
hold off
```

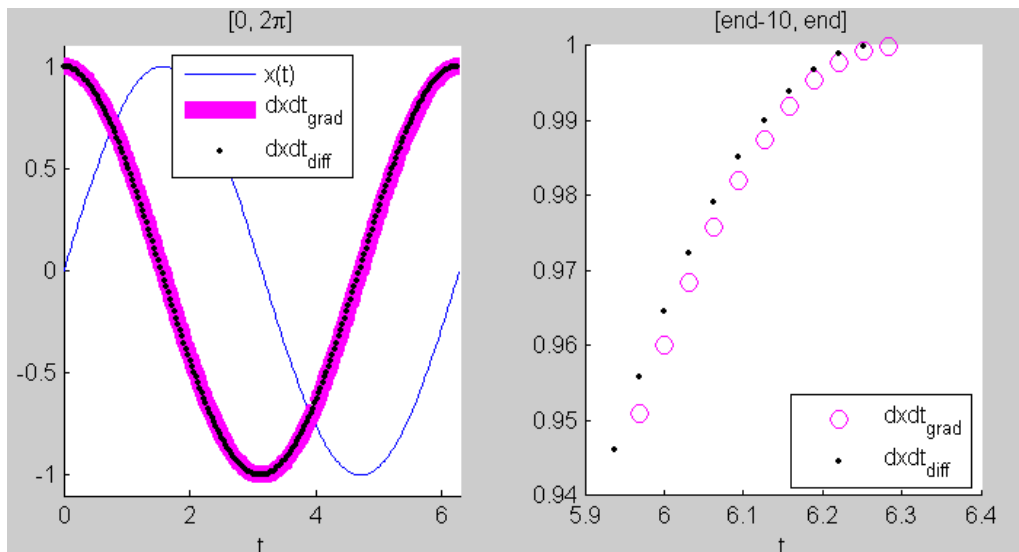


图 4.1-3 diff 和 gradient 求数值近似导数的异同比较

## 1.1.2 数值求和与近似数值积分

【例 4.1-4】求积分  $s(x) = \int_0^{\pi/2} y(t)dt$ ，其中  $y = 0.2 + \sin(t)$ 。

```
clear
d=pi/8;
t=0:d:pi/2;
y=0.2+sin(t);
s=sum(y);
s_sa=d*s;
s_ta=d*trapz(y);
disp(['sum 求得积分',blanks(3),'trapz 求得积分'])
disp([s_sa, s_ta])
t2=[t,t(end)+d];
y2=[y,nan];
stairs(t2,y2,':k')
hold on
plot(t,y,'r','LineWidth',3)
h=stem(t,y,'LineWidth',2);
set(h(1),'MarkerSize',10)
axis([0,pi/2+d,0,1.5])
hold off
shg
sum 求得积分    trapz 求得积分
    1.5762      1.3013
```

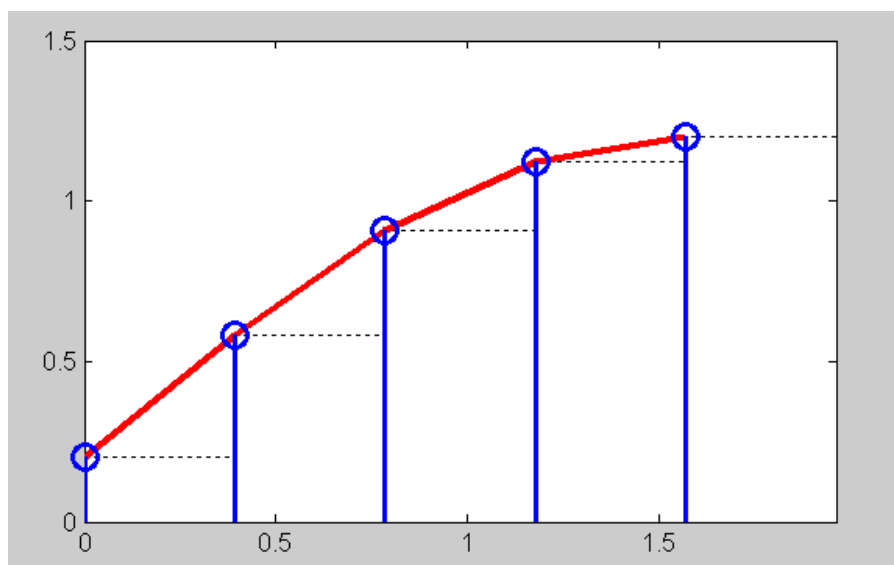


图 4.1-4 sum 和 trapz 求积模式示意

### 1.1.3 计算精度可控的数值积分

【例 4.1-5】求  $I = \int_0^1 e^{-x^2} dx$ 。

```
syms x
Isym=vpa(int(exp(-x^2),x,0,1))
Isym =
.74682413281242702539946743613185
```

```
format long
d=0.001;x=0:d:1;
Itrapz=d*trapz(exp(-x.*x))
Itrapz =
0.74682407149919
```

```
fx='exp(-x.^2)';
Ic=quad(fx,0,1,1e-8)
Ic =
0.74682413285445
```

【例 4.1-6】求  $s = \int_1^2 \int_0^1 x^y dx dy$ 。

```
syms x y
s=vpa(int(int(x^y,x,0,1),y,1,2))
s =
.40546510810816438197801311546432
```

```
format long
```

```

s_n=dblquad(@(x,y)x.^y,0,1,1,2)
s_n =
    0.40546626724351

```

## 1.1.4 函数极值的数值求解

【例 4.1-7】已知  $y = (x + \pi) \cdot e^{|\sin(x+\pi)|}$ ，在  $-\pi/2 \leq x \leq \pi/2$  区间，求函数的极小值。

(1)

```

syms x
y=(x+pi)*exp(abs(sin(x+pi)));
yd=diff(y,x);
xs0=solve(yd)
yd_xs0=vpa(subs(yd,x,xs0),6)
y_xs0=vpa(subs(y,x,xs0),6)
y_m_pi=vpa(subs(y,x,-pi/2),6)
y_p_pi=vpa(subs(y,x,pi/2),6)
xs0 =
-1.0676598444985783372948670854801
yd_xs0 =
.1e-4
y_xs0 =
4.98043
y_m_pi =
4.26987
y_p_pi =
12.8096

```

(2)

```

x1=-pi/2;x2=pi/2;
yx=@(x)(x+pi)*exp(abs(sin(x+pi)));
[xn0,fval,exitflag,output]=fminbnd(yx,x1,x2)
xn0 =
-1.2999e-005
fval =
    3.1416
exitflag =
    1
output =
    iterations: 21
    funcCount: 24
    algorithm: 'golden section search, parabolic interpolation'

```



```
message: [1x112 char]
```

(3)

```
xx=-pi/2:pi/200:pi/2;  
yxx=(xx+pi).*exp(abs(sin(xx+pi)));  
plot(xx,yxx)  
xlabel('x'),grid on
```

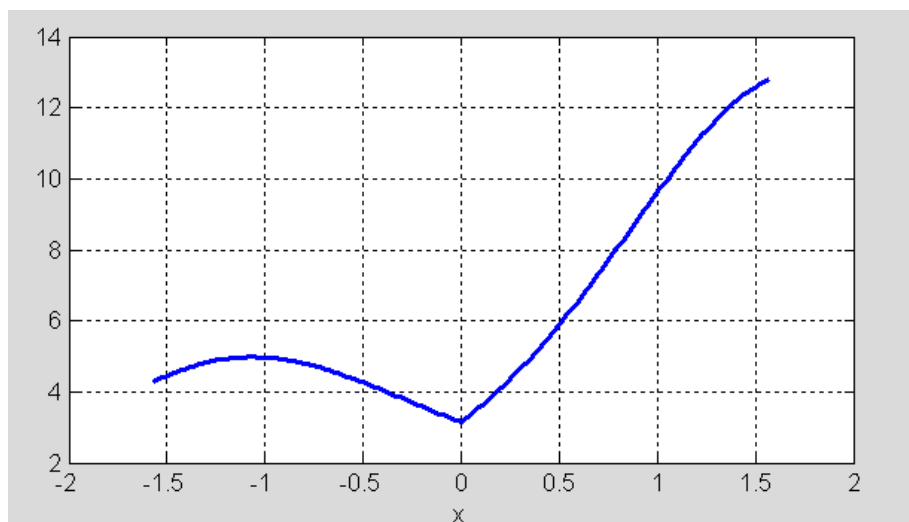


图 4.1-5 在 $[-\pi/2, \pi/2]$ 区间中的函数曲线

```
[xx,yy]=ginput(1)  
xx=  
1.5054e-008  
yy=  
3.1416
```

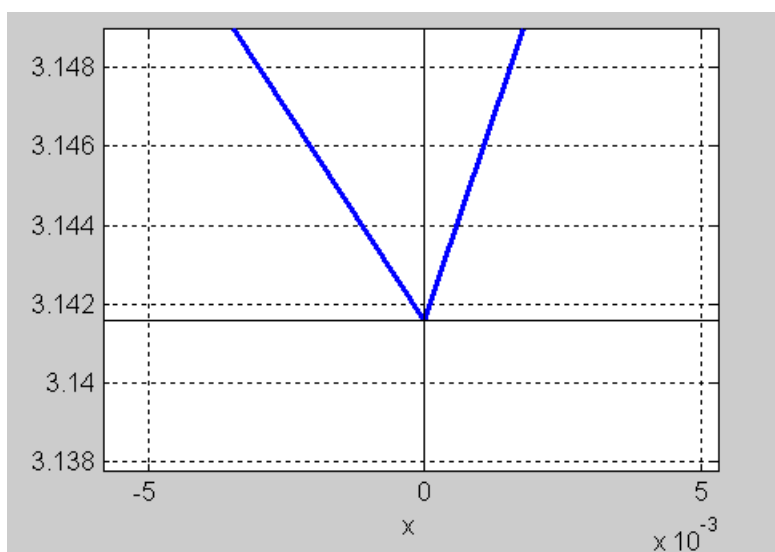


图 4.1-6 函数极值点附近的局部放大和交互式取值

【例 4.1-8】求  $f(x, y) = 100(y - x^2)^2 + (1 - x)^2$  的极小值点。它即是著名的 Rosenbrock's

"Banana" 测试函数，它的理论极小值是  $x = 1, y = 1$ 。

(1)

```
ff=@(x)(100*(x(2)-x(1).^2)^2+(1-x(1))^2);
```

(2)

```
x0=[-5,-2,2,5;-5,-2,2,5];
[sx,sfval,sexit,soutput]=fminsearch(ff,x0)
sx =
    1.0000    -0.6897     0.4151     8.0886
    1.0000    -1.9168     4.9643     7.8004
sfval =
    2.4112e-010
sexit =
     1
soutput =
    iterations: 384
    funcCount: 615
    algorithm: 'Nelder-Mead simplex direct search'
    message: [1x196 char]
```

(3) 检查目标函数值

```
format short e
disp([ff(sx(:,1)),ff(sx(:,2)),ff(sx(:,3)),ff(sx(:,4))])
    2.4112e-010    5.7525e+002    2.2967e+003    3.3211e+005
```

## 1.1.5 常微分方程的数值解

【例 4.1-9】求微分方程  $\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0$ ， $\mu = 2$ ，在初始条件

$x(0) = 1, \frac{dx(0)}{dt} = 0$  情况下的解，并图示。（见图 4.1-7 和 4.1-8）

(1)

(2)

[DyDt.m]

```
function ydot=DyDt(t,y)
mu=2;
ydot=[y(2);mu*(1-y(1)^2)*y(2)-y(1)];
```

(3) 解算微分方程

```
tspan=[0,30];  
y0=[1;0];  
[tt,yy]=ode45(@DyDt,tspan,y0);  
plot(tt,yy(:,1))  
xlabel('t'),title('x(t)')
```

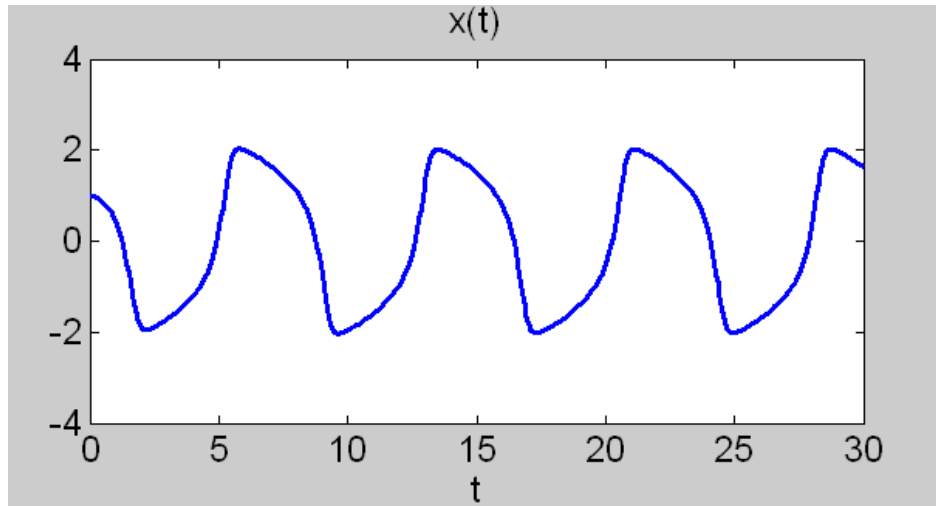


图 4.1-7 微分方程解

(4)

```
plot(yy(:,1),yy(:,2))  
xlabel('位移'),ylabel('速度')
```

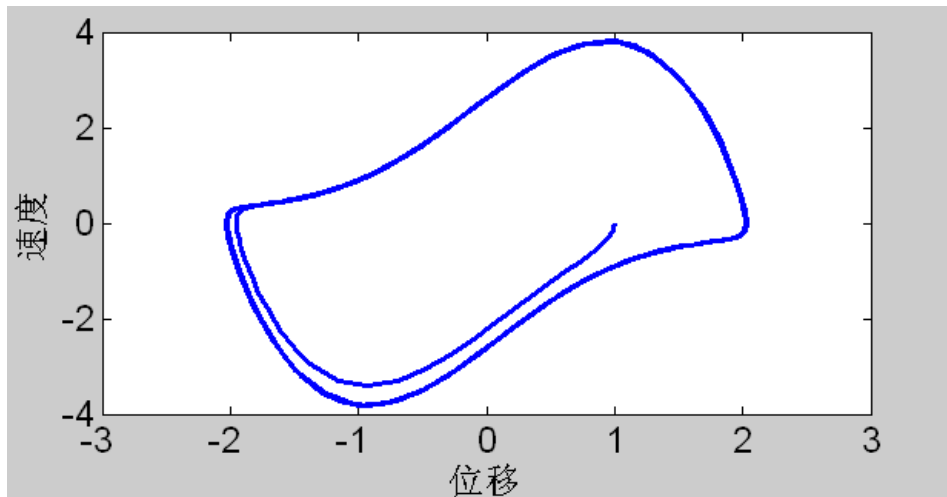


图 4.1-8 平面相轨迹

## 1.2 矩阵和代数方程

### 1.2.1 矩阵运算和特征参数

## 一 矩阵运算

【例 4.2-1】已知矩阵  $\mathbf{A}_{2 \times 4}$ ， $\mathbf{B}_{4 \times 3}$ ，采用三种不同的编程求这两个矩阵的乘积

$$\mathbf{C}_{2 \times 3} = \mathbf{A}_{2 \times 4} \mathbf{B}_{4 \times 3}。$$

(1)

```
clear
rand('state',12)
A=rand(2,4);B=rand(4,3);
%-----
C1=zeros(size(A,1),size(B,2));
for ii=1:size(A,1)
    for jj=1:size(B,2)
        for k=1:size(A,2)
            C1(ii,jj)=C1(ii,jj)+A(ii,k)*B(k,jj);
        end
    end
end
C1
C1 =
    1.0559    0.9859    1.2761
    1.7654    1.8483    1.8811
```

(2)

```
%-----
C2=zeros(size(A,1),size(B,2));
for jj=1:size(B,2)
    for k=1:size(B,1)
        C2(:,jj)=C2(:,jj)+A(:,k)*B(k,jj);
    end
end
C2
C2 =
    1.0559    0.9859    1.2761
    1.7654    1.8483    1.8811
```

(3)

```
C3=A*B,
C3 =
    1.0559    0.9859    1.2761
    1.7654    1.8483    1.8811
```

(3)

```

C3_C3=norm(C3-C1,'fro')
C3_C2=norm(C3-C2,'fro')
C3_C3 =
    0
C3_C2 =
    0

```

【例 4.2-2】观察矩阵的转置操作和数组转置操作的差别。

```

format rat
A=magic(2)+j*pascal(2)
A =
    1      +    1i      3      +    1i
    4      +    1i      2      +    2i

%
A1=A'
A2=A.'
A1 =
    1      -    1i      4      -    1i
    3      -    1i      2      -    2i
A2 =
    1      +    1i      4      +    1i
    3      +    1i      2      +    2i

B1=A*A'
B2=A.*A'
C1=A*A.'
C2=A.*A.'
B1 =
    12      13      -    1i
    13      25
B2 =
    2      13      +    1i
    13      8
C1 =
    8      +    8i      7      +    13i
    7      +    13i      15      +    16i
C2 =
    0      +    2i      11      +    7i
    11      +    7i      0      +    8i

```

## 二 矩阵的标量特征参数

【例 4.2-3】矩阵标量特征参数计算示例。

```
A=reshape(1:9,3,3)
r=rank(A)
d3=det(A)
d2=det(A(1:2,1:2))
t=trace(A)
A =
     1     4     7
     2     5     8
     3     6     9
r =
     2
d3 =
     0
d2 =
    -3
t =
    15
```

【例 4.2-4】矩阵标量特征参数的性质。

```
format short g
rand('state',0)
A=rand(3,3);
B=rand(3,3);
C=rand(3,4);
D=rand(4,3);

tAB=trace(A*B)
tBA=trace(B*A)
tCD=trace(C*D)
tDC=trace(D*C)
tAB =
    3.6697
tBA =
    3.6697
tCD =
    2.1544
tDC =
    2.1544
```

```
d_A_B=det(A)*det(B)
```

```
dAB=det(A*B)
```

```
dBA=det(B*A)
```

```
d_A_B =  
      0.0846
```

```
dAB =  
      0.0846
```

```
dBA =  
      0.0846
```

```
dCD=det(C*D)
```

```
dDC=det(D*C)
```

```
dCD =  
    -0.012362
```

```
dDC =  
    -2.1145e-018
```

## 1.2.2 矩阵的变换和特征值分解

【例 4.2-5】行阶梯阵简化指令 `rref` 计算结果的含义。

(1)

```
A=magic(4)
```

```
[R,ci]=rref(A)
```

```
A =  
    16     2     3    13  
     5    11    10     8  
     9     7     6    12  
     4    14    15     1
```

```
R =  
     1     0     0     1  
     0     1     0     3  
     0     0     1    -3  
     0     0     0     0
```

```
ci =  
     1     2     3
```

(2)

```
r_A=length(ci)
```

```
r_A =  
     3
```

(3)

```

aa=A(:,1:3)*R(1:3,4)
err=norm(A(:,4)-aa)
aa =
    13
     8
    12
     1
err =
     0

```

【例 4.2-6】矩阵零空间及其含义。

```

A=reshape(1:15,5,3);
X=null(A)
S=A*X
n=size(A,2);
l=size(X,2);
n-l==rank(A)
X =
   -0.4082
    0.8165
   -0.4082
S =
   1.0e-014 *
    0.2109
    0.2220
    0.3220
    0.3331
    0.4330
ans =
     1

```

【例 4.2-7】简单实阵的特征值分解。

(1)

```

A=[1,-3;2,2/3]
[V,D]=eig(A)
A =
    1.0000   -3.0000
    2.0000    0.6667
V =
    0.7746           0.7746
    0.0430 - 0.6310i    0.0430 + 0.6310i
D =
    0.8333 + 2.4438i    0

```



```
0 0.8333 - 2.4438i
```

(2)

```
[VR,DR]=cdf2rdf(V,D)
```

```
VR =
```

```
0.7746 0
0.0430 -0.6310
```

```
DR =
```

```
0.8333 2.4438
-2.4438 0.8333
```

(3)

```
A1=V*D/V
```

```
A1_1=real(A1)
```

```
A2=VR*DR/VR
```

```
err1=norm(A-A1,'fro')
```

```
err2=norm(A-A2,'fro')
```

```
A1 =
```

```
1.0000 - 0.0000i -3.0000
2.0000 + 0.0000i 0.6667
```

```
A1_1 =
```

```
1.0000 -3.0000
2.0000 0.6667
```

```
A2 =
```

```
1.0000 -3.0000
2.0000 0.6667
```

```
err1 =
```

```
4.4409e-016
```

```
err2 =
```

```
4.4409e-016
```

## 1.2.3 线性方程的解

### 一 线性方程解的一般结论

### 二 除法运算解方程

【例 4.2-8】求方程  $\begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix} \cdot \mathbf{x} = \begin{bmatrix} 13 \\ 14 \\ 15 \\ 16 \end{bmatrix}$  的解。

(1)

```
A=reshape(1:12,4,3);
b=(13:16)';
```

(2)

```
ra=rank(A)
rab=rank([A,b])
ra =
     2
rab =
     2
```

(3)

```
xs=A\b;
xg=null(A);
c=rand(1);
ba=A*(xs+c*xg)
norm(ba-b)
Warning: Rank deficient, rank = 2, tol = 1.8757e-014.
ba =
    13.0000
    14.0000
    15.0000
    16.0000
ans =
    1.3874e-014
```

### 三 矩阵逆

【例 4.2-9】“逆阵”法和“左除”法解恰定方程的性能对比

(1)

```
randn('state',0);
A=gallery('randsvd',300,2e13,2);
x=ones(300,1);
b=A*x;
cond(A)
ans =
    2.0069e+013
```

(2)

```
tic
xi=inv(A)*b;
ti=toc
eri=norm(x-xi)
```

```

rei=norm(A*xi-b)/norm(b)
ti =
    0.0341
eri =
    0.0839
rei =
    0.0048

```

```

(3)
tic;
xd=A\b;
td=toc
erd=norm(x-xd)
red=norm(A*xd-b)/norm(b)
td =
    0.0172
erd =
    0.0127
red =
    9.4744e-015

```

## 1.2.4 一般代数方程的解

【例 4.2-10】求  $f(t) = (\sin^2 t) \cdot e^{-0.1t} - 0.5|t|$  的零点。

```

(1)
S=solve('sin(t)^2*exp(-0.1*t)-0.5*abs(t)','t')
S =
    0.

```

```

(2)
y_C=inline('sin(t).^2.*exp(-0.1*t)-0.5*abs(t)','t');

t=-10:0.01:10;
Y=y_C(t);
clf,
plot(t,Y,'r');
hold on
plot(t,zeros(size(t)),'k');
xlabel('t');ylabel('y(t)')
hold off

```

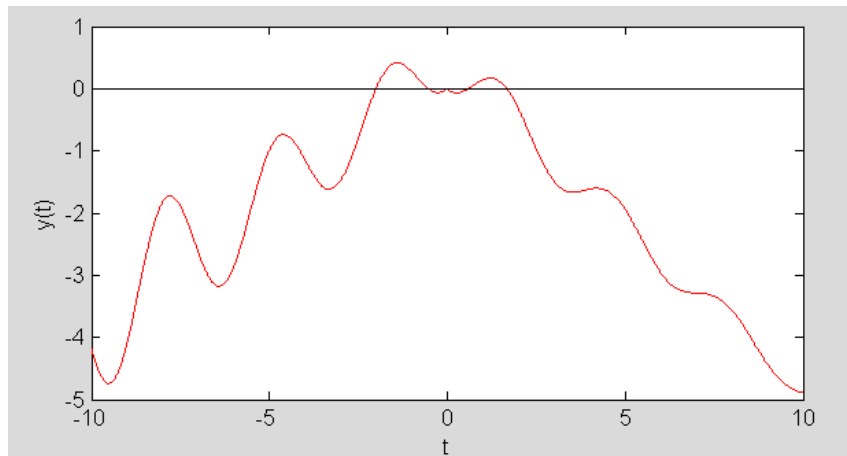


图 4.2-1 函数零点分布观察图

`zoom on`

`[tt,yy]=ginput(5);zoom off`

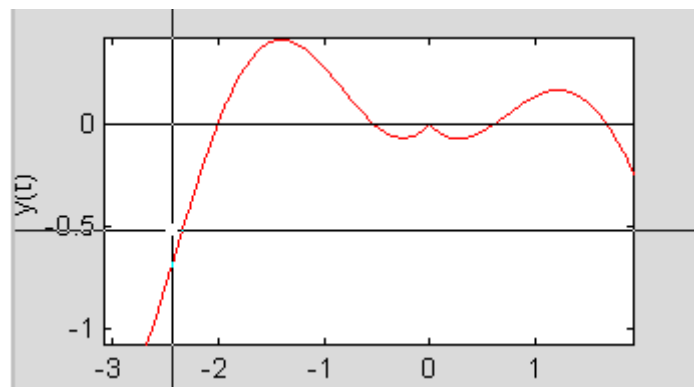


图 4.2-2 局部放大和利用鼠标取值图

`tt`

`tt =`

`-2.0039`

`-0.5184`

`-0.0042`

`0.6052`

`1.6717`

`[t4,y4]=fzero(y_C,0.1)`

`t4 =`

`0.5993`

`y4 =`

`1.1102e-016`

## 1.3 概率分布和统计分析

### 1.3.1 概率函数、分布函数、逆分布函数和随机数的发生

#### 一 二项分布(Binomial distribution)

【例 4.3-1】画出  $N=100, p=0.5$  情况下的二项分布概率特性曲线。

```
N=100;p=0.5;  
k=0:N;  
pdf=binopdf(k,N,p);  
cdf=binocdf(k,N,p);  
h=plotyy(k,pdf,k,cdf);  
set(get(h(1),'Children'),'Color','b','Marker','.', 'MarkerSize',13)  
set(get(h(1),'Ylabel'),'String','pdf')  
set(h(2),'Ycolor',[1,0,0])  
set(get(h(2),'Children'),'Color','r','Marker','+', 'MarkerSize',4)  
set(get(h(2),'Ylabel'),'String','cdf')  
xlabel('k')  
grid on
```

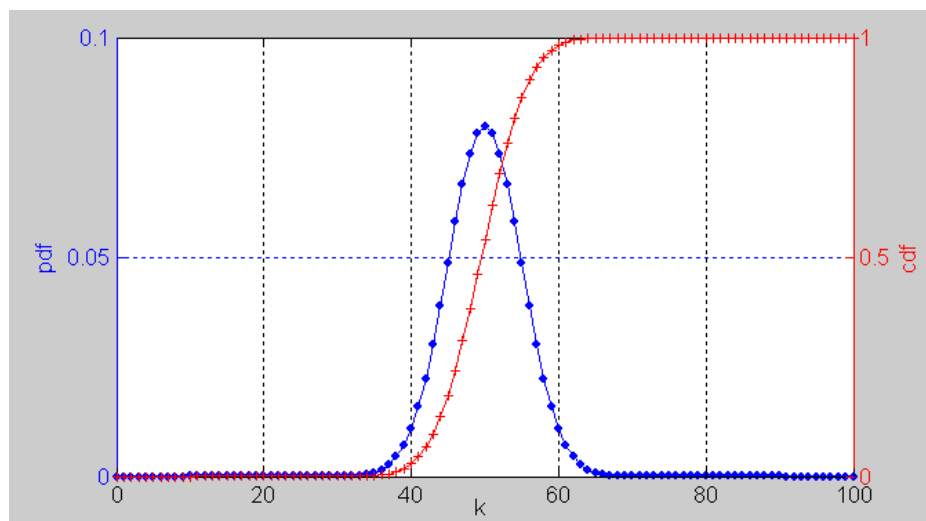


图 4.3-1 二项分布  $B(100, 0.5)$  的概率和累计概率曲线

#### 二 正态分布 (Normal distribution)

【例 4.3-2】正态分布标准差的几何表示。

```
mu=3;sigma=0.5;  
x=mu+sigma*[-3:-1,1:3];  
yf=normcdf(x,mu,sigma);  
P=[yf(4)-yf(3),yf(5)-yf(2),yf(6)-yf(1)];  
xd=1:0.1:5;  
yd=normpdf(xd,mu,sigma);
```

```

clf
for k=1:3
    %-----
    xx=x(4-k):sigma/10:x(3+k);
    yy=normpdf(xx,mu,sigma);
    %-----
    subplot(3,1,k),plot(xd,yd,'b');
    hold on
    fill([x(4-k),xx,x(3+k)],[0,yy,0],'g');
    hold off
    if k<2
        text(3.8,0.6,['{\mu}-{\sigma},{\mu}+{\sigma}'])
    else
        kk=int2str(k);
        text(3.8,0.6,['[{\mu}-',kk,'{\sigma},{\mu}+',kk,'{\sigma}]'])
    end
    text(2.8,0.3,num2str(P(k)));shg
end
xlabel('x');shg

```

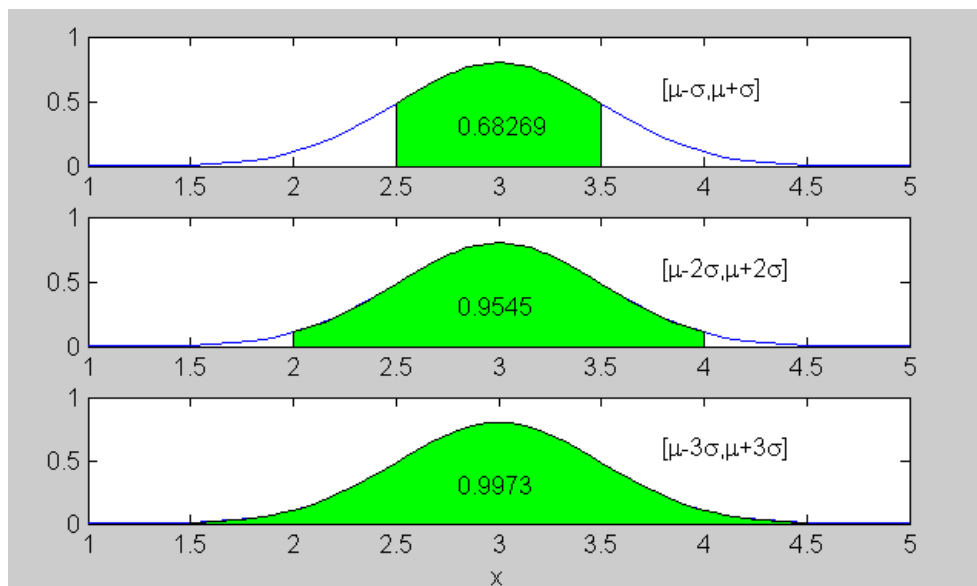


图 4.3-2 均值两侧一、二、三倍标准差之间的概率

### 三 各种概率分布的交互式观察界面

【例 4.3-3】概率分布交互界面使用方法介绍。

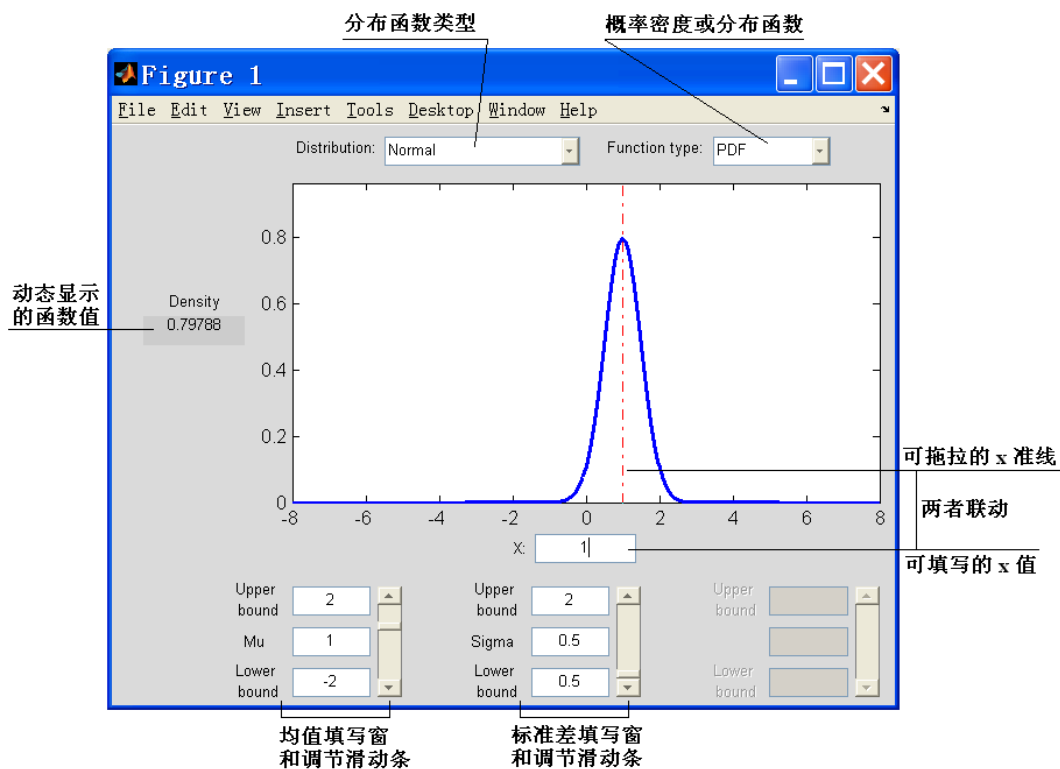


图 4.3-3 概率分布交互界面

### 1.3.2 随机数发生器和 统计分析指令

【例 4.3-4】随机数据的统计量。

```

randn('state',0)
A=randn(1000,4);
AMAX=max(A)
AMIN=min(A)
CM=mean(A)
MA=mean(mean(A))
S=std(A)
var(A)-S.^2
C=cov(A)
diag(C)'-var(A)
p=corrcoef(A)
AMAX =
    2.7316    3.2025    3.4128    3.0868
AMIN =
   -2.6442   -3.0737   -3.5027   -3.0461
CM =
   -0.0431    0.0455    0.0177    0.0263
MA =
    0.0116
S =
    0.9435    1.0313    1.0248    0.9913

```

```
ans =
    1.0e-015 *
         0    -0.2220         0         0
C =
    0.8902    -0.0528    0.0462    0.0078
   -0.0528    1.0635    0.0025    0.0408
    0.0462    0.0025    1.0502   -0.0150
    0.0078    0.0408   -0.0150    0.9826
ans =
         0         0         0         0
p =
    1.0000   -0.0543    0.0478    0.0083
   -0.0543    1.0000    0.0024    0.0399
    0.0478    0.0024    1.0000   -0.0147
    0.0083    0.0399   -0.0147    1.0000
```

【例 4.3-5】产生 1000 个服从  $N(2,0.5^2)$  的随机数。

```
mu=2;s=0.5;
randn('state',22)
x=randn(1000,1);
y=s*x+mu;
z=s*(x+mu);
subplot(3,1,1),histfit(x),axis([-5,5,0,100]),ylabel('x')
subplot(3,1,2),histfit(y),axis([-5,5,0,100]),ylabel('y')
subplot(3,1,3),histfit(z),axis([-5,5,0,100]),ylabel('z')
```

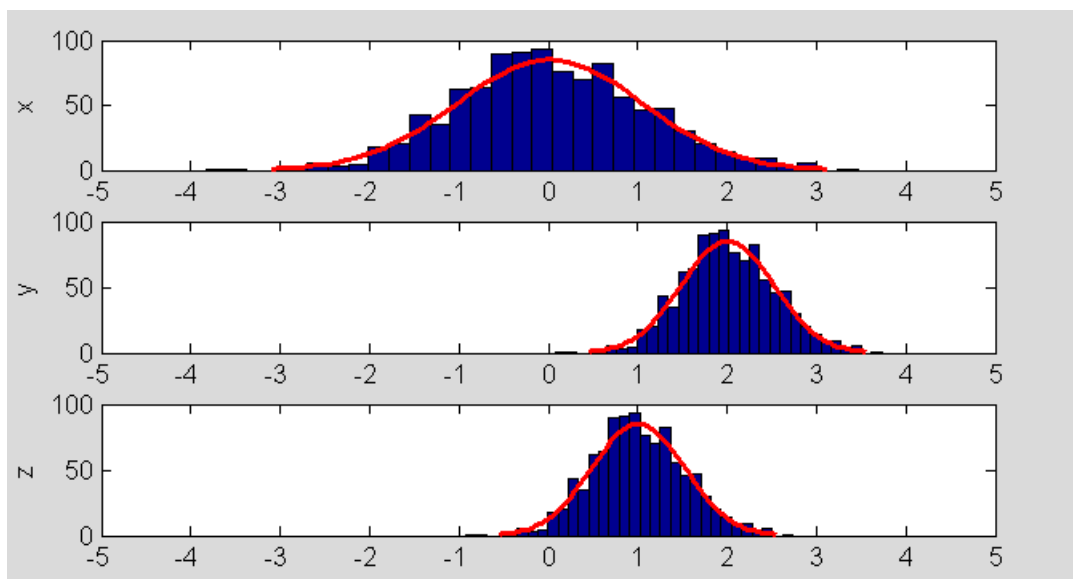


图 4.3-4 均值为 2 标准差为 0.5 的随机数样本  $z$



## 1.4 多项式运算和卷积

### 1.4.1 多项式的运算函数

#### 一 多项式表达方式的约定

#### 二 多项式运算函数

【例 4.4-1】求  $\frac{(s^2 + 2)(s + 4)(s + 1)}{s^3 + s + 1}$  的“商”及“余”多项式。

(1)

```
format rat
p1=conv([1,0,2],conv([1,4],[1,1]));
p2=[1 0 1 1];
[q,r]=deconv(p1,p2);
cq='商多项式为 '; cr='余多项式为 ';
disp([cq,poly2str(q,'s')]),disp([cr,poly2str(r,'s')])
商多项式为      s + 5
余多项式为      5 s^2 + 4 s + 3
```

(2)

```
qp2=conv(q,p2);
pp1=qp2+[zeros(1,length(qp2)-length(r)),r];
pp1==p1
ans =
      1      1      1      1      1
```

【例 4.4-2】矩阵和特征多项式，特征值和多项式根。

(1)

```
A=[11 12 13;14 15 16;17 18 19];
PA=poly(A)
PPA=poly2str(PA,'s')
PA =
      1.0000  -45.0000  -18.0000   0.0000
PPA =
      s^3 - 45 s^2 - 18 s + 2.163e-014
```

(2)

```
s=eig(A)
r=roots(PPA)
s =
      45.3965
```

```

-0.3965
0.0000
r =
45.3965
-0.3965
0.0000

(3)
n = length(PA);
AA = diag(ones(1,n-2,class(PA)),-1);
AA(1,:) = -PA(2:n) ./ PA(1);
AA
sr = eig(AA)
AA =
45.0000    18.0000   -0.0000
 1.0000         0         0
      0     1.0000         0
sr =
45.3965
-0.3965
0.0000

```

【例 4.4-3】构造指定特征根的多项式。

```

R=[-0.5,-0.3+0.4*i,-0.3-0.4*i];
P=poly(R)
PR=real(P)
PPR=poly2str(PR,'x')
P =
1.0000    1.1000    0.5500    0.1250
PR =
1.0000    1.1000    0.5500    0.1250
PPR =
x^3 + 1.1 x^2 + 0.55 x + 0.125

```

【例 4.4-4】多项式求值指令 polyval 与 polyvalm 的本质差别。

```

(1)
clear
p=[1,2,3];
poly2str(p,'x')
x=[1,2;3,4]
ans =
x^2 + 2 x + 3

```

```
X =
     1     2
     3     4
```

(2)

```
va=X.^2+2*X+3
Va=polyval(p,X)
va =
     6    11
    18    27
Va =
     6    11
    18    27
```

(3)

```
vm=X^2+2*X+3*eye(2)
Vm=polyvalm(p,X)
vm =
    12    14
    21    33
Vm =
    12    14
    21    33
```

(4)

```
cp=poly(X);
poly2str(cp,'x')
cpXa=polyval(cp,X)
cpX=polyvalm(cp,X)
ans =
    x^2 - 5 x - 2
cpXa =
    -6    -8
    -8    -6
cpX =
    1.0e-015 *
    0.2220         0
         0    0.2220
```

## 1.4.2 多项式拟合和最小二乘法

## 一 多项式拟合

【例 4.4-5】 给定数据组  $x_0, y_0$ ，求拟合三阶多项式，并图示拟合情况。（见图 4.4-1）

```
x0=0:0.1:1;  
y0=[-.447,1.978,3.11,5.25,5.02,4.66,4.01,4.58,3.45,5.35,9.22];  
  
n=3;  
P=polyfit(x0,y0,n)  
  
xx=0:0.01:1;  
yy=polyval(P,xx);  
plot(xx,yy,'-b',x0,y0,'.r','MarkerSize',20)  
legend('拟合曲线','原始数据','Location','SouthEast')  
xlabel('x')  
P =  
    56.6915   -87.1174    40.0070   -0.9043
```

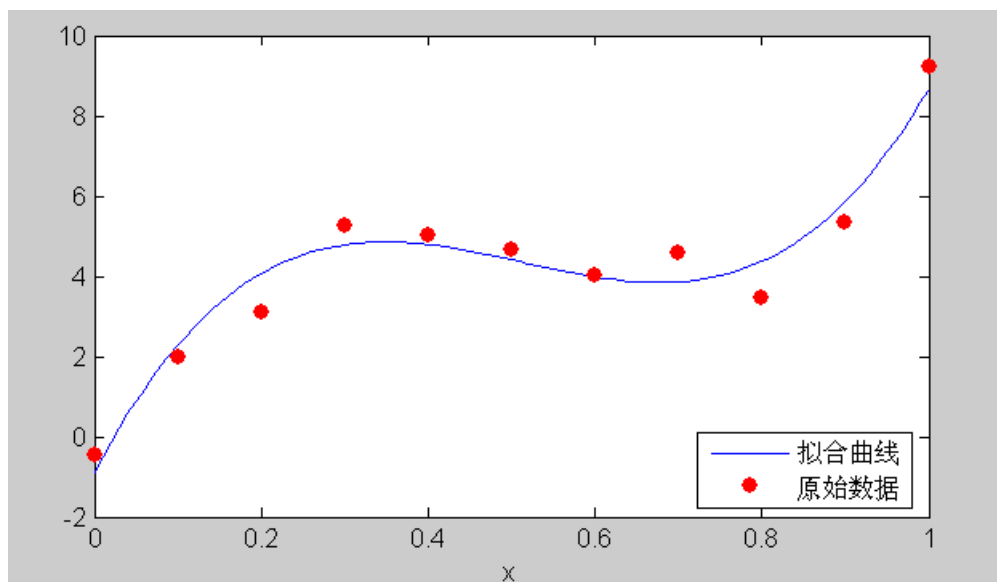


图 4.4-1 采用三阶多项式所得的拟合曲线

## 二 最小二乘问题

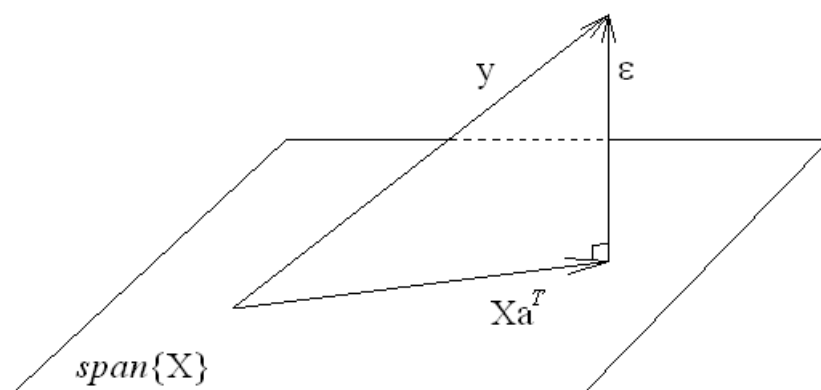


图 4.4-2 最小二乘的几何解释

【例 4.4-6】采用与例 4.4-5 相同的数据组  $x_0, y_0$ ，运用式(4.4-3)求拟合多项式的系数。

```
x0=(0:0.1:1)';
y0=[-.447,1.978,3.11,5.25,5.02,4.66,4.01,4.58,3.45,5.35,9.22]';
m=length(x0);
n=3;

X=zeros(m,n+1);
for k=1:n
    X(:,n-k+1)=(x0.^k);
end
X(:,n+1)=ones(m,1);

aT=(X\y0)'
```

aT =

```
56.6915 -87.1174 40.0070 -0.9043
```

### 1.4.3 两个有限长序列的卷积

【例 4.4-7】有序列  $A(n) = \begin{cases} 1 & n = 3, 4, \dots, 12 \\ 0 & \text{else} \end{cases}$  和  $B(n) = \begin{cases} 1 & n = 2, 3, \dots, 9 \\ 0 & \text{else} \end{cases}$ ，求

这两个序列的卷积。

(1)

```
N1=3;N2=12;
A=ones(1,(N2-N1+1));
```

```

M1=2;M2=9;
B=ones(1,(M2-M1+1));
Nc1=N1+M1;Nc2=N2+M2;
kcc=Nc1:Nc2;
for n=Nc1:Nc2
    w=0;
    for k=N1:N2
        kk=k-N1+1;
        t=n-k;
        if t>=M1&t<=M2
            tt=t-M1+1;
            w=w+A(kk)*B(tt);
        end
    end
    nn=n-Nc1+1;
    cc(nn)=w;
end
kcc,cc
kcc =
     5     6     7     8     9    10    11    12    13    14    15    16    17
18    19    20    21
cc =
     1     2     3     4     5     6     7     8     8     8     7     6     5
4     3     2     1

```

(2)

```

N1=3;N2=12;
a=ones(1,N2+1);a(1:N1)=0;
M1=2;M2=9;
b=ones(1,M2+1);b(1:M1)=0;
c=conv(a,b);
kc=0:(N2+M2);
kc,c
kc =
     0     1     2     3     4     5     6     7     8     9    10    11    12
13    14    15    16    17    18    19    20    21
c =
     0     0     0     0     0     1     2     3     4     5     6     7     8
8     8     7     6     5     4     3     2     1

```

(3)

```

N1=3;N2=12;
M1=2;M2=9;
A=ones(1,(N2-N1+1));

```

```

B=ones(1,(M2-M1+1));
C=conv(A,B);
Nc1=N1+M1;Nc2=N2+M2;
KC=Nc1:Nc2;
KC,C
KC =
      5      6      7      8      9     10     11     12     13     14     15     16     17
18     19     20     21
C =
      1      2      3      4      5      6      7      8      8      8      7      6      5
4      3      2      1

```

(3)

```

subplot(2,1,1),stem(kc,c), text(20,6,'0 起点法')
CC=[zeros(1,KC(1)),C];
subplot(2,1,2),stem(kc,CC),text(18,6,'非平凡区间法')
xlabel('n')

```

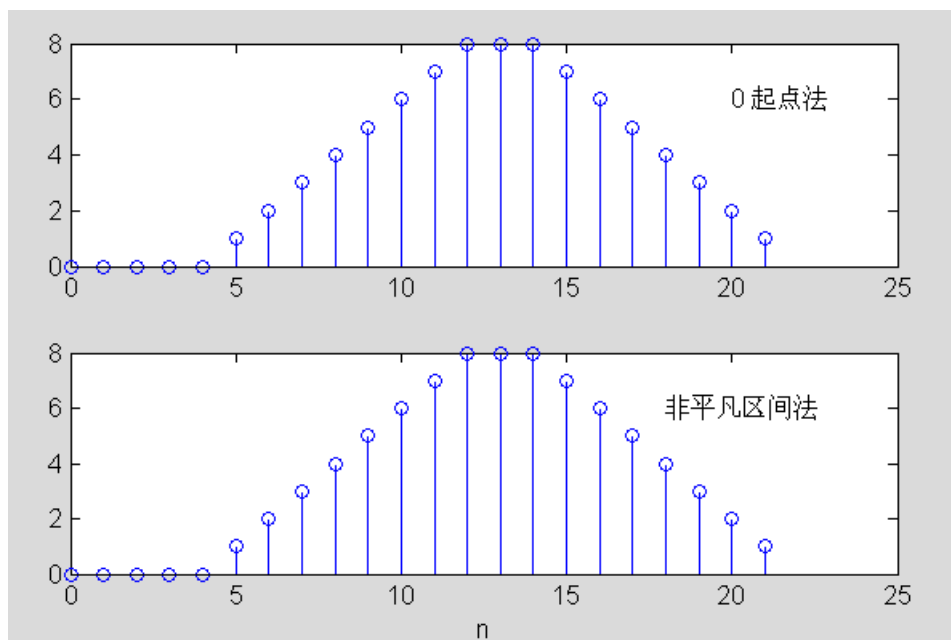
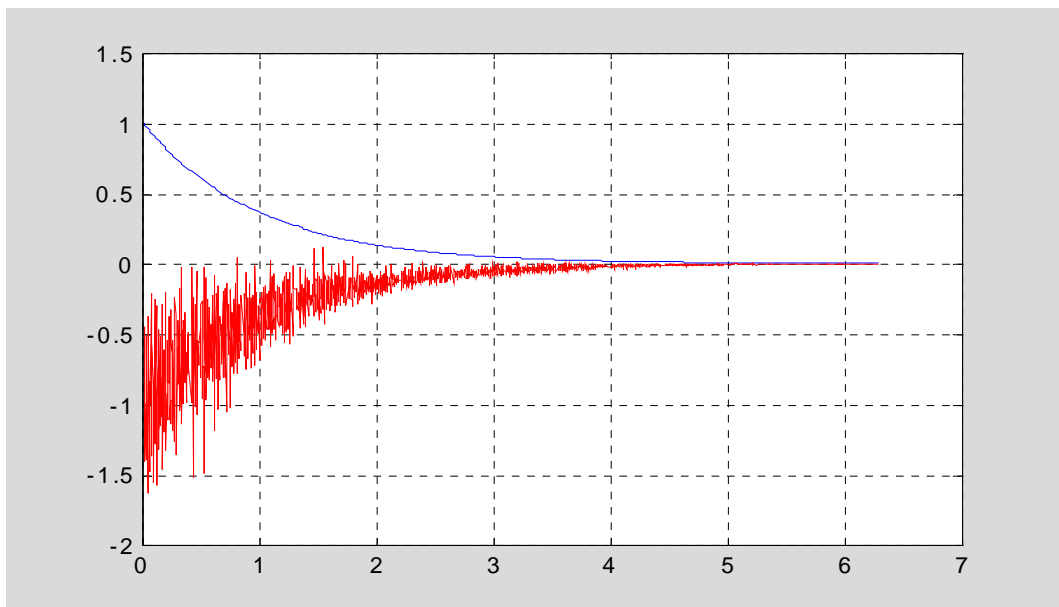


图 4.4-3 借助 conv 指令时两种不同序列记述法所得的卷积序列

## 习题 4

1. 根据题给的模拟实际测量数据的一组  $t$  和  $y(t)$  试用数值差分 `diff` 或数值梯度 `gradient` 指令计算  $y'(t)$ ，然后把  $y(t)$  和  $y'(t)$  曲线绘制在同一张图上，观察数值求导的后果。（模拟数据从 `prob_data401.mat` 获得）

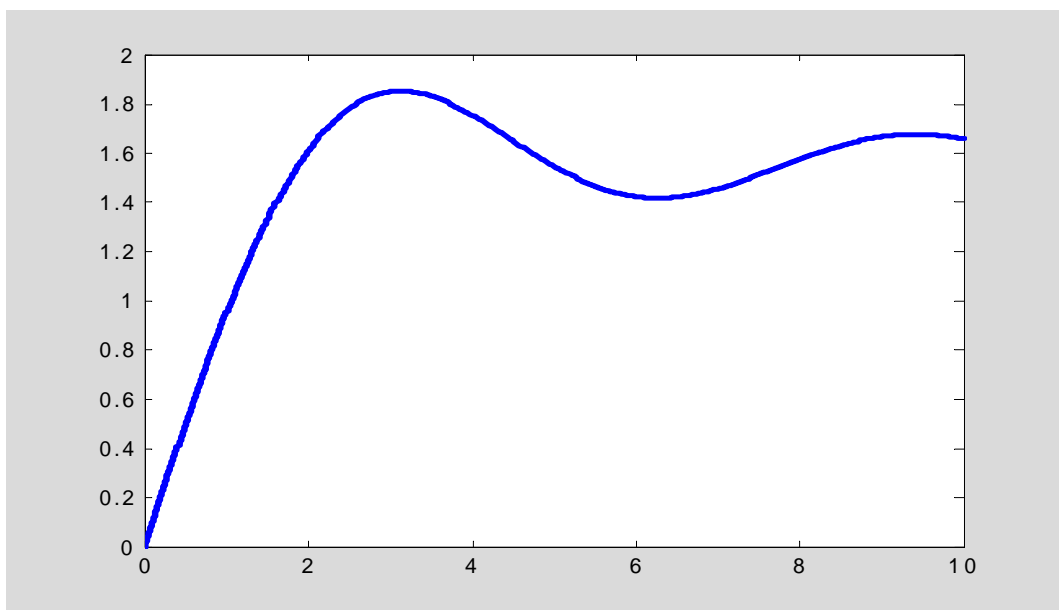
【答案】



2. 采用数值计算方法，画出  $y(x) = \int_0^x \frac{\sin t}{t} dt$  在  $[0, 10]$  区间曲线，并计算  $y(4.5)$ 。

【答案】

**s45 =**  
**1.6541**





3. 求函数  $f(x) = e^{\sin^3 x}$  的数值积分  $s = \int_0^\pi f(x)dx$ ，并请采用符号计算尝试复算。

【答案】

```
s =
    5.1370
```

```
Warning: Explicit integral could not be found.
> In sym.int at 58
ss =
int(exp(sin(x)^3),x = 0 .. pi)
```

4. 用 quad 求取  $\int_{-5\pi}^{1.7\pi} e^{-|x|} |\sin x| dx$  的数值积分，并保证积分的绝对精度为  $10^{-9}$ 。

【答案】

```
sq =
    1.08784993815498
```

5. 求函数  $f(t) = (\sin 5t)^2 e^{0.06t^2} - 1.5t \cos 2t + 1.8|t + 0.5|$  在区间  $[-5, 5]$  中的最小值点。

【答案】

```
最小值点是      -1.28498111480531
相应目标值是    -0.18604801006545
```

6. 设  $\frac{d^2 y(t)}{dt^2} - 3\frac{dy(t)}{dt} + 2y(t) = 1$ ,  $y(0) = 1$ ,  $\frac{dy(0)}{dt} = 0$ ，用数值法和符号法求  $y(t)|_{t=0.5}$ 。

【答案】

数值解

```
y_05 =
    0.78958020790127
```

符号解

```
ys =
1/2-1/2*exp(2*t)+exp(t)
ys_05 =
    .78958035647060552916850705213780
```

7. 已知矩阵  $A = \text{magic}(8)$ , (1) 求该矩阵的“值空间基阵”  $B$  ; (2) 写出“A 的任何列可用基向量线性表出”的验证程序 (提示: 利用 `rref` 检验)。

【答案】

三组不同的基

**B1 =**

```
64      2      3
 9     55     54
17     47     46
40     26     27
32     34     35
41     23     22
49     15     14
 8     58     59
```

**B2 =**

```
-0.3536    0.5401    0.3536
-0.3536   -0.3858   -0.3536
-0.3536   -0.2315   -0.3536
-0.3536    0.0772    0.3536
-0.3536   -0.0772    0.3536
-0.3536    0.2315   -0.3536
-0.3536    0.3858   -0.3536
-0.3536   -0.5401    0.3536
```

**B3 =**

```
0.3536    0.6270    0.3913
0.3536   -0.4815   -0.2458
0.3536   -0.3361   -0.1004
0.3536    0.1906   -0.0451
0.3536    0.0451   -0.1906
0.3536    0.1004    0.3361
0.3536    0.2458    0.4815
0.3536   -0.3913   -0.6270
```

8. 已知由 MATLAB 指令创建的矩阵  $A = \text{gallery}(5)$ , 试对该矩阵进行特征值分解, 并通过验算观察发生的现象。

【答案】

用 `eig` 得到不正确的特征值

**ans =**

```
-0.0181                -0.0054 - 0.0171i   -0.0054 + 0.0171i
```

$$0.0144 - 0.0104i \quad 0.0144 + 0.0104i$$

应采用 jordan 才能得到正确的特征值

$$DJ = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

9. 求矩阵  $\mathbf{Ax} = \mathbf{b}$  的解, A 为 3 阶魔方阵, b 是  $(3 \times 1)$  的全 1 列向量。

〔答案〕

$$\mathbf{x} = \begin{pmatrix} 0.0667 \\ 0.0667 \\ 0.0667 \end{pmatrix}$$

10. 求矩阵  $\mathbf{Ax} = \mathbf{b}$  的解, A 为 4 阶魔方阵, b 是  $(4 \times 1)$  的全 1 列向量。

〔答案〕

解不唯一

$$\mathbf{x} = \begin{pmatrix} -0.0074 \\ -0.0809 \\ 0.1397 \\ 0.0662 \end{pmatrix} + \begin{pmatrix} 0.0588 \\ 0.1176 \\ -0.0588 \\ 0 \end{pmatrix} t$$

下面逆阵法的解不正确

$$\mathbf{xx} = \begin{pmatrix} 0.0313 \\ -0.1250 \\ 0.1250 \\ 0.0313 \end{pmatrix}$$

11. 求矩阵  $\mathbf{Ax} = \mathbf{b}$  的解,  $\mathbf{A}$  为 4 阶魔方阵,  $\mathbf{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ 。

【答案】

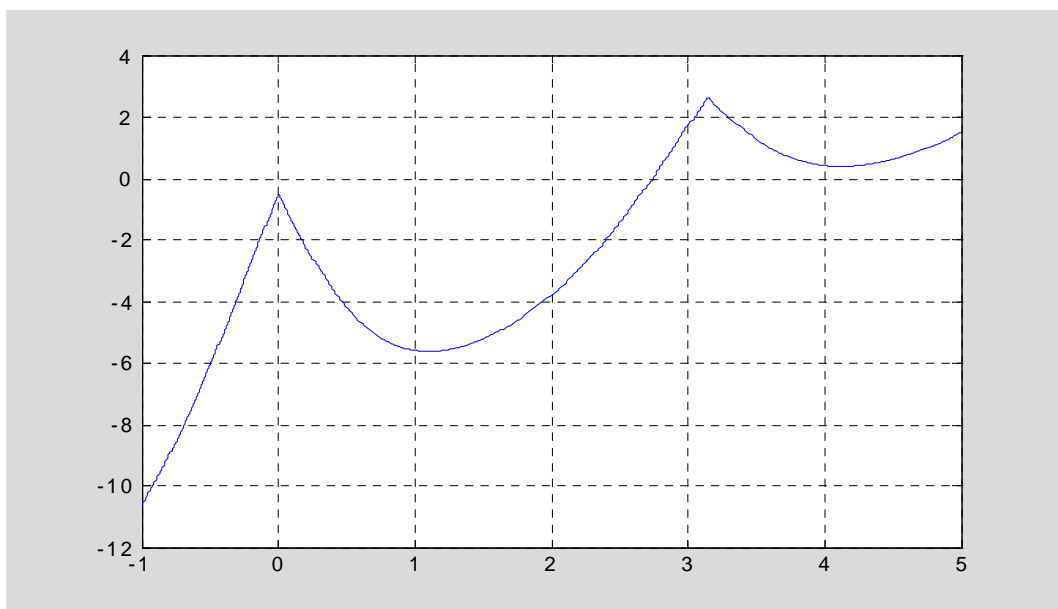
没有准确解, 以下是伪逆阵求得的最小二乘解

$\mathbf{x} =$   
 $\begin{bmatrix} 0.0235 \\ 0.1235 \\ 0.1235 \\ 0.0235 \end{bmatrix}$

12. 求  $-0.5 + t - 10e^{-0.2t} |\sin[\sin t]| = 0$  的实数解。

【答案】

$t =$   
 $2.7341$   
 $yt =$   
 $2.2204e-015$



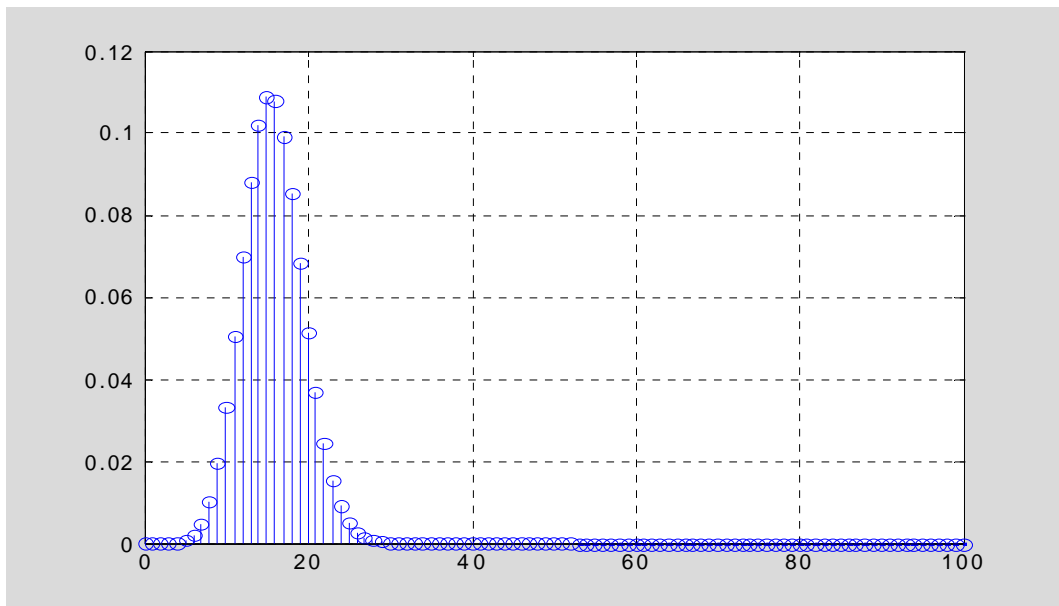
13. 求解二元函数方程组  $\begin{cases} \sin(x - y) = 0 \\ \cos(x + y) = 0 \end{cases}$  的解。(用符号法解)

〔答案〕

```
X =  
[ 1/4*pi]  
[ -1/4*pi]  
Y =  
[ 1/4*pi]  
[ -1/4*pi]
```

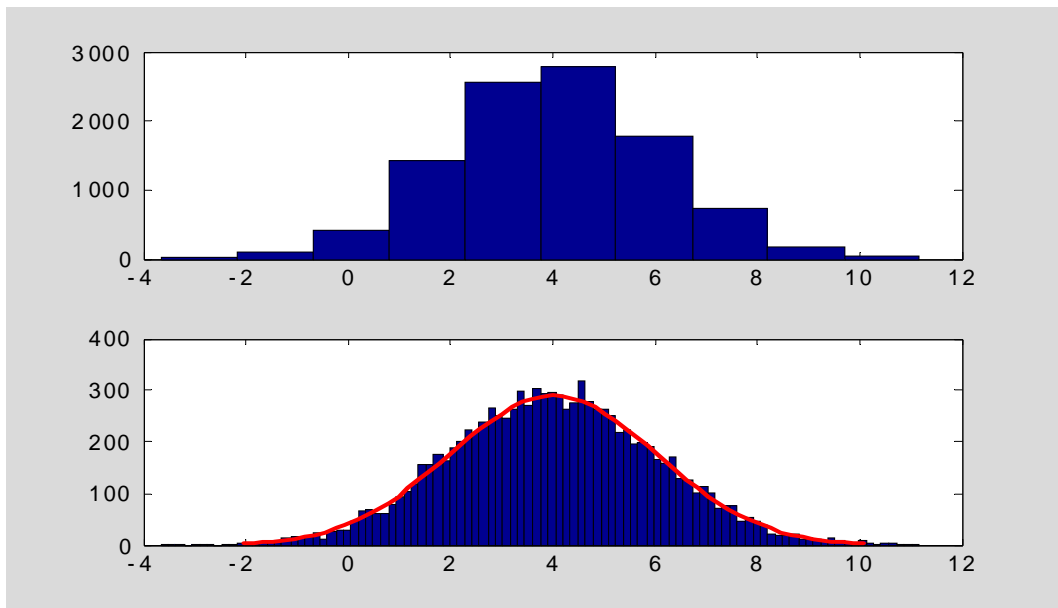
14. 假定某窑工艺瓷器的烧制成品合格率为 0.157，现该窑烧制 100 件瓷器，请画出合格产品数的概率分布曲线。

〔答案〕



15. 试产生均值为 4，标准差为 2 的(10000×1)的正态分布随机数组 a， 分别用 hist 和 histfit 绘制该数组的频数直方图，观察两张图形的差异。除 histfit 上的拟合红线外，你能使这两个指令绘出相同的频数直方图吗？

〔答案〕



16. 从数据文件 prob\_data4016.mat 得到随机数组 R，下面有一段求取随机数组全部数据最大值、均值和标准差的程序。

`Mx=max(max(R)),Me=mean(mean(R)),St=std(std(R)),`

试问该程序所得的结果都正确吗？假如不正确，请写出正确的程序。

【答案】

```
Mx1 =
    0.9997
Me1 =
    0.5052
S1 =
    0.2919
```

17. 已知有理分式  $R(x) = \frac{N(x)}{D(x)}$ ，其中  $N(x) = (3x^3 + x)(x^3 + 0.5)$ ，

$D(x) = (x^2 + 2x - 2)(5x^3 + 2x^2 + 1)$ 。(1)求该分式的商多项式  $Q(x)$  和余多项式  $r(x)$ 。

(2) 用程序验算  $D(x)Q(x) + r(x) = N(x)$  是否成立。

【答案】

```
Q =
    0.6000    -1.4400
```

```

r =
    -0.0000    0.0000    21.8800    -5.3400    -5.5200    4.5800
   -2.8800

```

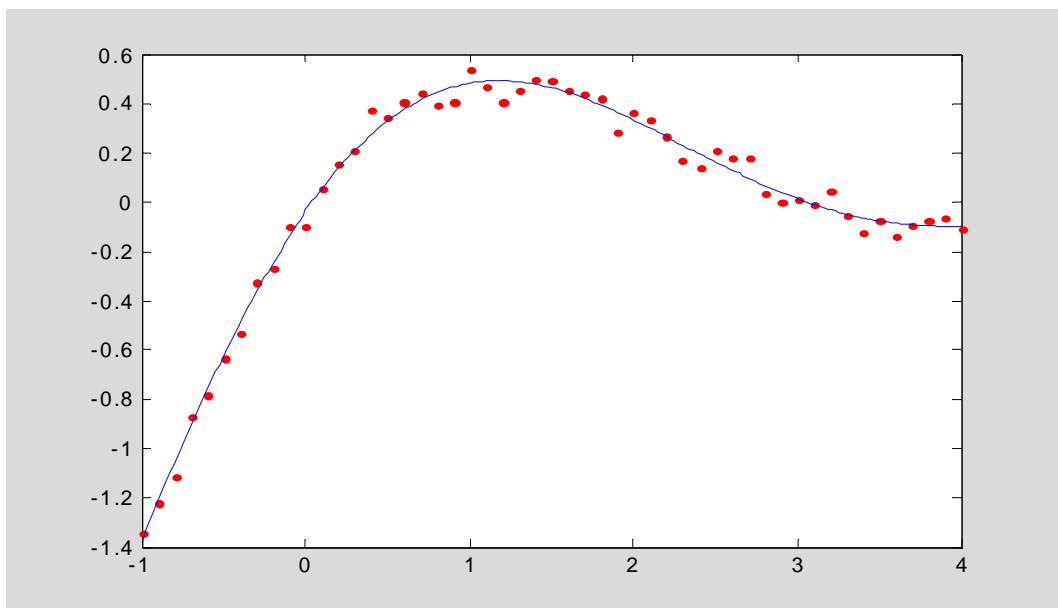
18. 现有一组实验数据  $x, y$  (数据从 prob\_data418.mat 获得), 试求这组数据的 5 阶拟合多项式。

【答案】

```

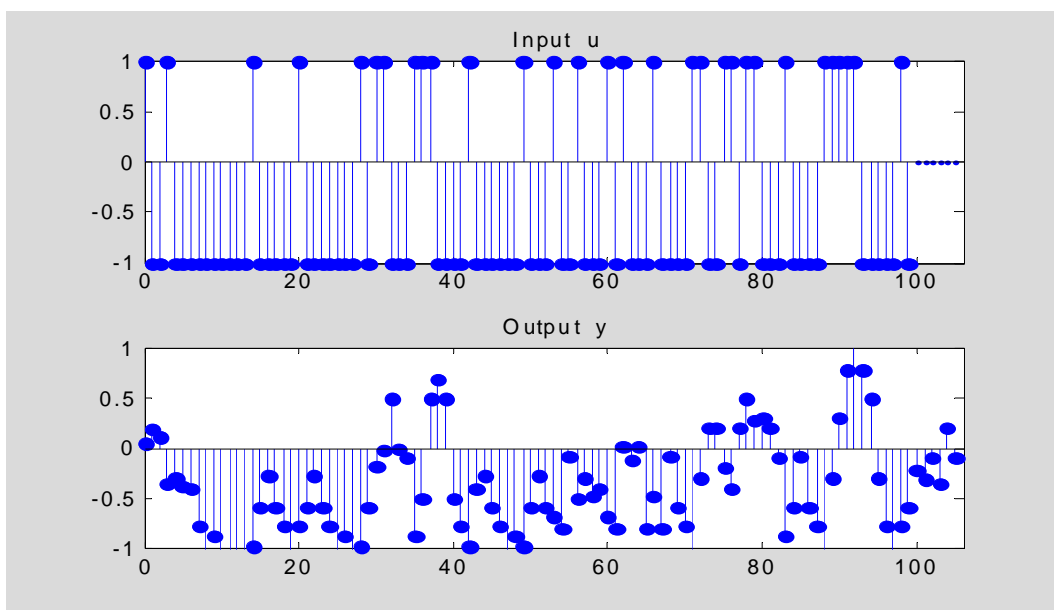
P =
   -0.0039    0.0338   -0.0227   -0.4456    0.9590   -0.0364

```



19. 已知系统冲激响应为  $h(n)=[0.05,0.24,0.40,0.24,0.15,-0.1,0.1]$  , 系统输入  $u(n)$ 由指令 `randn('state',1);u=2*(randn(1,100)>0.5)-1` 产生, 该输入信号的起始作用时刻为 0。试用直杆图 (提示: 用 `stem` 指令) 画出分别显示该系统输入、输出信号的两张子图。

【答案】



# 第 1 章 数据和函数的可视化

视觉是人们感受世界、认识自然的最重要依靠。数据可视化的目的在于：通过图形，从一堆杂乱的离散数据中观察数据间的内在关系，感受由图形所传递的内在本质。MATLAB 一向注重数据的图形表示，并不断地采用新技术改进和完备其可视化功能。

本章将系统地阐述：曲线、曲面绘制的基本技法和指令；如何使用线型、色彩、数据点标记凸现不同数据的特征；如何利用着色、灯光照明、烘托表现高维函数的性状；如何生成和运用标识，画龙点睛般地注释图形等。

本章的图形指令只涉及 MATLAB 的“高层”绘图指令。这种指令的形态和格式友善，易于理解和使用。整章内容遵循由浅入深、由基础到高级、由算例带归纳的原则。所有算例都是运行实例，易于读者实践试验，并从中掌握一般规律。

## 1.1 引导

### 1.1.1 离散数据和离散函数的可视化

众所周知：一对实数标量  $(x, y)$  可表示为平面上的一个点；进而，一对实数“向量”  $(\mathbf{x}, \mathbf{y})$  可表现为平面上的一组点。MATLAB 就是利用这种几何比拟法实现了离散数据可视化。

至于离散函数可视化的步骤是：先根据离散函数特征选定一组自变量



$\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ ；再根据所给离散函数  $y_n = f(x_n)$  算得相应的  $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$ ，

然后在平面上几何地表现这组向量对  $(\mathbf{x}, \mathbf{y})$ 。

【例 5.1-1】图形表示离散函数  $y = |n|$ 。（图 5.1-1）

```
n=(-10:10)';
y=abs(n);
plot(n,y,'r.','MarkerSize',20)
axis equal
grid on
xlabel('n')
```

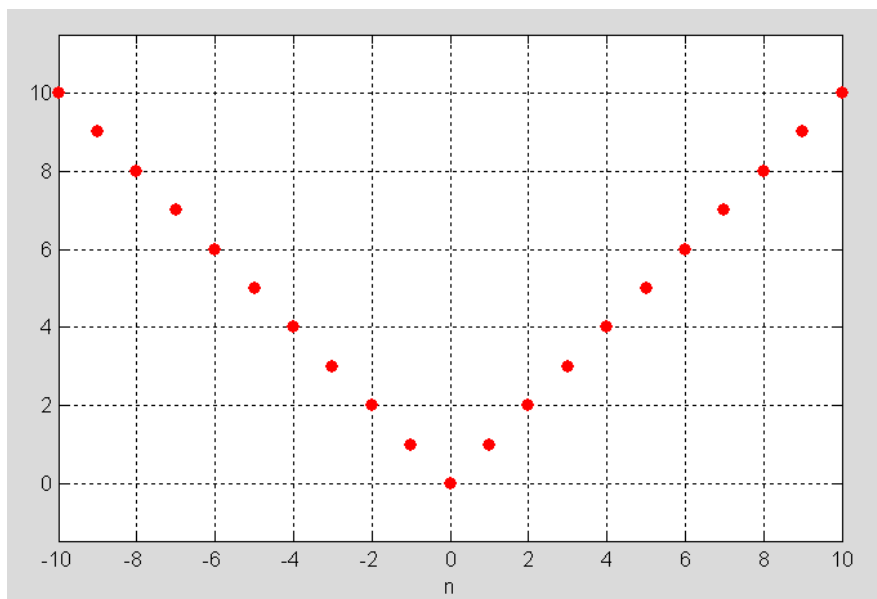


图 5.1-1 离散函数的可视化

## 1.1.2 连续函数的可视化

连续函数可视化包含三个重要环节：一，从连续函数获得一组采样数据，即选定一组自变量采样点（包括采样的起点、终点和采样步长），并计算相应的函数值；二，离散数据的可视化；三，图形上离散点的连续化。

【例 5.1-2】用图形表示连续调制波形  $y = \sin(t)\sin(9t)$ 。（图 5.1-2）

```
t1=(0:11)/11*pi;
t2=(0:400)/400*pi;
t3=(0:50)/50*pi;
y1=sin(t1).*sin(9*t1);
y2=sin(t2).*sin(9*t2);
y3=sin(t3).*sin(9*t3);
subplot(2,2,1),plot(t1,y1,'r.') %<7>
```

```

axis([0,pi,-1,1]),title('(1)点过少的离散图形')
subplot(2,2,2),plot(t1,y1,t1,y1,'r.') %<9>
axis([0,pi,-1,1]),title('(2)点过少的连续图形')
subplot(2,2,3),plot(t2,y2,'r.') %<11>
axis([0,pi,-1,1]),title('(3)点密集的离散图形')
subplot(2,2,4),plot(t3,y3) %<13>
axis([0,pi,-1,1]),title('(4)点足够的连续图形')

```

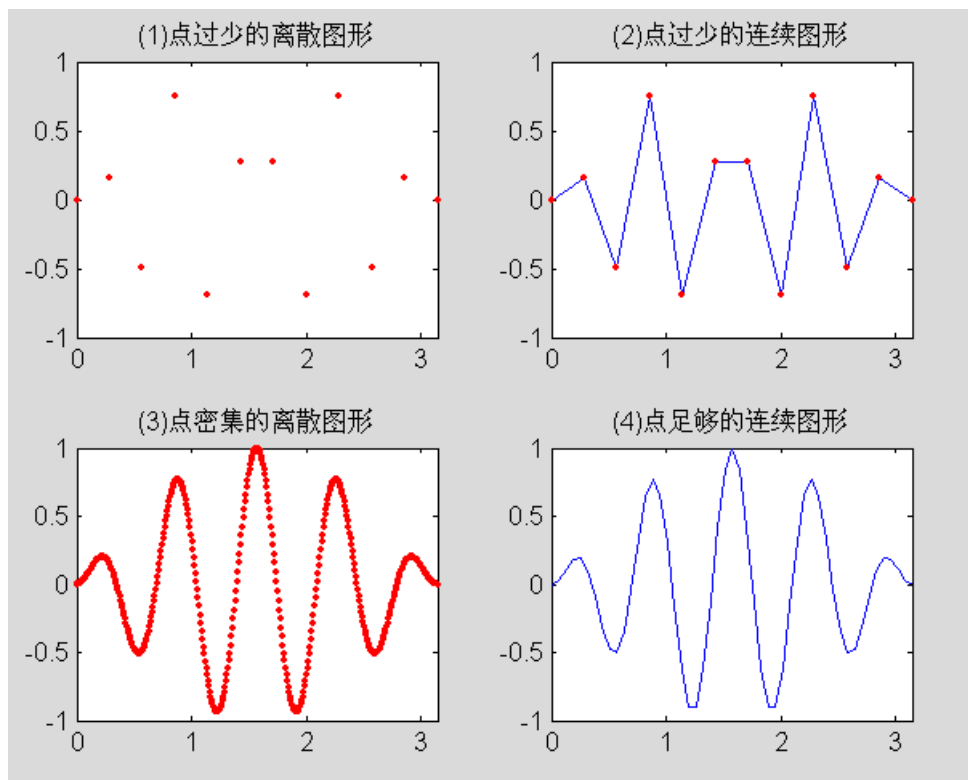


图 5.1-2 连续函数的图形表现方法

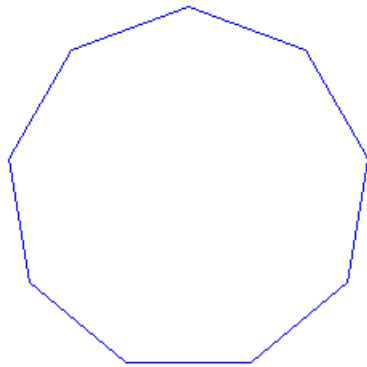
【例 5.1-3】绘制奇数正多边形及圆。(图 5.1-3)

```

N=9;
t=0:2*pi/N:2*pi;
x=sin(t);y=cos(t);
tt=reshape(t,2,(N+1)/2);
tt=flipud(tt);
tt=tt(:);
xx=sin(tt);yy=cos(tt);
subplot(1,2,1),plot(x,y)
title('(1) 正常排序图形'),axis equal off,shg
subplot(1,2,2),plot(xx,yy)
title('(2) 非正常排序图形'),axis equal off,shg

```

(1) 正常排序图形



(2) 非正常排序图形

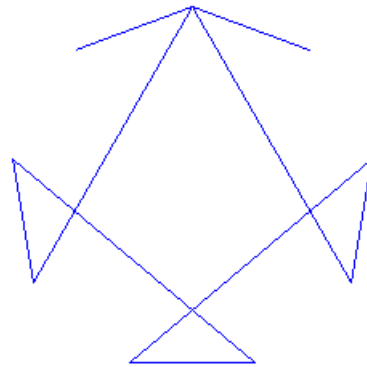


图 5.1-3 自变量排列次序对连续曲线图形的影响

## 1.2 二维曲线和图形

MATLAB 提供了多种二维图形的绘制指令（见表 5.2-1），但其中，最重要、最基本的指令是 **plot**。其他许多特殊绘图指令或以它为基础而形成，或使用场合较少。出于简明考虑，本节着重介绍 **plot** 的使用。

### 1.2.1 二维曲线绘制的基本指令 **plot**

#### 一 基本调用格式 **plot(x,y,'s')**

#### 二 **plot** 的衍生调用格式

##### （1）单色或多色绘制多条曲线

<b>plot(X,Y,'s')</b>	用 s 指定的点形线型色彩绘制多条曲线
<b>plot(X,Y)</b>	采用默认的色彩次序用细实线绘制多条曲线

##### （2）多三元组绘制多条曲线

**plot(X1,Y1,'s1',X2,Y2,'s2',... ,Xn,Yn,'sn')**

### (3) 单输入量绘线

**plot(Y)**

## 三 plot的属性可控调用格式

**plot(x, y, 's', 'PropertyName', PropertyValue, ...)**

【例 5.2-1】二维曲线绘图指令演示之一。(图 5.2-1)

```
clf
t=(0:pi/50:2*pi)';
k=0.4:0.1:1;
Y=cos(t)*k;
subplot(1,2,1),plot(t,Y,'LineWidth',1.5)
title('By plot(t,Y)'),xlabel('t')
subplot(1,2,2),plot(Y,'LineWidth',1.5)
title('By plot(Y)'),xlabel('row subscript of Y')
```

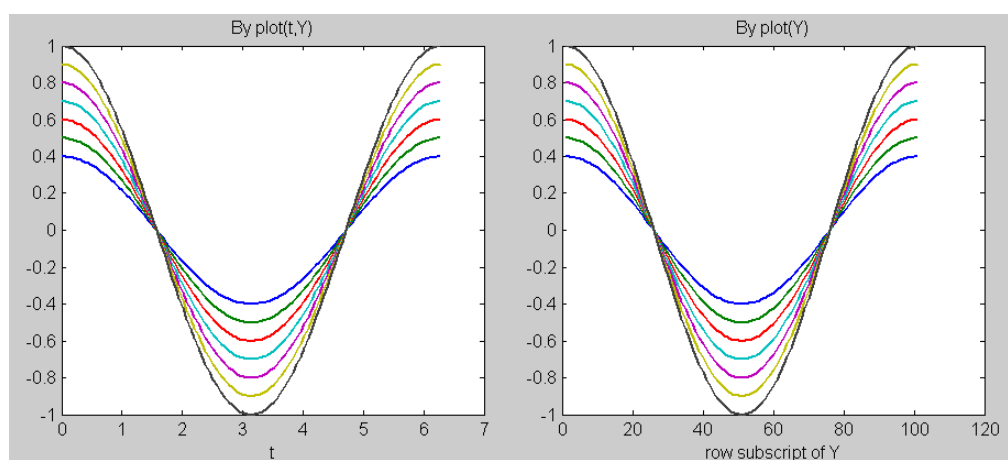


图 5.2-1 plot(t,Y)与 plot(Y)所绘曲线的区别

【例 5.2-2】用图形表示连续调制波形  $y = \sin(t)\sin(9t)$  及其包络线。(图 5.2-2)

```
t=(0:pi/100:pi)';           %<1>
y1=sin(t)*[1,-1];           %<2>
y2=sin(t).*sin(9*t);         %<3>
t3=pi*(0:9)/9;               %
                               %<4>
y3=sin(t3).*sin(9*t3);       %
                               %<5>
```

```

plot(t,y1,'r:',t,y2,'-bo') %
    <6>
hold on
plot(t3,y3,'s','MarkerSize',10,'MarkerEdgeColor',[0,1,0],'MarkerFaceC
olor',[1,0.8,0]) %
    <8>
axis([0,pi,-1,1]) %<9>
hold off %<10>
%以下指令供读者比较用。使用时，指令前的 % 号要去除。
%属性影响该指令中的所有线对象中的离散点。
%plot(t,y1,'r:',t,y2,'-bo',t3,y3,'s','MarkerSize',10,'MarkerEdgeColor
',[0,1,0],'MarkerFaceColor',[1,0.8,0]) % <10>

```

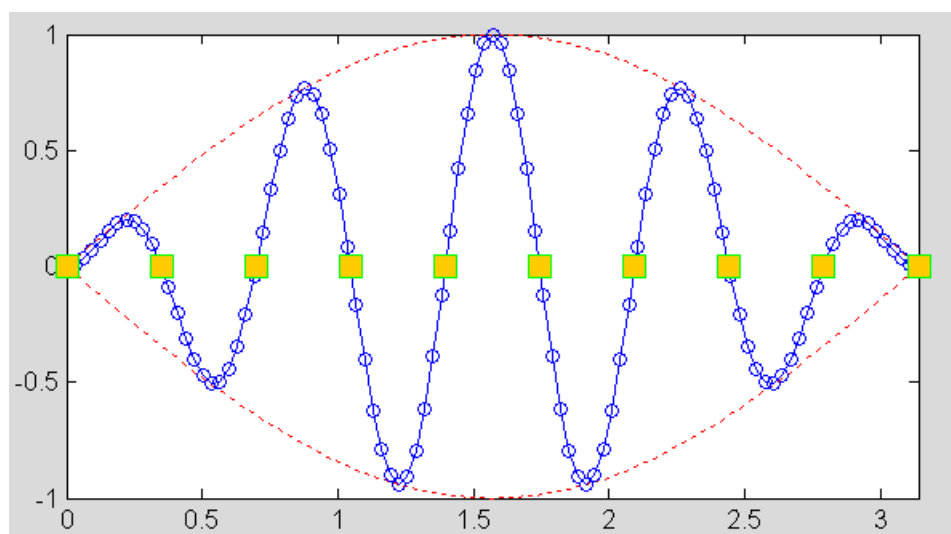


图 5.2-2 属性控制下所绘曲线

## 1.2.2 坐标控制和图形标识

MATLAB 对图形风格的控制比较完备友善。一方面，在最通用的层面上，它采用了一系列考虑周全的默认设置，因此在绘制图形时，无需人工干预，它就能根据所给数据自动地确定坐标取向、范围、刻度、高宽比，并给出相当令人满意的画面。另一方面，在适应用户的层面上，它又给出了一系列便于使用的指令，可让用户根据需要和喜欢去改变那些默认设置。

### 一 坐标轴的控制

【例 5.2-3】观察各种轴控制指令的影响。演示采用长轴为 3.25，短轴为 1.15 的椭圆。注意：采用多子图（图 5.2-3）表现时，图形形状不仅受“控制指令”影响，而且受整个图面“宽高比”及“子图数目”的影响。本书这样处理，是出于篇幅考虑。读者欲想准确体会控制指令的影响，请在全图状态下进行观察。

```

t=0:2*pi/99:2*pi;
x=1.15*cos(t);y=3.25*sin(t);

```

```

subplot(2,3,1),plot(x,y),axis normal,grid on,
title('Normal and Grid on')
subplot(2,3,2),plot(x,y),axis equal,grid on,title('Equal')
subplot(2,3,3),plot(x,y),axis square,grid on,title('Square')
subplot(2,3,4),plot(x,y),axis image,box off,title('Image and Box off')
subplot(2,3,5),plot(x,y),axis image fill,box off
title('Image and Fill')
subplot(2,3,6),plot(x,y),axis tight,box off,title('Tight')

```

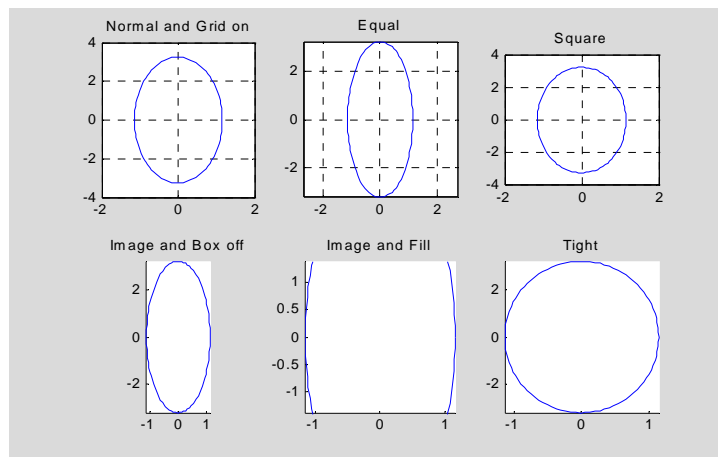


图 5.2-3 各种轴控制指令的不同影响

## 二 分格线和坐标框

<b>grid</b>	是否画分格线的双向切换指令（使当前分格线状态翻转）
<b>grid on</b>	画出分格线
<b>grid off</b>	不画分格线
<b>box</b>	坐标形式在封闭式和开启式之间切换指令
<b>box on</b>	使当前坐标呈封闭式
<b>box off</b>	使当前坐标呈开启形式

## 三 图形标识指令

图形标识包括：图名(Title)、坐标轴名(Label)、图形注释(Text)和图例(Legend)。标识指令的最简捷使用格式如下：

<b>title(S)</b>	书写图名
<b>xlabel(S)</b>	横坐标轴名
<b>ylabel(S)</b>	纵坐标轴名
<b>legend(S1,S2, ...)</b>	绘制曲线所用线型、色彩或数据点形图例
<b>text(xt,yt,S)</b>	在图面(xt, yt)坐标处书写字符注释

四 标识指令中字符的精细控制

表 5.2-7 图形标识用的希腊字母

指 令	字 符	指 令	字 符	指 令	字 符	指 令	字 符
<code>\alpha</code>	$\alpha$	<code>\eta</code>	$\eta$	<code>\Nu</code>	$\nu$	<code>\upsilon</code>	$\upsilon$
						<code>\Upsilon</code>	$\Upsilon$
<code>\beta</code>	$\beta$	<code>\theta</code>	$\theta$	<code>\xi</code>	$\xi$	<code>\phi</code>	$\phi$
		<code>\Theta</code>	$\Theta$	<code>\Xi</code>	$\Xi$	<code>\Phi</code>	$\Phi$
<code>\gamma</code>	$\gamma$	<code>\iota</code>	$\iota$	<code>\pi</code>	$\pi$	<code>\chi</code>	$\chi$
<code>\Gamma</code>	$\Gamma$			<code>\Pi</code>	$\Pi$		
<code>\delta</code>	$\delta$	<code>\kappa</code>	$\kappa$	<code>\rho</code>	$\rho$	<code>\psi</code>	$\psi$
<code>\Delta</code>	$\Delta$					<code>\Psi</code>	$\Psi$
<code>\epsilon</code>	$\epsilon$	<code>\lambda</code>	$\lambda$	<code>\sigma</code>	$\sigma$	<code>\omega</code>	$\omega$
		<code>\Lambda</code>	$\Lambda$	<code>\Sigma</code>	$\Sigma$	<code>\Omega</code>	$\Omega$
<code>\zeta</code>	$\zeta$	<code>\mu</code>	$\mu$	<code>\tau</code>	$\tau$		
使 用 示 例							
指 令	效果	指 令	效果	指 令		效果	
<code>'sin\beta'</code>	$\sin \beta$	<code>'\zeta\omega'</code>	$\zeta\omega$	<code>'\itA{\in}R^{m\times n}'</code>		$A \in R^{m \times n}$	

表 5.2-8 图形标识用的其他特殊字符

指 令	符	指 令	符	指 令	符	指 令	符	指 令	符
<code>\approx</code>	$\approx$	<code>\propto</code>	$\propto$	<code>\exists</code>	$\exists$	<code>\cap</code>	$\cap$	<code>\downarrow</code>	$\downarrow$
<code>\cong</code>	$\cong$	<code>\sim</code>	$\sim$	<code>\forall</code>	$\forall$	<code>\cup</code>	$\cup$	<code>\leftarrow</code>	$\leftarrow$
<code>\div</code>	$\div$	<code>\times</code>	$\times$	<code>\in</code>	$\in$	<code>\subset</code>	$\subset$	<code>\leftrightarrow</code>	$\leftrightarrow$
<code>\equiv</code>	$\equiv$	<code>\oplus</code>	$\oplus$	<code>\infty</code>	$\infty$	<code>\subseteq</code>	$\subseteq$	<code>\rightarrow</code>	$\rightarrow$
<code>\geq</code>	$\geq$	<code>\oslash</code>	$\oslash$	<code>\perp</code>	$\perp$	<code>\supset</code>	$\supset$	<code>\uparrow</code>	$\uparrow$
<code>\leq</code>	$\leq$	<code>\otimes</code>	$\otimes$	<code>\prime</code>	$\prime$	<code>\supseteq</code>	$\supseteq$	<code>\circ</code>	$\circ$
<code>\neq</code>	$\neq$	<code>\int</code>	$\int$	<code>\cdot</code>	$\cdot$	<code>\Im</code>	$\Im$	<code>\bullet</code>	$\bullet$
<code>\pm</code>	$\pm$	<code>\partial</code>	$\partial$	<code>\ldots</code>	$\ldots$	<code>\Re</code>	$\Re$	<code>\copyright</code>	$\copyright$

表 5.2-9 上下标的控制指令

	指令	arg 取值	举 例	
			示例指令	效 果

上标	<code>^{\arg}</code>	任何合法字符	<code>'\ite^{\-t}sint'</code>	$e^{-t} \sin t$
下标	<code>_{\arg}</code>	任何合法字符	<code>'x~{\chi}_{\alpha}^2(3)'</code>	$x \sim \chi_{\alpha}^2(3)$

表 5.2-10 字体式样设置规则

字体	指 令	arg 取 值	举 例	
			示例指令	效果
名称	<code>\fontname{\arg}</code>	arial ; courier ; roman ; 宋体; 隶书; 黑体 .....	<code>'\fontname{courier}Example 1'</code> <code>'\fontname{隶书}范例 2'</code>	Example 1 范例 2
风格	<code>\arg</code>	bf (黑体) it (斜体 一) sl (斜体 二) rm (正体)	<code>'\bfExample 3'</code> <code>'\itExample 4'</code>	<b>Example 3</b> <i>Example 4</i>
大小	<code>\fontsize{\arg}</code>	正整数。 默 认 值 为 10 (Points 磅)。	<code>'\fontsize{14}Example 5'</code> <code>'\fontsize{6}Example 6'</code>	<b>Example 5</b> Example 6

【例 5.2-4】本例非常简单，专供试验标识用（图 5.2-4）。读者在指令窗中反复调用这两条指令就可以检查自己对指令、标识的理解是否正确。当然每次试验时，第<5>条指令中的字符串应按读者的需要改变。

```

clf;t=0:pi/50:2*pi;
y=sin(t);
plot(t,y)
axis([0,2*pi,-1.2,1.2])
text(pi/2,1,'\fontsize{16}\leftarrow\itsin(t)\fontname{隶书}极大值')
%<5>

title('y=sin(t)')
xlabel('t')
ylabel('y')

```

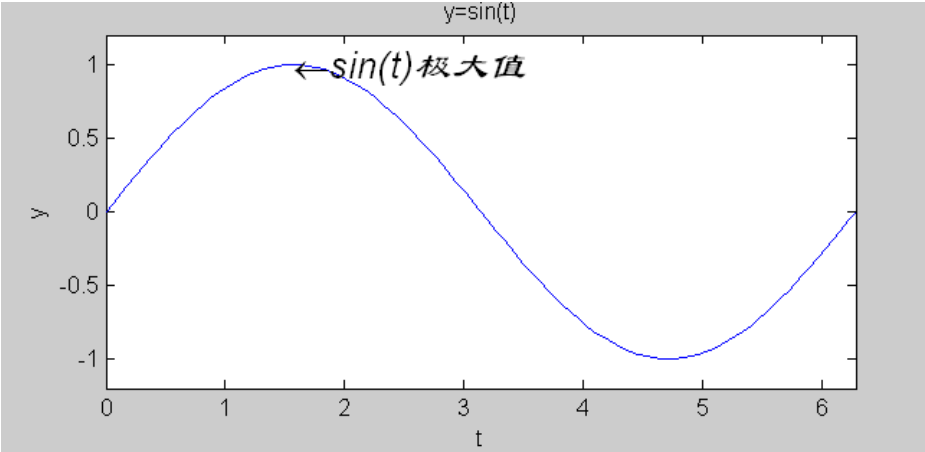




图 5.2-4 试验标识的图形

【例 5.2-5】通过绘制二阶系统阶跃响应，综合演示图形标识。本例比较综合，涉及的指令较广。请读者耐心读、实际做、再看例后说明，定会有匪浅收益。（图 5.2-5）

```

clf;t=6*pi*(0:100)/100;
y=1-exp(-0.3*t).*cos(0.7*t);
plot(t,y,'r-','LineWidth',3)
    %<3>
hold on
tt=t(find(abs(y-1)>0.05));ts=max(tt);           %<5>
plot(ts,0.95,'bo','MarkerSize',10)           %<6>
hold off
axis([-inf,6*pi,0.6,inf])
set(gca,'Xtick',[2*pi,4*pi,6*pi],'Ytick',[0.95,1,1.05,max(y)]) %<9>
set(gca,'XtickLabel',{'2*pi','4*pi','6*pi'}) %<10>
set(gca,'YtickLabel',{'0.95','1','1.05','max(y)'}) %<11>
grid on
text(13.5,1.2,'\fontsize{12}{\alpha}=0.3')
    %<13>
text(13.5,1.1,'\fontsize{12}{\omega}=0.7')
    %<14>

cell_string{1}='\fontsize{12}\uparrow';
    %<15>
cell_string{2}='\fontsize{16} \fontname{隶书}镇定时间';
cell_string{3}='\fontsize{6} ';
cell_string{4}=['\fontsize{14}\rmt_{s} = ' num2str(ts)];
    %<18>
text(ts,0.85,cell_string,'Color','b','HorizontalAlignment','Center')
    %<19>
title('\fontsize{14}\it y = 1 - e^{\ -\alpha t}cos{\omegat}') %<20>
xlabel('\fontsize{14} \bft \rightarrow')
ylabel('\fontsize{14} \bfy \rightarrow') %<22>

```

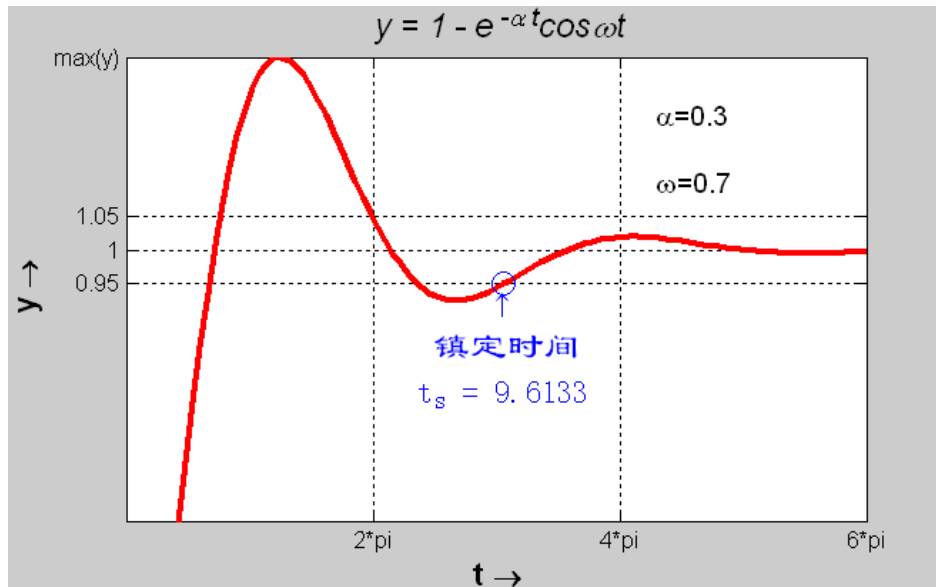


图 5.2-5 二阶阶跃响应图的标识

## 1.2.3 多次叠绘、双纵坐标和多子图

### 一 多次叠绘

**hold on** 使当前轴及图形保持而不被刷新，准备接受此后将绘制的新曲线。  
**hold off** 使当前轴及图形不再具备不被刷新的性质。  
**hold** 当前图形是否具备刷新性质的双向切换开关。

【例 5.2-6】利用 hold 绘制离散信号通过零阶保持器后产生的波形。（图 5.2-6）

```
t=2*pi*(0:20)/20;
y=cos(t).*exp(-0.4*t);
stem(t,y,'g','Color','k');
hold on
stairs(t,y,':r','LineWidth',3)
hold off
legend('\fontsize{14}\it stem','\fontsize{14}\it stairs')
box on
```

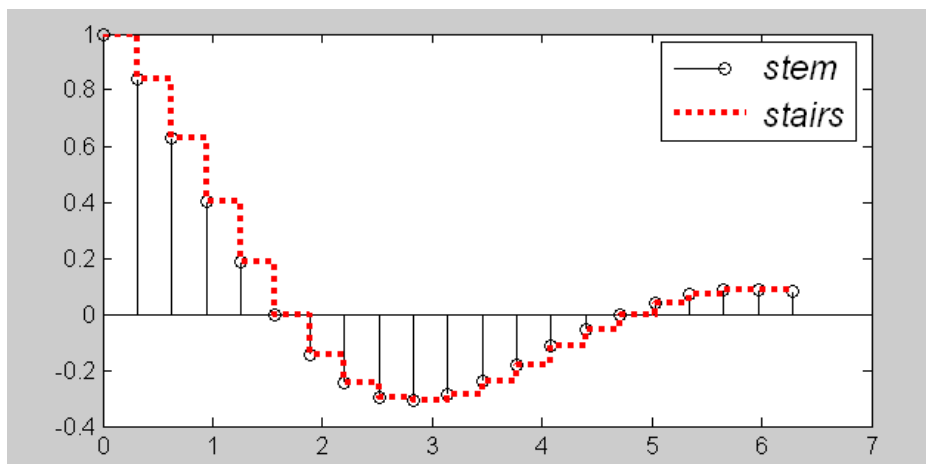


图 5.2-6 离散信号的重构

## 二 双纵坐标图

### **plotyy(X1,Y1,X2,Y2)**

以左、右不同纵轴绘制 X1-Y1、X2-Y2 两条曲线。

### **plotyy(X1,Y1,X2,Y2,'FUN')**

以左、右不同纵轴把 X1-Y1、X2-Y2 绘制成 FUN 指定形式的两条曲线。

### **plotyy(X1,Y1,X2,Y2,'FUN1','FUN2')**

以左、右不同纵轴把 X1-Y1、X2-Y2 绘制成 FUN1、FUN2 指定的不同形式的两条曲线。

【例 5.2-7】画出函数  $y = x \sin x$  和积分  $s = \int_0^x (x \sin x) dx$  在区间  $[0,4]$  上的曲线(图 5.2-7)。

```

clf;dx=0.1;x=0:dx:4;y=x.*sin(x);
s=cumtrapz(y)*dx;                                     %<2>
a=plotyy(x,y,x,s,'stem','plot');
    %<3>
text(0.5,1.5,'\fontsize{14}\ity=xsinx')
    %<4>
sint='\fontsize{16}\int_{\fontsize{8}0}^{\ x}';           %<5>
ss=['\fontsize{14}\its=',sint,'\fontsize{14}\itxsinx dx']; %<6>
text(2.5,3.5,ss)
    %<7>
set(get(a(1),'Ylabel'),'String','被积函数 \ity=xsinx')
    %<8>
set(get(a(2),'Ylabel'),'String',ss)
    %<9>
xlabel('x')

```

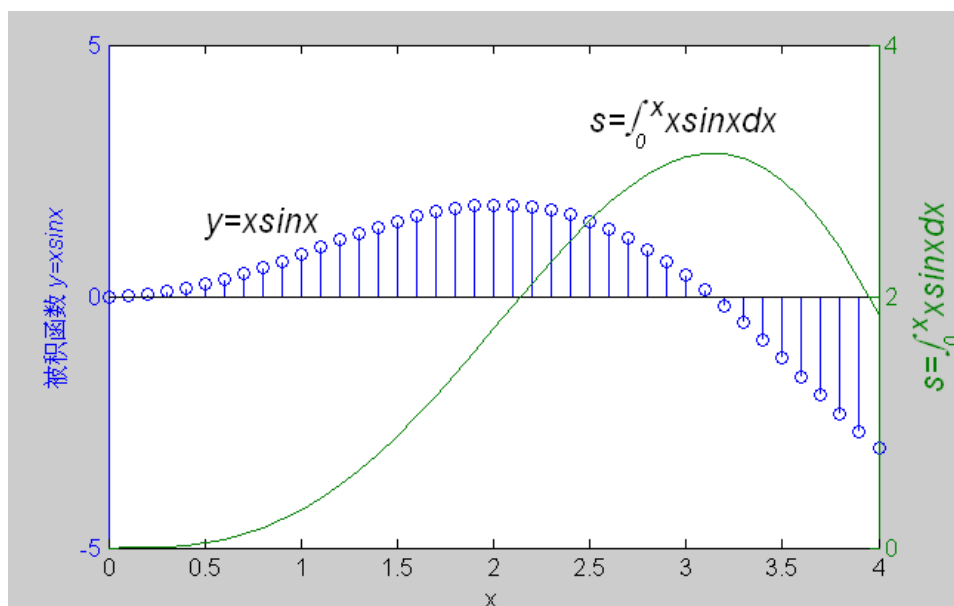


图 5.2-7 函数和积分

### 三 多子图

**subplot(m,n,k)**

使  $(m \times n)$  幅子图中的第  $k$  幅成为当前图

**subplot('position',[left bottom width height])** 在指定位置上开辟子图，并成为当前图。

【例 5.2-8】演示 subplot 指令对图形窗的分割（图 5.2-8）。

```
clf;t=(pi*(0:1000)/1000)';
y1=sin(t);y2=sin(10*t);y12=sin(t).*sin(10*t);
subplot(2,2,1),plot(t,y1);axis([0,pi,-1,1])
subplot(2,2,2),plot(t,y2);axis([0,pi,-1,1])
subplot('position',[0.2,0.1,0.6,0.4])
plot(t,y12,'b-',t,[y1,-y1],'r:')
axis([0,pi,-1,1])
```

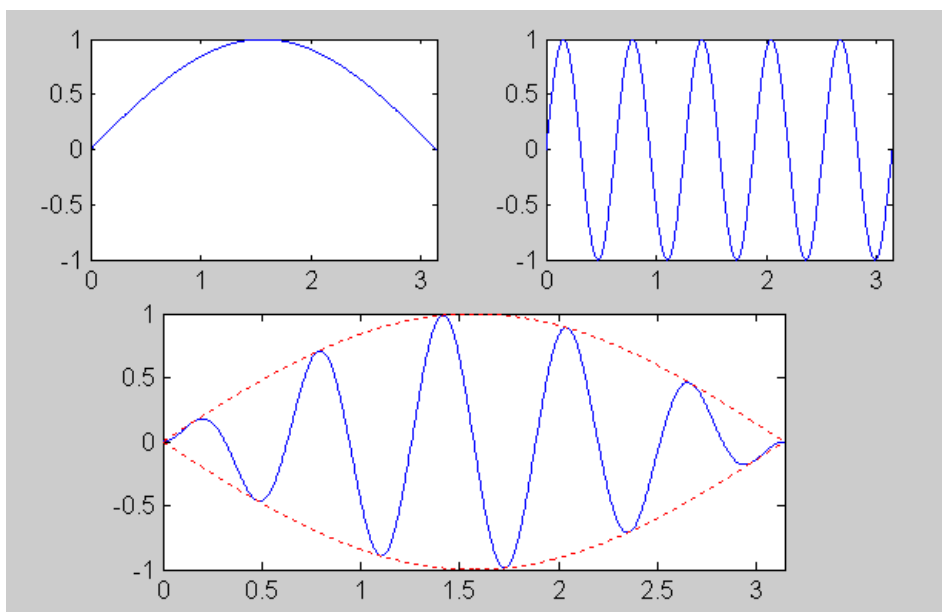


图 5.2-8 多子图的布置

## 1.2.4 获取二维图形数据的指令ginput

`[x,y]=ginput(n)` 用鼠标从二维图形上获取  $n$  个点的数据坐标  $(x,y)$ 。

【例 5.2-9】采用图解法求  $(x+2)^x = 2$  的解。

(1)

```
clf
x=-1:0.01:5;
y=(x+2).^x-2;
plot(x,y)
grid on
```

(2)

```
[x,y]=ginput(1);
```

(3)

```
format long
x,y
x =
    0.69828692903537
y =
   -5.884401711382421e-010
```

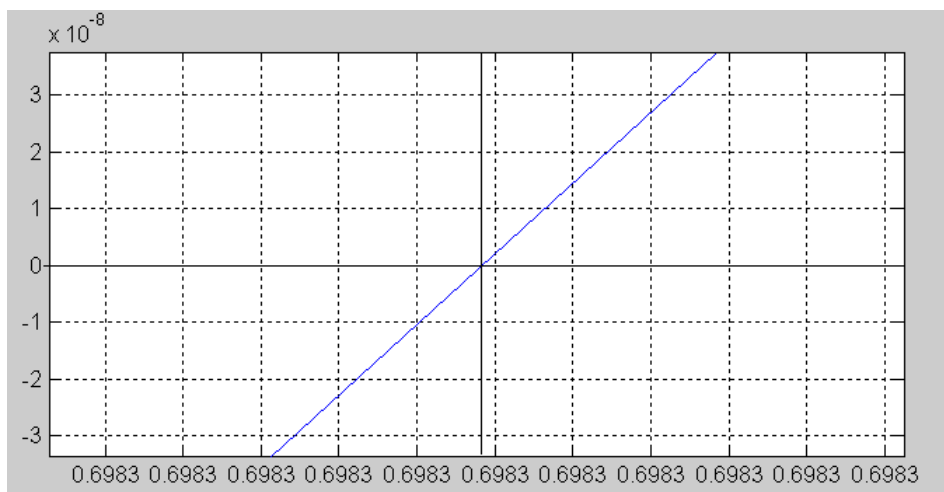


图 5.2-9 局部放大图

## 1.3 三维曲线和曲面

### 1.3.1 三维线图指令plot3

**plot3(X,Y,Z,'s')**

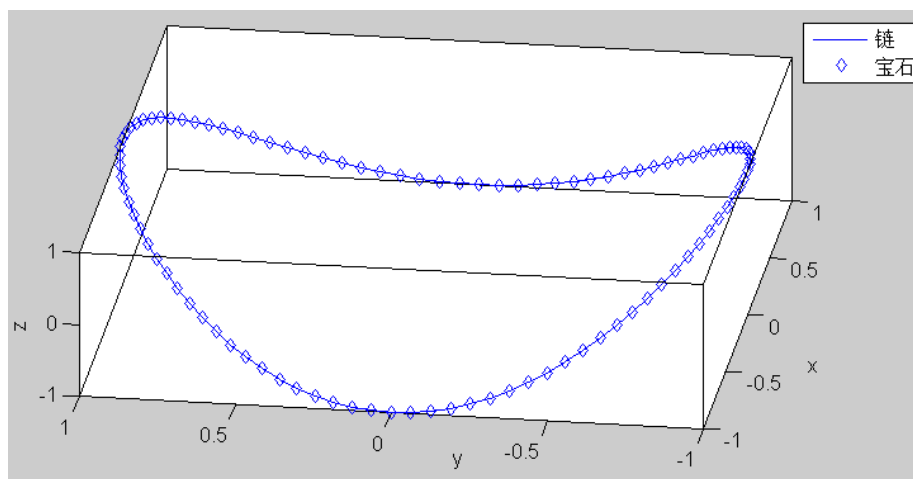
用 s 指定的点形线型色彩绘制曲线

**plot3(X1,Y1,Z1,'s1',X2,Y2,Z2,'s2', ...)**

用 s1, s2 指定的点形线型色彩绘制多类曲线

【例 5.3-1】三维曲线绘图。本例演示：三维曲线的参数方程；线型、点形和图例。（图 5.3-1）

```
t=(0:0.02:2)*pi;
x=sin(t);y=cos(t);z=cos(2*t);
plot3(x,y,z,'b-','x,y,z','bd')
view([-82,58]),box on
xlabel('x'),ylabel('y'),zlabel('z')
legend('链','宝石')
```



## 1.3.2 三维曲面/网线图

### 一 三维曲面/网线可视化的数据准备

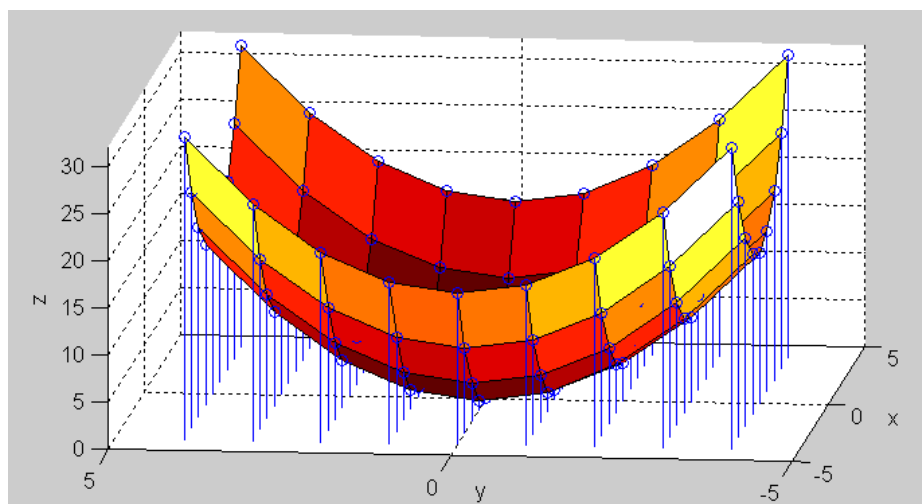
画函数  $z = f(x, y)$  所代表的三维空间曲面，需要做以下数据准备：

### 二 绘制曲面/网线图的基本指令

<b>surf(Z)</b>	以 Z 矩阵列、行下标为 $x, y$ 轴自变量，画曲面图。
<b>surf(X,Y,Z)</b>	最常用的曲面图调用格式
<b>surf(X,Y,Z,C)</b>	最完整调用格式，画由 C 指定用色的曲面图。
<b>mesh(Z)</b>	以 Z 矩阵列、行下标为 $x, y$ 轴自变量，画网线图。
<b>mesh(X,Y,Z)</b>	最常用的网线图调用格式
<b>mesh(X,Y,Z,C)</b>	最完整调用格式，画由 C 指定用色的网线图。

【例 5.3-2】用曲面图表现函数  $z = x^2 + y^2$ 。（图 5.3-2）。

```
clf
x=-4:4;y=x;
[X,Y]=meshgrid(x,y);
Z=X.^2+Y.^2;
surf(X,Y,Z);
colormap(hot)
hold on
stem3(X,Y,Z,'bo')
hold off
xlabel('x'),ylabel('y'),zlabel('z')
axis([-5,5,-5,5,0,inf])
view([-84,21])
```



● 图 5.3-2 曲面图和格点

●  
●

### 1.3.3 曲面/网线图的精细修饰

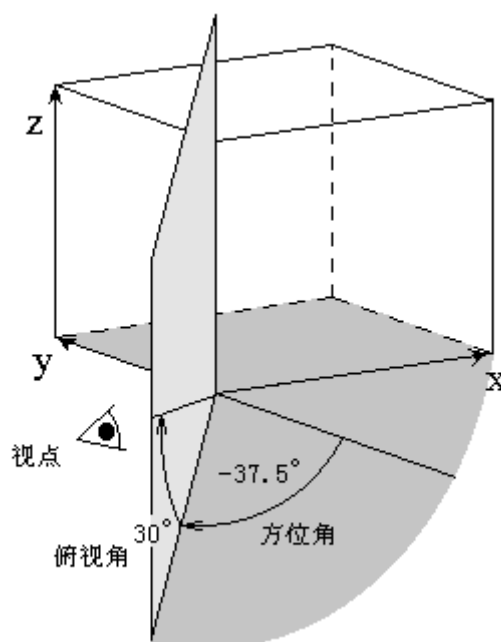
#### 一 视角控制view

**view([az,el])**

通过方位角、俯视角设置视点

**view([vx,vy,vz])**

通过直角坐标设置视点



● 图 5.3-3 视点设置参数示意

#### 二 色图colormap

**colormap(CM)**

设置当前图形窗的着色色图为 CM



表 5.3-1 MATLAB 的预定义色图矩阵 CM

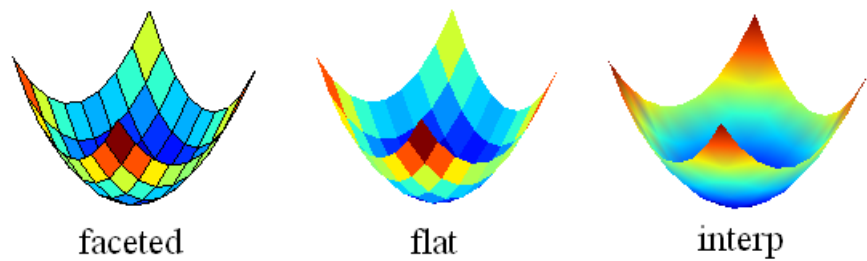
CM	含 义	CM	含 义
autumn	红、黄浓淡色	jet	蓝头红尾饱和值色
bone	蓝色调浓淡色	lines	采用 plot 绘线色
colorcube	三浓淡多彩交错色	pink	淡粉红色图
cool	青、品红浓淡色	prism	光谱交错色
copper	纯铜色调线性浓淡色	spring	青、黄浓淡色
flag	红-白-蓝-黑交错色	summer	绿、黄浓淡色
gray	灰色调线性浓淡色	winter	蓝、绿浓淡色
hot	黑、红、黄、白浓淡色	white	全白色
hsv	两端为红的饱和值色		
注意：jet 时默认色图。			

三 浓淡处理shading

shading options          图形对象着色的浓淡处理

【例 5.3-3】三种浓淡处理方式比较。（图 5.3-4）

```
clf
x=-4:4;y=x;
[X,Y]=meshgrid(x,y);
Z=X.^2+Y.^2;
surf(X,Y,Z)
colormap(jet)
subplot(1,3,1),surf(Z),axis off
subplot(1,3,2),surf(Z),axis off,shading flat
subplot(1,3,3),surf(Z),axis off,shading interp
set(gcf,'Color','w')
```



● 图 5.3-4 三种浓淡处理方式比较

## 四 透明度控制 **alpha**

**alpha(v)** 对面、块、象三种图形对象的透明度加以控制

【例 5.3-4】半透明的表面图。（见图 5.3-5）

```
clf
surf(peaks)
shading interp
alpha(0.5)
colormap(summer)
```

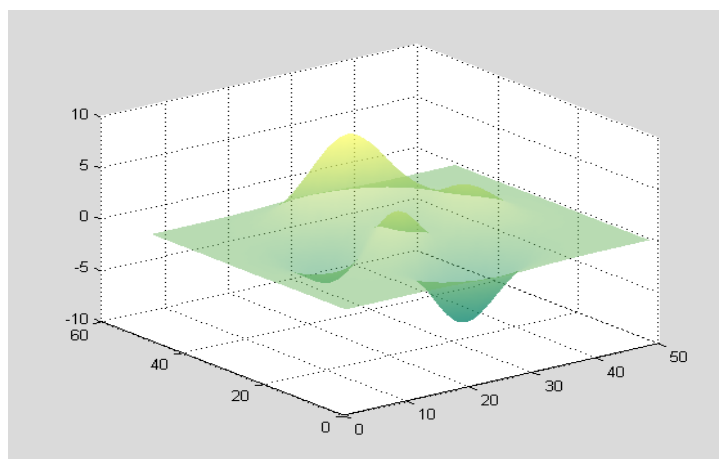


图 5.3-5 半透明薄膜

## 五 灯光设置**light**

**light('color',option1,'style',option2,'position',option3)** 灯光设置

## 六 照明模式**lighting**

**lighting options** 设置照明模式

## 七 控制光反射的材质指令**material**

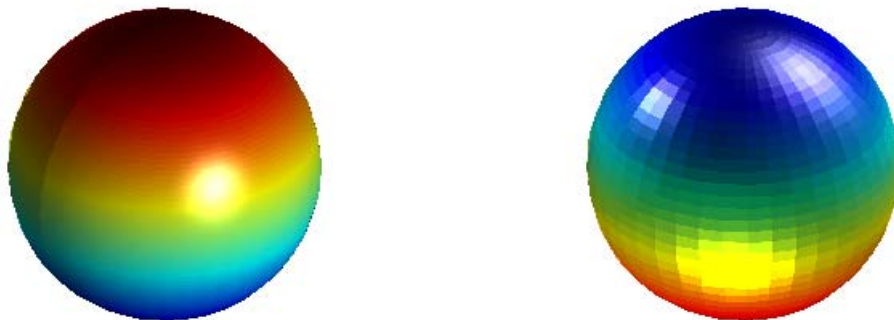
**material options** 使用预定义反射模式

【例 5.3-5】灯光、照明、材质指令所表现的图形。（图 5.3-6）

```

clf;
[X,Y,Z]=sphere(40);
colormap(jet)
    %<3>
subplot(1,2,1),surf(X,Y,Z),axis equal off,shading interp
    %<4>
light ('position',[0 -10 1.5],'style','infinite')
    %<5>
lighting phong
    %<6>
material shiny
    %<7>
subplot(1,2,2),surf(X,Y,Z,-Z),axis equal off,shading flat    %<8>
light;lighting flat
    %<9>
light('position',[-1,-1,-2],'color','y')    %<10>
light('position',[-1,0.5,1],'style','local','color','w')
    %<11>
set(gcf,'Color','w')

```



- 图 5.3-6 灯光、照明、材质指令所表现的图形

- 
- 

### 1.3.4 透视、镂空和裁切

#### 一 图形的透视

**hidden off**      透视被叠压的图形  
**hidden on**      消隐被叠压的图形

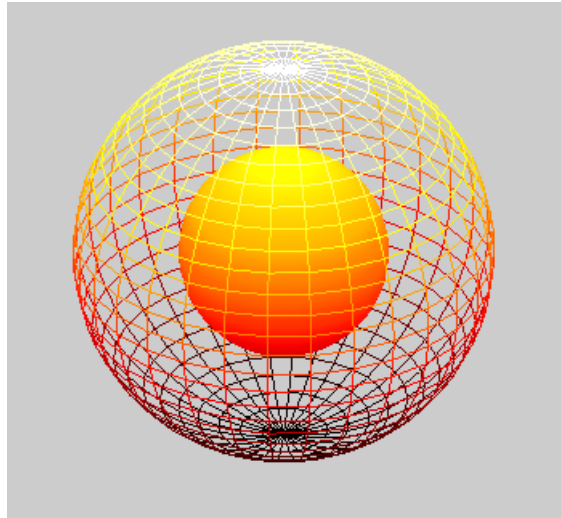
【例 5.3-6】透视演示（图 5.3-7）。

```
[X0,Y0,Z0]=sphere(30);
```

```

X=2*X0;Y=2*Y0;Z=2*Z0;
surf(X0,Y0,Z0);
shading interp
hold on,mesh(X,Y,Z),colormap(hot)
hold off
hidden off
axis equal,axis off

```



● 图 5.3-7 剔透玲珑球

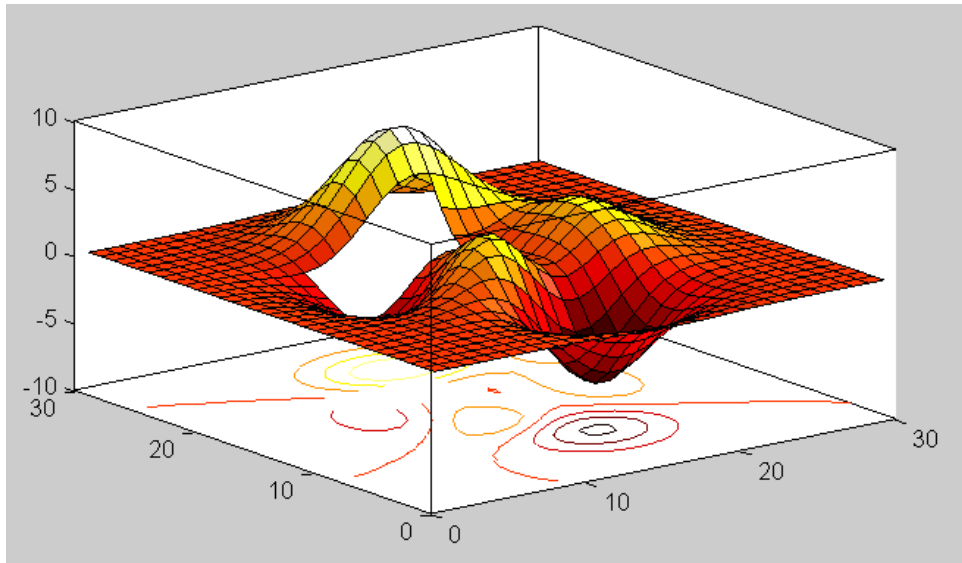
## 二 图形的镂空

【例 5.3-7】演示：如何利用“非数”NaN，对图形进行镂空处理。（图 5.3-8）

```

P=peaks(30);
P(18:20,9:15)=NaN;
surfc(P);
colormap(hot)
light('position',[50,-10,5])
material([0.9,0.9,0.6,15,0.4])
grid off,box on

```



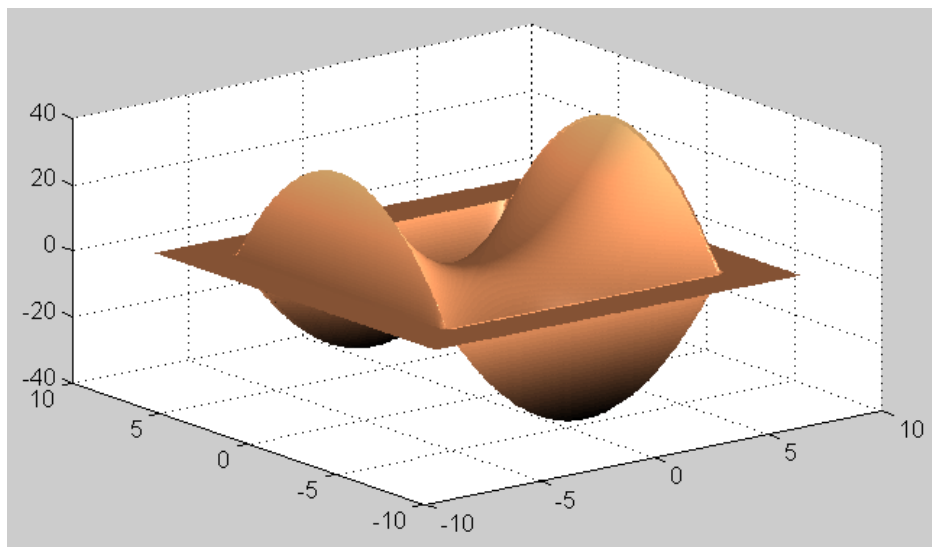
● 图 5.3-8 鞍方孔的曲面

●  
●

### 三 裁切

【例 5.3-8】表现切面（图 5.3-9）。

```
clf,x=[-8:0.1:8];y=x;[X,Y]=meshgrid(x,y);ZZ=X.^2-Y.^2;
ii=find(abs(X)>6|abs(Y)>6);
ZZ(ii)=zeros(size(ii));
surf(X,Y,ZZ),shading interp,colormap(copper)
light('position',[0,-15,1]);lighting phong
material([0.8,0.8,0.5,10,0.5])
```



● 图 5.3-9 经裁切处理后的图形

●  
●

## 1.4 高维可视化

### 1.4.1 二维半图指令pcolor, contour, contourf

【例 5.4-1】“二维半”指令 pcolor, contour, contourf 的应用。(图 5.4-1)

```
clf;clear;[X,Y,Z]=peaks(40);  
n=6;  
subplot(1,2,1),  
pcolor(X,Y,Z)  
shading interp  
zmax=max(max(Z));zmin=min(min(Z));  
caxis([zmin,zmax])  
colorbar  
hold on  
C=contour(X,Y,Z,n,'k:');  
clabel(C)  
hold off  
subplot(1,2,2)  
[C,h]=contourf(X,Y,Z,n,'k:');  
clabel(C,h)  
colormap(cool)  
set(gcf,'Color','w')
```

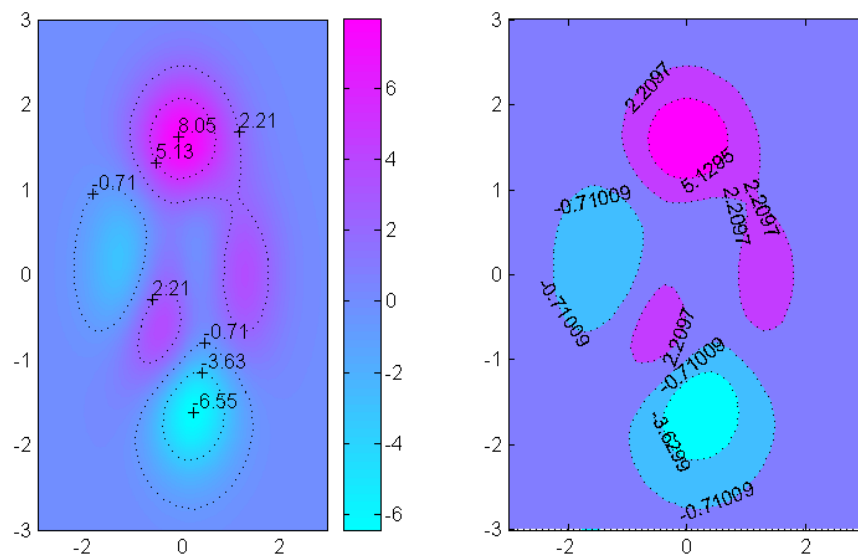


图 5.4-1 “二维半”指令的演示

### 1.4.2 四维表现

人对自然界的理解和思维是多维的。人的感官不仅善于接受一维、二维、三维的几何信息，而且对几何物体的运动，对颜色、声音、气味、触感等反应灵敏。从这意义上讲，MATLAB

色彩控制、动画等指令为四维或更高维表现提供了手段。

## 一 准四维表现

【例 5.4-2】用颜色表现  $z = f(x, y)$  函数的其他特征（如梯度、曲率等）。（图 5.4-2）

```
clf
x=3*pi*(-1:1/15:1);y=x;[X,Y]=meshgrid(x,y);
R=sqrt(X.^2+Y.^2)+eps;Z=sin(R)./R;
[dzdx,dzdy]=gradient(Z);
dzdr=sqrt(dzdx.^2+dzdy.^2);           %<4>
dz2=del2(Z);                           %<5>
subplot(1,2,1),surf(X,Y,Z,abs(dzdr))
shading faceted;
colorbar('SouthOutside')
brighten(0.6);
colormap hsv
title('No. 1    surf(X,Y,Z,abs(dzdr))')
subplot(1,2,2);surf(X,Y,Z,abs(dz2))
shading faceted
colorbar('NorthOutside')
title('No. 2    surf(X,Y,Z,abs(dz2))')
```

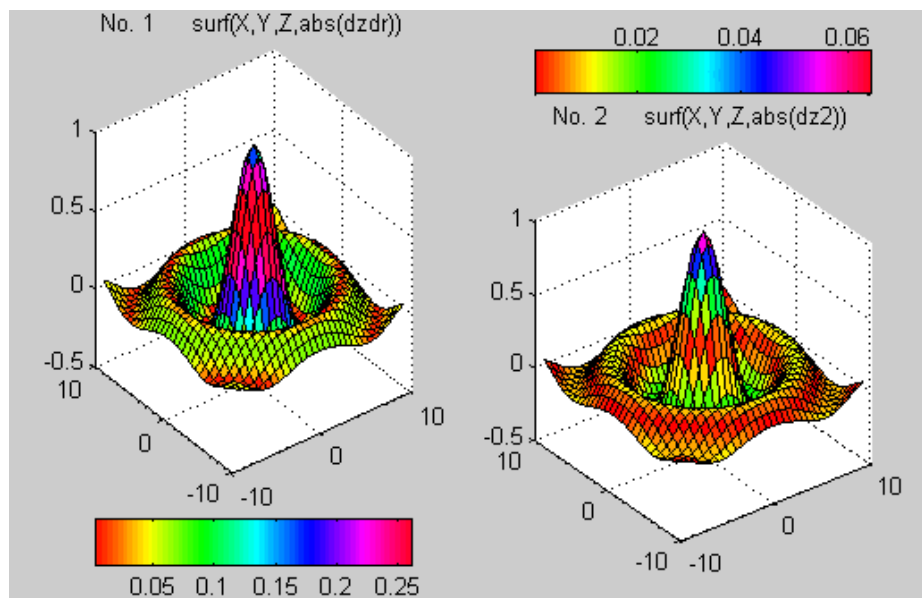


图 5.4-2 色彩分别表现函数的径向导数和曲率特征

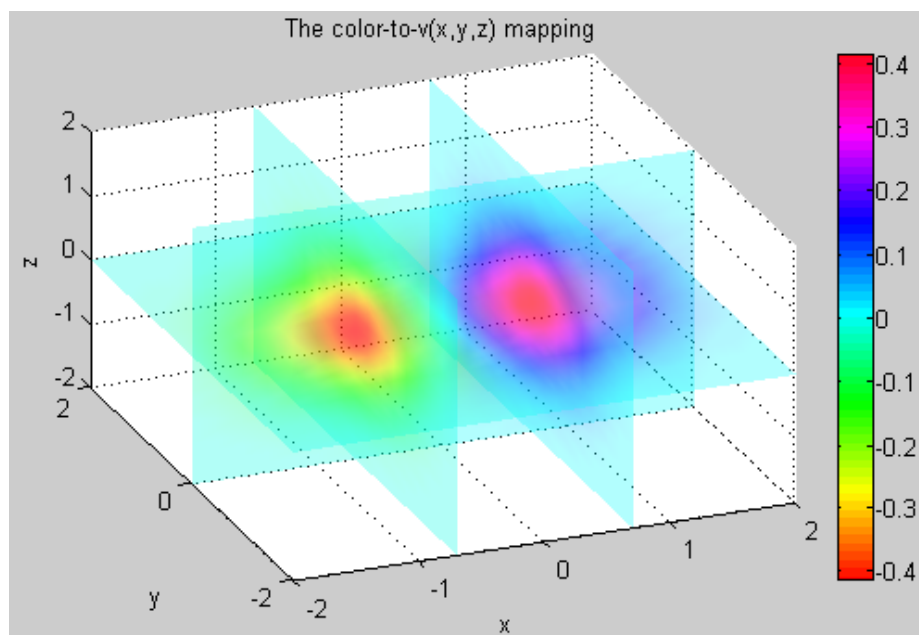
## 二 切片图

`[X,Y,Z]=meshgrid(x,y,z)`  
`slice(X,Y,Z,V,sx,sy,sz)`

由采样向量产生三维自变量“格点”数组  
三元函数切片图

【例 5.4-3】图形表现  $v = xe^{-x^2-y^2-z^2}$ 。(图 5.4-3)

```
clf
[x,y,z] = meshgrid(-2:.2:2,-2:.25:2,-2:.16:2);
v = x.*exp(-x.^2-y.^2-z.^2);
xs = [-0.7,0.7]; ys = 0; zs = 0;
slice(x,y,z,v,xs,ys,zs)
colorbar
shading interp
colormap hsv
xlabel('x'),ylabel('y'),zlabel('z')
title('The color-to-v(x,y,z) mapping')
view([-22,39])
alpha(0.3)
```



- 图 5.4-3 切片图
- 
- 

### 1.4.3 动态图形

在 MATLAB 的“上层”图形指令中的彗星轨线指令、色图变幻指令、影片动画指令，能很方便地使图形及色彩产生动态变化效果。

由于在 Notebook 和硬拷贝下，这种动态变化效果都无法表现，因此本节所有例题都不提供图形，而只给出有关指令。当读者在 MATLAB 指令窗中运作这些指令后，便可在图形窗中看到相应的动态图形。



## 一 彗星状轨迹图

`comet(x,y,p)`            二维彗星轨线  
`comet3(x,y,z,p)`        三维彗星轨线

【例 5.4-4】简单二维彗星示例。（请读者自己在指令窗中运行以下指令。）

```
shg;n=2;t=n*pi*(0:0.000005:1);x=sin(t);y=cos(t);  
plot(x,y,'g');axis square  
hold on  
comet(x,y,0.0001)  
hold off
```

## 二 色图的变幻

`spinmap`  
`spinmap(t)`  
`spinmap(inf)`  
`spinmap(t,inc)`

【例 5.4-5】指令 `spinmap` 的应用。（图 5.4-4）

```
ezsurf('x*y','circ');shading flat;view([-18,28])  
C=summer;  
CC=[C;flipud( C )];  
colormap(CC)  
spinmap(30,4)
```

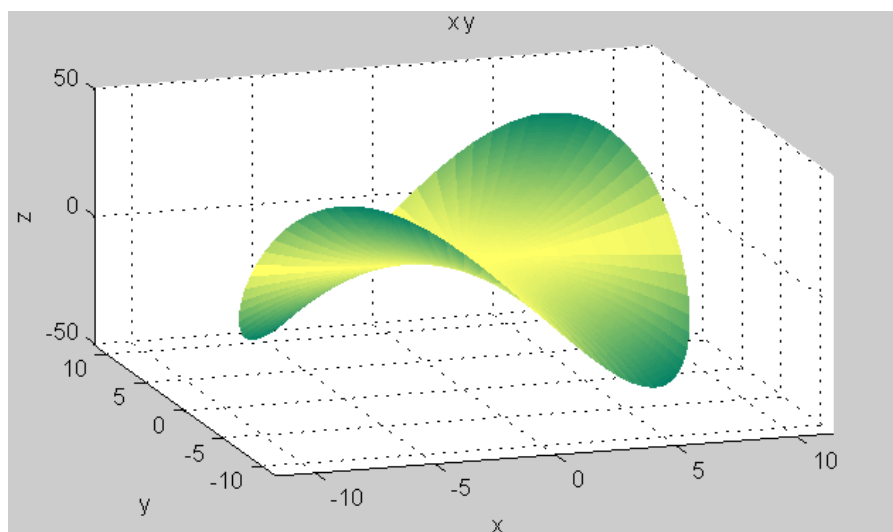


图 5.4-4 用于色图变幻演示的图形

### 三 影片动画

```
M(i)=getframe
movie(M,k)
```

【例 5.4-6】三维图形的影片动画。（因印刷关系，无法表现本例。请读者自己在指令窗中运行以下指令。）

```
clf;shg,x=3*pi*(-1:0.05:1);y=x;[X,Y]=meshgrid(x,y);
R=sqrt(X.^2+Y.^2)+eps; Z=sin(R)./R;
h=surf(X,Y,Z);colormap(jet);axis off
n=12;mmm=moviein(n);
for i=1:n
    rotate(h,[0 0 1],25);
    mmm(:,i)=getframe;
end
movie(mmm,5,10)
```

### 四 实时动画

【例 5.4-7】制作红色小球沿一条带封闭路径的下旋螺线运动的实时动画（图 5.4-5）。

(1) anim\_zzy1.m

```
function f=anim_zzy1(K,ki)
```

```
% anim_zzy1.m          演示红色小球沿一条封闭螺线运动的实时动画
```

```
% 仅演示实时动画的调用格式为          anim_zzy1(K)
```

```
% 既演示实时动画又拍摄照片的调用格式为          f=anim_zzy1(K,ki)
```

```
% K          红球运动的循环数（不小于1）
```

```
% ki          指定拍摄照片的瞬间，取 1 到 1034 间的任意整数。
```

```
% f          存储拍摄的照片数据，可用image(f.cdata)观察照片。
```

```
%
t1=(0:1000)/1000*10*pi;x1=cos(t1);y1=sin(t1);z1=-t1;
t2=(0:10)/10;x2=x1(end)*(1-t2);y2=y1(end)*(1-t2);z2=z1(end)*ones(size(x2));
t3=t2;z3=(1-t3)*z1(end);x3=zeros(size(z3));y3=x3;
t4=t2;x4=t4;y4=zeros(size(x4));z4=y4;
x=[x1 x2 x3 x4];y=[y1 y2 y3 y4];z=[z1 z2 z3 z4];
```

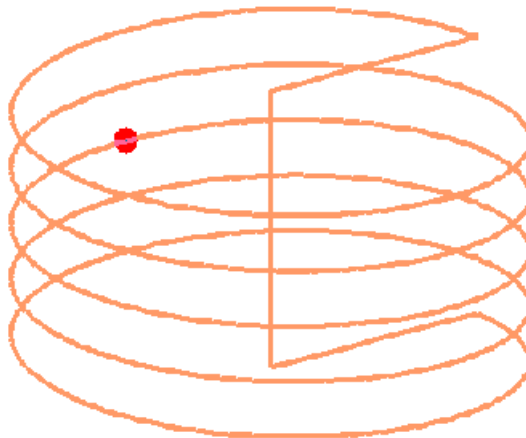
```

plot3(x,y,z,'Color',[1,0.6,0.4],'LineWidth',2.5)
axis off
%
h=line('xdata',x(1),'ydata',y(1),'zdata',z(1),'Color',[1 0 0],'Marker','.',
'MarkerSize',40,'EraseMode','xor'); >
%
n=length(x);i=2;j=1;
while 1
    set(h,'xdata',x(i),'ydata',y(i),'zdata',z(i));
    drawnow; % <22>
    pause(0.0005) % <23>
    i=i+1;
    if nargin==2 & nargout==1
        if(i==ki&j==1);f=getframe(gcf);end % <26>
    end
    if i>n
        i=1;j=j+1;
        if j>K;break;end
    end
end
end

(2)
f=anim_zzy1(2,450);

(3)
image(f.cdata),axis off

```



● 图 5.4-5 红球沿下旋螺线运动的瞬间照片

## 1.5 图形窗功能简介

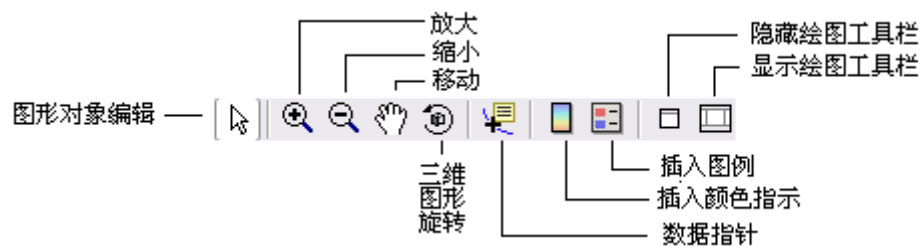


图 5.5-1 图形窗工具条专用按键

【例 5.5-1】利用图形窗的编辑功能，绘制图 5.5-2 所示的连续调制波形  $y = \sin(t) \sin(9t)$  及其包络线。

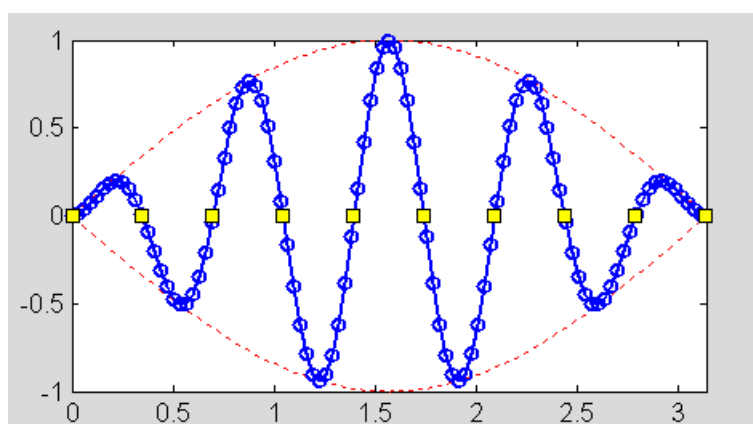


图 5.5-2 个性化的图形

(1)

```
t=(0:pi/100:pi)';
y1=sin(t)*[1,-1];
y2=sin(t).*sin(9*t);
t3=pi*(0:9)/9;
y3=zeros(size(t3));
plot(t,y1,t,y2,t3,y3)
```

(2)

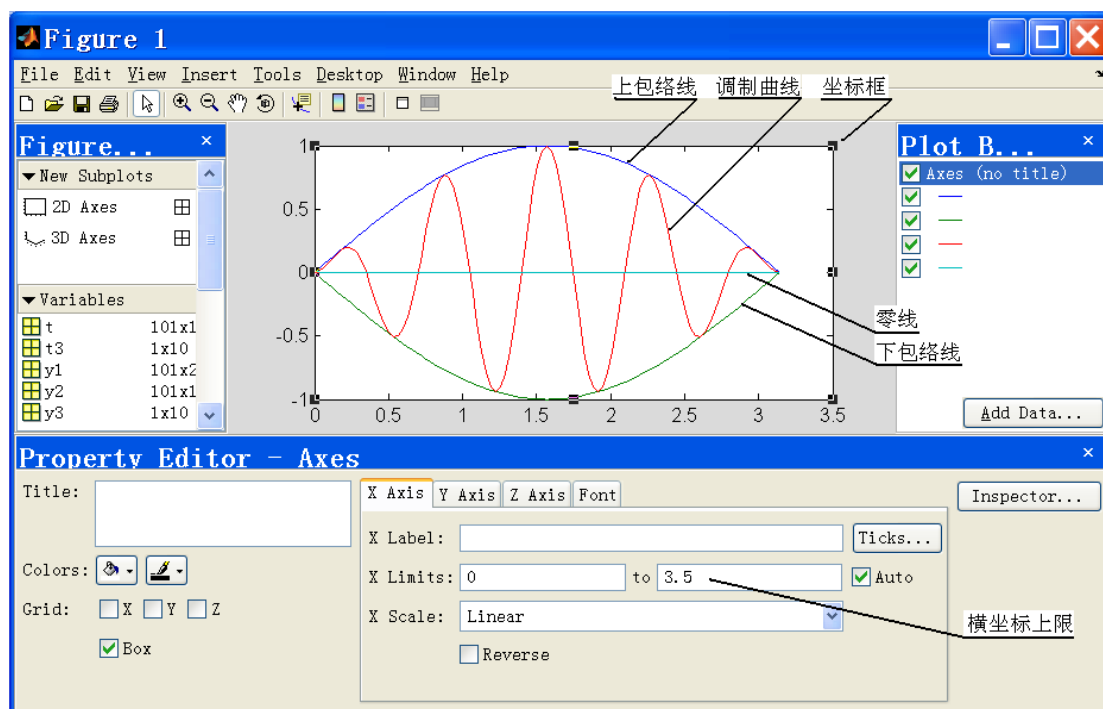


图 5.5-3 编辑工作模式下的图形窗界面

(3)

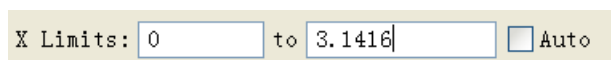


图 5.5-4 横坐标上限设置示例

(4)

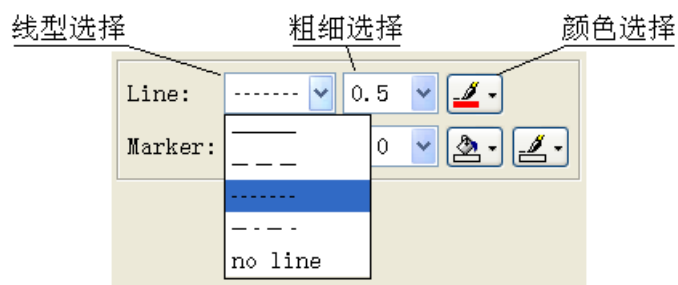


图 5.5-5 包络线的设置示例

(5)

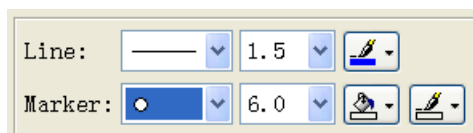


图 5.5-6 调制曲线的设置示例

(6)

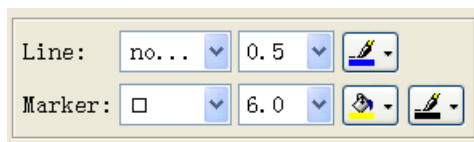


图 5.5-7 调制曲线的设置示例

## 习题 5

1. 已知椭圆的长、短轴  $a = 4, b = 2$ ，用“小红点线”画椭圆  $\begin{cases} x = a \cos t \\ y = b \sin t \end{cases}$ 。（参见图 p5-1）

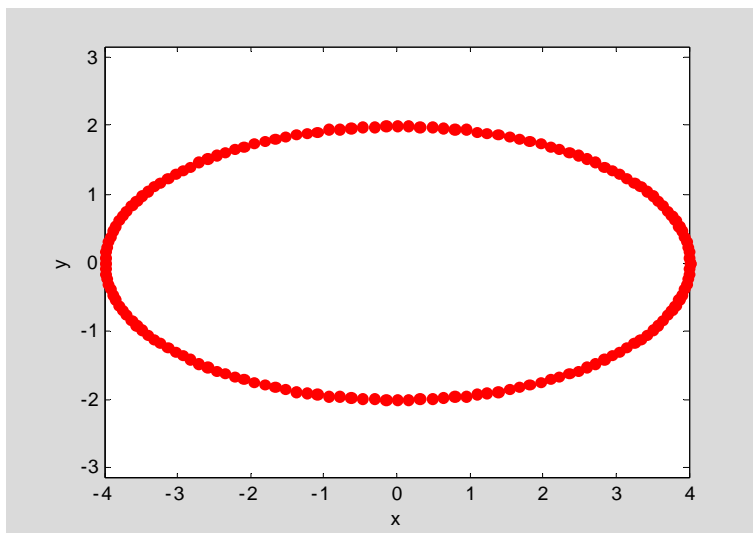


图 p5-1

2. 根据表达式  $\rho = 1 - \cos \theta$  绘制如图 p5-2 的心脏线。（提示：采用极坐标绘线指令 polar）

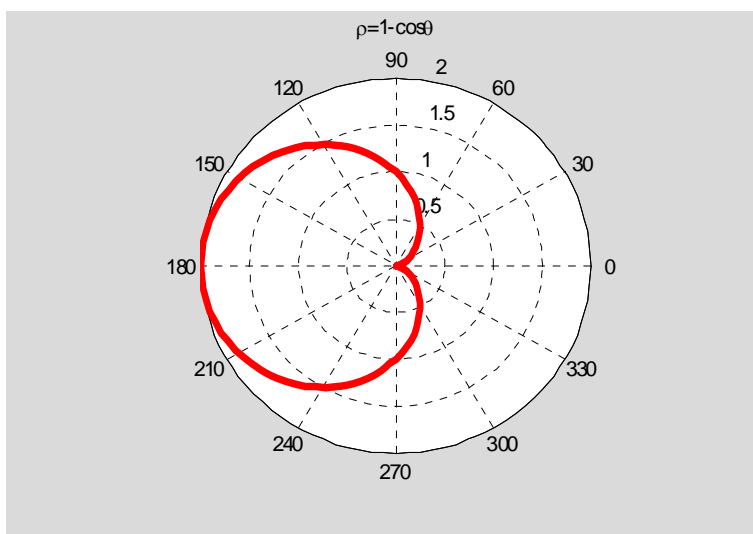


图 p5-2

3. A,B,C 三个城市上半年每个月的国民生产总值见表 p5.1。试画出如图 p5-3 所示的三城市上半年每月生产总值的累计直方图。（提示：使用指令 bar）

表 p5.1 各城市生产总值数据（单位：亿元）

城市	1 月	2 月	3 月	4 月	5 月	6 月
A	170	120	180	200	190	220
B	120	100	110	180	170	180
C	70	50	80	100	95	120

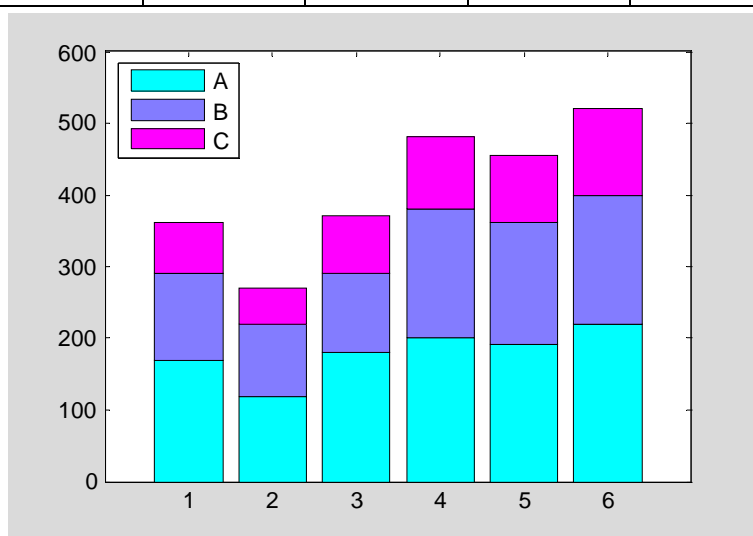


图 p5-3

4. 二阶线性系统的归一化（即令  $\omega_n = 1$ ）冲激响应可表示为：

$$y(t) = \begin{cases} \frac{1}{\beta} e^{-\zeta t} \sin(\beta t) & 0 \leq \zeta < 1 \\ te^{-t} & \zeta = 1 \\ \frac{1}{2\beta} [e^{-(\zeta-\beta)t} - e^{-(\zeta+\beta)t}] & \zeta > 1 \end{cases}, \text{ 其中 } \beta = \sqrt{1-\zeta^2}, \zeta \text{ 为阻尼}$$

系数。试在同一张图上，画出  $\zeta = 0.2:0.2:1.4$  不同取值时的各条曲线，时间区间为

$t \in [0, 18]$ 。此外，要求（1） $\zeta < 1$  的各条曲线用蓝色， $\zeta > 1$  的用红色， $\zeta = 1$  的用黑

色，且线较粗；（2） $\zeta = 0.2$  和  $\zeta = 1.4$  的两条曲线要有醒目的阻尼系数标志。（本题要

求图形参见图 p5-4）

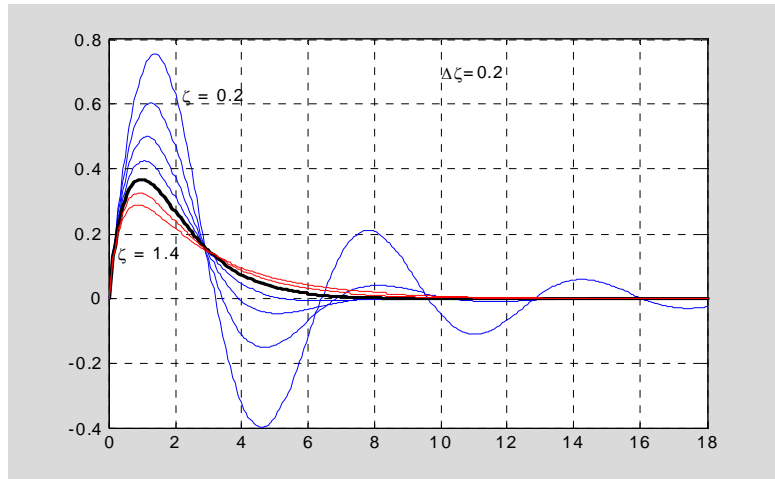


图 p5-4

5. 用绿实线绘制  $x = \sin(t)$ ,  $y = \cos(t)$ ,  $z = t$  的三维曲线, 曲线如图 p5-5 所示。(提示: 使用 `plot3` 指令)

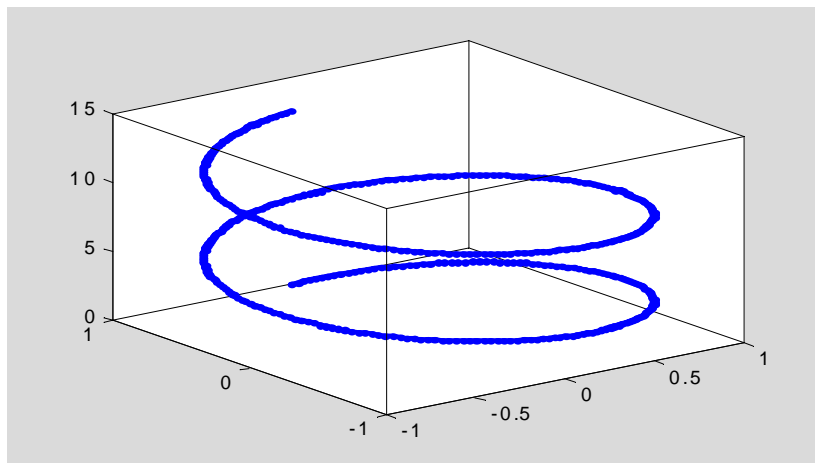


图 p5-5

6. 在区域  $x, y \in [-3, 3]$ , 绘制  $z = 4xe^{-x^2-y^2}$  的如图 p5-6 的三维 (透视) 网格曲面。(不得使用 `ezmesh`)



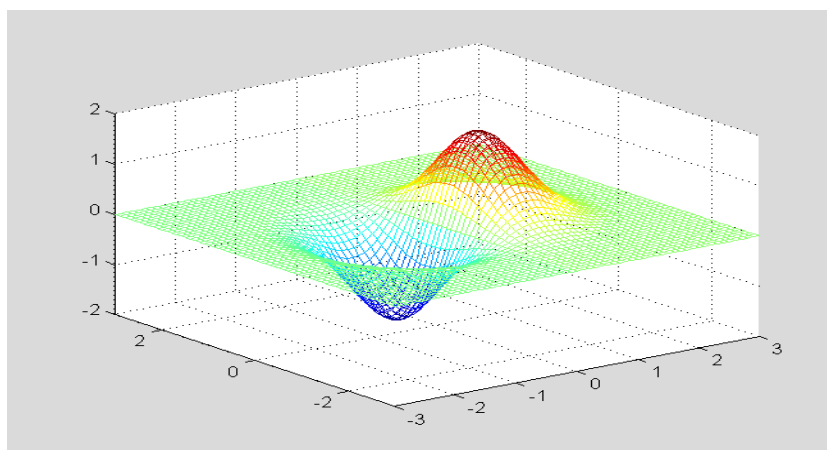


图 p5-6

7. 在  $x, y \in [-4\pi, 4\pi]$  区间里, 根据表达式  $z = \frac{\sin(x+y)}{x+y}$ , 绘制如图 p5-7 所示的曲面。

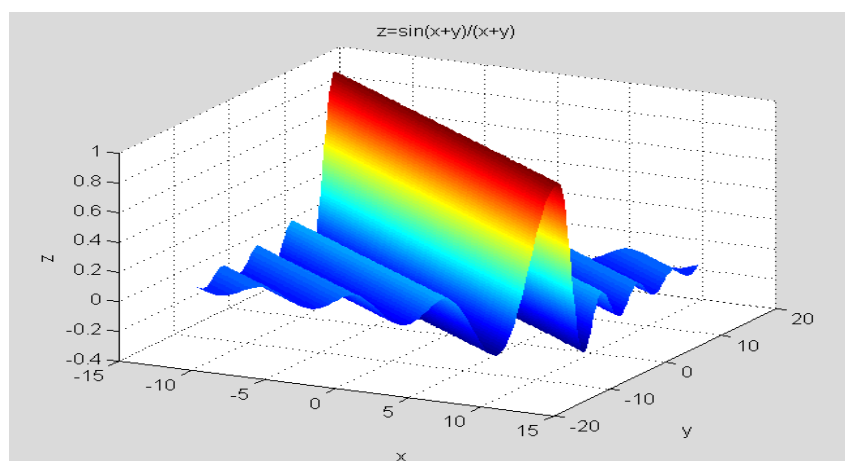


图 p5-7

8. 试用图解法回答:

(1) 方程组 
$$\begin{cases} \frac{y}{(1+x^2+y^2)} = 0.1 \\ \sin(x+\cos(y)) = 0 \end{cases}$$
 有多少个实数解?

- (2) 求出离  $x=0, y=0$  最近、且满足该方程组的一个近似解。

【答案】

- (1) 有 6 个

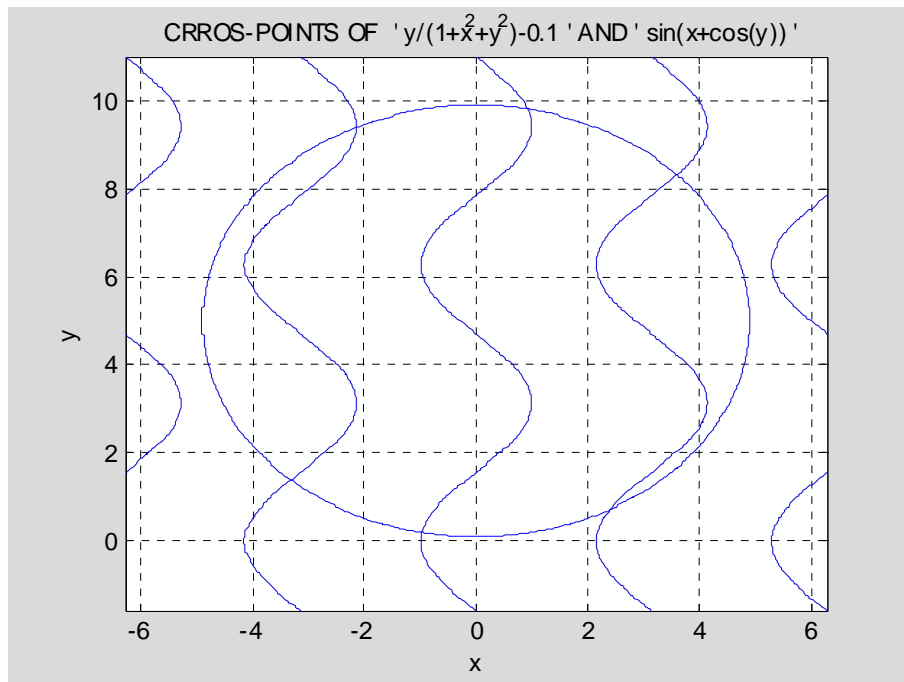


图 p5-8

(2) 离  $x = 0, y = 0$  最近的一个近似解为

```
x00 =  
-0.9801  
y00 =  
0.2005
```

9. 制作如文件 prob509.p(在光盘的 mfiles 文件夹上)运行时那样的色图变幻(参见图 p5-9)。

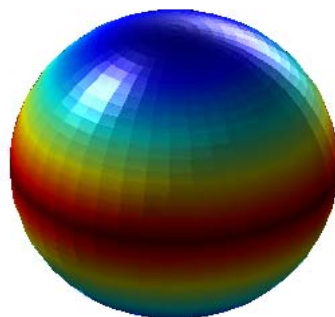


图 p5-9

10. 在  $[0, 4\pi]$  区间内, 根据  $y(t, x) = e^{-0.2x} \sin(\frac{\pi}{24}t - x)$  通过图形曲线表现“行波”。图 p5-9 显示的是该动画中的一个静止图形。做题前, 请先从 prob\_data510.mat 文件 (在光盘的 mfiles 文件夹上) 中获取数据 MOV, 然后运行 shg,axis off,movie(MOV,5,12) 观察演示。

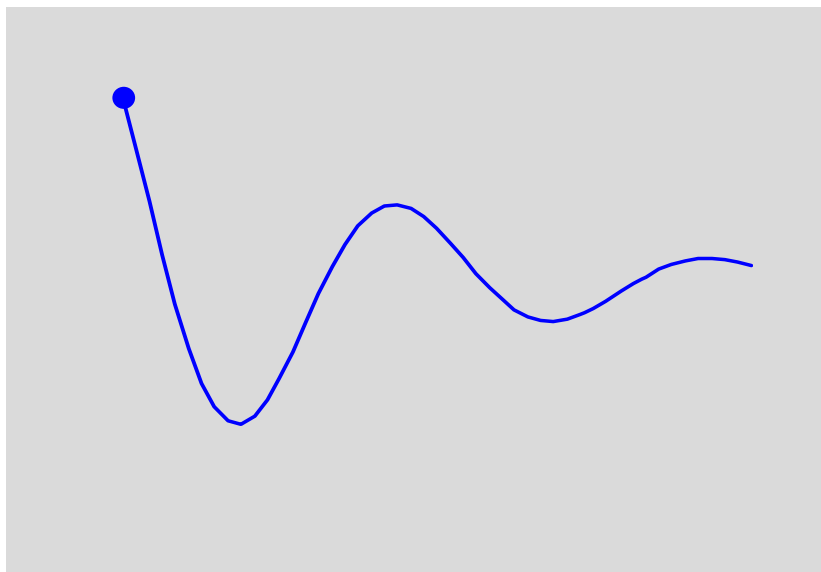


图 p5-10

11. 编写使红色小球沿三叶线  $\rho = \cos(3\theta)$  运动的程序。具体参见演示程序 prob511.p (在光盘的 mfiles 文件夹上) 的运行实况。图 p5-11 显示的是该动画中的一个静止图形。

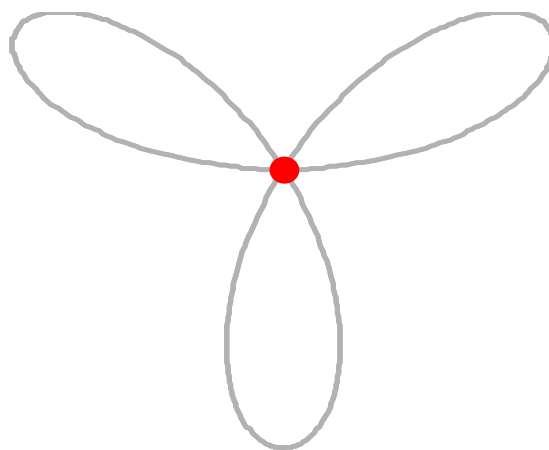


图 p5-11

## 第 1 章 M文件和函数句柄

从总体上说, 单独的 MATLAB 指令, 及由若干指令构成的孤立表达式, 只能解算或仿

真简单理论数学模型所描述的问题。复杂些的问题通常需要用一组理论数学算式。这些问题的解算及仿真就需要借助较复杂的 MATLAB 程序进行。这种 MATLAB 程序由较多的 MATLAB 指令和多种多样的 MATLAB 表达式组成，并循着一定的执行次序运行。这种程序的扩展名为 m。这就是脚本 M 文件或函数 M 文件。

本章系统介绍编写 MATLAB 程序时最常用到的四种控制结构、M 函数文件的构造、主函数、子函数以及匿名函数。最后一节专门叙述函数句柄的创建、观察和调用。

## 1.1 MATLAB 控制流

作为一种计算机编程语言，MATLAB 提供了多种用于程序流控制的描述关键词 (Keyword)。本节只介绍其中最常用的条件控制 (if, switch) 和循环控制 (for, while, continue, break)。由于 MATLAB 的这些指令与其它语言相应指令的用法十分相似，因此本节只结合 MATLAB 给定的描述关键词，对这四种指令进行简要的说明。

### 1.1.1 if-else-end 条件控制

if-else-end 指令为程序流提供了一种分支控制，它最常见的使用方式见表 6.1-1。

表 6.1-1 if-else-end 分支结构的使用方式

单 分 支	双 分 支	多 分 支
<b>if</b> <b>expr</b> <b>(commands)</b> <b>end</b>	<b>if</b> <b>expr</b> <b>(commands1)</b> <b>else</b> <b>(commands2)</b> <b>end</b>	<b>if</b> <b>expr1</b> <b>(commands)</b> <b>elseif</b> <b>expr2</b> <b>(commands)</b> ..... <b>else</b> <b>(commandsk)</b> <b>end</b>
当 <b>expr</b> 给出“逻辑 1”时， <b>(commands)</b> 指令组才被执行。	当 <b>expr</b> 给出“逻辑 1”时， <b>(commands1)</b> 指令组被执行；否则， <b>(commands2)</b> 被执行。	<b>expr1, expr2, ...</b> 中，首先给出“逻辑 1”的那个分支的指令组被执行；否则， <b>(commandsk)</b> 被执行。 该使用方法常被 switch-case 所取代。

【例 6.1-1】已知函数  $y = \begin{cases} x & x < -1 \\ x^3 & -1 \leq x < 1 \\ e^{-x+1} & 1 \leq x \end{cases}$ ，编写能对任意一组输入 x 值求相应 y 值的

程序。

(1)

**function** y=exm060101(x)

**% y=exm060101(x)      Function calculate of example 6.1-1**

**n=length(x);**

```

for k=1:n
    if x(k)<-1
        y(k)=x(k);
    elseif x(k)>=1
        y(k)=exp(1-x(k));
    else
        y(k)=x(k)^3;
    end
end

```

(2)

(3)

```
x=[-2,-1.2,-0.4,0.8,1,6]
```

```
y=exm060101(x)
```

```
x =
```

```
-2.0000   -1.2000   -0.4000    0.8000    1.0000    6.0000
```

```
y =
```

```
-2.0000   -1.2000   -0.0640    0.5120    1.0000    0.0067
```

## 1.1.2 switch-case控制结构

表 6.1-2 switch-case 控制结构的使用方式

指令格式	含义
<b>switch</b> <b>expr</b>  <b>case</b> <b>value_1</b>  ( <b>commands1</b> )  <b>case</b> <b>value_2</b>  ( <b>commands2</b> )  <b>case</b> <b>value_k</b>  ( <b>commandsk</b> )  <b>otherwise</b>	<ul style="list-style-type: none"> <li>● <b>expr</b> 为根据此前给定变量进行计算的表达式</li> <li>● <b>value_1</b> 是给定的数值、字符串标量（或单元数组）</li> <li>● 若 <b>expr</b> 结果与 <b>value_1</b>（或其中的单元元素）相等，则执行。</li> <li>● <b>value_k</b> 是给定的数值、字符串标量（或单元数组）</li> <li>● 若 <b>expr</b> 结果与 <b>value_k</b>（或其中的单元元素）相等，则执行。</li> <li>● 该情况是以上的“并”的“补”</li> <li>● 若所有 <b>case</b> 都不发生，则执行该组命令。</li> </ul>

<p>(commands)</p> <p>end</p>	
------------------------------	--

【例 6.1-2】已知学生的名字和百分制分数。要求根据学生的百分制分数，分别采用“满分”、“优秀”、“良好”、“及格”和“不及格”等表示学生的学习成绩。

```
clear;
for k=1:10
    a(k)={89+k};b(k)={79+k};c(k)={69+k};d(k)={59+k};
end;
c=[d,c];
%
A=cell(3,5);
A(1,:)={'Jack','Marry','Peter','Rose','Tom'};
A(2,:)={72,83,56,94,100};
%
for k=1:5
    switch A{2,k}
    case 100
        r='满分';
    case a
        r='优秀';
    case b
        r='良好';
    case c
        r='及格';
    otherwise
        r='不及格';
    end
    A(3,k)={r};
end
A
A =
    'Jack'    'Marry'    'Peter'    'Rose'    'Tom'
    [ 72]    [ 83]    [ 56]    [ 94]    [ 100]
    '及格'    '良好'    '不及格'    '优秀'    '满分'
```

### 1.1.3 for循环和while循环

表 6.1-3 循环结构的使用方式

for 循环	while 循环
<b>for</b> ix=array  (commands)  <b>end</b>	<b>while</b> expression  (commands)  <b>end</b>

【例 6.1-3】创建 Hilbert 矩阵。

(1) Hilbert 矩阵  $a(i, j) = \frac{1}{i + j - 1}$ 。

(2)

```
K=5;
A = zeros(K,K) ;
for m = 1:K
    for n = 1:K
        A(m,n) = 1/(m+n -1);
    end
end
format rat
A
format short g
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
1/5	1/6	1/7	1/8	1/9

(3)

```
clear
tic
K=1000;
for m = 1:K
    for n = 1:K
        A1(m,n) = 1/(m+n -1);
    end
end
t1=toc
t1 =
```

17.434

```
tic
K=1000;
```

```

A2 = zeros(K,K);
for m = 1:K
    for n = 1:K
        A2(m,n) = 1/(m+n -1);
    end
end
t2=toc
t2 =
    2.1895

```

```

(4)
tic
N=1000;
n=repmat(1:N,N,1);
m=n';
A3=1./(n+m-1);
t3=toc
t3 =
    0.13909

```

【例 6.1-4】编写计算  $S = \sum_{n=1}^N \frac{1}{\sum_{k=1}^n k}$ ，其中  $N = \arg \min \left\{ \frac{1}{\sum_{k=1}^N k} \leq \varepsilon \right\}$ ， $\varepsilon$  是预先给定的控制

精度。

(1)

```

function [S,N]=exm060104(epsilon)
% [S,N]=exm060104(epsilon)
%          Calculate the sum of a special series S=1+1/(1+2)+...+1/(1+2+...+N)
% S          Sum of a special series
% N          The minimum among all numbers to have 1/sum(1:N)<epsilon
% epsilon    Given accuracy

k=0;
s=0;
d=inf;
S=0;
while d>epsilon
    k=k+1;
    s=s+k;
    d=1/s;

```



```

        S=S+d;
end
N=k;

(2)
(3)
[S,N]=exm060104(0.0001)
S =
    1.9859
N =
    141

syms k n N
SINF=limit(symsum(1/symsum(k,1,n),n,1,N),N,inf)
SINF =
    2

N=141;
SN=vpa(symsum(1/symsum(k,1,n),n,1,N))
SN =
    1.9859154929577464788732394366197

```

【例 6.1-5】创建  $n$  阶魔方矩阵，限定条件是  $n$  为能被 4 整除的偶数。

(1) 所谓魔方矩阵 (Magic matrix)，是指那矩阵由 1 到  $n^2$  的正整数按照一定规则排列而成，

并且每列、每行、每条对角线元素的和都等于  $\frac{n(n^2+1)}{2}$ 。就生成规则而言，魔方矩阵可分

成三类：一， $n$  为奇数；二， $n$  为不能被 4 整除的偶数；三， $n$  为能被 4 整除的偶数。

(2)

```

% exm060105.m      生成一类魔方矩阵，该魔方矩阵的阶 n 为能被 4 整除的偶数。

% A              为魔方矩阵

% n              魔方矩阵的阶数

clear
clc
while 1
    %<6>

    n=input('请输入一个能被 4 整除的正整数！ n = ');

```

```

    if mod(n,4)==0
        %<8>
        break
    %<9>
    end
    %<10>
end
    %<11>
G=logical(eye(4,4)+rot90(eye(4,4)));
m=n/4;
K=repmat(G,m,m);
N=n^2;
A=reshape(1:N,n,n);
A(K)=N-A(K)+1

(3)

(4)

(5)
s0=round(n*(n*n+1)/2);
disp([int2str(n),' 阶魔方矩阵的标称和是 ',int2str(s0)])
Ns0=round(2*(n+1));
B=A';
SC=sum(A);
SR=sum(B);
Sd=sum(diag(A));
Sdi=sum(diag(B));
LS=[SC,SR,Sd,Sdi]==s0;
NS=round(sum(LS));
if NS==Ns0
    disp('经验证, A 是魔方矩阵。')
else
    disp('经验证, A 不是魔方矩阵。')
end
12 阶魔方矩阵的标称和是    870
经验证, A 是魔方矩阵。

```

### 1.1.4 控制程序流的其它常用指令

表 6.1-4 控制程序流的其它常用指令

指令及使用格式	使 用 说 明
---------	---------

<b>v=input('message')</b> <b>v=input('message','s')</b>	该指令执行时，“控制权”交给键盘；待输入结束，按下 Enter 键，“控制权”交还 MATLAB。message 是提示用的字符串。
<b>keyboard</b>	遇到 keyboard 时，将“控制权”交给键盘，用户可以从键盘输入各种 MATLAB 指令。
<b>break</b>	break 指令，或导致包含该指令的 while、for 循环终止，或在 if - end、switch - case、try - catch 中导致中断。参见例 6.1-5。
<b>continue</b>	跳过位于它之后的循环体中其他指令，而执行循环的下一个迭代。见例 6.1-5。
<b>pause</b> <b>pause(n)</b>	第一种格式使程序暂停执行，等待用户按任意键继续； 第二种格式使程序暂停 n 秒后，再继续执行。
<b>return</b>	结束 return 指令所在函数的执行，而把控制转至主调函数或者指令窗。否则，只有待整个被调函数执行完后，才会转出。

## 1.2 脚本文件和函数文件

### 1.2.1 M脚本文件

对于一些比较简单的问题，从指令窗中直接输入指令进行计算是十分轻松简单的事。但随指令数的增加，或随控制流复杂度的增加，或重复计算要求的提出，直接从指令窗进行计算就显得烦琐。而此时脚本文件最为适宜。“脚本”本身反映这样一个事实：MATLAB 只是按文件所写的指令执行。关于 M 脚本文件的编写，请参见 1.8 节。

这种文件的构成比较简单。其特点是：

- 它只是一串按用户意图排列而成的（包括控制流向指令在内的）MATLAB 指令集合。
- 脚本文件运行后，产生的所有变量都驻留在 MATLAB 基本工作空间（Base workspace）中。只要用户不使用 clear 指令加以清除，只要 MATLAB 指令窗不关闭，这些变量将一直保存在基本工作空间中。基本空间随 MATLAB 的启动而产生；只有当关闭 MATLAB 时，该基本空间才被删除。

### 1.2.2 M函数文件

与脚本文件不同，函数文件（Function file）犹如一个“黑箱”。从外界只看到：传给它的输入量和送出来的计算结果。而内部运作是藏而不见的。它的特点是：

### 1.2.3 局部变量和全局变量

#### （1）局部（Local）变量

存在于函数空间内部的中间变量，产生于该函数的运行过程中，其影响范围也仅限于该函数本身。正由于这种空间、时间上的局部性，中间变量被称为局部变量。

#### （2）全局（Global）变量

通过 global 指令，MATLAB 也允许几个不同的函数空间以及基本工作空间共享同一个变量。这种被共享的变量称为全局变量。

### 1.2.4 M函数文件的一般结构

由于从结构上看，脚本文件只是比函数文件少一个“函数申明行”，所以只需描绘清楚函数文件的结构，脚本文件的结构也就无须多费笔墨了。

【例 6.2-1】编写一个 M 函数文件。它具有以下功能：（A）根据指定的半径，画出蓝色圆周线；（B）可以通过输入字符串，改变圆周线的颜色、线型；（C）假若需要输出圆面积，则绘出圆。

（1）

```
function [S,L]=exm060201(N,R,str)
```

```
% exm060201.m      The area and perimeter of a regular polygon (正多边形的面积和周长)
```

```
%           N           The number of sides
```

```
%           R           The circumradius
```

```
%           str         A line specification to determine line type/color
```

```
%           S           The area of the regular polygon
```

```
%           L           The perimeter of the regular polygon
```

```
% exm060201          用蓝实线画半径为 1 的圆
```

```
% exm060201(N)       用蓝实线画外接半径为 1 的正 N 边形
```

```
% exm060201(N,R)     用蓝实线画外接半径为 R 的正 N 边形
```

```
% exm060201(N,R,str) 用 str 指定的线画外接半径为 R 的正 N 边形
```

```
% S=exm060201(...)   给出多边形面积 S ,并画相应正多边形填色图
```

```
% [S,L]=exm060201(...) 给出多边形面积 S 和周长L,并画相应正多边形填色图
```

```
% Zhang Zhiyong 编写于 2006-1-31
```

```
switch nargin
```

```
    case 0
```

```
        N=100;R=1;str='-b';
```

```
    case 1
```

```
        R=1;str='-b';
```

```
    case 2
```

```
        str='-b';
```

```
    case 3
```

```
        ;
```

```
    otherwise
```

```
        error('输入量太多。');
```

```

end;
t=0:2*pi/N:2*pi;
x=sin(t);y=cos(t);
if nargout==0
    plot(x,y,str);
elseif nargout>2

    error('输出量太多。');

else
    S=N*R*R*sin(2*pi/N)/2;
    L=2*N*R*sin(pi/N);
    fill(x,y,str)
end
axis equal square
box on
shg

(2)
[S,L]=exm060201(6,2,'-g')
S =
    10.3923
L =
    12.0000

```

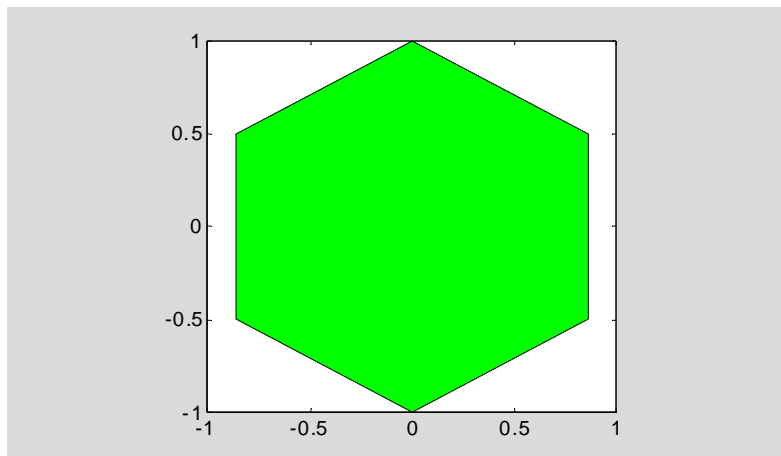


图 6.2-1 绿色正六边形

## 1.3 MATLAB的函数类别

从扩展名M观察,MATLAB的M文件分为M脚本文件和M函数文件。那么,在MATLAB中,函数 Function 又被细分为:主函数;子函数;嵌套函数;私用函数;匿名函数等。限于篇幅,本节只对主函数、子函数及匿名函数进行阐述。

### 1.3.1 主函数 (Primary function)

### 1.3.2 子函数 (Subfunction)

【例 6.3-1】编写一个内含子函数的 M 函数绘图文件。

(1)

```
function Hr=exm060301(flag )
% exm060301.m      Demo for handles of primary functions and subfunctions
%
%      flag      可以取字符串 'line' 或 'circle' 。
%
%      Hr      子函数cirline的句柄

t=(0:50)/50*2*pi;
x=sin(t);
y=cos(t);
Hr=@cirline;
feval(Hr,flag,x,y,t)
% -----subfunction-----
function cirline(wd,x,y,t)
% cirline(wd,x,y,t) 是位于exm060301.m 函数体内的子函数
%
%      wd      接受字符串 'line' 或 'circle'
%
%      t      画线用的独立参变量
%
%      x      由 t 产生的横坐标变量
%
%      y      由 t 产生的纵坐标变量

switch wd
case 'line'
    plot(t, x, 'b', t, y, 'r', 'LineWidth', 2)
case 'circle'
    plot(x, y, '-g', 'LineWidth', 8),
    axis square off
otherwise
    error('输入宗量只能取 "line" 或 "circle" ！ ')
end
shg
```

```
(2)
HH=exm060301('circle')
HH =
    @cirline
```

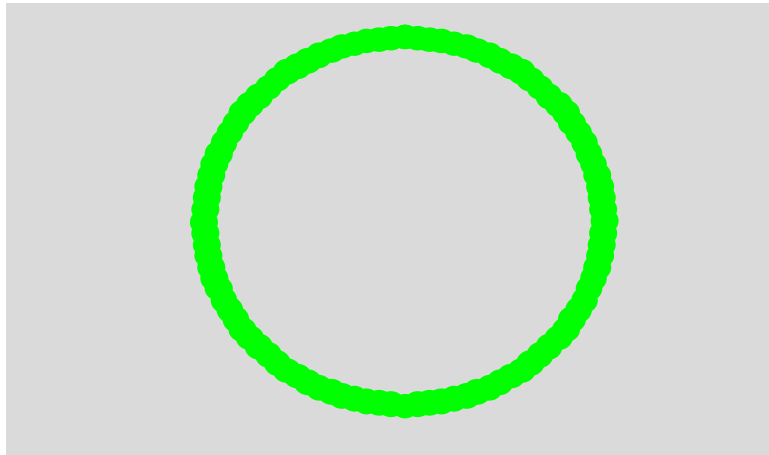


图 6.3-1 绿色圆周线

```
(3)
t=0:2*pi/5:2*pi;x=cos(t);y=sin(t);%为绘制正五边形准备数据
HH('circle',x,y,t)                %利用句柄绘图
```

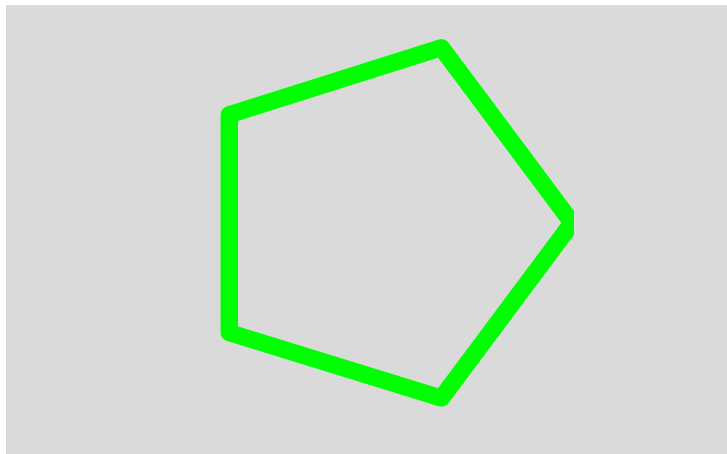


图 6.3-2 由子函数绘制的绿色正五边形

### 1.3.3 匿名函数（Anonymous function）

(1)

**FH= @(arglist) expr**

(2)

**FH(arglist)**                      直接调用格式

**feval(FH, arglist)**

间接调用格式

## 1.4 函数句柄

函数句柄（Function handle）是 MATLAB 的一种数据类型。它携带着“相应函数创建句柄时的路径、视野、函数名，以及可能存在的重载方法”。

引入函数句柄的理由是：使 feval 及借助于它的泛函指令工作更可靠；使“函数调用”象“变量调用”一样灵活方便；提高函数调用速度，特别在反复调用情况下更显效率；提高软件重用性，扩大子函数和私用函数的可调用范围；可迅速获得同名重载函数的位置、类型信息。

### 1.4.1 函数句柄的创建和观察

【例 6.4-1】为 MATLAB 的 magic 函数创建函数句柄，并观察其内涵。

(1)

```
hm=@magic
hm =
    @magic
```

(2)

```
class(hm)
isa(hm, 'function_handle')
ans =
function_handle
ans =
    1
```

(3)

```
CC=functions(hm)
CC =
    function: 'magic'
           type: 'simple'
           file: 'D:\MATLAB71\toolbox\matlab\elmat\magic.m'
```

(4)

```
M1=hm(4)
M1 =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```



(5)

```
M2=feval(hm,4)
```

```
M2 =
```

```
16     2     3    13
 5    11    10     8
 9     7     6    12
 4    14    15     1
```

## 1.4.2 函数句柄的基本用法

在第 6.3-3 节和第 6.4-1 节中，已经简单介绍和演示了函数句柄两种调用格式的简单形式。本节将更为准确地介绍两种函数句柄调用格式的一般形式。

**[argout1, argout2, ..., argoutn]=FunName(argin1, argin2, ..., arginn)**

**[argout1, argout2, ..., argoutn]=Hfun(argin1, argin2, ..., arginn)**

直接调用格式

**[argout1, argout2, ..., argoutn]=feval(Hfun, argin1, argin2, ..., arginn)**

间接调用格式

【例 6.4-2】借助本书算例 6.2-1 建立的函数文件 exm060201.m。

(1)

(2)

```
[S,L]=exm060201(3,2,'-r')
```

```
S =
```

```
5.1962
```

```
L =
```

```
10.3923
```

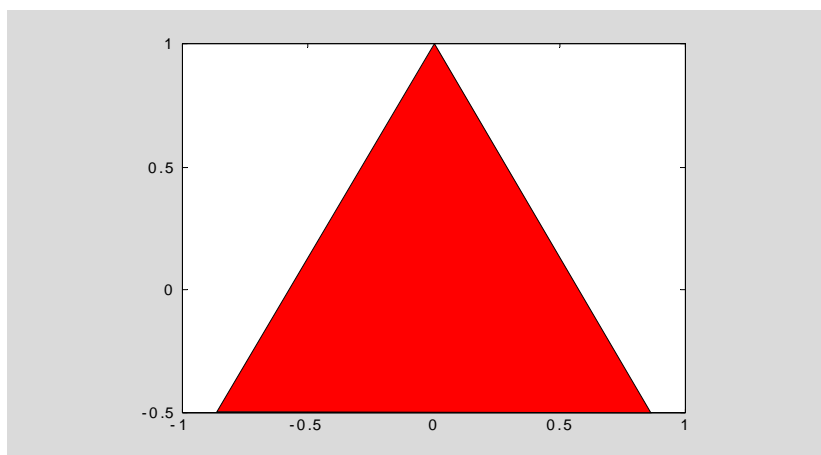


图 6.4-1 直接调用函数 exm060201 所画的红色等边三角形

(3)

```
Hexm=@exm060201,
```

```
Hexm =  
    @exm060201
```

(4)

```
which('exm060201')  
'exm060201' not found.
```

(5)

```
[S,L]=exm060201(3,2,'-r')  
??? Undefined command/function 'exm060201'.
```

(6)

```
[S,L]=Hexm(3,2,'-r')  
S =  
    5.1962  
L =  
   10.3923
```

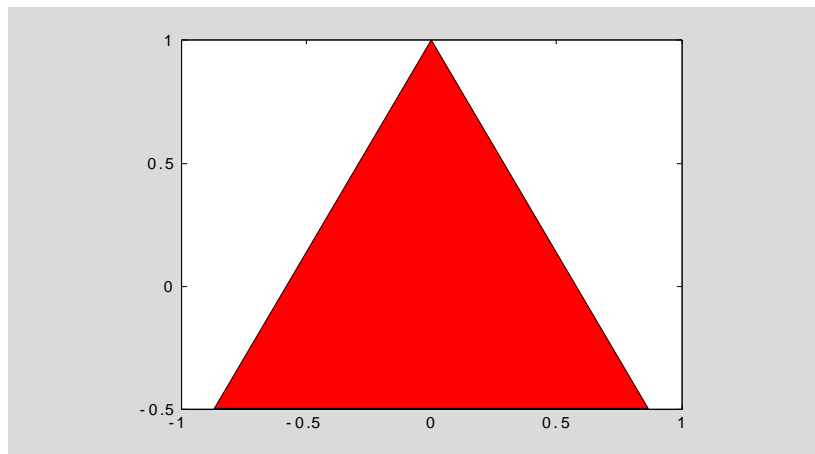


图 6.4-2 在新视野中里用函数句柄 Hexm 所画的红色等边三角形

【例 6.4-3】借助本书算例 6.2-1 建立的函数文件 exm060201.m。

(1)

```
clear
```

(2)

```
Hexm_2=@exm060201  
Hexm_2 =  
    @exm060201
```

(3)

```
class(Hexm_2)  
isa(Hexm_2,'function_handle')
```

```
ans =
function_handle
ans =
1
```

(4)

```
[S,L]=Hexm_2(3,2,'-r')
??? Undefined command/function 'exm060201'.
```

## 习题 6

1. 请分别写出用 for 和 while 循环语句计算  $K = \sum_{i=0}^{1000000} 0.2^i = 1 + 0.2 + 0.2^2 \cdots + 0.2^{1000000}$  的

程序。此外，还请写出避免循环的数值、符号计算程序。

```
s1 =
1.2500
```

2. 编写一个函数 M 文件，它的功能：没有输入量时，画出单位圆（见图 p6-1）；输入量是大于 2 的自然数 N 时，绘制正 N 边形，图名应反映显示多边形的真实边数（见图 p6-2）；输入量是“非自然数”时，给出“出错提示”。此外，函数 M 文件应有 H1 行、帮助说明和程序编写人姓名。

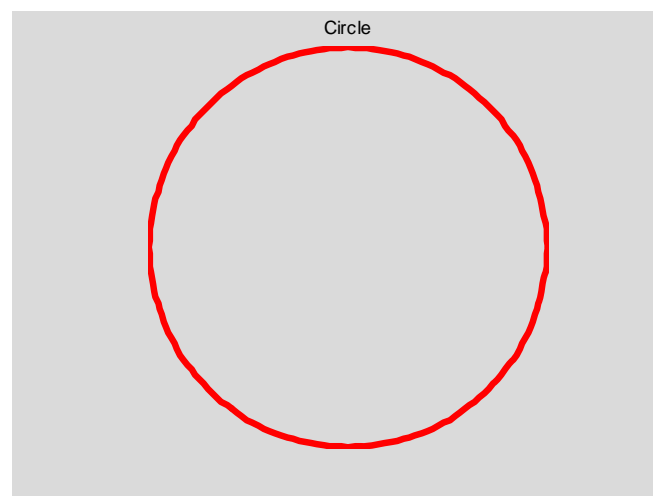


图 p6-1

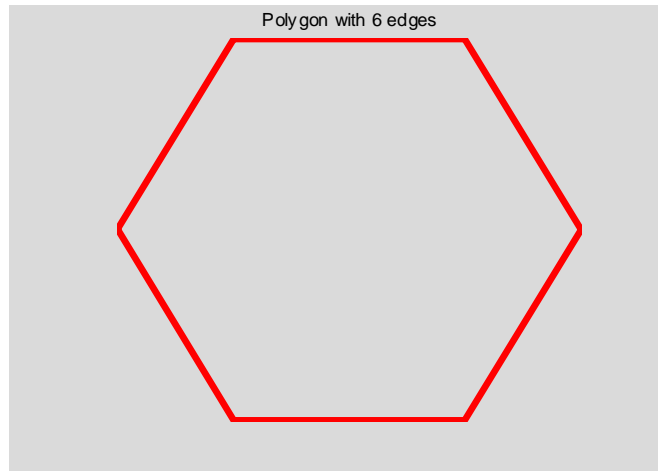


图 p6-2

```
function prob_solve602(n)
% prob_solve602(n)      plot a circle or a polygon with n edges
% prob_solve602        plot a circle
% n                    应为大于2的自然数

% By ZZY, 2006-2-15
if nargin==0
    t=0:pi/100:2*pi;
    x=exp(i*t);
    str='Circle';
else
    if (nargin~=0)&(n<=2)
        error('输入量应是大于2的自然数')
    end;
    if n-round(n)~=0          %检查非自然数
        error('输入量应是大于2的自然数')
    end;
    t=(0:n)/n*2*pi;
    x=exp(i*t);
    str=['Polygon with ', int2str(n), ' edges'];    % 合成字符串
end
plot(real(x),imag(x),'r','LineWidth',4)
title(str)
axis square image off
shg
```

3. 用泛函指令 `fminbnd` 求  $y(x) = -e^{-x} |\sin[\cos x]|$  在  $x=0$  附近的极小值。`fminbnd` 的第一个输入量要求使用匿名函数表达。

**x =**

-0.8634

4. 在 matlab 的 \toolbox\matlab\elmat\private 文件夹上有一个“烟圈矩阵”发生函数 smoke.m。运行指令 smoke(3,0,'double')，将生成一个 3 阶伪特征根矩阵如下

$$A = \begin{bmatrix} -0.5000 + 0.8660i & 1.0000 & 0 \\ 0 & -0.5000 - 0.8660i & 0 \\ 1.0000 & 1.0000 & 1.0000 \end{bmatrix}$$

现在的问题是：在 MATLAB 当前目录为\work 情况下，如何利用函数句柄调用 smoke.m 函数，产生 3 阶伪特征根矩阵。请写出相应的程序或操作步骤。

```
clear
b=pwd                                %获取当前目录名字符串
which('smoke')                       %检查在当前目录下能否看到 smoke.m
b_d=b;
b_d(end-4:end)=[];                  %在 b 字符串中去除最后的四个字符，即\work。
str=[b_d,'\toolbox\matlab\elmat\private'];
cd(str)                              %把 smoke.m 所在目录设置为当前目录。
SM=@smoke;                          %创建 smoke.m 的函数句柄
cd(b)                                %使 work 恢复为当前目录
disp(' ')
A=SM(3,0,'double')                  %产生一个 3 阶“伪特征根”矩阵
disp(' ')
pwd                                  %显示当前所在目录，以证实符合题意。
which('smoke')                       %再次检查在当前目录下能否看到 smoke.m。
disp(' ')
```

## 第 1 章 SIMULINK交互式 仿真集成环境

SIMULINK 是 MATLAB 最重要的组件之一，它向用户提供一个动态系统建模、仿真和综合分析的集成环境。在这环境中，用户无须书写大量的程序，而只需通过简单直观的鼠标操作，选取适当的库模块，就可构造出复杂的仿真模型。SIMULINK 的主要优点：

- 适应面广。可构造的系统包括：线性、非线性系统；离散、连续及混合系统；单任务、多任务离散事件系统。
- 结构和流程清晰。它外表以方块图形式呈现，采用分层结构。既适于自上而下的设计流

程，又适于自下而上逆程设计。

- 仿真更为精细。它提供的许多模块更接近实际，为用户摆脱理想化假设的无奈开辟了途径。
- 模型内码更容易向 DPS，FPGA 等硬件移植。

基于本书定位，为避免内容空泛，本节对于 SIMULINK 将不采用横断分层描述，即不对 SIMULINK 库、模块、信号线勾画标识等进行分节阐述。本节将以四个典型算例为准线，纵向描述 SIMULINK 的使用要领。

## 1.1 连续时间系统的建模与仿真

### 1.1.1 基于微分方程的SIMULINK建模

本节将从微分方程出发，以算例形式详细讲述 SIMULINK 模型的创建和运行。

【例 7.1-1】在图 7.1-1 所示的系统中，已知质量  $m = 1 \text{ kg}$ ，阻尼  $b = 2 \text{ N}\cdot\text{sec}/\text{m}$ ，弹簧系数  $k = 100 \text{ N}/\text{m}$ ，且质量块的初始位移  $x(0) = 0.05 \text{ m}$ ，其初始速度  $x'(0) = 0 \text{ m}/\text{sec}$ ，要求创建该系统的 SIMULINK 模型，并进行仿真运行。

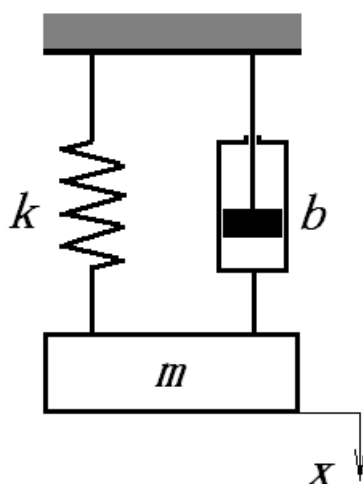


图 7.1-1 弹簧—质量—阻尼系统

#### (1) 建立理论数学模型

对于无外力作用的“弹簧—质量—阻尼”系统，据牛顿定律可写出

$$mx'' + bx' + kx = 0 \quad (7.1)$$

代入具体数值并整理，可得

$$x'' = -2x' - 100x \quad (7.2)$$

#### (2) 建模的基本思路

#### (3)

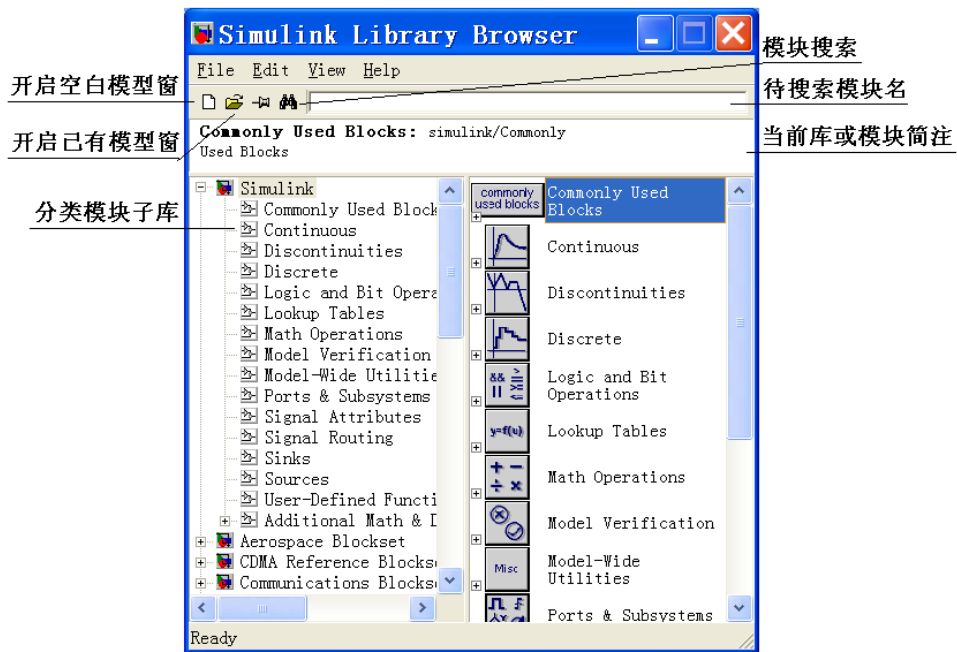


图 7.1-2

(4) 开启空白（新建）模型窗

(5)

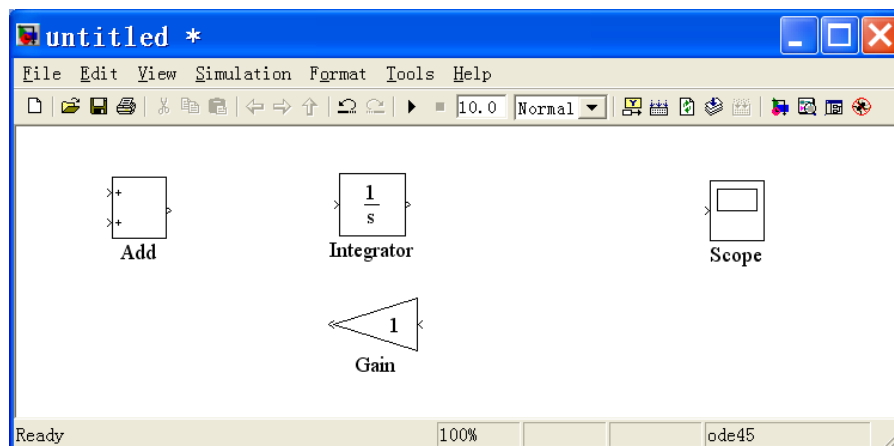


图 7.1-3

(6)

(7)

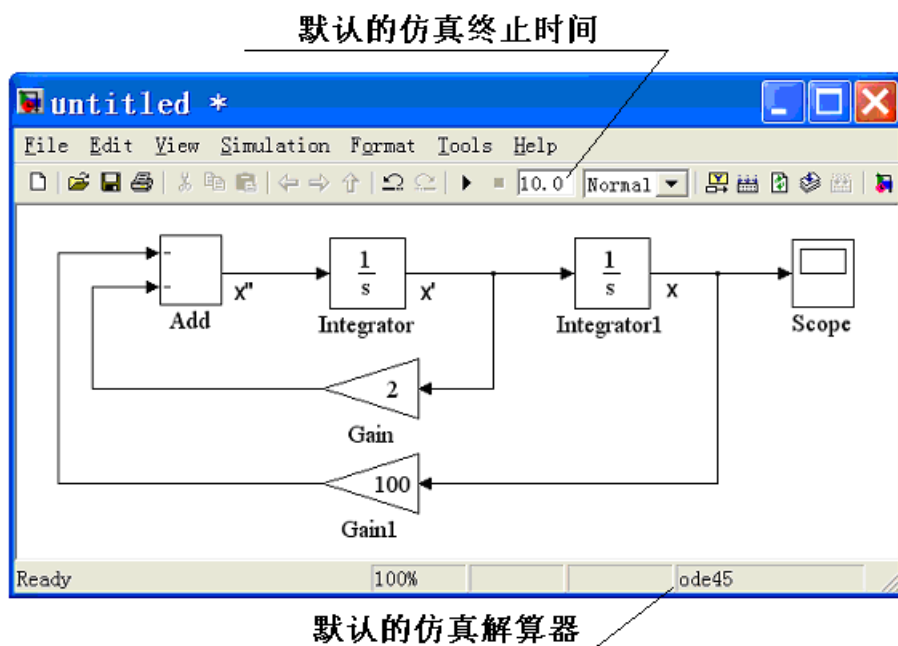


图 7.1-4

(8)

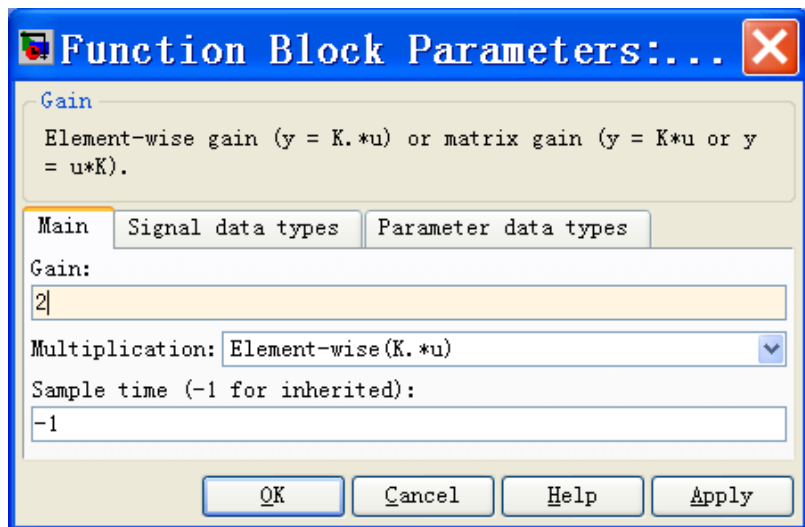


图 7.1-5



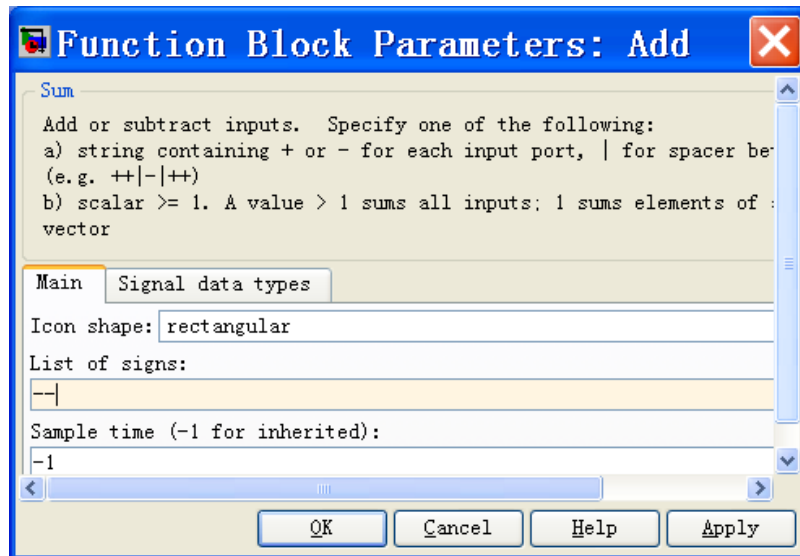


图 7.1-6

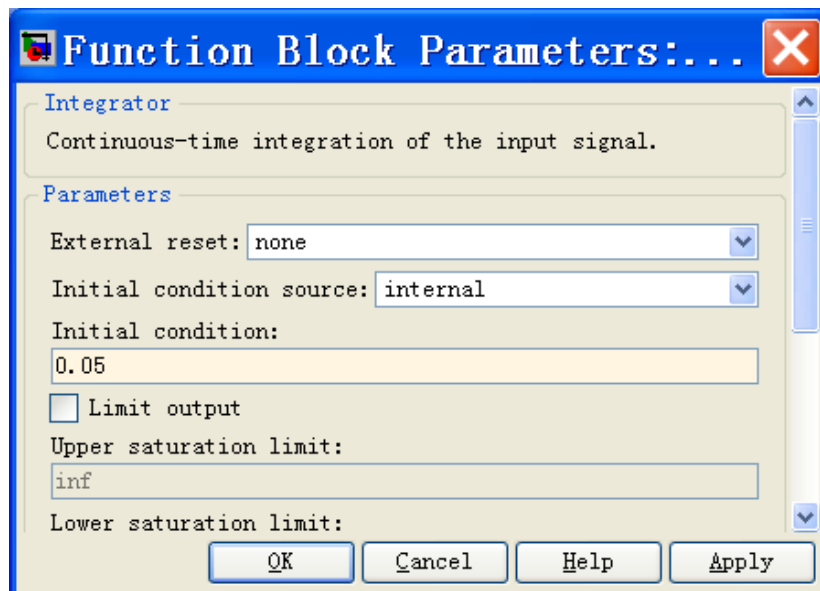


图 7.1-7

- (9)
- (10)
- (11)

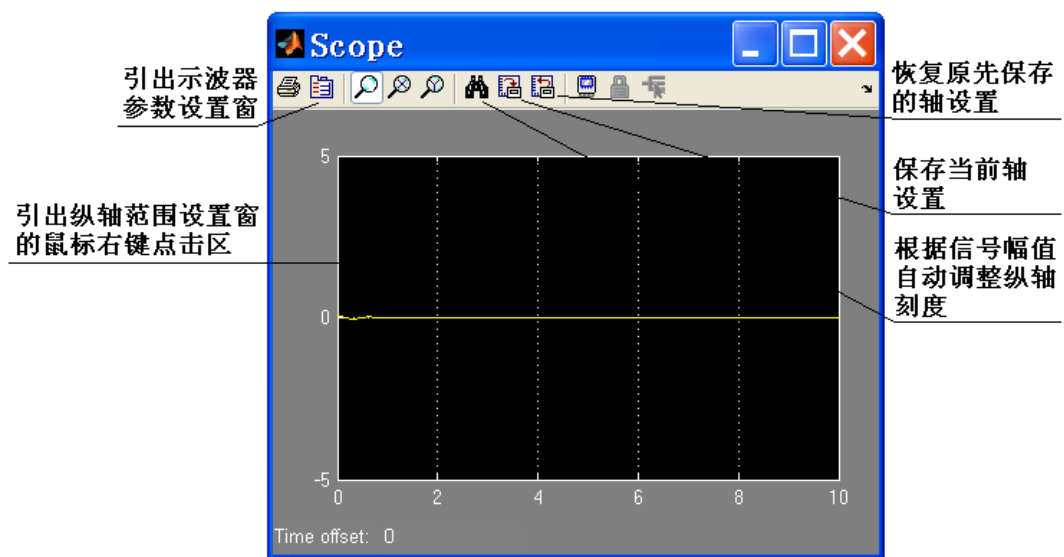


图 7.1-8

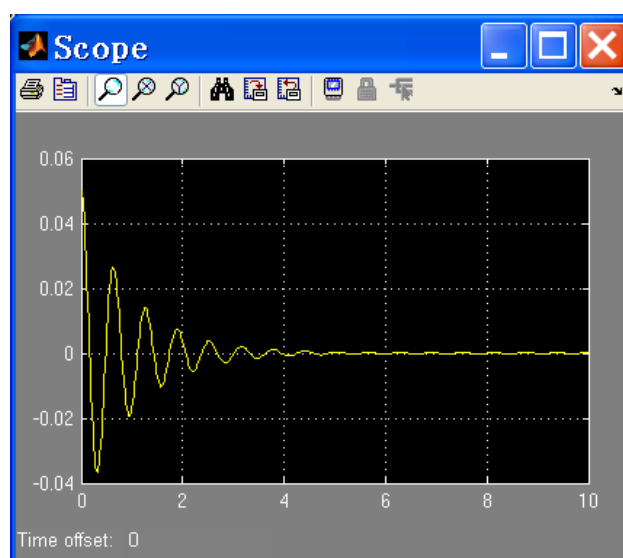


图 7.1-9

(12)

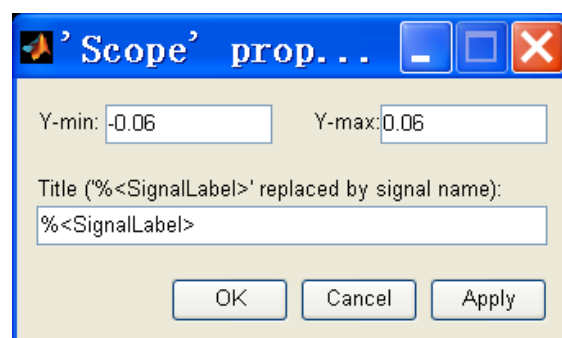


图 7.1-10

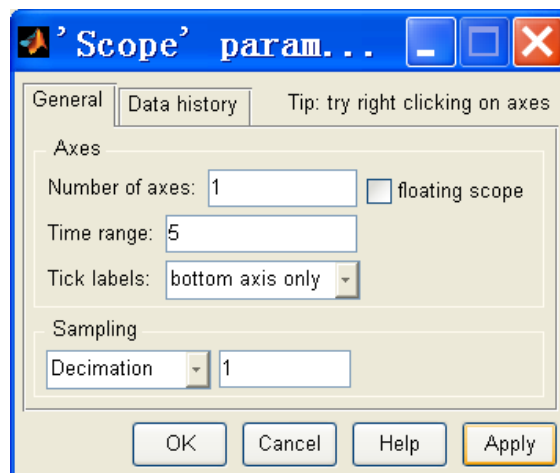


图 7.1-11

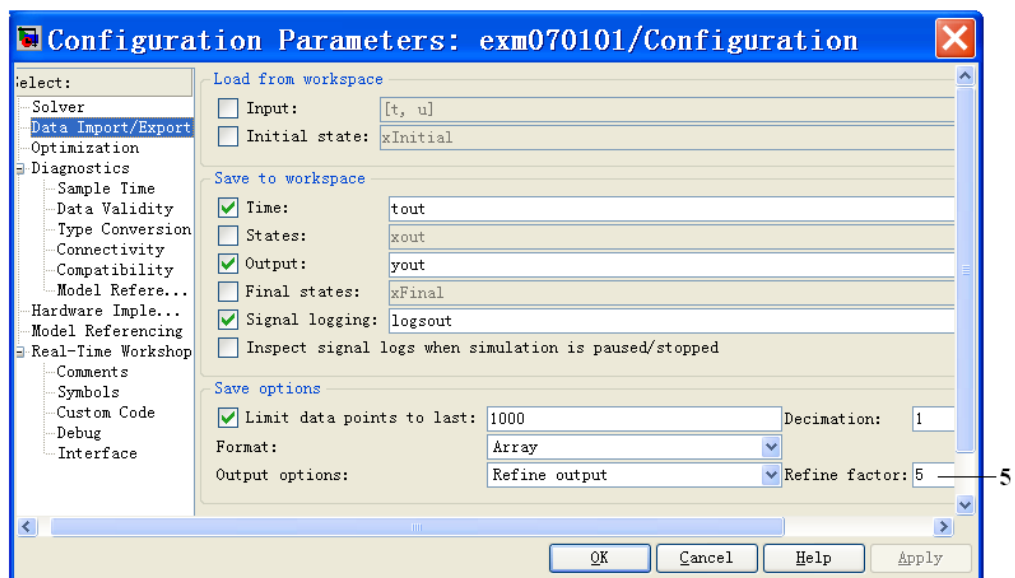


图 7.1-12

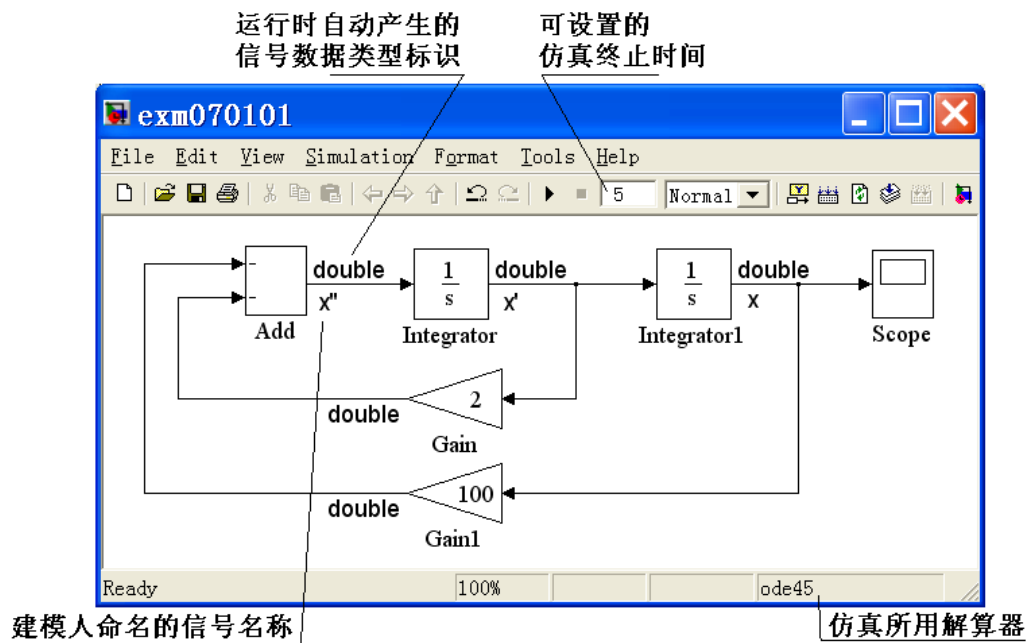


图 7.1-13

(13)

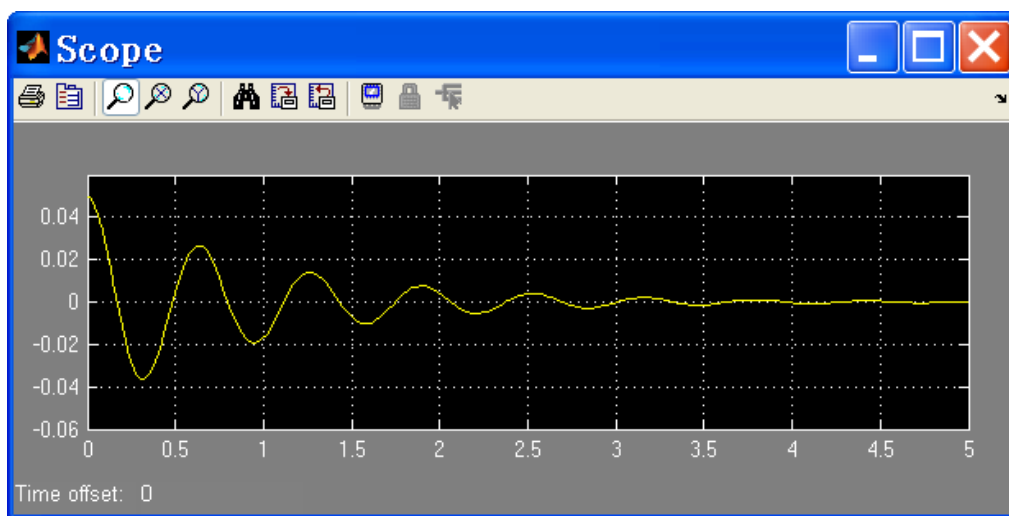


图 7.1-14

## 1.1.2 基于传递函数的SIMULINK建模

【例 7.1-2】对于图 7.1-15 所示的多环控制系统，(1) 求系统传递函数  $G(s) = \frac{Y(s)}{U(s)}$ ；(2)

求该系统的单位阶跃响应。

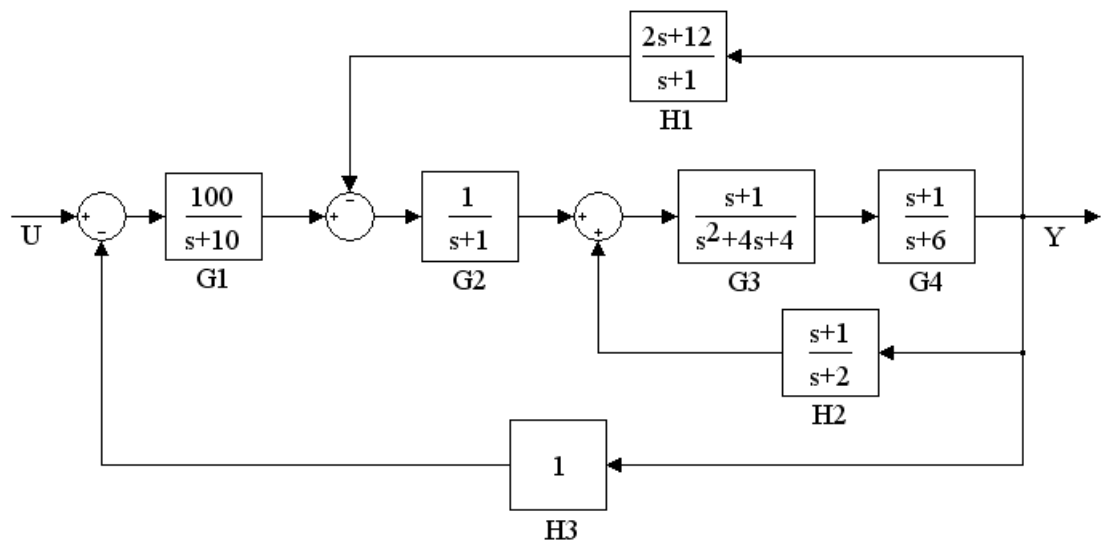


图 7.1-15 多环控制系统

(1)

(2)

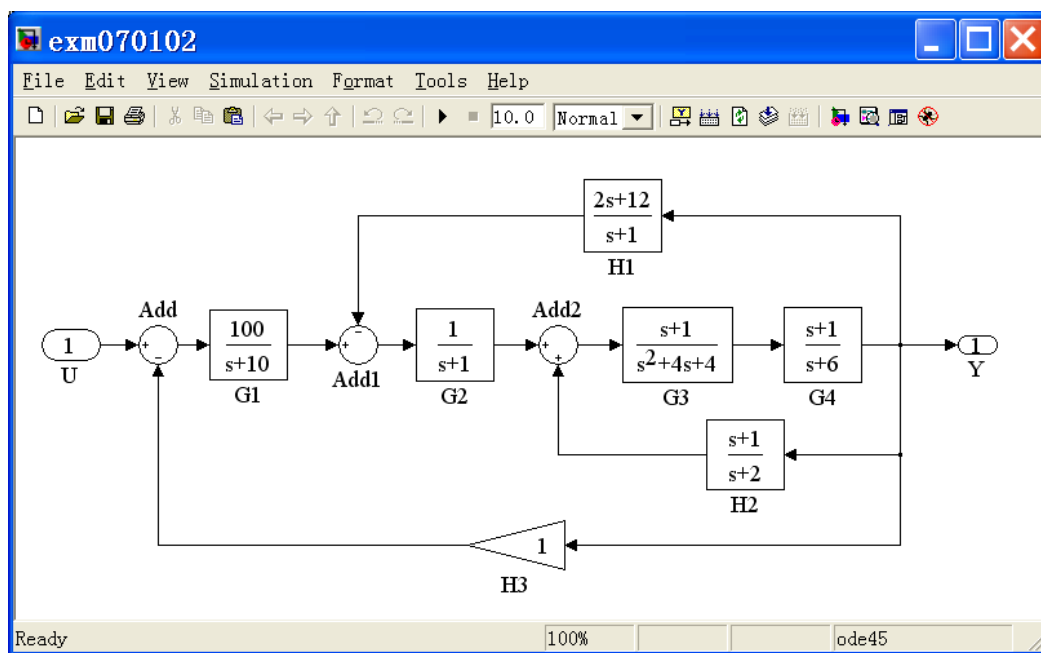


图 7.1-16

(3)

```
[A,B,C,D]=linmod2('exm070102');
STF=tf(minreal(ss(A,B,C,D)))
[Num,Den]=tfdata(STF);
Num{:},Den{:}
2 states removed.
```

Transfer function:

```

-7.471e-015 s^4 - 2.989e-014 s^3 + 100 s^2 + 300 s + 200
-----
s^5 + 21 s^4 + 157 s^3 + 663 s^2 + 1301 s + 910

ans =
      0   -0.0000   -0.0000  100.0000  300.0000  200.0000
ans =
 1.0e+003 *
      0.0010      0.0210      0.1570      0.6630      1.3010      0.9100

```

```

(4)
t0=(0:0.1:5)';
[y,t]=step(STF,t0);
plot(t,y,'LineWidth',3)
grid on
axis([0,5,0,0.4])
xlabel('t'),ylabel('y')

```

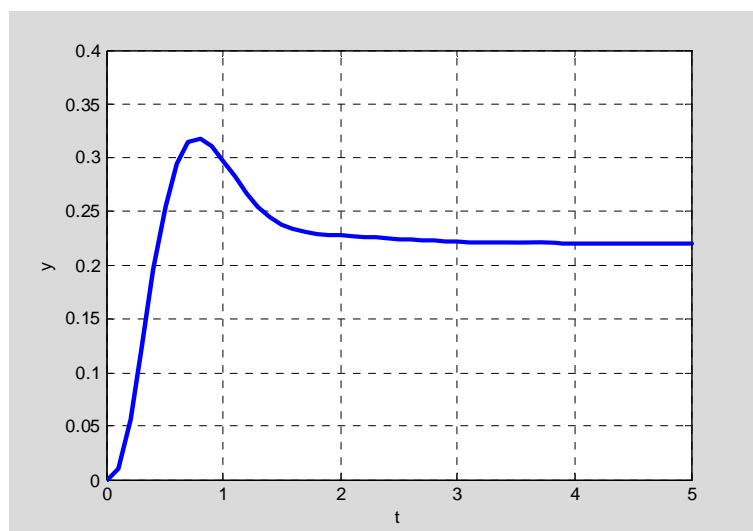


图 7.1-17

## 1.2 离散时间系统的建模与仿真

【例 7.2 -1】构建一个低通滤波系统的 SIMULINK 模型。输入信号是一个受正态噪声干扰的采样信号  $x(kT_s) = \sin(2\pi \cdot 10 \cdot kT_s) + 1.5 \cos(2\pi \cdot 100 \cdot kT_s) + n(kT_s)$ ，在此  $T_s = 0.001$ （秒），而  $n(kT) \sim N(0, 0.2^2)$ ；采用 8 阶 Butterworth 低通滤波器，以便从输入信号中过滤获得 10Hz 的输出信号。

(1)

$$y(k) = F(z)x(k)$$

$$F(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \cdots + b(n+1)z^{-n}}{1 + a(2)z^{-1} + \cdots + a(n+1)z^{-n}}$$

(2)

(3)

(4)

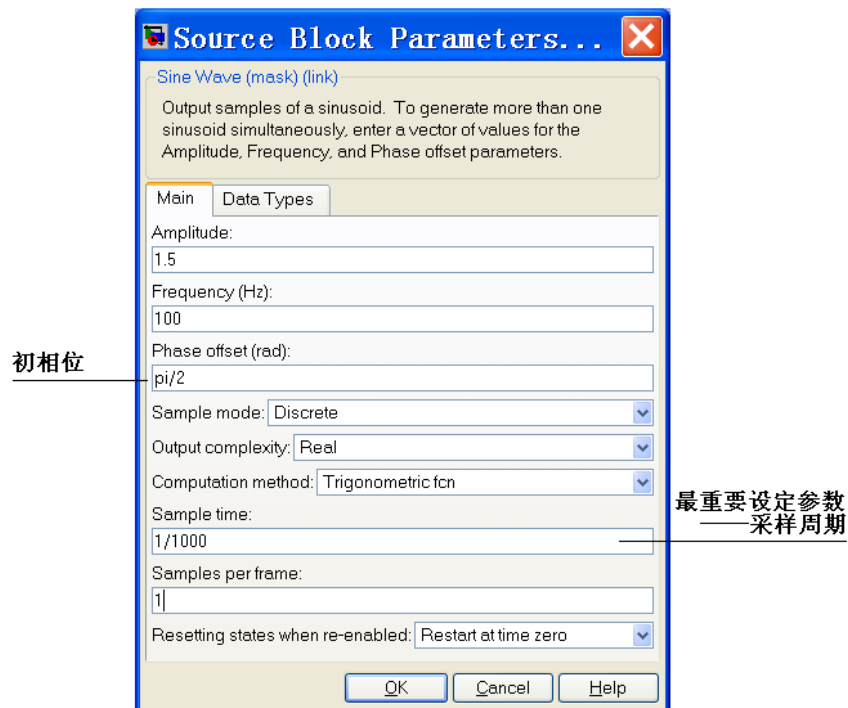


图 7.2-1

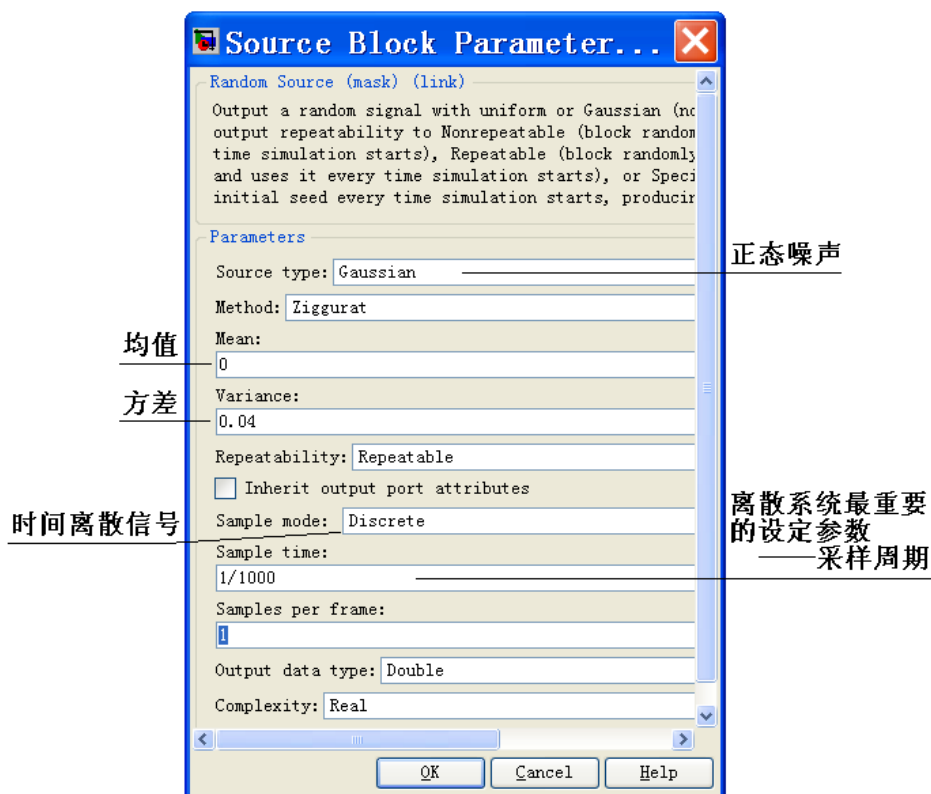


图 7.2-2

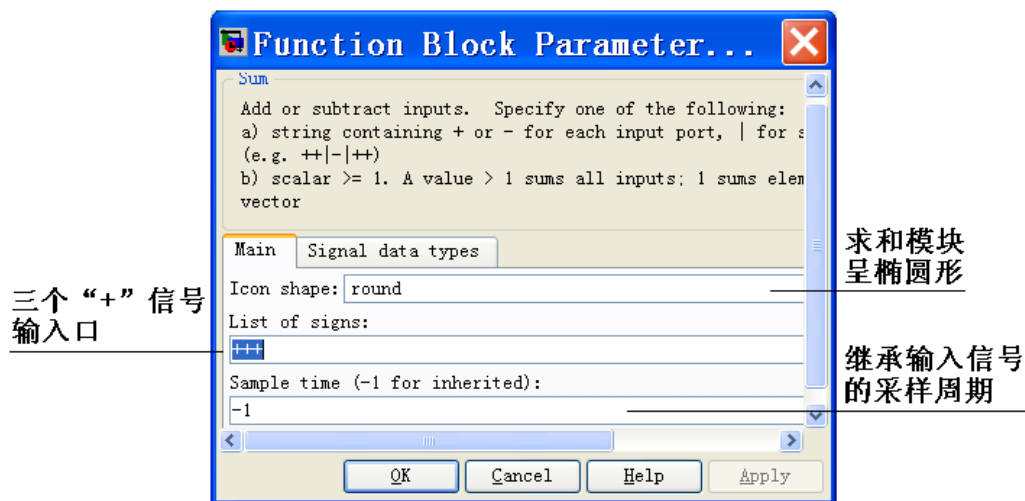


图 7.2-3



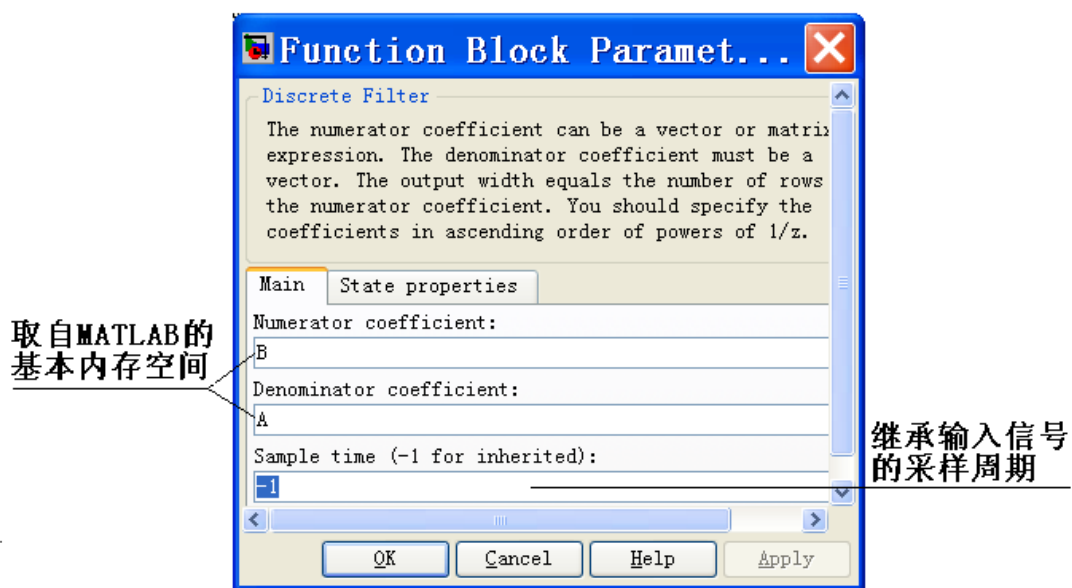


图 7.2-4

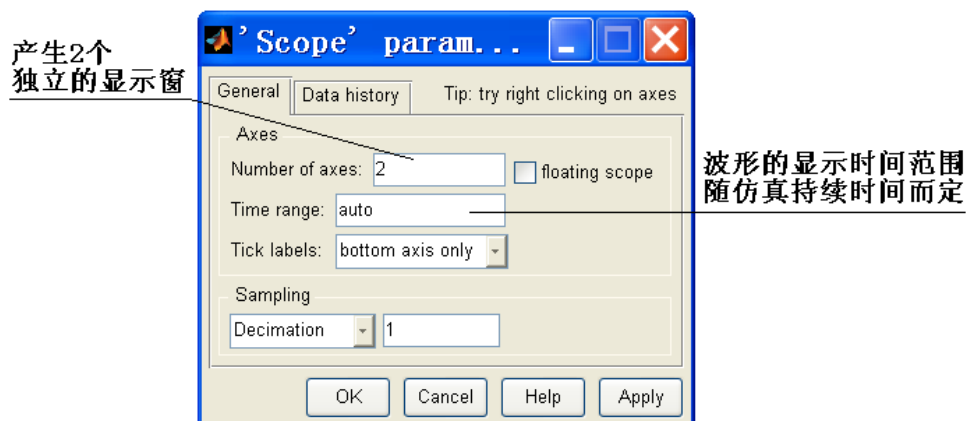


图 7.2-5

(5)

(6)

仿真终止时间设置为 1

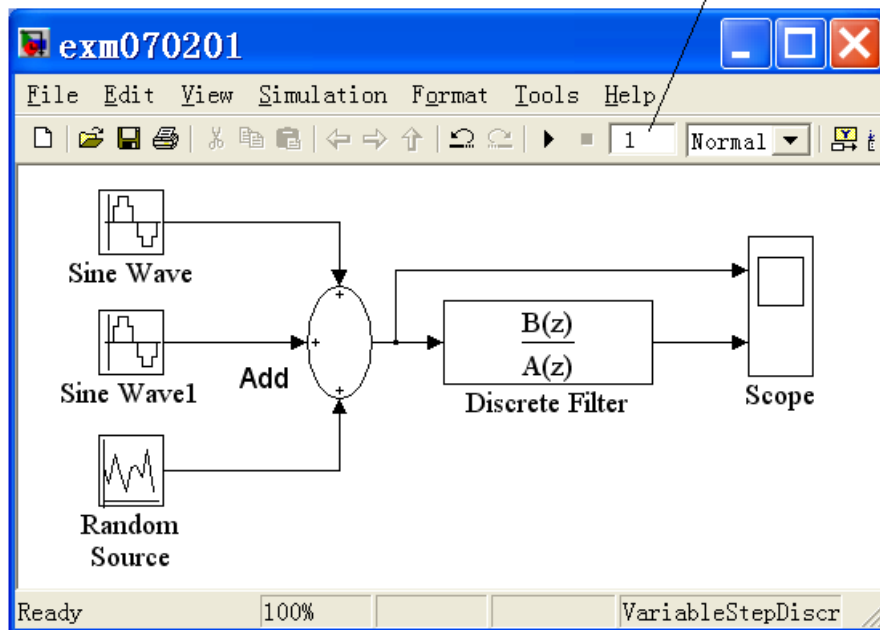


图 7.2-6

(7)

```
Ts=0.001;  
fs=1/Ts;  
fn=fs/2;  
fc=30;  
[B,A]=butter(8,fc/fn);
```

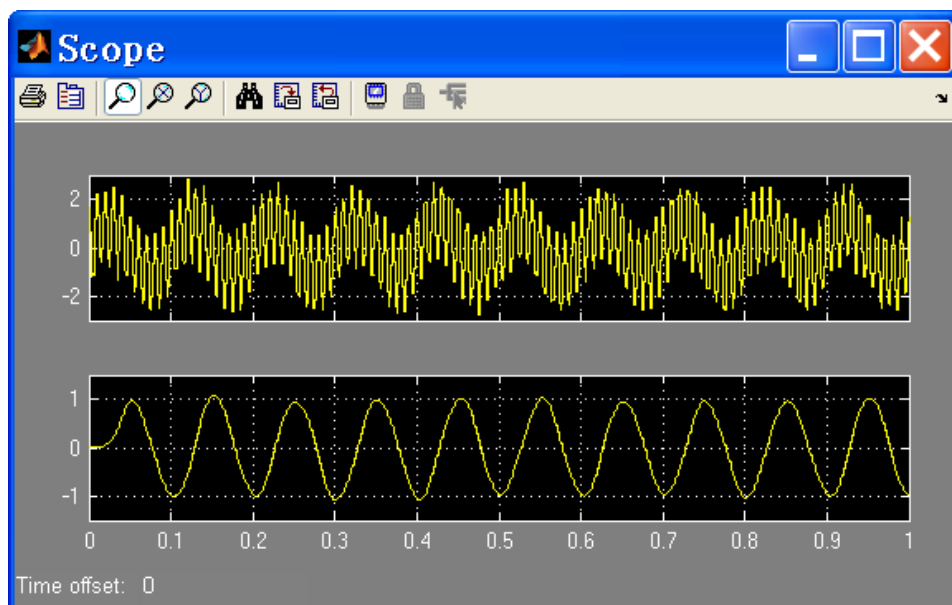


图 7.2-7

(8)

### 1.3 SIMULINK实现的元件级电路仿真

就仿真模型逼近被仿真系统的真实程度而言，前两节所建 SIMULINK 模型属于功能级仿真模型。这种模型所使用的模块与真实的物理器件之间不存在一一对应的关系；这种模型的构建以抽象了的数学模型为基础。较早的 SIMULINK，就是进行功能级仿真的软件环境。但最近发布的几个 SIMULINK 版本，已经在某些专业领域把仿真推进到了元器件级。

本节将利用 SimPowerSystems 中的库模块构建一个在元器件级上对应的电路模型，然后通过该模型进行电量的瞬态分析。

【例 7.3-1】在图 7.3-1 所示的电路中，已知  $L = 1(H)$ ， $C = 1(F)$ ， $R_1 = 0.5(\Omega)$ ， $R_2 = R_3 = 1(\Omega)$ ， $V_C(0^-) = -1(V)$ ， $i_L(0^-) = 1(A)$ ， $V_s = 10(V)$ ，开关 K 在  $t = 0$  时闭合。试采用 SIMULINK 的 SimPowerSystems 模块库器件进行元件级仿真。

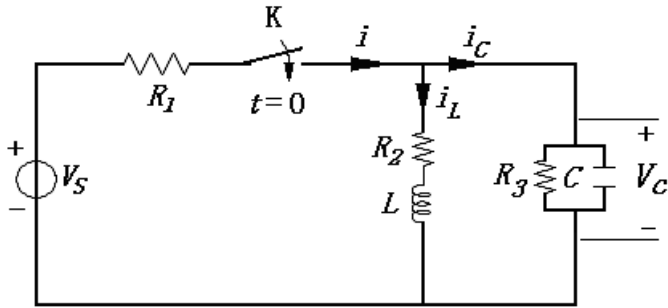


图 7.3-1 二阶 RLC 电路

- (1)
- (2)

表 7.3-1 建模所需模块的来源和参数设置

模型窗中 器件名称	库模块名称	所在库位置	参数设置
直流电压 源 $V_s$	DC Voltage Source	Electrical Sources	
电阻 $R_1$	Series RLC Branch	Elements	
电阻 $R_2$ 和 电感 $L$ 串联支路	Series RLC Branch	Elements	
电阻 $R_3$ 和 电容 $C$ 并联支路	Parallel RLC Branch	Elements	
开关 $K$	Breaker	Elements	

电流测量器 Mi	Current Measurement	Measurements	
电压测量器 Mv	Voltage Measurement	Measurements	
示波器 Si	Scope	Simulink\Sinks	
示波器 Sv	Scope	Simulink\Sinks	

(3)

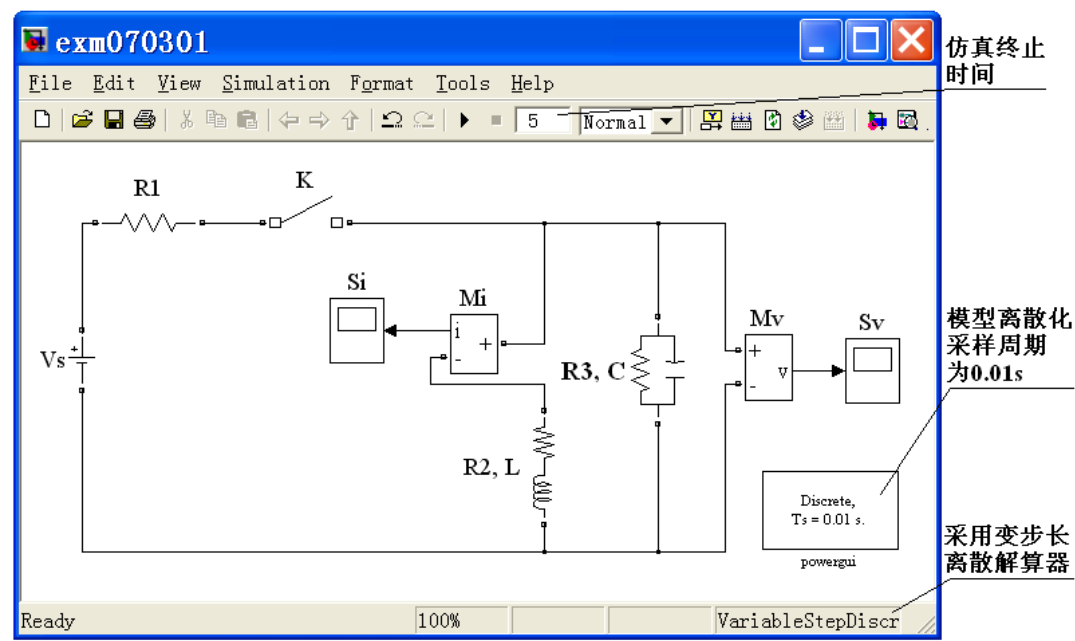


图 7.3-2

(4)

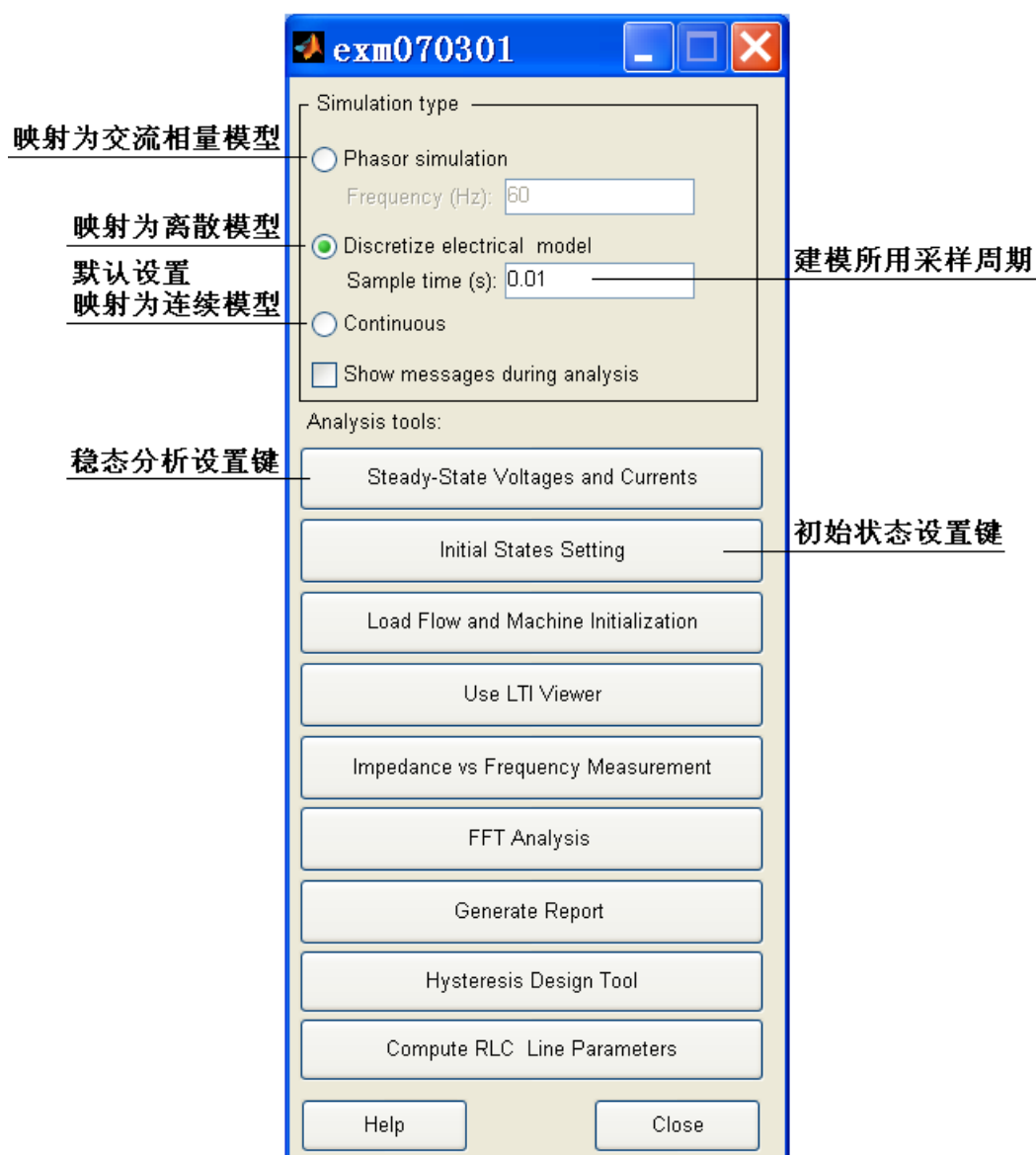


图 7.3-3

(5)

(6)

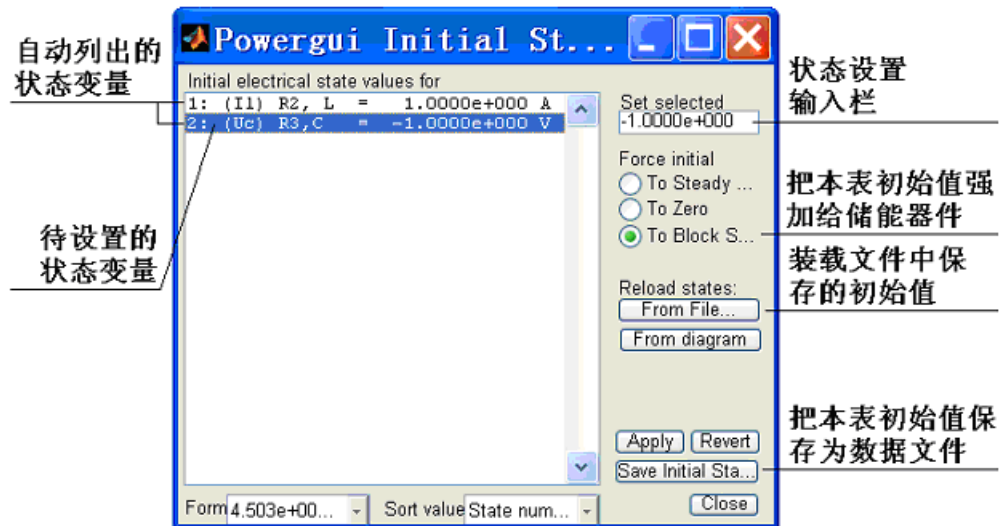


图 7.3-4

(7)

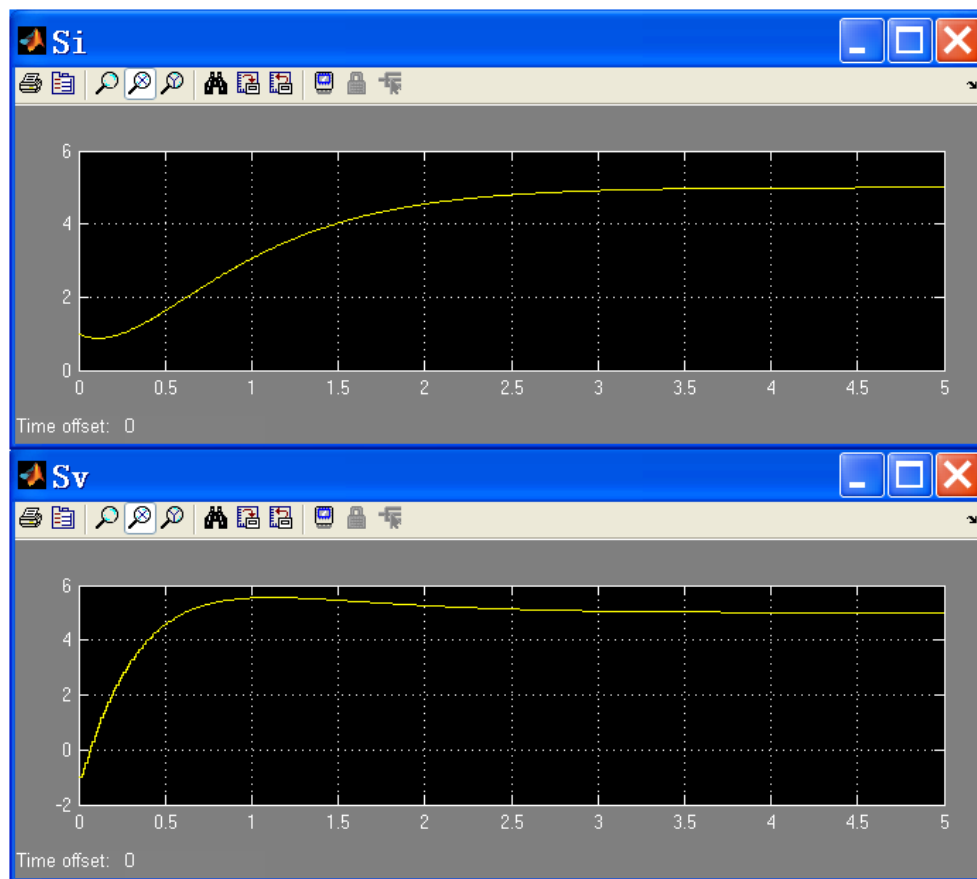


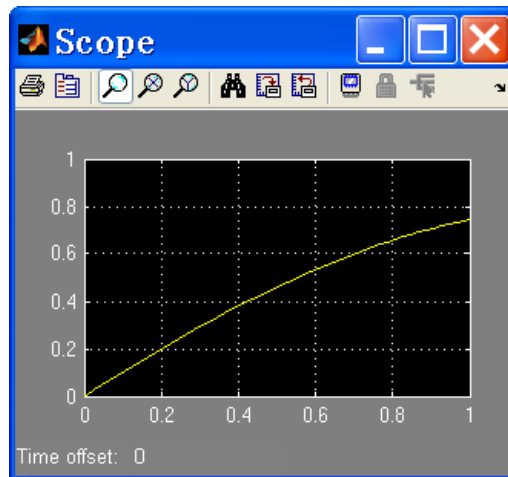
图 7.3-5

## 习题 7

1. 利用 SIMULINK 求解  $I(t) = \int_0^t e^{-x^2} dx$  在区间  $t \in [0,1]$  积分，并求出积分值  $I(1)$ 。（参见例 4.1-5）

〔答案〕

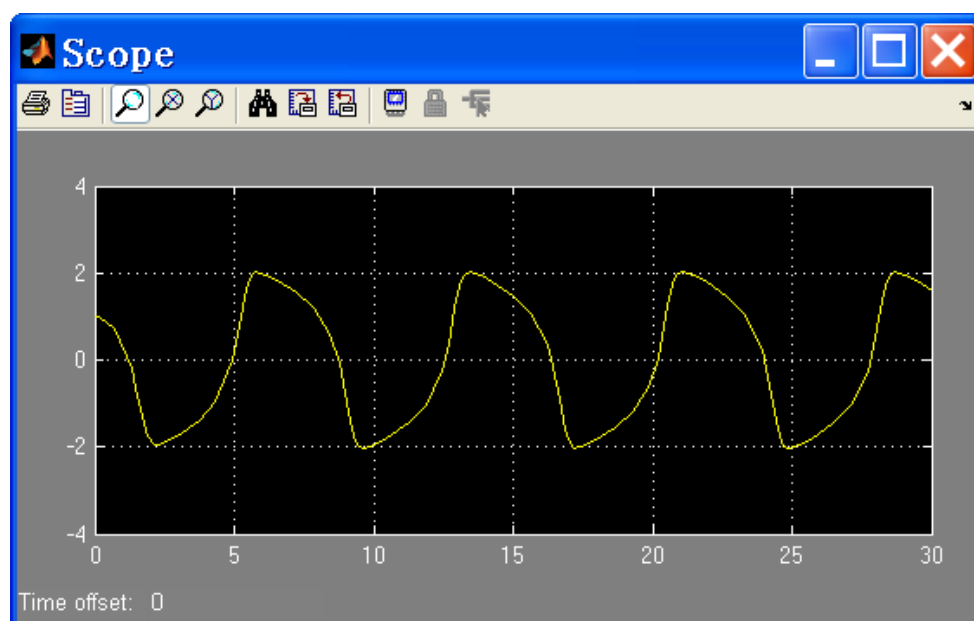
0.7468



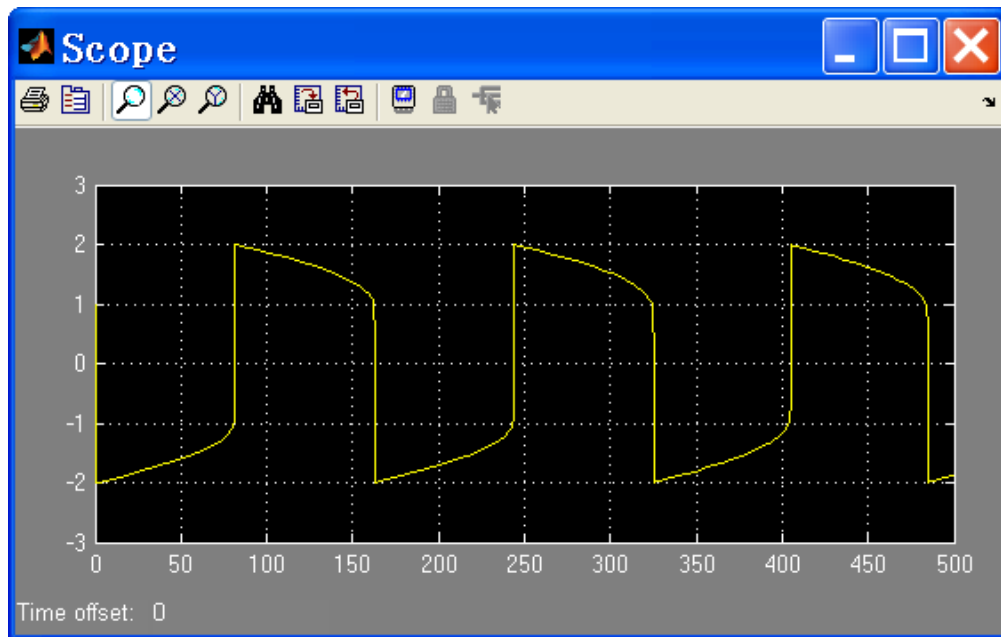
2. 利用 SIMULINK 求解微分方程  $\frac{d^2 x}{dt^2} - \mu(1-x^2)\frac{dx}{dt} + x = 0$ ，方程的初始条件为  $x(0) = 1, \frac{dx(0)}{dt} = 0$ 。在增益模块“Gain”取值分别为 2 和 100 的情况下（即数学表达式中  $\mu = 2, 100$ ）运行，给出运行结果。（参见例 4.1-9）

〔答案〕

$$\mu = 2$$



$$\mu = 100$$



3. 已知某系统的框图如图 p7-1 所示，求该系统的传递函数。

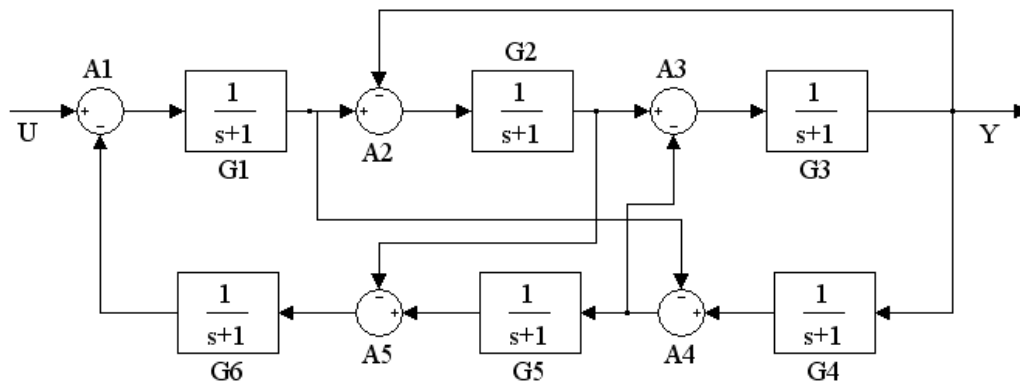


图 p7-1

〔答案〕

Transfer function:

$$s^4 + 5 s^3 + 9 s^2 + 7 s + 2$$

---


$$s^6 + 6 s^5 + 17 s^4 + 26 s^3 + 22 s^2 + 8 s + 1$$

Zero/pole/gain:

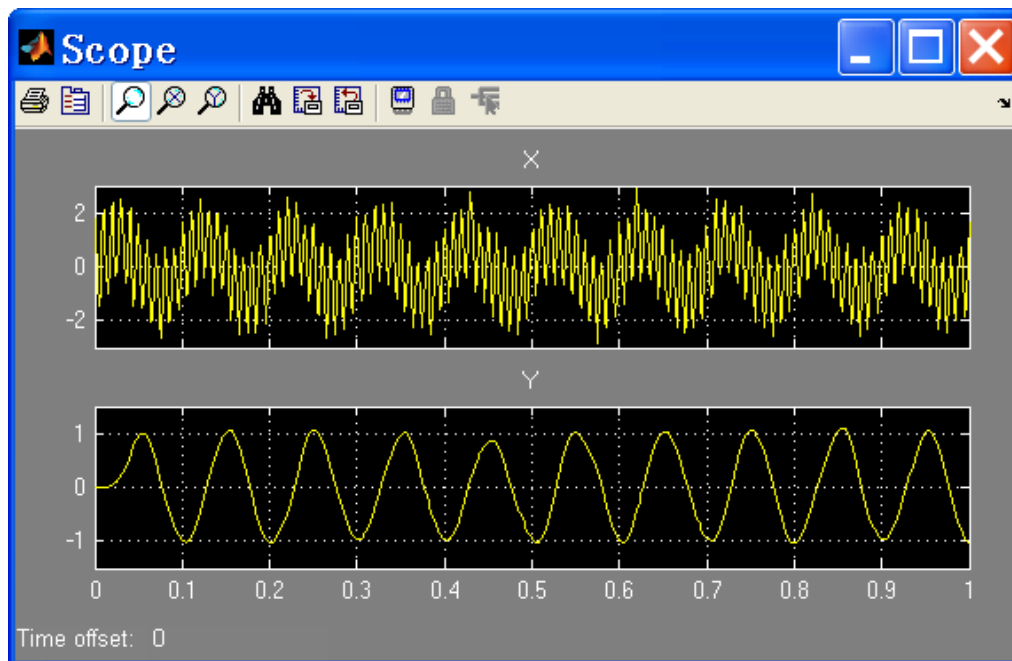
$$(s+2) (s+1)^3$$



$$(s+0.3177)^2 (s^2 + 2.682s + 3.148)^2$$

4. 采用 SIMULINK 基本库和 Signal Processing Blockset 库的“连续”时间的模块构建的 SIMULINK 模型解决第 7 章算例 7.2-1。

〔答案〕



5. 在如图 p7-2 所示的交流电路中, 其中  $Z_1 = j1 (\Omega)$ ,  $Z_2 = Z_4 = Z_5 = 1 (\Omega)$ ,  $Z_3 = -j1 (\Omega)$ ,

$\dot{V} = 20\angle 120^\circ (\text{V})$ ,  $\dot{I} = 10\angle 45^\circ (\text{A})$ ,  $f = 50 (\text{Hz})$ , 求  $Z_3$  支路中的电流和  $Z_5$  两端的电压。

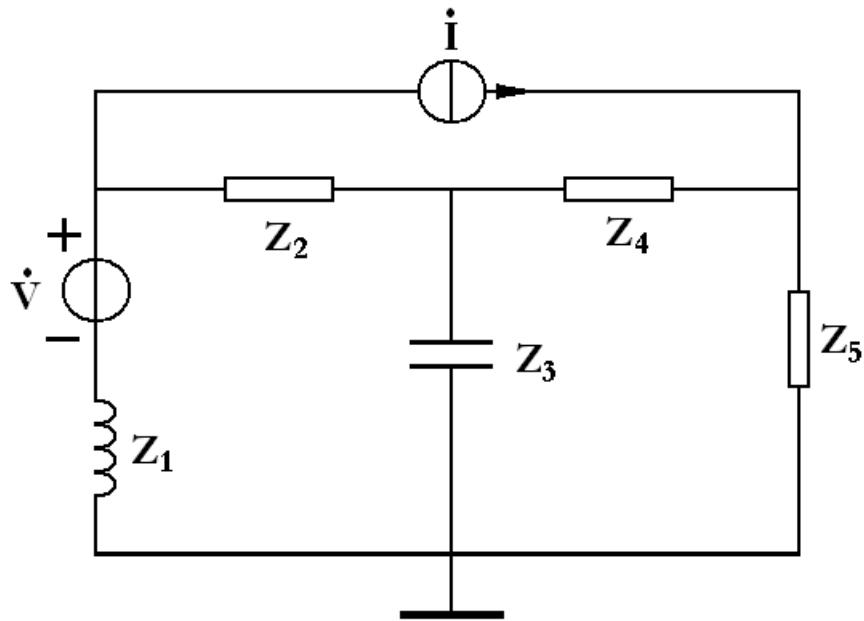
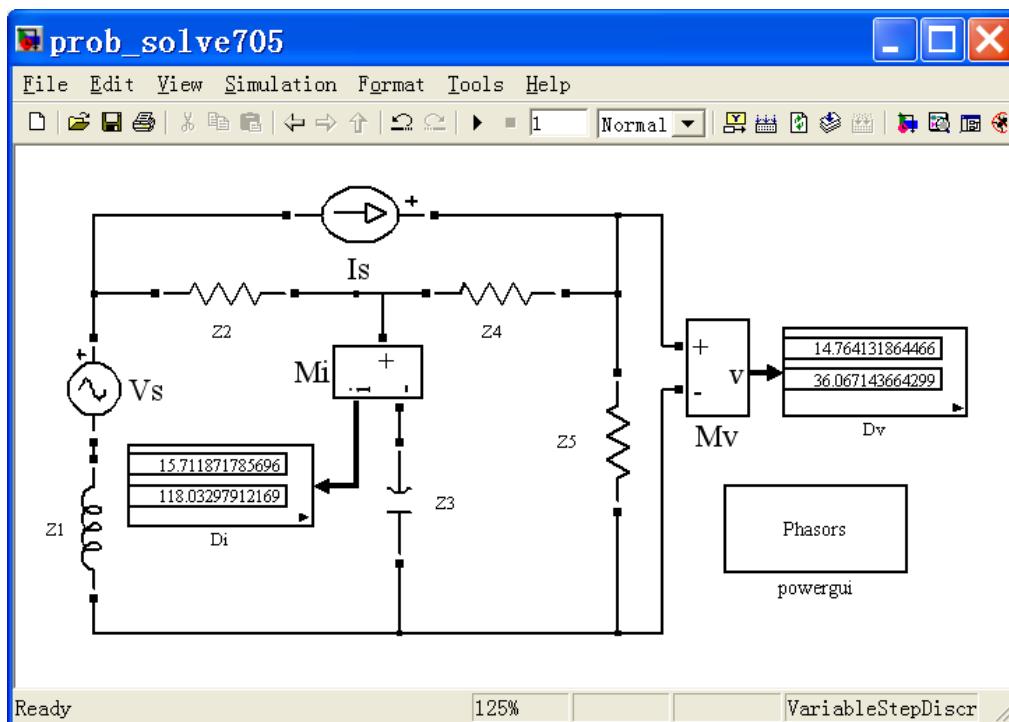


图 p7-2

【答案】



## 第 1 章 Notebook

Notebook 的功能在于：使用户能在 Word 环境中“随心所欲地享用”MATLAB 的浩瀚科技资源，为用户营造融文字处理、科学计算、工程设计于一体的完美工作环境。

MATLAB Notebook 制作的 M-book 文档不仅拥有 MS-Word 的全部文字处理功能，而且具备 MATLAB 无与伦比的数学解算能力和灵活自如的计算结果可视化能力。它既可以看作解决各种计算问题的字处理软件，也可以看作具备完善文字编辑功能的科技应用软件。

M-book 文档最显著的特点是它的“活”性：

- 它为论文、科技报告、讲义教材、学生作业的撰写营造了文字语言思维和科学计算思维的和谐环境。
- 用 M-book 写成的电子著作、电子文稿、讲义教材不仅图文并茂，而且动静结合。那些由 MATLAB 指令构成的例题、演示，都可供读者亲自操作，举一反三，从而在“手脑并用”的环境中由此及彼、由浅入深。

## 1.1 Notebook的配置和启动

### 1.1.1 Notebook的配置

#### (1) 与 MATLAB 适配的 Word

随 MATLAB 版本的升级，与其适配的版本也会发生变化。以 MATLAB R2006a 为例，能用来配置 Notebook 环境的 Word 版本是 2000，2002，2003。

#### (2) Notebook 的配置

在 Windows 已经装有前述 Word 版本的前提下，在 MATLAB 中配置 Notebook 环境十分简便。只要在 MATLAB 指令窗中运行以下指令，配置过程将自动进行。

```
notebook -setup
```

假如指令窗中出现如下信息，就表示配置成功。

```
Setup complete
```

### 1.1.2 Notebook的启动

#### 一 创建新的M-book文件

##### (1) 在 Word 默认窗口（即 Normal.dot）下创建新的 M-book 文档

- 选择 Word 窗口的下拉菜单项{文件：新建}；
- 在弹出的对话框中，选择“M-book”模板，按 [确定] 键；
- 于是，Word 的窗口由原先的默认式样变成“M-book”式样（如图 8.1-1 所示），自动开启一个新的 MATLAB 作为其服务器，而不管此前 Windows 平台上是否已经开启了 MATLAB。
- 注意：M-book 模板引出的 MATLAB 与直接开启的 MATLAB 的当前目录位置有所不同。

前者的当前目录是 `matlab\bin\win32`。为了保证 MATLAB 正常工作, 建议读者把目录改置为 `matlab\work`。

## M-book文档特有菜单

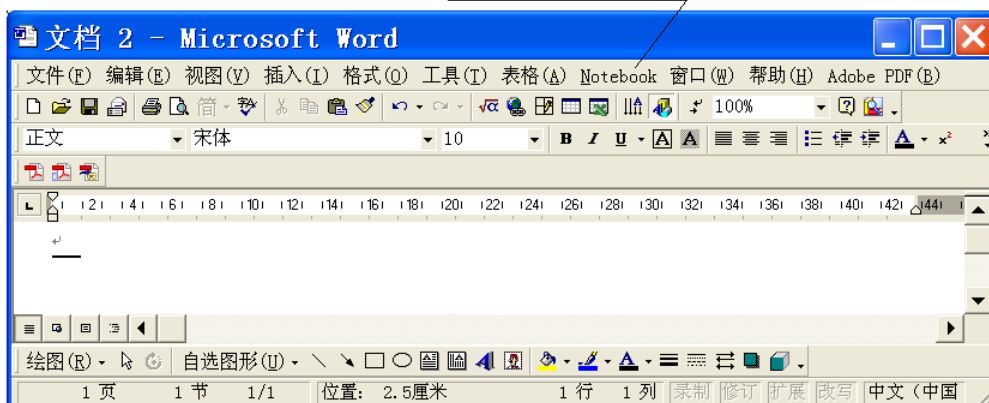


图 8.1-1 新建 M-book 文档的界面

## (2) 在 MATLAB 中创建新的 M-book 文件

在 MATLAB 指令窗中运行以下指令，都可以创建新的 M-book 文件

**notebook** 引出一个未命名的 M-book 文档界面

<b>notebook</b>	<b>NewFileName</b>	在当前目录上创建名为 NewFileName 的空白文件（慎用！）
-----------------	--------------------	-----------------------------------

〔说明〕

- 第一个指令引出的是使用 **M-book.dot** 模板的未命名文档界面（如图 8.1-1 所示）。
- 第二个指令不仅引出 **M-book** 界面，而且自动在当前目录上产生一个名为 **NewFileName.doc** 的空白文件。
- 第二个指令的使用要特别小心。要特别注意：**NewFileName** 千万不要与当前目录上已经存在的文件同名。否则，将把扩展名为 **doc** 的原有同名文件改变为一个空白文件，而且发生这种改变前没有任何提示可控操作。
- 以上指令运行后，不会引出新的 **MATLAB**，而只是把原有的 **MATLAB** 用作自动服务器，此时在 **MATLAB** 指令窗中会出现“Warning: MATLAB is now an automation server”的信息。

## 二 打开已有的M-book文件

(1) 在 Word 默认窗口下打开已有的 M-book 文件

- 在 Word 默认的窗口下打开已有 M-book 文件的方法与打开一般 Word 文件没有两样。最常用的方法是选中下拉菜单项 {文件: 打开}, 然后从弹出的对话框中选择所需要编辑的 M-book 文件。
- 以上指令运行后, 自动开启一个新的 MATLAB 作为 M-book 的服务器, 而不管此前 Windows 平台上是否已经开启了 MATLAB。

(2) 在资源管理器中打开已有的 M-book 文件

- 在资源管理器中，双击已有的 M-book 文件。

- 经以上操作后，会自动开启一个新的 MATLAB 作为 此 M-book 文件的服务器，而不管此前 Windows 平台上是否已经开启了 MATLAB。

### (3) 在 MATLAB 当前目录窗中打开已有的 M-book 文件

- 在 MATLAB 当前目录窗中，双击已有的 M-book 文件。
- 经以上操作后，会再开启一个新的 MATLAB 作为该 M-book 文件的服务器。

### (4) 在 MATLAB 指令窗中开启已有的 M-book 文件

- 在 MATLAB 指令窗中，运行指令 `notebook('FN.doc')`，在此 FN 是已有的 M-book 文件名。特别提醒：FN 之后一定要带扩展名 doc，否则将造成“清空原文件”的严重后果。
- 该指令运行后，将把当前 MATLAB 设置成文件 FN.doc 的自动服务器，而不再开启新的 MATLAB。

## 1.2 M-book模板的使用

M-book.dot 模板的外形和使用方法，几乎与普通 Word 模板 Normal.dot 完全相同。因此，在 M-book 中，文字、图象、表格、数学公式等的输入、排版、编辑方法，与在普通 Word 文档没有什么区别。

M-book 的特点在于：该模板以 MATLAB 为其计算服务器。这些特殊功能集中地反映在{Notebook} 下拉菜单中，参见图 8.2-1 。

定义输入细胞	Define Input Cell	定义自初始化细胞
	Define AutoInit Cell	
	Define Calc Zone	
取消细胞定义	Undefine Cells	
	Purge Selected Output Cells	删除输出细胞
	Group Cells	
	Ungroup Cells	
	Hide Cell Markers	
	Toggle Graph Output for Cell	
	Evaluate Cell	运行细胞中指令
	Evaluate Calc Zone	
	Evaluate M-book	
	Evaluate Loop...	
输出结果的格式设置	Bring MATLAB to Front	
	Notebook Options...	

图 8.2-1 {Notebook} 下拉菜单的常用功能项

### 1.2.1 输入细胞（群）的创建和运行

#### 一 细胞（群）

在 Notebook 中，凡参与 Word 和 MATLAB 之间信息交换的部分，就称之为“细胞（群）”

(Cells or Cell group)。由 M-book 送向 MATLAB 的指令，称为“输入”细胞 (Input cells)；由 MATLAB 返回 M-book 的计算结果，称为“输出”细胞 (Output cells)。

(输入)细胞和(输入)细胞群没有根本的区别，也没有必要刻意区别。输入、输出细胞不必成对生存：输入细胞可以单独存在，但输出细胞必须依赖输入细胞而存在。

## 二 基本操作

较之普通 Word, Notebook 最宝贵的东西就是输入细胞 (群)。从应用上讲，学会了输入细胞 (群) 的创建和运行，就意味着掌握了 Notebook。

细胞 (群) 创建、正确运行的两个基本操作：

- 以普通文本形式输入的必须是 MATLAB 指令。特别注意：标点符号必须是在英文状态下输入的。
- 不管文本形式的一条指令有多长，不管一行有多少条文本形式指令，不管有多少行文本形式指令，只要能用鼠标把它们同时“点亮”选中，那么可以被创建或运行，具体如下：
  - (A) 在文本内容“点亮”后，按组合键 [Ctrl-Enter]，或选中下拉菜单项 {Evaluate Cell}，那么被“点亮”部分就被激活成输入细胞 (群)，文字颜色将呈现为象征生命的绿色。与此同时，细胞所含指令被送进 MATLAB 运行，最后在该输入细胞的下方嵌入计算结果 (数据或图形)。这就是输出细胞，它的文字用蓝色显示。
  - (B) 在文本内容“点亮”后，按组合键 [Alt-D]，或选中下拉菜单项 {Define Input Cell}，那么被“点亮”部分只是变成了输入细胞 (群)，也没送去运行，当然也就没有运行结果。

## 三 输入细胞 (群) 操作示例

【例 8.2-1】演示：创建并运行输入细胞的基本操作方法。

### (1) 输入细胞的单纯生成法

在“正文”段落里，英文状态下，按普通的文本输入方式，键入如下一行指令；用鼠标把该行内容“点亮”；然后按组合键 [Alt-D]，那文本形式的指令就变成了“绿色”的输入细胞。但并不送去运算，当然也不会输出任何结果。

```
xx=(1:5)/5*pi;yy=sin(xx).*exp(xx)
```

### (2) 输入细胞生成、运行同时进行的操作方法

在英文状态下，在“正文”段落里，按普通的文本输入方式，键入如下一行指令；用鼠标把该行内容“点亮”；然后按组合键 [Ctrl-Enter]；于是该指令就会自动变成“绿色”输入细胞，并给出“蓝色”运算结果，即输出细胞。

```
x=(1:4)/4*pi;y=sin(x).*exp(x)
y =
    1.5509    4.8105    7.4605    0.0000
```

【说明】

- 在 M-book 中，最常用最可靠的操作是：“点亮”待运算指令，按组合键 [Ctrl-Enter]。

它的功能是：产生并运行当前细胞（群）。

- 在中文 M-book 文档中，特别注意：不要把中文标点混杂在 MATLAB 指令中。否则，或产生运行错误，或造成死机。

【例 8.2-2】演示：生成完整图形的所有指令必须定义在同一细胞（群）中。

```
t=0:0.1:10;y=1-cos(t).*exp(-t); %<1>
tt=[0,10,10,0];
yy=[0.95,0.95,1.05,1.05];
fill(tt,yy,'g'),axis([0,10,0,1.2]),xlabel('t'),ylabel('y') %<4>
hold on %<5>
plot(t,y,'k','LineWidth',4) %<6>
hold off %<7>
ymax=max(y) %<8>
ymax =
1.0669
```

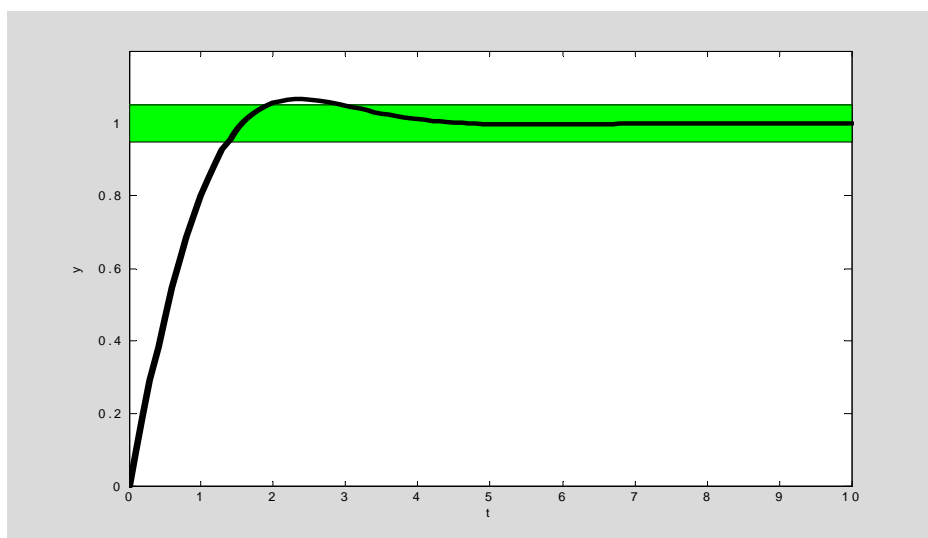


图 8.2-2 细胞群产生的完整图形

【说明】

- 使上述程序正确运行的最简单的方法是：用鼠标把从<1>到<8>的指令全部“点亮”，然后按组合键 [Ctrl-Enter]，就能得到正确结果。
- 保证程序正确运行的起码条件是：指令<4><5><6>应该在同一个输入细胞（群）中。否则会产生多幅图形，其中只有最后一幅是完整的。
- 不管输入细胞中显示运算结果的指令次序如何，在输出细胞中，非图形结果（包括数值、字符、符号对象等）总安排在图形的前面显示。

## 1.2.2 Notebook菜单的其他选项

## 一 自初始化细胞及其应用

### (1) 自初始化细胞

自初始化细胞 (AutoInit Cell) 与输入细胞功能的唯一不同是：当用户启动一个 M-book 文件时，包含在该文件中的自初始化细胞会自动被送去运算。而输入细胞不具备这种功能。若用户需要在打开文件时，对 MATLAB 工作内存进行初始化工作，那么自初始化细胞特别有用。

自初始化细胞有两种来源：一，文本形式的 MATLAB 命令；二，已经存在的输入细胞。为把它们变成自初始化细胞，先用鼠标选亮它们，然后选用 {Notebook: Define AutoInit Cell} 菜单选项即可。

### (2) 工作内存的初始化

M-book 所有计算都在 MATLAB 中进行，参与运算的所有变量都储存在 MATLAB 工作内存。各 M-book 文件和 MATLAB 指令窗分享同一个“计算引擎 (Compute Engine)”和同一个工作内存。工作内存中的变量是各 M-book 文件和 MATLAB 指令窗工作后共同产生的。对此，用户应有清醒认识。记住这条工作原理，将能使 M-book 文件灵活自如。

当用户同时打开几个 M-book 文件，或在 MATLAB 指令窗和 M-book 文件间交互运作时，要特别注意不同文件和窗口之间变量的相互影响。假如要保证某 M-book 文件独占 MATLAB 工作内存，保证该文件的输入输出数据间的一致性，一个有效的办法是：把 clear 定义为该文件的第一个自初始化细胞。

## 二 整个M-book文件的运行

{Notebook} 菜单中的 {Evaluate M-book} 选项可以运行整个 M-book 文件，即把文档中所有输入细胞送到 MATLAB 中去运行。不管光标处在该文档的什么地方，运行总是从文件首部开始。在整个 M-book 文件运行时，它不但会把所有原输出细胞中的内容刷新；而且会补写新的输出细胞。这个命令在保证整个 M-book 文件中所有指令、数据、图形的一致性方面十分有用。

在此提醒用户注意：假如原 M-book 文档的输出细胞自生成后没有再经历过编辑（如图形“对中”，输出细胞位置的前后“搬动”，输出细胞前后空行的“删除”等）操作，那么使用 {Evaluate M-book} 选项可得到良好的预期效果。否则，有可能造成整个版面的混乱。

实际上，M-book 模板的输出细胞采用“两端对齐”方式排版。对非图形输出来说，这是符合普通排版习惯的；但对图形输出来说，就显得别扭，因此，一般把图形“对中”。这样，以后的 {Evaluate M-book} 操作，就引起版面混乱。

此外，假如在原先的输入细胞后没有空行相隔，而紧接着普通文档的标题，那么 {Evaluate M-book} 操作，将导致标题错乱。

总之，慎用 {Evaluate M-book} 操作，尤其对较大的 M-book 文件。

## 三 删去M-book 文件所有输出细胞

{Notebook: Purge Output Cells} 菜单选项的作用是删去 M-book 文件中的所有输出细胞。它的具体操作步骤是：运行下拉菜单项 {编辑：全选}，使整个文件选亮；然后再运行下拉菜单项 {Notebook: Purge Output Cells}，所有输出细胞就被删去。这个指令在撰写报告、布置作业时常会用到。



### 1.2.3 输出细胞的格式控制

输出细胞容纳 MATLAB 的各种输出结果：数据、图形、错误信息。输出数据的有效数字、图形的大小都可以借助如图 8.2-3 所示的对话框加以控制。打开控制对话框的方法是：选中{Notebook: Notebook Options}菜单选项。



图 8.2-3 控制输出细胞格式的对话框

#### 一 输出数据的表示法

在 MATLAB 指令窗中，用 `format` 指令控制的输出数据表示方式共有 8 种：“Short”、“Long”、“Hex”、“Bank”、“Plus”、“Short e”、“Long e”、“Rational”。

在 M-book 中，输出细胞中数据的表示方式有三种控制方法：

- (1) 利用图 8.2-3 所示对话框中 {Numeric Format} 子框下拉列表进行设置。
- (2) 借助 M-book 输入细胞中的 `format` 指令进行设置。
- (3) 在 MATLAB 指令窗中，用 `format` 指令进行设置。

以上三种方法的效果相同，读者应视情况选用。

#### 二 输出数据间的空行控制

图 8.2-3 所示对话框中的 {Loose} 和 {Compact} 选项用来控制输入细胞与输出细胞之间的空白区间。比如，选择 {Loose} 后，在 M-book 文档的输入细胞和输出细胞之间加入一个空行。注意：这种控制方法与输入细胞群中的“`format loose`”、“`format compact`”命令有不同的功能。后者控制的将是输出细胞与输出细胞之间空行。

在此再需强调，不同输出格式给出不同的数据显示精度，但内部存储及运算都是以相同的双精度进行的。

### 三 图形的嵌入控制

在默认情况下，图 8.2-3 所示对话框中的“镶嵌选项” {Embed Figures in M-book} 处于“勾选”状态。此时，输出图形将被镶嵌在 M-book 文档中。假如“镶嵌选项”不勾选，那么在 M-book 文档中，将肯定没有输出图形。“镶嵌选项”的控制作用，将影响其后运行的所有输入细胞中的绘图输出。

【例 8.2-3】在同一细胞群中虽然包含绘制两幅图形的指令，但实际上只能把最后一个绘制的图形镶嵌进 M-book。

```
surf(peaks)                %绘制曲面图
colormap(hot)
t=(0:50)/50*pi;y=sin(t);
plot(t,y)                  %绘制曲线图
```

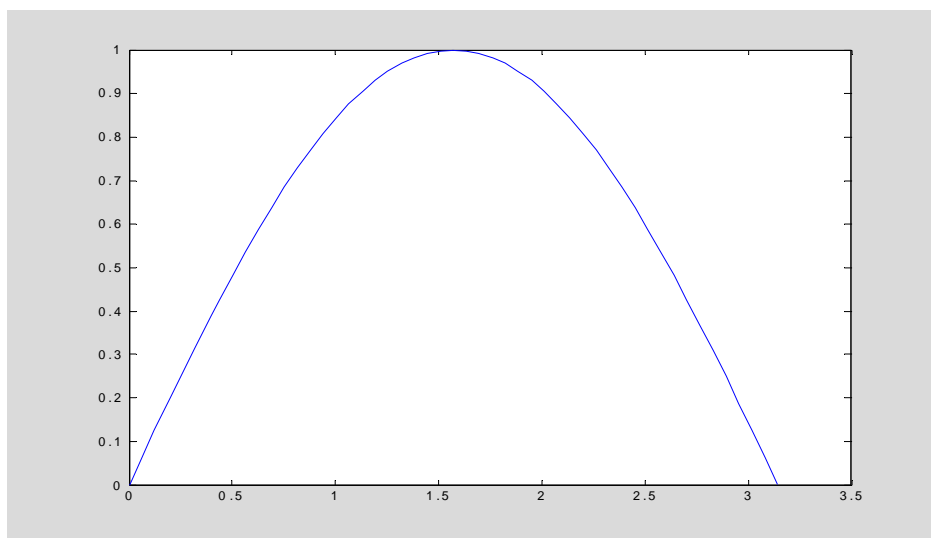


图 8.2-4 同一细胞群中最后一幅绘制的曲线图

### 四 嵌入图形大小的控制

在图 8.2-3 所示的 Notebook Options 对话框下方有三个栏目：“Units”、“Width”、“Height”。用户可通过对这三栏的设置，决定嵌入图形框的大小。

注意：由于嵌入图形框的大小由人为指定，一般不会顾及其宽高比是否与 MATLAB 图形窗相同。这样，诸如 axis square, axis equal 等控制宽高比的指令将不能发挥原先的作用。

图形一旦被嵌入 M-book，就象普通 Word 图形一样，可以被移动、缩放、剪裁和编辑。有关内容，请读者查阅 Word 书籍。

### 五 嵌入图形的背景色问题

在默认情况下，正常嵌入图形的背景色应是“灰/白”的。假如由于某种原因，所嵌图形出现“灰/黑”背景色，那么可采取以下两种措施的一种，尝试纠正：

- 打开 Notebook Options 对话框，确认“镶嵌选项”处于“勾选”状态，并再次点动 [OK] 键。然后，再重新运行输入细胞。
- 在 MATLAB 指令窗中，运行 `whitebg('white')`，或运行 `close;colordef`

`white`，然后再重新运行输入细胞。

## 六 M-book处理活动画面的能力

M-book 输出细胞中图形的产生机理是：在模板“宏”的控制下，输入细胞中的绘图指令被送去 MATLAB 计算后，MATLAB 依然是在图形窗中产生图形，只有该窗中的最终画面才被“复制”到剪贴板，并再从那里被粘贴到 M-book。

由此机理可知，M-book 模板既不可能表现活动画面，也不可能用于画面的交互操作。正因为如此，本书每凡涉及动画和交互画面时，都一再提醒读者，那种指令应在 MATLAB 指令窗中运行。

尽管如此，但仍可以从 M-book 发出动画指令，把图形窗带到前台，在图形窗中表现画面的活动情景。有兴趣的读者可以试试例 8.2-4 中的指令。这种处理方法在制作计算机演讲稿时，有可能会用到。

【例 8.2-4】在 M-book 中尝试导出动画图形。

假若下列 M 文件在 MATLAB 的搜索路径上，那么运行以下指令可以带出动画图形窗。

```
anim_zzyl(1),shg
```

〔说明〕

- 在运行以上指令前，应该先使“镶嵌选项”处于不勾选状态。

### 1.3 使用 M-book 模板的若干参考技法

- 文档中的 MATLAB 指令必须在英文状态下输入；指令中的标点符号必须在英文状态下输入。
- 续行号不能使用于输入细胞。
- 不管一条指令多长，只要不用“硬回车”换行，总可以被鼠标全部“点亮”并按组合键 [Ctrl-Enter] 后正确地运行。
- MATLAB 指令在 M-book 中的运行速度比在指令窗中慢得多。因此，符号计算指令、编译指令等在 M-book 中运行时，有可能发生“运行时间过长”或“出错”的警告。遇到这种情况，用户最好还是让那些指令在指令窗中直接运行。