

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、变量编辑器、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 R2010a（即 MATLAB 7.10）版要求 WinXP 或 Windows Vista 平台。下面介绍从光盘上安装 MATLAB 的方法。

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。

在安装 MATLAB.R2010a 时，会出现一个界面，该界面上有两个选项：Typical 和 Custom。由于近年电脑的硬盘容量很大，所以一般用户为方便计，直接点选“Typical”即可。

安装完成后，一般会产生两个目录：

- MATLAB 软件所在的目录
 - 该目录位置及目录名，都是用户在安装过程中指定的。比如，C:\MATLAB R2010a。
 - 该目录包含 MATLAB 运作所需的所有文件，如启动文件、各种工具包等。
- MATLAB 自动生成的供用户使用的工作目录
 - 该目录是由安装 MATLAB 时自动生成的，是专供用户存放操作 MATLAB 中产生的中间文件使用的。
 - 该工作目录的名称是 MATLAB。它一般登录在 C:\Documents and Settings\acer\My Documents 文件夹下。（注意：这文件夹名中的 acer 会随电脑不同而变。）
 - 该工作目录 C:\Documents and Settings\acer\My Documents\MATLAB 被自动记录在 MATLAB 的搜索路径中。因此，在这目录上的 M 文件、MAT 文件、MDL 文件等都能被 MATLAB 搜索到。


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。

1.2.2 Desktop 操作桌面简介

MATLAB R2010a 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、MATLAB 工作内存空间（Workspace）浏览器、历史指令（Command History）窗。

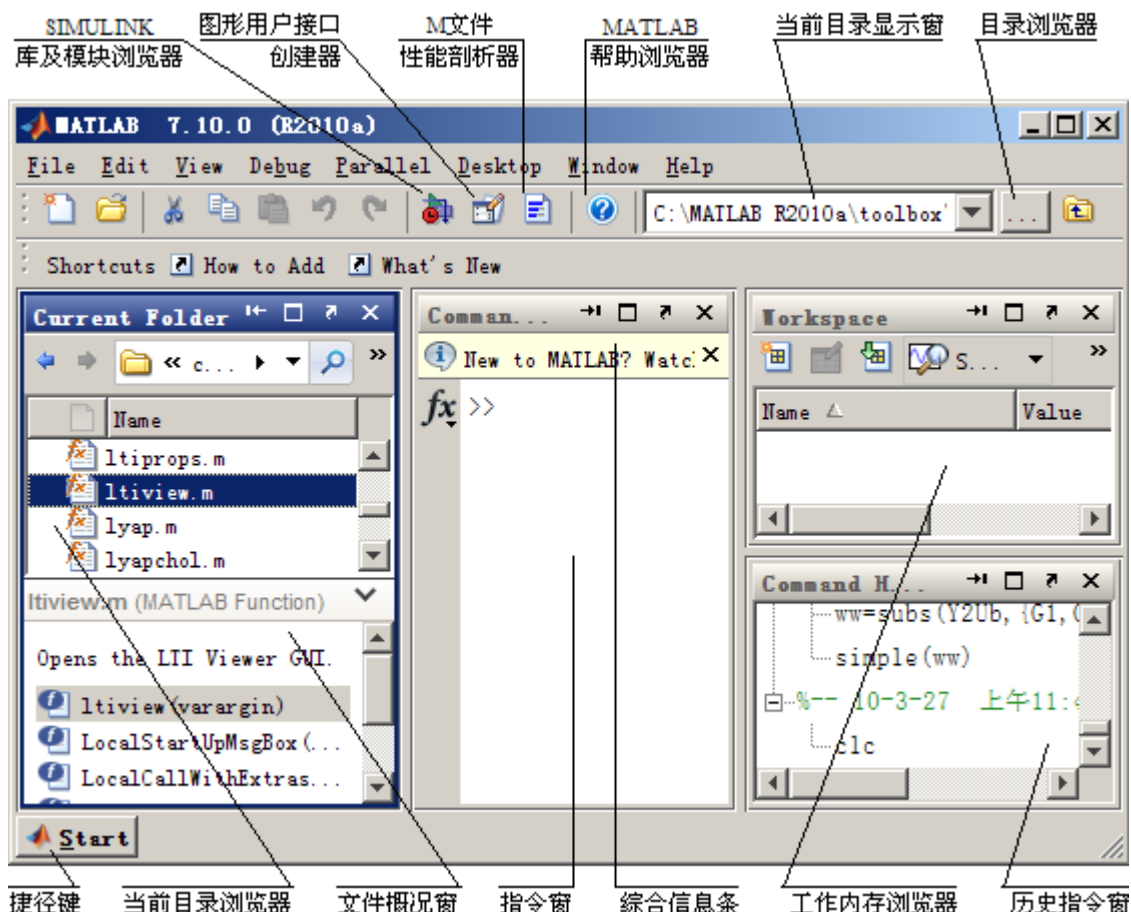


图 1.2-1 Desktop 操作桌面的默认外貌

- 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。

- 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。

此外，在当前目录浏览器正下方，还有一个“文件概况窗”。该窗显示所选文件的概况信息。比如该窗会展示：M 函数文件的 H1 行内容，最基本的函数格式；所包含的内嵌函数和其它子函数。

- 工作空间浏览器

该浏览器默认地位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。

- 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。


- 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。

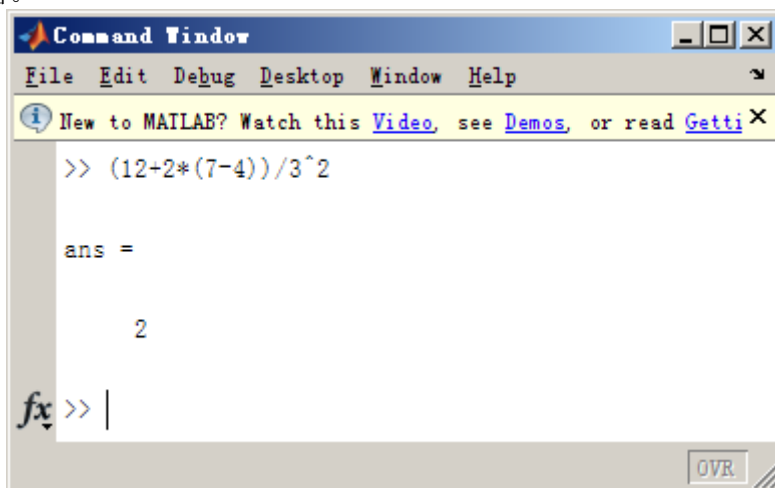



图 1.3-1 几何独立的指令窗

【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。

1 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} 。

2 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。

3 MATLAB 默认的数学常数

MATLAB 为一些数学常数(Math Contants)预定义了变量名，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 为数学常数预定义的变量名

预定义变量	含 义	预定义变量	含 义
eps	浮点数相对精度 2^{-52}	NaN 或 nan	不是一个数 (Not a Number), 如 $0/0$, ∞/∞
i 或 j	虚单元 $i = j = \sqrt{-1}$	pi	圆周率 π
Inf 或 inf	无穷大, 如 $1/0$	realmax	最大正实数, 默认 $1.7977\text{e}+308$
intmax	可表达的最大正整数, 默认 (2147483647)	realmin	最小正实数, 默认 $2.2251\text{e}-308$
intmin	可表达的最小负整数, 默认 (-2147483648)		

【说明】

- 假如用户对表中任何一个预定义变量进行赋值, 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。
- 关于它们的更详细的帮助信息, 可在 MATLAB 帮助浏览器左侧 `Contents` 页的 <MATLAB/Functions/Mathematics/Math Constants> 找到。

【例 1.3-3】运用以下指令, 以便初步了解关于常数的预定义变量。本例演示: 各常数的含义。

```
format short e
RMAd=realmax('double')      % 双精度类型 (默认) 时最大实数
RMAs=realmax('single')      % 单精度类型时最大实数
RMAd =
    1.7977e+308
RMAs =
    3.4028e+038

IMA64=intmax('int64')        % int64 整数类型时最大正整数
IMA32=intmax                 % int32 (默认) 整数类型时最大正整数
IMA16=intmax('int16')        % int16 整数类型时最大正整数

IMA64 =
    9223372036854775807
IMA32 =
    2147483647
IMA16 =
    32767

e1=eps                       % 双精度类型时的相对精度
e2=eps(2)                    % 表达 2 时的绝对精度
e1 =
    2.220446049250313e-016
e2 =
    4.440892098500626e-016

pi
ans =
    3.141592653589793
```

4 运算符和表达式

(1) 经典教科书上的算术运算符 (Arithmetic Operations) 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 $b ./ a$
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 (1×1) 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。
- 关于它们的更详细的帮助信息, 可在 MATLAB 帮助浏览器左侧 Contents 页的 <MATLAB/ User Guide/ Programming Fundamentals/ Basic Program Components/ Operators/ Arithmetic Operations>节点找到。
- 关于它们的帮助信息, 也可在 MATLAB 帮助浏览器左上方的搜索栏中输入 Arithmetic Operations, 经搜索获得。

(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。

5 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 $z_1 = 4 + 3i$, $z_2 = 1 + 2i$, $z_3 = 2e^{\frac{\pi}{6}i}$ 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 +      3i
```

【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 +      2i
z3 =
      1.7321 +      1i
z =
      1.884 +      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 =
      1.884
image_z =
      5.2631
magnitude_z =
      5.5902
angle_z_radian =
      1.2271
angle_z_degree =
      70.305
```

【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')
z12 =
      5.0000 + 5.0000i
```

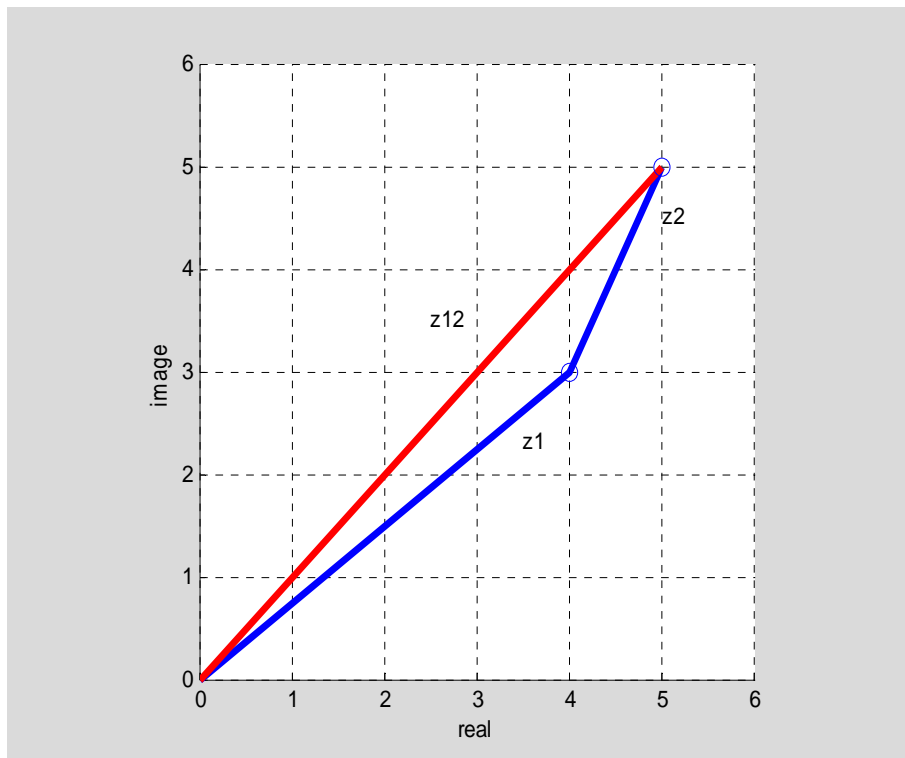



图 1.3-2 两个复数相加

【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
hold off
```

%保证屏幕显示呈真圆

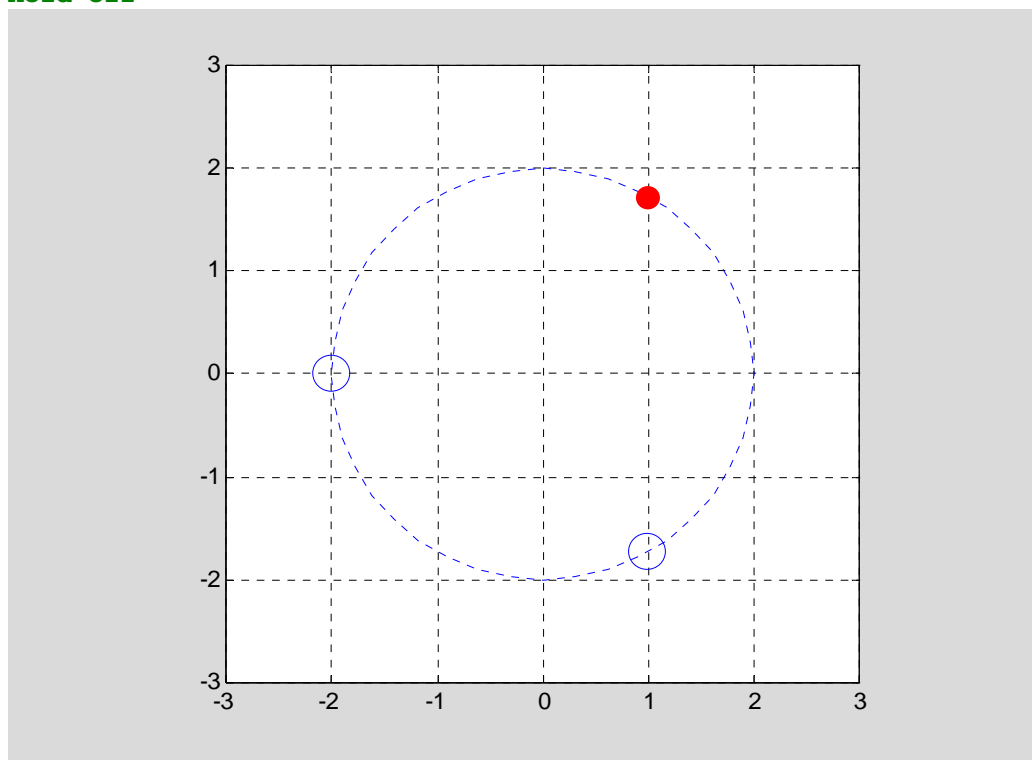


图 1.3-3 (-8) 的全部三次方根分布

【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。

6 面向数组设计的运算——MATLAB 特点之二

在 MATLAB 中，标量数据被看作 (1×1) 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')
```

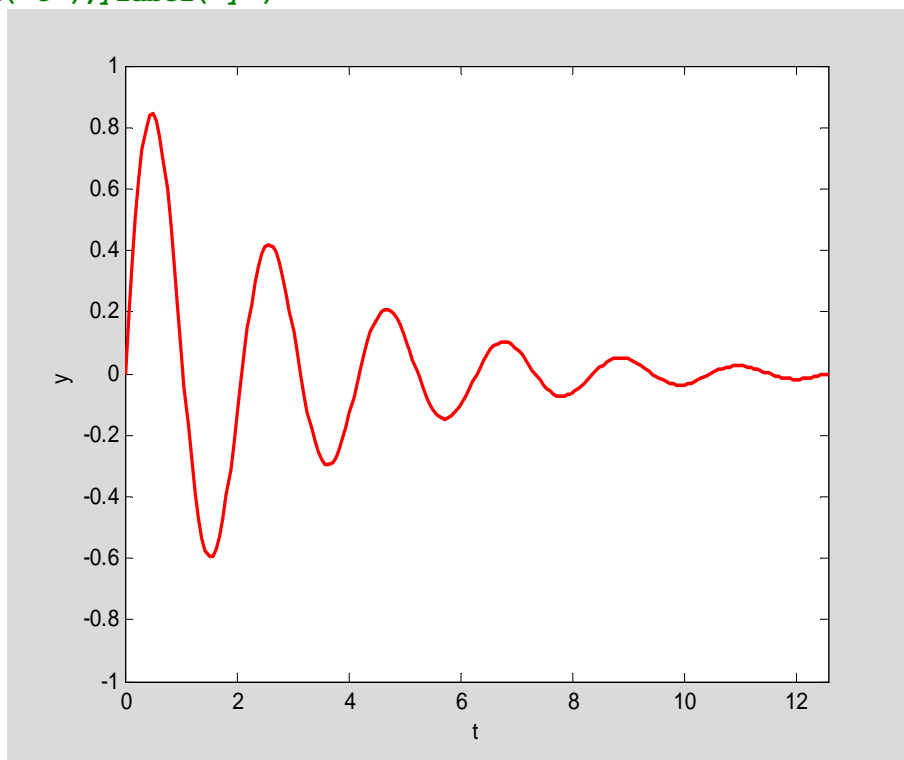


图 1.3-4 衰减振荡曲线

【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（ \mathbf{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

1 默认的输入显示方式

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：
 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。
 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。
 输入的字符串自动呈现为紫色字体。

2 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
format short	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
format long	小数点后 15 位数字表示	3.141592653589793
format short e	5 位科学记数表示	3.1416e+00

format long e	15 位科学记数表示	3.14159265358979e+00
format short g	从 format short 和 format short e 中自动选择最佳记数方式	3.1416
format long g	从 format long 和 format long e 中自动选择最佳记数方式	3.14159265358979
format rat	近似有理数表示	355/113
format hex	十六进制表示	400921fb54442d18
format +	显示大矩阵用。正数、负数、零 分别用 +, -, 空格表示。	+
format bank	(金融) 元、角、分表示	3.14
format compact	显示变量之间没有空行	
format loose	在显示变量之间有空行	
【说明】 format short 显示格式是默认的显示格式。 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。		

3 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； 用做单下标援引时，表示全部元素构成的长列； 用做多下标援引时，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； 在数组援引时用； 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】 <ul style="list-style-type: none"> ● 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。 ● 关于它们的更详细的帮助信息，可在 MATLAB 帮助浏览器左侧 Contents 页的 <MATLAB/ User Guide/ Programming Fundamentals/ Basic Program Components/ Symbol Reference>节点找到。 		

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
ans	最新计算结果的默认变量名	edit	打开 M 文件编辑器
cd	设置当前工作目录。	exit	关闭/退出 MATLAB
clf	清除图形窗	help	在指令窗中显示帮助信息
clc	清除指令窗中显示内容	more	使其后的显示内容分页进行
clear	清除 MATLAB 工作空间中保存的变量	quit	关闭/退出 MATLAB
dir	列出指定目录下的文件和子目录清单	return	返回到上层调用程序；结束键盘模式
doc	在 MATLAB 浏览器中，显示帮助信息	type	显示指定 M 文件的内容
diary	把指令窗输入记录为文件	which	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。

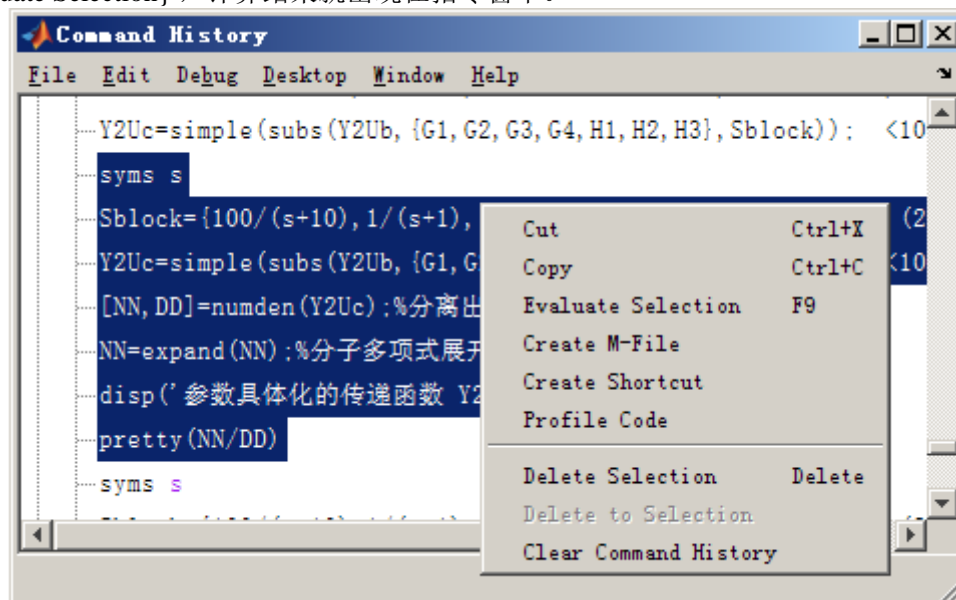


图 1.5-1 再运行历史指令的演示

【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。

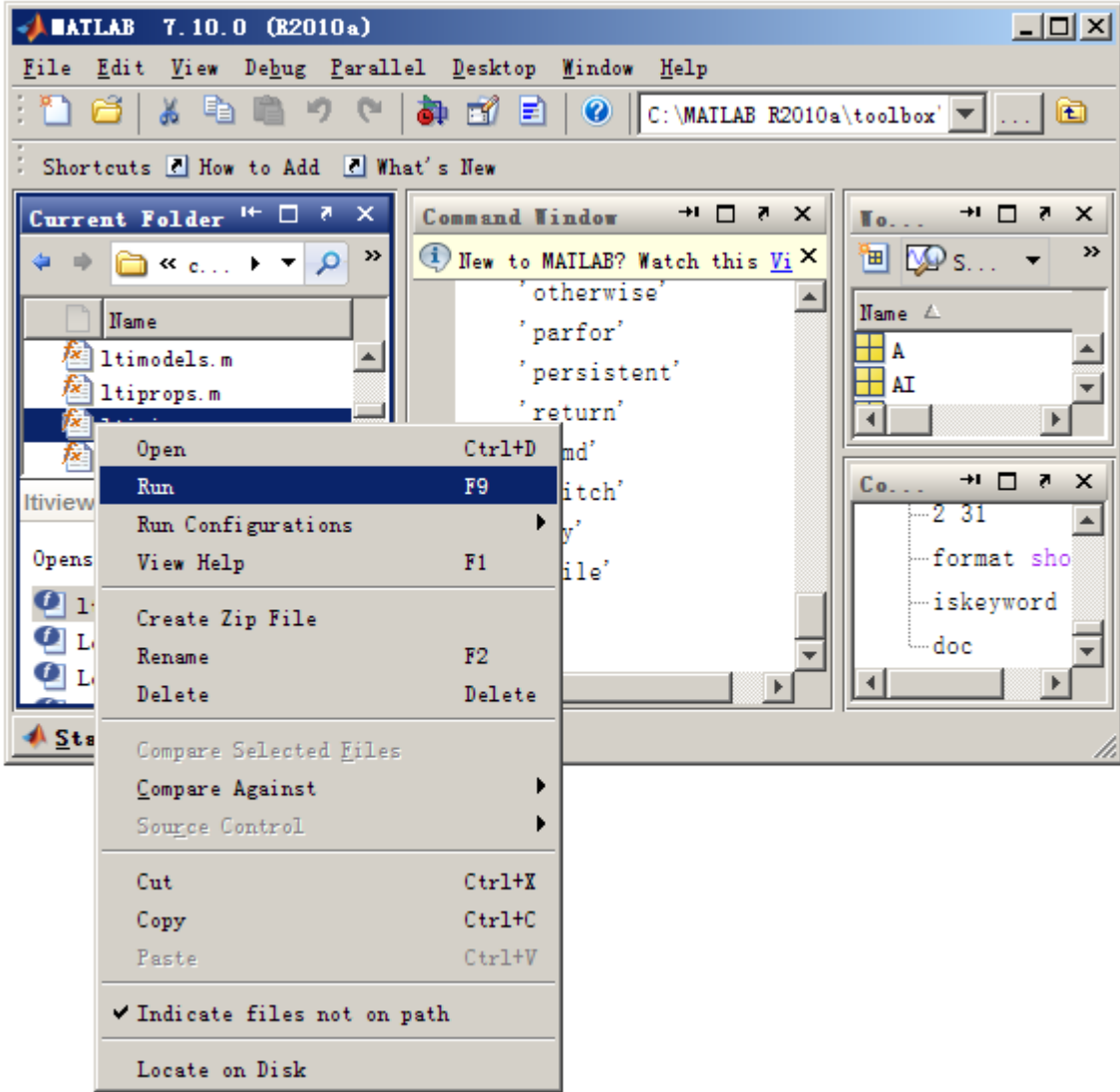


图 1.6-1 当前目录浏览器和适配的弹出菜单

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：C:\Documents and Settings\acer\My Documents\MATLAB 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。
- 若使用 notebook 文档启动 MATLAB 窗口，则当前目录将是 MATLAB 所在的根目录。提醒读者：千万不要把 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) 用户目录

MATLAB R2010a 在安装过程中，会自动生成一个目录 C:\Documents and Settings\acer\My Documents\MATLAB。该目录专供存放用户自己的各类 MATLAB 文件。

假若用户想另建一个工作目录，采用 Windows 规范操作就可实现。

(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”或 MATLAB 为自动开设的“C:\Documents and Settings\acer\My Documents\MATLAB”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

1 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。

2 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。

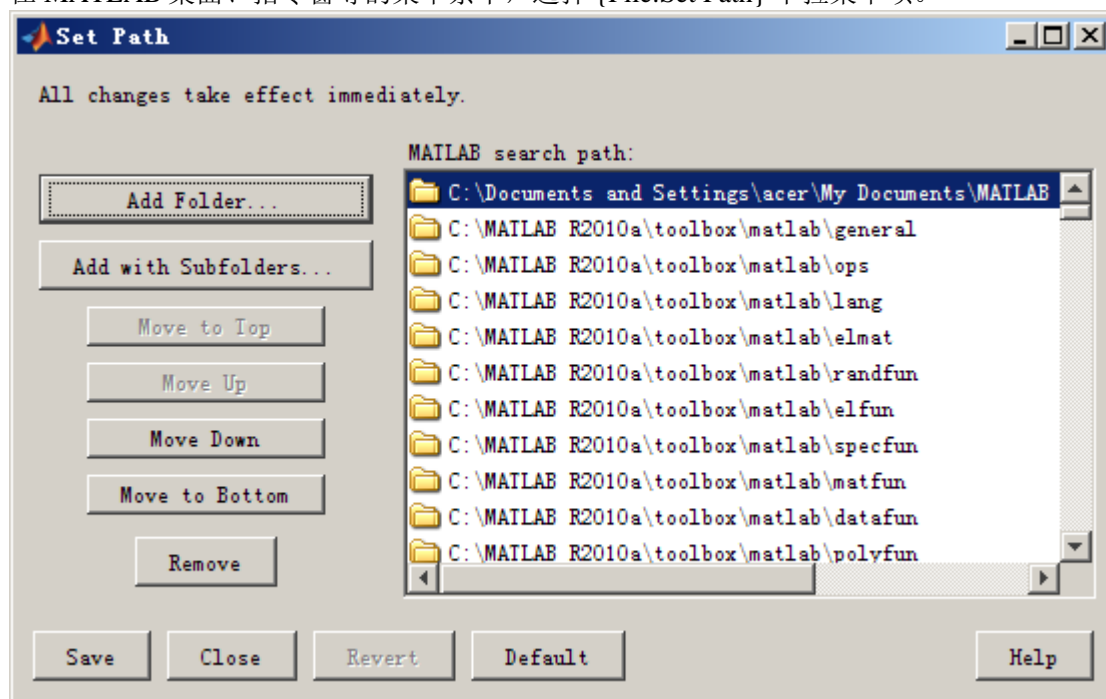


图 1.6-2 路径设置对话框

【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。

3 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，那么在上一环境下用 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 工作空间浏览器和变量编辑器

1.7.1 工作空间浏览器和变量可视化

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。

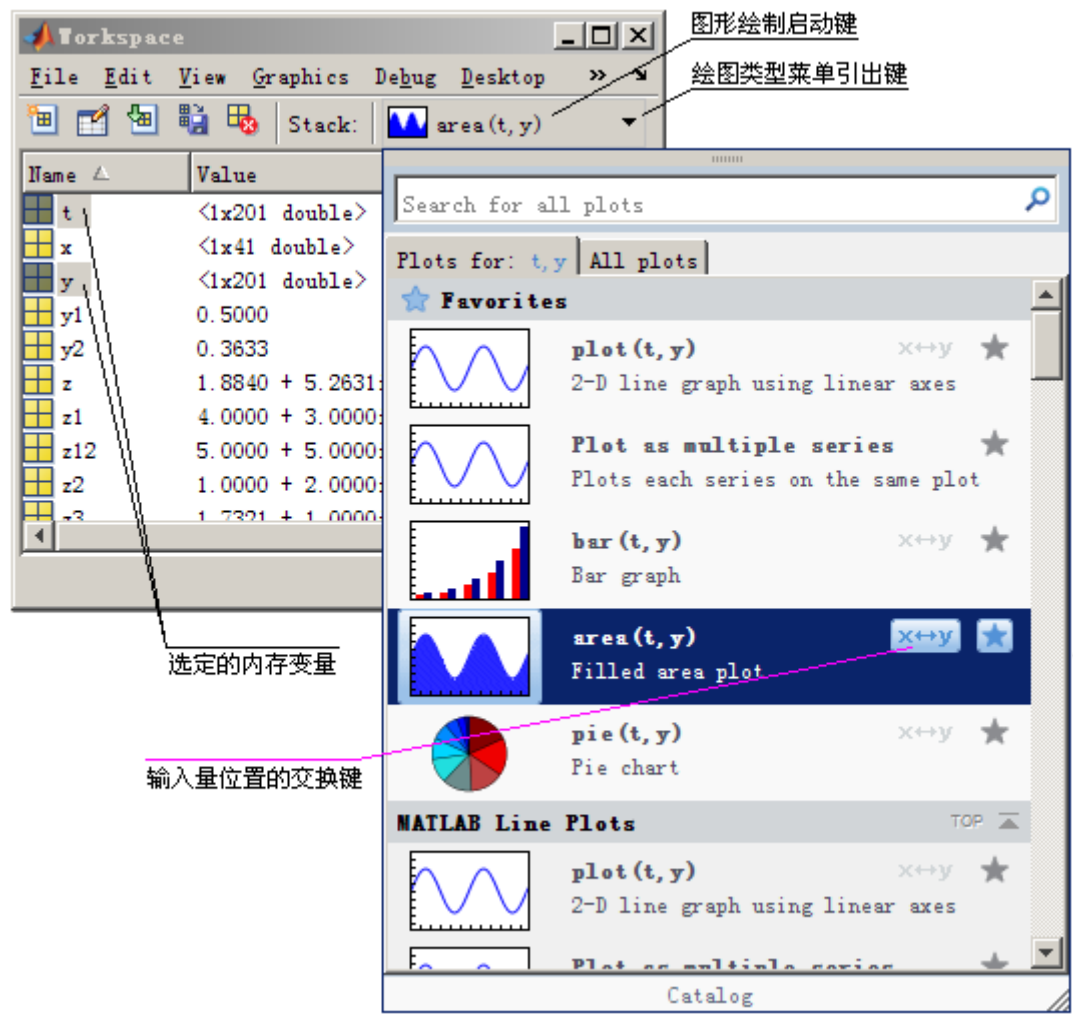








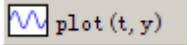
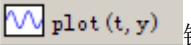




图 1.7-1 工作空间浏览器及“绘图工具”图标的展开

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
 新变量创建	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 Variable Editor 变量编辑器（如图 1.7-3）；在变量编辑器中，向各元素输入数据；最后，对该变量进行重命名。
 变量内容显示	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“Variable Editor”变量编辑器中。（参见第 1.7-3 节）
 向内存 装载文件数据	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。
 把变量 保存进文件	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）

 plot(t,y) 图形绘制启动键	点击  键绘制出选定类型的图形。（详见例 1.7-1）
 绘图类型 菜单引出键	点击  键，引出绘图类型菜单供选择。（详见例 1.7-1）

【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 t 和 y 之间的关系图形。（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) 绘图变量的选定

在“工作空间浏览器”中，用鼠标点亮所需图示的变量 y 和 t 。

(2) 选定绘图的类型

点击[绘图类型菜单引出键]，引出绘图类型菜单。假如需要绘制“填色面图”，则用鼠标点中“填色面图”栏（参见图 1.7-1），即可。

(3) 绘图变量位置的交换

由于在选择变量时，先点选 y ，后选 t ，所以在绘图指令显示出 $\text{area}(y,t)$ 。显然，指令 area 的两个输入量位置不正确。为纠正这种错误，用鼠标点击[输入量位置交换键]，参见图 1.7-1。

(2) 图形的绘制

经过以上操作后，再双击“填色面图”菜单，就绘制出如图 1.7-2 所示的图形。

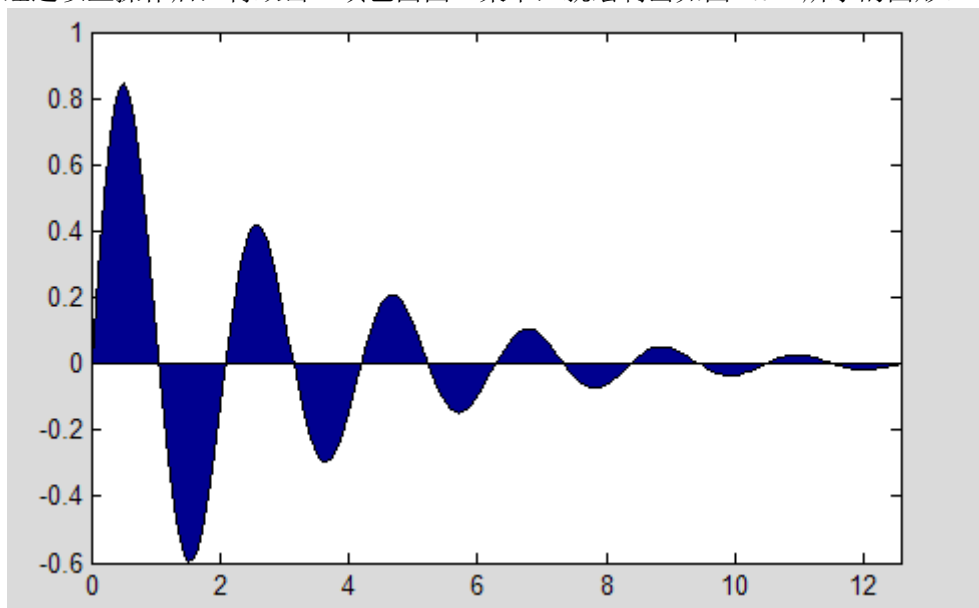


图 1.7-2 数组 (t, y) 表示的曲线

【说明】

- 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。正是出于这种考虑，MATLAB 提供了“图示数据”的多种途径。

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 。

1 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  Attributes

ans           1x1             8  double
t             1x201          1608  double
y             1x201          1608  double      s
```

〔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。

2 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。

3 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 Variable Editor 变量编辑器

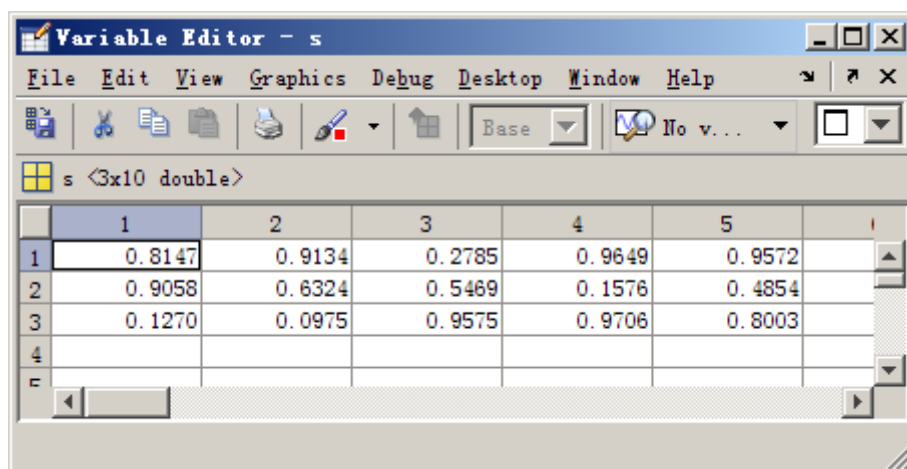



图 1.7-3 变量编辑器


双击工作空间浏览器中的变量图标，将引出如图 1.7-3 所示的变量编辑器 Variable 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。


现在的变量编辑器不但能观察和编辑“双精度”数组，也能观察和编辑“字符串”数组、“胞元”数组、和“构架”数组，而且还能借助“数据链接（Data link）”和“数据刷（Data brush）”与图形窗中的图形相关联（参见第 5.5 节）。

1.7.4 数据文件和变量的存取

1 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。
- 点击工作空间浏览器工具条图标, 或选中弹出菜单中的 {Save As} 项，就弹出 Windows 标准的目录和文件名输入对话框。
- 选定数据文件的保存目录。数据文件应保存在，或用户选定的目录上，或 MATLAB 自动生成的用户工作目录，即 C:\Documents and Settings\acer\My Documents\MATLAB 目录上。注意：假如不有意识地选定目录，那么数据文件将被保存在 MATLAB 的当前目录上。
- 输入数据文件名，如 mydata（注意：.mat 扩展名会自动生成），点击[保存]键，就完成 MAT 数据文件的产生。

2 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}, 引出 Windows 标准的目录和文件选择对话框。
- 在用户希望的目录上，选中 MAT 数据文件（如光盘 For2010a\mfiles 子目录上的 prob_data401.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。

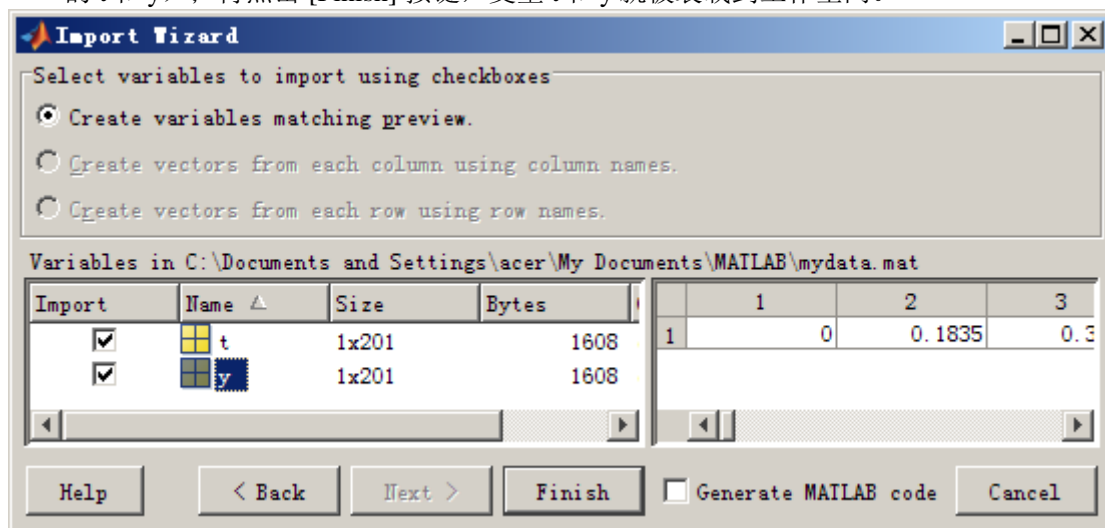


图 1.7-4 向工作空间装载变量的输入向导

3 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。

save FileName

把全部内存变量保存为 FileName.mat 文件

save FileName v1 v2

把变量 v1, v2 保存为 FileName.mat 文件

save FileName v1 v2 -append	把变量 v1, v2 添加到 FileName.mat 文件中
save FileName v1 v2 -ascii	把变量 v1, v2 保存为 FileName 8 位 ASCII 文件
save FileName v1 v2 -ascii -double	把变量 v1, v2 保存为 FileName 16 位 ASCII 文件
load FileName	把 FileName.mat 文件中的全部变量装入内存
load FileName v1 v2	把 FileName.mat 文件中的 v1, v2 变量装入内存
load FileName v1 v2 -ascii	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

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

（2）清空内存，从 saf.mat 向内存装载变量 Z

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

【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

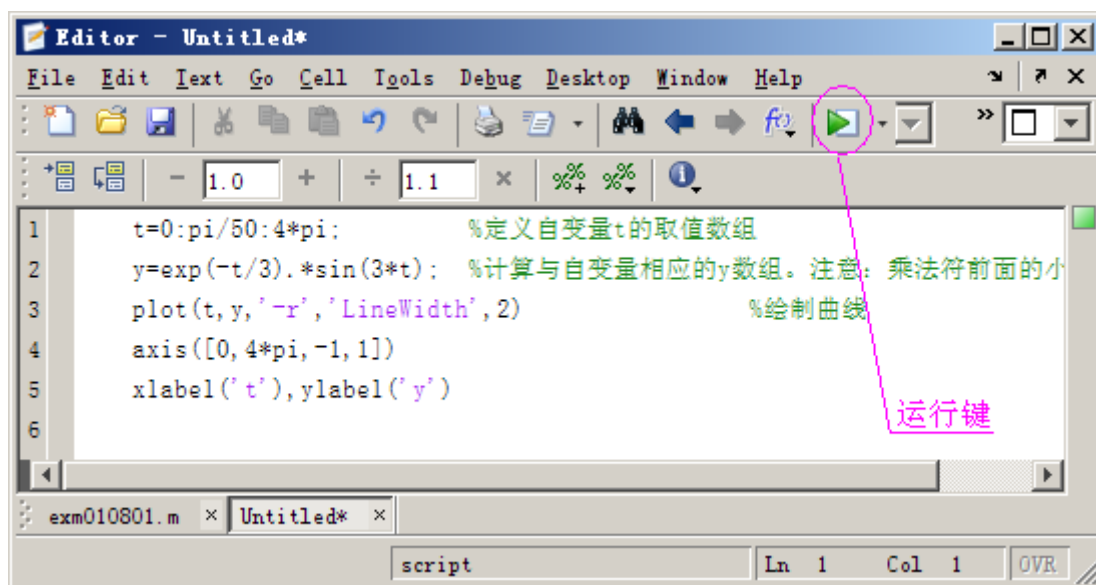




图 1.8-1 M 文件编辑器示意图

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：

- 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，参见图 1.8-2。
- 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。
- 保存文件的操作是 Windows 标准操作。首先，选择 C:\Documents and Settings\acer\My Documents\MATLAB 为文件保存目录，然后以 `exm010801` 为文件名进行保存，于是就得到了 `exm010801.m` 文件。
- 因为 C:\Documents and Settings\acer\My Documents\MATLAB 是 MATLAB 安装时自动生成“在搜索路径上的”目录，所以直接点击 M 文件编辑器上的工具图标  运行键，就可以得到如图 1.3-4 的曲线。

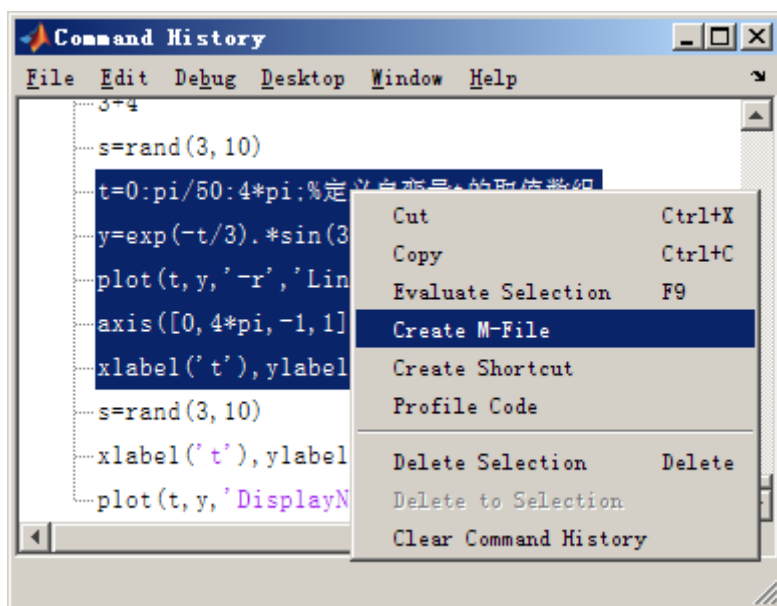


图 1.8-2 利用历史指令创建 M 文件

【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。参见图 1.8-1。

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 帮助体系的三种主要形式

表 1.9-1 MATLAB 的帮助体系

帮助形式	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。	直接从指令窗中，通过 <code>help</code> 指令获得；所有包含在 M 文件之中的帮助注释内容。
帮助导航系统	HTML 形式；系统叙述 MATLAB 规则和用法；适于系统阅读和交叉查阅；最重要的帮助形式。	位于 <code>matlab\help</code> 目录下；通过帮助浏览器获得；HTML 和 XML 文件，物理上独立于 M 文件，是次生性帮助文件；本书重点介绍。
Web 网帮助系统	包括各种 PDF 文件、视频演示文件、各种讨论组等	mathworks 公司网站；MATLAB 操作界面下拉菜单 {Help>Web Resources>}；本书不作介绍。

1.9.2 常用帮助指令

1 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：

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

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

doc ToolboxName 列出指定名称工具包中的所有函数名
doc FunName 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。

2 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。

lookfor KeyWord 对 M 文件 H1 行进行单词条检索
docsearch 对 HTML 子系统进行多词条检索

【说明】

- 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 指令使用格式
 - 格式：**docsearch('Word1 Word2')**，或 **docsearch('Word1 OR Word2')**
搜索是对“每个词条”按“或”逻辑进行。
 - 格式：**docsearch('Word*')**
*是通配符。凡词头为 Word 的词条将都被检索。
 - 格式：**docsearch(' "Word1 Word2" ')**
将对由 Word1 Word2 构成的合成词组进行搜索
- 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。该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 帮助浏览器


1 帮助浏览器的导出

帮助浏览器 (Help Brower) 搜索的资源是 Mathworks 专门创建的 HTML 随“机”帮助系统。它的内容来源于所有 M 文件，但更详细。它的界面友善，交叉查阅尤其方便。这是

用户寻求帮助的最主要资源。

引出如图 1.9-1 所示帮助浏览器的方法有以下几种：

(1) MATLAB Desktop 操作界面上操作法

- 方法一：点击工具条的  图标；
- 方法二：选中下拉菜单项 {Help>MATLAB Help>Product Help}

(2) 指令窗操作法

- 在指令窗中运行 helpbrowser 或 helpdesk。

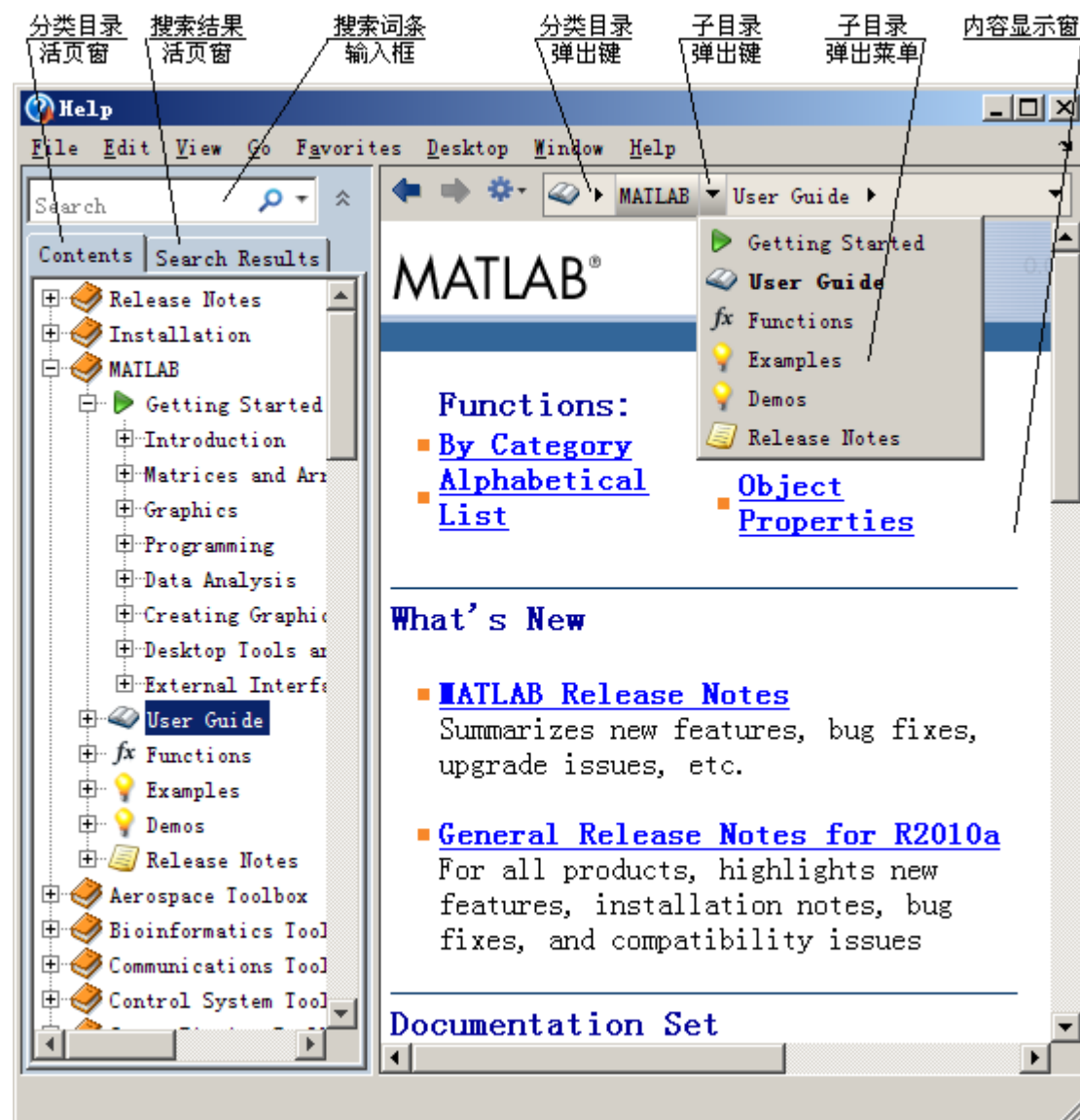


图 1.9-1 默认的帮助浏览器界面

2 帮助浏览器界面简介





(1) 总体外观


图 1.9-1 显示的帮助浏览器界面，的结构如下：






- 左侧检索区：
 - Contents 分类目录活页窗
 - Search Results 检索结果活页窗
 - (Search) 搜索词条输入框


- 右侧显示区
 - 检索内容显示窗
 - 目录弹出工具图标：分类目录弹出图标；子目录弹出图标。

(2) Contents 分类目录活页窗







- 一级目录分三类： M 码类； SIMULINK 类； 外延应用类。
 - 关于 M 码类目录的说明

前三个 “一级目录”分别是：Release Notes 目录，发布新产品、版本升级、老版本的修订、兼容性等信息；Installation 目录，发布 MATLAB 及各工具包的安装和激活信息；MATLAB 目录，介绍 MATLAB 功能、规则、和基本函数指令等帮助信息。

关于 M 码工具包的分类目录。各工具包目录按英文字母表排序；通用性较强的工具有： Optimization Toolbox 优化工具包， Statistics Toolbox 统计工具包， Symbolic Math Toolbox 符号计算工具包。
 - 关于 SIMULINK 的分类目录

第一个目录是 SIMULINK，它介绍 SIMULINK 功能、规则、和基本模块等信息。其他工具库目录按英文字母表排序。
 - 关于 外延应用的分类目录

各工具库都按英文字母表排序
- 二级目录按功能分成五类

■  快速入门	最简捷的入门介绍，新手必读。
■  用户指南	系统叙述该软件包的具体应用规则，及注意事项。
■  库模块使用说明	按字母排序逐块解释库模块的使用要领和相关连接。
■  函数指令使用说明	按字母排序逐条解释函数指令的调用格式。
■  运用实例和演示	算例和演示程序
■  版本说明	说明版本新增、更新内容和兼容状况。

(3) 搜索词条输入框

- 在搜索框里，既可以输入函数指令名（如 inv），又可以输入各种专业词条（如 inverse matrix）。输入确认，按[Enter]键。
- 词条搜索规则
 - 格式：Word1 Word2，或 Word1 OR Word2
搜索是对“每个词条”按“或”逻辑进行。
 - 格式：Word*
*是通配符。凡词头为 Word 的词条将都被检索。
 - 格式："Word1 Word2"
将对由 Word1 Word2 构成的合成词组进行搜索

(4) Search Results 搜索活页窗

假如在搜索框中，输入词组 laplace transform，按[Enter]键进行搜索，那么帮助浏览器将呈现如图 1.9-2 的界面。图中，因为输入词组中的单词被空格分开，所以各单词分别被搜索，并被彩化。

- 左侧搜索结果列表

有如下三种可能的排列方式：

 - Relevance 相关性排列方式 这是默认排列方式。这是按照该段文字与“被搜索词组”相关数量大小排列的。这便于用户找到最集中、详细的帮助信息。
 - Type 二级目录类型排列方式 如果用鼠标点帮助浏览器击左侧的[Type]按键，那么搜索结果将按“二级目录类型（参见图 1.9-2）”排列。这便于用户找到最容易入门的帮助信息。

- Product 产品类型排列方式 如果用鼠标点击帮助浏览器左侧的[Product]按键，那么搜索结果将按产品名称，即工具包名称，字母顺序排列。这便于用户找到希望工具包中的针对性较强的帮助信息。

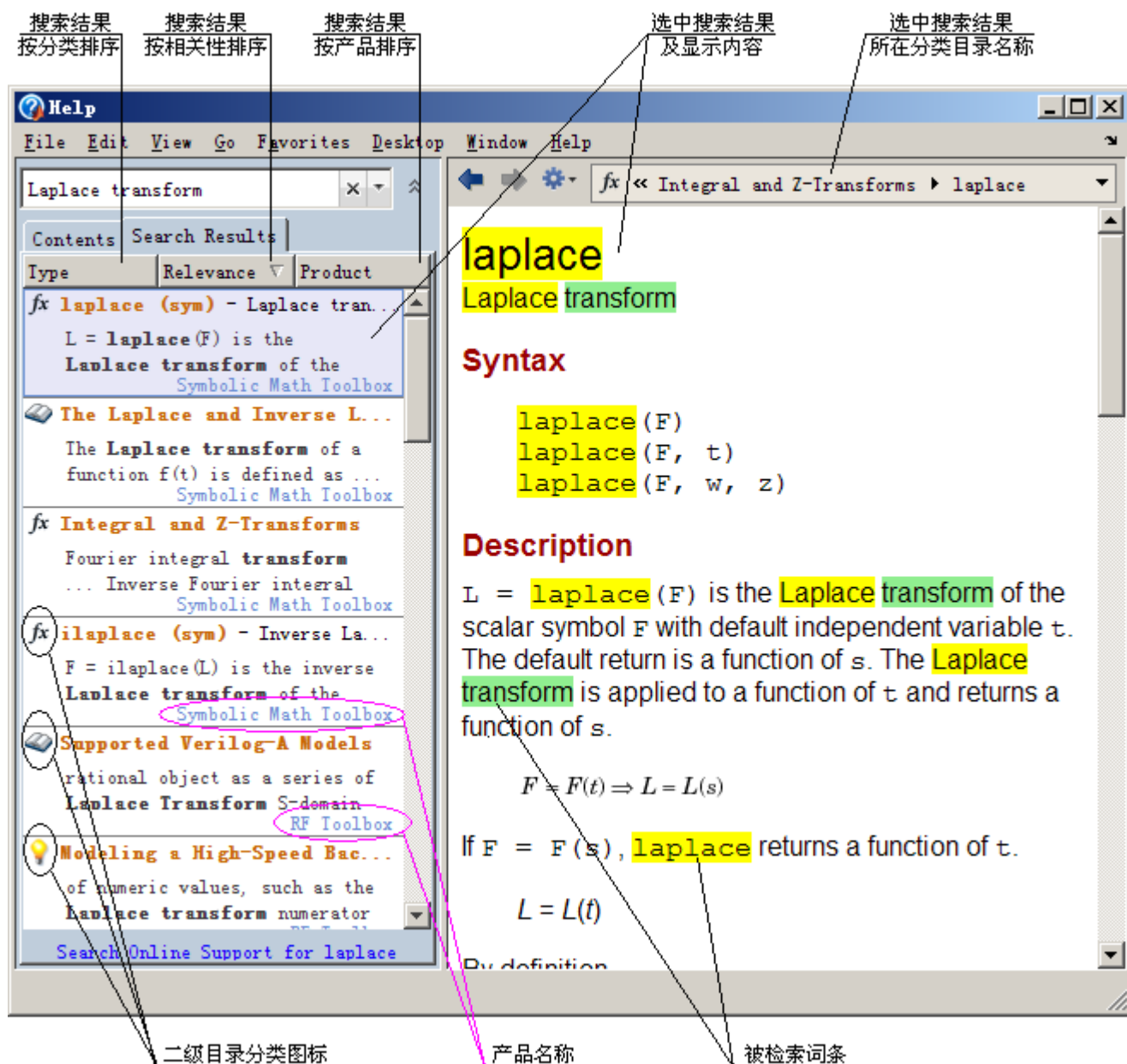



图 1.9-2 按相关性罗列的搜索结果

3 帮助浏览器默认显示的利用

当点击 MATLAB 工作台的工具条图标 , 或选中下拉菜单项 {Help>MATLAB Help>Product Help}, 而引出如图 1.9-1 的帮助浏览器时, 该浏览器右侧默认地显示出清晰、简捷的“通往各种帮助文件的超链接通道”(参看图 1.9-1 的右侧)。具体如下:

- 函数指令和图形对象超链接通道 便于查找指令和图形对象属性
- 版本信息超链接通道 为具有较大量 MATLAB 历史资源的用户而设
- 详细使用说明超链接通道 为希望全面了解某种功能而开设
- 功用演示超链接通道 向用户提供包括视频在内的多种演示帮助
- PDF 文件超链接通道 向用户更适于阅读和打印的帮助文件

- MathWorks 网站资源超链接通道 供用户查阅 MATLAB 各种资源或问题解答

(1) 函数指令和图形对象超链接通道

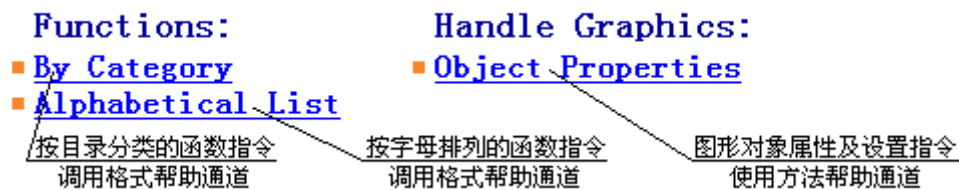


图 1.9-3 函数及图柄帮助通道区

- By Category 按目录分类排列的函数指令帮助通道
适用场合： 不知道具体指令名，但知道指令应该具有的功能；
需要了解、比较功能相近的指令；
挑选最适用的指令。
通道特点： 功能清楚，可比较选择；但查询速度较慢。
- Alphabetical List 按字母排列的函数指令帮助通道
适用场合： 指令名清楚已知，但准确的调用格式模糊；
指令名清楚已知，了解多种调用格式。
通道特点： 查询速度最快；但缺少比较。
- Object Properties 图形对象属性及舌质指令帮助通道
适用场合： 需对 MATLAB 所绘图形进行个性化“底层”操作的场合；
了解、选用“图形对象属性”及“操作指令”调用格式的场合。
通道特点： 层次清楚；属性分列清晰；超链接交互查阅方便。

(2) 版本信息超链接通道

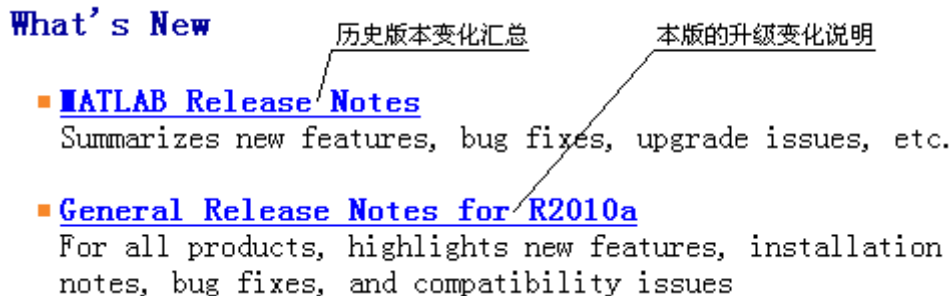


图 1.9-4 版本信息帮助通道区

- MATLAB Release Notes 历史版本变化汇总
适用场合： MATLAB 新手不必看；
需要了解 MATLAB 历史演进；
较多、较重要的 MATLAB 历史版本编写的文件资源启用前；
MATLAB 历史版本写成文件运行产生不明原因错误的场合。
通道特点： 对各版变化、补丁修正描述清晰。
- General Release Notes for R2010a 本版本升级变化说明
使用场合： MATLAB 新手可浏览新版本对环境的要求；
利用 MATLAB 设计较大型软件库的用户；
MATLAB 历史版本写成文件运行产生不明原因错误的场合。

(3) 详细使用说明超链接通道

Documentation Set

- ▶ [Getting Started](#) ————— [快速入门](#)
- ▶ [User Guides](#) ————— [用户指南](#)
- [Getting Help](#) ————— [帮助指南](#)
Provides instructions for using help functions, the Help browser, and other resources
- [Examples in Documentation](#) ————— [分功能详解](#)
Lists major examples in the MATLAB documentation
- [Programming Tips](#) ————— [编程技巧](#)
Provides helpful techniques and shortcuts for programming in MATLAB

图 1.9-5 帮助文档通道区

- Getting Started 快速入门
适用场合： 新手应先读和必读，克服生疏感；
感受新版本工作环境。
通道特点： 所包含的网络接口通道，可直接观看 Mathworks 制作的入门视频；
所涉材料浅显易懂。
- User Guider 用户指南
适用场合： 用户指南是所有帮助内容中最重要的部分；
建议每个用户快速浏览阅读，以形成对 MATLAB 编程环境的宏观了解；
遇问题时重点研读，以深入理解使程序真实反映数学模型的注意事项。
通道特点： 是数学模型和程序模型之间的桥梁。
- Getting Help 帮助指南
适用场合：
- Examples in Documentation 分功能详解
适用场合： 建议每个用户浏览比较细的分功能列表；
学习、掌握某功能，如怎样合并矩阵、怎样消除数据中的确定性趋势等。
通道特点： 既包含基本编程技能，又包含最实用的数学处理方法；
每种功能单独分列，适于片段学习。
- Programming Tips 编程技巧
适用场合： 建议用户快速浏览编程技巧；
适于快速了解 MATLAB 功能和使用要领。
通道特点： 包含的各条目短小精干。

(4) 功用演示超链接通道

Product Demos

- [MATLAB Demos](#) ————— [功能演示](#)
Presents a collection of demos that you can run from the Help browser to help you learn the product

图 1.9-6 功能演示通道区

- MATLAB Demos 功能演示

适用场合： 感受 MATLAB 界面各图标、菜单的功能，以及交互操作手法；
了解、学习典型的 M 码文件的指令运用和编写技巧；
了解、学习 GUI 图形用户接口的功能和编写技巧。

通道特点： Video 视频演示需网络支持；
GUI 资源、及部分 M 码资源所提供的帮助具有独特性；
资源既包括“入门引导”型（初学者适宜），又有“学科专业”型（科研人员适宜），还有“编程技巧”型（对 MATLAB 较熟悉者适宜）。

(5) PDF 文件超链接通道

Printable (PDF) Documentation on the Web

■ Printable versions of the MATLAB documentation and related papers on the Web PDF格式帮助文件

图 1.9-7 PDF 文档通道区

- Printable versions PDF 格式帮助文件

适用场合： 需要系统阅读、学习 User Guider 用户指南。

通道特点： 需要得到网络支持；
该文件编框架和内容与超文本形式的“User Guider 用户指南”相似。

(6) MathWorks 网站资源超链接通道



图 1.9-8 MathWorks 网站资源通道区

适用场合： 了解 MATLAB 产品、第三方相关产品信息；
向 MathWorks 公司进行技术咨询。

通道特点： 需网络支持。

习题 1

1. 数字 1.5e2, 1.5e3 中的哪个与 1500 相同？
2. 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
3. 在 MATLAB 环境中，比 1 大的最小数是多少？
4. 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

5. 指令 clear, clf, clc 各有什么用处?
6. 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。”

7. 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

8. 试为例 1.3-5 编写一个解题用的 M 脚本文件?

第 2 章 符号计算

2.1 符号对象和符号表达式

2.1.1 符号对象的创建和衍生

1 生成符号对象的基本规则

2 符号数字和符号常数

【例 2.1-1】

```
a=pi+sqrt(5)
sa=sym('pi+sqrt(5)')
Ca=class(a)
Csa=class(sa)
vpa(sa-a)
a =
    5.3777
sa =
pi + 5^(1/2)
Ca =
double
Csa =
sym
ans =
0.0000000000000000013822375841085200048593542564188
```

3 基本符号变量

4 自由符号变量

【例 2.1-2】。

(1)

```
syms u v w z a5
f=sym('3');
Eq=sin(f)*u*z^2+v*z+f*w-a5;
```

(2)

```
symvar(Eq)
ans =
[ a5, u, v, w, z]
```

```
symvar(Eq,100)
ans =
[ w, z, v, u, a5]
```

```
symvar(Eq,1)
ans =
w
```

(3)

```
result_1=solve(Eq)
result_1 =
```

```
a5/3 - (v*z)/3 - (u*sin(3)*z^2)/3
```

(4)

```
result_2=solve(Eq,z)
```

```
result_2 =
```

```
-(v - (v^2 + 4*a5*u*sin(3) - 12*u*w*sin(3))^(1/2))/(2*u*sin(3))  
-(v + (v^2 + 4*a5*u*sin(3) - 12*u*w*sin(3))^(1/2))/(2*u*sin(3))
```

【例 2.1-3】

(1)

```
syms a b x X Y
```

```
k=sym('3');
```

```
z=sym('c*sqrt(d)+y*sin(t)');
```

```
EXPR=a*z*X+(b*x^2+k)*Y;
```

(2)

```
symvar(EXPR)
```

```
ans =
```

```
[ X, Y, a, b, c, d, t, x, y]
```

(3)

```
symvar(EXPR,10)
```

```
ans =
```

```
[ x, y, t, d, c, b, a, X, Y]
```

(4)

```
symvar(EXPR,1)
```

```
ans =
```

```
x
```

(5)

```
symvar(EXPR,3)
```

```
ans =
```

```
[ x, y, t]
```

(6)

```
E3=sym('a*sqrt(theta)')
```

```
??? Error using ==> sym.sym>sym.sym/scalarsym at 382
```

```
Error: argument must be of 'Type::Arithmetical' [sqrt]
```

```
Error in ==> sym.sym>sym.sym/char2sym at 337
```

```
Scell = scalarsym(x);
```

```
Error in ==> sym.sym>sym.sym/symchar at 182
```

```
Scell = char2sym(x);
```

```
Error in ==> sym.sym>sym.sym at 120
```

```
S = cell2sym(S,symchar(x,a,nargin));
```

```
E4=sym('a*sqrt(theta123)')
```

```
E4 =
```

```
a*theta123^(1/2)
```

```
E5=sym('a*sqrt(theta*t)')
```

```
E5 =
```

```
a*(t*theta)^(1/2)
```


【例 2.1-4】

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

2.1.2 符号计算中的算符

2.1.3 符号计算中的函数指令

2.1.4 符号对象的识别

【例 2.1-5】。

(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      Attributes

  Mc        1x9        18    char
  Mn        2x2        32    double
  Ms        2x2        60    sym

```

2.1.5 符号运算机理和变量假设

- 1 符号运算的工作机理
- 2 对符号变量的限定性假设
- 3 清除变量和撤销假设

【例 2.1-6】

```

(1)
syms x clear
f=x^3+4.75*x+2.5;
rf=solve(f,x)
rf =
    -1/2
    1/4 - (79^(1/2)*i)/4
    (79^(1/2)*i)/4 + 1/4

evalin(symengine,'getprop(x)')
ans =
C_

(2)
syms x real
rfr=solve(f,x)
rfr =
-1/2

evalin(symengine,'getprop(x)')
ans =
R_

(3)
clear x
syms x
g=x^2+x+5;
rg=solve(g,x)
Warning: Explicit solution could not be found.
> In solve at 98
rg =
[ empty sym ]

```

```
(4)
syms x clear
rg=solve(g,x)
rg =
- (19^(1/2)*i)/2 - 1/2
(19^(1/2)*i)/2 - 1/2
```

【例 2.1-7】

```
(1)
clear all
reset(symengine)
Da=1.2;Dw=1/3;
syms sa sw sx sy sz
syms A B positive
syms C real

(2)
whos
```

Name	Size	Bytes	Class	Attributes
A	1x1	58	sym	
B	1x1	58	sym	
C	1x1	58	sym	
Da	1x1	8	double	
Dw	1x1	8	double	
sa	1x1	60	sym	
sw	1x1	60	sym	
sx	1x1	60	sym	
sy	1x1	60	sym	
sz	1x1	60	sym	

```
(3)
syms
```

'A'	'B'	'C'	'sa'	'sw'	'sx'	'sy'	'sz'
-----	-----	-----	------	------	------	------	------

```
(4)
evalin(symengine,'anames(Properties)')
ans =
{A, B, C}
```

```
(5)
clear A
syms
```

'B'	'C'	'ans'	'sa'	'sw'	'sx'	'sy'	'sz'
-----	-----	-------	------	------	------	------	------

```
evalin(symengine,'anames(Properties)')
ans =
{A, B, C}
```

```
(6)

syms B clear
syms
```

'B'	'C'	'ans'	'sa'	'sw'	'sx'	'sy'	'sz'
-----	-----	-------	------	------	------	------	------

```
evalin(symengine,'anames(Properties)')
ans =
{A, C}
```

2.1.6 符号帮助体系

【例 2.1-8】

(1)

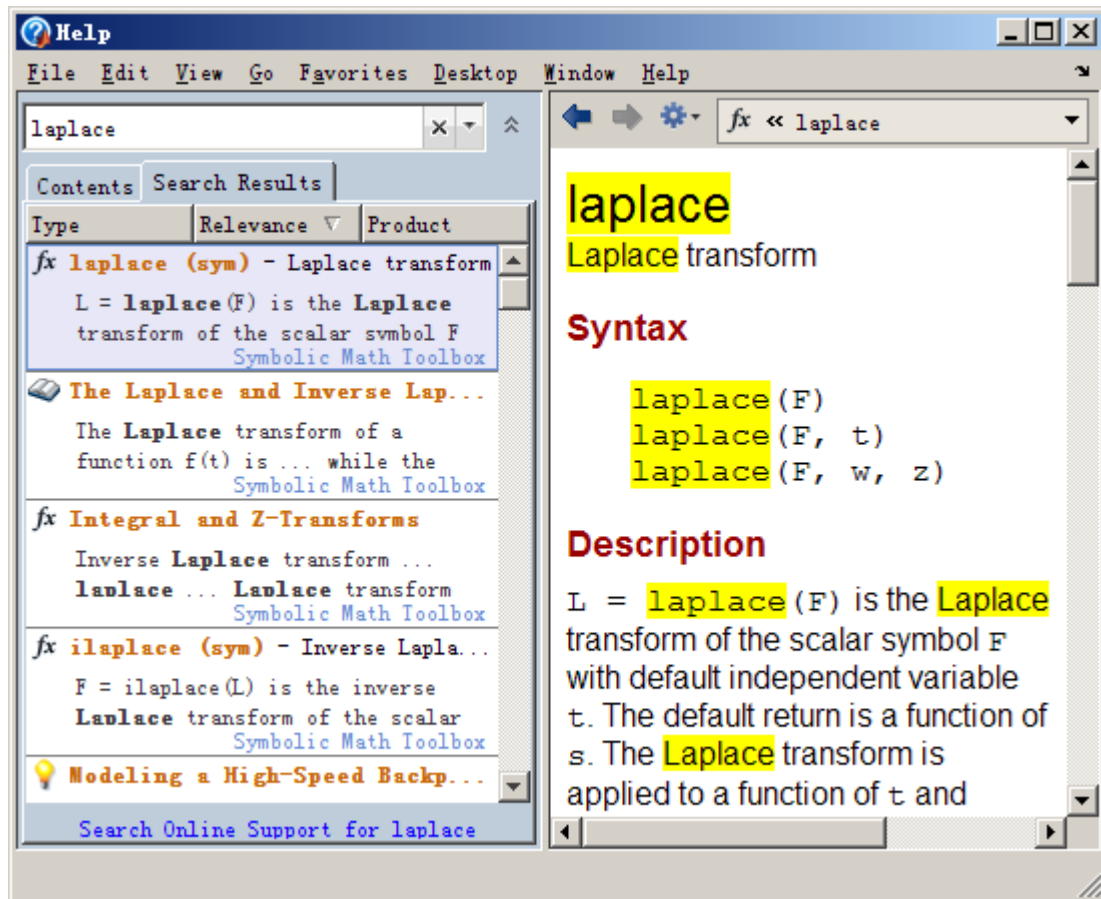


图 2.1-1

(2)

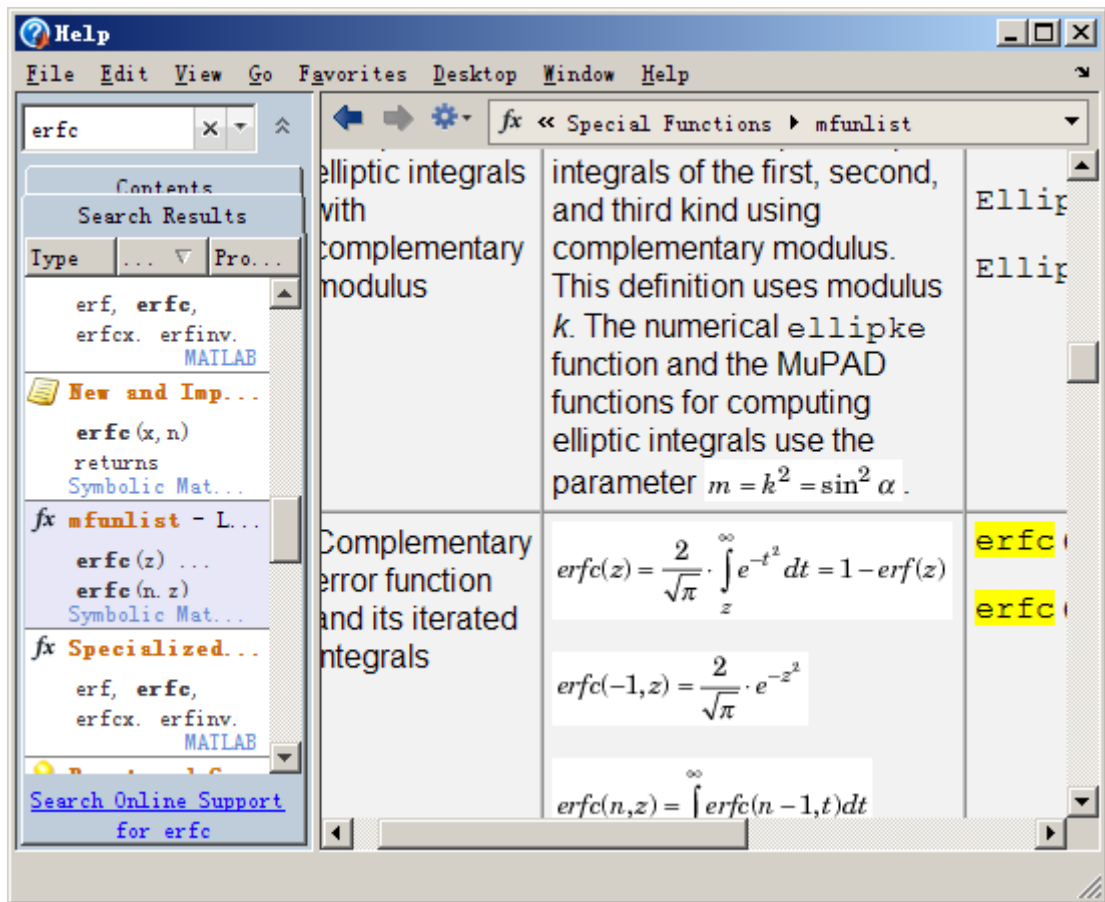


图 2.1-2

- (3)

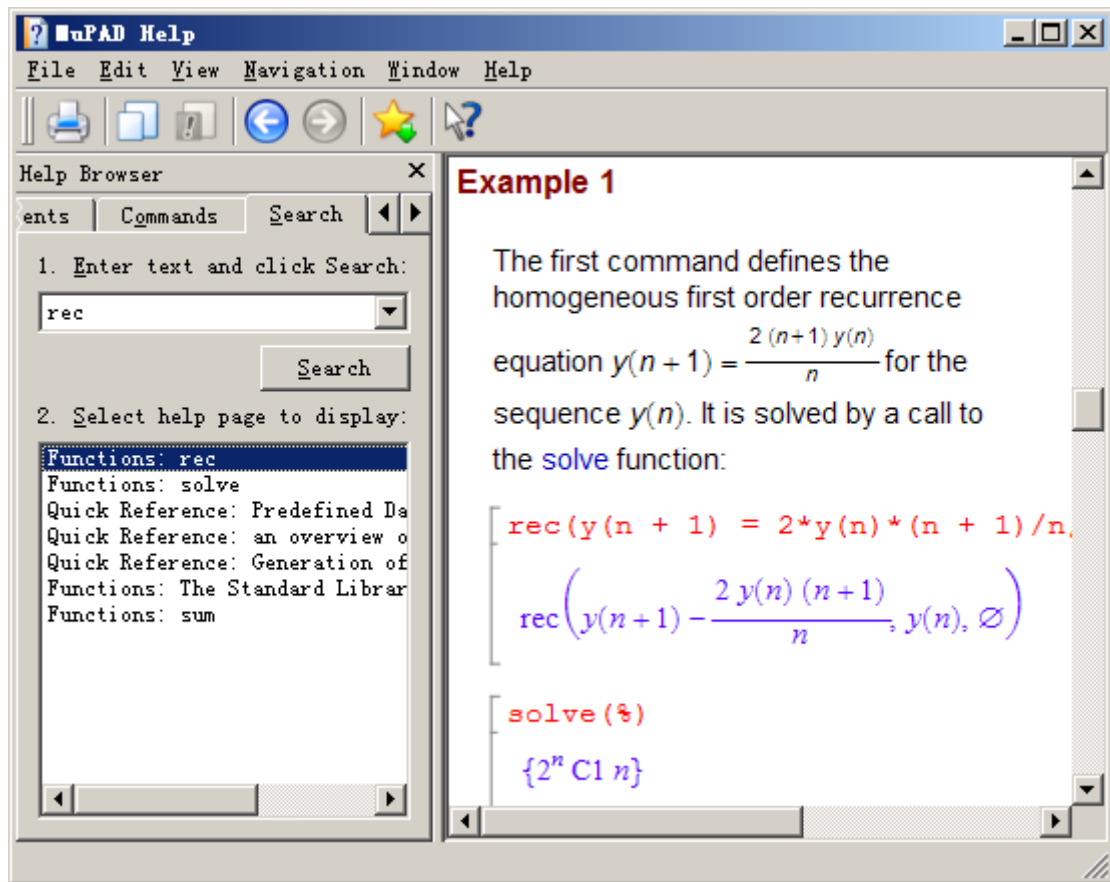


图 2.1-3

2.2 符号数字及表达式的操作

2.2.1 双精度数字与符号数字之间的转换

1 双精度数字向符号数字的转换

2 符号数字向双精度数字转换

2.2.2 符号数字的任意精度表达形式

【例 2.2-1】

(1)

```
reset(symengine)
```

```
sa=sym('1/3+sqrt(2)')
```

```
sa =  
2^(1/2) + 1/3
```

(2)

```
digits
```

```
Digits = 32
```

```
format long
a=1/3+sqrt(2)
sa_Plus_a=vpa(sa+a,20)
sa_Minus_a=vpa(sa-a,20)
```

```
a =
    1.747546895706428
sa_Plus_a =
    3.4950937914128567869
sa_Minus_a =
   -0.000000000000000022658064826339973669
```

```
(3)
sa32=vpa(sa)           %                               <8>
digits(48)             %                               <9>
sa5=vpa(sa,5)          %                               <10>
sa48=vpa(sa)           %                               <11>
sa32 =
1.747546895706428382135022057543
sa5 =
1.7475
sa48 =
1.74754689570642838213502205754303141190300520871
```

```
(4)
a
a2=a+1/10^12
a_16=vpa(a,16);a2_16=vpa(a2,16);
a_12=vpa(a,12);a2_12=vpa(a2,12);
a =
    1.747546895706428
a2 =
    1.747546895707429

a_16==a2_16
ans =
    0

a_12==a2_12
ans =
    0
```

2.2.3 符号表达式的基本操作

【例 2.2-2】

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


2.2.4 表达式中的置换操作

1 公因子法简化表达

【例 2.2-3】。

```
(1)
clear
A=sym('[a b;c d]')
[V,D]=eig(A)
A =
[ a, b]
[ c, d]
V =
[(a/2+d/2-a^2-2*a*d+d^2+4*b*c)^(1/2)/2)/c-d/c, (a/2+d/2+(a^2-2*a*d+
d^2+4*b*c)^(1/2)/2)/c-d/c]
[
1,
1]
D =
[a/2 + d/2 - (a^2 - 2*a*d + d^2 + 4*b*c)^(1/2)/2, 0]
[
0, a/2 + d/2 + (a^2 - 2*a*d + d^2 + 4*b*c)^(1/2)/2]

(2)
subexpr([V;D]) % <4>
who %
sigma =
(a^2 - 2*a*d + d^2 + 4*b*c)^(1/2)
ans =
[(a/2 + d/2 - sigma/2)/c - d/c, (a/2 + d/2 + sigma/2)/c - d/c]
[
1,
1]
[
a/2 + d/2 - sigma/2,
0]
[
0,
a/2 + d/2 + sigma/2]

Your variables are:

A      D      V      ans      sigma

(3)
Dw=subexpr(D,'w')
w =
(a^2 - 2*a*d + d^2 + 4*b*c)^(1/2)
Dw =
[ a/2 + d/2 - w/2, 0]
[
0, a/2 + d/2 + w/2]

(4)
[RVD,w]=subexpr([V;D],'w') % <7>
RVD =
[(a/2 + d/2 - w/2)/c - d/c, (a/2 + d/2 + w/2)/c - d/c]
[
1,
1]
[
a/2 + d/2 - w/2,
0]
[
0,
a/2 + d/2 + w/2]
w =
(a^2 - 2*a*d + d^2 + 4*b*c)^(1/2)
```

2 通用置换指令

【例 2.2-4】

```
(1)
clear
```

```
syms a b x;
f=a*sin(x)+b
f =
b + a*sin(x)
```

```
(2)
f1=subs(f,sin(x),'log(y)')
class(f1)
f1 =
b + a*log(y)
ans =
sym
```

```
(3)
f2=subs(f,a,3.11)
class(f2)
f2 =
b + (311*sin(x))/100
ans =
sym
```

```
(4)
f3=subs(f,{a,b,x},{2,5,sym('pi/3')})
class(f3)
f3 =
3^(1/2) + 5
ans =
sym
```

```
(5)
format
format compact
f4=subs(f,{a,b,x},{2,5,pi/3})
class(f4)
f4 =
    6.7321
ans =
double
```

```
(6)
f5=subs(f,x,0:pi/2:pi)
class(f5)
f5 =
[ b, a + b, b]
ans =
sym
```

```
(7)
t=0:pi/10:2*pi;
f6=subs(f,{a,b,x},{2,3,t})
plot(t,f6)
f6 =
Columns 1 through 8
    3.0000    3.6180    4.1756    4.6180    4.9021    5.0000    4.9021
    4.6180
Columns 9 through 16
    4.1756    3.6180    3.0000    2.3820    1.8244    1.3820    1.0979
    1.0000
Columns 17 through 21
    1.0979    1.3820    1.8244    2.3820    3.0000
```

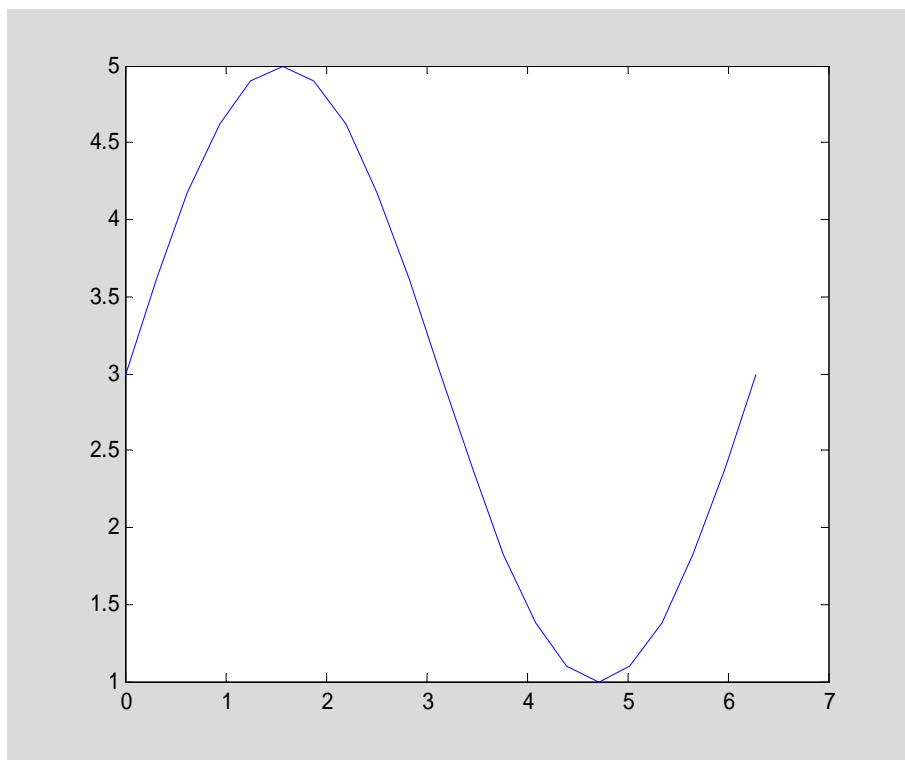


图 2.2-1

```
(8)
k=(0.5:0.1:1)';
f6=subs(subs(f,{a,b},{k,2}),x,t);
size(f6)
plot(t,f6)
ans =
    6    21
```

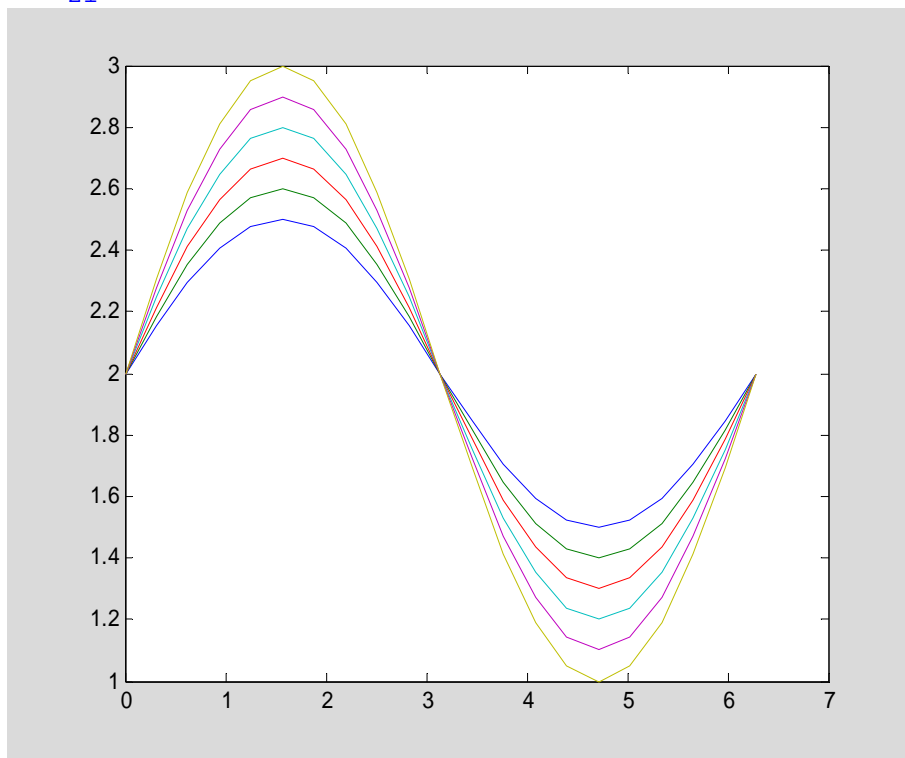


图 2.2-2

2.3 符号微积分

2.3.1 极限和导数的符号计算

【例 2.3-1】

```
syms t x k
s=sin(k*t)/(k*t);
f=(1-1/x)^(k*x);
Lsk=limit(s,0)
Ls1=subs(Lsk,k,1)
Lf=limit(f,x,inf)
Lf1=vpa(subs(Lf,k,sym('-1')),48)
Lsk =
1
Ls1 =
1
Lf =
1/exp(k)
Lf1 =
2.7182818284590452353602874713526624977572470937
```

【例 2.3-2】

```
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】

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

【例 2.3-4】

```
(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) % <5>
df_p0=limit((subs(f_p,x,d)-subs(f_p,x,0))/d,d,0)% <6>
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) % <8>
df_n0=limit((subs(f_n,x,0)-subs(f_n,x,-d))/d,d,0) % <9>
df_n =
-cos(x)
df_n0 =
-1

(3)
f=sin(abs(x));
dfdx=diff(f,x) % <11>
dfdx0=subs(dfdx,x,0) % <12>
dfdx =
cos(abs(x))*sign(x)
dfdx0 =
0

(4)
clf
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',3) % <14>
plot(xn,subs(df_n,x,xn),'--r','LineWidth',3)
plot(xp,subs(df_p,x,xp),' :r','LineWidth',3)
legend(char(f),char(df_n),char(df_p),'Location','NorthEast') % <17>
grid on
xlabel('x')
hold off

```

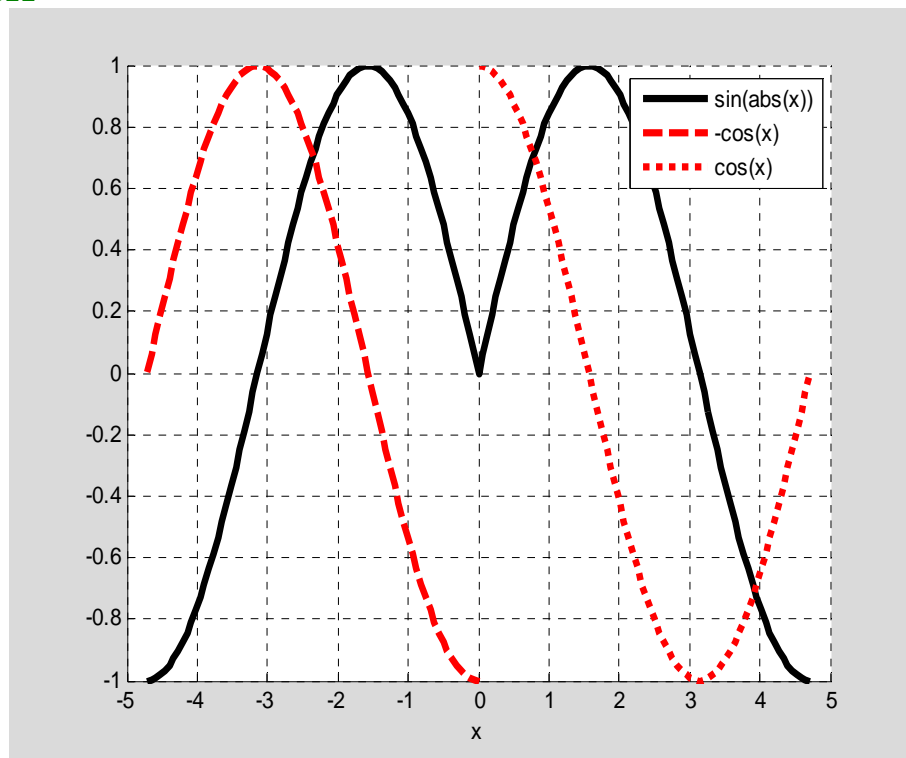


图 2.3-1

【例 2.3-5】

```

(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)))*(cos(y(x))*diff(y(x), x) + 1) = cos(y(x))*diff(y(x),
x)

```

```

(2)
dgdxdx1=subs(dgdxdx,'diff(y(x),x)','dydx') % <5>
dgdxdx1 =
-sin(x + sin(y(x)))*(dydx*cos(y(x)) + 1) = dydx*cos(y(x))

```

```

(3)
dydx=solve(dgdxdx1,'dydx')
dydx =
-sin(x + sin(y(x)))/(cos(y(x)) + cos(y(x))*sin(x + sin(y(x))))

```

【例 2.3-6】

```

(1)
syms x
r=taylor(x*exp(x),9,x,0) % <2>
pretty(r) %
r =
x^8/5040 + x^7/720 + x^6/120 + x^5/24 + x^4/6 + x^3/2 + x^2 + x

```

$$\frac{x^8}{5040} + \frac{x^7}{720} + \frac{x^6}{120} + \frac{x^5}{24} + \frac{x^4}{6} + \frac{x^3}{2} + x^2 + x$$

```

(2)
R = evalin(symengine,'series(x*exp(x),x=0,8)') % <4>
pretty(R) %
R =
x + x^2 + x^3/2 + x^4/6 + x^5/24 + x^6/120 + x^7/720 + x^8/5040 + O(x^9)

```

$$x + x^2 + \frac{x^3}{2} + \frac{x^4}{6} + \frac{x^5}{24} + \frac{x^6}{120} + \frac{x^7}{720} + \frac{x^8}{5040} + O(x^9)$$

【例 2.3-7】

```

TL1=evalin(symengine,'mtaylor(sin(x^2+y),[x,y],8)')
TL1 =
(x^6*y^2)/12 - x^6/6 + (x^4*y^3)/12 - (x^4*y)/2 - (x^2*y^6)/720 +
(x^2*y^4)/24 - (x^2*y^2)/2 + x^2 - y^7/5040 + y^5/120 - y^3/6 + y
class(TL1)
ans =
sym

```

2.3.2 序列/级数的符号求和

【例 2.3-8】。

(1)

```
syms n k
f1=1/(k*(k+1));
s1=symsum(f1,k,1,n)
s1 =
1 - 1/(n + 1)
```

(2)

```
f2=x^(2*k-1)/(2*k-1);
s2=symsum(f2,k,1,inf)
s2 =
piecewise([abs(x) < 1, atanh(x)])
```

(3)

```
f3=[1/(2*k-1)^2,(-1)^k/k];
s3=symsum(f3,k,1,inf)
s3 =
[ pi^2/8, -log(2)]
```

2.3.3 符号积分

【例 2.3-9】

```
clear
syms a b x
f1=x*log(x)
s1=int(f1,x)
s1=simple(s1) % <5>
f1 =
x*log(x)
s1 =
(x^2*(log(x) - 1/2))/2
s1 =
x^2*(log(x)/2 - 1/4)
```

【例 2.3-10】

```
f2=[a*x,b*x^2;1/x,sin(x)]
disp(' ')
disp('The integral of f is')
pretty(int(f2))
f2 =
[ a*x, b*x^2]
[ 1/x, sin(x)]
```

The integral of f is

$$\begin{bmatrix} \frac{a x^2}{2}, & \frac{b x^3}{3} \\ \log(x), & -\cos(x) \end{bmatrix}$$

【例 2.3-11】

```
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)
Warning: Explicit integral could not be found.
F2 =
(14912*2^(1/4))/4641 - (6072064*2^(1/2))/348075 + (64*2^(3/4))/225 +
1610027357/6563700
VF2 =
224.921535733311431597907100328046757677071376012
```

【例 2.3-12】。

(1)

```
syms a r theta phi
r=a*theta;
x=r*cos(theta);
y=r*sin(theta);
dLdth=sqrt(diff(x,theta)^2+diff(y,theta)^2);
warning off % <5>
L=simple(int(dLdth,theta,0,phi)) % <6>
L =
(phi*(a^2*phi^2 + a^2)^(1/2) + log(phi + (phi^2 +
1)^(1/2)))*(a^2)^(1/2))/2
```

(2)

```
L_2pi=subs(L,[a,phi],sym('[1,2*pi]'))
L_2pi_vpa=vpa(L_2pi)
L_2pi =
log(2*pi + (4*pi^2 + 1)^(1/2))/2 + pi*(4*pi^2 + 1)^(1/2)
L_2pi_vpa =
21.2562941482090988007025122725661088234709310476
```

(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]); % <14>
set(h1,'Color','r','LineWidth',5)%
title(' ') % <16>
legend('螺线长度-幅角曲线','阿基米德螺线')
hold off %
```

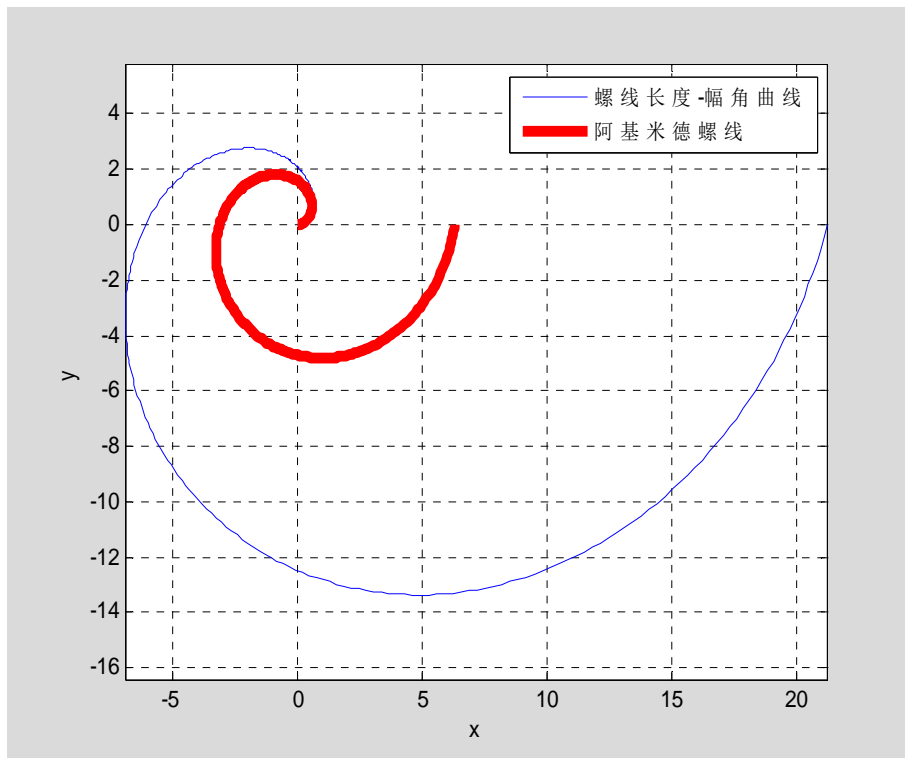


图 2.3-2 阿基米德螺线（粗红）和螺线长度函数（细蓝）

2.4 微分方程的符号解法

2.4.1 符号解法和数值解法的互补作用

2.4.2 求微分方程符号解的一般指令

2.4.3 微分方程符号解示例

【例 2.4-1】

```
clear all % <1>
S=dsolve('Dx=y,Dy=-x') %
disp(' ')
disp(['微分方程的解',blanks(2),'x',blanks(22),'y'])
disp([S.x,S.y])
S =
    x: [1x1 sym]
    y: [1x1 sym]
```

```
微分方程的解    x                                y
[ C2*cos(t) + C1*sin(t), C1*cos(t) - C2*sin(t)]
```

【例 2.4-2】

(1)

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

```

        x^2/4
C3*x - C3^2

(2)
clf,hold on %
h1=ezplot(y(1),[-6,6,-4,8],1); % <4>
set(h1,'Color','r','LineWidth',5)
for k=-2:0.5:2 % <6>
    y2=subs(y(2),'C3',k); %<7>
    ezplot(y2,[-6,6,-4,8],1)
end % <9>
hold off %
box on
legend('奇解','通解','Location','Best')
ylabel('y')
title(['\fontsize{14}微分方程',' (y'')^2 - xy'' + y = 0 ','的解'])

```

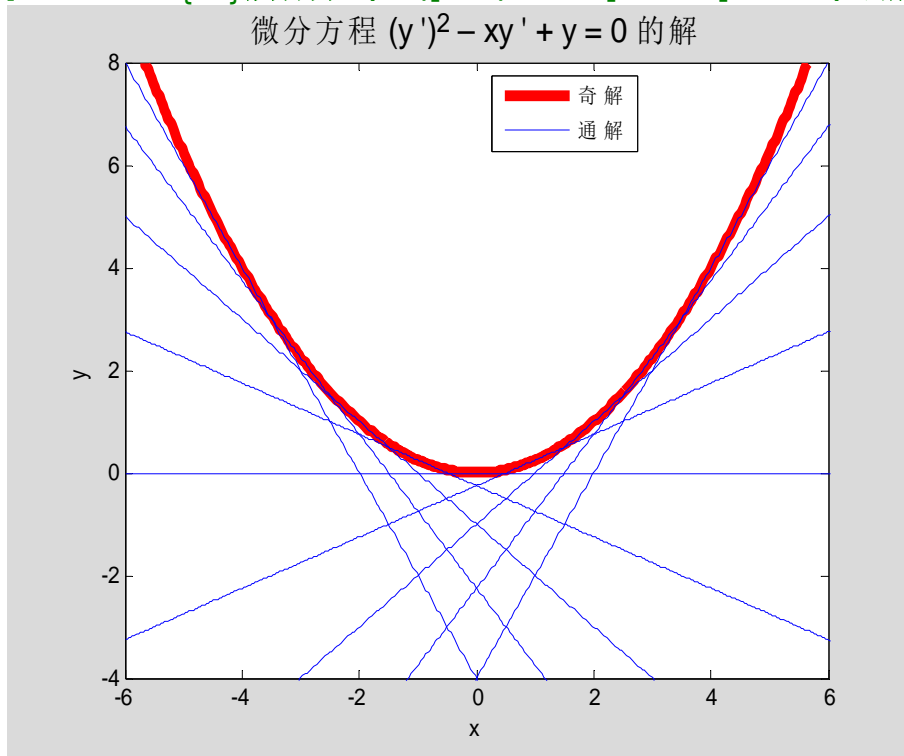


图 2.4-1 通解和奇解曲线

【例 2.4-3】

```

(1)
y=dsolve('x*D2y-3*Dy=x^2','y(1)=0,y(5)=0','x')
y =
(31*x^4)/468 - x^3/3 + 125/468

(2)
xn=-1:6;
yn=subs(y,'x',xn)
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
yn =

```

0.6667 0.2671 0 -1.3397 -3.3675 -4.1090 0.0000
14.1132

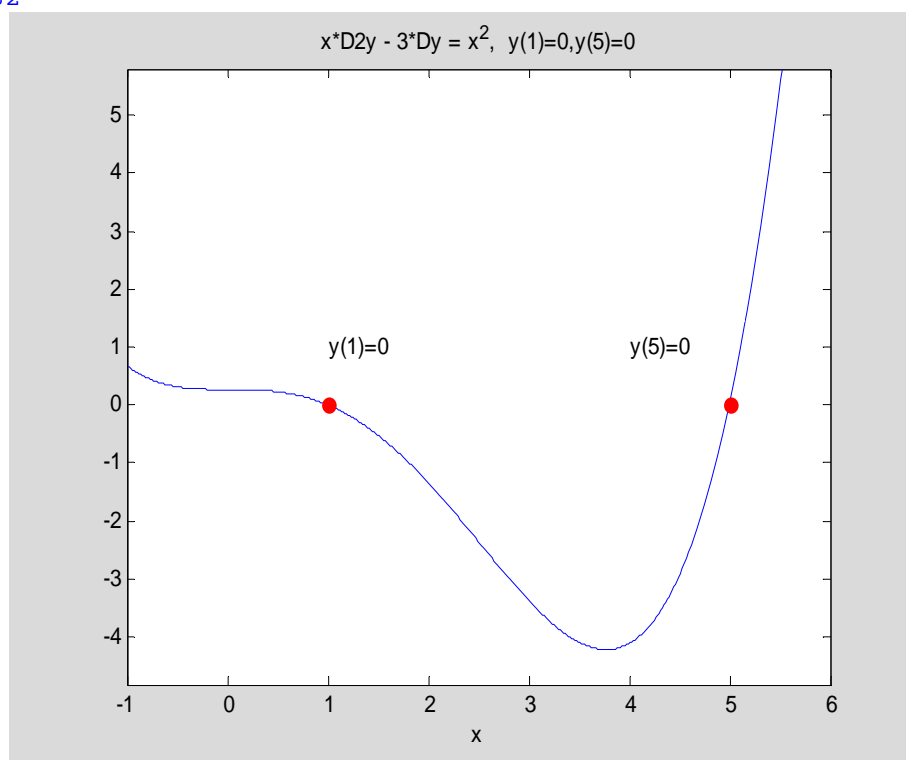


图 2.4-2 两点边值问题的解曲线

2.5 符号变换和符号卷积

2.5.1 Fourier 变换及其反变换

【例 2.5-1】

(1)

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

(2)

```
Ut=ifourier(UT,w,t)
SUt=simple(Ut) % <5>
Ut =
(pi + pi*(2*heaviside(t) - 1))/(2*pi)
SUt =
heaviside(t)
```

(3)

```
t=-2:0.01:2;
ut=heaviside(t);
kk=find(t==0); % <8>
plot(t(kk),ut(kk),'r','MarkerSize',30)
hold on
ut(kk)=NaN; % <10>
plot(t,ut,'-r','LineWidth',3)
```

```

plot([t(kk),t(kk)],[ut(kk-1),ut(kk+1)],'or','MarkerSize',10)
%
hold off
grid on
axis([-2,2,-0.2,1.2])
xlabel('\fontsize{14}t'),ylabel('\fontsize{14}ut')
title('\fontsize{14}Heaviside(t)')

```

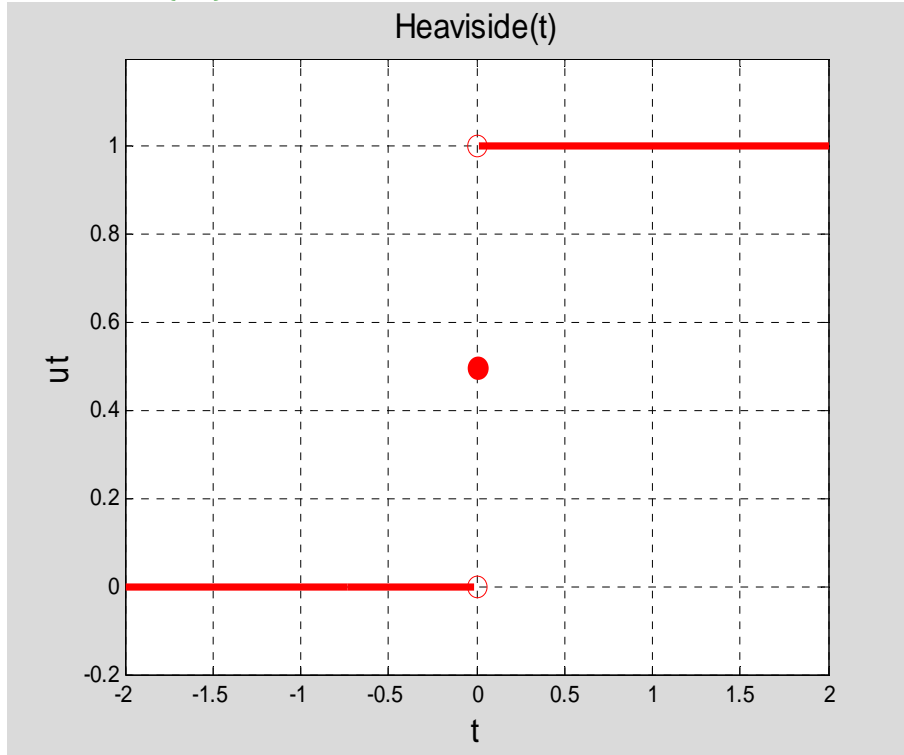


图 2.5-1 Heaviside(t) 定义的单位阶跃函数

【例 2.5-2】

(1)

```

syms A t w tao
yt=A*(heaviside(t+tao/2)-heaviside(t-tao/2)); %
Yw=fourier(yt,t,w) %
Yw_fy=simplify(Yw) %
Yw_fy_e=simple(Yw_fy) %
Yw =
A*((1/exp((tao*w*i)/2))*(- pi*dirac(-w) + i/w) - exp((tao*w*i)/2)*(-
pi*dirac(-w) + i/w))
Yw_fy =
(2*A*sin((tao*w)/2))/w
Yw_fy_e =
(2*A*sin((tao*w)/2))/w

```

(2)

```

Yt=ifourier(Yw_fy_e,w,t) %
Yt_e=simple(Yt) % <7>
Yt =
(A*transform::fourier(sin((tao*w)/2)/w, w, t))/pi
Yt_e =
-A*(heaviside(t - tao/2) - heaviside(t + 1/2*tao))

```

(3)

```
t3=3;
```

```

tn=-3:0.1:3;
yt13=subs(yt,{A,tao},{1,t3})
yt13n=subs(yt13,'t',tn);
kk=find(tn==-t3/2|tn==t3/2); % <13>
plot(tn(kk),yt13n(kk),'.r','MarkerSize',30)%
yt13n(kk)=NaN; % <15>
hold on
plot(tn,yt13n,'-r','LineWidth',3)
hold off
grid on
axis([-3,3,-0.5,1.5])
yt13 =
heaviside(t + 3/2) - heaviside(t - 3/2)

```

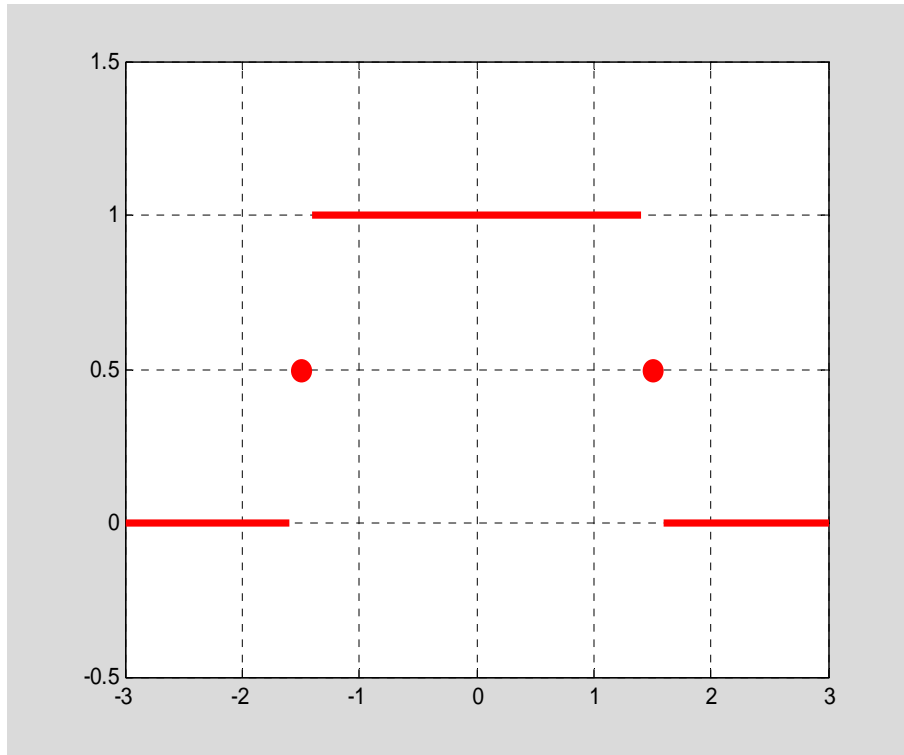


图 2.5-2 由 Heaviside(t) 构造的矩形波

(4)

```

Yw13=subs(Yw_fy_e,{A,tao},{1,t3});
subplot(2,1,1),ezplot(Yw13),grid on
subplot(2,1,2),ezplot(abs(Yw13)),grid on

```

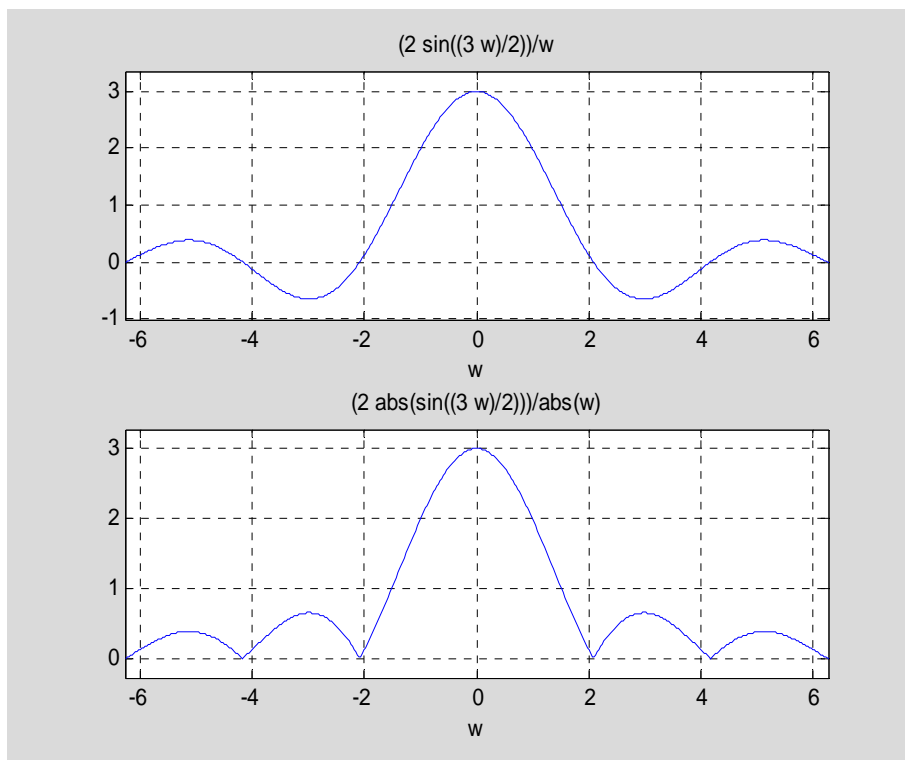


图 2.5-3 矩形脉冲的频率曲线和幅度频谱

【例 2.5-3】

(1)

```
clear
syms t x w
ft=exp(-(t-x))*heaviside(t-x);    % <3>
gt=exp(-(t-x));                    % <4>
```

(2)

```
F1=simple(fourier(ft,t,w))    %
G1=simple(fourier(gt,t,w))    %
F1 =
(1/exp(w*x*i))/(1 + w*i)
G1 =
transform::fourier(exp(x - t), t, -w)
```

(3)

```
F2=simple(fourier(ft,t))    %
F3=simple(fourier(ft))    %
F2 =
-exp(t^2*i)/(- 1 + t*i)
F3 =
-(1/exp(t*w*i))/(- 1 + w*i)
```

2.5.2 Laplace 变换及其反变换

【例 2.5-4】

(1)

```
syms t s a b
```



```

f1=exp(-a*t)*sin(b*t) % <2>
F1=laplace(f1,t,s)
f1 =
sin(b*t)/exp(a*t)
F1 =
b/((a + s)^2 + b^2)

(2)
sym a clear % <4>
f2=heaviside(t-a)
F2=laplace(f2,t,s) %
ans =
a
f2 =
heaviside(t - a)
F2 =
laplace(heaviside(t - a), t, s)

syms a positive % <7>
F3=laplace(f2) %
F3 =
1/(s*exp(a*s))

(4)
f4=dirac(t-b); %
F4=laplace(f4,t,s) %
F4 =
piecewise([b < 0, 0], [0 <= b, 1/exp(b*s)])

f5=dirac(t-a); % <11>
F5=laplace(f5,t,s) %
ft_F5=ilaplace(F5,s,t) %
F5 =
1/exp(a*s)
ft_F5 =
dirac(a - t)

(5)
n=sym('n','clear'); % <14>
F6=laplace(t^n,t,s) %
F6 =
piecewise([-1 < Re(n), gamma(n + 1)/s^(n + 1)])

n=sym('n','positive') % <16>
F6=laplace(t^n,t,s) % <17>
n =
n
F6 =
gamma(n + 1)/s^(n + 1)

```

2.5.3 Z 变换及其反变换

【例 2.5-5】

```

(1)
clear
syms n z clear % <2>
gn=6*(1-(1/2)^n) %
G=simple(ztrans(gn,n,z)); %
pretty(G) %

```

```

gn =
6 - 6*(1/2)^n

      6 z
-----
      2
2 z  - 3 z + 1

(2)
syms n w T z clear          % <6>
fwn=sin(w*n*T);             %
FW=ztrans(fwn,n,z);         %
pretty(FW),disp(' ')
inv_FW=iztrans(FW,z,n)      %

      z sin(T w)
-----
      2
z  - 2 cos(T w) z + 1

inv_FW =
sin(T*n*w)

(3)
syms n z clear              % <11>
f1=1;
F1=ztrans(f1,n,z);
pretty(F1)
inv_F1=iztrans(F1,z,n)

      z
-----
      z - 1
inv_F1 =
1

(4)
clear
syms n z clear              % <16>
delta=sym('kroneckerDelta(n, 0)'); % <17>
KD=ztrans(delta,n,z)
inv_KD=iztrans(KD)
KD =
1
inv_KD =
kroneckerDelta(n, 0)

(5)
syms n z clear              % <20>
k=sym('k','positive');      % <21>
fd=sym('f(n)*kroneckerDelta(n-k, 0)');
FD=ztrans(fd,n,z)
inv_FD=iztrans(FD,z,n)
FD =
piecewise([k in Z_, f(k)/z^k], [Otherwise, 0])
inv_FD =
piecewise([k in Z_, f(k)*kroneckerDelta(k - n, 0)], [Otherwise, 0])

(6)
syms a z n clear
GZ=exp(-a/z);               %

```

```
gn=iztrans(GZ,z,n) % <28>
gn =
(-a)^n/factorial(n)
```

2.5.4 符号卷积

【例 2.5-6】

```
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 =
-(1/exp(t) - 1/exp(t/T))/(T - 1)
```

【例 2.5-7】

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

2.6 符号矩阵分析和代数方程解

2.6.1 符号矩阵分析

【例 2.6-1】

(1)

```
syms a11 a12 a21 a22
A=[a11,a12;a21,a22]
DA=det(A)
IA=inv(A)

A =
[ a11, a12]
[ a21, a22]
DA =
a11*a22 - a12*a21
IA =
[ a22/(a11*a22 - a12*a21), -a12/(a11*a22 - a12*a21)]
[ -a21/(a11*a22 - a12*a21), a11/(a11*a22 - a12*a21)]
```

(2)

```
EA=subexpr(eig(A),'D')
D =
(a11^2 - 2*a11*a22 + a22^2 + 4*a12*a21)^(1/2)
EA =
a11/2 + a22/2 - D/2
a11/2 + a22/2 + D/2
```

【例 2.6-2】

(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 =
[ 3^(1/2)/2,      1/2]
[      1/2, 3^(1/2)/2]
GA =
[ (3^(1/2)*cos(t))/2 - sin(t)/2, cos(t)/2 - (3^(1/2)*sin(t))/2]
[ cos(t)/2 + (3^(1/2)*sin(t))/2, sin(t)/2 + (3^(1/2)*cos(t))/2]

(2)
clf
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')
Lstr=['旋转前的 v1','旋转前的 v2','旋转后的 u1','旋转后的 u2']; % <16>
legend(Lstr,'Location','South') % <17>
hold off
grid on

```

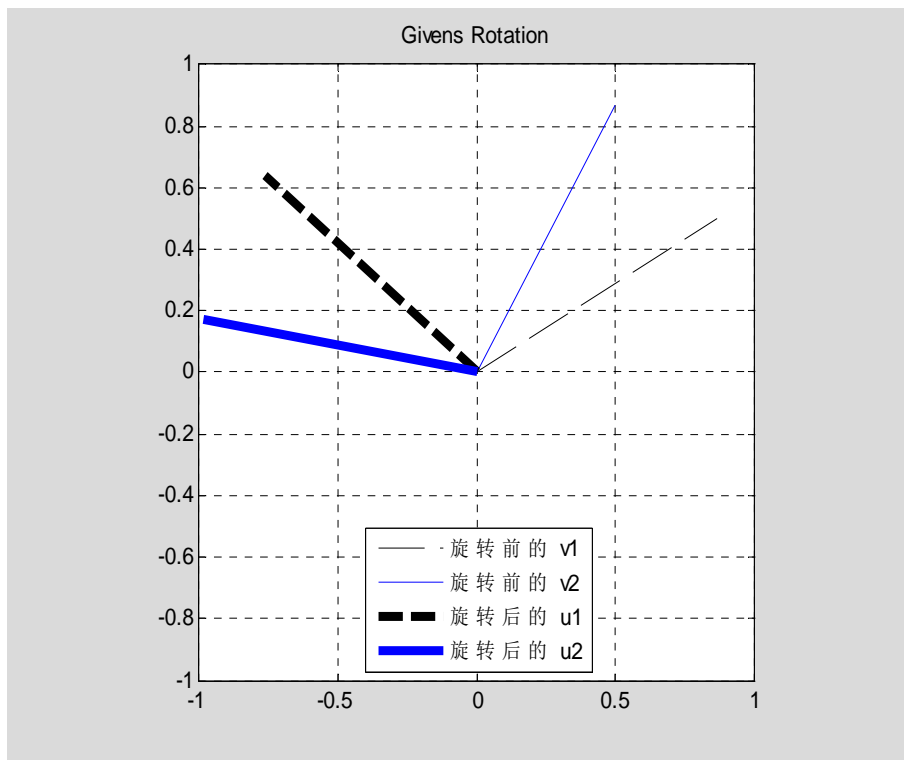


图 5.6-1 Givens 旋转的几何意义

2.6.2 线性方程组的符号解

【例 2.6-3】

(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 %

```

```

X1 =
1
8
8
9

(2)
eq1=sym('d+n/2+p/2-q'); % <4>
eq2=sym('d+n-p+q-10'); %
eq3=sym('d-n/4-p+q'); %
eq4=sym('-8*d-n+p+q-1'); %
S=solve(eq1,eq2,eq3,eq4,'d','n','p','q'); % <8>
disp([' d',' n',' p',' q'])
disp([S.d,S.n,S.p,S.q]) % <10>
      d n p q
[ 1, 8, 8, 9]

```

2.6.3 一般代数方程组的解

【例 2.6-4】

```

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

```

【例 2.6-5】

```

syms d n p q
eq1=d+n/2+p/2-q;eq2=n+d+q-p-10;eq3=q+d-n/4-p; % <2>
S=solve(eq1,eq2,eq3,d,n,p,q); % <3>
disp([' S.d',' S.n',' S.p',' S.q'])
disp([S.d,S.n,S.p,S.q])
      S.d S.n S.p S.q
[ z/3 - 2, 8, (4*z)/3 - 4, z]

```

【例 2.6-6】

```

clear all,syms x;
s=solve('(x+2)^x=2','x')
xs=(s(1)+2)^s(1) % 验算 <3>
s =
matrix([[0.69829942170241042826920133106081]])
xs =
2.0

```

2.7 代数状态方程求符号传递函数

2.7.1 结构框图的代数状态方程解法

【例 2.7-1】

(1)

(2)

(2.7-2)

(3)

(2.7-4)

(4)

```
syms G1 G2 G3 G4 H1 H2 H3
A=[ 0, 0, 0, 0, 0, 0, -G1;
    G2, 0, 0, 0, 0, 0, -G2, 0;
    0, G3, 0, 0, G3, 0, 0, 0;
    0, 0, G4, 0, 0, 0, 0, 0;
    0, 0, 0, H2, 0, 0, 0, 0;
    0, 0, 0, H1, 0, 0, 0, 0;
    0, 0, 0, H3, 0, 0, 0, 0];
b=[ G1; 0; 0; 0; 0; 0; 0; 0];
c=[ 0, 0, 0, 1, 0, 0, 0, 0];
Y2Ua=c*((eye(size(A))-A)\b); %利用“左除”取代“求逆”，计算传递函数
disp([blanks(5),'传递函数 Y2Ua 为'])
pretty(Y2Ua)
```

传递函数 Y2Ua 为

$$\frac{G1 \ G2 \ G3 \ G4}{G2 \ G3 \ G4 \ H1 - G3 \ G4 \ H2 + G1 \ G2 \ G3 \ G4 \ H3 + 1}$$

2.7.2 信号流图的代数状态方程解法

【例 2.7-2】

(1)

(2.7-5)

此

(2.7-6)

(2)

(3)

```
syms G1 G2 G3 G4 H1 H2 H3
A=[ 0, 0, 0, 0, -H3;
    G1, 0, 0, 0, -H1;
    0, G2, 0, 0, H2;
    0, 0, G3, 0, 0;
    0, 0, 0, G4, 0];
b=[ 1; 0; 0; 0; 0];
c=[ 0, 0, 0, 0, 1];
Y2Ub=c*((eye(size(A))-A)\b); %
disp([blanks(5),'传递函数 Y2Ub 为'])
pretty(Y2Ub)
```

传递函数 Y2Ub 为

$$\frac{G1 \ G2 \ G3 \ G4}{G2 \ G3 \ G4 \ H1 - G3 \ G4 \ H2 + G1 \ G2 \ G3 \ G4 \ H3 + 1}$$

(4)

```
syms s
```

```

Sblock={100/(s+10),1/(s+1),(s+1)/(s^2+4*s+4),(s+1)/(s+6),(2*s+12)/(s+
1),(s+1)/(s+2),1}; % <9>
ww=subs(Y2Ub,{G1,G2,G3,G4,H1,H2,H3},Sblock); % <10>
Y2Uc=simple(ww);
[NN,DD]=numden(Y2Uc); %
NN=expand(NN); %
disp('参数具体化的传递函数 Y2Uc 为')
pretty(NN/DD)
参数具体化的传递函数 Y2Uc 为

```

$$\frac{100 s^2 + 300 s + 200}{s^5 + 21 s^4 + 157 s^3 + 663 s^2 + 1301 s + 910}$$

●

2.8 符号计算结果的可视化

2.8.1 直接可视化符号表达式

1 单独立变量符号函数的可视化

【例 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') %
y =
(2*cos((3^(1/2)*t)/2))/(3*exp(t/2))
s =
1/3-(2*cos((3^1/2)*t)/2)/2-3^(1/2)*sin((3^(1/2)*t)/2))/(3*exp(t/2
))

```

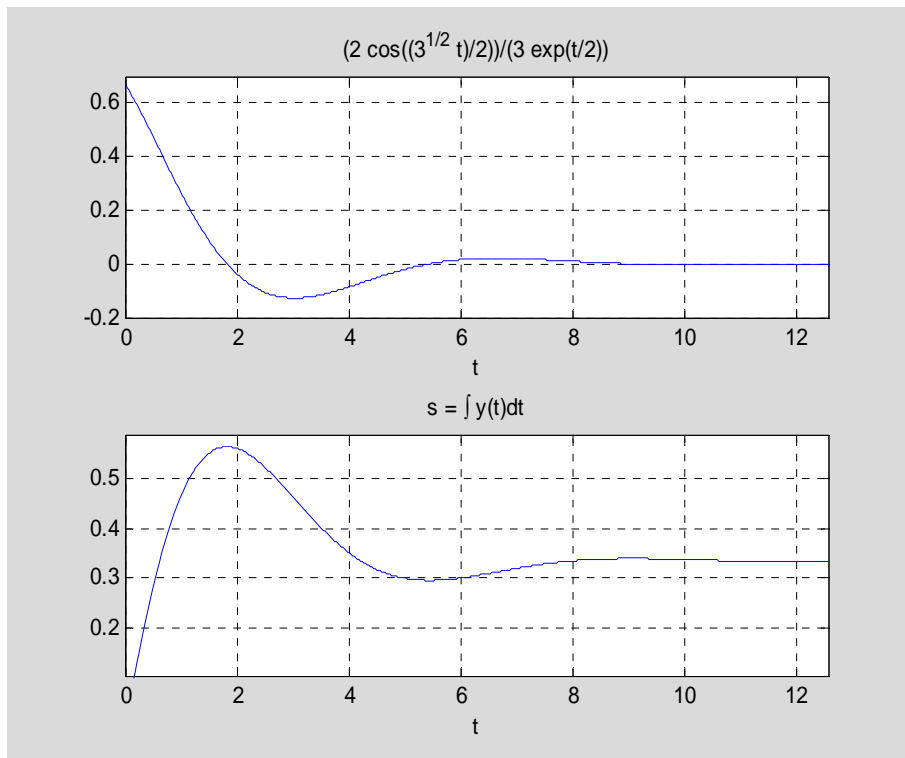



图 2.8-1 ezplot 使用示例

2 双独立变量符号函数的可视化

【例 2.8-2】

```

clf
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]) %

```

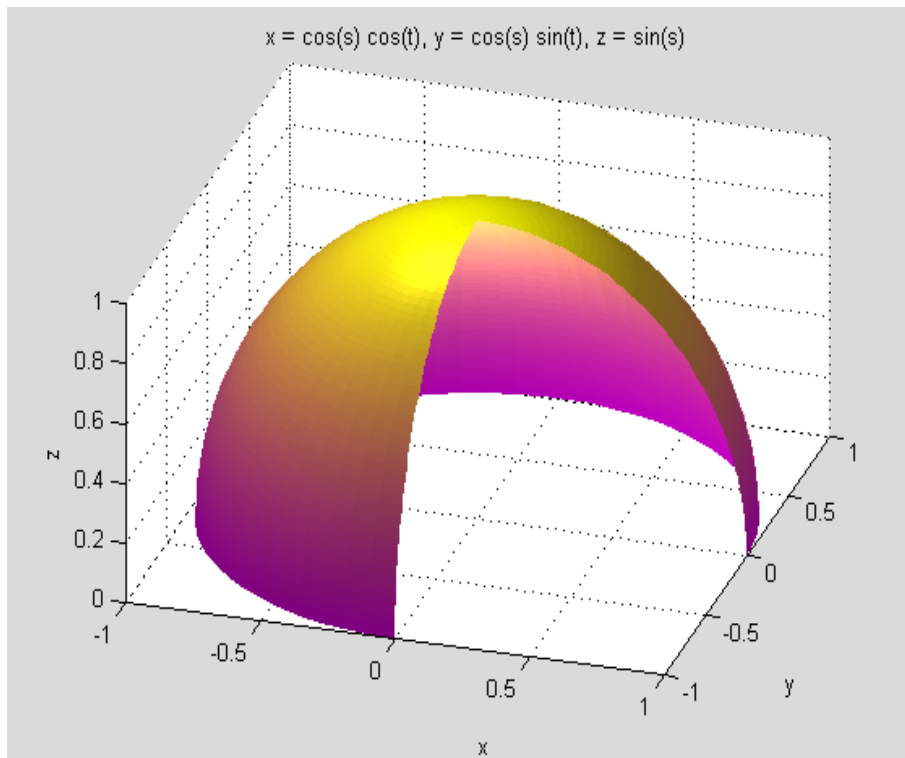


图 2.8-2 ezsurf 在参变量格式下绘制的图形

2.8.2 符号计算结果的数值化绘图

【例 2.8-3】

(1)

```
clear
syms x y real           %
fx=1-2/(1+exp(x));      %
disp('f(x)=')
pretty(fx)
disp(' ')
fxint=int(fx,x,0,x)      %
f(x)=
```

$$1 - \frac{2}{\exp(x) + 1}$$

```
fxint =
log((exp(x) + 1)^2/4) - x
```

(2)

```
xk=0:0.1:2;             %
fxk=subs(fx,x,xk);       %
fxintk=subs(fxint,x,xk); %
plot(xk,fxk,'g',xk,fxintk,'r','LineWidth',2.5)
%
title
xlabel('x')
legend('f(x)', '\int^x_0 f(x) dx','Location','best')
```

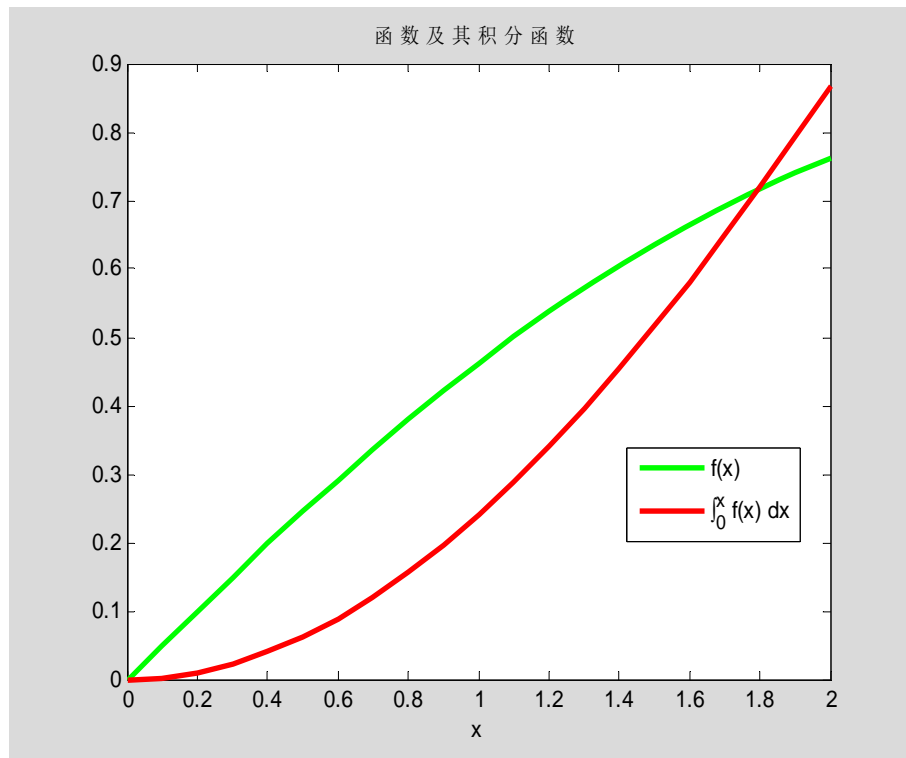


图 2.8-3 用双精度数据绘曲线

(3)

```
gy=subs(finverse(fx),x,y) %
gyint=int(gy,y,0,y) %
gy =
log(-(y + 1)/(y - 1))
gyint =
piecewise([y < 1, log(1 - y^2) + y*log(y + 1) - y*log(1 - y)], [1 <= y,
log(y^2 - 1) + y*log(-(y + 1)/(y - 1)) - pi*i])
```

(4)

```
gf=simplify(subs(gy,y,fx)) %
gf =
x
```

(5)

```
yk=subs(fx,x,xk); %
gyintk=subs(gyint,y,yk); %
GYintk=xk.*fxk-fxintk; %
plot(yk,gyintk,'r','LineWidth',5)
hold on
plot(yk,GYintk,'+k')
hold off
xlabel('y')
legend('直接法计算反函数积分', '互补法求反函数积分','location','best')
```

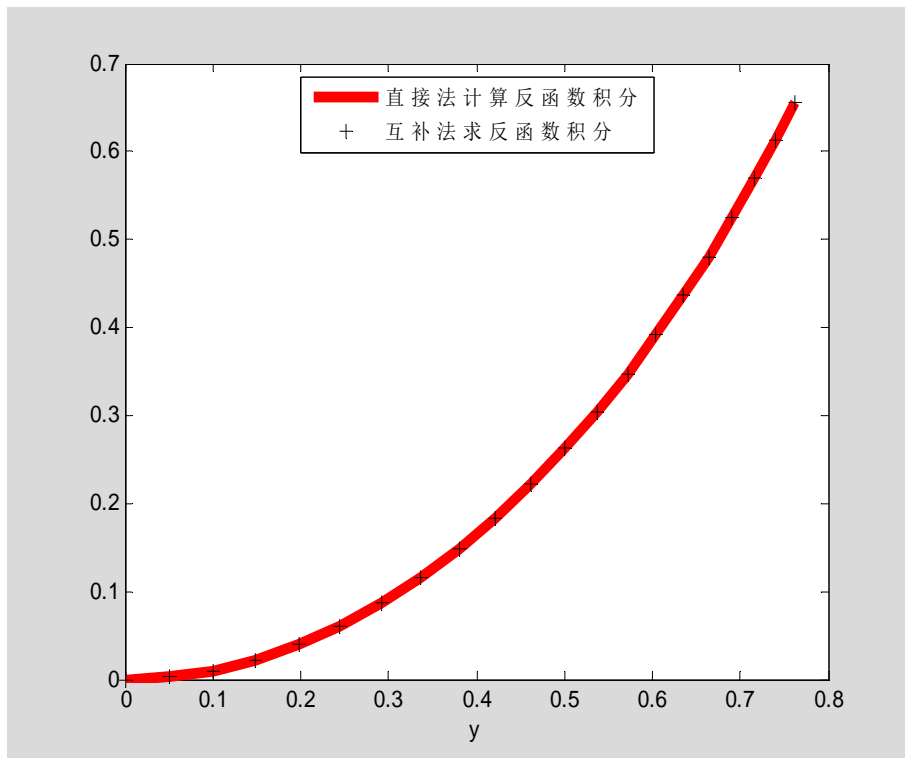


图 2.8-4 反函数积分两种算法结果比较

2.8.3 可视化与数据探索

【例 2.8-4】

(1)

```
TL1=evalin(symengine,'mtaylor(sin(x^2+y),[x,y],8)')%
TL1 =
(x^6*y^2)/12 - x^6/6 + (x^4*y^3)/12 - (x^4*y)/2 - (x^2*y^6)/720 +
(x^2*y^4)/24 - (x^2*y^2)/2 + x^2 - y^7/5040 + y^5/120 - y^3/6 + y
```

(2)

```
Fxy=sym('sin(x^2+y)')
Fxy_TL1=Fxy-TL1 %
Fxy =
sin(x^2 + y)
Fxy_TL1 =
sin(x^2 + y) - y + (x^2*y^2)/2 - (x^2*y^4)/24 - (x^4*y^3)/12 +
(x^2*y^6)/720 - (x^6*y^2)/12 + (x^4*y)/2 - x^2 + x^6/6 + y^3/6 - y^5/120
+ y^7/5040
```

(3)

```
figure(1) %
ezsurf(Fxy,[-2,2,-3,3]) %
shading interp %
view([-63,52]) %
colormap(spring)
light,light('position',[-10,4,50],'style','local','color','r')
```

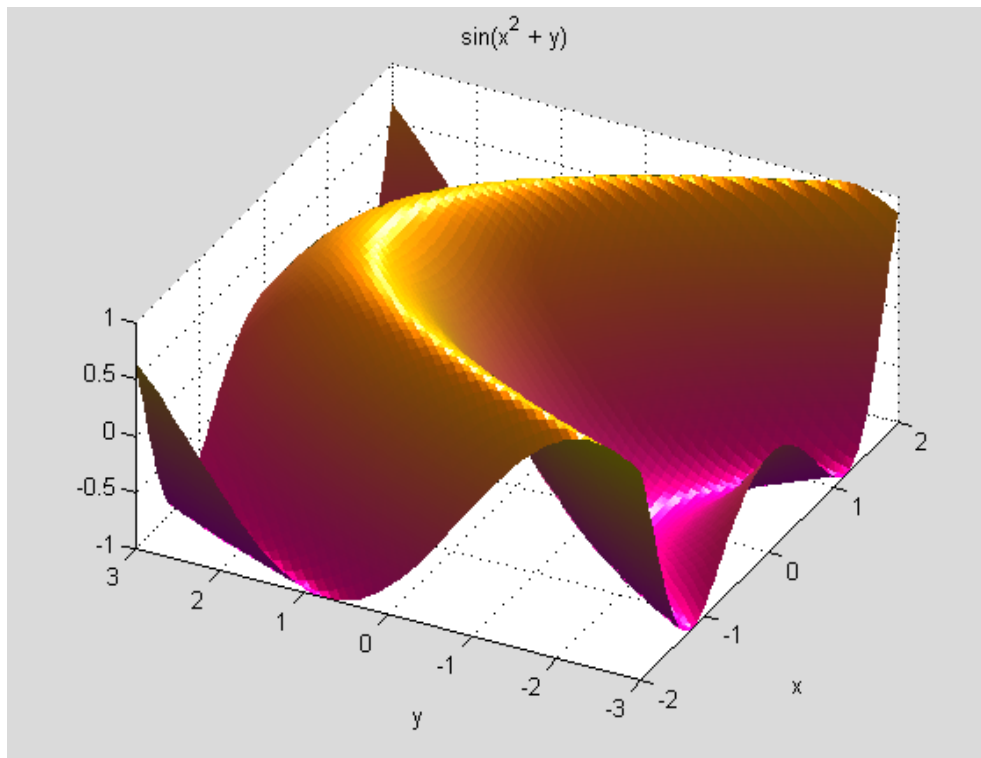


图 2.8-5 原函数在较大范围内的图形

```
(4)
figure(2)
ezsurf(TL1,[-2,2,-3,3])
shading interp
view([-43,54])
colormap(spring)
light
light('position',[-10,2,2],'style','local','color',[0.8,0.3,0.3])
light('position',[-2,-10,2],'style','local','color',[0.4,0.5,0.7])
```

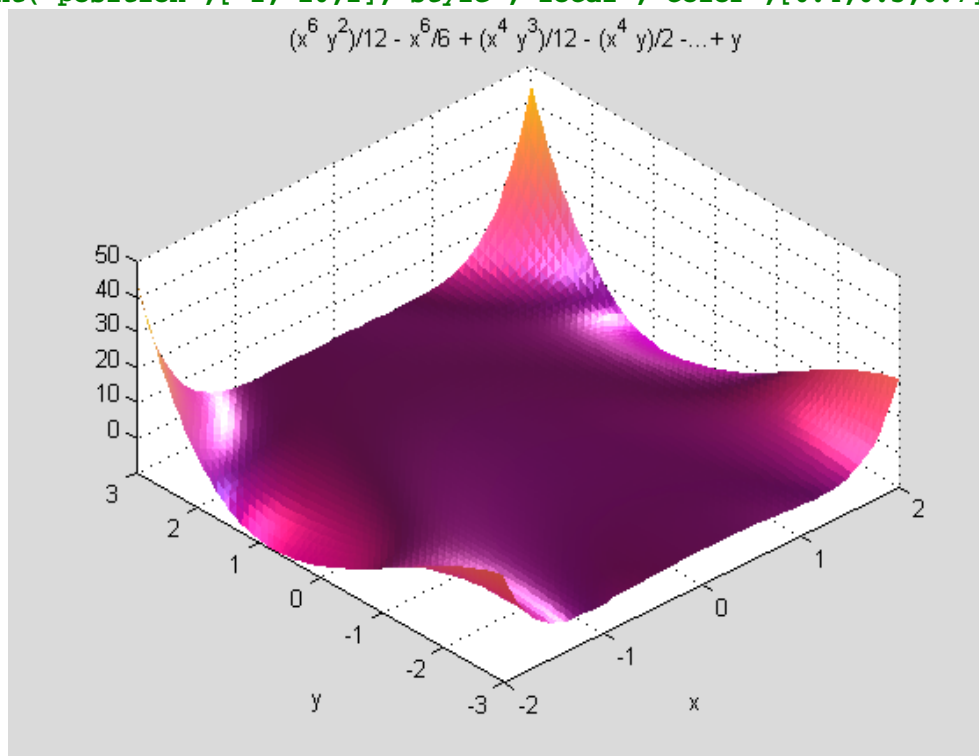
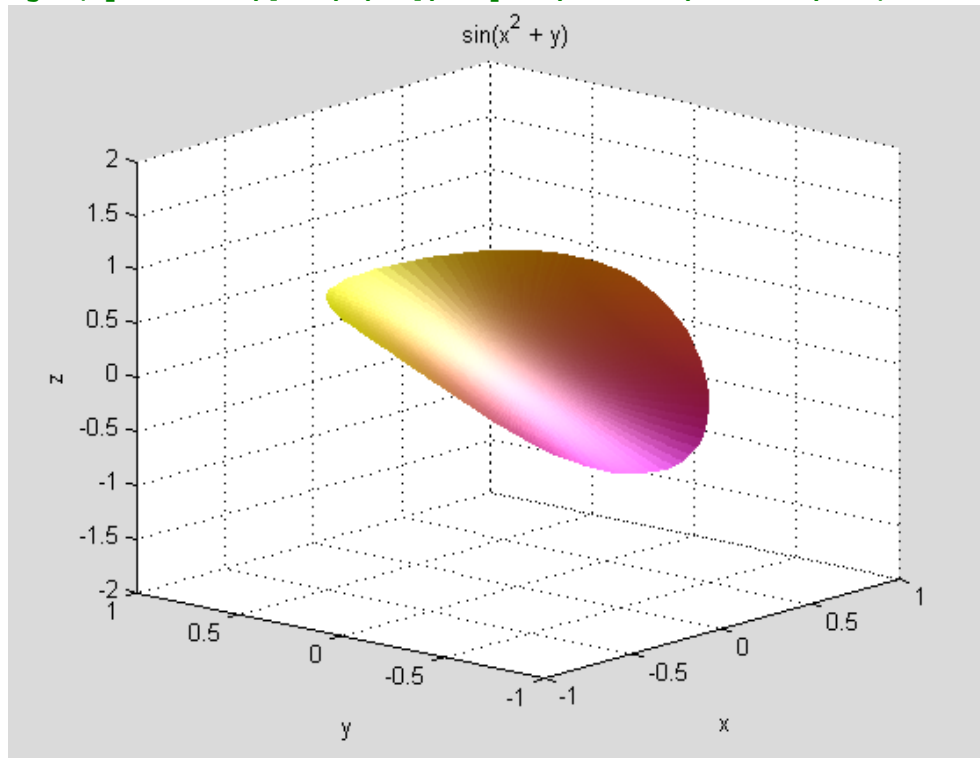


图 2.8-6 Taylor 展开在较大范围内的图形

```
(5)
figure(3)
ezsurf(Fxy,[-0.5,0.5,-0.5,0.5],'circ')      %
axis([-1,1,-1,1,-2,2])
shading interp
colormap(spring)
view([-49,17])
light
light('position',[-30,0,-2],'style','local','color','r')
```



2'r'

图 2.8-7 原函数在小范围内的图形

```
(6)
figure(4)
ezsurf(Fxy_TL1,[-0.5,0.5],'circ')
shading interp
colormap(spring)
view([-53,34])
light
light('position',[-10,15,0],'style','local','color',[0.8,0.3,0.3])
```

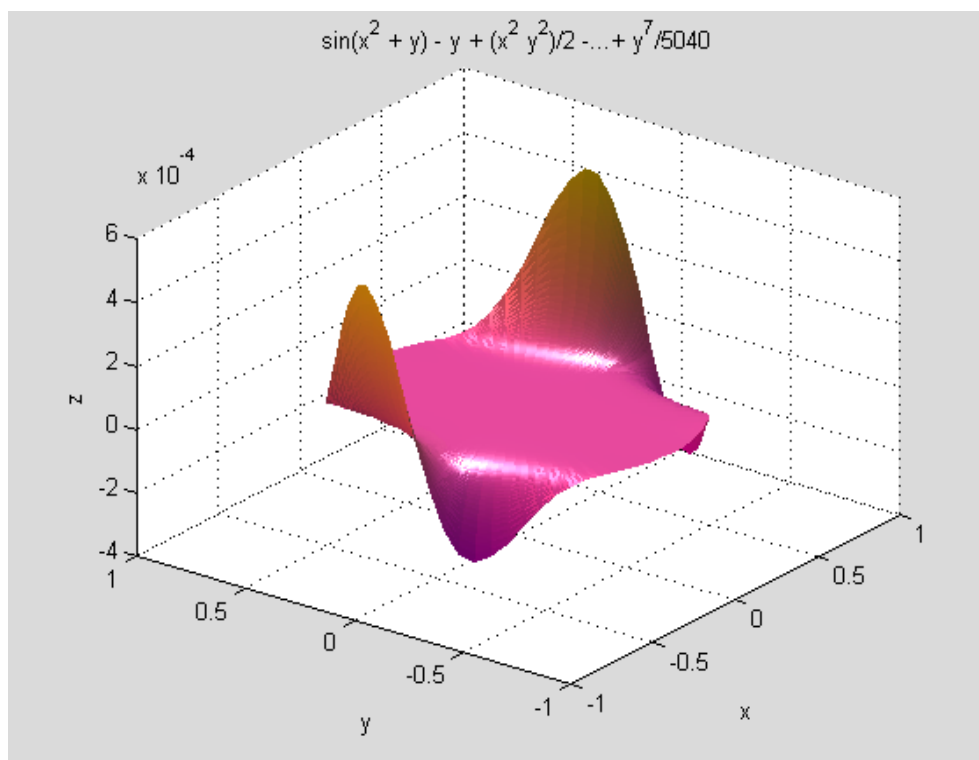


图 2.8-8 小范围内的误差曲面

2.9 符号计算资源深入利用

2.9.1 符号表达式、串操作及数值计算 M 码间的转换

2.9.2 符号工具包资源表达式转换成 M 码函数

1 把符号包资源转换成 M 码函数的示例

【例 2.9-1】

(1)

```
s=dsolve('x*D2y-3*Dy=x^2','y(a)=yL,y(b)=yR','x')
s =
(a^4*b^3 + 3*yR*a^4 - a^3*b^4 - 3*yL*b^4)/(3*a^4 - 3*b^4) - x^3/3 +
(x^4*(a^3 - b^3 + 3*yL - 3*yR))/(3*a^4 - 3*b^4)
```

(2)

```
Hs=matlabFunction(s,'file','exm050901_ZZY','vars',{'x','a','b','yL','yR'},'outputs',{'y'})
Hs =
@exm050901_ZZY % <2>
```

(3)

```
a=1;b=5;yL=0;yR=0;
xn=-1:6;yn=Hs(xn,a,b,yL,yR) % <4>
x=-1:0.2:6;y=Hs(x,a,b,yL,yR); % <5>
plot(x,y,'b-'),hold on
```

```

plot([1,5],[0,0],'.r','MarkerSize',20),hold off
title(['xy{\prime\prime}- 3y{\prime}=x^2',' ','y(1)=0, y(5)=0'])
text(1,1,'y(1)=0'),text(4,1,'y(5)=0')
xlabel('x'),ylabel('y')
yn =
Columns 1 through 7
    0.6667    0.2671    0.0000   -1.3397   -3.3675   -4.1090    0.0000
Column 8
    14.1132

```

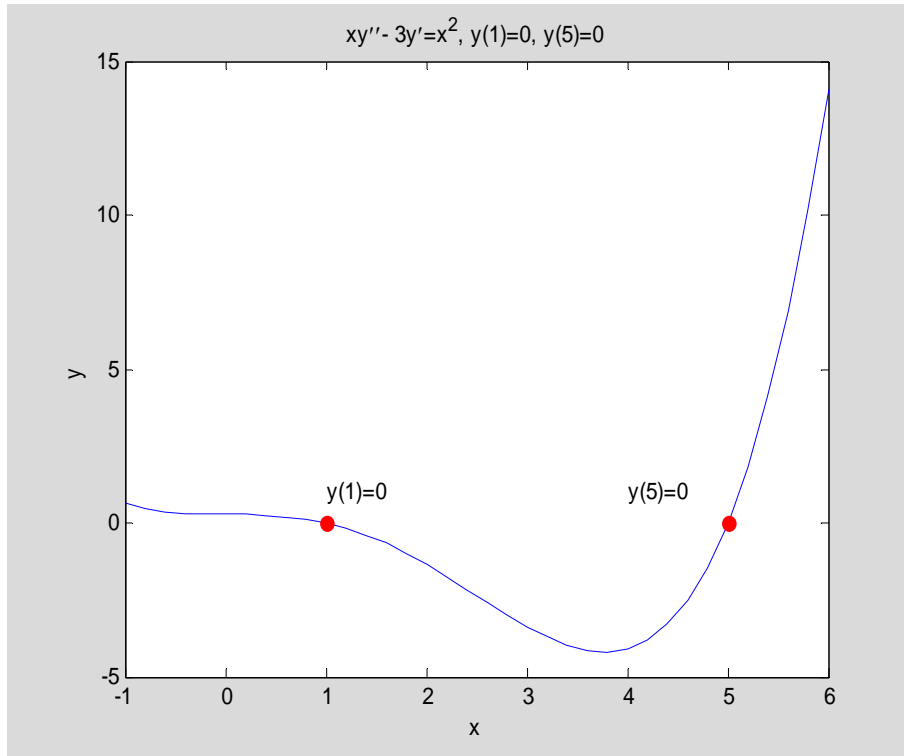


图 5.9-2 两点边值问题的解曲线

(4)

```
function y = exm050901_ZZY(x,a,b,yL,yR)
```

```

%
%
%
%
%
%
%
%
%
%
%

```

```

t7 = a.^2;
t8 = b.^2;
t9 = t7.^2;
t10 = t8.^2;
t11 = x.^2;
t12 = 3.*t10;
t13 = t12 - 3.*t9;
t14 = 1./t13;
y = t14.*(3.*t10.*yL - 3.*t9.*yR + a.*t10.*t7 - b.*t8.*t9) - (x.*t11)./3 - t14.*(3.*yL - 3.*yR +

```


a.*t7 - b.*t8).*t11.^2;

(5)

```
ym=exm050901_ZZY(xn,a,b,yL,yR) % <12>
ym =
    Columns 1 through 7
    0.6667    0.2671    0.0000   -1.3397   -3.3675   -4.1090    0.0000
    Column 8
    14.1132
```

■ [

【例 2.9-2】

(1)

```
function y = exm050902_ZZY(de,x,a,b,yL,yR,flag)
%
%
%
%
%
%
%
%
```

```
if nargin~=7
    error('输入量数目应有 7 个 !')
end
s=dsolve(de,'y(a)=yL,y(b)=yR','x');
Hs=matlabFunction(s,'vars',{'x','a','b','yL','yR'}); % <14>
y=Hs(x,a,b,yL,yR);
if flag==1
    plot(x,y,'-b',[a,b],[yL,yR],'*r')
    title(de)
    xlabel('x'),ylabel('y')
    shg
end
```

(2)

```
x=-1:6;
a=1;b=5;yL=0;yR=0;
de1='x*D2y-3*Dy=x^2';
y = exm050902_ZZY(de1,x,a,b,yL,yR,0)
y =
    0.6667    0.2671    0.0000   -1.3397   -3.3675   -4.1090    0
    14.1132
```

(3)

```
de2='x*D2y-3*Dy=3*x^2+x';
x=-1:0.2:6;
y = exm050902_ZZY(de2,x,3,6,-2,10,1);
```

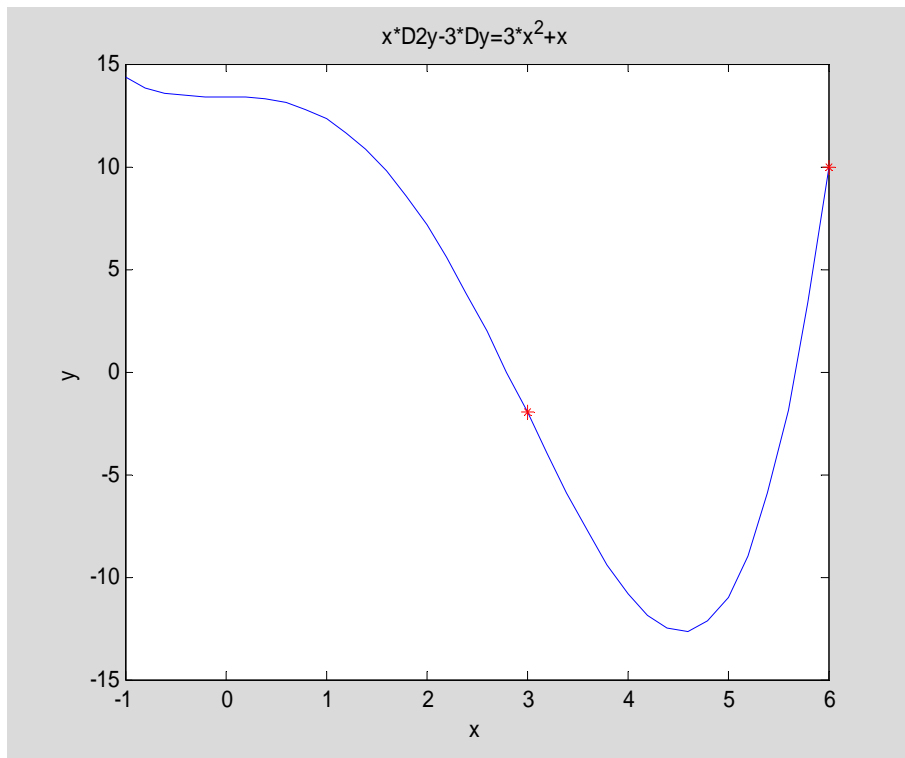


图 5.9-3 用 exm050902_ZZY 求解另一微分方程所得解

2.9.3 借助 mfun 调用 MuPAD 特殊函数

【例 2.9-3】

(1)

```
syms t
gt=1/log(t);
gt_0=limit(gt, t,0,'right') %
gt_0 =
0
```

(2)

```
ezplot(gt,[0,1]) %
grid on
legend('gt')
```

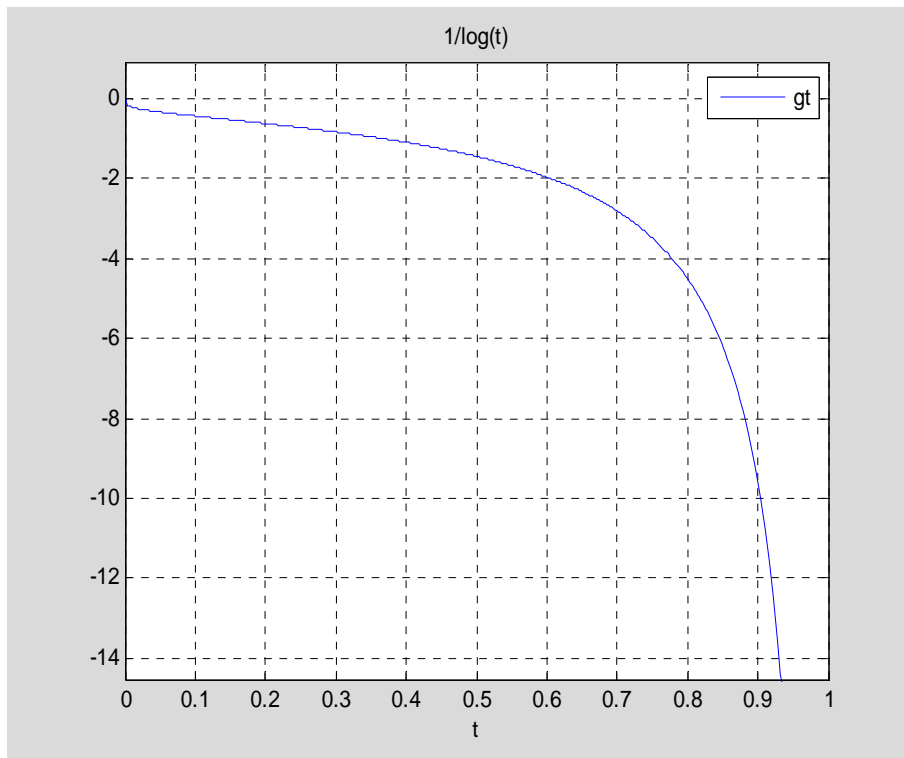


图 5.10-1 在关心区间内的被积函数

(3)

```
fx=int(gt,t,0,x) % <7>
Warning: Explicit integral could not be found.
fx =
piecewise([x < 1, Li(x)], [Otherwise, int(1/log(t), t = 0..x)])
```

(4)

```
x=0:0.05:0.9; % <8>
fxMfun=mfun('Li',x) % <9>
%
fxMfun =
Columns 1 through 8
NaN -0.0131 -0.0324 -0.0564 -0.0851 -0.1187 -0.1574
-0.2019
Columns 9 through 16
-0.2529 -0.3114 -0.3787 -0.4564 -0.5469 -0.6534 -0.7809
-0.9369
Columns 17 through 19
-1.1340 -1.3959 -1.7758
```

(5)

```
hold on
plot(x,fxMfun,'--r','LineWidth',3)
legend('gt','fxMfun','Location','Best')
title(' ') %
hold off
```

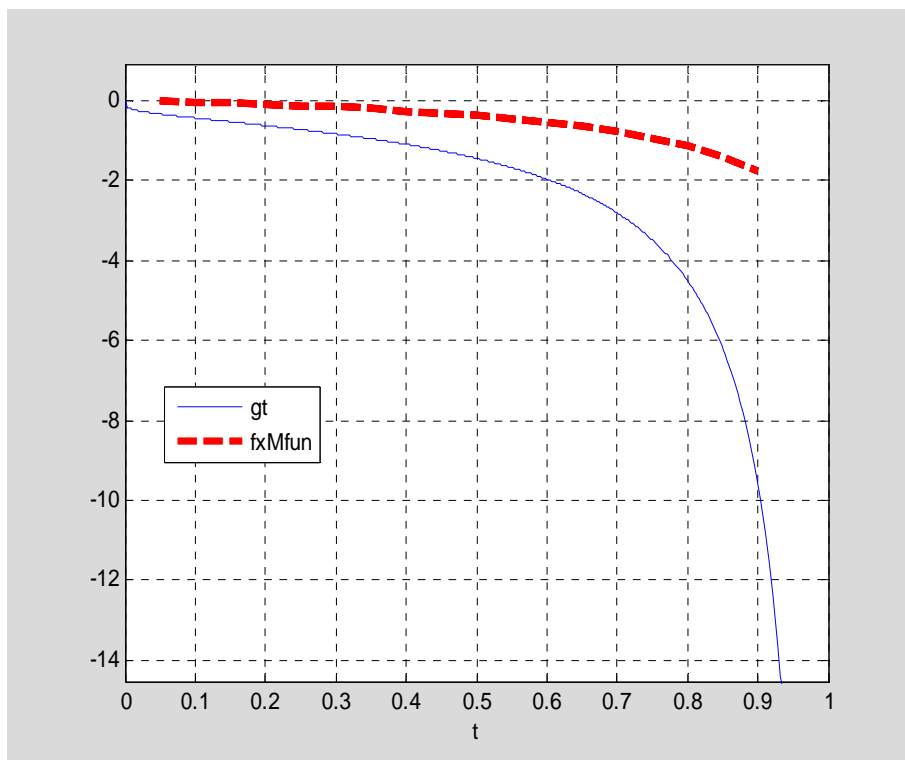


图 5.10-2 被积函数曲线 gt 和积分曲线 fx

(6)

```
fx_matlab=-expint(-log(x)) % <15>
%
fx_matlab =
Columns 1 through 8
NaN -0.0131 -0.0324 -0.0564 -0.0851 -0.1187 -0.1574
-0.2019
Columns 9 through 16
-0.2529 -0.3114 -0.3787 -0.4564 -0.5469 -0.6534 -0.7809
-0.9369
Columns 17 through 19
-1.1340 -1.3959 -1.7758
```

习题 2

- 说出以下四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？
 $3/7+0.1$, $\text{sym}(3/7+0.1)$, $\text{vpa}(\text{sym}(3/7+0.1))$
- 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。
 $\text{sym}(\sin(w*t))$, $\text{sym}(a*\exp(-X))$, $\text{sym}(z*\exp(j*th))$
- 求以下两个方程的解：（提示：关于符号变量的假设要注意）
 (1) 试写出求三阶方程 $x^3 - 44.5 = 0$ 正实根的程序。注意：只要正实根，不要出现其他根。
 (2) 试求二阶方程 $x^2 - ax + a^2 = 0$ 在 $a > 0$ 时的根。
- 观察一个数（在此用@记述）在以下四条不同指令作用下的异同：
 $a = @$, $b = \text{sym}(@)$, $c = \text{sym}(@, 'd')$, $d = \text{sym}('@')$
 在此，@ 分别代表具体数值 $7/3$, $\pi/3$, $\pi*3^{1/3}$ ；而异同通过 $\text{vpa}(\text{abs}(a-d))$, $\text{vpa}(\text{abs}(b-d))$, $\text{vpa}(\text{abs}(c-d))$ 等来观察。

5. 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。
6. 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 $\sum_{k=0}^{\infty} f(k)z^{-k}$ 。(提示: symsum。
实际上, 这就是根据定义求 Z 变换问题。)
7. 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 $\ln x$; 注意限定性假设)
8. (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 $\frac{dy}{dt} \Big|_{t=\frac{\pi}{2}^+}$ 。
9. 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。(提示: int, vpa, ezplot)
10. 计算二重积分 $\int_1^2 \int_1^{x^2} (x^2 + y^2) dy dx$ 。
11. 在 $[0, 2\pi]$ 区间, 画出 $y(x) = \int_0^x \frac{\sin t}{t} dt$ 曲线, 并计算 $y(4.5)$ 。(提示: int, subs)
12. 在 $n > 0$ 的限制下, 求 $y(n) = \int_0^{\frac{\pi}{2}} \sin^n x dx$ 的一般积分表达式, 并计算 $y(\frac{1}{3})$ 的 32 位有效数字表达。(提示: 注意限定条件; 注意题目要求 32 位有效)
13. 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 $y(k) = \sum_{n=0}^k h(n)x(k-n)$ 。(提示: symsum, subs)
14. 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: 直接卷积法, 变换法均可;)
15. 求 $f(t) = Ae^{-\alpha|t|}$, $\alpha > 0$ 的 Fourier 变换。(提示: 注意限定)
16. 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。(提示: 注意限定; simple)
17. 求 $F(s) = \frac{s+3}{s^3 + 3s^2 + 6s + 4}$ 的 Laplace 反变换。
18. 利用符号运算证明 Laplace 变换的时域求导性质: $L\left[\frac{df(t)}{dt}\right] = s \cdot L[f(t)] - f(0)$ 。(提示: 用 **sym('f(t)')** 定义函数 $f(t)$)
19. 求 $f(k) = ke^{-\lambda k T}$ 的 Z 变换表达式。
20. 求方程 $x^2 + y^2 = 1, xy = 2$ 的解。(提示: 正确使用 solve)
21. 求图 p2-1 所示信号流图的系统传递函数, 并对照胡寿松主编“自动控制原理”中的例

2-21 结果，进行局部性验证。（提示：在局部性验证时，把不存在支路的符号增益设置为 0）

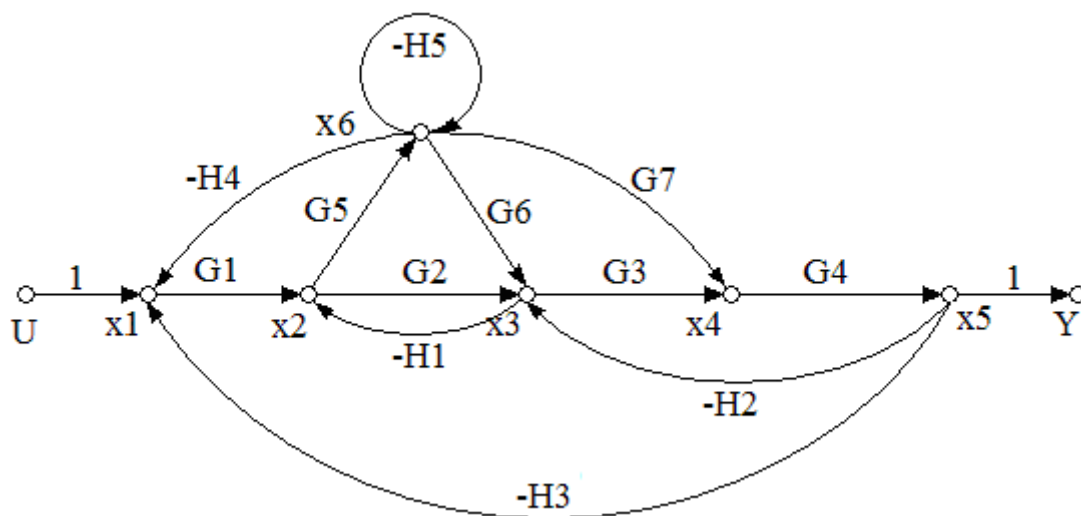


图 p2-1

22. 采用代数状态方程法求图 p2-2 所示结构框图的传递函数 $\frac{Y}{U}$ 和 $\frac{Y}{W}$ 。（提示：列出正确的状态方程

的状态方程 $\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}U + \mathbf{f}W \\ Y = \mathbf{c}\mathbf{x} + \mathbf{d}U + \mathbf{g}W \end{cases}$ ，进而写出相关输入输出对之间的传递函数表达式

$$\frac{Y}{U} = \mathbf{c}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d \text{ 和 } \frac{Y}{W} = \mathbf{c}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{f} + g$$

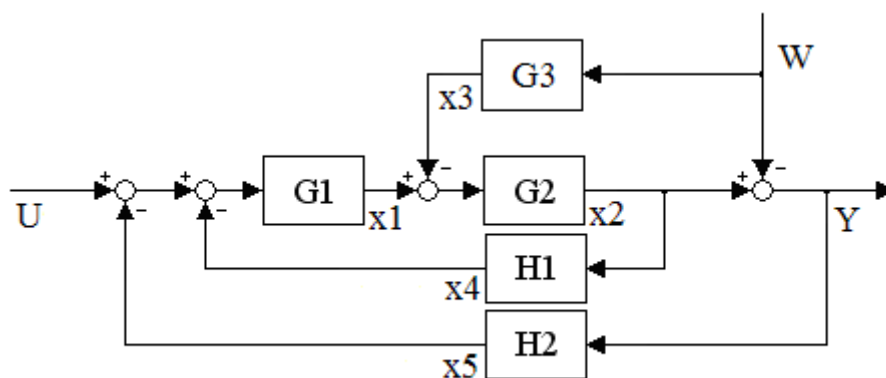


图 p2-2

23. 求微分方程 $yy' / 5 + x / 4 = 0$ 的通解，并绘制任意常数为 1 时，如图 p2-3 所示的解曲线图形。（提示：通解中任意常数的替代；构造能完整反映所有解的统一表达式，然后绘图。）

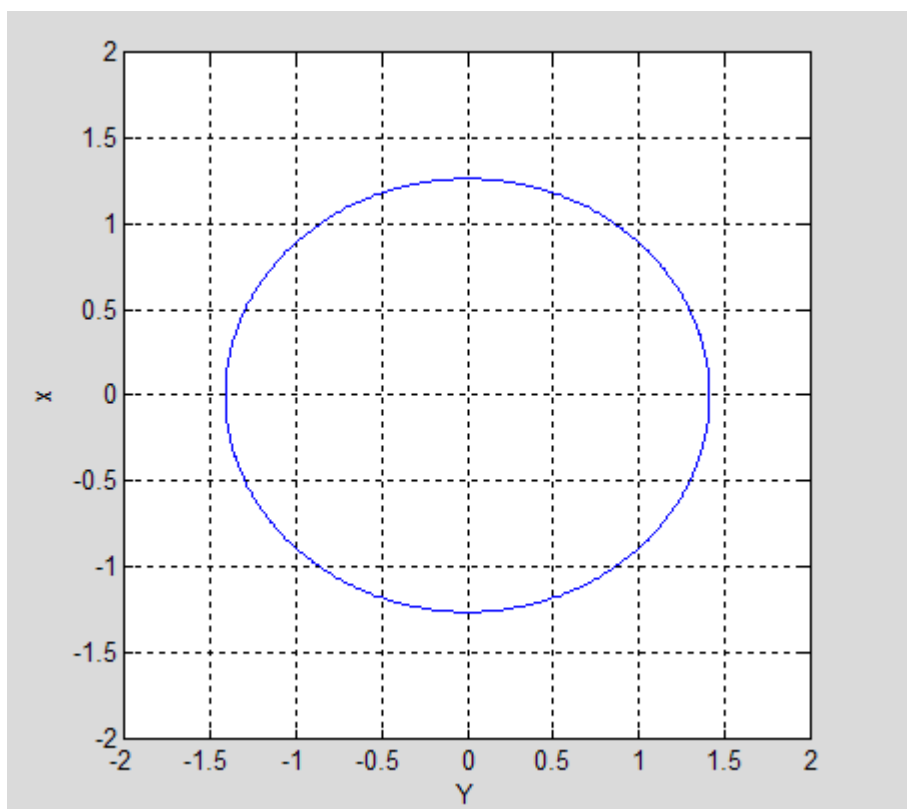


图 p2-3 微分方程的解曲线

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

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

第 3 章 数值数组及向量化运算

3.1 数值计算的特点和地位

【例 3.1-1】

(1)

```
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)

```
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
```

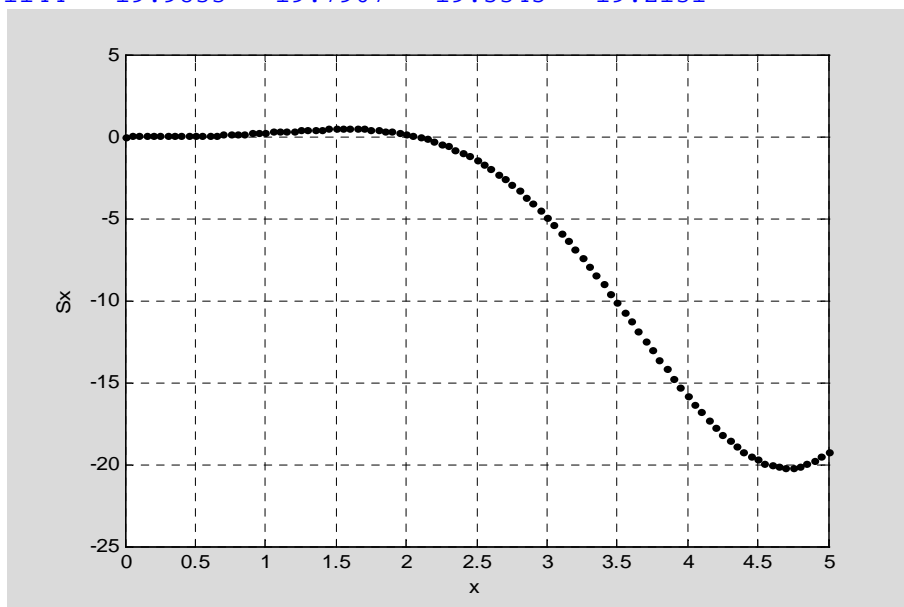


图 3.1-1 在区间[0, 5]采样点上算得的定积分值

【例 3.1-2】

(1)

```
syms t x
ft=exp(-sin(t))
sx=int(ft,t,0,4)
ft =
1/exp(sin(t))
Warning: Explicit integral could not be found.
sx =
int(1/exp(sin(t)), t = 0..4)
```

(2)

```
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
```

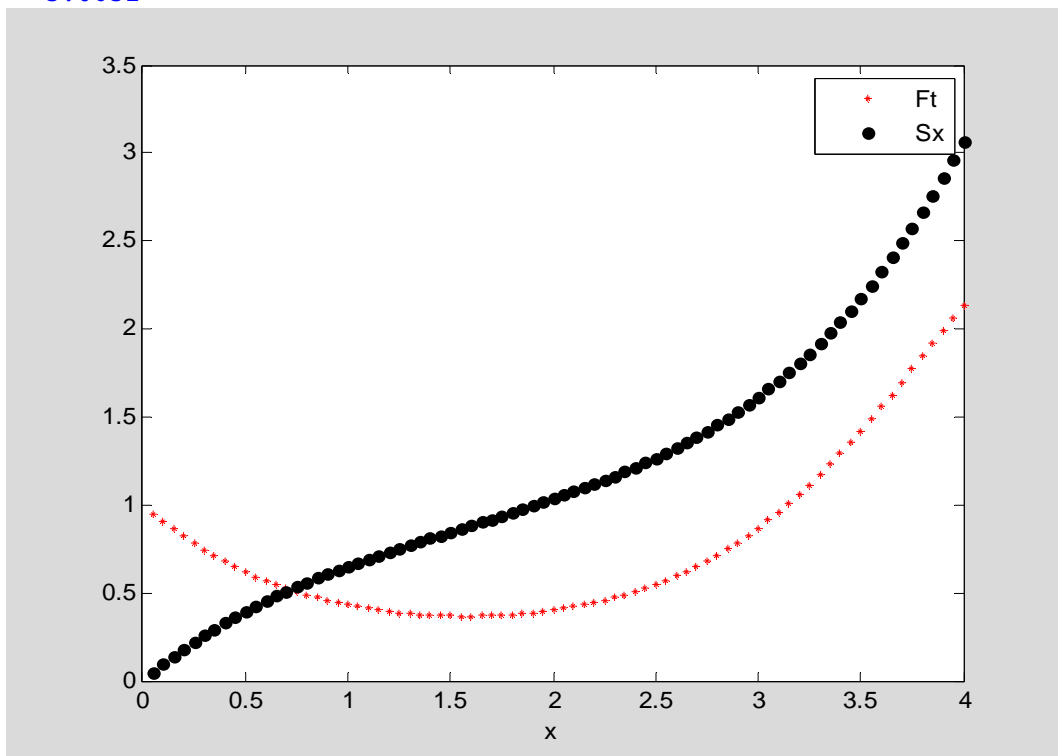


图 3.1-2 在区间[0, 4]中间的被积函数及其原函数的离散计算结果

3.2 数值数组的创建和寻访

3.2.1 一维数组的创建

1 其他类型一维数组的创建

【例 3.2-1】

```
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 7  
    1.0000    0.9000    0.8000    0.7000    0.6000    0.5000    0.4000  
Columns 8 through 11  
    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('twister',0) %  
c2=rand(1,5) %  
c2 =  
    0.5488    0.7152    0.6028    0.5449    0.4237
```

【例 3.2-2】

```
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
```

2 中规模数组的数组编辑器创建法

【例 3.2-3】试用变量编辑器，把如下(3×6)的数组输入 MATLAB 内存，并命名为 A18。

0.6459	0.9637	0.5289	0.0710	0.8326	0.9786
0.4376	0.3834	0.5680	0.0871	0.7782	0.7992
0.8918	0.7917	0.9256	0.0202	0.8700	0.4615

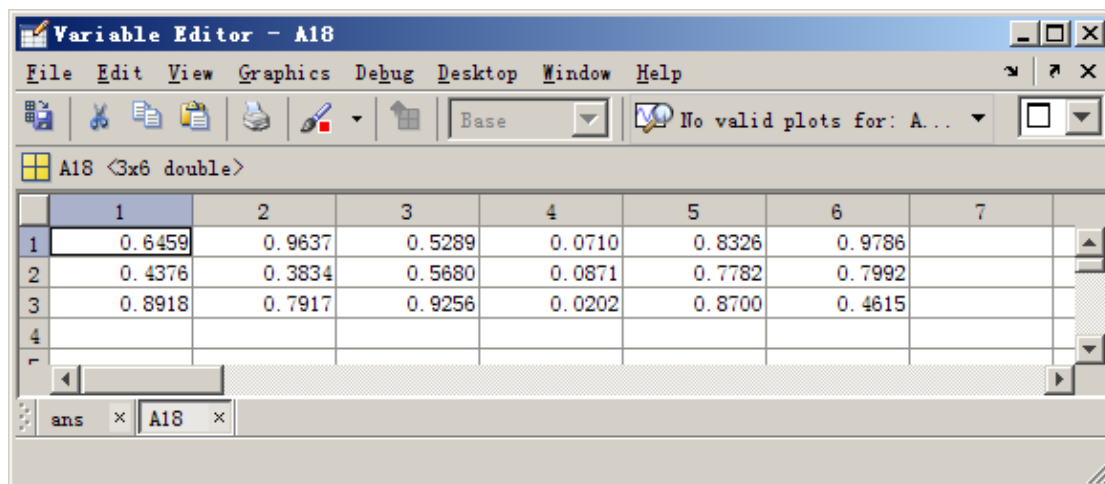


图 3.2-1 利用数组编辑器创建中规模数组

3 中规模的 M 文件创建法

【例 3.2-4】

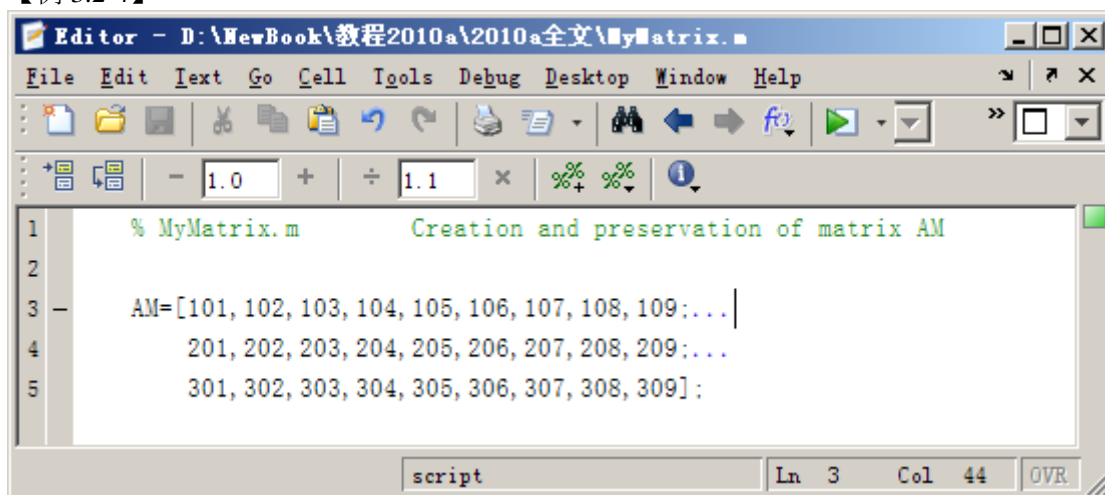


图 3.2-2 利用 M 文件创建数组

4 利用 MATLAB 函数创建数组

【例 3.2-5】

```

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 12
    -1    -1    -3     1    -3     1    -3     3     3    -3    -3     1
     1    -3    -1    -1     3    -1    -3    -1     3    -3    -1     1
    -3    -3    -1     1    -3     1     3     1    -3     3     3    -1
Columns 13 through 20
     1     3    -1    -1    -1     1    -1    -3
     3     3     3     3    -3    -3    -3     1
    -3     1    -3    -1    -3    -1     1     1

```

3.2.2 二维数组元素的标识和寻访

【例 3.2-6】

```

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

```

3.2.3 数组构造技法综合

【例 3.2-7】

```

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】

```

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

```

3.3 数组运算

【例 3.3-1】

(1)

```

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); % <7>
for k=1:L
    r(k)=vr(k)/ir(k); %
end % <10>
%
sr=0; % <12>
for k=1:L
    sr=sr+r(k); %
end
rm=sr/L % <16>
rm =
30.5247

```

(2)

```

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 =
Columns 1 through 8
31.7857 30.0000 30.9000 29.4483 30.6780 29.8837 30.0669
30.8171
Columns 9 through 10
31.4935 30.1739
rm =
30.5247

```

【例 3.3-2】

(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
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])

```

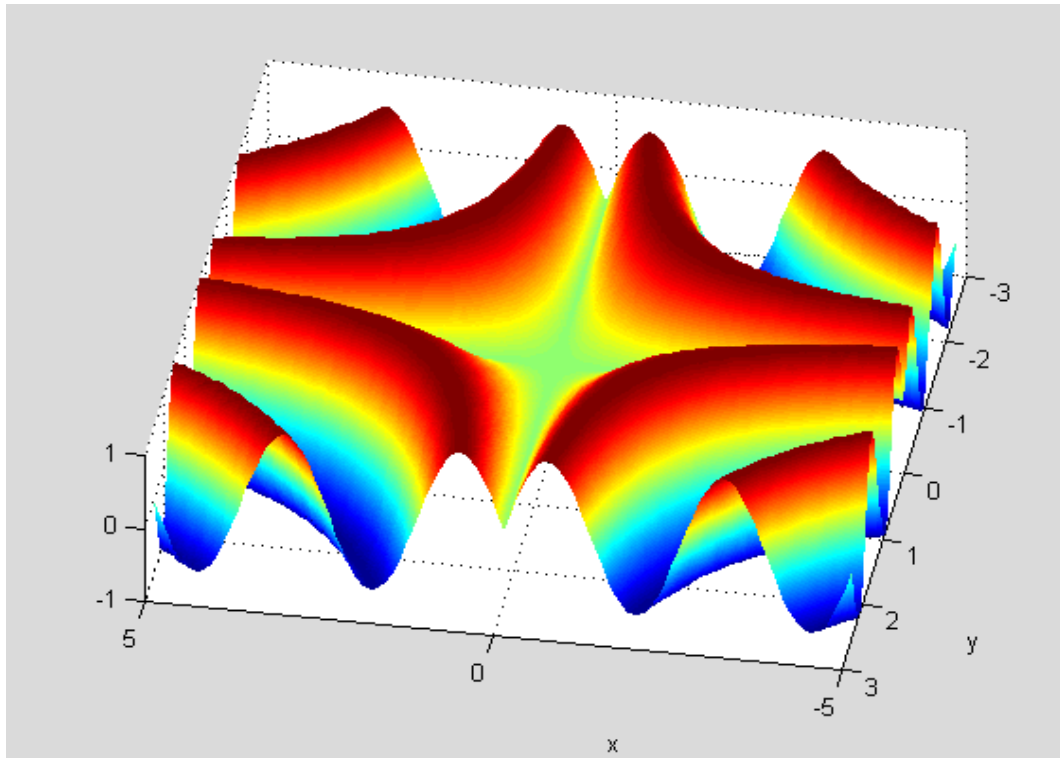


图 3.3-1 指定域上的二元函数图形

3.4 “非数”和“空”数组

【例 3.4-1】

(1)

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

```
a =  
    NaN
```

```
b =  
    NaN
```

```
c =  
    NaN
```

(2)

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

```
ans =  
    NaN
```

```
ans =  
    NaN
```

(3)

```
class(a)    %
```

```
isnan(a)    %
```

```
ans =
```

```
double
```

```
ans =  
    1
```

【例 3.4-2】

```
%
```

```
rand('twister',0) %
```

```
R=rand(2,5);R(1,5)=NaN;R(2,3)=NaN
```



```

R =
    0.5488    0.6028    0.4237    0.4376         NaN
    0.7152    0.5449         NaN    0.8918    0.3834

LR=isnan(R)           % <3>
LR =
     0     0     0     0     1
     0     0     1     0     0

%
si=find(LR)           %
[ri,ci]=ind2sub(size(R),si) %
[rj,cj]=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

```

3.4.1 “空”数组

【例 3.4-3】

(1)

```

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)

```

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)
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

```

3.5 关系操作和逻辑操作

【例 3.5-1】

```

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】

```

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')

```

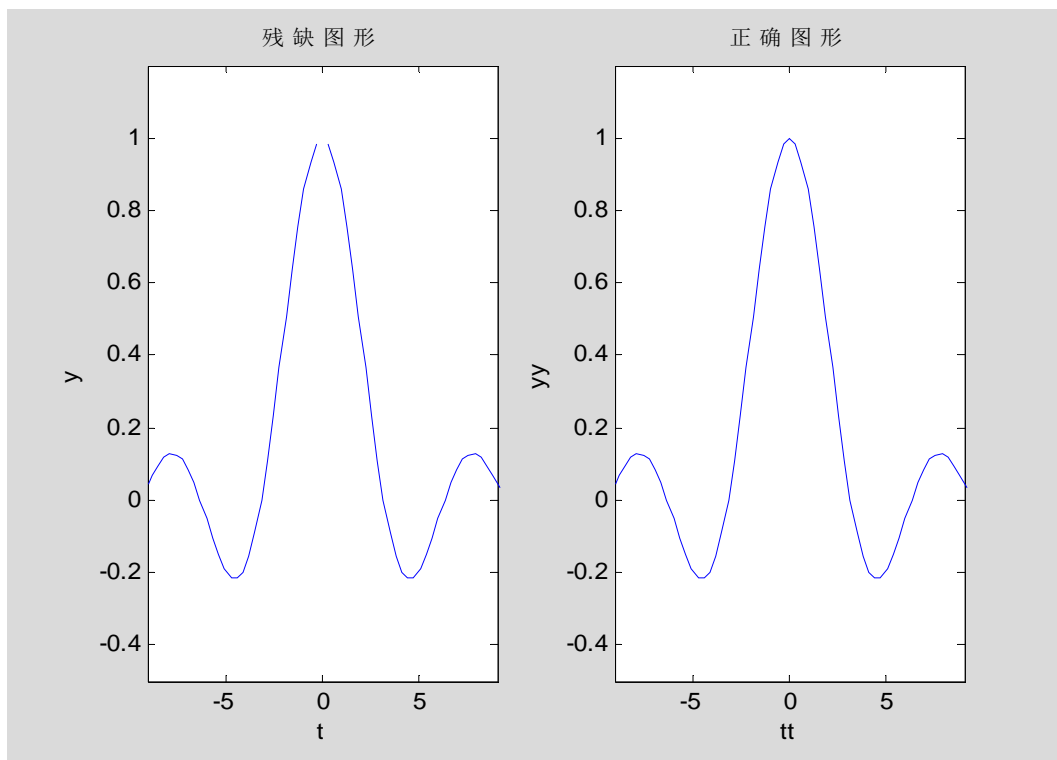


图 3.5-1 采用近似极限处理前后的图形对照

3.5.1 逻辑操作

【例 3.5-3】

(1)

```
A=[-2,-1,0,0,1,2,3]
L1=~(A>1)           %
L2=(A>0)&(A<2)      %
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)

```
A,B=[0,-1,1,0,1,-2,-3]
C=xor(A,B)          %
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】

```
clear,t=linspace(0,3*pi,500);y=sin(t); %
%
z1=((t<pi)|(t>2*pi)).*y;                % <3>
w=(t>pi/3&t<2*pi/3)+(t>7*pi/3&t<8*pi/3);% <4>
wn=~w;                                  % <5>
z2=w*sin(pi/3)+wn.*z1;                  % <6>
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')

```

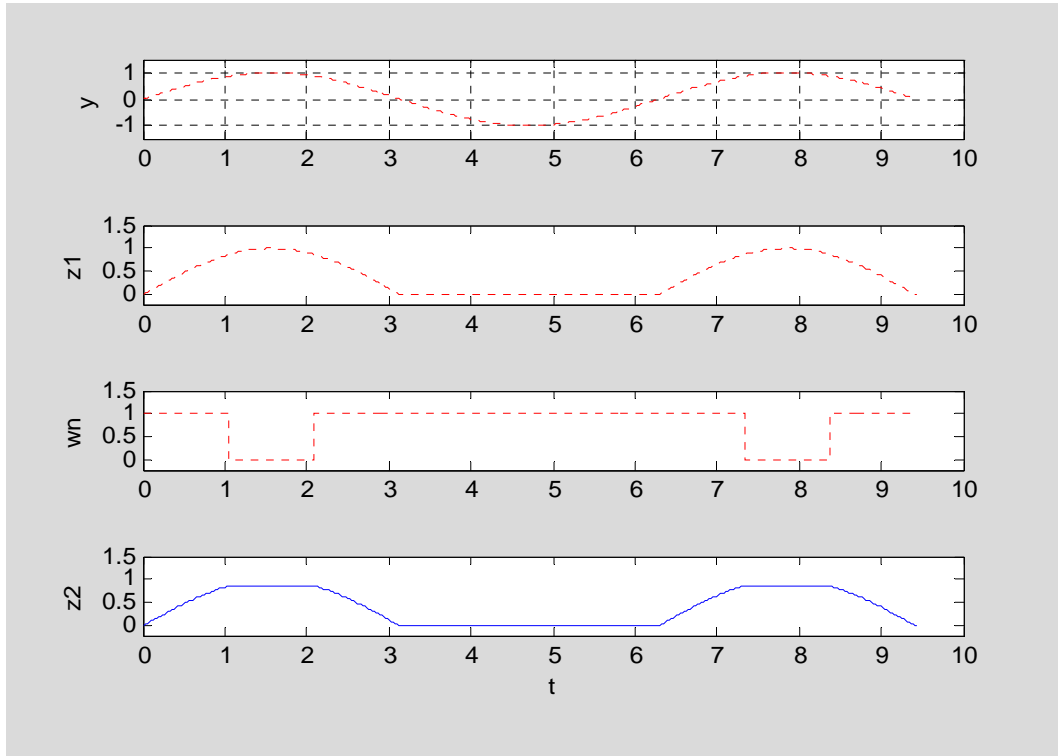


图 3.5-2 逐段解析函数的产生

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。（提示：冒号生成法，定点生成法）
2. 由指令 `rand('twister',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”。（提示：find 和 sub2ind）
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。（提示：rand, randn, randsrc 等都可以用来产生所需码。注意：“关系符==”、“求和指令 sum”的应用。）
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。（提示：数组乘、矩阵乘。注意：范数指令 norm 用途。）
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。（注意：体验数组运算的简捷。）
6. 先运行 `clear, format long, rand('twister',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。（提示：diag）
7. 先运行指令 `x=-3*pi/15:3*pi; y=x; [X,Y]=meshgrid(x,y); warning off;`

$Z = \sin(X) \cdot \sin(Y) ./ X ./ Y$; 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“无裂缝”图形的全部指令。(提示: `isnan, sum, eps`)

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, 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 (1×10) 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。

(提示: 本题专为揭示 `sum` 对行数组的作用而设计。仔细观察以下程序运行后所得到的 \mathbf{Sa} 正确吗? `for k=10:-1:1; A=reshape(1:10*k,k,10); Sa(k,:)=sum(A); end; Sa`)

第 4 章 数值计算

4.1.1 近似数值极限及导数

【例 4.1-1】

```
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】

(1)

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

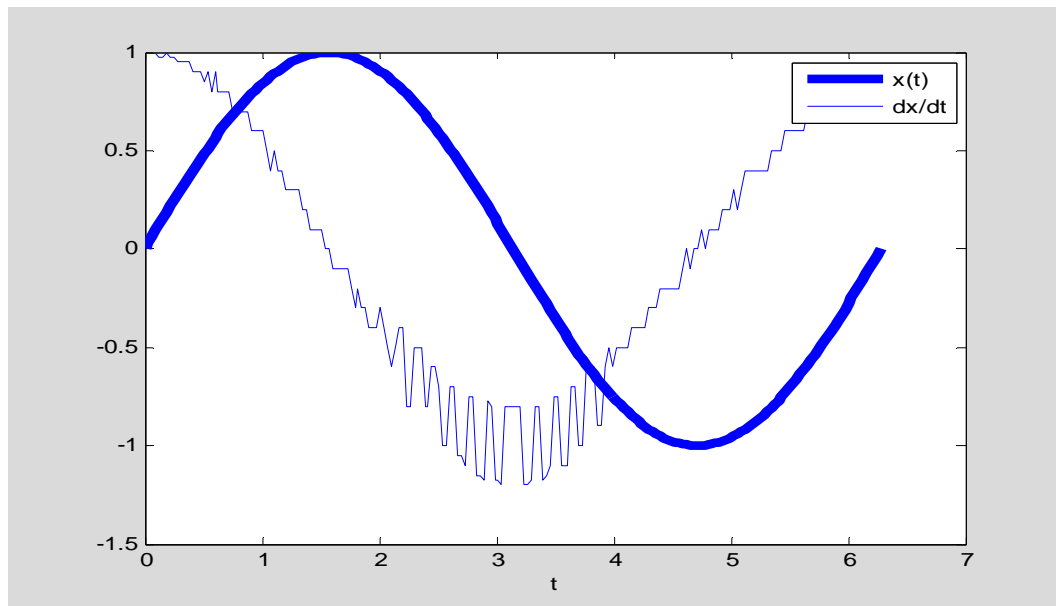


图 4.1-1 增量过小引起有效数字严重丢失后的毛刺曲线

```
(2)
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')
```

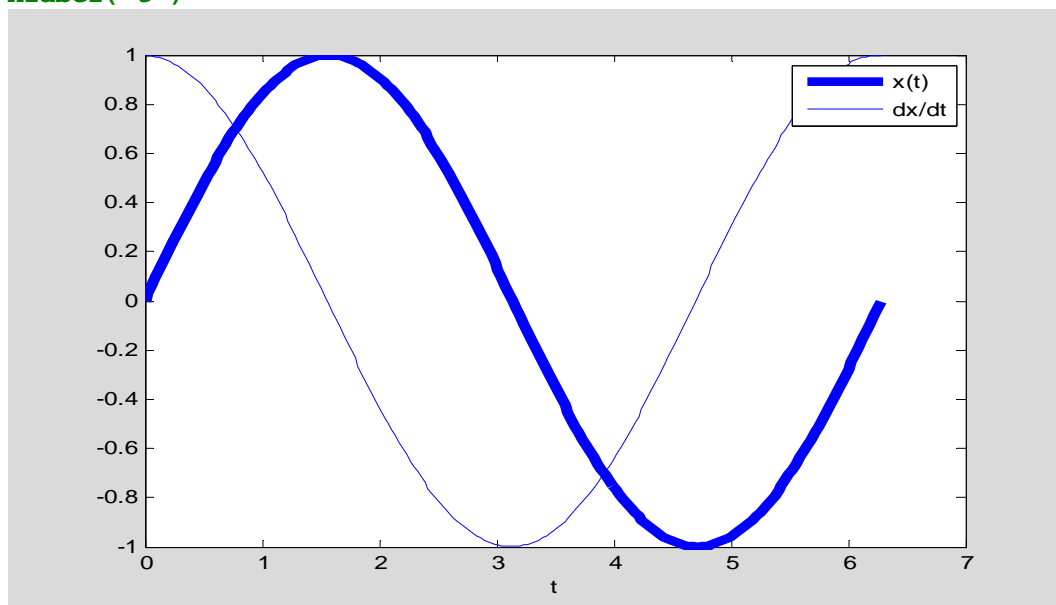


图 4.1-2 增量适当所得导函数比较光滑

【例 4.1-3】

```
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);%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

```

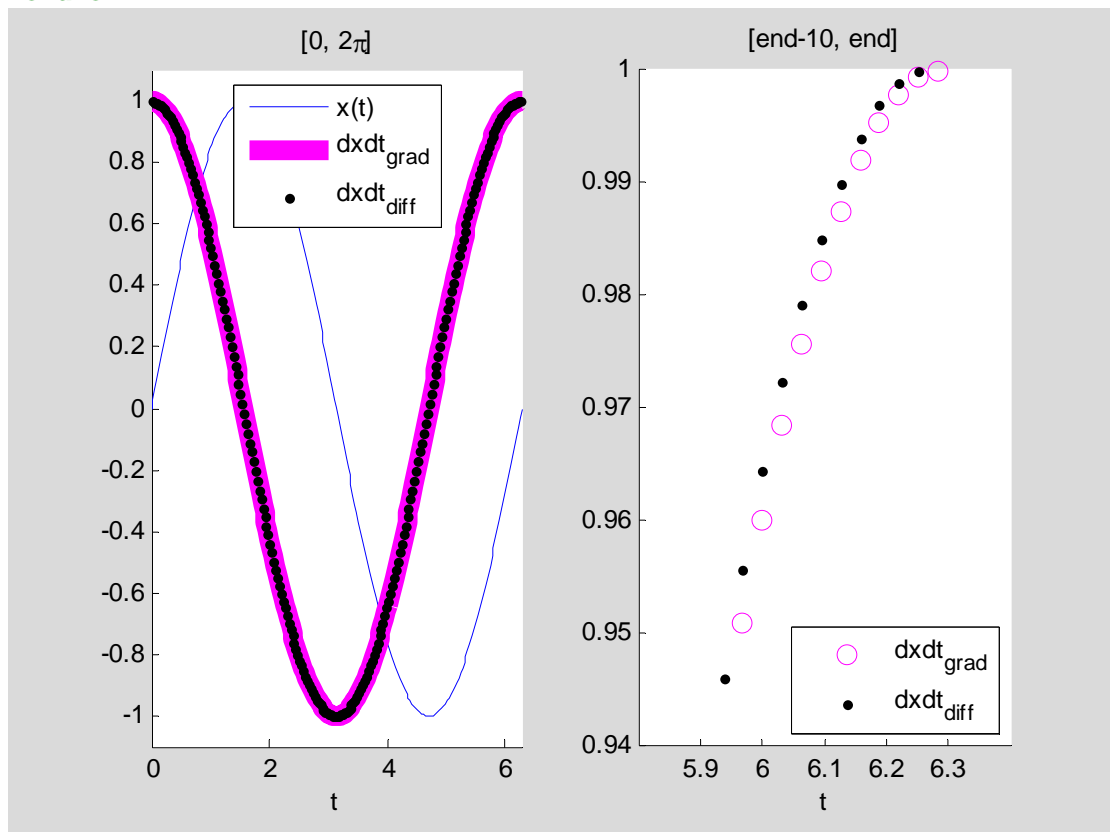


图 4.1-3 diff 和 gradient 求数值近似导数的异同比较

4.1.2 数值求和与近似数值积分

【例 4.1-4】

```

clear
d=pi/8;          %
t=0:d:pi/2;      %
y=0.2+sin(t);    %
s=sum(y);         %

```



```

s_sa=d*s; % <6>
s_ta=d*trapz(y); % <7>
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

```

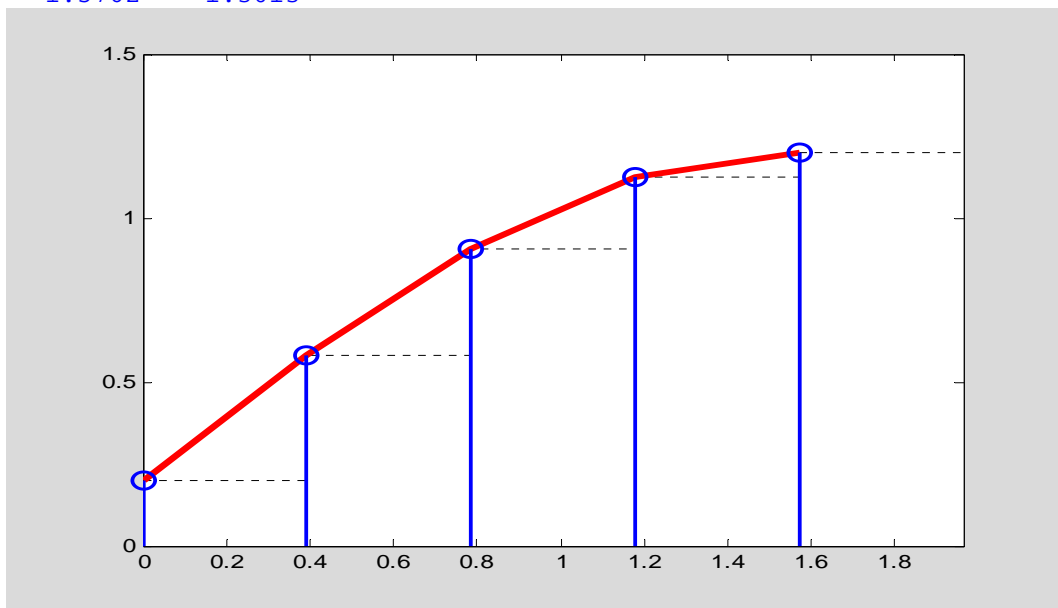


图 4.1-4 sum 和 trapz 求积模式示意

4.1.3 计算精度可控的数值积分

【例 4.1-5】

(1)

```

syms x
Isym=vpa(int(exp(-x^2),x,0,1))
Isym =
0.74682413281242702539946743613185

```

(2)

```

format long %
d=0.001;x=0:d:1;
Itrapz=d*trapz(exp(-x.*x))
Itrapz =
0.746824071499185

```

(3)

```

fx='exp(-x.^2)'; %
Ic=quad(fx,0,1,1e-8) % <7>
Ic =

```

0.746824132854452

【例 4.1-6】

(1)

```
syms x y
s=vpa(int(int(x^y,x,0,1),y,1,2))           % <2>
Warning: Explicit integral could not be found.
s =
0.40546510810816438197801311546435
```

(2)

```
format long
s_n=dblquad(@(x,y)x.^y,0,1,1,2)           % <4>
s_n =
0.405466267243508
```

4.1.4 函数极值的数值求解

【例 4.1-7】

(1)

```
syms x
y=sin(x)^2*exp(-0.1*x)-0.5*sin(x)*(x+0.1);
yd=diff(y,x);                               %
xs0=solve(yd,x)                             % <4>
yd_xs0=vpa(subs(yd,x,xs0),6)               %
y_xs0=vpa(subs(y,x,xs0),6)                %
xs0 =
matrix([[0.050838341410271656880659496266968]])
yd_xs0 =
2.2958874039497802890014385492622*10^(-41)
y_xs0 =
-0.001263317776974196724544154344118
```

(2)

```
x1=-10;x2=10;                               %
yx=@(x)(sin(x)^2*exp(-0.1*x)-0.5*sin(x)*(x+0.1)); %
[ xn0,fval,exitflag,output]=fminbnd(yx,x1,x2) % <9>
xn0 =
2.514797840754235
fval =
-0.499312445280039
exitflag =
1
output =
iterations: 13
funcCount: 14
algorithm: 'golden section search, parabolic interpolation'
message: [1x112 char]
```

(3)

```
xx=-10:pi/200:10;                           %
yxx=subs(y,x,xx);
plot(xx,yxx)
xlabel('x'),grid on
```

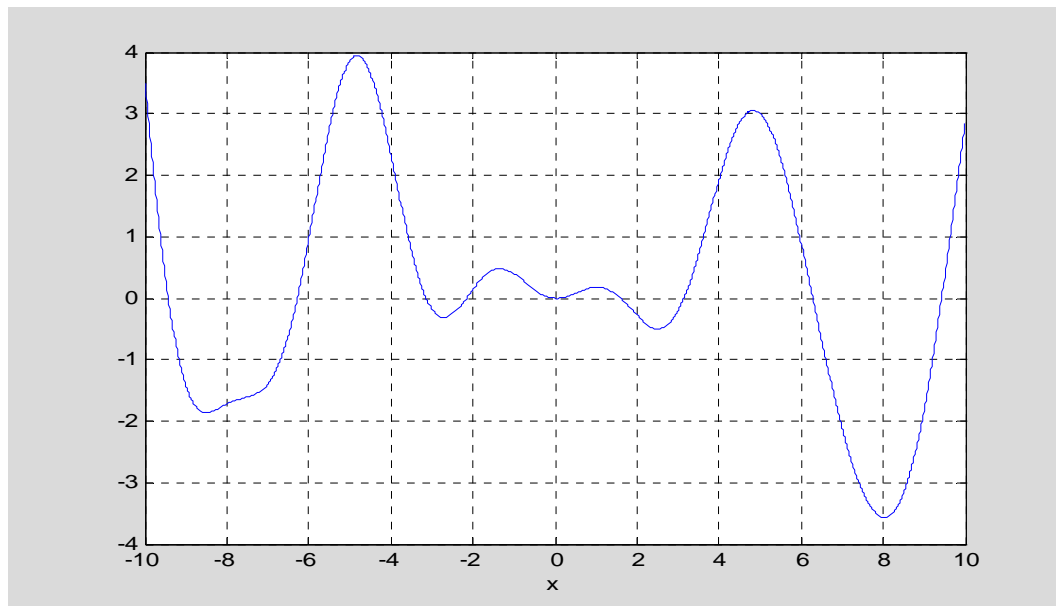


图 4.1-5 在[-10, 10]区间中的函数曲线

```
(4)
x11=6;x2=10;           %
yx=@(x)(sin(x)^2*exp(-0.1*x)-0.5*sin(x)*(x+0.1));
                           %
[xn00,fval,exitflag,output]=fminbnd(yx,x11,x2)           % <16>
                           %
xn00 =
    8.023562824723015
fval =
   -3.568014059128578
exitflag =
     1
output =
    iterations: 9
    funcCount: 10
    algorithm: 'golden section search, parabolic interpolation'
    message: [1x112 char]
```

【例 4.1-8】

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

(2)
format short g
x0=[-5,-2,2,5;-5,-2,2,5]; %提供 4 个搜索起点
[sx,sfval,sexit,soutput]=fminsearch(ff,x0)
                           %sx 给出一组使优化函数值非减的局部极小点

sx =
    0.99998    -0.68971    0.41507    8.0886
    0.99997    -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)
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
```

4.1.5 常微分方程的数值解

【例 4.1-9】

(1)

(3)

```
tspan=[0,30];           %
y0=[1;0];               %
[tt,yy]=ode45(@DyDt,tspan,y0);           %<3>
plot(tt,yy(:,1))
xlabel('t'),title('x(t)')
```

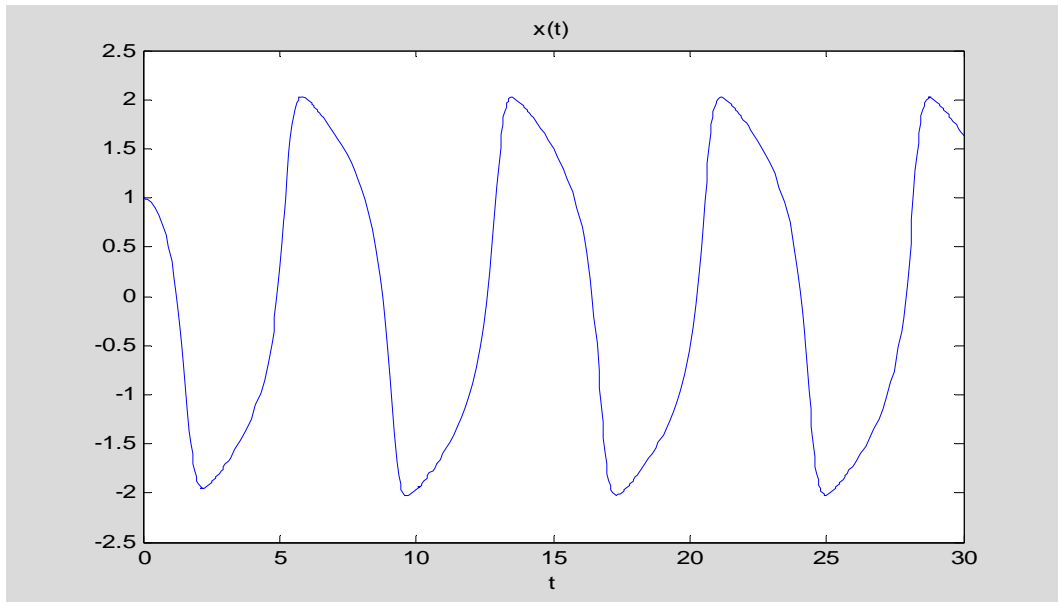


图 4.1-6 微分方程解

(4)

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

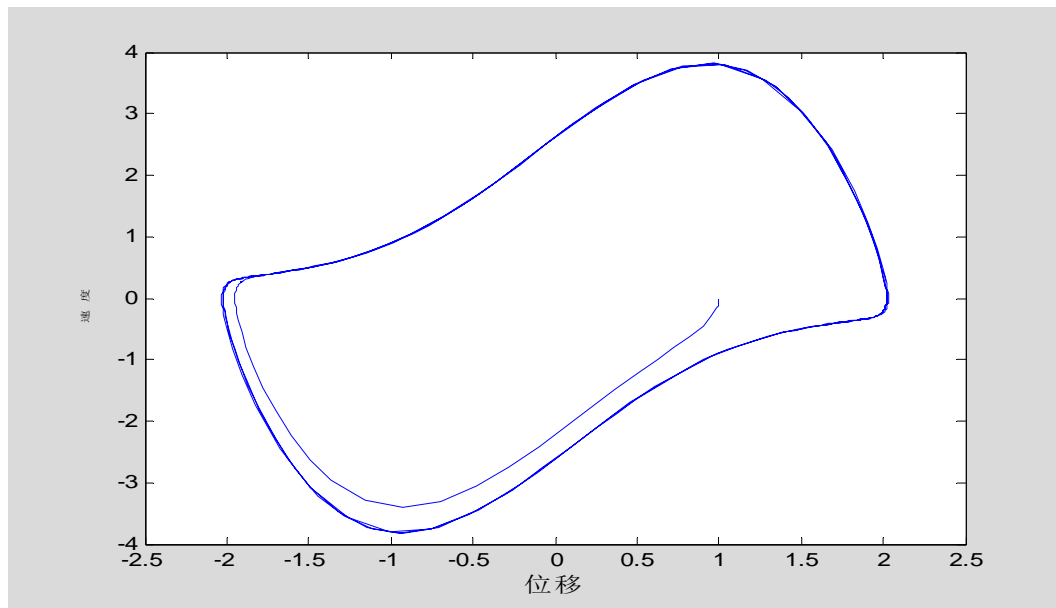


图 4.1-7 平面相轨迹

4.2 矩阵和代数方程

4.2.1 矩阵运算和特征参数

1 矩阵运算

【例 4.2-1】

(1)

```
clear
rand('twister',12)
A=rand(2,4);B=rand(4,3);
C1=zeros(size(A,1),size(B,2));           % <4>
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 =
    0.7337    0.8396    0.3689
    1.0624    1.1734    1.3787
```

(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 =
    0.7337    0.8396    0.3689
    1.0624    1.1734    1.3787
```

```

(3)
C3=A*B, %
C3 =
    0.7337    0.8396    0.3689
    1.0624    1.1734    1.3787

(4)
C3_C3=norm(C3-C1,'fro') %
C3_C2=norm(C3-C2,'fro') %
C3_C3 =
    0
C3_C2 =
    0

```

【例 4.2-2】

```

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      +    1i      25
B2 =
    2      13      +    1i
    13      -    1i      8
C1 =
    8      +    8i      7      +    13i
    7      +    13i      15      +    16i
C2 =
    0      +    2i      11      +    7i
    11      +    7i      0      +    8i

```

2 矩阵的标量特征参数

【例 4.2-3】

```

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】

```

format short      %
rand('twister',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 =
    2.6030
tBA =
    2.6030
tCD =
    4.1191
tDC =
    4.1191

d_A_B=det(A)*det(B)
dAB=det(A*B)
dBA=det(B*A)
d_A_B =
    0.0094
dAB =
    0.0094
dBA =
    0.0094

dCD=det(C*D)
dDC=det(D*C)      %
dCD =
    0.0424
dDC =
   -2.6800e-018

```

4.2.2 矩阵的变换和特征值分解

【例 4.2-5】

```

(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】

```

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
     -0.1776
     -0.2665
     -0.3553
     -0.5329
ans =
       1

```

【例 4.2-7】

(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 =
    6.7532e-016
err2 =
    4.4409e-016

```

4.2.3 线性方程的解

1 线性方程解的一般结论

【例 4.2-8】

```

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

(2)
ra=rank(A)           %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 =
    1.1374e-014

```

2 矩阵逆

【例 4.2-9】

(1)

```
randn('state',0);
A=gallery('randsvd',300,2e13,2); %
x=ones(300,1); %
b=A*x; %
cond(A) %
ans =
    1.9978e+013
```

(2)

```
tic %
xi=inv(A)*b; %
ti=toc %
eri=norm(x-xi) %
rei=norm(A*xi-b)/norm(b) %
ti =
    0.0294
eri =
    0.1003
rei =
    0.0053
```

(3)

```
tic;
xd=A\b; %
td=toc
erd=norm(x-xd)
red=norm(A*xd-b)/norm(b)
td =
    0.0134
erd =
    0.0939
red =
    8.4835e-015
```

4.2.4 一般代数方程的解

【例 4.2-10】

(1)

```
syms t
ft=sin(t)^2*exp(-0.1*t)-0.5*abs(t);
S=solve(ft,t) %
ftS=subs(ft,t,S) %
S =
matrix([[0]])
ftS =
0
```

<3>

(2)

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

(B)

```

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

```

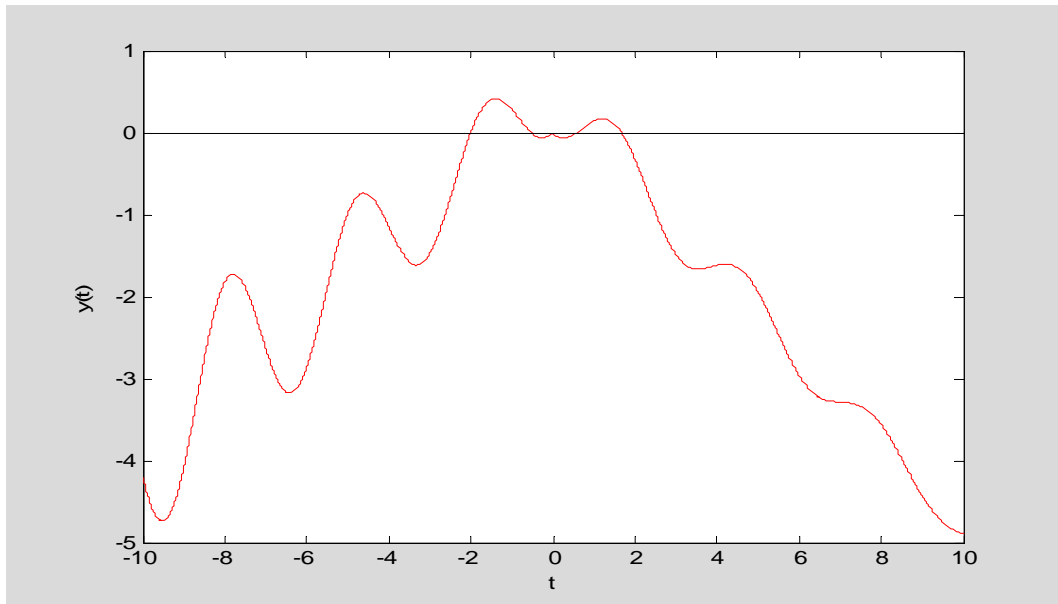


图 4.2-1 函数零点分布观察图

(C)

```

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

```

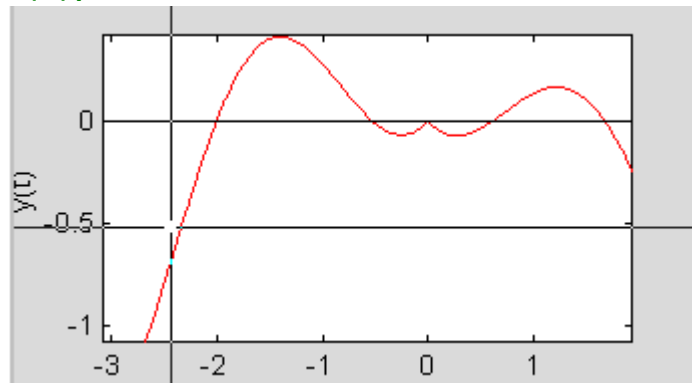


图 4.2-2 局部放大和利用鼠标取值图

```

tt %
tt =

```

```

-2.0039
-0.5184
-0.0042
0.6052
1.6717

```

(D)

```

[t4,y4]=fzero(y_C,0.1) % <17>
t4 =

```

```

0.5993
y4 =
1.1102e-016

```

4.3 概率分布和统计分析

1 二项分布(Binomial distribution)

【例 4.3-1】

```

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

```

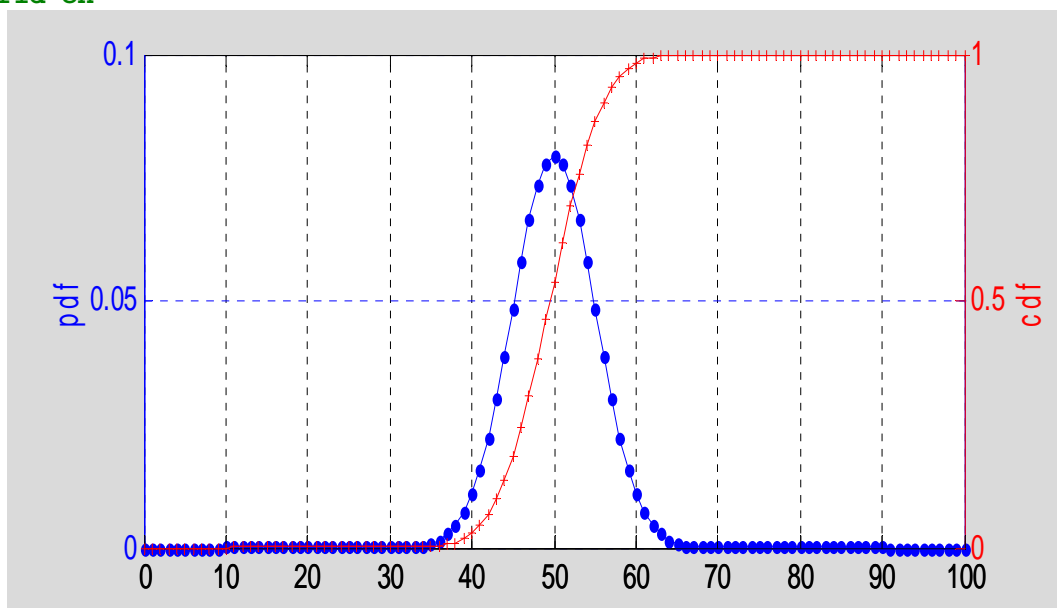


图 4.3-1 二项分布 B(100, 0.5)的概率和累计概率曲线

2 正态分布 (Normal distribution)

【例 4.3-2】

```

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

```

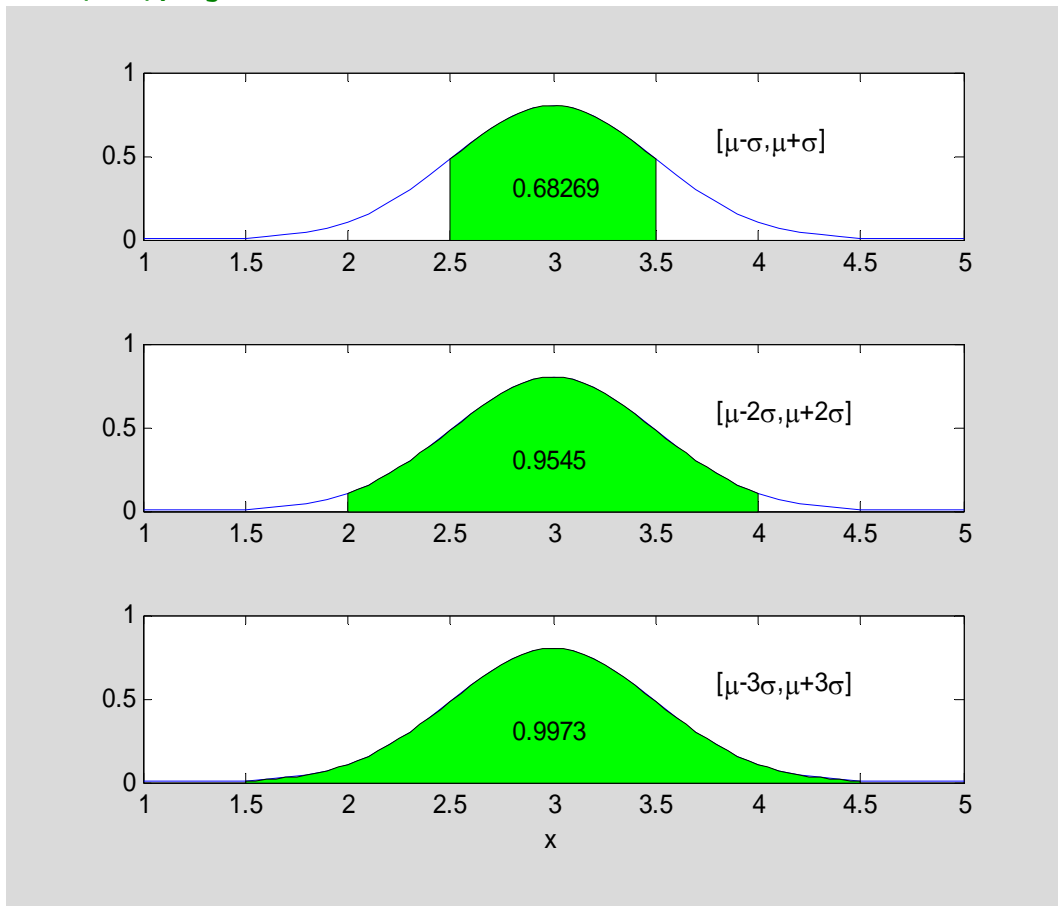


图 4.3-2 均值两侧一、二、三倍标准差之间的概率

3 各种概率分布的交互式观察界面

【例 4.3-3】

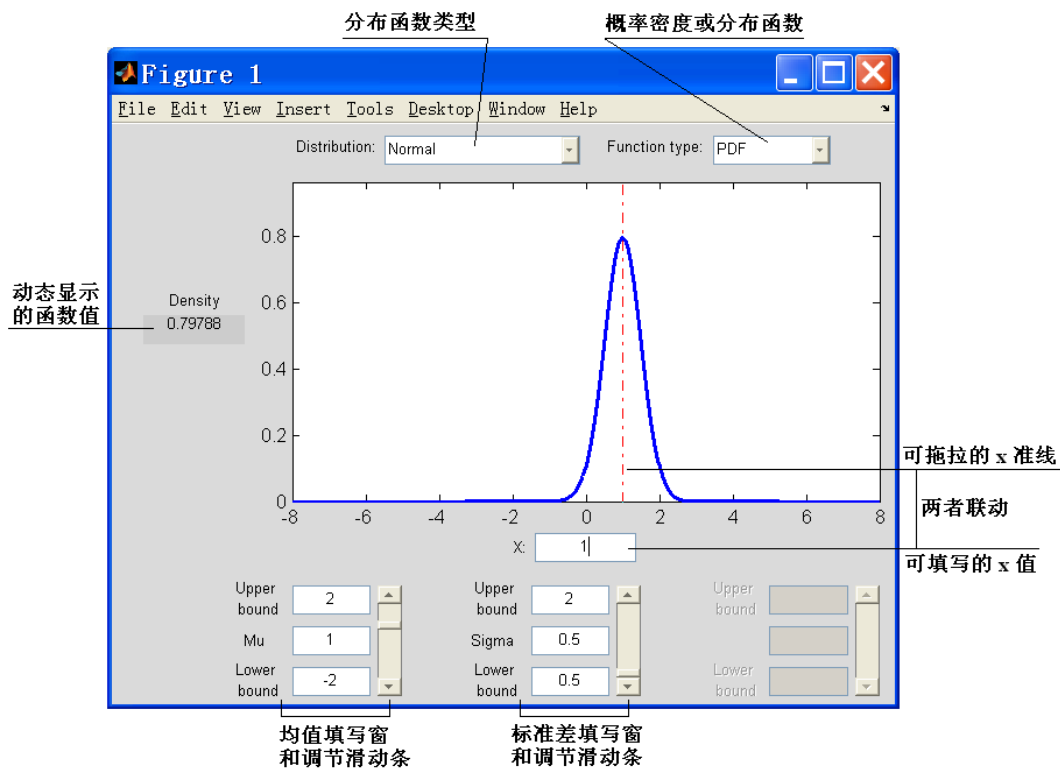


图 4.3-3 概率分布交互界面

4.3.2 随机数发生器和 统计分析指令

【例 4.3-4】

```

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 =
    1.0e-014 *
    -0.0111    -0.1554    -0.0888         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】

```
mu=2;s=0.5;
randn('state',22)
x=randn(1000,1);      %      <3>
y=s*x+mu;             %      <4>
z=s*(x+mu);           %      <5>
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')
```

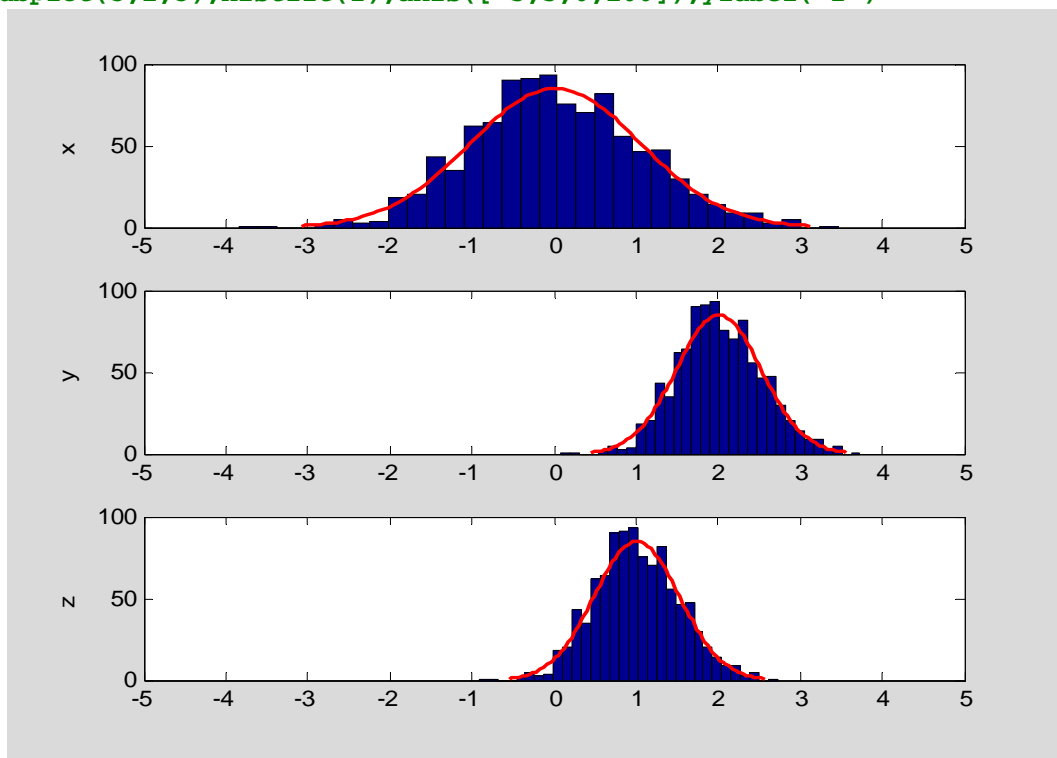


图 4.3-4 均值为 2 标准差为 0.5 的随机数样本 z

4.4 多项式运算和卷积

1 多项式表达方式的约定

2 多项式运算函数

【例 4.4-1】

(1)

```
format rat %
```

```

p1=conv([1,0,2],conv([1,4],[1,1]));      %          <2>
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')]) %<6>
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+r;                            %          <8>
pp1==p1                                %          <9>
ans =
      1      1      1      1      1

```

【例 4.4-2】

```

(1)
format short
A=[11 12 13;14 15 16;17 18 19];        %
PA=poly(A)                             %          <2>
PPA=poly2str(PA,'s')                   %
PA =
    1.0000   -45.0000   -18.0000     0.0000
PPA =
    s^3 - 45 s^2 - 18 s + 1.6206e-014

```

```

(2)
s=eig(A)
r=roots(PA)
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);  %          <7>
AA(1,:) = -PA(2:n) ./ PA(1);           %          <8>
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】

```

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】

(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 % <5>
Va=polyval(p,X)
va =
     6    11
    18    27
Va =
     6    11
    18    27
```

(3)

```
vm=X^2+2*X+3*eye(2) % <7>
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
```

4.4.2 多项式拟合和最小二乘法

1 多项式拟合

【例 4.4-5】

```

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

```

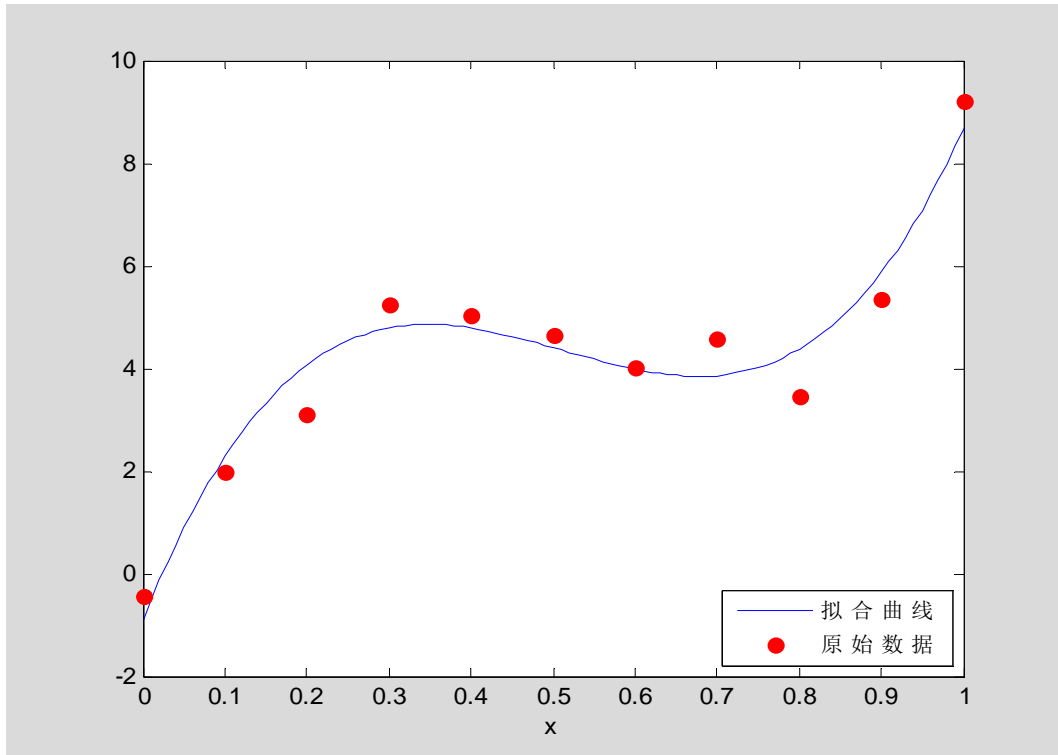


图 4.4-1 采用三阶多项式所得的拟合曲线

2 最小二乘问题

【例 4.4-6】

```

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

```

4.4.3 两个有限长序列的卷积

【例 4.4-7】

(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 =
    Columns 1 through 13
         5         6         7         8         9        10        11        12        13        14        15        16
17
    Columns 14 through 17
        18        19        20        21
cc =
    Columns 1 through 13
         1         2         3         4         5         6         7         8         8         8         7         6
5
    Columns 14 through 17
         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 =
    Columns 1 through 13
         0         1         2         3         4         5         6         7         8         9        10        11
12
    Columns 14 through 22
        13        14        15        16        17        18        19        20        21
c =
    Columns 1 through 13
         0         0         0         0         0         1         2         3         4         5         6         7
8
    Columns 14 through 22
         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 =
    Columns 1 through 13
         5         6         7         8         9        10        11        12        13        14        15        16
17
    Columns 14 through 17
        18        19        20        21
C =
    Columns 1 through 13
         1         2         3         4         5         6         7         8         8         8         7         6
5
    Columns 14 through 17
         4         3         2         1

```

(4)

```

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')

```

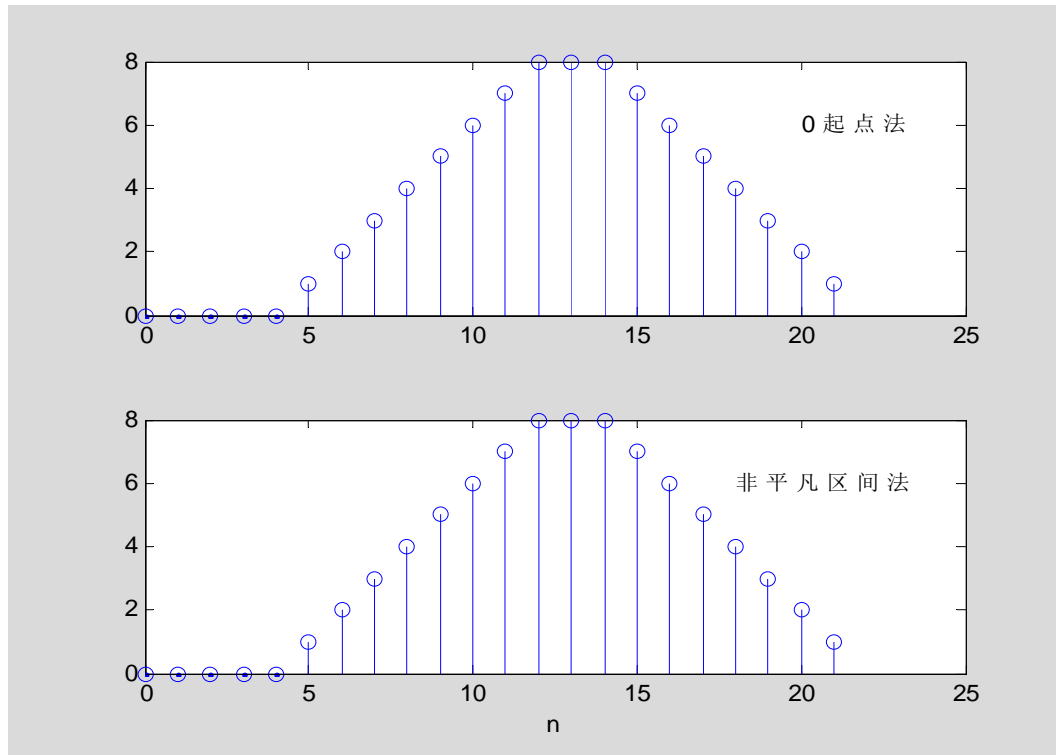


图 4.4-3 借助 conv 指令时两种不同序列记述法所得的卷积序列

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）（提示：自变量 t 采样间距太小。）

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 $y(4.5)$ 。
(提示: cumtrapz 快捷, 在精度要求不高处可用; quad 也可试。巧用 find。)
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。(提示: 各种数值法均可试。)
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} 。(体验: 试用 trapz, 如何算得同样精度的积分。)
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。
(提示: 作图观察。)
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 $y(t)|_{t=0.5}$ 。
(提示: 注意 ode45 和 dsolve 的用法。)
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: 方法很多; 建议使用 rref 体验。)
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。(提示: condeig)
9. 求矩阵 $Ax = b$ 的解, A 为 3 阶魔方阵, b 是 (3×1) 的全 1 列向量。(提示: 用 rref, inv, / 体验。)
10. 求矩阵 $Ax = b$ 的解, A 为 4 阶魔方阵, b 是 (4×1) 的全 1 列向量。(提示: 用 rref, inv, / 体验。)
11. 求矩阵 $Ax = b$ 的解, A 为 4 阶魔方阵, $b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ 。(提示: 用 rref, inv, / 体验。)
12. 求 $-0.5 + t - 10e^{-0.2t} |\sin[\sin t]| = 0$ 的实数解。(提示: 发挥作图法功用)
13. 求解二元函数方程组 $\begin{cases} \sin(x - y) = 0 \\ \cos(x + y) = 0 \end{cases}$ 的解。(提示: 可尝试符号法解; 试用 contour 作图求解; 比较之。此题有无数解。)
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。(提示: 二项式分布概率指令 binopdf; stem)
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?(提示: 体验 normrnd; 理解 hist(Y, m) 指令格式。)
16. 从数据文件 prob_data416.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。

$$Mx = \max(\max(R)), Me = \text{mean}(\text{mean}(R)), St = \text{std}(\text{std}(R)),$$
 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。(提示: load; R(:)。)
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。(提示: 采用范数指令 norm 验算。)
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。(提示: load, polyfit, polyval)

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。试画出类似图 p4-1 所示的系统输入、输出信号图形。（提示：注意输入信号尾部的处理；NaN 的使用。）

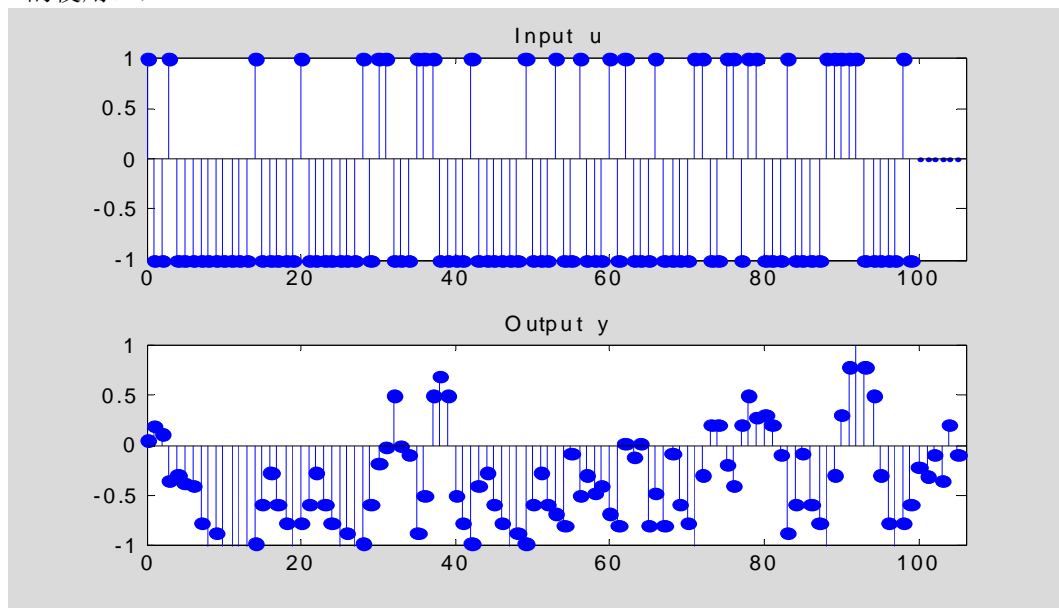


图 p4-1

第 5 章 数据和函数的可视化

5.1 引导

5.1.1 离散数据和离散函数的可视化

【例 5.1-1】

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

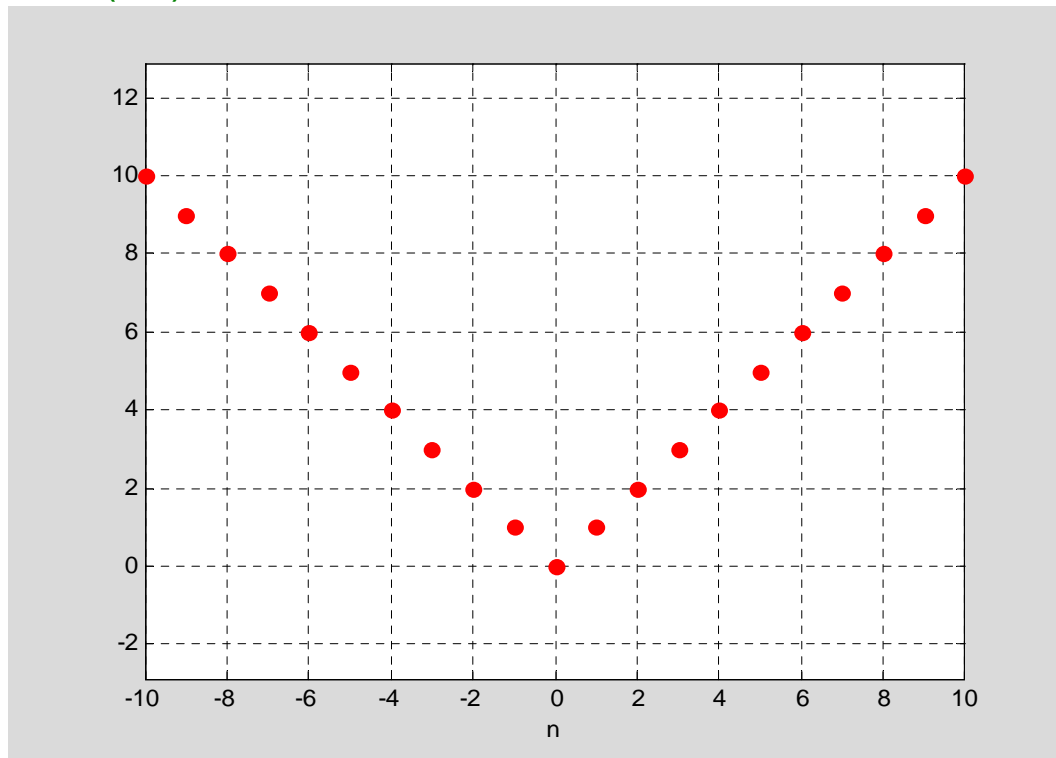


图 5.1-1 离散函数的可视化

5.1.2 连续函数的可视化

【例 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.')
```

```

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

```

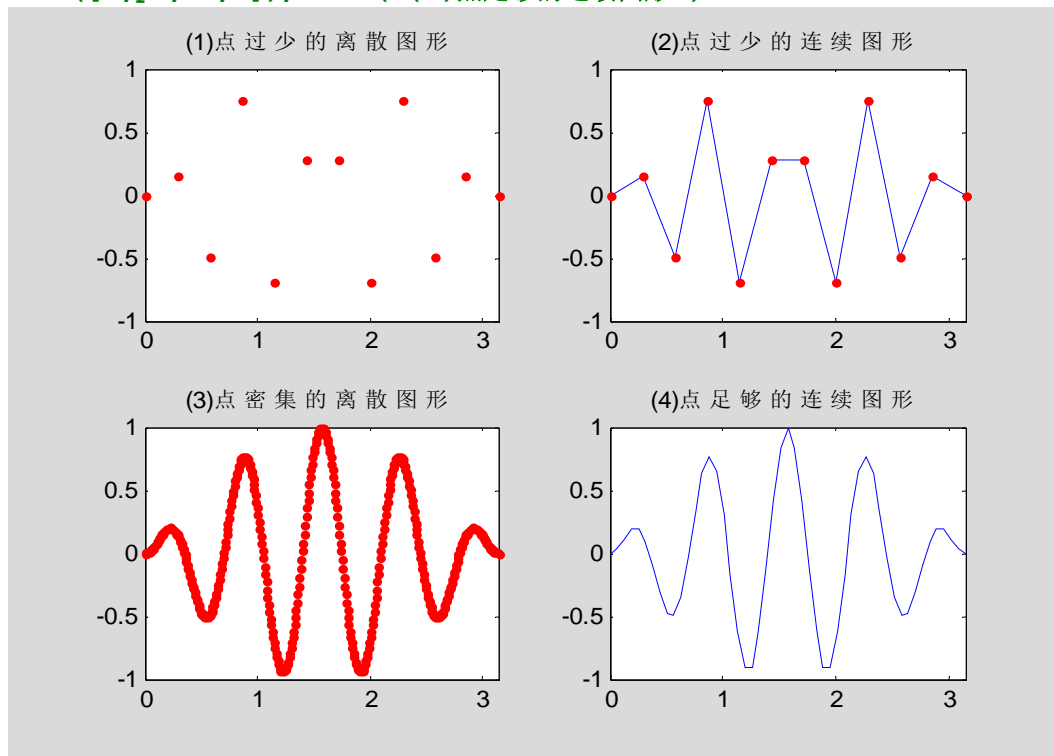


图 5.1-2 连续函数的图形表现方法

【例 5.1-3】

```

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

```

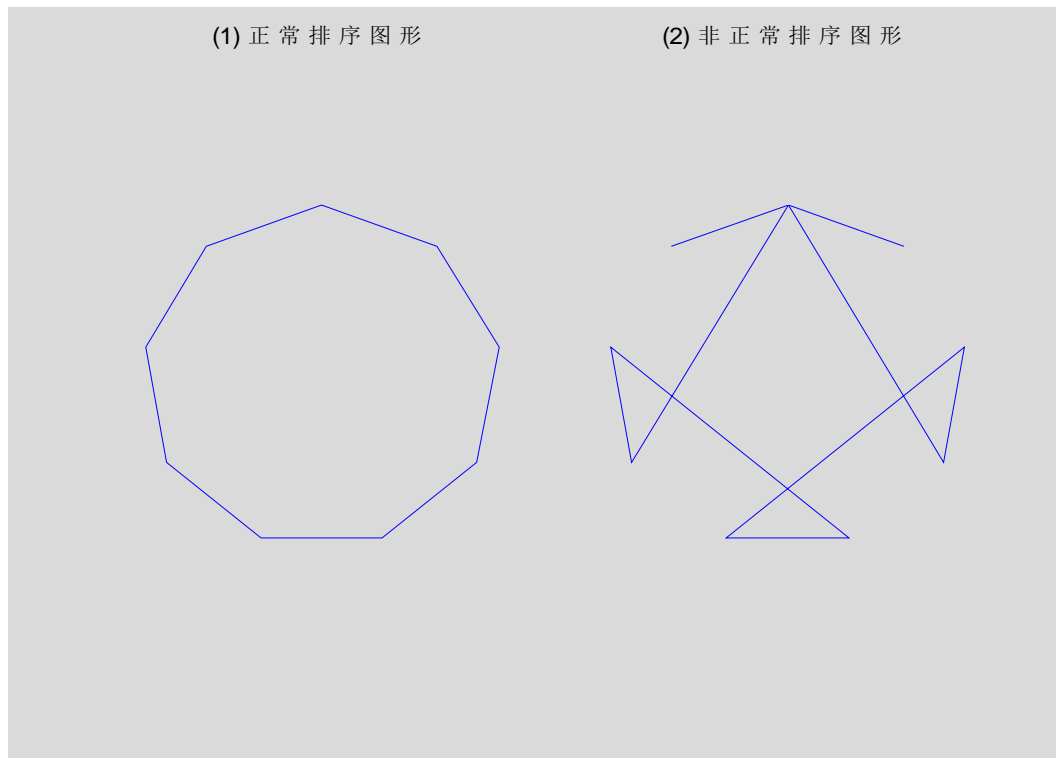



图 5.1-3 自变量排列次序对连续曲线图形的影响

5.2 二维曲线和图形

5.2.1 二维曲线绘制的基本指令 plot

【例 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')
```

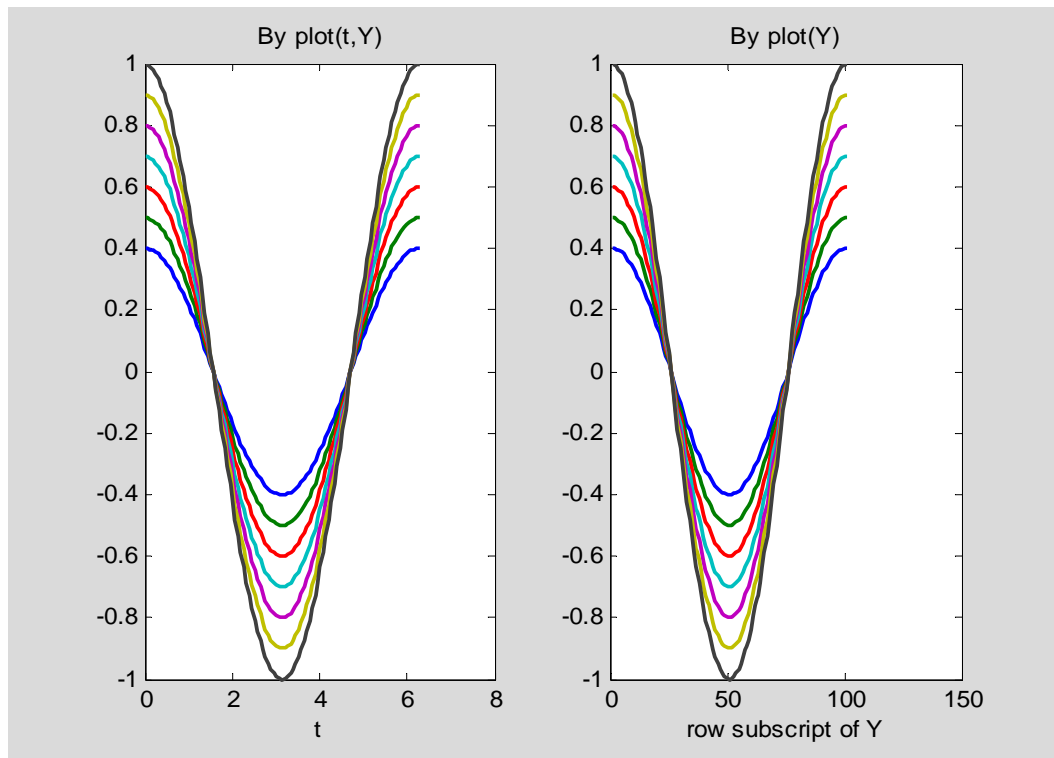


图 5.2-1 plot(t,Y)与 plot(Y)所绘曲线的区别

【例 5.2-2】

```
t=(0:pi/100:pi)';
y1=sin(t)*[1,-1];
y2=sin(t).*sin(9*t);
t3=pi*(0:9)/9;
y3=sin(t3).*sin(9*t3);
plot(t,y1,'r:',t,y2,'-bo')
hold on
plot(t3,y3,'s','MarkerSize',10,'MarkerEdgeColor',[0,1,0],'MarkerFaceC
olor',[1,0.8,0])
axis([0,pi,-1,1])
hold off
%plot(t,y1,'r:',t,y2,'-bo',t3,y3,'s','MarkerSize',10,'MarkerEdgeColor
',[0,1,0],'MarkerFaceColor',[1,0.8,0])
```

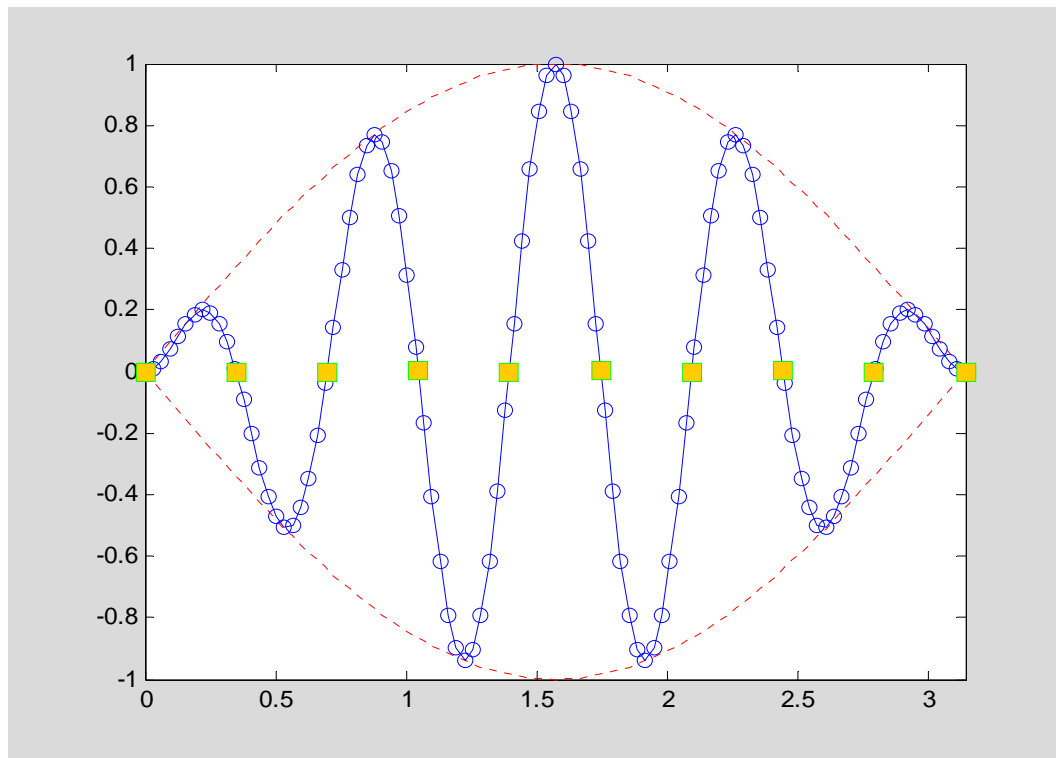


图 5.2-2 属性控制下所绘曲线

5.2.2 坐标控制和图形标识

【例 5.2-3】

```
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')
```

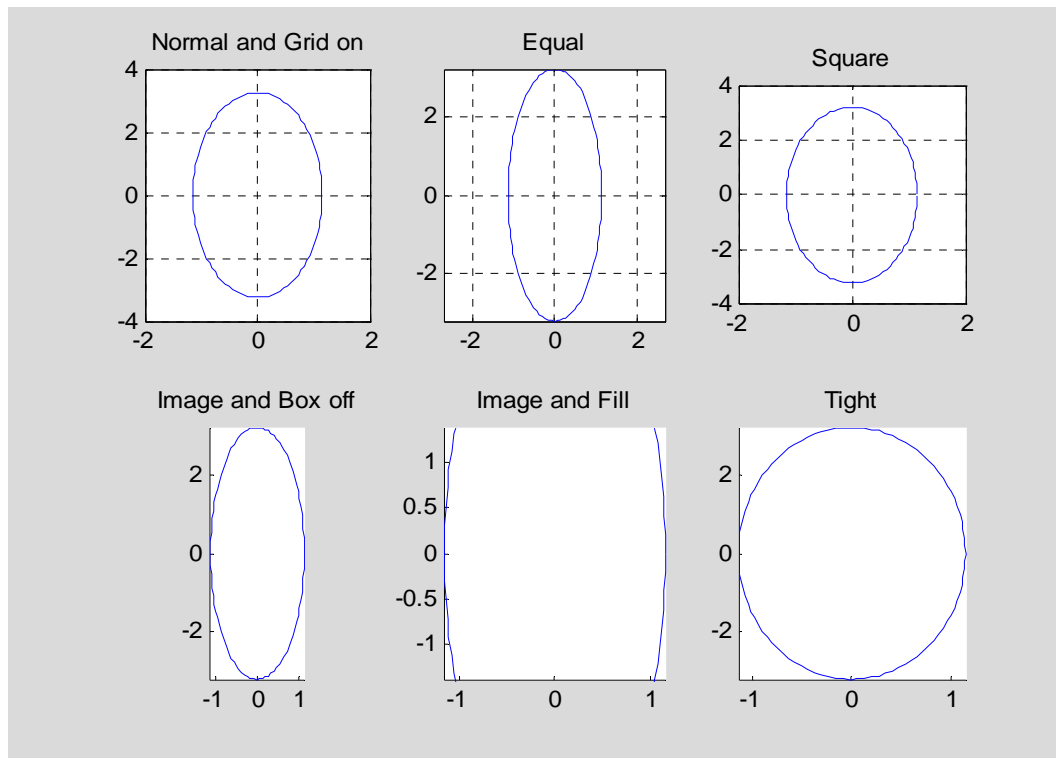


图 5.2-3 各种轴控制指令的不同影响

【例 5.2-4】

```

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

```

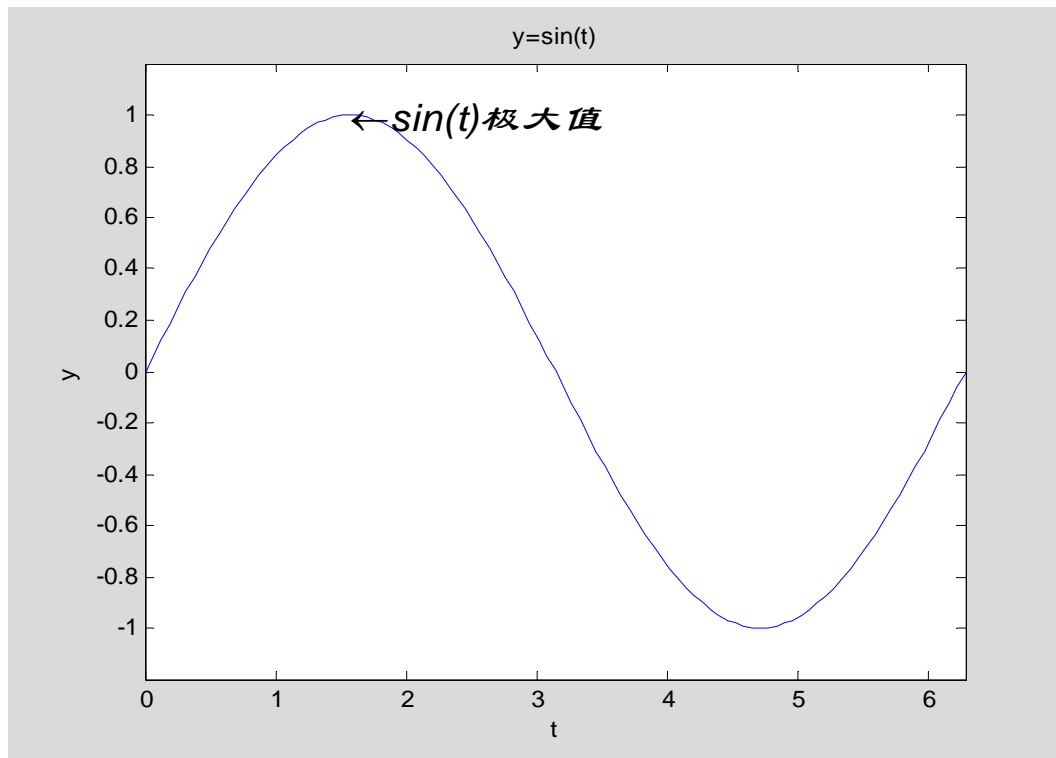


图 5.2-4 试验标识的图形

【例 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)
hold on
tt=t(find(abs(y-1)>0.05));ts=max(tt);
plot(ts,0.95,'bo','MarkerSize',10)
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)])
set(gca,'XtickLabel',{'2*pi','4*pi','6*pi'})
set(gca,'YtickLabel',{'0.95','1','1.05','max(y)'})
grid on
text(13.5,1.2,'\fontsize{12}{\alpha}=0.3')
text(13.5,1.1,'\fontsize{12}{\omega}=0.7')
cell_string{1}='\fontsize{12}\uparrow';
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)];
text(ts,0.85,cell_string,'Color','b','HorizontalAlignment','Center')
title('\fontsize{14}\it y = 1 - e^{\alpha t}cos{\omegat}')
xlabel('\fontsize{14} \bft \rightarrow')
ylabel('\fontsize{14} \bfy \rightarrow')

```

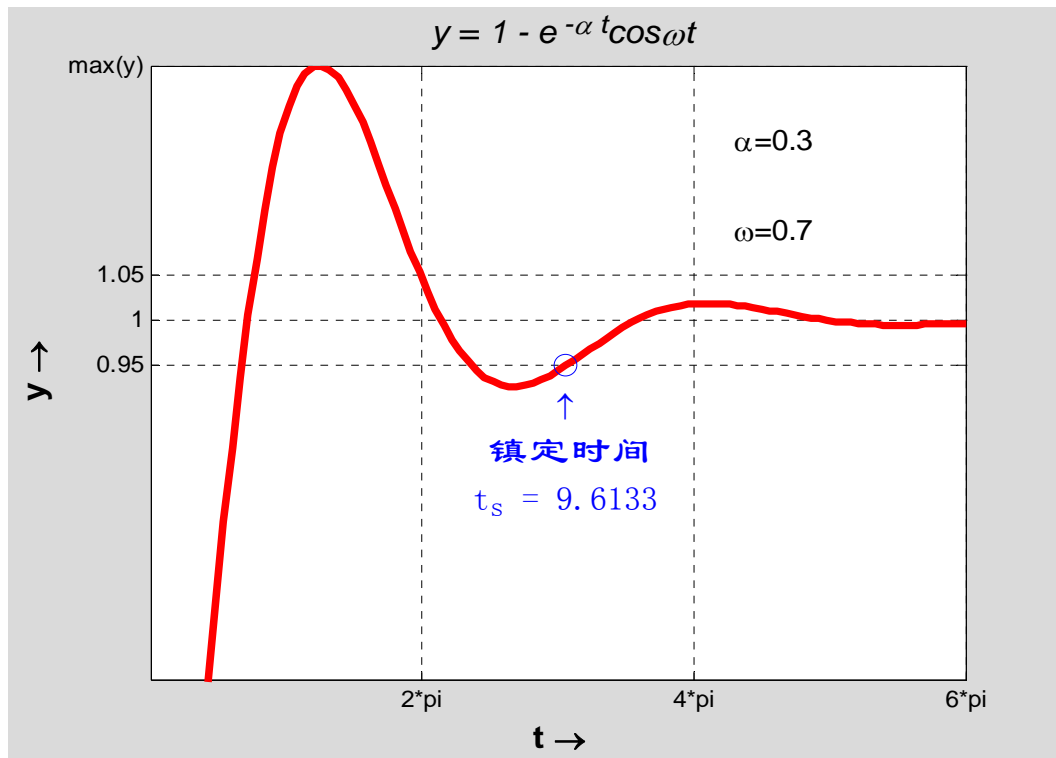


图 5.2-5 二阶阶跃响应图的标识

5.2.3 多次叠绘、双纵坐标和多子图

【例 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
```

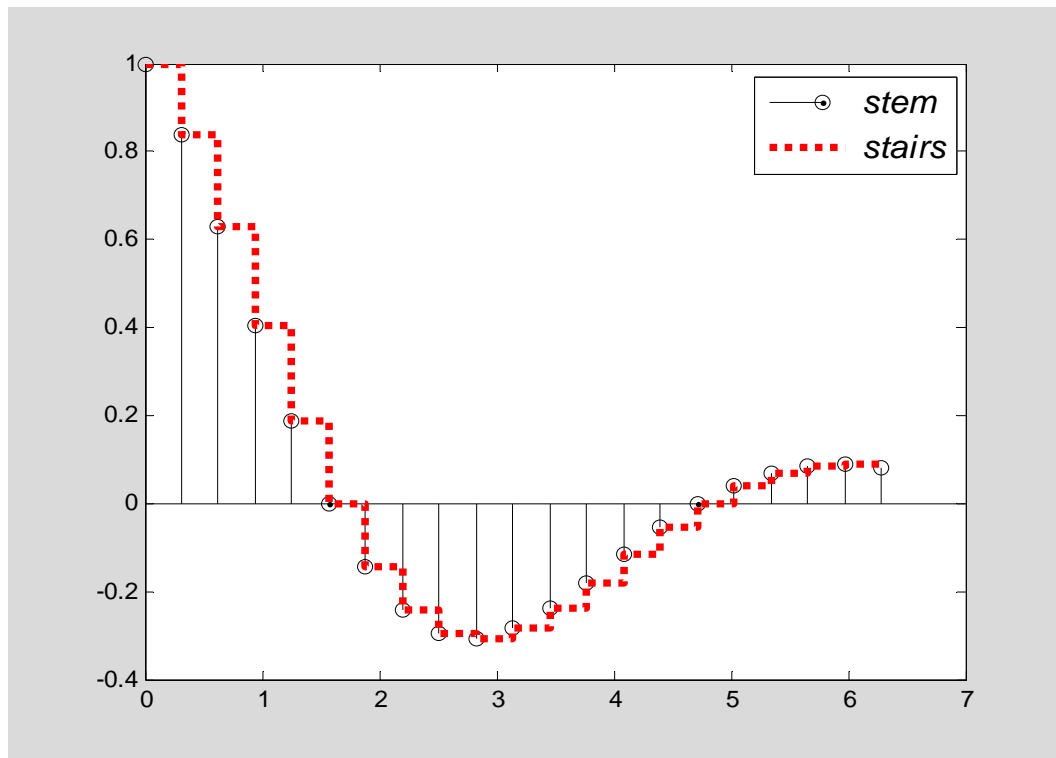


图 5.2-6 离散信号的重构

【例 5.2-7】

```

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

```

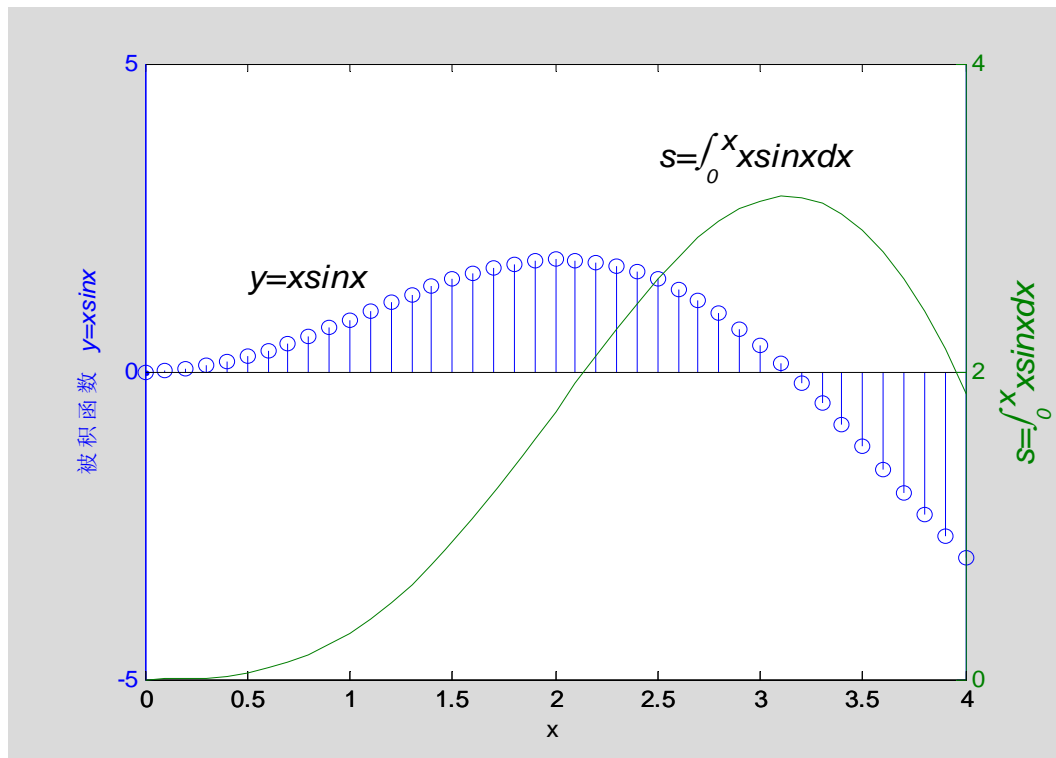


图 5.2-7 函数和积分

【例 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',[0.2,0.1,0.6,0.40])
plot(t,y12,'b-',t,[y1,-y1],'r:')
axis([0,pi,-1,1])
```

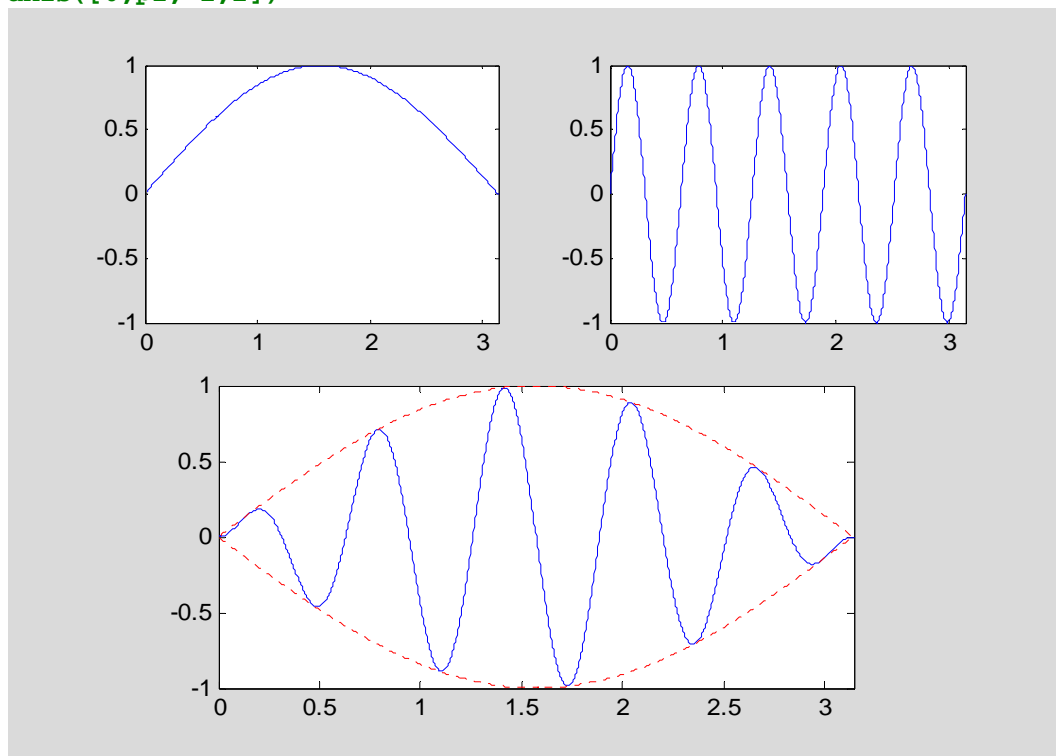


图 5.2-8 多子图的布置

5.2.4 获取二维图形数据的指令 ginput

【例 5.2-9】

```
clf  
x=0:0.01:1;  
y=(x+2).^x-2;  
plot(x,y);  
grid on
```

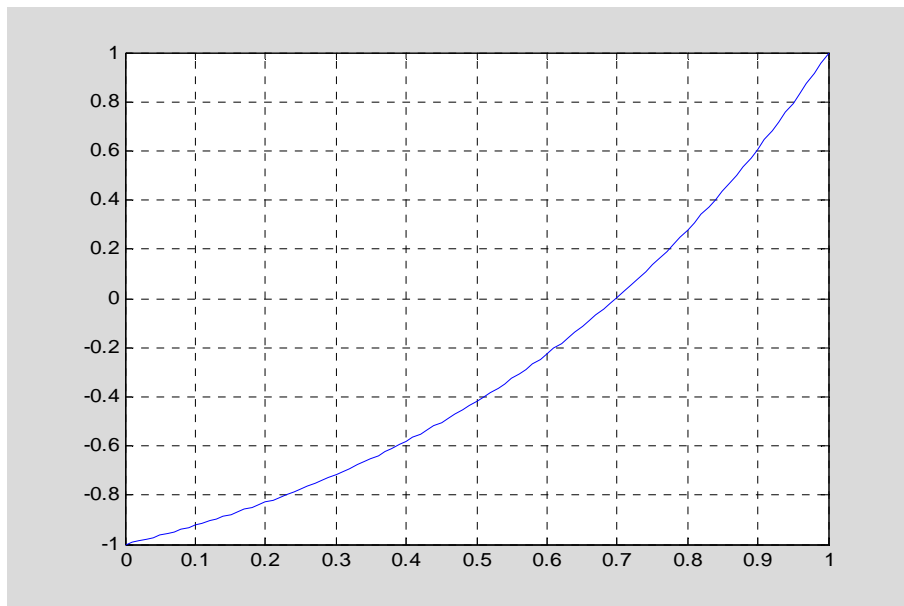


图 5.2-9 经多次试探后画出的图形

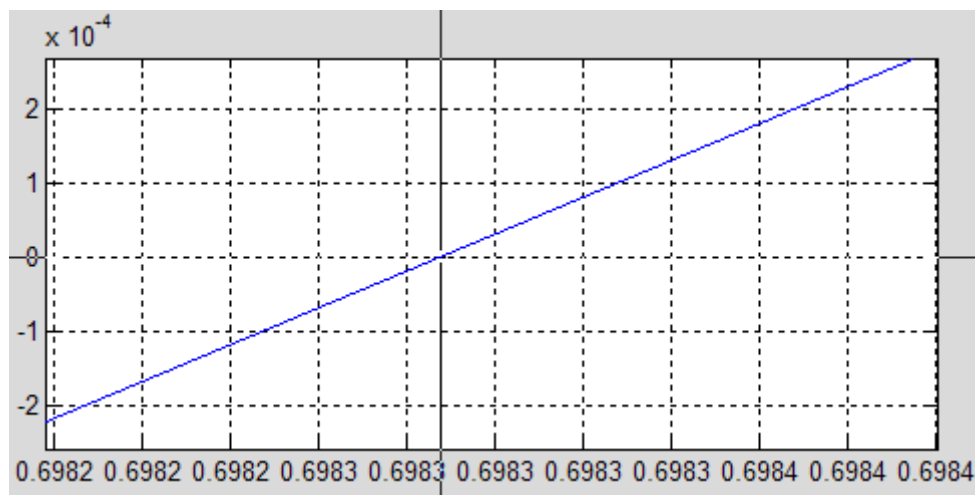


图 5.2-10 局部放大图

```
[x,y]=ginput(1);  
format long  
x,y  
x =  
    0.69828692903537  
y =  
   -5.884401711382421e-010
```

5.3 三维曲线和曲面

5.3.1 三维线图指令 plot3

【例 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','Location','best')
```

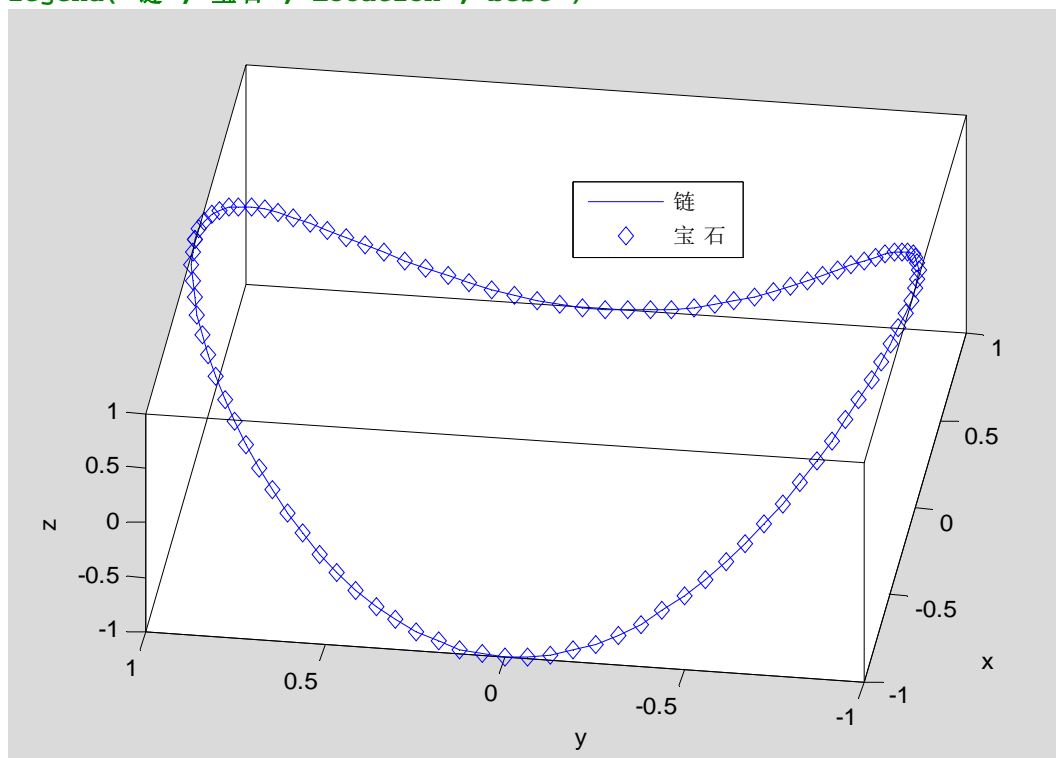


图 5.3-1 宝石项链

5.3.2 三维曲面/网线图

【例 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])
```

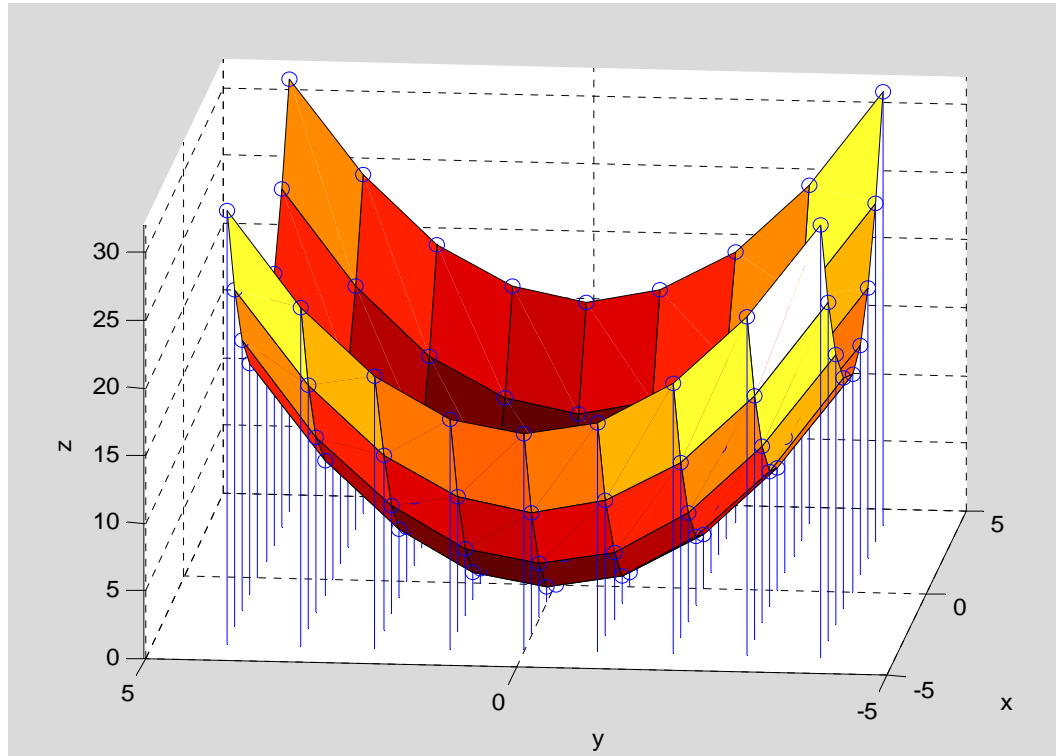


图 5.3-2 曲面图和格点

5.3.3 曲面/网线图的精细修饰

【例 5.3-3】

```
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')
```

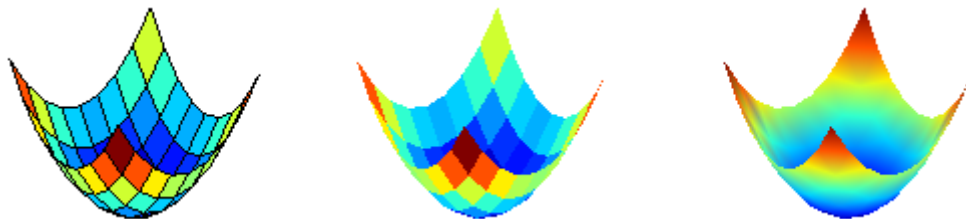


图 5.3-4 三种浓淡处理方式 **faceted/flat/interp** 的效果比较

【例 5.3-4】

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

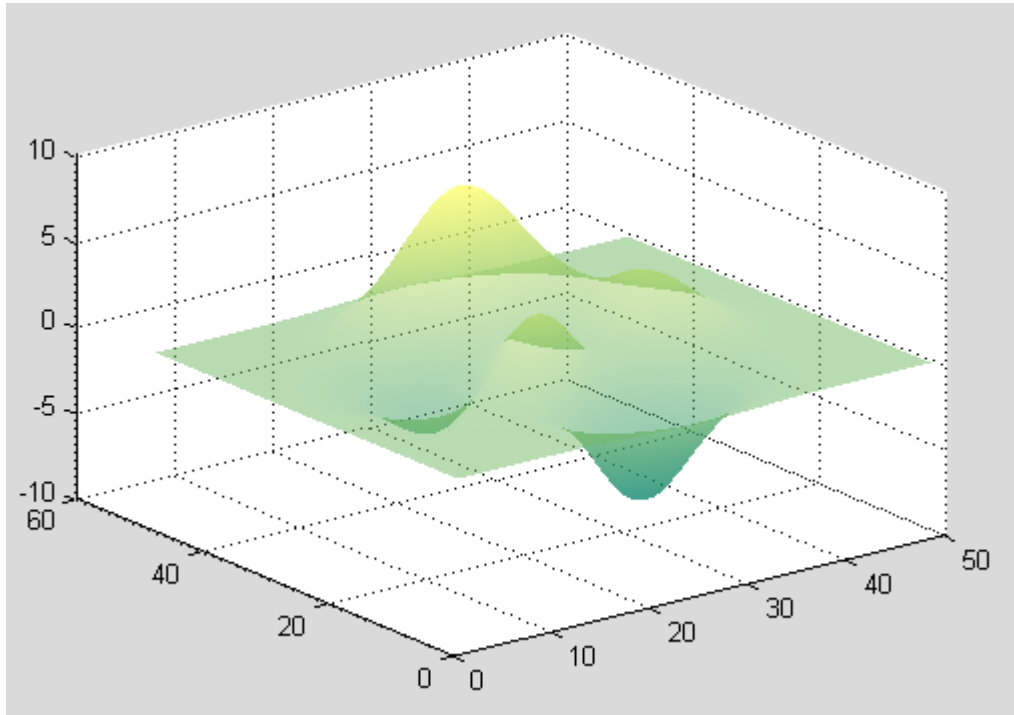


图 5.3-5 半透明薄膜

【例 5.3-5】

```
clf;
[X,Y,Z]=sphere(40);
colormap(jet)
subplot(1,2,1),surf(X,Y,Z),axis equal off,shading interp
light('position',[0 -10 1.5],'style','infinite')
lighting phong
material shiny
subplot(1,2,2),surf(X,Y,Z,-Z),axis equal off,shading flat
light;lighting flat
light('position',[-1,-1,-2],'color','y')
light('position',[-1,0.5,1],'style','local','color','w')
set(gcf,'Color','w')
```

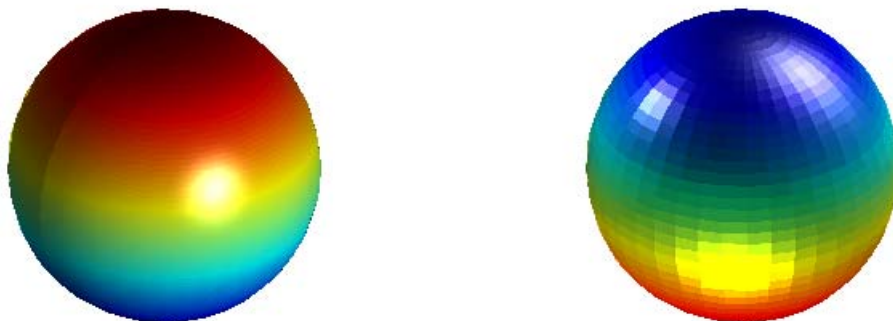


图 5.3-6 灯光、照明、材质指令所表现的图形

5.3.4 透视、镂空和裁切

【例 5.3-6】

```
[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
```

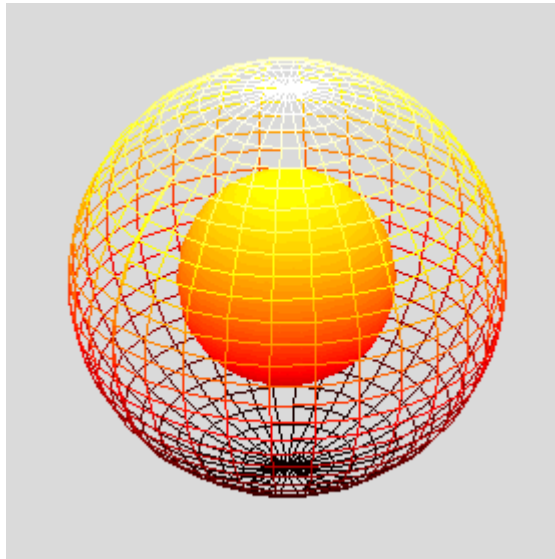


图 5.3-7 剔透玲珑球

【例 5.3-7】

```
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
```

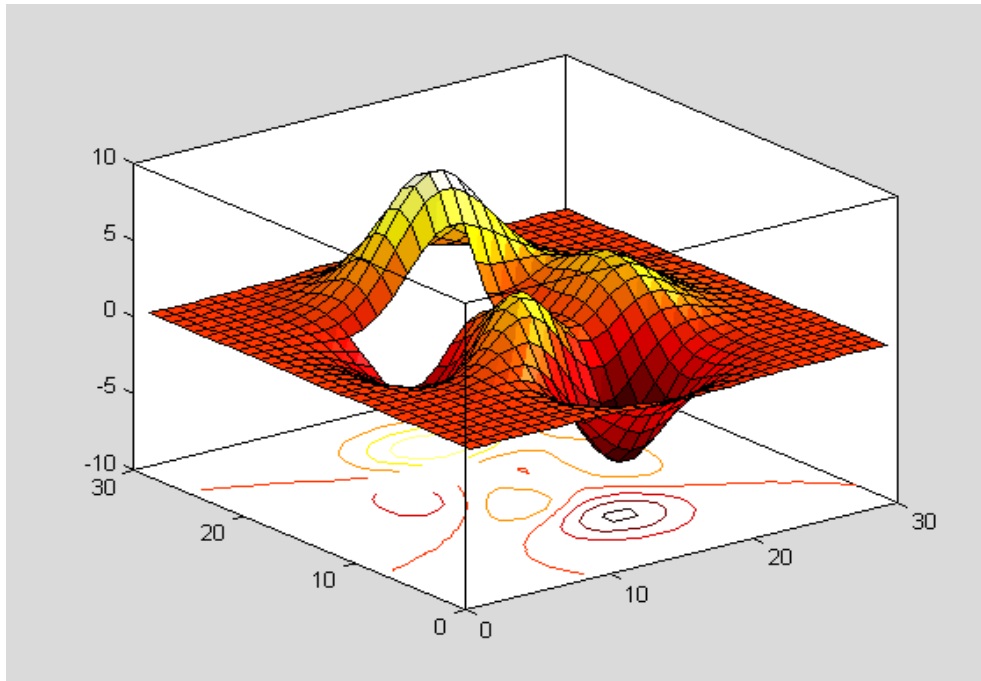


图 5.3-8 镂空方孔的曲面

【例 5.3-8】

```
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])
```

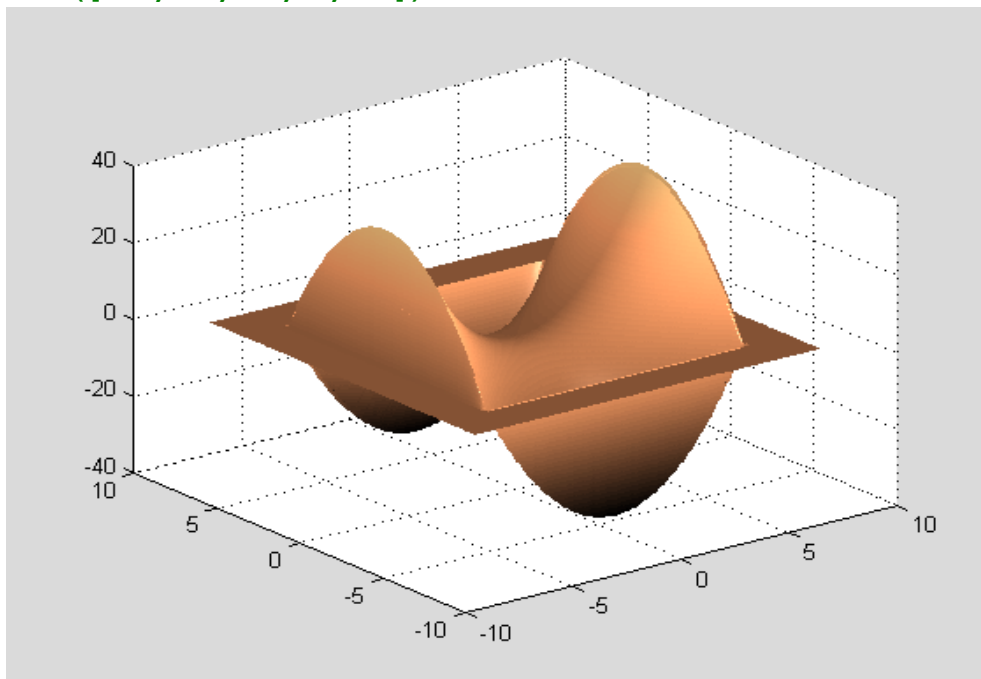


图 5.3-9 经裁切处理后的图形

5.4 高维可视化

5.4.1 二维半图指令 pcolor, contour, contourf

【例 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')
```

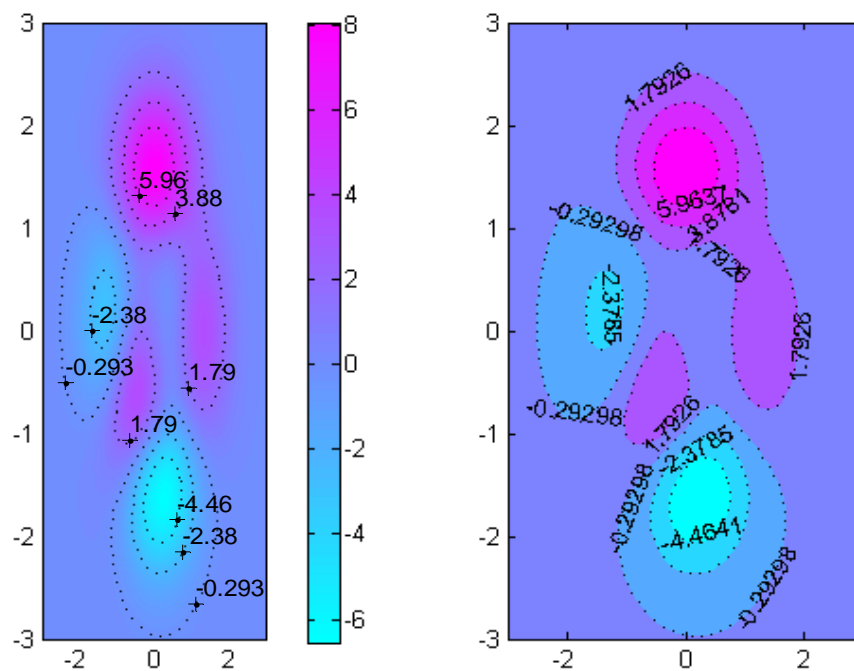


图 5.4-1 “二维半”指令的演示

5.4.2 四维表现

【例 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);
dz2=del2(Z);
```

```

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))')

```

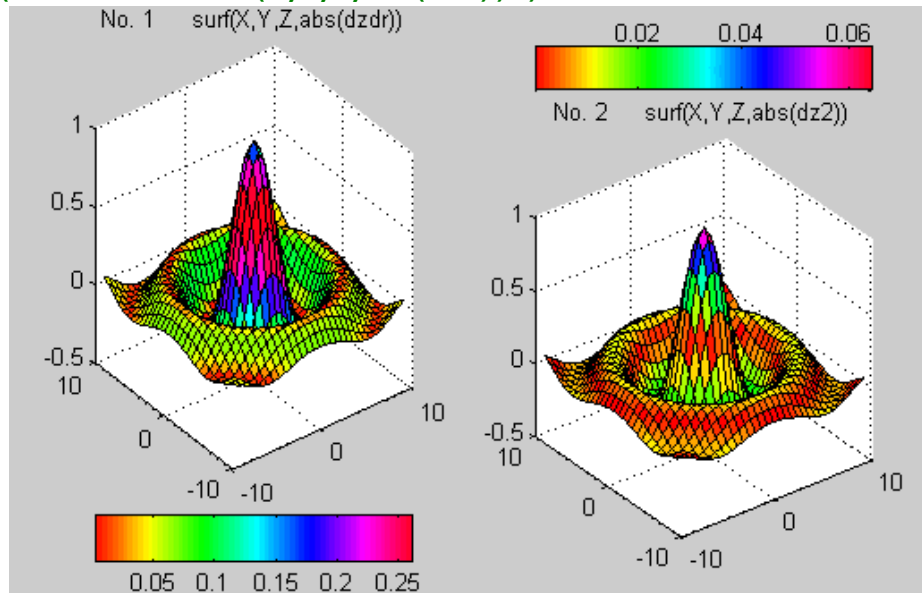


图 5.4-2 色彩分别表现函数的径向导数和曲率特征

【例 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)

```

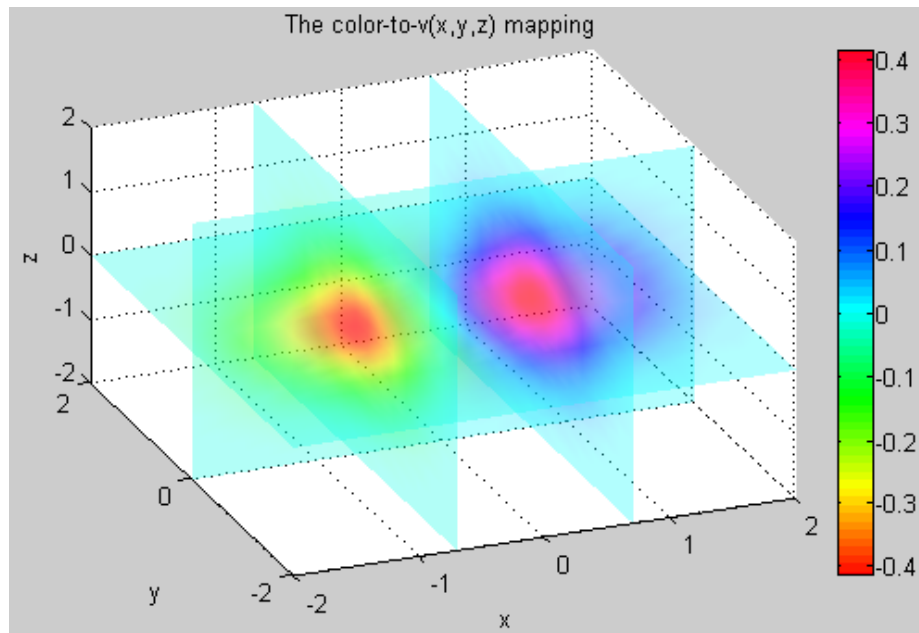



图 5.4-3 切片图

【例 5.4-4】
`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`

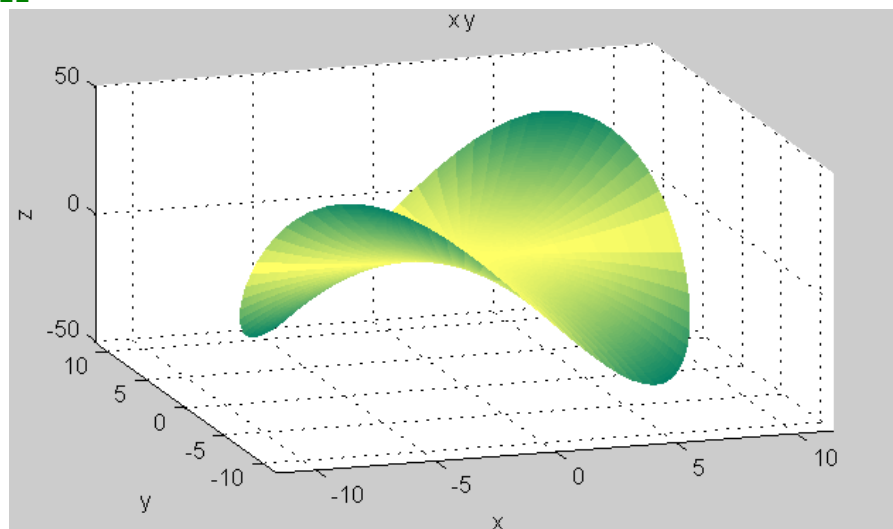


图 5.4-4 用于色图变幻演示的图形

【例 5.4-5】
`ezsurf('x*y','circ');shading flat;view([-18,28])`
`C=summer;`
`CC=[C;flipud(C)];`
`colormap(CC)`
`spinmap(30,4)`

【例 5.4-6】
 (1)
`clf`
`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
close

```

```

(2)
shg,axis off
movie(mmm,5,10)

```

【例 5.4-7】

(1)

```

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];
n=length(x);
if nargin<2
    ki=fix(n/2);
end
y=[y1 y2 y3 y4];z=[z1 z2 z3 z4];
shg
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');
KK=K*n;
text(-1,-0.85,-36,'倒计时')
KK=KK-1;
htext=text(-1,-1,-40,int2str(KK));
%
i=2;j=1;
while 1
    set(h,'xdata',x(i),'ydata',y(i),'zdata',z(i));
    drawnow; % <23>
    pause(0.0005) % <24>
    i=i+1;
    KK=KK-1;
    set(htext,'string',int2str(KK)) % <27>
    if nargin==2 && narginout==1 %
        if(i==ki&&j==1);f=getframe(gcf);end % <29>
    end
    if i>n
        i=1;j=j+1;
        if j>K;break;end
    end
end
end

```

(2)

```

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

```

(3)

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

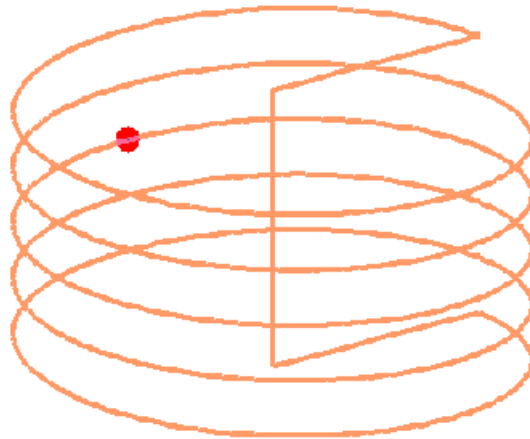


图 5.4-5 红球沿下旋螺线运动的瞬间照片

5.5 图形窗功能简介

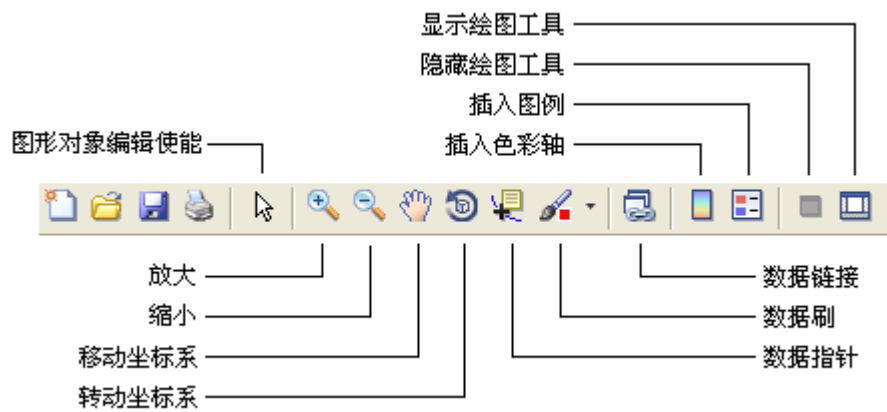


图 5.5-1 图形窗工具条专用按键

【例 5.5-1】

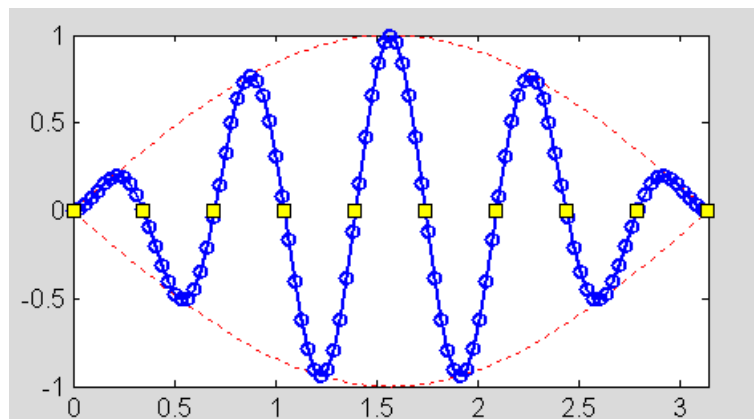


图 5.5-2 个性化的图形

(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)

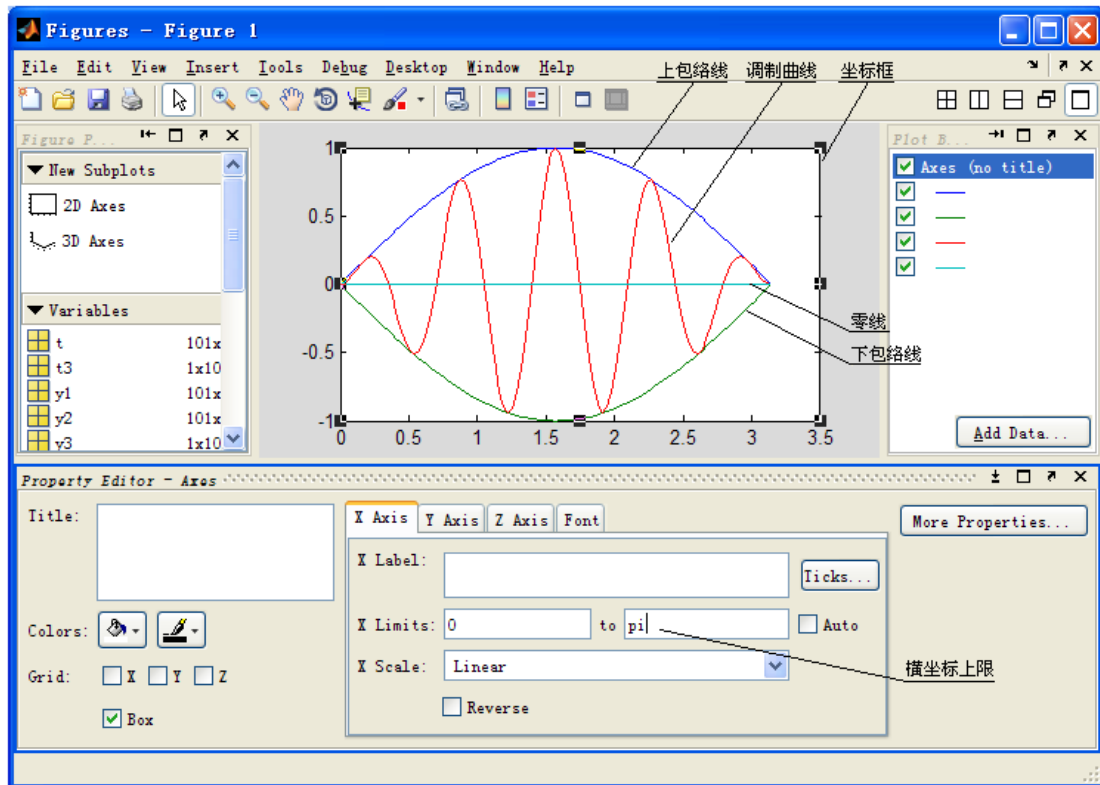


图 5.5-3 编辑工作模式下的图形窗界面

(3)

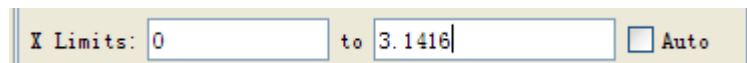


图 5.5-4 横坐标上限设置示例

(4)

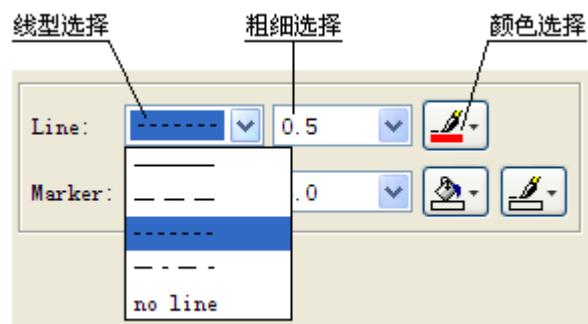


图 5.5-5 包络线的设置示例

(5)

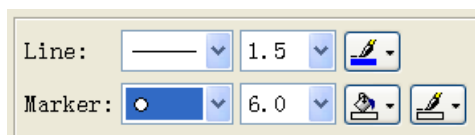


图 5.5-6 调制曲线的设置示例

(6)

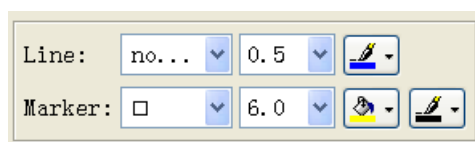


图 5.5-7 调制曲线的设置示例

习题 5

1. 已知椭圆的长、短轴 $a = 4, b = 2$ ，用“小红点线”画如图 p5-1 所示的椭圆 $\begin{cases} x = a \cos t \\ y = b \sin t \end{cases}$ 。
- (提示：参量 t ；点的大小；axis equal)

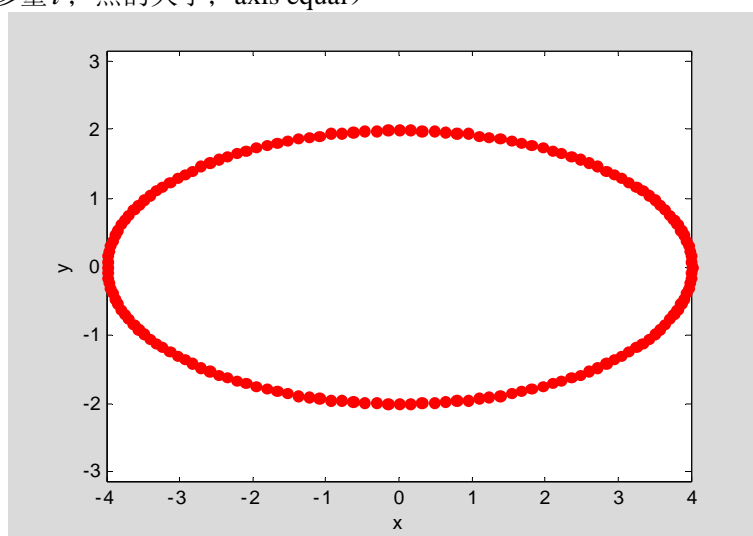


图 p5-1

2. 根据表达式 $\rho = 1 - \cos \theta$ 绘制如图 p5-2 的心脏线。(提示：polar；注意 title 中特殊字符；线宽；axis square。可以用 plot 试试。)

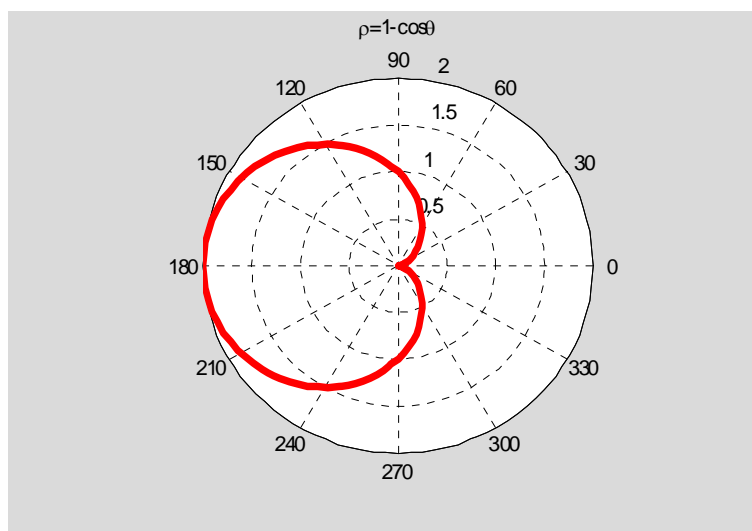


图 p5-2

3. A,B,C 三个城市上半年每个月的国民生产总值表 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：bar(x,Y,'style'); colormap(cool); legend。）

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

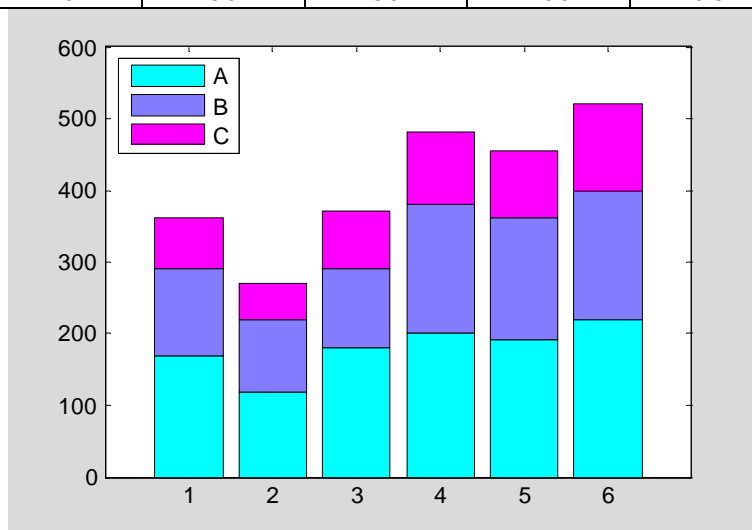


图 p5-3

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

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

（1）希望在同一张图上，绘制 $t \in [0,18]$ 区间内 $\zeta = 0.2:0.2:1.4$ 不同取值时的各条曲线（参见图 p5-4）。在此图上， $\zeta < 1$ 的各条曲线为细蓝线； $\zeta = 1$ 为粗黑线； $\zeta > 1$

为细红线；并且对最上方及最下方的两条曲线给出 $\zeta = 0.2$ 和 $\zeta = 1.4$ 的醒目标志。（2）读者运行题下程序 exmp504.m，可以发现该程序画出的曲线中没有“粗黑线”。你能讲出原因吗？如何对 exmp504.m 作最少的修改（比如只改一条指令），就可画出所需图形。（提示：该题深层次地暴露数值计算可能存在的隐患。）

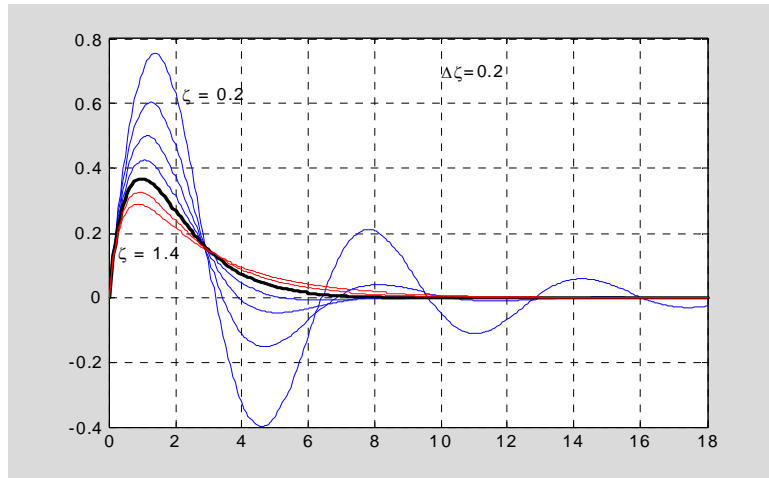


图 p5-4

```
% exmp504.m
clc,clf,clear;
t=(0:0.05:18)';
N=length(t);
zeta=0.2:0.2:1.4;
L=length(zeta);
y=zeros(N,L);
hold on
for k=1:L
    zk=zeta(k);
    beta=sqrt(abs(1-zk^2));
    if zk<1 %满足此条件，绘蓝色线
        y=1/beta*exp(-zk*t).*sin(beta*t);
        plot(t,y,'b')
        if zk<0.4
            text(2.2,0.63,'\zeta = 0.2')
        end
    elseif zk==1 %满足此条件，绘黑色线
        y=t.*exp(-t);
        plot(t,y,'k','LineWidth',2)
    else %其余，绘红色线
        y=(exp(-(zk-beta)*t)-exp(-(zk+beta)*t))/(2*beta);
        plot(t,y,'r')
        if zk>1.2
            text(0.3,0.14,'\zeta = 1.4')
        end
    end
end
end
text(10,0.7,'\Delta\zeta=0.2')
axis([0,18,-0.4,0.8])
hold off
box on
grid on
```

供第 4 道习题使用的程序

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：参变量；plot3；线色线粗。）

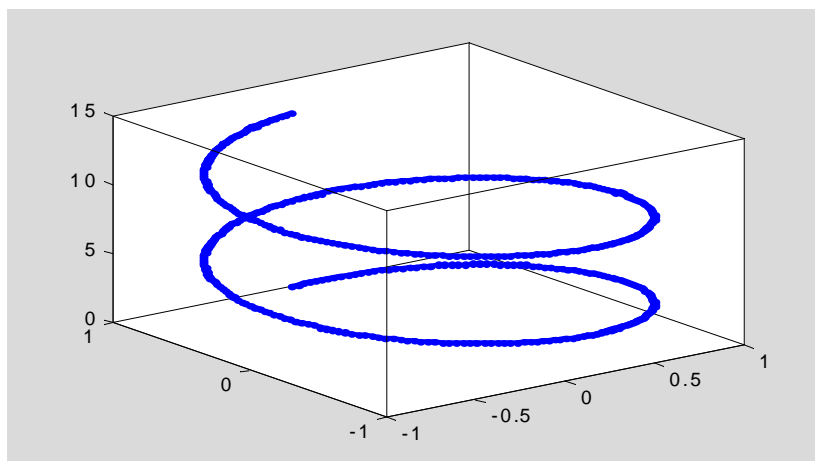


图 p5-5

6. 采用两种不同方法绘制 $z = 4xe^{-x^2-y^2}$ 在 $x, y \in [-3,3]$ 的如图 p5-6 的三维（透视）网格曲面。（提示：ezmesh; mesh; hidden）

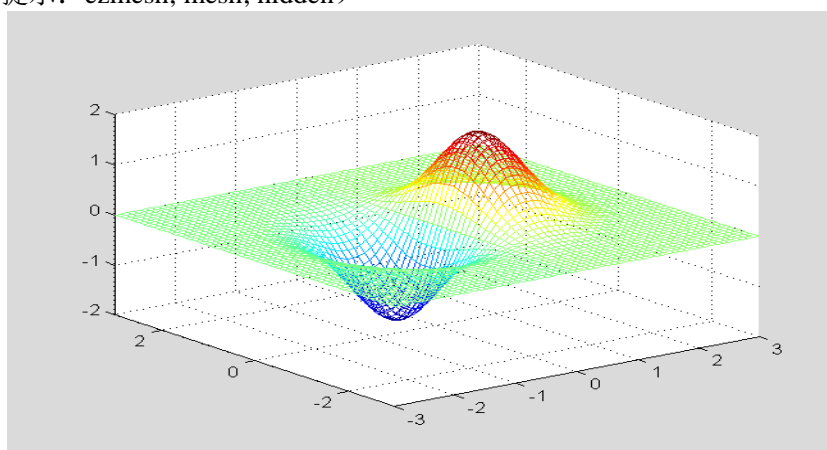


图 p5-6

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。（提示：NaN 的处理）

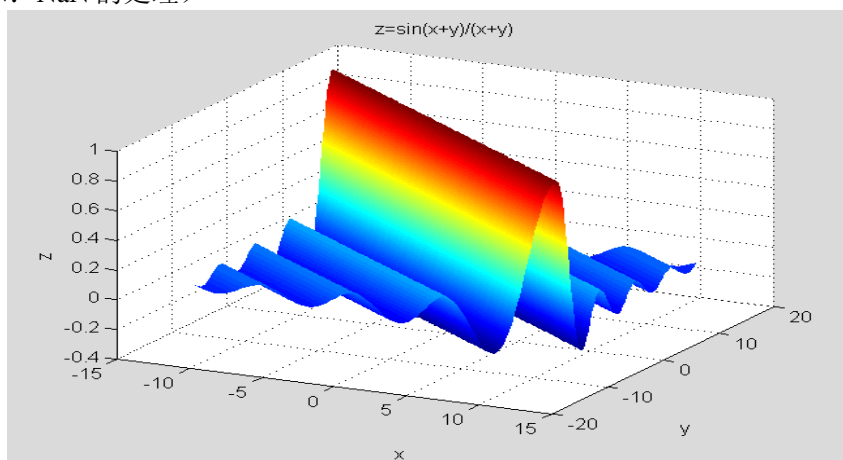


图 p5-7

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? (提示: 图解法; ezplot; ginput)
- (2) 求出离 $x=0, y=0$ 最近、且满足该方程组的一个近似解。
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 p5-8)。
(提示: [jet; flipud(jet)]; colormap; spinmap)

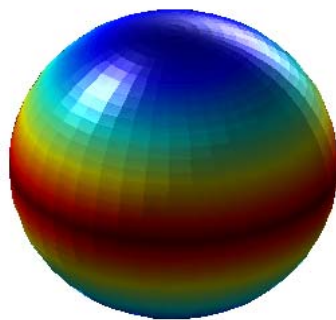


图 p5-8

10. 在 $[0, 4\pi]$ 区间内, 根据 $y(t, x) = e^{-0.2x} \sin(\frac{\pi}{24}t - x)$, 通过如图 p5-9 所示曲线表现“行波”。做题前, 请先运行 prob510.p 文件, 观察演示。(提示: 采用实时动画; 使用两个 line 对象; background 擦除模式; 使用 pause 控制动画速度。)

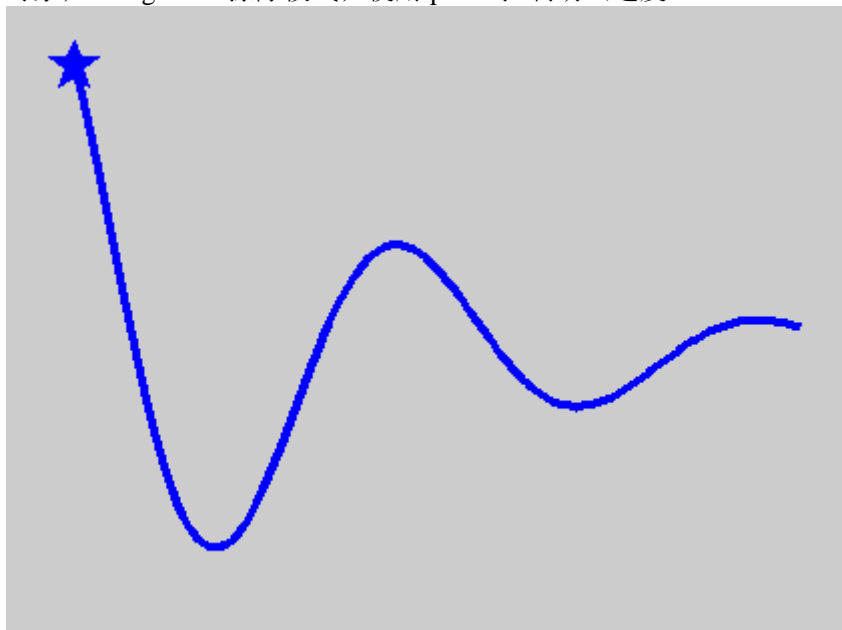


图 p5-9

11. 利用影片动画法, 据函数 $f(x, t) = \sin(x) \sin(t)$ 制作如图 p5-10 所示驻波动画。在做题

前，先运行 prob511.p 产生的演示动画。（提示：用 2 个 line 分别产生带图柄的线和点对象；擦除模式为 background；用 set 通过线图柄操作线位置；getframe；movie）

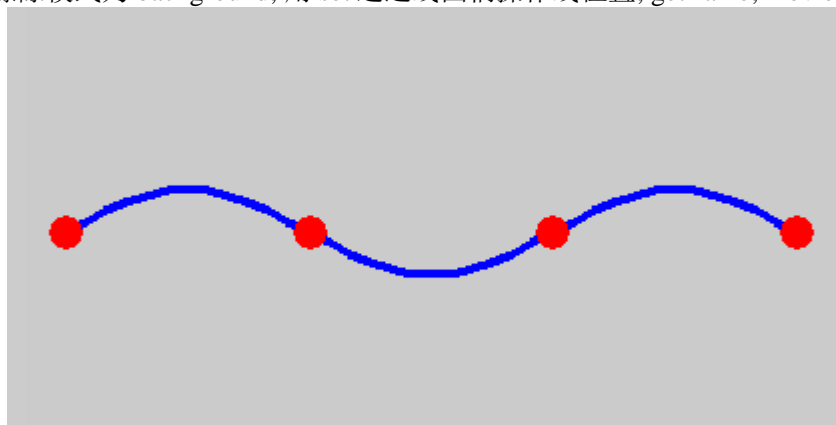


图 p5-10

12. 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 prob512.p（在光盘的 mfiles 文件夹上）的运行实况。图 p5-10 显示的是该动画中的一个静止图形。（提示：用参量方程表达三叶线；用 line 绘制线对象；用 line 创建红点的图柄，擦除模式用 xor；用 set 操作红点坐标，构成动画；drawnow。）

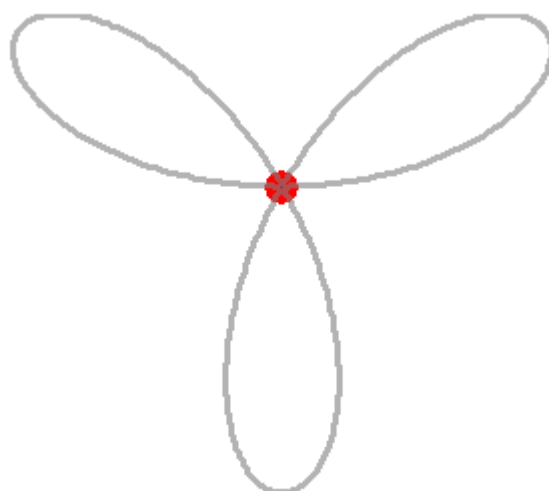


图 p5-11

第 6 章 M 文件和函数句柄

6.1 MATLAB 控制流

6.1.1 if-else-end 条件控制

【例 6.1-1】

(1)

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

```
%
```

```
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)

```
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
```

6.1.2 switch-case 控制结构

【例 6.1-2】

```
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'};% <7>
```

```
A(2,:)={72,83,56,94,100}; % <8>
```

```
%
```

```
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'

```

6.1.3 for 循环和 while 循环

【例 6.1-3】

(1)

```

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

```

(2)

```

%
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

```
(3)
%
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】

(1)

```
function [S,N]=exm060104(epsilon)
```

```
%
```

```
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)

```
[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】

(1)

```
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
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 %
请输入一个能被 4 整除的正整数! n = A =
144    13    25   108    96    61    73    60    48   109   121    12
     2   131   119    38    50    83    71    86    98    35    23   134
     3   130   118    39    51    82    70    87    99    34    22   135
   141    16    28   105    93    64    76    57    45   112   124    9
   140    17    29   104    92    65    77    56    44   113   125    8
     6   127   115    42    54    79    67    90   102    31    19   138
     7   126   114    43    55    78    66    91   103    30    18   139
   137    20    32   101    89    68    80    53    41   116   128    5
   136    21    33   100    88    69    81    52    40   117   129    4
    10   123   111    46    58    75    63    94   106    27    15   142
    11   122   110    47    59    74    62    95   107    26    14   143
   133    24    36    97    85    72    84    49    37   120   132    1

```

(2)

```

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。

```

6.1.4 控制程序流的其它常用指令

6.2 脚本文件和函数文件

6.2.1 M 脚本文件

6.2.2 M 函数文件

6.2.3 局部变量和全局变量

6.2.4 M 函数文件的一般结构

【例 6.2-1】

(1)

```
function [S,L]=exm060201(N,R,str)
```

```
%
```

```
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=R*sin(t);y=R*cos(t);
```

```
if nargin==0
```

```
    plot(x,y,str);
```

```
elseif nargin>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
```

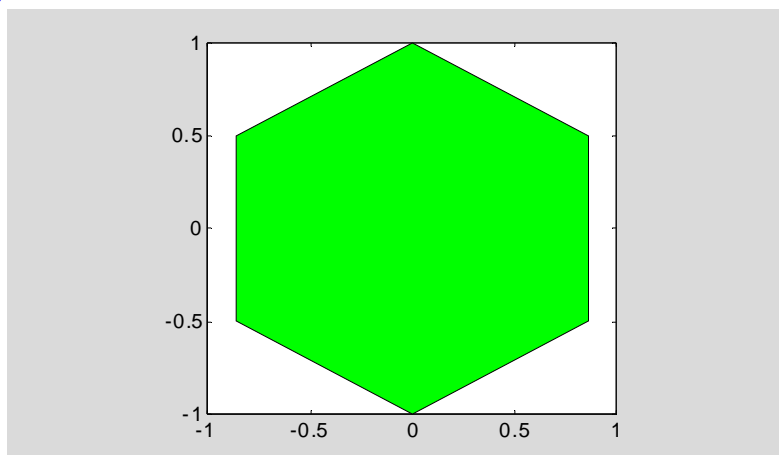


图 6.2-1 绿色正六边形

6.3 MATLAB 的函数类别

6.3.1 主函数 (Primary function)

6.3.2 子函数 (Subfunction)

【例 6.3-1】

(1)

```
function Hr=exm060301(flag)
%
t=(0:50)/50*2*pi;
x=sin(t);
y=cos(t);
Hr=@cirline;           %
feval(Hr,flag,x,y,t)
%
function cirline(wd,x,y,t)
%
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
```

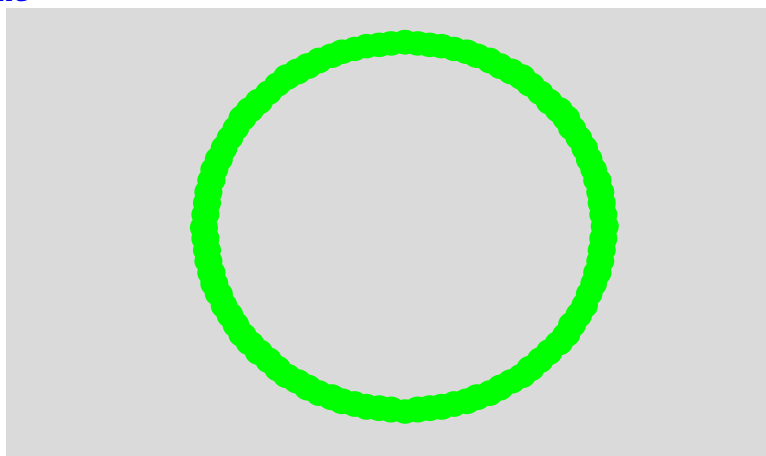


图 6.3-1 绿色圆周线

(3)

```
t=0:2*pi/5:2*pi;x=cos(t);y=sin(t); %
HH('circle',x,y,t)                  %
```

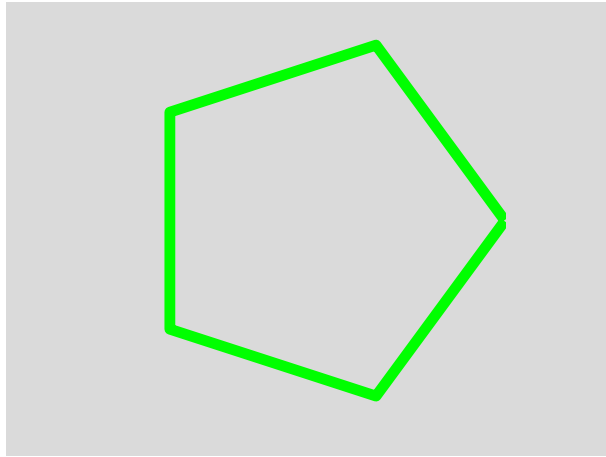



图 6.3-2 由子函数绘制的绿色正五边形

6.3.3 匿名函数 (Anonymous function)

6.4 函数句柄

6.4.1 函数句柄的创建和观察

【例 6.4-1】

(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: 'C:\MATLAB R2010a\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
```

6.4.2 函数句柄的基本用法

【例 6.4-2】

(1)

```
[S,L]=exm060201(3,2,'-r') %
S =
    5.1962
L =
   10.3923
```

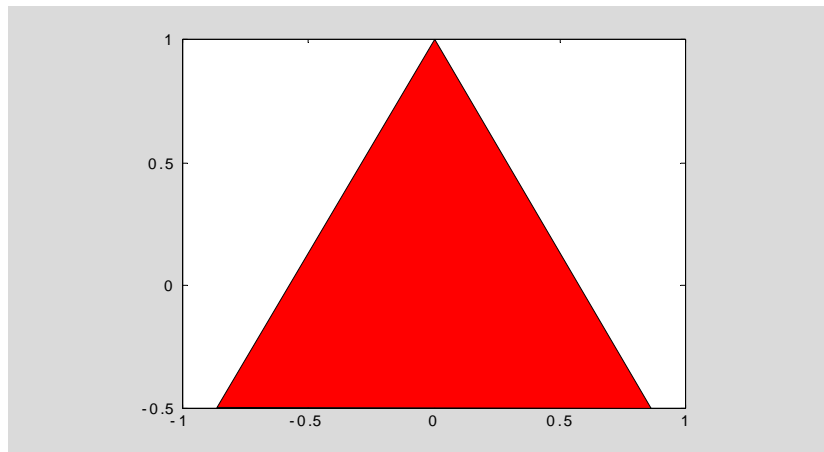


图 6.4-1 直接调用函数 exm060201 所画的红色等边三角形

(2)

```
Hexm=@exm060201, %
Hexm =
    @exm060201
```

(3)

```
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
```

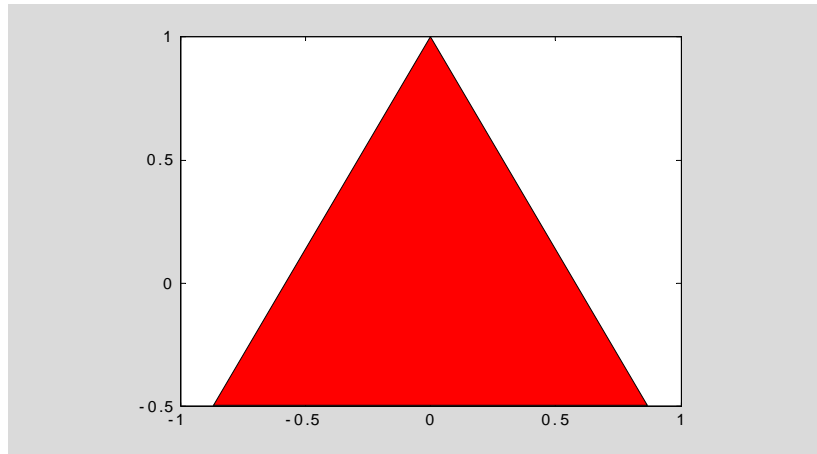


图 6.4-2 在新视野中里用函数句柄 Hexm 所画的红色等边三角形

【例 6.4-3】

(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 $K = \sum_{i=0}^{1000000} 0.2^i = 1 + 0.2 + 0.2^2 + \dots + 0.2^{1000000}$ 的程序。此外，还请写出避免循环的数值、符号计算程序。（提示：sum 和“指数采用数组”配合；tic, toc 可用以记录计算所花的时间。）
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。（提示：nargin, error, int2str）

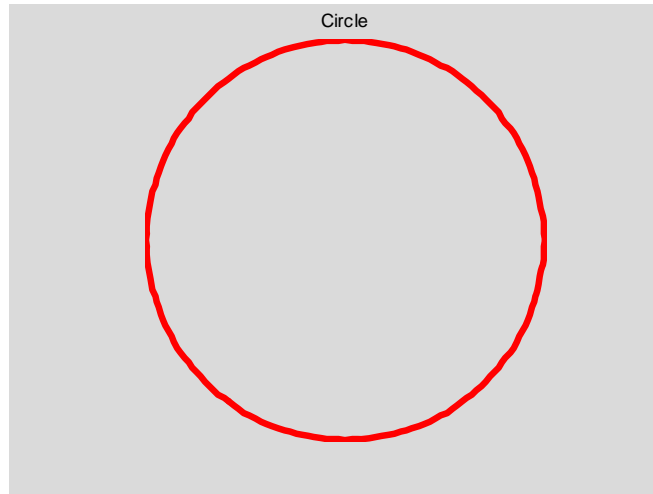


图 p6-1

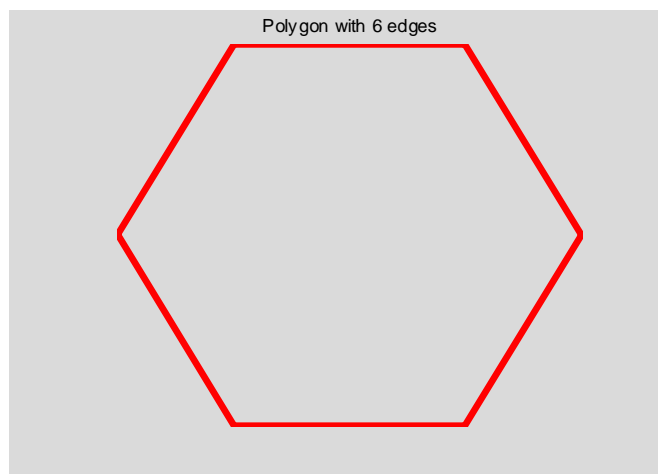


图 p6-2

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。（提示：注意搜索范围的选择；假如极值在边界附近，进一步扩大搜索范围是合理的选择。）
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

$$A = \begin{bmatrix} -0.5000 + 0.8660i & 1.0000 & 0 \\ 0 & -0.5000 - 0.8660i & 1.0000 \\ 1.0000 & 0 & 1.0000 \end{bmatrix}$$

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。（提示：注意函数句柄创建的有效性；若想编写能完全自动执行的解题程序，注意使用 `cd`, `pwd`, `which`, 及字符串的拆分、合并操作技术。）

第 7 章 SIMULINK 交互式 仿真集成环境

7.1 连续时间系统的建模与仿真

7.1.1 基于微分方程的 SIMULINK 建模

【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。本例演示，据物理定理建立微分方程，并以此微分方程创建 SIMULINK 模型的完整步骤：微分方程的整理；模块的复制；信号线的构画；模块参数设置；示波器的调整；仿真参数设置。

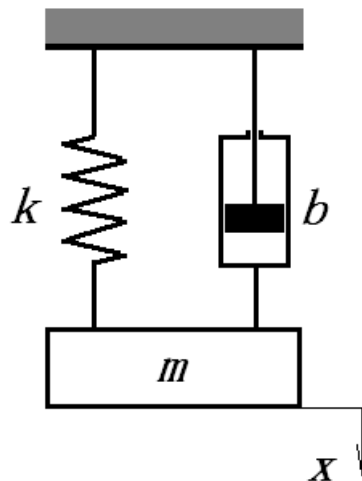


图 7.1-1 弹簧—质量—阻尼系统

- (1) 建立理论数学模型
- (2) 建模的基本思路
- (3) 打开 SIMULINK 模块库

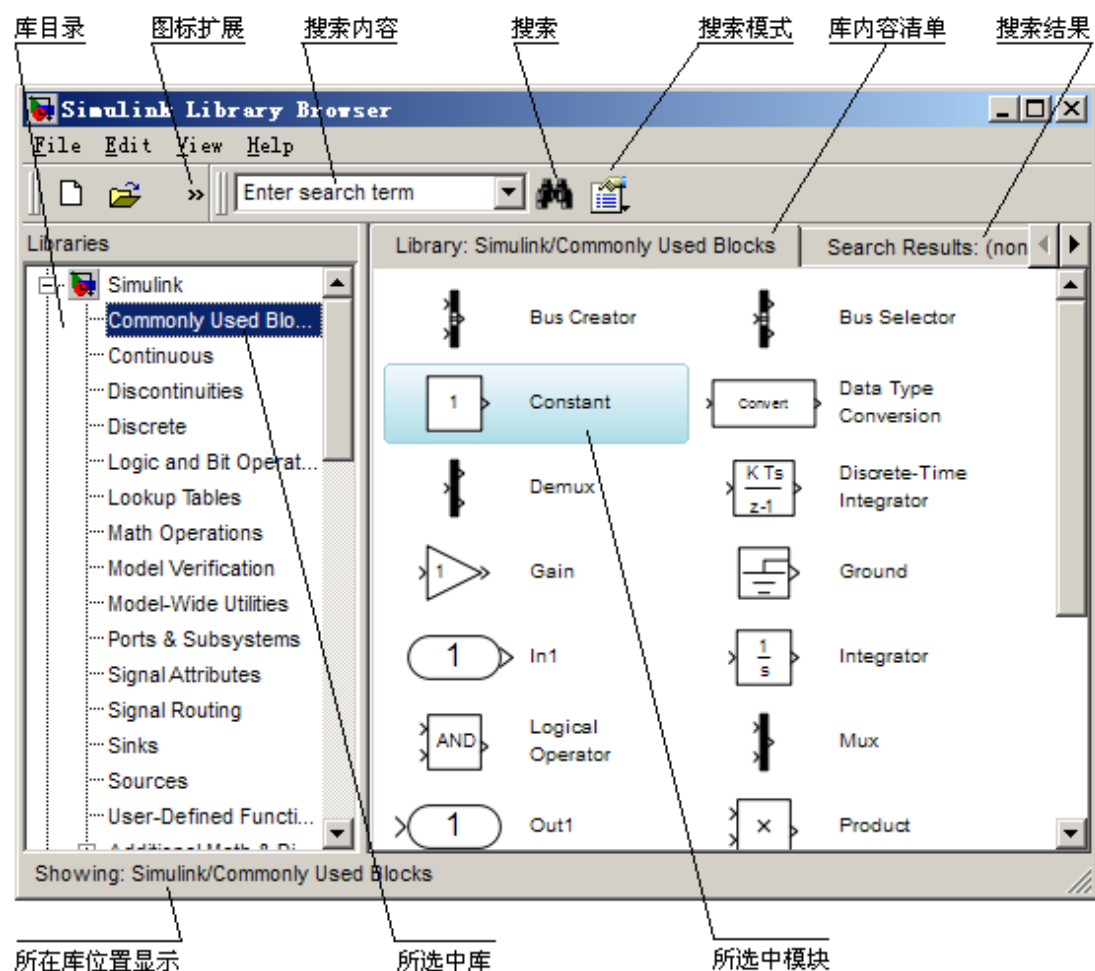


图 7.1-2 SIMULINK 模块库浏览器

- (4) 开启空白（新建）模型窗
- (5) 从模块库复制所需模块到空白（新建）模型窗

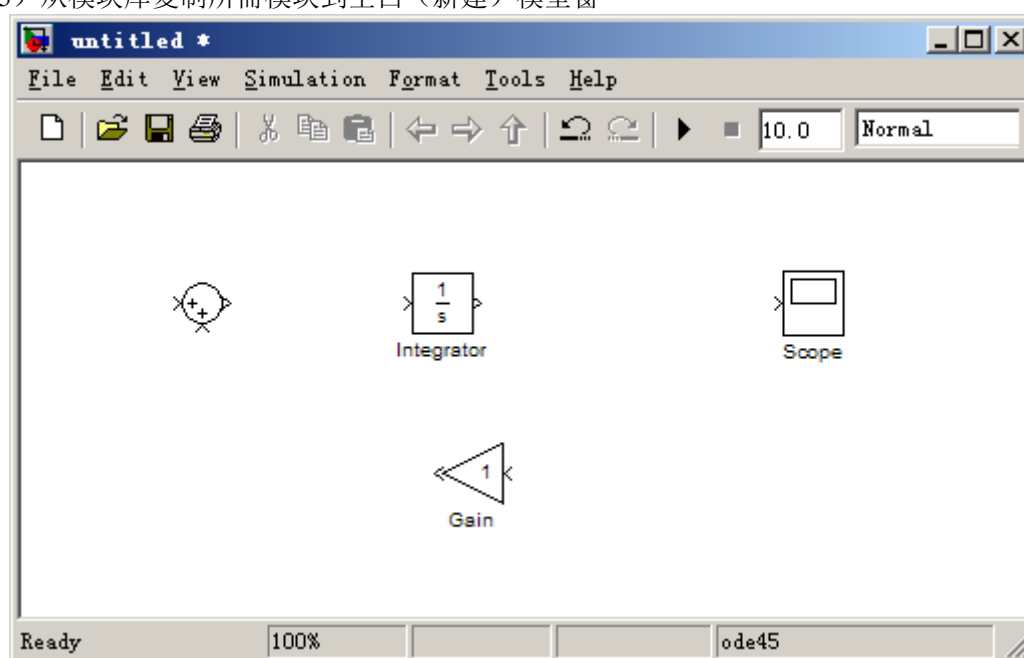


图 7.1-3 已经复制进库模块的新建模型窗

- (6) 新建模型窗中的模型再复制
- (7) 模块间信号线的连接

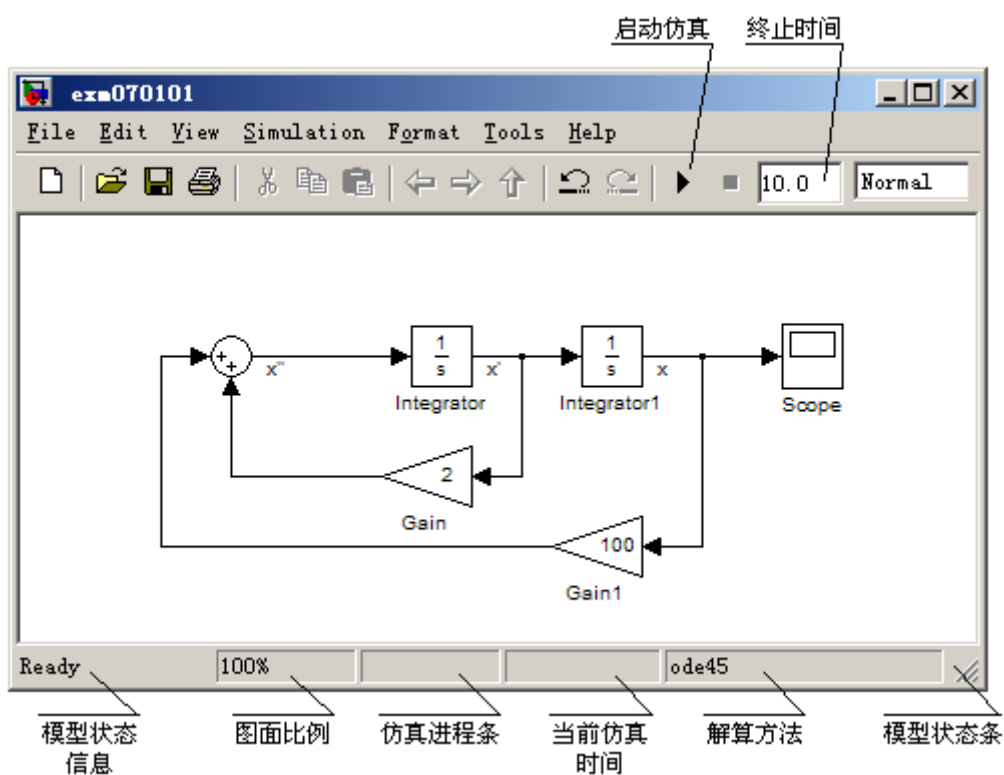


图 7.1-4 已构建完成的新模型窗

- (8) 根据理论数学模型设置模块参数

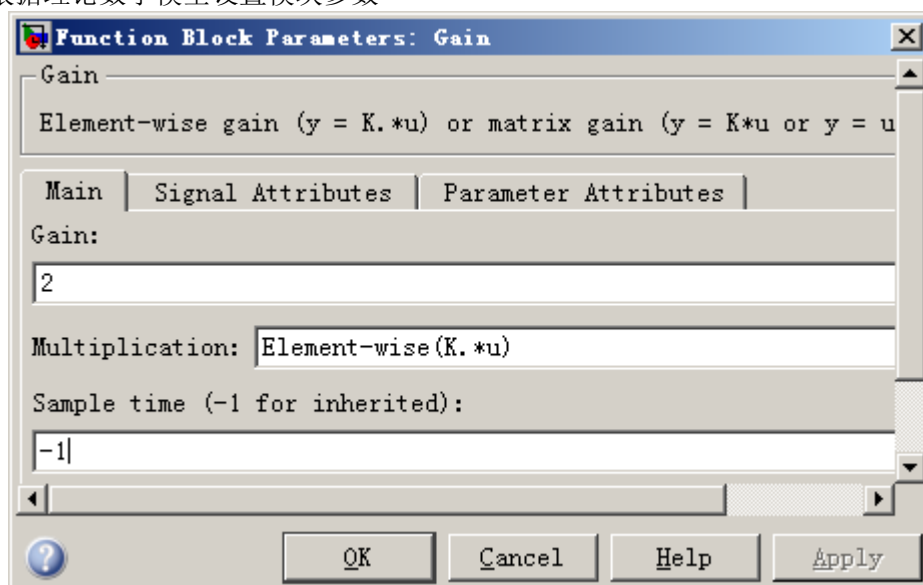


图 7.1-5 参数已经修改为 2 的<Gain>增益模块参数设置窗

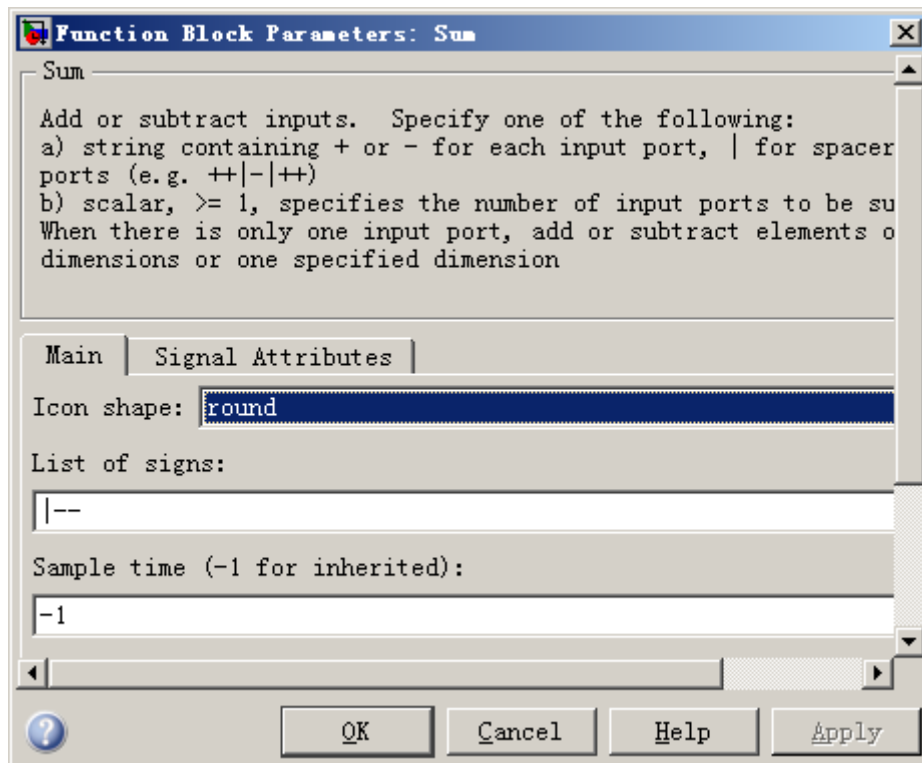


图 7.1-6 改变输入符号的求和模块参数设置窗

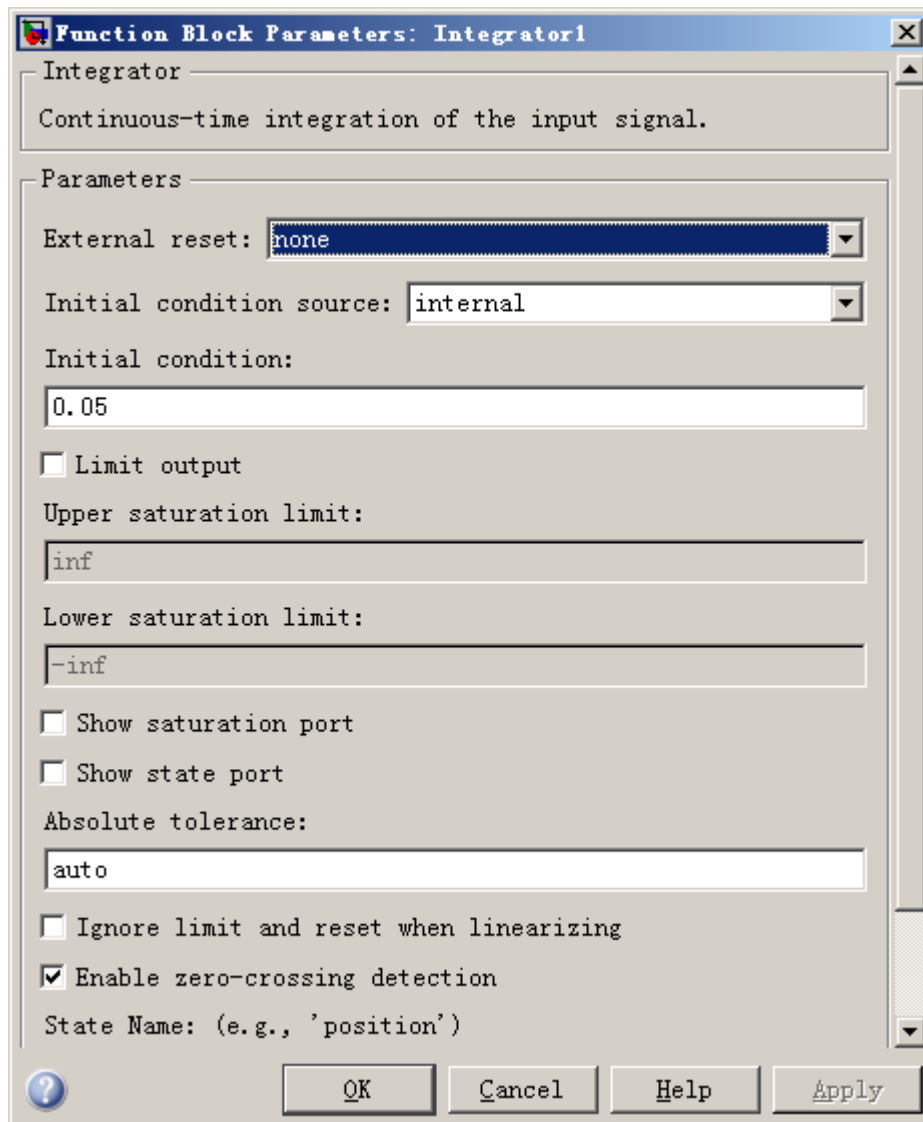


图 7.1-7 实现初始位移 0.05 设置的<Integrator1>设置窗

- (9) 仿真运行参数采用默认解算器“ode45”、默认“变步长”和默认仿真终止时间 10。
- (10) 把新建模型保存为 exm070101.mdl。
- (11) 试运行，以便发现问题加以改善。

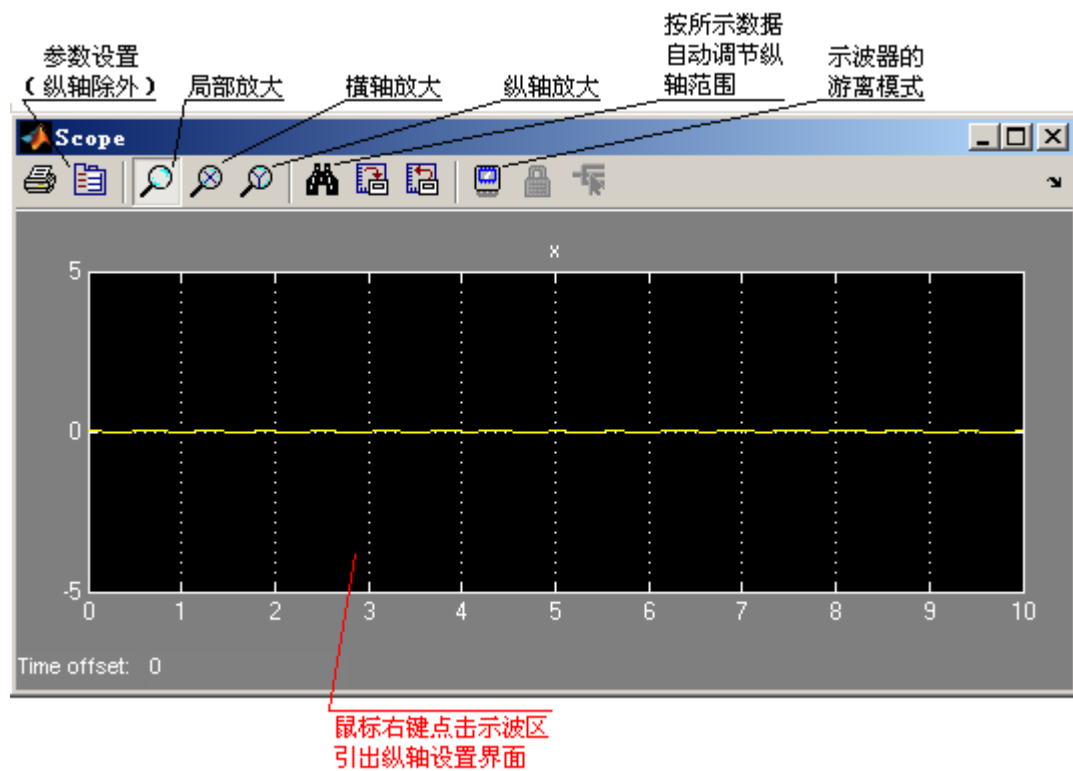


图 7.1-8 坐标范围设置不当时的信号显示

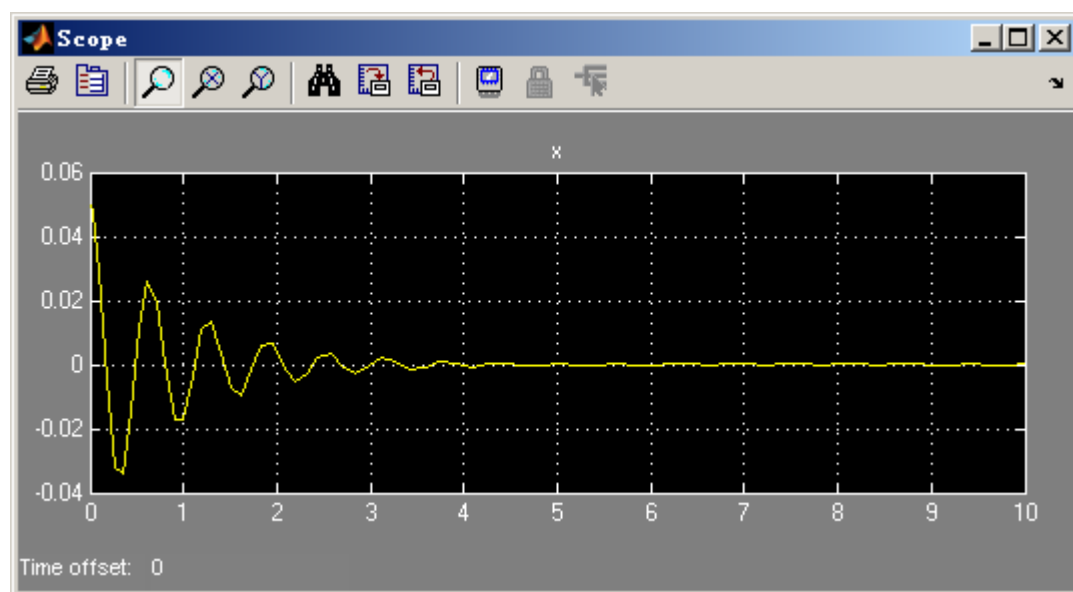


图 7.1-9 采用轴自动设置功能后的信号显示

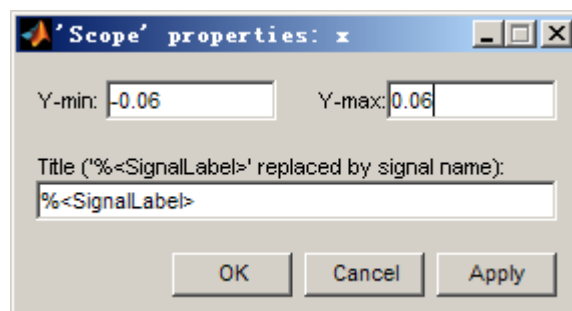


图 7.1-10 对显示屏的纵坐标范围进行设置

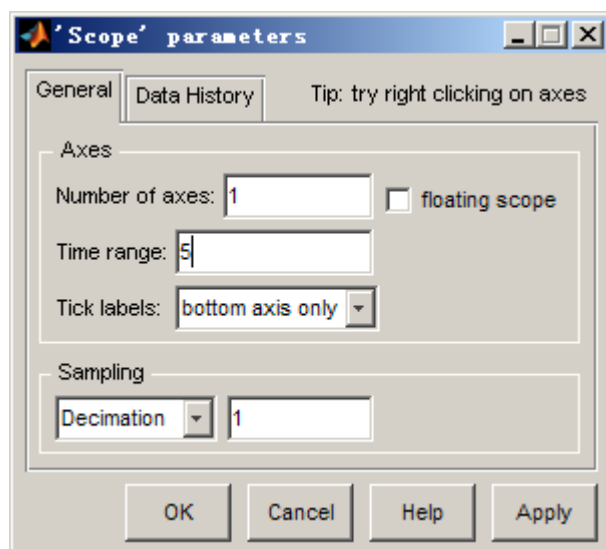


图 7.1-11 对示波器时间显示范围的设置

(12) 据试运行结果，进行仿真参数的再设置。

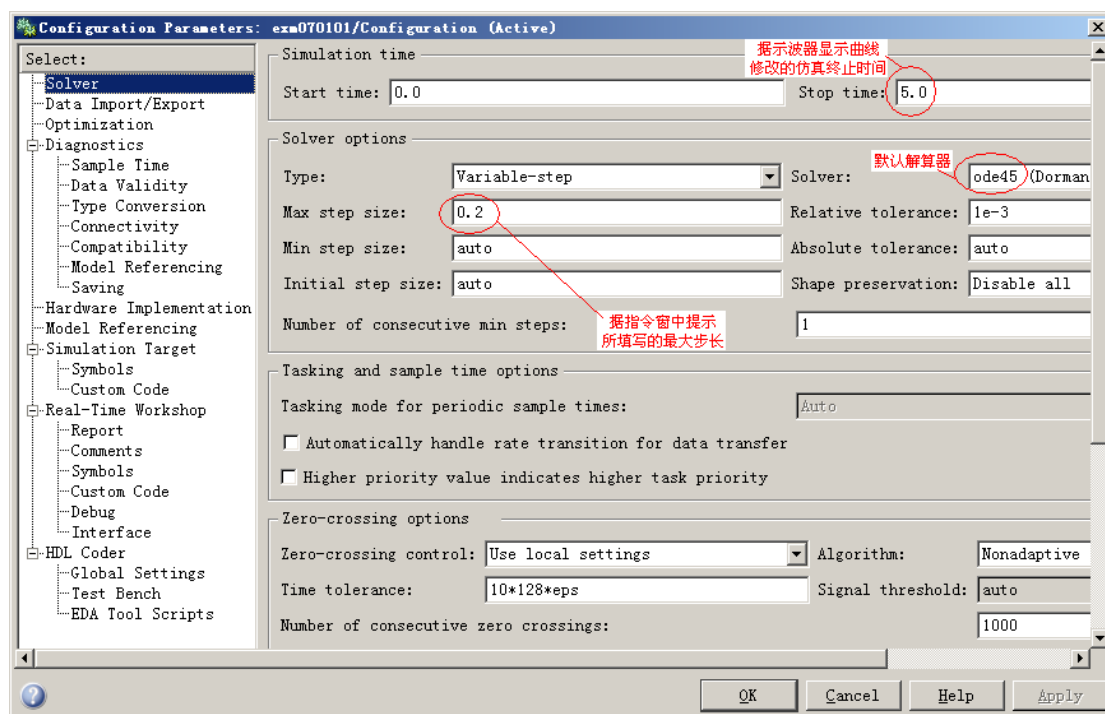


图 7.1-12 通过仿真参数配置窗设置输出曲线光滑因子

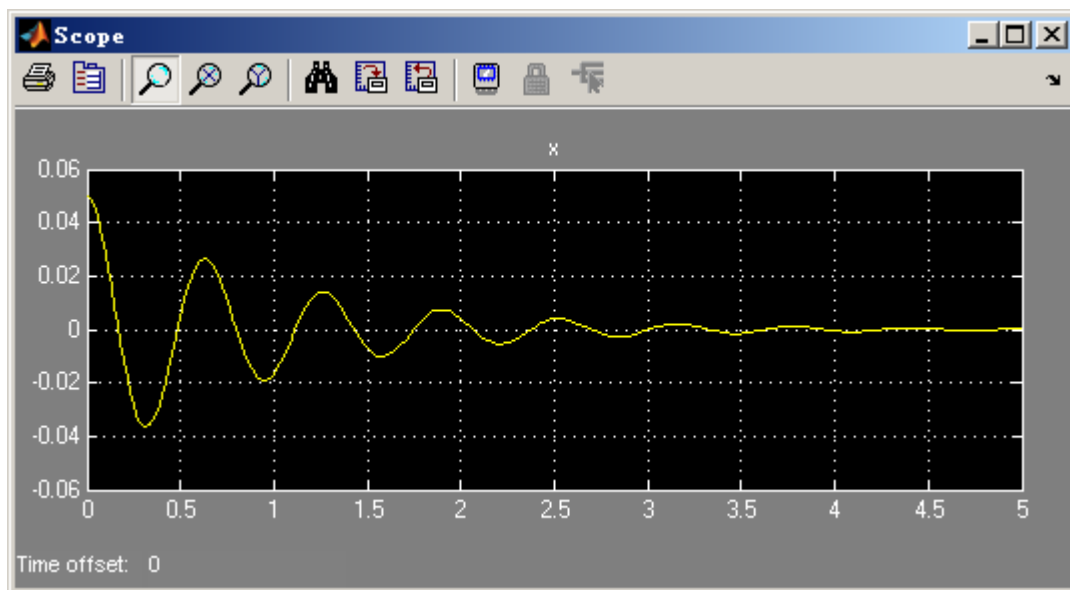


图 7.1-15 适当地显示仿真所得的位移变化曲线

7.1.2 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。

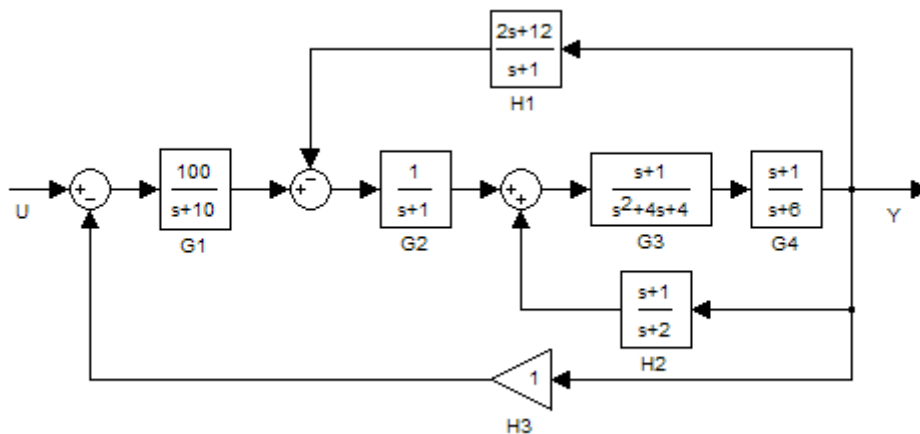


图 7.1-15 多环控制系统

- （1）建模的基本思路
- （2）构造“用于系统传递函数计算”的 SIMULINK 模型

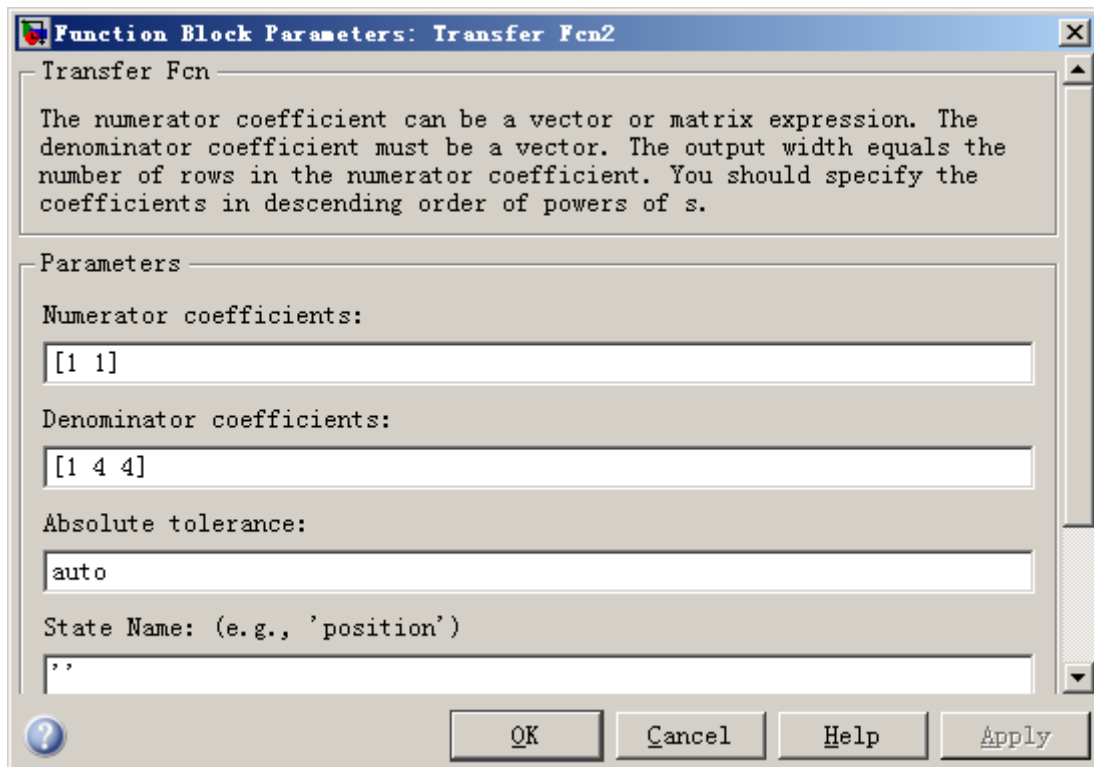


图 7.1-16 传递函数参数设置对话框

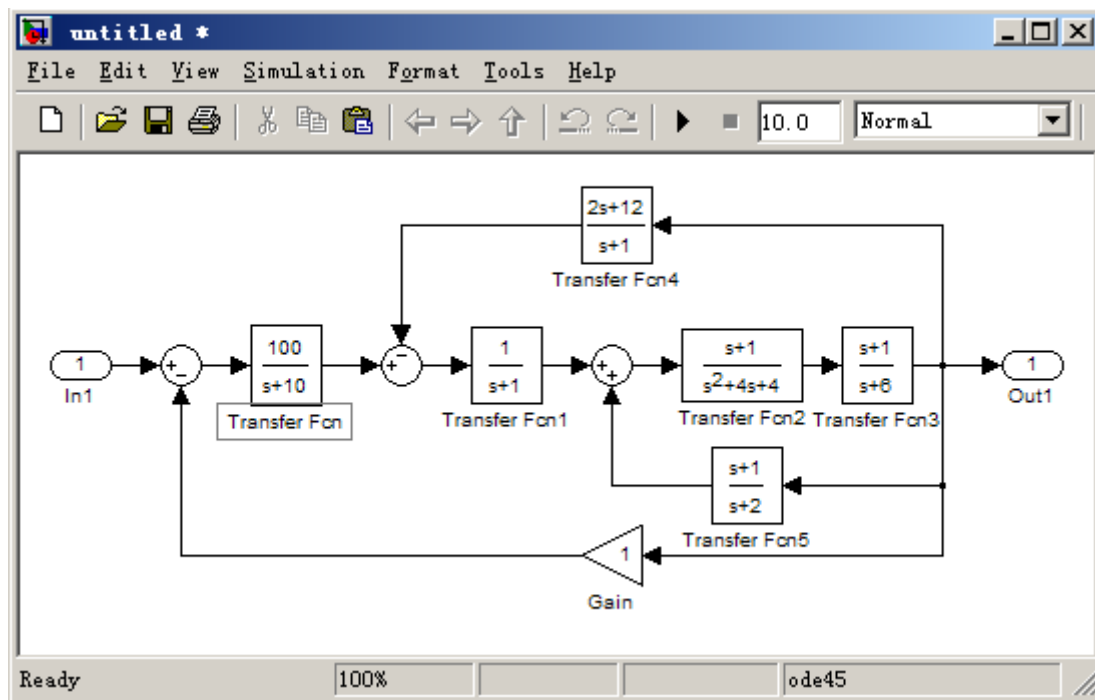


图 7.1-17 完成连接和参数设置的模型

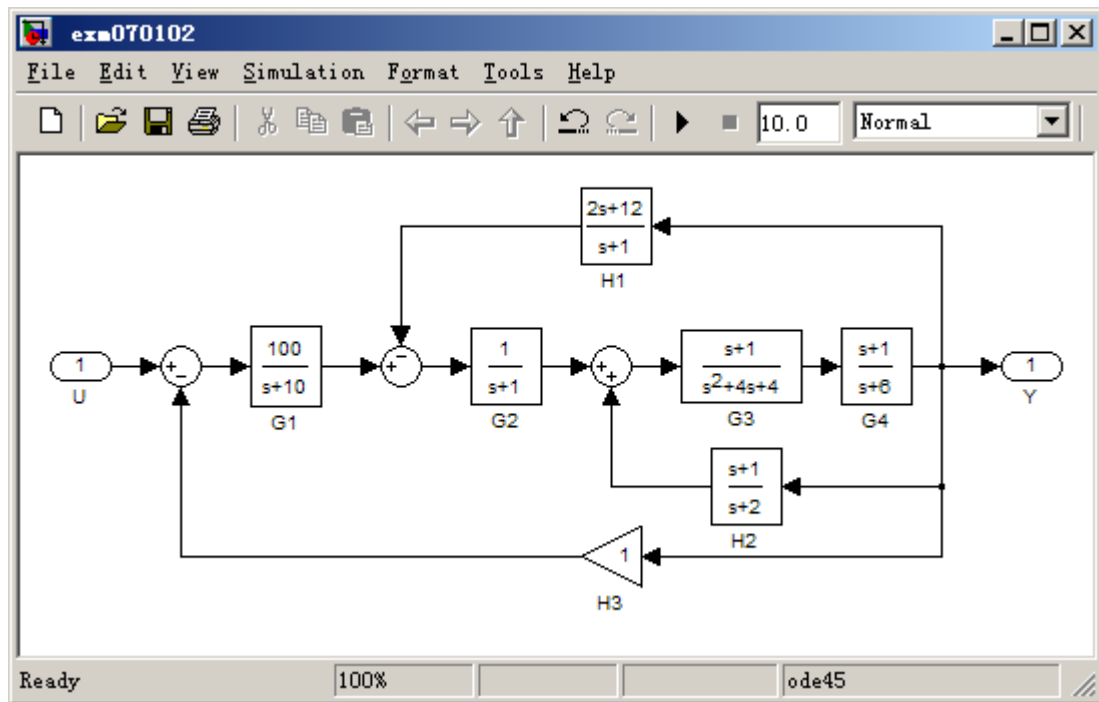


图 7.1-18 用于系统传递函数计算的模型

```
(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:
          100 s^2 + 300 s + 200
-----
s^5 + 21 s^4 + 157 s^3 + 663 s^2 + 1301 s + 910

ans =
          0          0          0 100.0000 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')
```

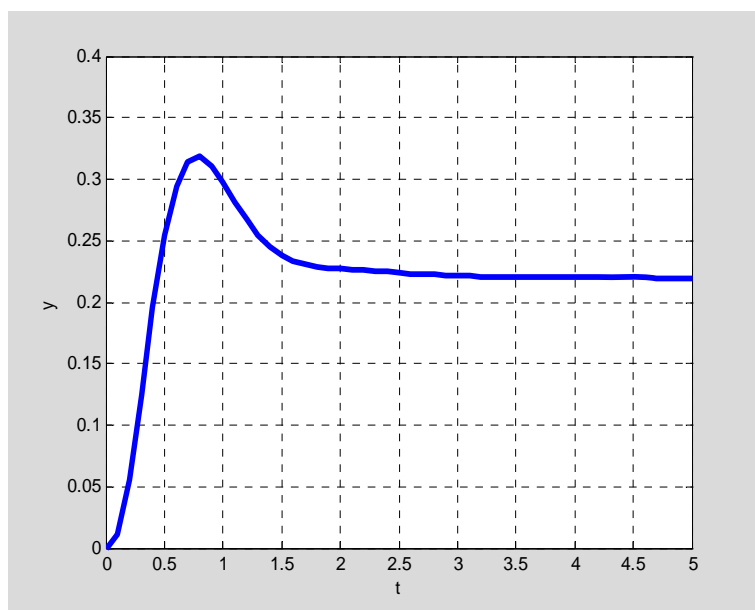


图 7.1-19 系统的单位阶跃响应

7.2 离散时间系统的建模与仿真

【例 7.2-1】设计一个数字低通滤波器 $F(z)$ ，从受噪声干扰的多频率混合信号 $x(t)$ 中获取 10Hz 的信号。

$$x(t) = \sin(2\pi \cdot 10 \cdot t) + 1.5 \cos(2\pi \cdot 100 \cdot t) + n(t) \quad (7.3)$$

在此， $n(t) \sim N(0, 0.2^2)$ ， $t = k \cdot \frac{1}{f_s} = k \cdot T_s$ 。采样频率取 $f_s = 1000$ (Hz)，即采样

周期 $T_s = 0.001$ (秒)。本例演示：纯离散时间系统建模的完整过程；离散时间仿真模型中采样周期的设定；SIMULINK 模块参数与 MATLAB 内存变量之间的数据传递；影响模块几何结构的参数。

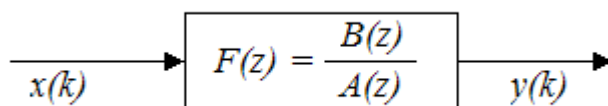


图 7.2-1 数字滤波示意图

- (1) 分析
- (2) 模型的构建

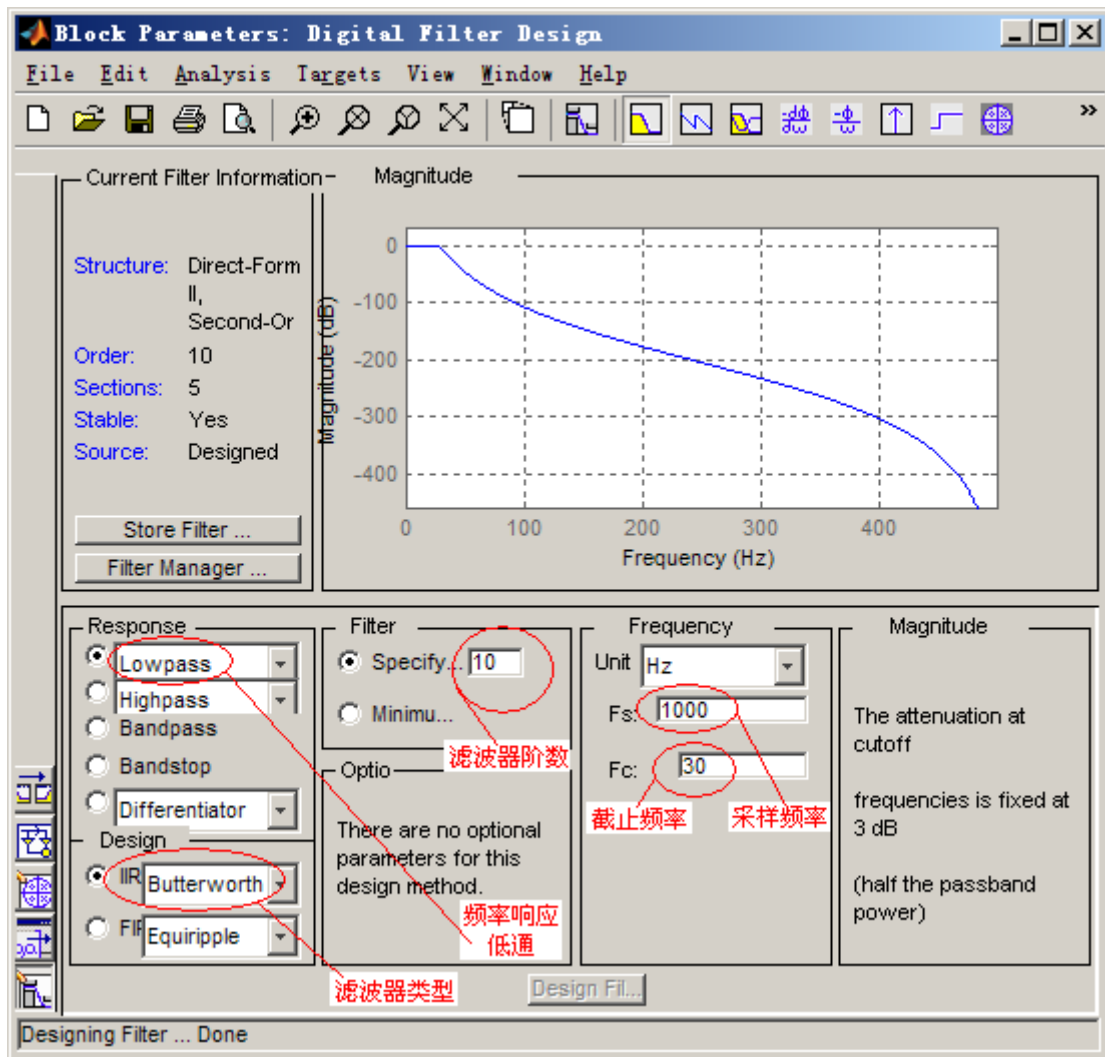


图 7.2-2 数字滤波设计模块参数设置

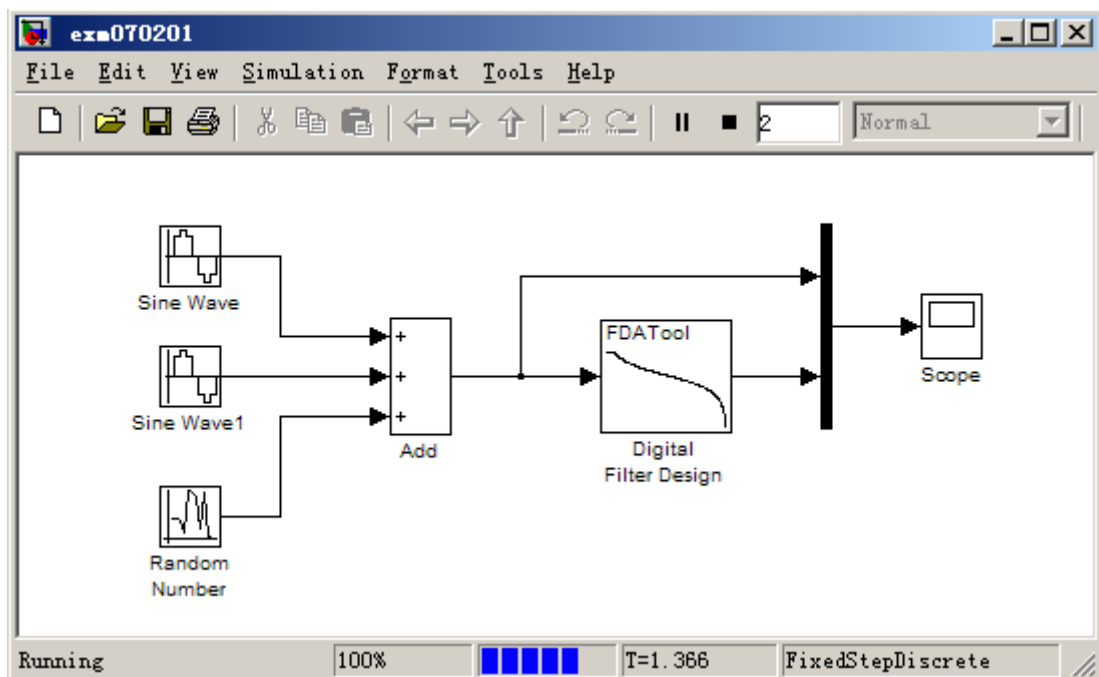


图 7.2-3 仿真进程中的 SIMULINK 滤波模型

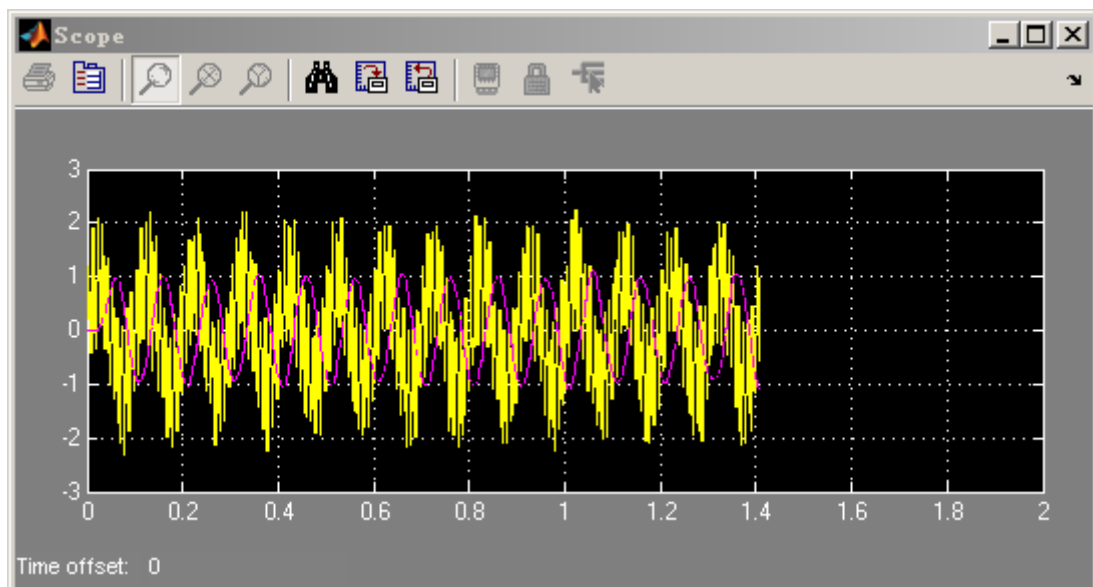


图 7.2-4 仿真进程中的示波图形

7.3 SIMULINK 实现的元件级电路仿真

【例 7.3-1】在图 7.3-1 所示的电路中，已知 $L = 0.3(H)$ ， $C = 0.3(F)$ ， $R_1 = 2(\Omega)$ ， $R_2 = 0.01(\Omega)$ ， $R_3 = 5(\Omega)$ ， $V_C(0^-) = -1(V)$ ， $i_L(0^-) = 1(A)$ ， $V_s = 10(V)$ ，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，求 i_L 和 V_C 。

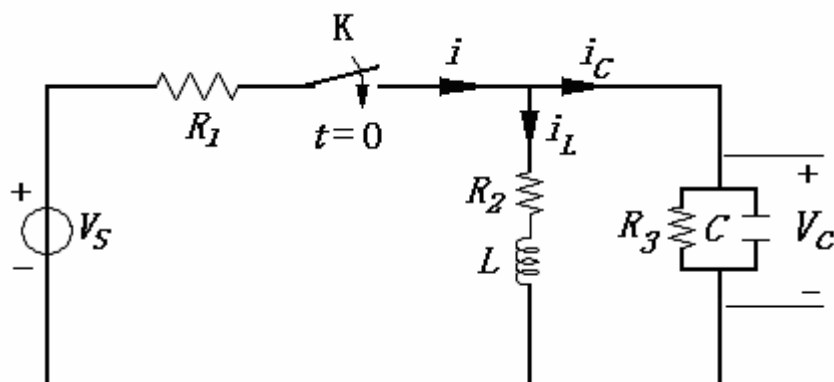


图 7.3-1 二阶 RLC 电路

- (1) 建模的基本思路
- (2) 仿真模型所用器件的来源及参数设置
- (3) 关于仿真模型元器件间连线的说明

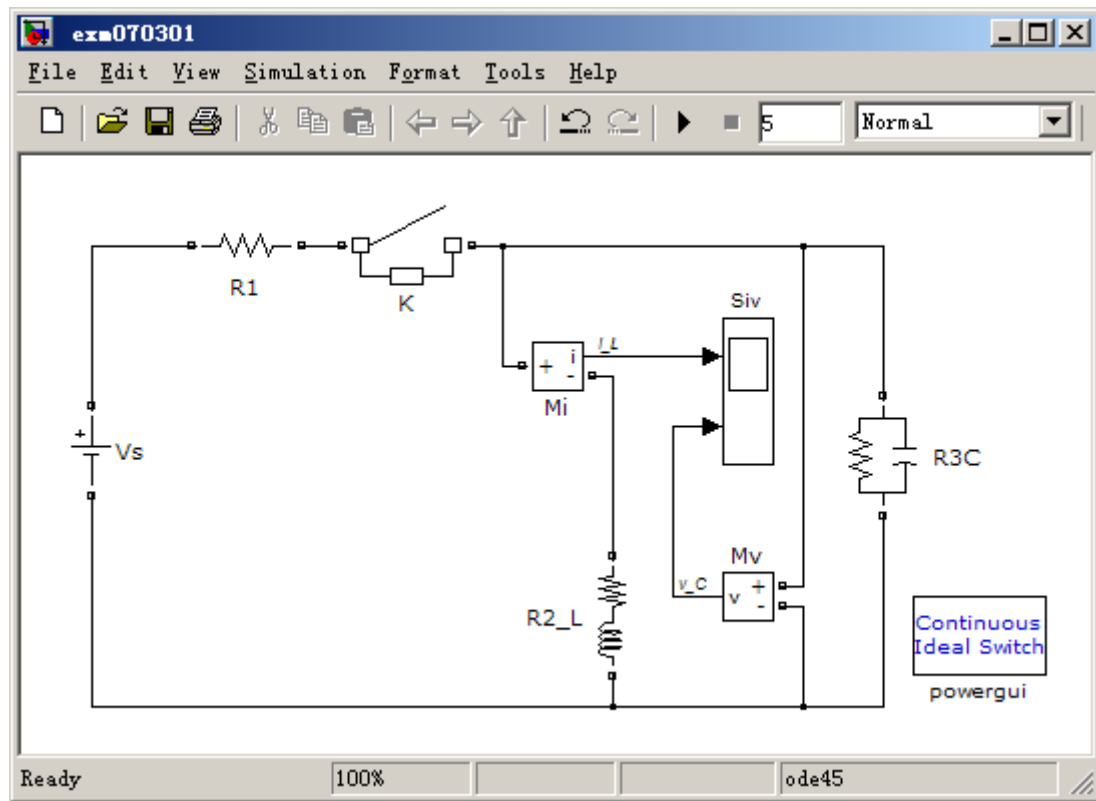


图 7.3-2 元件级仿真模型 exm070301

(4) 营造 SimPowerSystems 仿真环境的<powergui>模块



图 7.3-3 exm070301 模型的 powergui 对话框

(5) 理想开关的设置

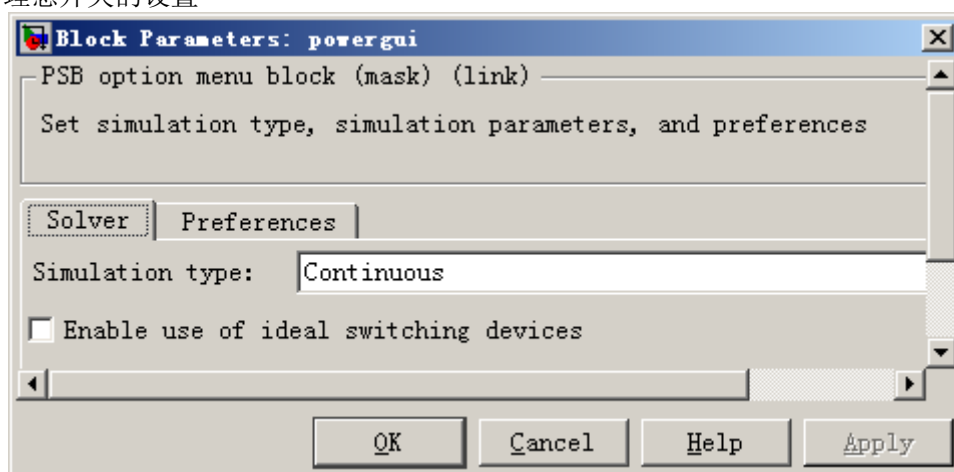


图 7.3-4 仿真类型和理想开关设置窗的默认界面

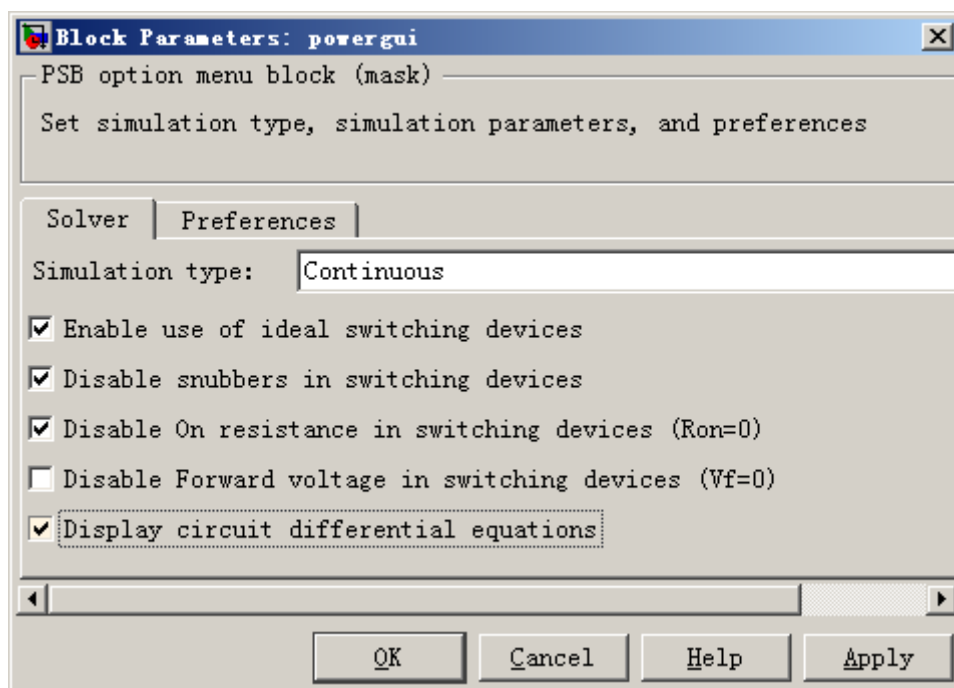


图 7.3-5 仿真模式及理想开关设置对话框

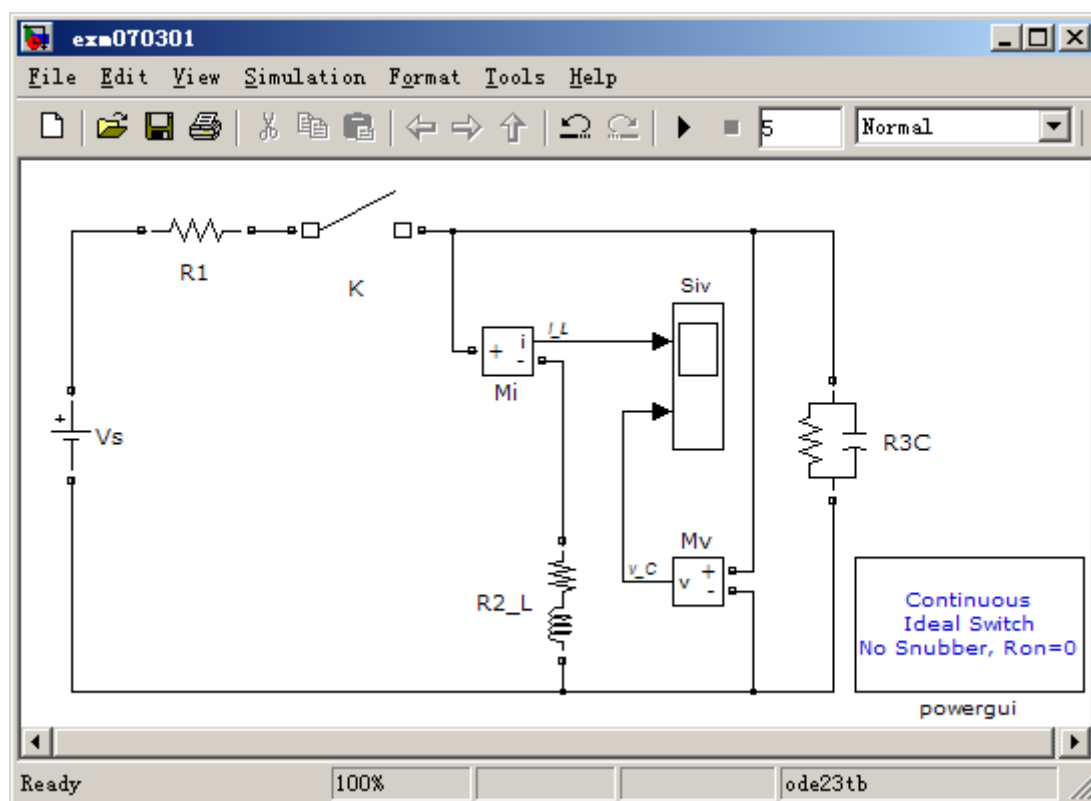


图 7.3-6 理想开关设置后的模型窗

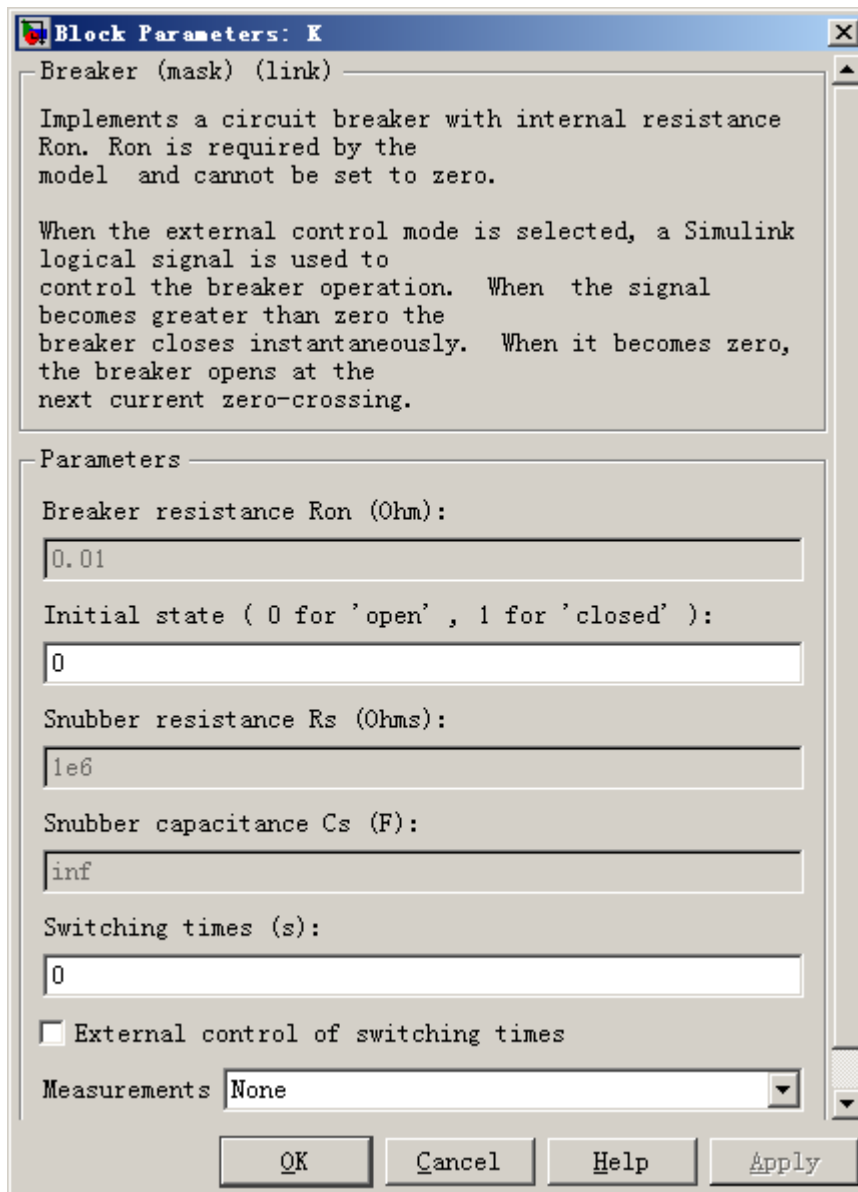


图 7.3-7 开关经失能设置后的对话框

- (6) 仿真类型的选定
- (7) 模型初始状态的设定

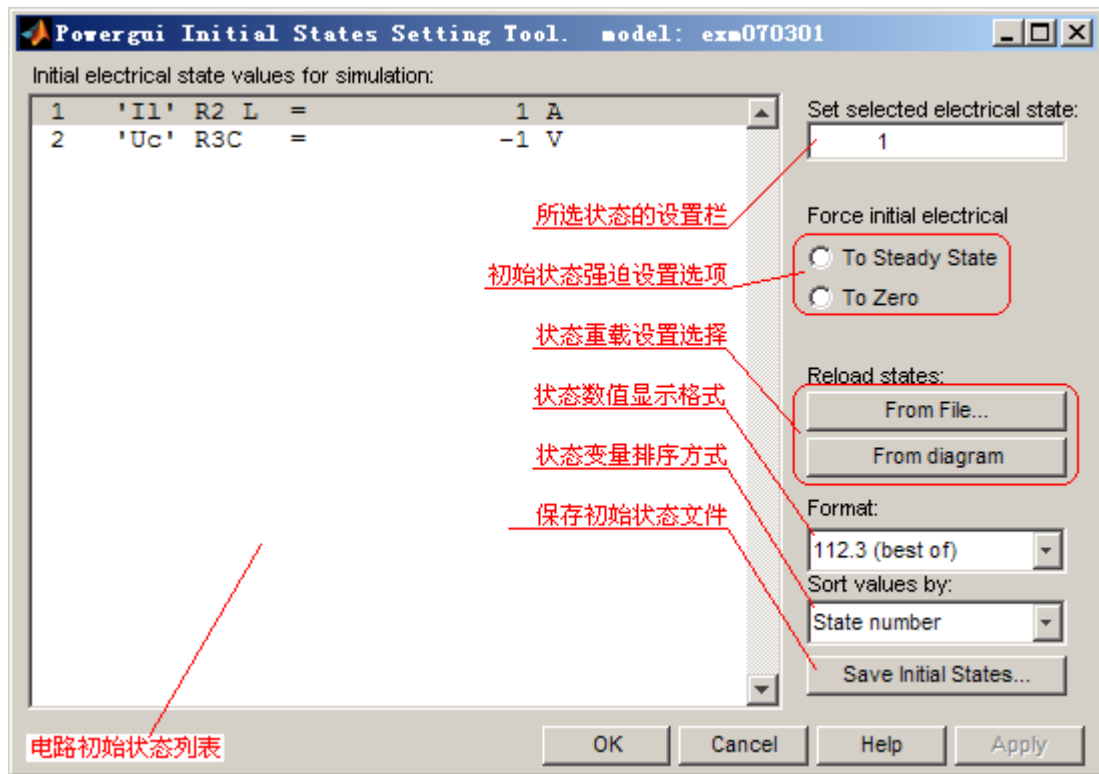


图 7.3-8 powergui 的初始值设置窗

(8) 仿真运行

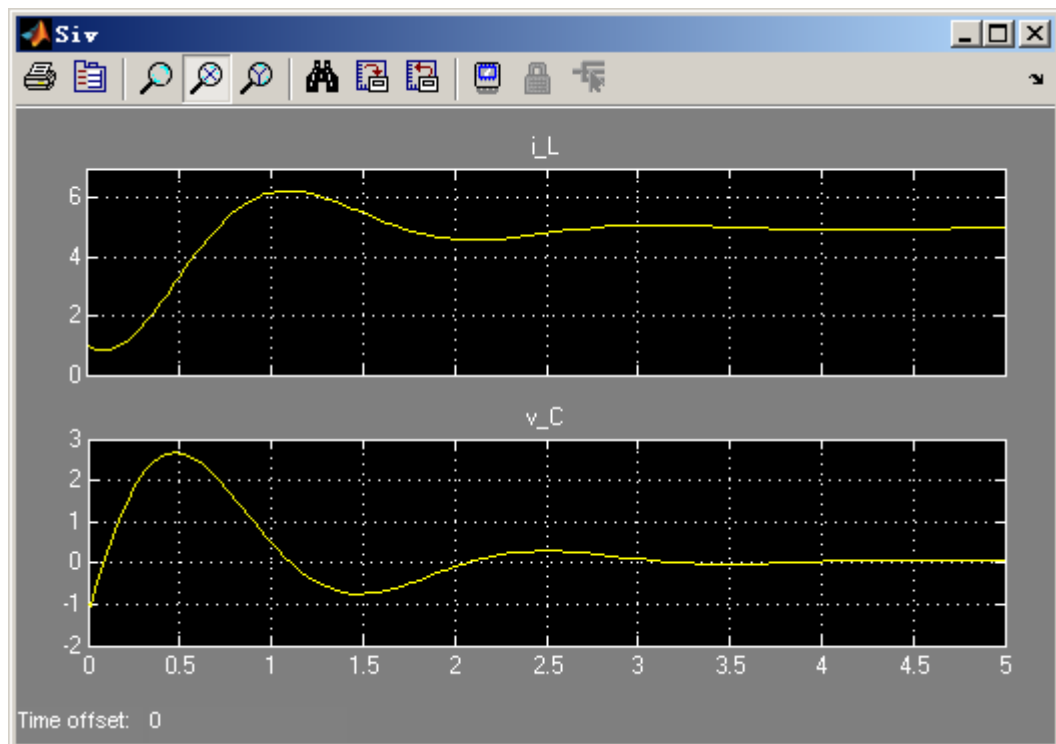


图 7.3-9 仿真所得的电感电流和电容电压变化曲线

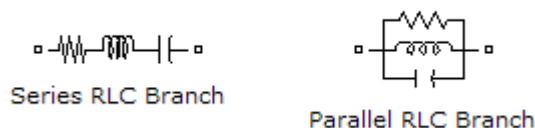


图 7.3-10 RLC 库器件原型

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 $I(1)$ 。（提示：时间变量由 Clock 产生；注意使用 Product, Math function, Integrator, Display, Scope 等库模块；计算结果可与例 4.1-5 对照。）
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。（提示：注意使用 Constant, Product, Add, Gain, Integrator, Scope 等库模块；注意初始状态设置；针对不同 μ ，采用不同解算器，并设置不同仿真终止时间；运算结果可与例 4.1-9 对照。）
3. 已知某系统的框图如图 p7-1 所示，求该系统的传递函数。（提示：参照例 7.1-2）

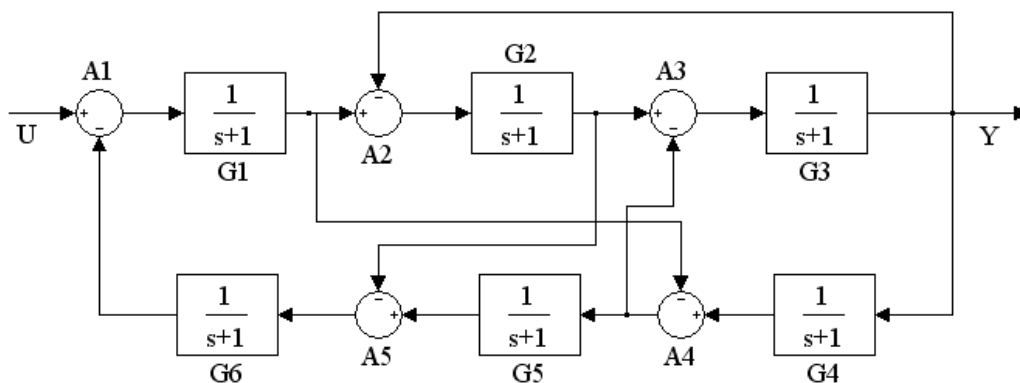


图 p7-1

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。（提示：可使用 Sine Wave, Analog Filter Design 等库模块；注意采样时间设置）
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。（提示：采用相量分析法；要注意电压源、电流源库模块的频率、相角设置；特别要注意模块所采用的幅值是“峰幅值”，它应是“有效值”的 $\sqrt{2}$ 倍；电压、电流测量模块的输出信号，可选择 Magnitude-Angle 幅相模式；powergui 模块的仿真类型 Simulation type 应选择“相量 Phasor”；仿真终止时间非 0 即可；仿真可采用纯离散解算器 discrete。）

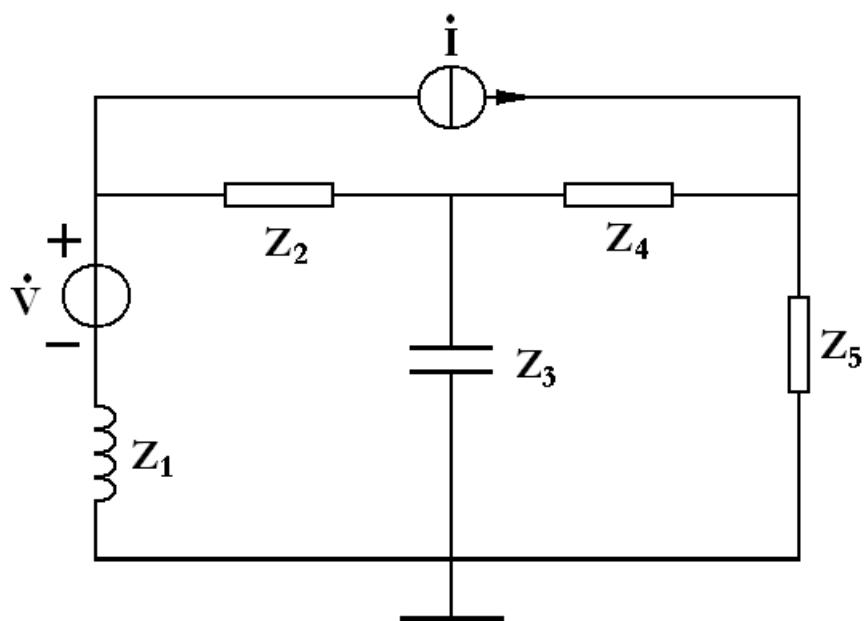


图 p7 -2

第 8 章 图形用户界面 (GUI)

假如读者所从事的数据分析、解方程、计算结果可视工作比较单一，那么一般不会考虑图形用户界面（Graphical User Interfaces，GUI）的制作。但是如果读者想向别人提供某种新的设计分析工具，想体现某种新的设计分析理念，想进行某种技术、方法的演示，那么图形用户界面也许是最好的选择之一。

MATLAB 本身提供了很多的图形用户界面，如 sisotool “单输入单输出控制系统设计工具”、fdatool “滤波器设计和分析工具”等。这些工具的出现不仅提高了设计和分析效率，而且改变原先的设计模式，引出了新的设计思想，改变了和正在改变着人们的设计、分析理念。正是出于这种观察，作者将图形用户界面内容列入本教程。

本章由四个精心设计的算例组成。这四个算例，一方面尽可能多地向读者展现构成 GUI 的各种控件或组件，另方面借助算例中回调函数的编写较快地将读者引向 GUI 制作的纵深。

本书作者相信：读者只要耐心地按照示例进行操作，便能在愉快感受 GUI 成功制作的同时，事半功倍地掌握 GUI 的制作技巧。

8.1 图形用户界面入门示例

【例 8.1-1】为演示归一化二阶系统 $G(s) = \frac{1}{s^2 + 2\zeta s + 1}$ 中阻尼比 ζ 对单位阶跃响应的影响，需要制作如图 8.1-1 所示的用户界面。要求：在界面右侧的编辑框中输入阻尼比 ζ 的具体“大于 0 的数值标量”，就能在坐标轴上画出相应曲线。

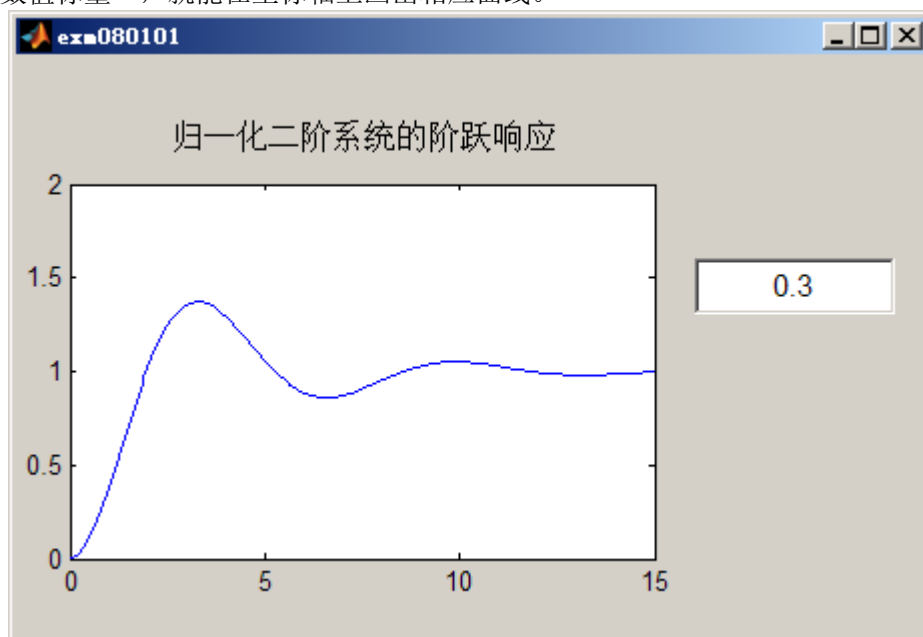


图 8.1-1 待制作的二阶系统单位阶跃响应演示界面

(1) GUIDE 开发环境的引入

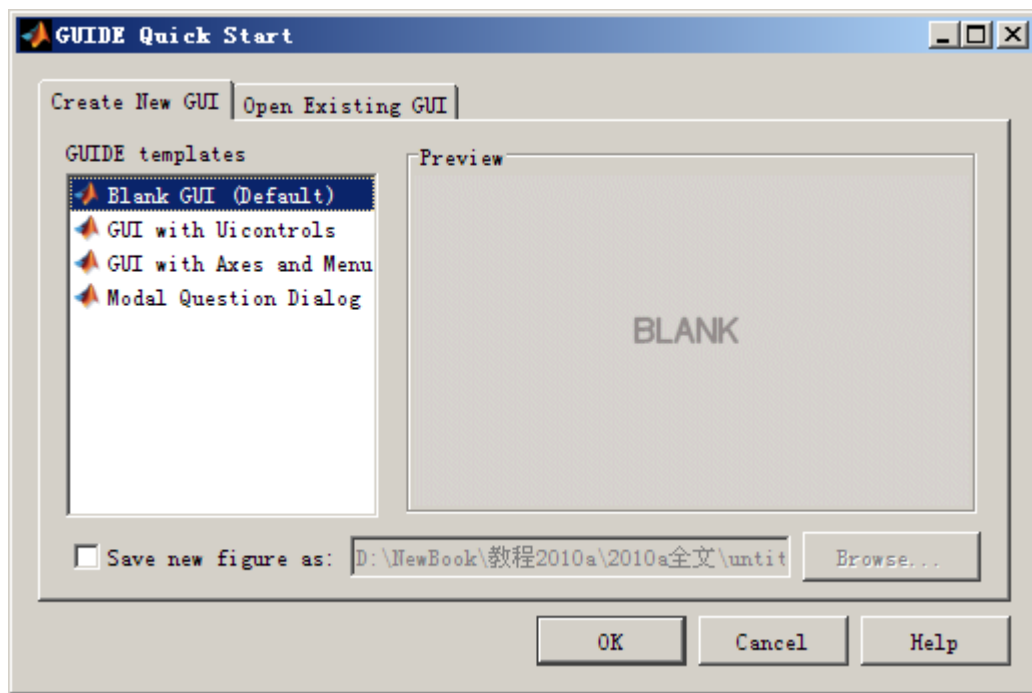


图 8.1-2 进入 GUIDE 开发环境的默认引导对话框

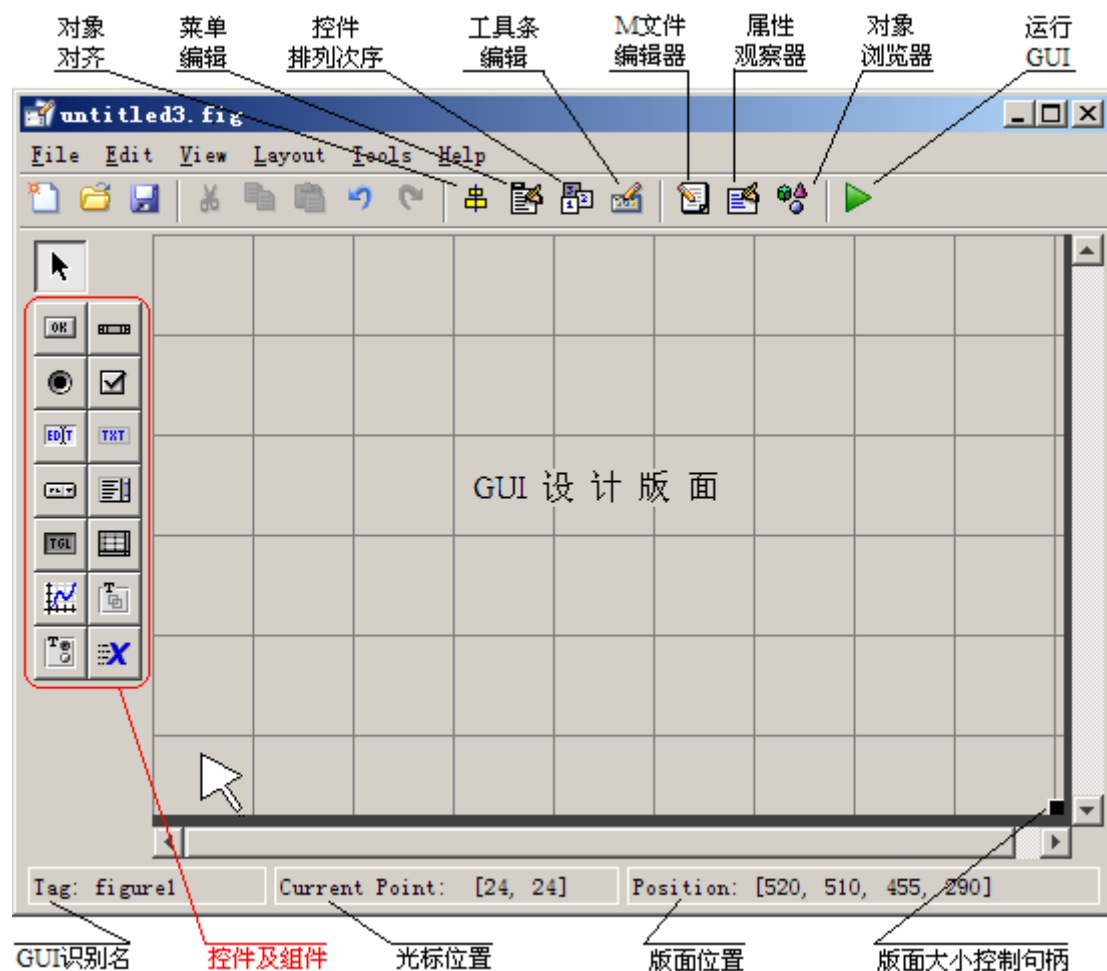


图 8.1-3 默认的空白用户界面开发环境 GUIDE

(2) 根据要求选配界面组件

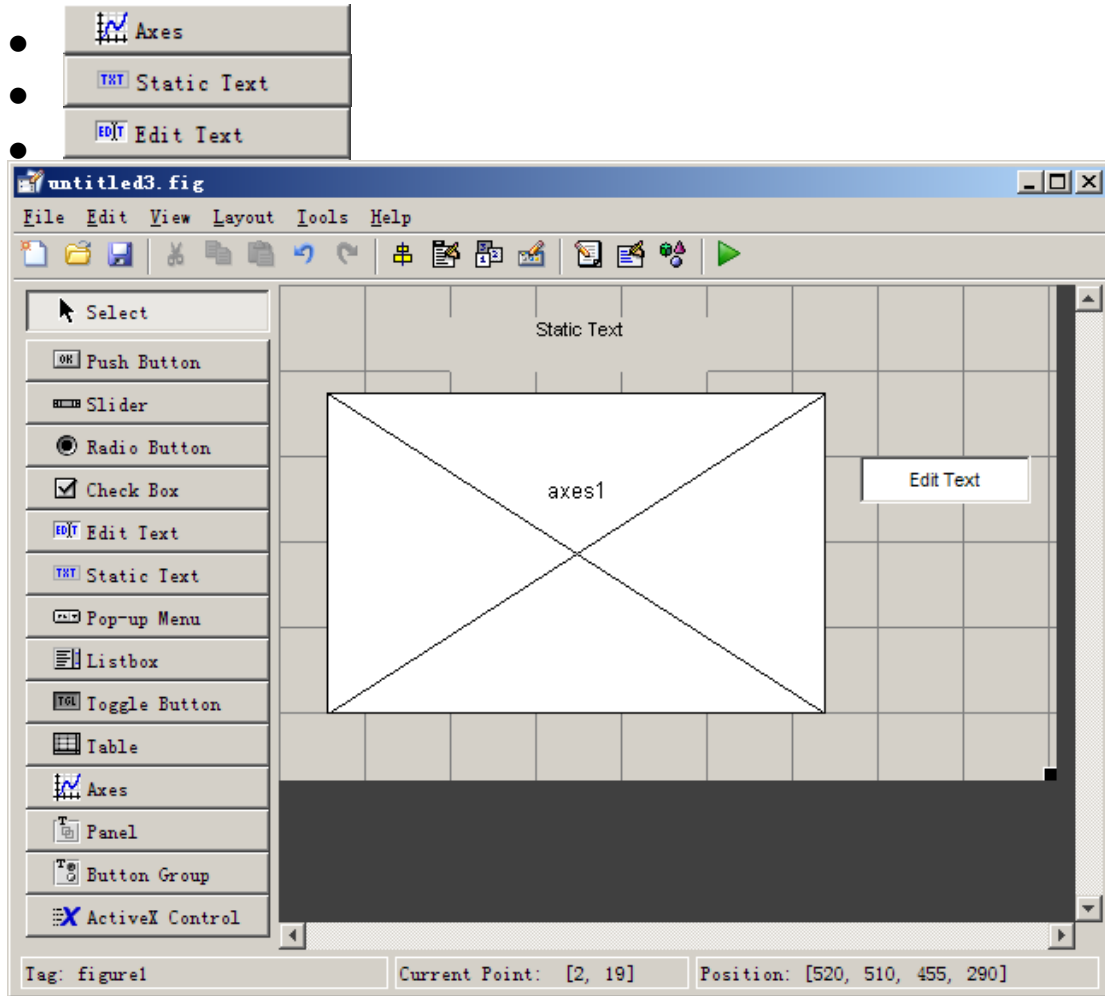


图 8.1-4 选配界面组件后的版面编辑器

(3) 界面组件的参数设置

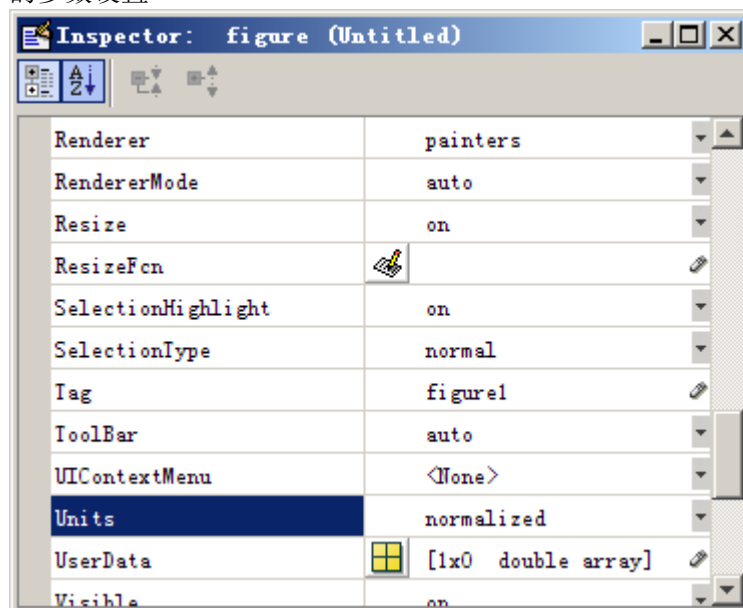


图 8.1-5 窗属性编辑器和设置的属性值

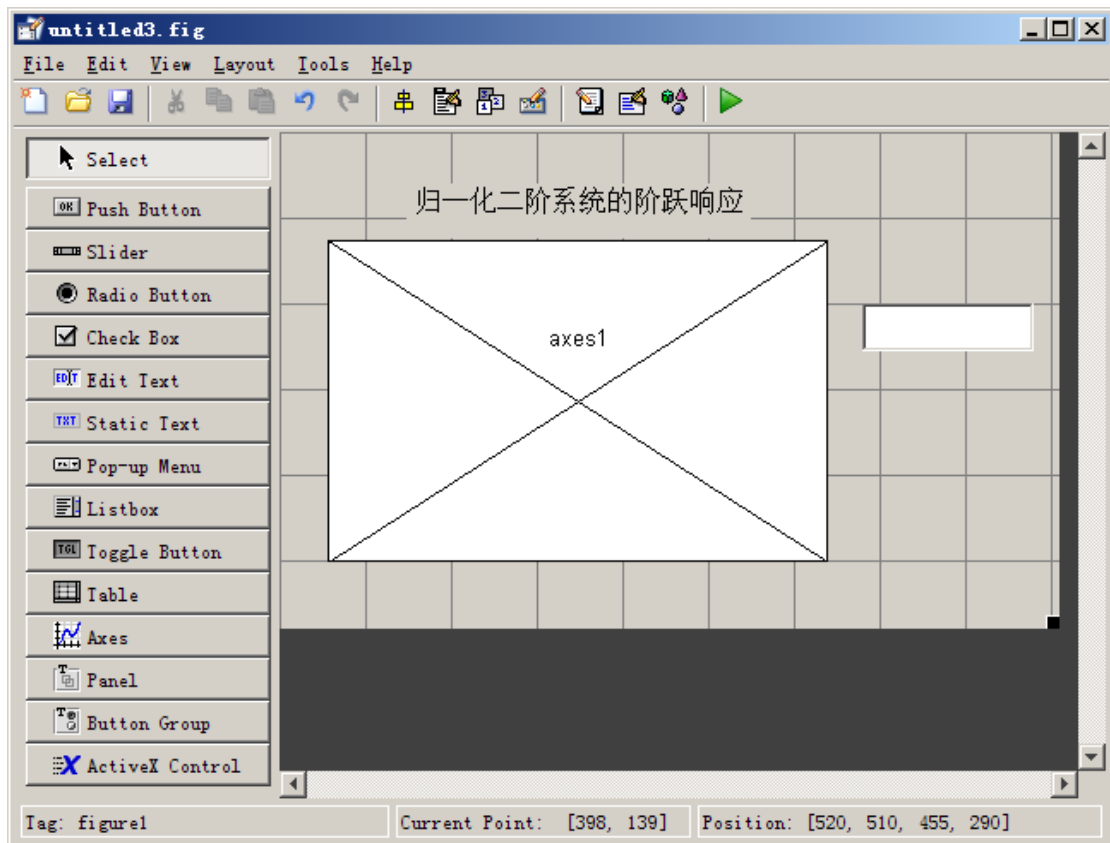


图 8.1-6 经属性设置后的设计版面

(5) 初步设计界面的保存

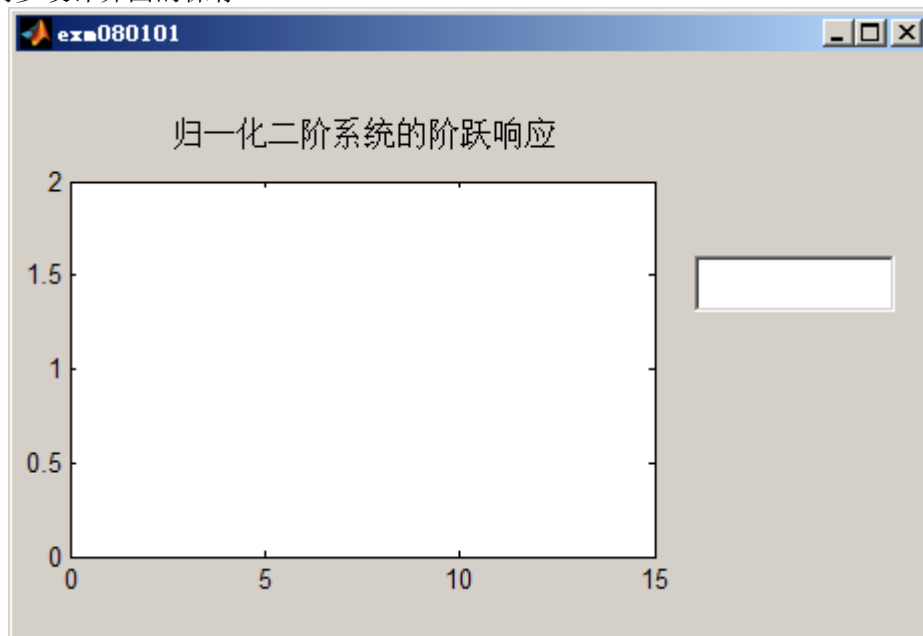


图 8.1-7 仅在外形上与题给要求一致的 GUI

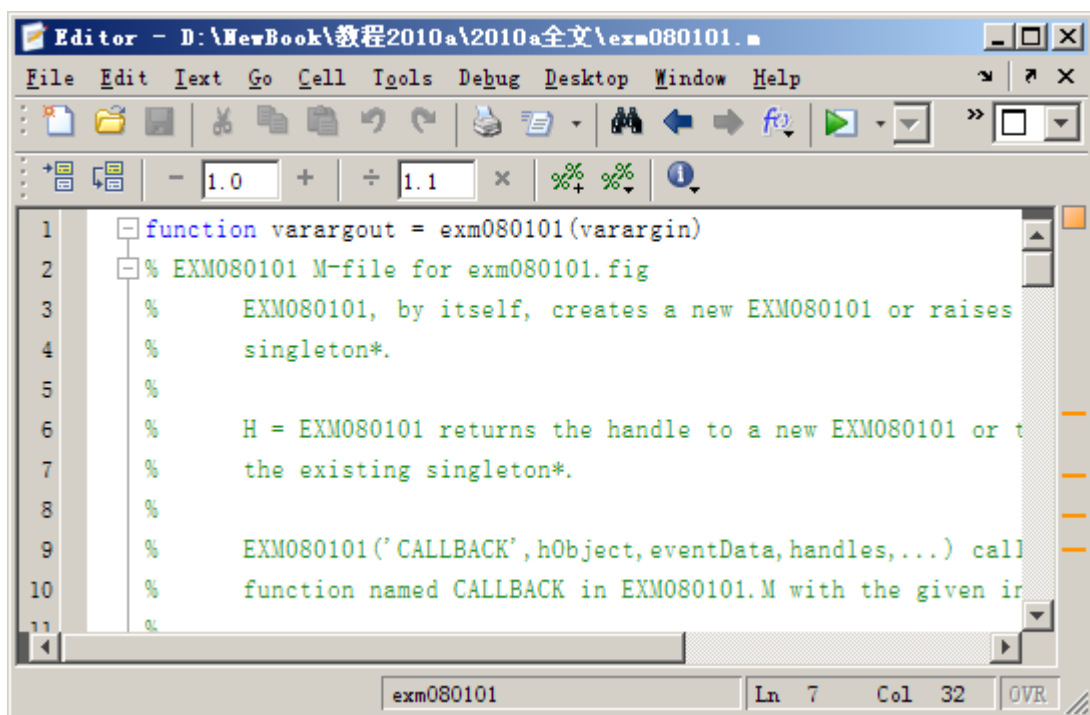


图 8.1-8 版面编辑器自动生成的设计界面 M 文件

(6) 回调程序的编写

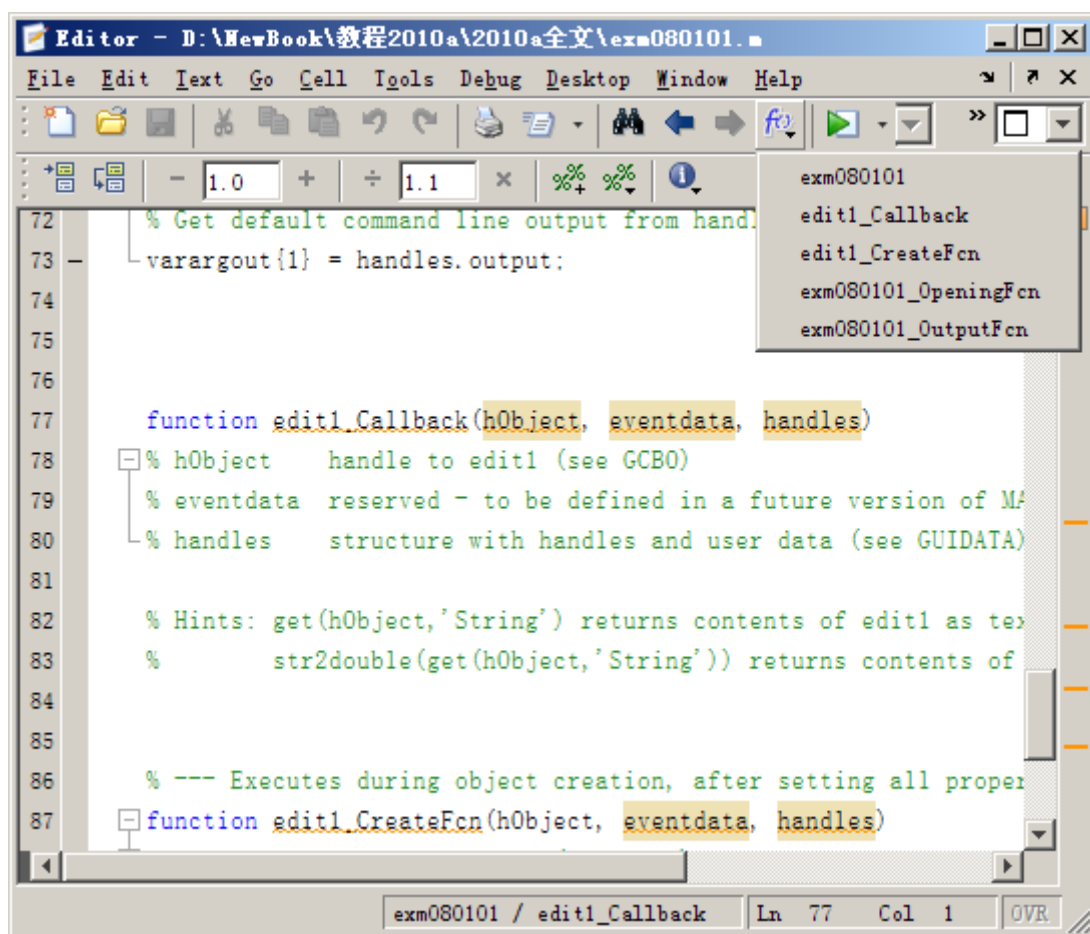


图 8.1-9 GUIDE 生成的“空白回调函数体”示图

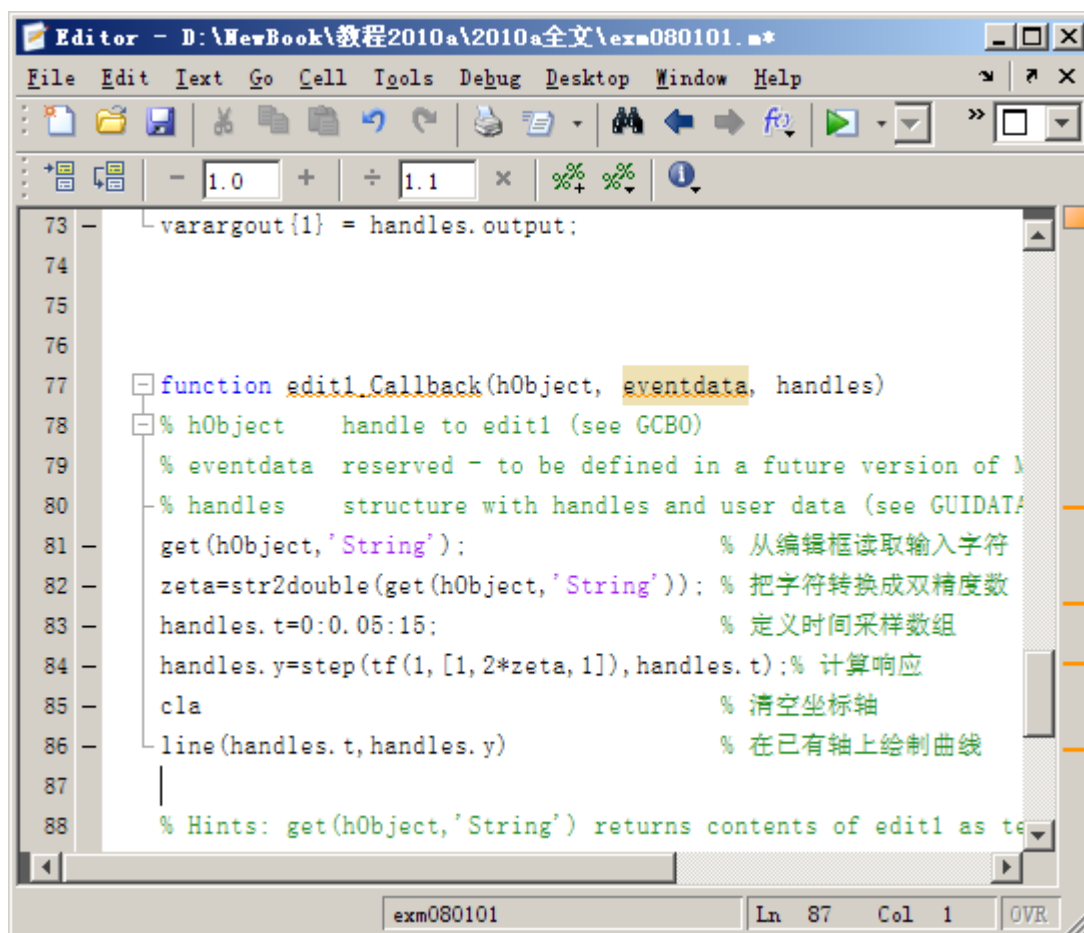


图 8.1-10 填写工作指令后的回调子函数

(7) 所建界面的使用

8.2 控件创建及应用示例

【例 8.2-1】为归一化二阶系统 $G(s) = \frac{1}{s^2 + 2\zeta s + 1}$ 单位阶跃响应制作如图 8.2-1 所示的用户

界面。要求：（1）通过编辑框和滑键都能输入阻尼比。（2）刚启动的界面初始形态如图 8.2-1 所示。（3）在刚启动的初始界面上，响应曲线用红线绘制；而一旦界面被操作，则响应曲线将用蓝线绘制。（4）在列表框中的三个选项可以任意组合。

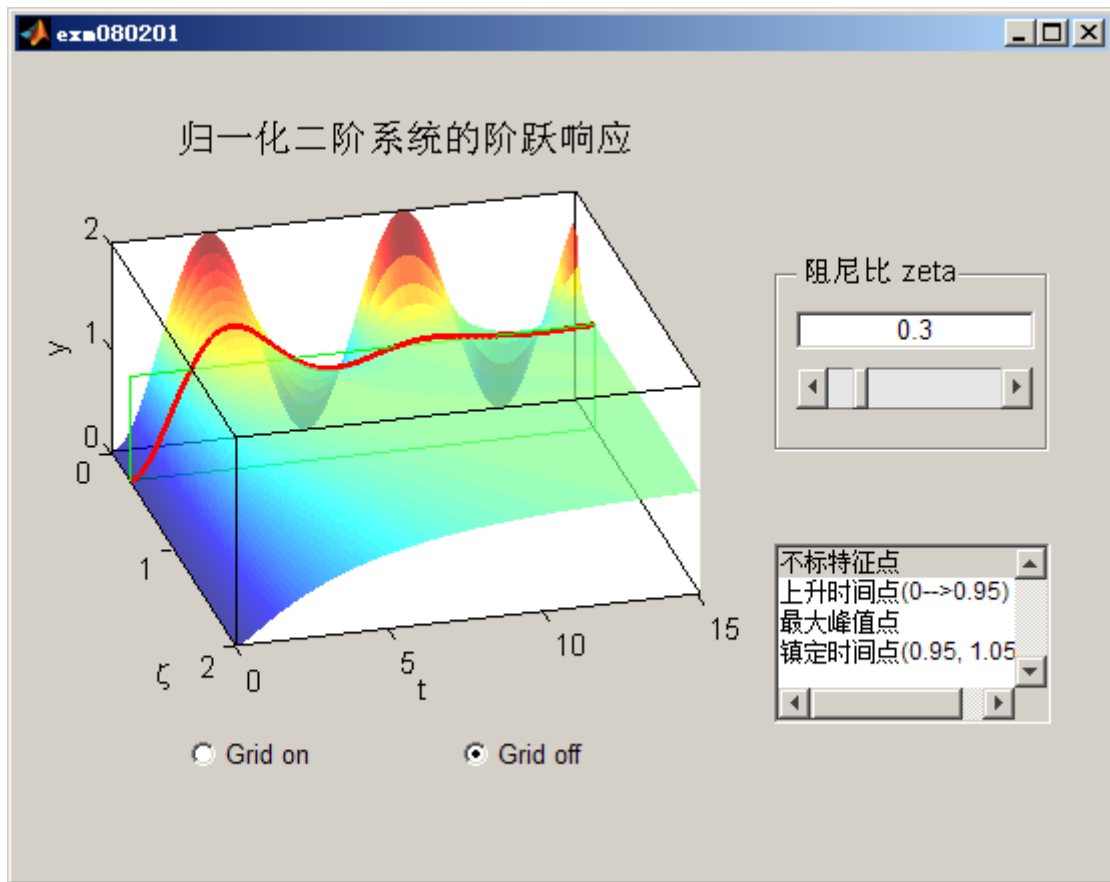


图 8.2-1 用户界面的初始状态

- (1) GUI 版面编辑器的开启
- (2) 对未来界面窗属性设置
- (3) 为版面设计区引入坐标参照系

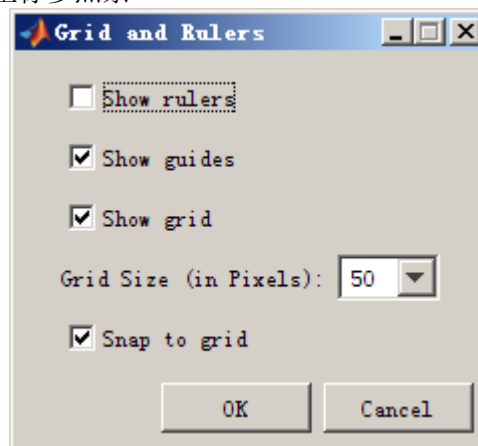


图 8.2-2 “格尺”对话框（默认状态）

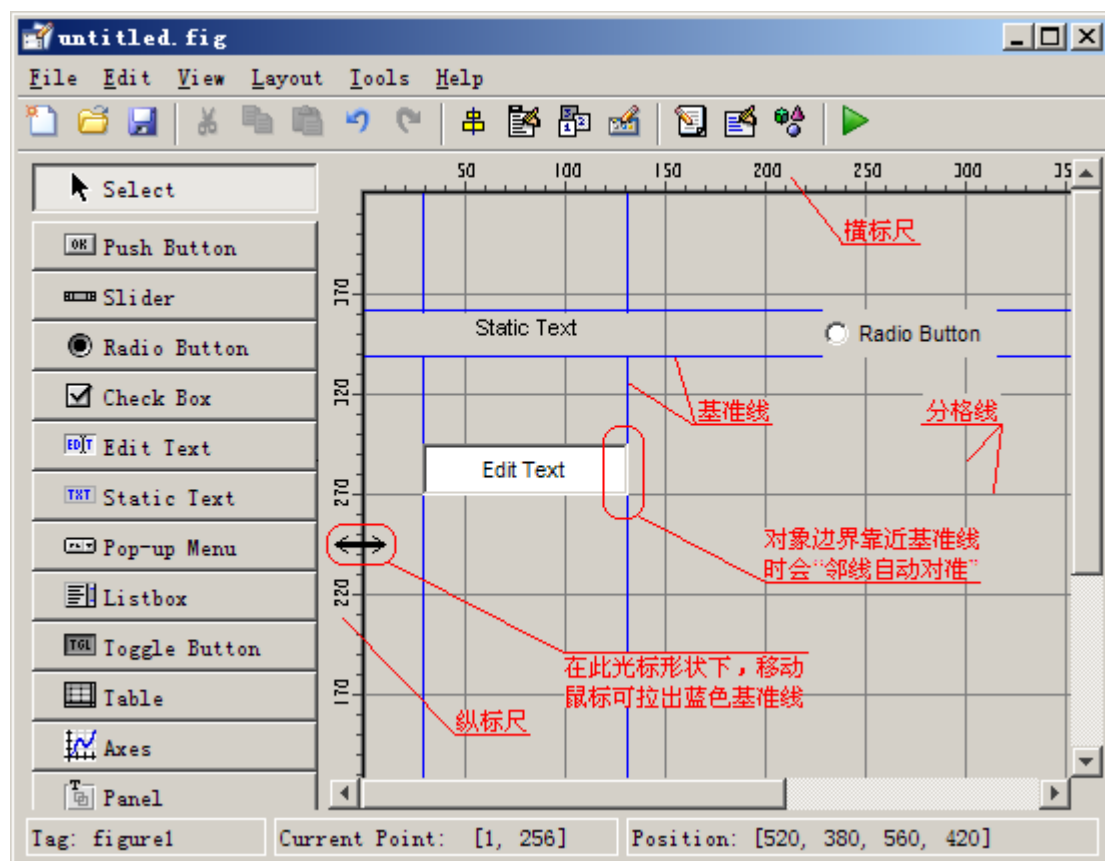


图 8.2-3 一般坐标参照工具示意

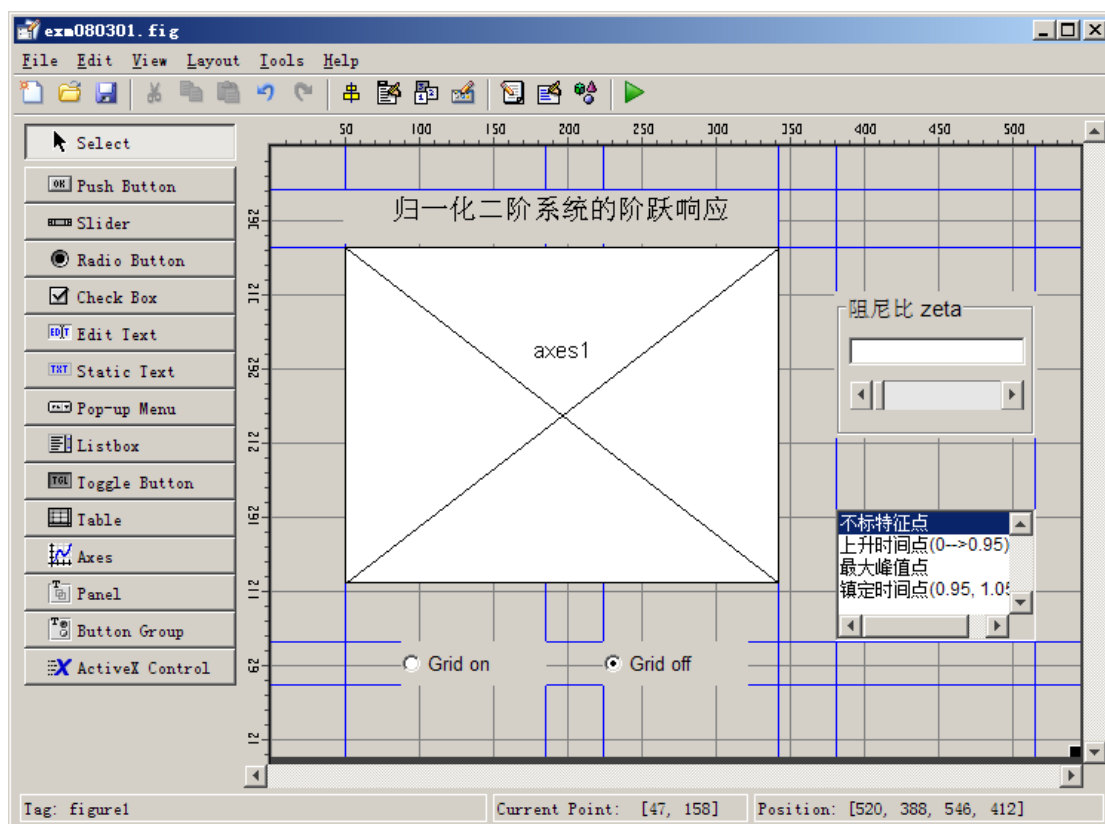


图 8.2-4 设计本例用户界面的版面编辑器

(4) 根据题目要求进行界面构建

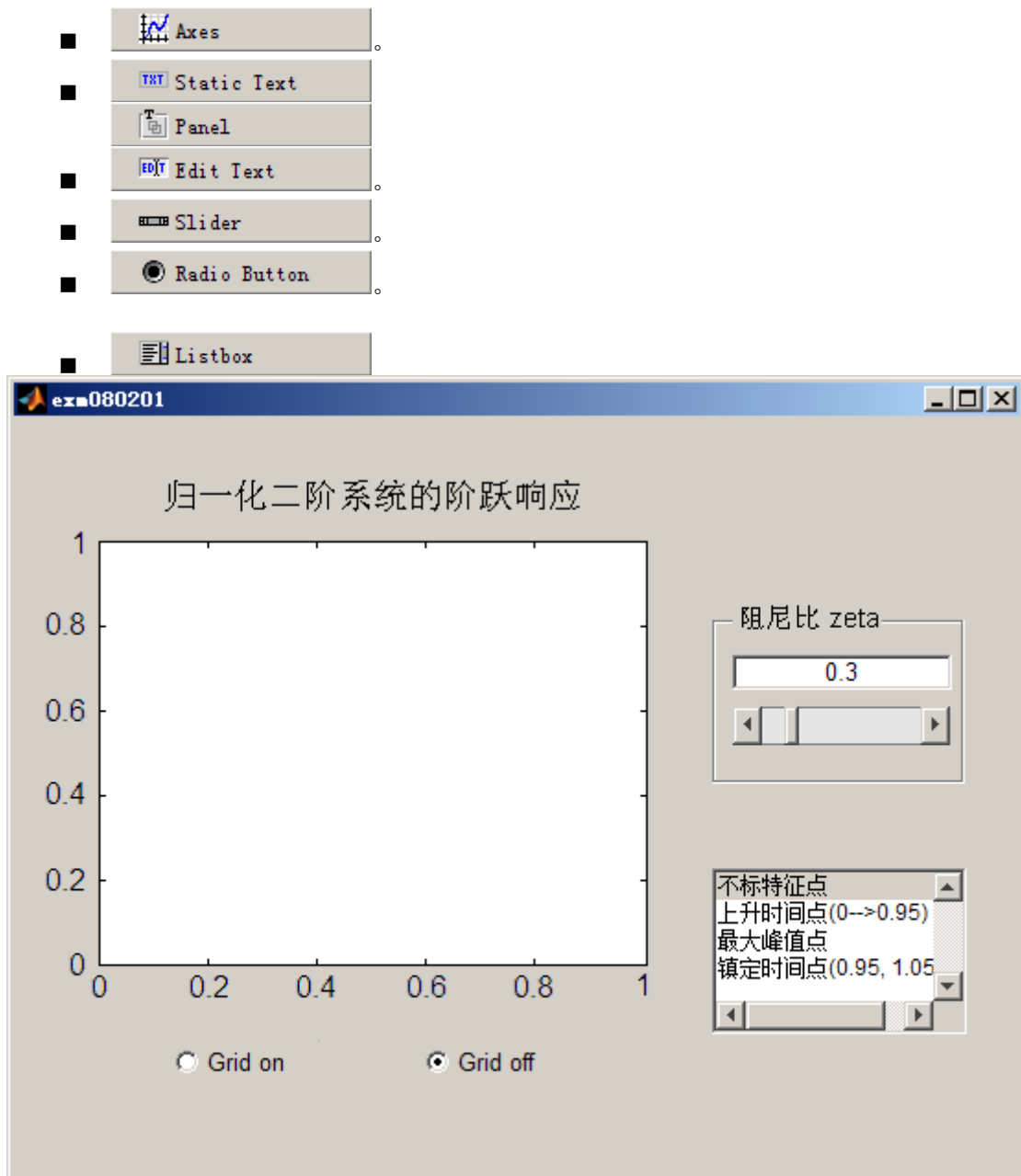


图 8.2-5 执行文件尚未填写时的生成界面

(5) 由版面编辑器自动产生的 exm080201.m 文件的结构

(6) 编写初始化程序

```
function exm080201_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to exm080201 (see VARARGIN)
%U_Start --- “%U_Stat” 和 “%U_End” 之间的 M 码均由作者编写-----U Start
zeta=0.3;
set(handles.edit1,'String',num2str(zeta))
set(handles.slider1,'Value',zeta)
set(handles.gridon,'Value',0)
set(handles.gridoff,'Value',1)
```

```

set(handles.listbox1,'Value',1)
handles.t=0:0.05:15;
handles.Color='Red';
handles.zeta=zeta;
handles.flag=0;
handles=surfplot(handles);
handles.flag=1;
handles.Color='Blue';
%U_End-----U_End
handles.output = hObject;
guidata(hObject, handles);

```

(7) 可编辑框的回调子函数

```

function edit1_Callback(hObject, eventdata, handles)
% hObject
% eventdata reserved - to be defined in a future version of MATLAB
% handles
%U_Start-----U_Start
sz=get(hObject,'String');
zeta=str2double(sz);
set(handles.slider1,'Value',zeta)
handles.zeta=zeta;
handles=surfplot(handles);
guidata(hObject, handles);
%U_End-----U_End

```

(8) 滑键回调子函数

```

function slider1_Callback(hObject, eventdata, handles)
%U_Start-----U_Start
zeta=get(hObject,'Value');
set(handles.edit1,'String',num2str(zeta))
handles.zeta=zeta;
handles=surfplot(handles);

guidata(hObject, handles);
%U_End-----U_End

```

(9) 无线电按键回调子函数

```

function gridon_Callback(hObject, eventdata, handles)
%U_Start-----U_Start
set(handles.gridoff,'Value',0)
grid on
%U_End-----U_End

function gridoff_Callback(hObject, eventdata, handles)
%U_Start-----U_Start
set(handles.gridon,'Value',0)
grid off
%U_End-----U_End

```

(10) 列表框回调子函数

```

function listbox1_Callback(hObject, eventdata, handles)
%U_Start-----U_Start
listindex=get(hObject,'Value');

```

```

if any(listindex==1)
    set(handles.listbox1,'Value',1)
end
handles.flag=0;
handles=surfplot(handles);
handles.flag=1;
guidata(hObject, handles);
%U_End-----U_End

```

(11) 绘图子函数

```

function handles=surfplot(handles)
% handles=surfplot(handles)
% handles
%
zeta=handles.zeta;
t=handles.t;
listindex=get(handles.listbox1,'Value');
Nt=length(t);
if handles.flag==0
    cla
    zmin=get(handles.slider1,'Min');
    zmax=get(handles.slider1,'Max');
    zt=zmin:0.05:zmax;
    Nz=length(zt);
    [ZT,T]=meshgrid(zt,t);
    Y=zeros(Nt,Nz);
    for k=1:Nz
        Y(:,k)=step(tf(1,[1,2*z(k),1]),t);
    end
    surface(ZT,T,Y)
    shading flat
else
    delete(handles.g1)
    delete(handles.rline)
end
xz=ones(1,Nt)*zeta;
y1=ones(1,Nt)*1;
y=step(tf(1,[1,2*zeta,1]),t);
gz=[zeta,zeta,xz,zeta,zeta,xz];
gt=[t(1),t(1),t(end),t(end),fliplr(t)];
gy=[0,1,y1,1,0,0*y1];
handles.g1=line(gz,gt,gy,'Color','g','LineWidth',1);
handles.rline=line(xz,t,y,'Color',handles.Color,'LineWidth',2);
K=length(get(handles.listbox1,'Value'));
for jj=1:K
    switch listindex(jj)
        case 1

        case 2
            k95=min(find(y>0.95));k952=[(k95-1),k95];
            t95=interp1(y(k952),t(k952),0.95);
            line(zeta,t95,0.95,'marker','+','markeredgecolor','k','markersize',6);
        case 3
            [ym,km]=max(y);
            if km<Nt & (ym-1)>0
                line(zeta,t(km),ym,'marker','.', 'markeredgecolor','k','markersize',5);
            end
    end
end

```

```

case 4
    ii=max(find(abs(y-1)>0.05));
    if ii<Nt
        line(zeta,t(ii+1),y(ii+1),'Color','r','Marker','o','MarkerSize',5)
    end
end
end
xlabel('\zeta')
ylabel('t')
zlabel('y')
alpha(0.7)
view(75,44)

```

(12) 用户界面的运行

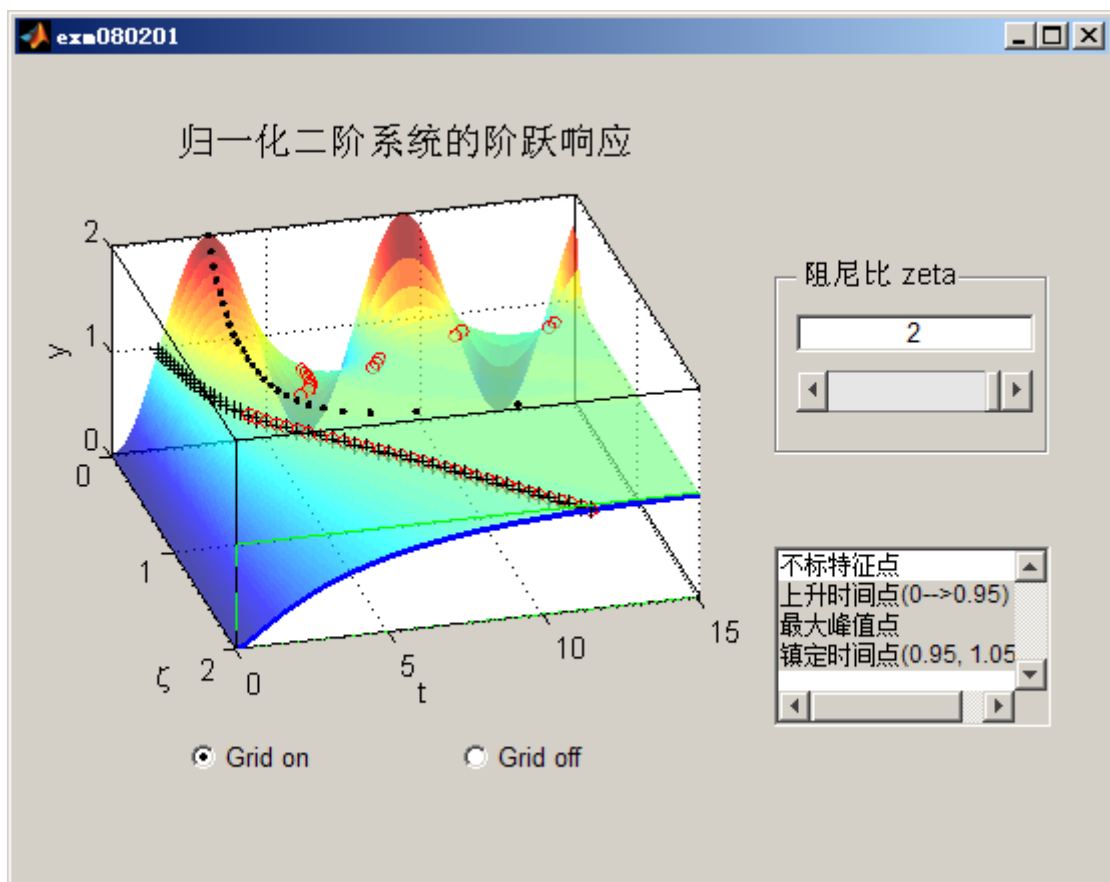


图 8.2-6 显示列表框全部选项的界面示例

【说明】

8.3 菜单及工具图标的设计示例

8.3.1 为界面配置标准菜单条和工具条

【例 8.3-1】在例 8.2-1 产生的界面上，配置 MATLAB 标准图形窗菜单。

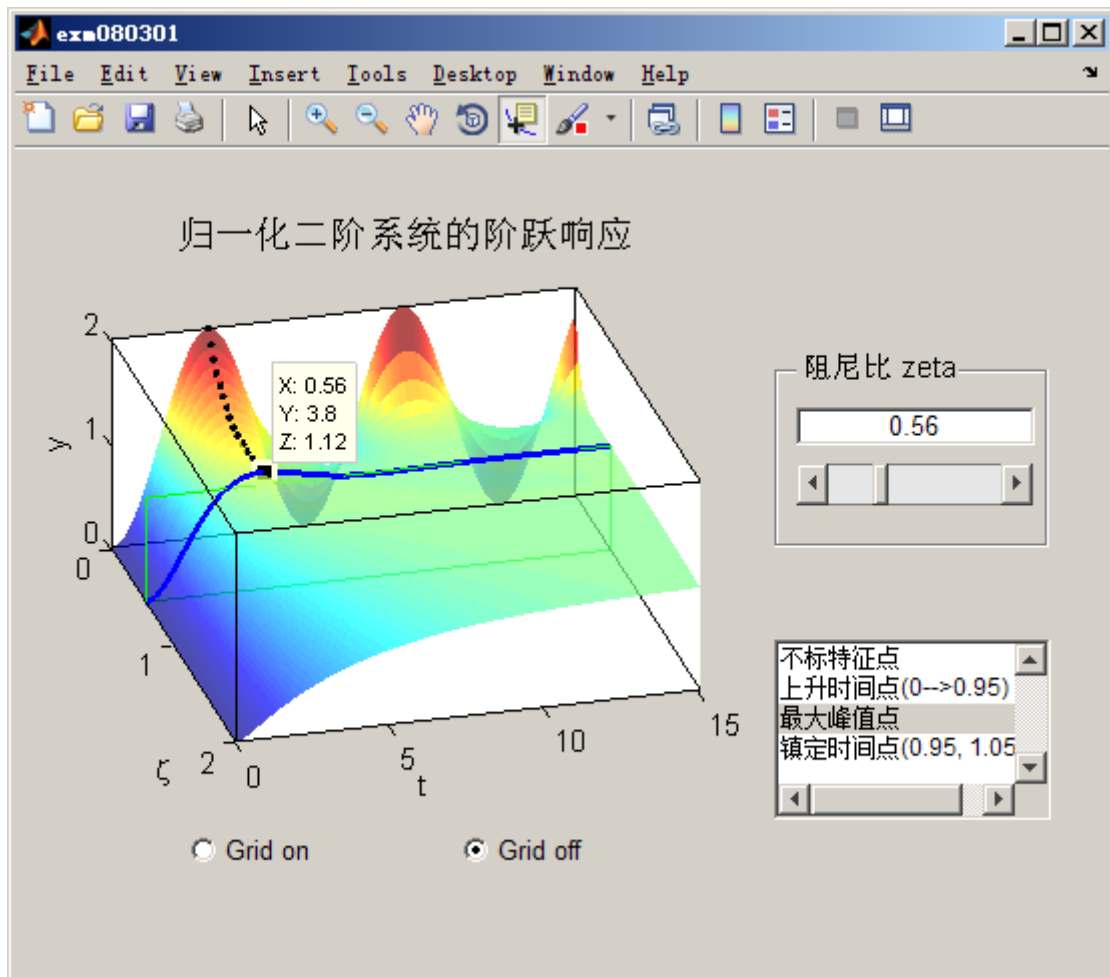



图 8.3-1 显示标准菜单和工具条的图形用户界面

- (1) 以例 8.2-1 的用户界面为基础制作新界面
- (2) 利用属性编辑器为界面配置标准菜单条和工具条
- (3) 界面操作示范

【说明】

8.3.2 菜单定制和标准图标选用

【例 8.3-2】制作如图 8.3-2 所示的图形用户界面。该用户界面上，不仅配置了一个定制菜单，用以控制界面上坐标框的是否封闭，而且还配置了一个“图形数据标识图标”。

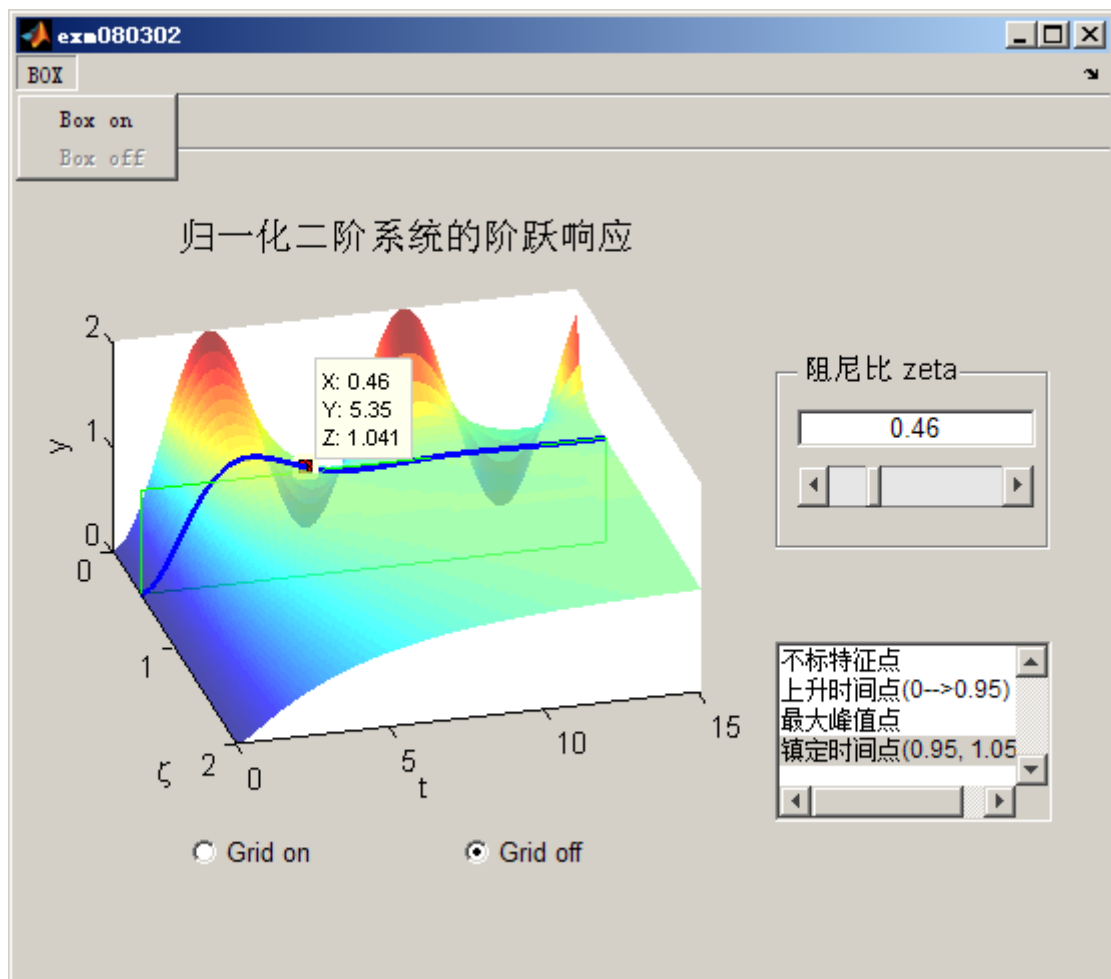


图 8.3-2 带定制菜单和选用标准工具图标的用户界面

- (1) 利用已有用户界面为基础制作新界面
- (2) 撤销原界面上的标准菜单和工具条
- (3) 定制菜单

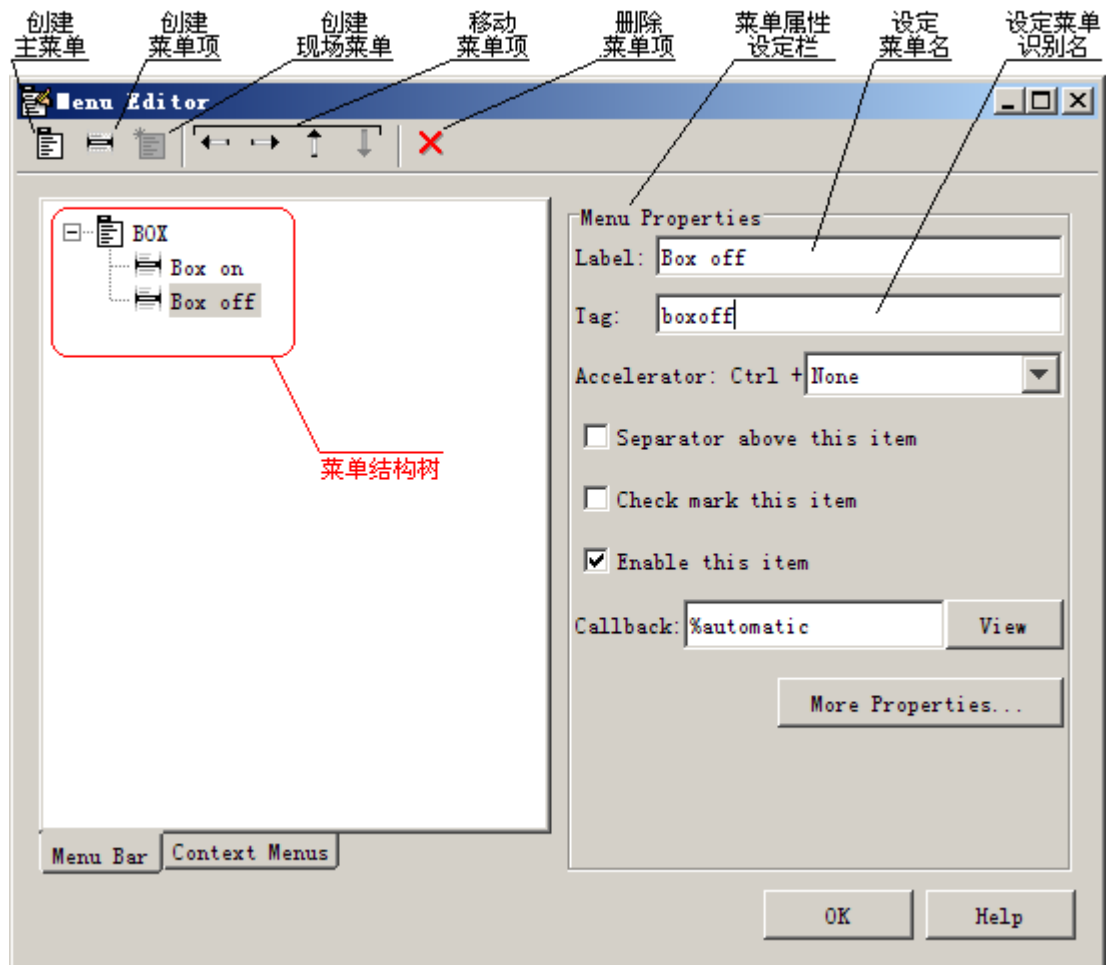


图 8.3-3 菜单编辑器的外形和功能分区

(4) 为定制菜单项编写回调子函数

```
function boxon_Callback(hObject, eventdata, handles)
%U_Start-----U_Start
box on
set(handles.boxon,'Enable','off')
set(handles.boxoff,'Enable','on')
%U_End-----U_End

function boxoff_Callback(hObject, eventdata, handles)
%U_Start-----U_Start
box off
set(handles.boxoff,'Enable','off')
set(handles.boxon,'Enable','on')
%U_End-----U_End
```

(5) 选用标准工具图标

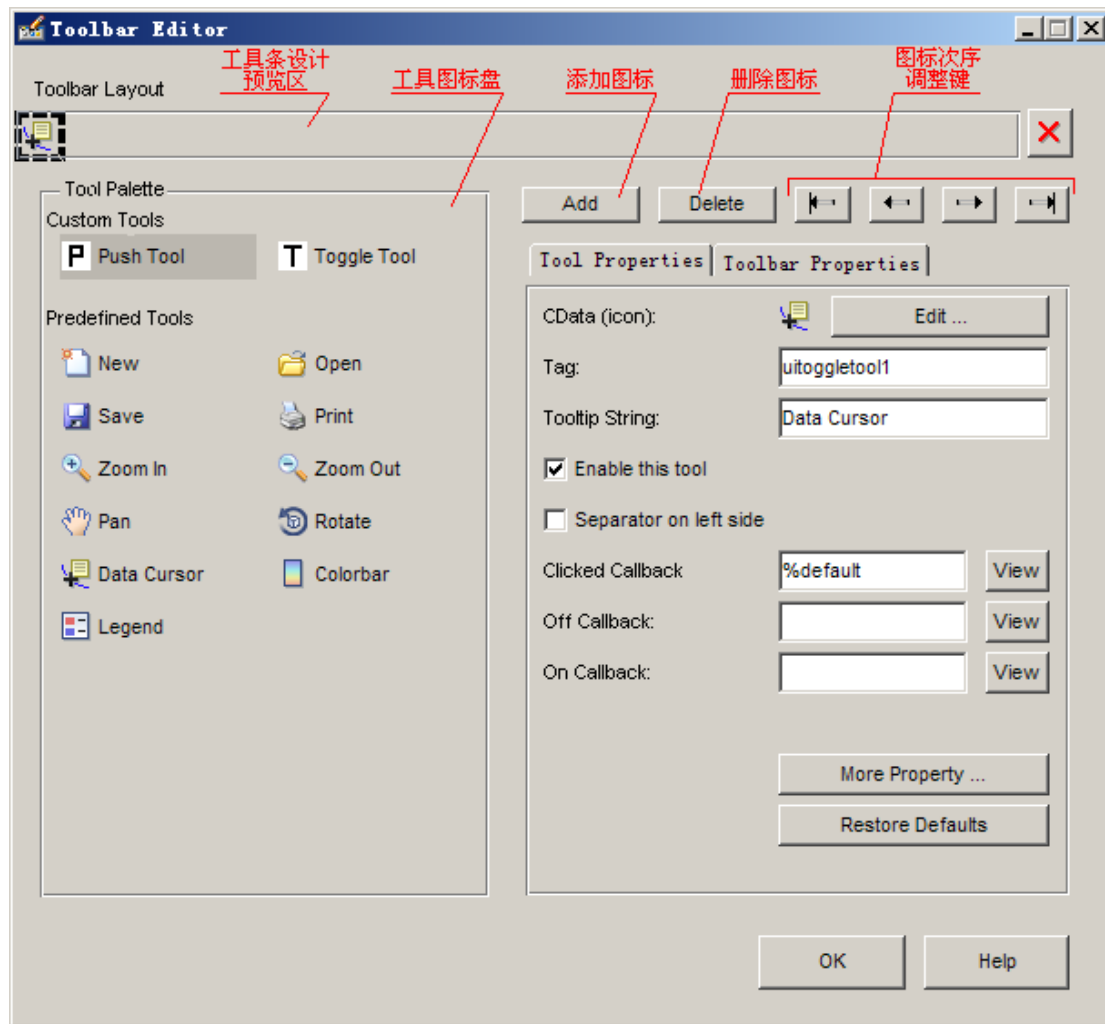


图 8.3-4 工具条编辑器的外形和功能分区

(4) 保存及生成题目所要求的用户界面

【说明】

习题 8

1. 以算例 8.3-2 的图形用户界面文件 exm080302.m 和 exm080302.fig 为基础，删除原界面上的两个无线电按键，以便获得如图 p8-1 的新图形用户界面。

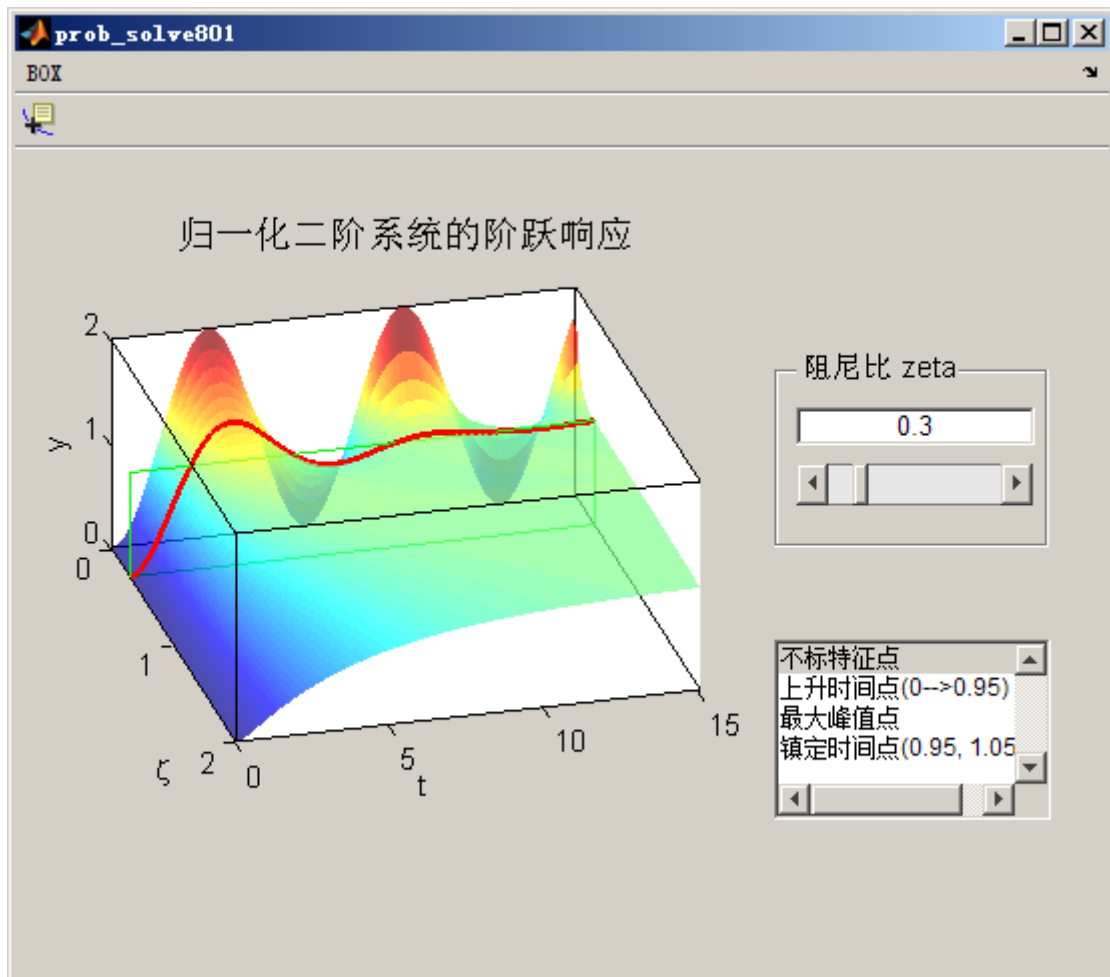


图 p8-1

2. 以算例 8.3-2 的图形用户界面文件 exm080302.m 和 exm080302.fig 为基础, 用双稳态按钮 Toggle Button 替代无线电按钮实现对坐标网格绘制的控制, 产生如图 p8-2 的新图形用户界面。

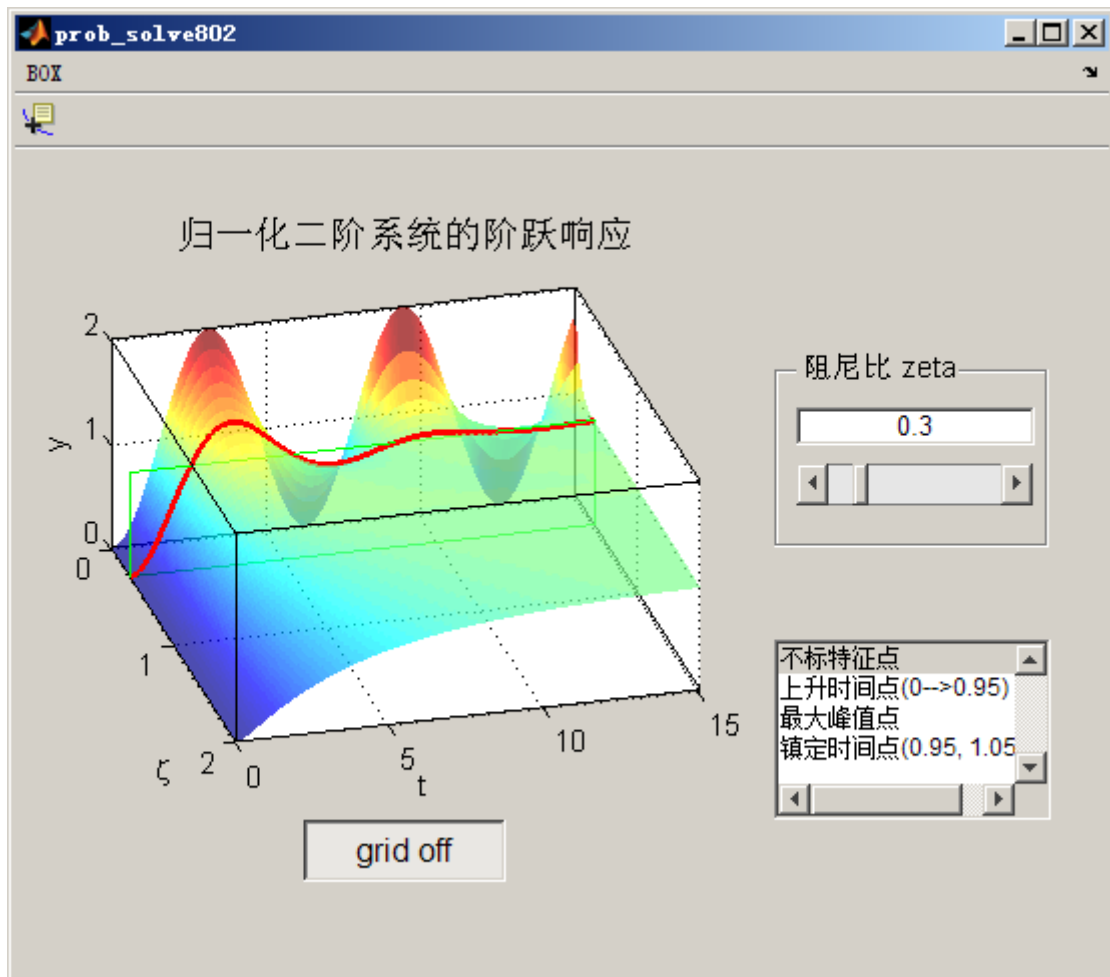


图 p8-2

第 9 章 Notebook

9.1 Notebook 的配置和启动

9.1.1 Notebook 的配置

9.1.2 Notebook 的启动

1 创建新的 M-book 文件

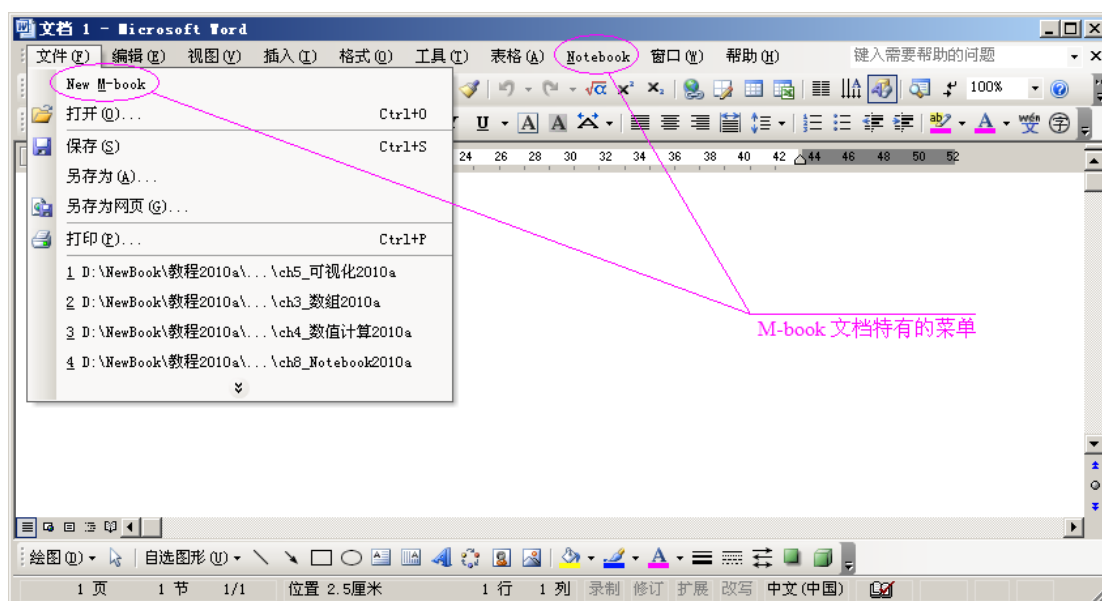


图 9.1-1 新建 M-book 文档的界面

2 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
- (2) 在资源管理器中打开已有的 M-book 文件
- (3) 在 MATLAB 当前目录窗中打开已有的 M-book 文件
- (4) 在 MATLAB 指令窗中开启已有的 M-book 文件

9.2 M-book 模板的使用

定义输入细胞	Define <u>I</u> ntput Cell	定义自初始化细胞
	Define <u>A</u> utoInit Cell	
	Define Calc <u>Z</u> one	
取消细胞定义	<u>U</u> ndefine Cells	删除选中的输出细胞
	<u>P</u> urge Selected Output Cells	
	<u>G</u> roup Cells	
	<u>U</u> ngroup Cells	
	H <u>i</u> de <u>C</u> ell Markers	
	T <u>o</u> GGLE Graph Output for Cell	
运行选中细胞	E <u>v</u> aluate <u>C</u> ell	
	E <u>v</u> aluate Calc <u>Z</u> one	
	E <u>v</u> aluate <u>M</u> -book	
	E <u>v</u> aluate <u>L</u> oop...	
	B <u>r</u> ing MATLAB to Front	引出输出数据格式及图形尺寸设置对话框
	N <u>o</u> tebook <u>O</u> ptions...	

图 9.2-1 {Notebook} 下拉菜单的常用功能项

9.2.1 输入细胞（群）的创建和运行

1 细胞（群）

2 输入细胞（群）操作示例

【例 9.2-1】

(1) 输入细胞的单纯生成法

```
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

```
x=(1:4)/4*pi;y=sin(x).*exp(x)
Y =
    1.5509    4.8105    7.4605    0.0000
```

【例 9.2-2】

```
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
```

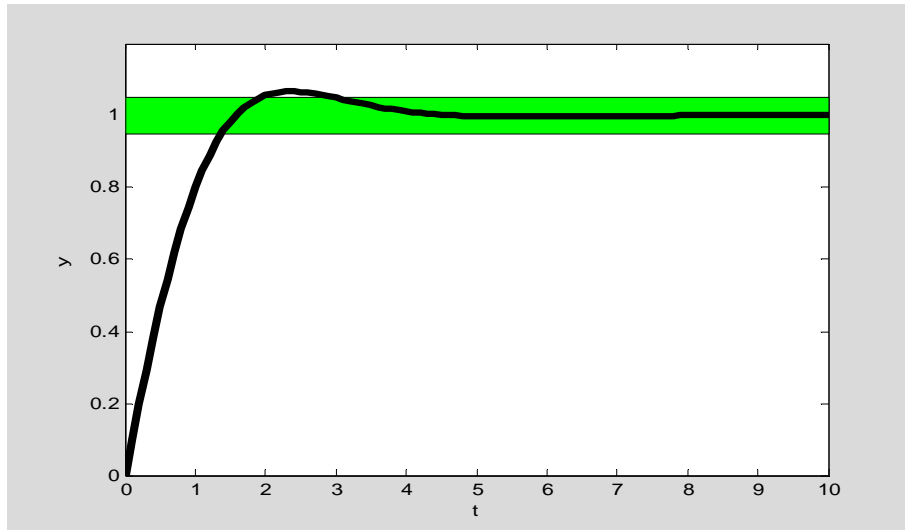


图 9.2-2 细胞群产生的完整图形

9.2.2 Notebook 菜单的其他选项

1 自初始化细胞及其应用

- (1) 自初始化细胞
- (2) 工作内存的初始化

2 删去 M-book 文件所有输出细胞

9.2.3 输出细胞的格式控制

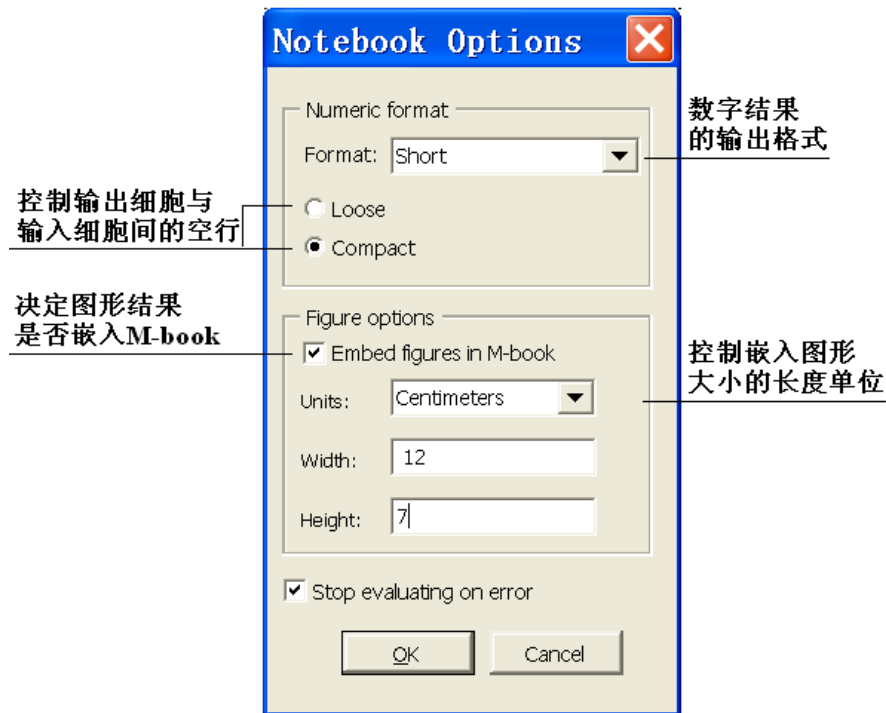


图 9.2-3 控制输出细胞格式的对话框

- 1 输出数据的表示法
- 2 输出数据间的空行控制
- 3 图形的嵌入控制

【例 9.2-3】

```

surf(peaks)           %
colormap(hot)
t=(0:50)/50*pi;y=sin(t);
plot(t,y)             %
  
```

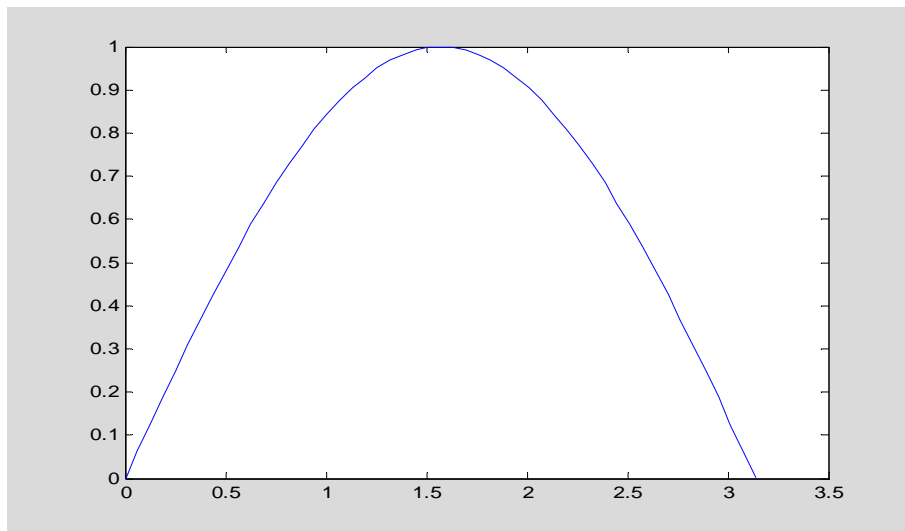


图 9.2-4 同一细胞群中最后一幅绘制的曲线图

4 嵌入图形大小的控制

5 嵌入图形的背景色问题

6 M-book 处理活动画面的能力

【例 9.2-4】

```
anim_zzy1(1),shg
```

9.3 使用 M-book 模板的若干参考技法

附录 A：字符串、胞元及构架数组

A.1 字符串数组

【例 A.1-1】

```
clear %
a=12345.6789 %
class(a) %
a_s=size(a) %
a =
    1.2346e+004
ans =
double
a_s =
     1     1
b='S' %
class(b) %
b_s=size(b) %
b =
S
ans =
char
b_s =
     1     1
whos %
    Name      Size      Bytes  Class  Attributes
    a         1x1         8  double
    a_s        1x2        16  double
    ans        1x4         8   char
    b          1x1         2   char
    b_s        1x2        16  double
```

【例 A.1-2】

(1)

```
a='This is an example.'
a =
This is an example.
```

(2)

```
size(a)
ans =
     1    19
```

(3)

```
A='这是算例。' %
A =
这是算例。
```

(4)

```
ab=[A(1:4),'A.1-2',A(5)] %
ab =
这是算例 A.1-2。
```

【例 A.1-3】

(1)

```
A=eye(2,4);           %
A_str1=int2str(A)      %
A_str1 =
1  0  0  0
0  1  0  0
```

(2)

```
rand('twister',0)
B=rand(2,4);           %
B3=num2str(B,3)        %
B3 =
0.549      0.603      0.424      0.438
0.715      0.545      0.646      0.892
```

【例 A.1-4】

```
clear                 %
a=2;                  %
w=3;                  %
t=0:0.01:10;         %
y=exp(-a*t).*sin(w*t); %
[y_max,i_max]=max(y); %
t_text=['t=',num2str(t(i_max))]; %      <7>
y_text=['y=',num2str(y_max)]; %      <8>
max_text=char('maximum',t_text,y_text); % <9>
tit=['y=exp(-',num2str(a),'t)*sin(',num2str(w),'t)']; % <11>
%
plot(t,zeros(size(t)),'k') %
hold on %
plot(t,y,'b') %
plot(t(i_max),y_max,'r.','MarkerSize',20) %
text(t(i_max)+0.3,y_max+0.05,max_text) % <16>
title(tit),xlabel('t'),ylabel('y') %
hold off
```

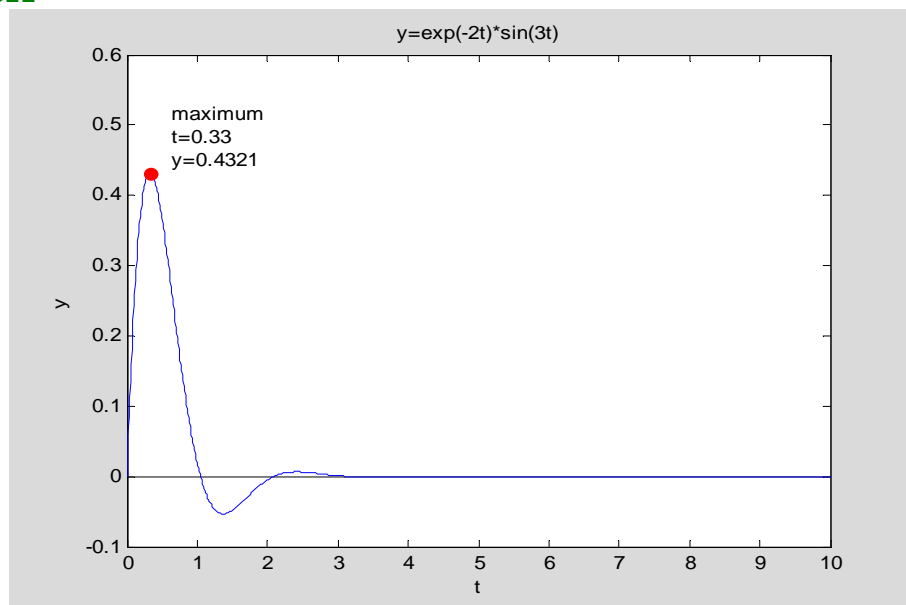


图 A.1-1 字符串运用示意图

A.2 胞元数组

【例 A.2-1】

```
clear
C_str='这是胞元数组创建算例 1';           %
R=reshape(1:9,3,3);                         %
Cn=[1+2i];                                  %
S_sym=sym('sin(-3*t)*exp(-t)');            %

%
B{1,1}=C_str;                               %<6>
B{1,2}=R;
B{2,1}=Cn;
B{2,2}=S_sym;                               %<9>

%
a=B(1,2)                                    %
class(a)
a =
    [3x3 double]
ans =
cell

%
b=B{1,2}                                    %
class(b)
b =
     1     4     7
     2     5     8
     3     6     9
ans =
double
```

A.3 构架数组

(1)

```
clear
G.name='一号房';                            %          <1>
G.volume=2000;                              %          <2>
G.temperature=[31.2,30.4,31.6,28.7];        %          <3>
G.humidity=[62.1,59.5,57.7,61.5;63,60,58.1,62.3]; %          <4>
```

(2)

```
G(2).name='二号房';
G(2).volume=2400;
```

(3)

```
G                                     %          <7>
G =
1x2 struct array with fields:
    name
    volume
    temperature
```

```

        humidity

(4)
G(1)
ans =
        name: '一号房'
        volume: 2000
        temperature: [31.2000 30.4000 31.6000 28.7000]
        humidity: [2x4 double]

(5)
G.humidity
ans =
        62.1000    59.5000    57.7000    61.5000
        63.0000    60.0000    58.1000    62.3000
ans =
    []

```

附录 B: 光盘使用说明

B.1 光盘文件的结构

在光盘上有如下两个文件夹:

- **mbook** 存放着包含本书全部算例的 M-book 形式 DOC 文档。
- **mfiles** 汇集了全书 90% 以上算例的 M 文件、MDL 文件和 FIG 文件。

B.2 光盘对软件环境的要求

- 需要 **Office2003**, **XP** 和 **MATLAB R2010a** 支持。(书中除 Simulink 模型 MDL 文件外, 其他绝大多数指令和文件在 MATLAB 其他新老版本中也都能运行。)
- 假如要完好运行 mbook 文件夹上的 DOC 文件, 则需要 **Notebook** 环境。(关于 Notebook 环境的设置参见第 8 章)

B.3 光盘文件的操作准备

在运行光盘文件之前, 应首先把 \mfiles 文件夹设置为当前目录或设置在 MATLAB 的搜索路径上(具体方法详细参见第 1.6 节)。

B.4 mbook 文件夹上 DOC 文件的使用

光盘 DOC 文件在 MATLAB **R2010a** 的 Notebook 环境中生成, DOC 文件各章节的编号、名称与印刷版完全一致。

(1) 光盘 DOC 文件的功用

- **弥补了印刷版丢失的色彩信息**

在 MATLAB 中, 用 M 文件编辑器或 Notebook 编写的指令或文件运行的结果(尤其图形)都采用不同的色彩鲜明地表现对象特征。但目前印刷版书籍出于价格和技术原因, 不得不牺牲色彩信息而采用“黑白”处理。读者借助本光盘可克服印刷版丢失色彩信息的缺憾。

- **为教师制作本教材电子讲稿提供模板**

据本人十多年 MATLAB 教学经验, 作者认为: 在 MATLAB 的课堂教学中, 电子讲稿最好使用 M-book 模板制作, 而不宜采用 PowerPoint 制作。正是出于这样的考虑, 本书光盘提供了各章的 DOC 文件。它们包含完整的章节结构和名称, 包含所有算例的题解要求和完整的解题程序。

主讲这门课程的老师, 可以根据自己经验和心得, 针对具体教学对象, 通过对本光盘 DOC 文件进行适当的剪裁, 增补数量不多但相当醒目的提示、警告和归纳性文字, 就可得到因地制宜的电子讲稿。

- **为学生完成电子作业提供模板**

学生平时作业最好使用 M-book 模板完成。一, 因为只有光盘电子版才准确描述了习题的要求(如图形色彩、光照等); 二, 只有在 M-book 文件中, 才能把文字解释、MATLAB 指令、计算结果及图形有机地结合在一起。

- **为习题提供答案**

由于有些习题答案(如图形等)必须通过光盘电子版才能较准确地表达, 所以本书相关章节的习题答案被放置在光盘中。

- 为读者提供了与印刷版对应的 Notebook 演练环境

本光盘中 DOC 文件的章节结构、算例编号与印刷版完全相同。因此在学习过程中,读者可在本光盘启动的 Notebook 环境中,或直接运行算例,观察运行结果;或改变若干指令,举一反三地观察运行结果的变化;或通过简单的复制操作,使相应指令在 MATLAB 指令窗中运行,而避免自己键入的错误。

(2) DOC 文件的开启

所有 DOC 文档都是在“Word + MATLAB R2010a”构成的 Notebook 环境中生成的。因此,在相同环境下开启是最佳选择。此时,文档具有“活性”。假如读者的 MATLAB 与 Word 联接正确,用鼠标双击光盘上的 DOC 文件,就能直接进入 Notebook 环境。

(3) 光盘 DOC 文件的使用方法

- 作为演练环境使用

在正常打开的光盘 DOC 文件中,读者只要把光标放在绿色的输入细胞内,按组合键 [Ctrl + Enter],就可使该输入细胞重新执行计算。在演练中,读者可以通过对指令的修改、变化和重新运行,观察运算结果的变化,从而达到举一反三的效果。

- 作为样板使用

先打开光盘 DOC 文件,然后删去原光盘文件内容,再写入读者自己所需的内容,最后通过菜单项的“另存为”操作保存为自己的文件。这样获得的文件能正常地在 Notebook 环境下工作,也就是既可以输入文字、公式,又可以运行 MATLAB 指令、嵌入数值或图形结果;既拥有 Word 的所有文字处理能力,又具备 MATLAB 的运算、表现能力。

B.5 mfiles 文件夹上的 M、MDL 文件的使用

所有算例都以 M 文件或 MDL 文件形式刻录在光盘的 \mfiles 文件夹中。

本光盘 M 文件应在 MATLAB R2010a 以上版本运行,涉及符号计算的应有 Symbolic Math Toolbox 3.2.3 以上版本适配;MDL 文件应有 SIMULINK7.1 以上版本适配。

(1) 光盘 M、MDL 文件的功用

- 提供可直接运作的 M 源码文件

只要有 MATLAB 环境,本光盘上的 M 文件就可以运行。它的适用条件比 \mbook 文件夹上的 DOC 文件宽松得多,也就是不管读者是否正确安装 Notebook,不管文件产生的是动画还是交互操作界面,它们都能在 MATLAB 环境中正确执行。

每个算例文件都是完整的,可在 MATLAB 环境中直接运行的,所得结果与印刷版相对应。但出于运行方式不同的考虑,有些光盘 M 文件与印刷版文件指令可能会存在少许差别,目的是为把算例特征表现得更充分。

此外,本光盘提供的 M 文件中,有许多是很通用的,读者只要稍加修改,就可为己所用。

- 弥补了印刷版没有 SIMULINK 模型文件的缺陷

由于 SIMULINK 工作特点的缘故,所以迄今为止所有涉及 SIMULINK 的印刷版书籍中都没有能直接运行的模型文件。这给读者带来许多困惑和麻烦:一,读者如想验证书中结论,那就不得不从建模做起;二,仿真模块中的参数设置常使初学者顾此失彼,从而造成仿真失败。本光盘上 MDL 模型文件都可直接在 MATLAB 中运行,进行验证。用户也可以在模型打开后,修改参数,观察变化。

(2) mfiles 文件夹上文件的放置规则

- exm 为前缀的文件都是可直接运行的算例文件

前缀后的编号与算例编号对应。最左边的两位数字为“章”编号标注,其后两位数字是“节”编号,最后两位数字是节内题号。具体举例如下:

【例 1.3-6】对应的 M 文件是 exm010306.m；

【例 7.1-2】对应的 MDL 模型文件为 exm070102.mdl；

而配用 M 文件是 exm070102s.m；

- **其他非 exm 前缀文件是被调用文件**

在 \mfiles 文件夹上还有一些不以 exm 为前缀的文件，它们不与算例直接对应，而是算例必不可少的被调用文件，或是做习题所需的演示文件、数据文件。在印刷版上可以找到有关它们的说明。

(3) M、MDL 的使用方法

直接在 MATLAB 指令窗中，运行（不带扩展名的）算例 M 文件名，就可得到相关结果。在此要再次提醒的是：必须把 \mfiles 文件夹设置在 MATLAB 的搜索路径上。

B.6 其他

在本书印刷版发行后，光盘软件的更新内容将通过光盘所带的 Readme 文件作简短的概述性说明。

附录 C MATLAB 指令索引

C.1 标点及特殊符号指令

+	加	(arith)	1.3.3, 2.1.2, 3.3.1, 4.2.1
-	减	(arith)	1.3.3, 2.1.2, 3.3.1, 4.2.1
*	矩阵乘	(arith)	1.3.3, 2.1.2, 3.3.1, 4.2.1
.*	数组乘	(arith)	1.3.3, 2.1.2, 3.3.1
^	矩阵乘方	(arith)	1.3.3, 2.1.2, 3.3.1
.^	数组乘方	(arith)	1.3.3, 2.1.2, 3.3.1
/	斜杠或右除	(slash)	3.3.1, 4.2.2
\	反斜杠或左除	(slash)	3.3.1, 4.2.3, 2.6.1
./或.\	数组除	(slash)	1.3.3, 2.1.2, 3.3.1, 3.3.2, 6.1.3
==	等号	(relop 或 eq)	2.1.2, 3.3.1, 3.5.1, 4.4.1, 6.1.3, 6.2.4
~=	不等于	(relop 或 ne)	2.1.2, 3.3.1, 3.5.1
<	小于	(relop 或 lt)	3.2.3, 3.3.1, 3.5.1
>	大于	(relop 或 gt)	3.3.1
<=	小于或等于	(relop 或 le)	3.3.1, 3.5.1
>=	大于或等于	(relop 或 ge)	3.3.1, 3.5.1
&	逻辑与	(relop 或 and)	3.3.1, 3.5.2
	逻辑或	(relop 或 or)	3.3.1, 3.5.2
~	逻辑非	(relop 或 not)	3.3.1, 3.5.2
:	冒号	(colon)	1.4.2, 3.1, 3.2.1, 3.2.3
()	圆括号	(paren)	1.3.3, 1.4.2
[]	方括号、空数组	(paren)	1.4.2, 3.2.2, 3.2.4, 3.4.2
{ }	花括号	(paren)	1.4.2, 2.7.2, 2.7.3 , 4.1.1, 5.2.2, 6.1.2, 7.1.2, 附录 A.2
@	创建函数句柄	(punct)	1.4.2, 4.1.3, 4.1.4, 6.3.2, 6.3.3, 6.4.2
.	小数点、数组运算标识、构架域	(punct)	1.4.2, 2.6.3, 附录 A.3
...	续行号	(punct)	1.3.2, 1.4.2
,	逗号	(punct)	1.3.3, 1.4.2, 3.2.2
;	分号	(punct)	1.3.3, 1.4.2, 3.2.2
%	注释号	(punct)	1.4.2, 6.2.4
=	赋值符号	(punct)	1.3.3
[,]	数组元素水平串接	(horzcat)	1.3.3
[;]	数组元素垂直串接	(vertcat)	1.3.3
'	用于形成单引号对	(punct)	1.3.3, 1.4.2, 4.1.3, 5.2.1, 5.2.2, 附录 A.1
.'	共轭转置号	(transpose)	2.1.2, 3.3.1, 4.2.1
.'	非共轭转置号	(transpose)	2.1.2, 3.3.1
_	下连符		1.4.2, 4.1.1

C.2 主要函数指令

A a

abs	模	1.3.3, 2.3.1, 3.3.1, 3.3.2
acos	反余弦	2.1.3
acosh	反双曲余弦	2.1.3
acot	反余切	2.1.3
acoth	反双曲余切	2.1.3
acsc	反余割	2.1.3
acsch	反双曲余割	2.1.3
all	所有元素均非零则为真	3.5.3
alpha	透明控制	5.3.3, 5.4.2
angle	相角	1.3.3, 3.3.1
ans	最新表达式的运算结果	1.3.2
any	有非零元则为真	3.5.3
area	面域图	5.2
asec	反正割	2.1.3, 3.3.1
asech	反双曲正割	2.1.3, 3.3.1
asin	反正弦	2.1.3, 3.3.1
asinh	反双曲正弦	2.1.3, 3.3.1
atan	反正切	2.1.3, 3.3.1
atan2	四象限反正切	2.1.3, 3.3.1
atanh	反双曲正切	2.1.3, 3.3.1
autumn	红、黄浓淡色图阵	5.3.3
axis	轴的刻度和表现	1.3.3, 3.5.1, 5.1.2, 5.2.2, 5.3.2, 5.3.3, 6.2.4

B b

bar	直方图	5.2
binocdf	二项分布累计概率	4.3.1
binopdf	二项分布概率密度	4.3.1
binornd	产生二项分布随机数组	4.3.1
blanks	空格字符	2.4.3
bode	对数频率特性曲线	7.1.2
bone	蓝色调浓淡色图阵	5.3.3
box	坐标封闭开关	4.1.1, 5.2.2, 5.3.4
break	终止最内循环	5.4.3, 6.1.3, 6.1.4
brighten	控制色彩的明暗	5.4.2
butter	ButterWorth 滤波器	7.2

C c

caxis	(伪) 颜色轴刻度	5.4.1
cd	设置当前工作目录	1.4.3, 1.7.4
cdf2rdf	复数对角型转换到	4.2.2

	实数块对角型	
ceil	朝正无穷大方向取整	3.3.1
cell	创建胞元数组	6.1.2
char	创建或转换为字符串	2.3.1, 2.9.1, 附录 A.1
charfcn	Maple 函数	2.5.3
Children	图形对象的子对象	4.3.1, 2.4.2
clabel	等高线标注	5.4.1
class	判别数据类别	2.1.1, 2.1.4, 2.3.1, 3.4.1, 3.4.2, 4.4.1, 6.4.1
clc	清除指令窗中显示内容	1.4.3, 6.1.3
clear	从内存中清除变量和函数	1.3.3, 1.4.3, 1.7.2, 2.1.5, 2.3.3, 6.1.3, 6.2.1
clf	清除当前图形窗图形	1.4.3, 2.4.3, 2.9.1, 5.2.1, 5.3.3
close	关闭图形窗	8.2.3
collect	合并同类项	2.2.3, 2.5.4
Color	图形对象色彩属性	4.3.1, 5.2.1, 5.4.3
colorbar	显示色条	5.4.1, 5.4.2
colorcube	三浓淡多彩交错色图阵	5.3.3
colordef	定义图形窗色彩	8.2.3
colormap	设置色图	2.9.1, 5.3.2, 5.3.3, 5.4.2, 5.4.3
colspace	矩阵列空间基	2.6.1
comet	彗星状轨迹图	5.4.3
comet3	三维彗星动态轨迹线图	5.4.3
compass	射线图;主用于方向和速度	5.2
cond	矩阵条件数	4.2.3
conj	复数共轭	3.3.1
continue	将控制转交给外层的 for 或 while 循环	6.1.3, 6.1.4
contour	等高线图	5.4.1
contourf	填色等高线图	5.4.1
conv	卷积和多项式相乘	4.4.1, 4.4.3
cool	青和品红浓淡色图阵	5.3.3
copper	线性变化纯铜色图阵	5.3.3
corrcoef	相关系数	4.3.2
cos	余弦	2.1.3, 3.3.1
cosh	双曲余弦	2.1.3, 3.3.1
cot	余切	2.1.3, 3.3.1
coth	双曲余切	2.1.3, 3.3.1
cov	协方差矩阵	4.3.2
csc	余割	2.1.3, 3.3.1
csch	双曲余割	2.1.3, 3.3.1
cumsum	元素累计和	4.1.2
cumtrapz	梯形法累计积分	3.1, 4.1.2, 5.2.3
D d		
dblquad	二重（闭型）数值	4.1.3

	积分指令	
deconv	解卷和多项式相除	4.4.1
del2	计算曲率	5.4.2
demos	演示函数	1.9.4
det	行列式的值	2.6.1
diag	创建对角阵, 抽取对角 向量	2.6.1, 3.2.2, 3.2.4, 6.1.3
diary	把指令窗输入记录为文件	1.4.3
diff	求导数, 差分和近似微分	2.1.3, 2.3.1 , 2.3.1, 2.3.3, 4.1.1
digits	控制符号数值的有效数 字位数	2.2.2
dir	列出目录清单	1.4.3, 1.7.4
dirac	单位冲激函数	2.5.2
disp	显示数值和字符串内容	2.3.3, 2.4.3, 2.6.3, 3.4.1
disttool	概率分布计算交互界面	4.3.1
doc	列出指定工具包中所有 函数名	1.9.2, 2.1.6
doc(symengine)	引出 MuPAD 帮助浏览器	2.1.6
docsearch	进行多词条检索	1.9.2
double	转化为双精度数值	2.2.1, 2.6.1
drawnow	刷新屏幕	5.4.3
dsolve	求解符号常微分方程	2.4.2, 2.4.3, 2.9.2
E e		
edit	打开 M 文件编辑器	1.4.3, 1.8.1
Ei	指数积分	2.1.6,
eig	矩阵特征值和特征向量	2.1.3, 2.2.4, 2.6.1, 4.2.2
emlBlock	符号表达式转换为 SIMULINK 模块	2.9.1
end	数组的最大下标, 结束 for, while, if 语句	3.1, 3.2.3, 4.1.1
eps	浮点相对误差	1.3.3, 3.3.2, 3.5.1, 4.1.1, 5.4.2
EraseMode	图形对象属性	5.4.3
erfc	补误差函数	2.1.6
error	显示错误信息	6.3.2
evalin	跨空间执行	2.1.3, 2.1.5, 2.1.6, 2.3.1, 2.3.2, 2.8.3, 2.9.1, 2.9.2
exit	关闭 MATLAB	1.4.3
exp	指数	2.1.3, 3.3.1
expand	对指定项展开	2.2.3, 2.7.2
expint	数值指数积分	2.9.3
expm	矩阵指数	2.1.3, 2.6.1
eye	单位阵	3.2.2, 3.4.2
ezcontour	画等位线简捷指令	2.8.1

ezcontourf	画填色等位线简捷指令	2.8.1
ezmesh	画网线图简捷指令	2.8.1
ezmeshc	画带等位线的网线图简捷指令	2.8.1
ezplot	画二维曲线简捷指令	2.1.3, 2.3.3, 2.4.3, 2.5.1, 2.8.1, 2.8.2
ezplot3	画三维曲线简捷指令	2.8.1
ezpolar	画极坐标曲线简捷指令	2.8.1
ezsurf	画曲面图简捷指令	2.1.3, 2.8.1, 2.8.2, 2.8.3
ezsurfz	画带等位线的曲面图简捷指令	2.8.1, 2.8.3
F f		
factor	进行因式或因式分解	2.2.3
factorial	n 的阶乘	2.5.3
false	按指定大小创建全 0 逻辑数组	3.5.3
feather	从 X 轴出发的复数向量图	5.2
feval	函数宏指令	2.9.1, 6.3.2, 6.3.3
fill	多边形填色图	4.3.1, 6.2.4
find	寻找满足条件的数组元素下标	2.5.1, 3.4.1, 5.3.4
findsym	确认表达式中自由符号变量	2.1.1, 2.3.1, 2.3.2, 2.4.3
finverse	求反函数	2.8.2
fix	朝零方向取整	3.3.1
flag	红-白-蓝-黑交错色图阵	5.3.3
fliplr	矩阵的左右翻转	3.2.4
flipud	矩阵的上下翻转	3.2.4, 5.1.2, 5.4.3
floor	朝负无穷大方向取整	3.3.1
fminbnd	求非线性函数极小值点	4.1.4
fminsearch	单纯形法求多元函数极小值点	4.1.4
for (end)	按规定次数重复执行语句	2.4.3, 3.3.2, 4.2.1, 6.1.3
format	设置数据输出格式	1.3.3, 1.4.1, 4.1.3, 4.1.4, 4.2.1, 4.4.1, 5.2.4, 6.1.3
fourier	Fourier 变换	2.5.1
fsolve	解非线性方程组	4.2.4
function	函数文件头	4.1.5
functions	观察函数句柄内涵	6.4.1
fzero	求单变量函数的零点	4.2.4
G g		
gallery	产生测试矩阵	3.2.2, 4.2.3
gca	获得当前轴的柄	5.2.2
gcf	获得当前图的柄	5.3.3

get	获得图柄	2.4.2, 4.3.1, 5.2.3, 8.1, 8.2
getframe	获得影片动画图象的帧	5.4.3
get_param	获取模块参数值	7.3
ginput	用鼠标在图上获取数据	4.1.4, 4.2.4, 5.2.4
global	定义全局变量	6.2.3
gradient	梯度	4.1.1, 5.4.2
gray	线性灰度色图阵	5.3.3
grid on/off	画坐标网格线	5.2.2, 5.3.4
guide	引出 GUIDE 开发环境	8.1, 8.2
guidata	保存用户界面更新数据	8.2
H h		
heaviside	单位阶跃函数	2.5.1, 2.5.2
help	在线帮助指令	1.9.2, 2.1.6, 6.2.4, 6.3.1, 6.3.2, 6.4.1
helpbrowser	引出帮助浏览器	1.9.3
helpdesk	引出帮助界面	1.9.3
helpwin	帮助浏览器中显示帮助信息	1.9.2
hidden	网线图消隐开关	5.3.4
hist	统计频数直方图	5.2, 4.3.2
histfit	带拟合曲线的统计频数直方图	4.3.2
hold on/off	图形的保持	2.4.3, 5.2.3
horner	转换成嵌套形式	2.2.3
hot	黑-红-黄-白交错色图阵	5.3.3
hsv	饱和色图阵	5.3.3
I i		
i, j	虚数单位	1.3.3
if-end	条件执行语句	3.3.2, 5.4.3
if-else-end	程序分支控制	6.1.1, 6.2.4
ifourier	Fourier 反变换	2.5.1
ilaplace	Laplace 反变换	2.1.3, 2.5.2
imag	复数虚部	1.3.3, 3.3.1
image	显示图像	5.4.3
impulse	系统冲激响应	7.1.2
ind2sub	据单下标换算出全下标	3.4.1
inf 或 Inf	无穷大	1.3.3, 3.4.1, 6.1.3
inline	创建内联函数	4.2.4
input	提示键盘输入	6.1.3, 6.1.4
int	计算积分	2.1.3, 2.3.3, 2.5.4, 2.9.3, 3.1
interp1	线性插值	8.2
intmax	可表达的最大正整数	1.3.3
intmin	可表达的最小负整数	1.3.3

int2str	整数转换为字符串	4.3.1, 6.1.3, 附录 A.1
inv	矩阵的逆	2.6.1, 4.2.3, 6.1.3
invhilb	求逆 Hilbert 阵	6.1.3
isa	判断指定变量类别	2.1.4, 3.5.3, 6.4.1
ischar	若是字符串则为真	3.5.3
isempty	若是空矩阵则为真	3.4.2, 3.5.3
isfinite	若是有限数则为真	3.5.3
isglobal	若是全局变量则为真	3.5.3
ishandle	是否图柄	3.5.3
isinf	若是无穷大则为真	3.5.3
isletter	串中是字母则为真	3.5.3
islogical	若是逻辑数则为真	3.5.3
isnan	若为非数则为真	3.4.1, 3.5.3
isnumeric	若是数值则为真	3.4.2, 3.5.3
isolate	Maple 的特殊指令	2.3.1
isprime	是否质数	3.5.3
isreal	若是实数矩阵则为真	3.5.3
isspace	串中是空格则为真	3.5.3
iztrans	Z 反变换	2.5.3

J j

jacobian	Jacobian 矩阵	2.3.1
jet	变异 HSV 色图阵	5.3.3
jordan	Jordan 分解	2.6.1, 4.2.2

K k

keyboard	键盘获得控制权	6.1.4
kroneckerDelta	Kronecker 单位脉冲函数	2.5.3

L l

laplace	Laplace 变换	2.1.6, 2.5.2, 2.5.4
legend	形成图例说明	2.6.1, 3.1.2, 4.1.2, 5.2.2, 5.2.3, 5.3.1
length	确定数组长度	4.2.2, 4.4.1, 4.4.2
light	灯光控制	2.8.1, 2.8.3, 5.3.4
lighting	设置照明模式	5.3.3
limit	求极限	2.3.1, 4.1.1, 6.1.3
line	创建线对象	5.4.3, 8.1
LineStyle	图形线对象属性-线型	5.2.1
LineWidth	图形属性-线宽	4.1.1, 5.2.1, 5.4.3
linmod2	从 SIMULINK 模型得到 系统的状态方程	7.1.2
linspace	线性等分向量	3.2.1
load	从磁盘调入数据变量	1.7.4

Location	图形对象属性-位置	4.1.1
log	自然对数	3.3.1, 3.4.1
log10	常用对数	3.3.1
log2	以 2 为底的对数	3.3.1
logical	将数值转化为逻辑值	3.5.3, 6.1.3
logspace	对数刻度向量	
lookfor	关键词检索	1.9.2, 6.2.4, 6.4.1
M m		
magic	魔方阵	3.2.2, 4.2.1, 4.2.2, 6.4.1
maple	进入 Maple 工作空间计算	2.1.3, 2.3.1, 2.8, 2.8.3
Marker	图形对象属性-点形	4.3.1, 5.2.1, 5.4.3
MarkerEdgeColor	图形对象属性-点边界色彩	5.2.1
MarkerFaceColor	图形对象属性-点域色彩	5.2.1
MarkerSize	图形对象属性-点大小	4.1.1, 5.2.1, 5.4.3
material	对象材质	2.8.2, 5.3.3, 5.3.4
matlabFunction	符号表达式转化为 M 函数	2.9.1, 2.9.2
max	最大值	4.3.2, 5.4.1, 8.2, 附录 A.1
mean	平均值	3.3.2, 4.3.2
mesh	三维网线图	5.3.2, 5.3.4, 5.4.2
meshgrid	用于三维曲面的分格线坐标	3.3.2, 5.3.2, 5.4.2
mfun	对 MuPAD 中若干经典特殊函数实施数值计算	2.1.3, 2.1.6, 2.9.1, 2.9.3
mfunlist	MuPAD 经典特殊函数列表	2.1.3, 2.1.6, 2.9.3
mhelp	查阅 Maple 中的库函数及其调用方法	2.1.3, 2.8, 2.8.2
min	最小值	4.3.2, 5.4.1, 8.2
minreal	状态方程最小实现	7.1.2
mkdir	创建目录	1.7.4
mod	模数求余	3.3.1, 6.1.3
more	命令窗口分页输出的控制开关	1.4.3
movie	播放影片动画	5.4.3
moviein	影片动画内存初始化	5.4.3
mtaylor	多元 Taylor 级数展开	2.3.1, 2.8.3
N n		
NaN 或 nan	非数	1.3.3, 3.2.3, 3.4.1, 4.1.2, 5.3.4
nargin	函数输入量的个数	1.3.3, 5.4.3, 6.2.4
nargout	函数输出量的个数	1.3.3, 5.4.3, 6.2.4

ndims	数组的维数	3.4.2
norm	矩阵或向量范数	3.3.2, 4.2.1, 4.2.2, 4.2.3
normcdf	正态分布累计概率	4.3.1
normpdf	正态分布概率密度	4.3.1
normrnd	产生正态分布随机数组	4.3.1
notebook	创建或打开 M-book 文件	8.1.1
null	零空间	2.6.1, 4.2.2, 4.2.3
num2str	把数值转换为字符串	4.3.1, 附录 A.1
numden	提取公因式	2.2.3

O o

ode45	高阶法解微分方程	4.1.5
ones	全 1 数组	3.2.1, 3.2.2, 3.4.2, 4.4.1
optimset	创建/编辑泛函指令的控制参数	4.1.4
orth	值空间	4.2.2

P p

pack	合并工作内存中的碎块	1.7.2
pascal	Pascal 矩阵	4.2.1
path	控制 MATLAB 的搜索路径	1.6.4
pathtool	修改搜索路径	1.6.4
pause	暂停	5.4.3, 6.1.4
pcolor	用颜色反映数据的伪色图	5.4.1
peaks	产生 peaks 图形数据	5.3.3, 5.3.4, 5.4.1
pi	3.1415926535897....	1.3.3
piecewise	MuPAD 分段函数	2.3.2, 2.5.2, 2.5.3, 2.8.2, 2.9.3
pie	饼形统计图	5.2
pink	淡粉红色图阵	5.3.3
plot	二维直角坐标曲线图	2.9.2, 3.5.1, 4.2.4, 5.2, 5.2.1
plot3	三维直角坐标曲线图	5.3.1, 5.4.3
plotyy	双纵坐标图	4.3.1, 5.2.3
polar	极坐标曲线图	5.2
poly	特征多项式, 由根创建多项式	2.6.1, 4.4.1
poly2sym	将多项式转换为符号多项式	2.9.1, 4.4.1
polyfit	多项式拟合	4.4.2
polyval	求多项式的值	4.4.1
polyvalm	求矩阵多项式的值	4.4.1
pow2	2 的幂	3.3.1
pretty	显示有理分式的易读形式	2.3.1, 2.3.3, 2.5.3, 2.7
prism	光谱色图阵	

prod 元素积

Q q

quad 低阶法数值积分 4.1.3

quadl 高阶法数值积分 4.1.3

quit 退出 MATLAB 1.4.3

quiver 二维箭头图；主用于场强、流向 5.2

R r

rand 均匀分布随机数组 3.2.1, 3.2.2, 3.4.1, 3.4.2, 4.2.1, 4.3.2

randn 正态分布随机数组 3.2.2, 4.2.3, 4.3.2

random 产生各种分布随机数组 3.2.2

randsrc 在指定字符集上产生均匀数组 3.2.2

rank 秩 2.6.1

real 复数实部 1.3.3, 3.3.1, 4.2.2, 4.4.1

realmax 最大浮点数 1.3.3

realmin 最小正浮点数 1.3.3

rec 解递推方程 2.1.6

rem 求余数 3.3.1

repmat 铺放模块数组 3.2.4, 6.1.3

reshape 矩阵变维 3.2.4, 4.2.1, 6.1.3

reset 重启 MuPAD 引擎 2.1.5, 2.2.2

residue 求部分分式表达 4.4.1

~~restart 重新启动 Maple 引擎 2.8.1~~

return 返回 1.4.3, 6.1.4, 6.2.2

roots 求多项式的根 4.4.1

rose 频数扇形图；主用于统计 5.2

rot90 矩阵逆时针旋转 90 度 3.2.4

rotate 旋转指令 5.4.3

round 四舍五入取整 3.3.1, 6.1.3

rref 转换为行阶梯形 2.6.1, 4.2.2

S s

save 把内存变量存入磁盘 1.7.4

sec 正割 2.1.3, 3.3.1

sech 双曲正割 2.1.3, 3.3.1

set 设置图形对象属性 2.4.3, 2.5.1, 4.3.1, 5.2.2, 5.2.3, 5.3.3, 5.4.3, 8.2

shading 图形渲染模式 2.9.1, 5.3.3

shg 显示图形窗 5.1.2, 6.2.4, 6.3.2, 8.2.3

sign 函数符号, 符号函数 2.3.1, 3.3.1

simple 运用各种指令化简符号表达式 2.2.3, 2.3.2, 2.5.1, 2.5.4

simplify	恒等式简化	2.2.3
simulink	打开 SIMULINK 集成环境	7.1.1
sin	正弦	2.1.3, 2.3.1, 3.3.1, 3.4.1
sinh	双曲正弦	2.1.3, 3.3.1
size	确定数组大小	2.1.4, 3.3.2, 3.4.1, 3.4.2, 4.2.1
slice	切片图	5.4.2
solve	求解代数方程组	2.1.3, 2.2.4 , 2.6.3, 4.2.4
sort	对数组或向量中元素排序	2.7
sphere	产生球面数据	5.3.3, 5.3.4
spinmap	颜色周期性变化操纵	5.4.3
spring	青、黄浓淡色图阵	5.3.3
sqrt	平方根	2.3.3, 3.3.1
square	轴属性为方型	1.3.3, 6.2.4
ss	产生状态方程 LTI 对象	7.1.2
stairs	阶梯形曲线图	5.2, 5.2.3
state	用于设置 randn 随机数 发生器状态的关键词	3.2.2, 4.2.3, 4.3.2
std	标准差	4.3.2
stem	杆图	4.4.3, 5.2, 5.2.3
stem3	三维离散杆图	5.3.2
step	计算阶跃响应	8.1
str2double	把字符串转换为双精度数	8.1
str2func	创建函数句柄	6.4.1
strcmp	比较字符串	6.1.2
String	图形对象属性-字符串	4.3.1
subexpr	运用符号变量置换子表 达式	2.2.4, 2.6.1
subplot	创建子图	2.8.1, 3.5.1, 3.5.2, 4.4.3, 5.2.2, 5.2.3
subs	通用置换指令	2.2.4, 2.3.1, 2.3.3, 2.5.1, 2.5.4, 2.7.2, 2.8.2
sum	元素和	4.1.2, 6.1.3
summer	绿、黄浓淡色图阵	5.4.3
surf	三维表面图	5.3.2, 5.3.4, 5.4.2
surfc	带等高线的三维表面图	5.3.4
surface	绘制曲面的底层指令	8.2
svd	奇异值分解	2.6.1
switch-case	多个条件分支	6.1.2, 6.2.4, 6.3.2
sym	产生符号对象	2.1.1, 2.1.5, 2.3.1, 2.2.1, 2.2.2, 2.3.3, 2.5.2, 2.9.1
symengine	MuPAD 符号计算引擎	2.1.5, 2.1.6, 2.2.2, 2.3.1, 2.3.2, 2.5.3, 2.8.3
syms	定义基本符号对象	2.1.1, 2.1.5, 2.3.1, 2.5.1
symsum	符号序列的求和	2.3.2, 6.1.3
symvar	认定基本或自由符号变量	2.1.1, 2.1.2, 2.3.1, 2.3.2, 2.3.3, 2.4.3, 2.6.3
sym2poly	符号多项式转换为 数值多项式系数	2.9.1

T t

tan	正切	2.1.3, 3.3.1
tanh	双曲正切	2.1.3, 3.3.1
taylor	Taylor 级数	2.3.1
text	图形上文字标注	2.4.3, 4.3.1, 4.4.3, 5.2.2
tf	产生传递函数 LTI 对象	7.1.2, 8.1
tfdata	从对象中提取传递函数 分子分母多项式系数	7.1.2
tic	秒表起动	4.2.3, 6.1.3
title	图形名	2.4.3, 3.5.1, 5.2.2
toc	秒表终止和显示	4.2.3, 6.1.3
trace	迹	4.2.1
trapz	梯形数值积分	4.1.2
tril	下三角分解	2.6.1
triu	上三角分解	2.6.1
true	按指定大小创建全 1 逻辑数组	3.5.3
triplequad	三重（闭型）数值积分指令	4.1.3
twister	用于设置 rand 指令随机数发生器状态的关键词	3.2.1, 3.4.1, 4.2.1, 4.3.2, A.1
type	显示文件内容	1.4.3

V v

var	求方差	4.3.2
version	MATLAB 版本	
vectorize	字符串表达式向量化	2.9.1
view	设定 3-D 图形观测点	2.8.2, 5.3.2, 5.3.3
vpa	给出数值型符号结果	2.2.2, 2.3.3, 4.1.4

W w

which	确定指定文件所在的目录	1.4.3, 3.4.2, 6.4.2
while end	不确定次数重复执行语句	3.3.2, 6.1.3
whitebg	图形底色控制	8.2.3
who	列出工作内存中的变量名	1.7.2, 1.7.4
whos	列出工作内存中的变量 细节	1.7.2, 2.1.4
winter	蓝、绿浓淡色图阵	5.3.3

X x

xlabel	X 轴名标注	3.5.1, 5.2.2, 5.3.1, 5.3.2
xor	异或	3.5.2

Y y

Ycolor	图形对象属性-纵轴颜色	4.3.1
ylabel	Y 轴名标注	3.5.1, 4.3.1, 5.2.2, 5.3.1, 5.3.2

Z z

zeros	全零矩阵	3.2.2, 3.4.2, 4.2.1, 5.3.4, 6.1.3
zlabel	Z 轴名标注	5.3.1, 5.3.2
zoom	二维图形的变焦放大	4.2.4
ztrans	Z 变换	2.5.3

C.3 Simulink 模块

Add	求和模块	7.1.1,7.1.2
Breaker	开关	7.3
Current Measurement	电流测量器	7.3
Dc Voltage Source	直流电压源	7.3
Digital Filter Design	数字滤波器设计模块	7.2
Discrete Filter	离散滤波器模块	7.1.2
Gain	增益模块	7.1.1,7.1.2
In1	输入端口模块	7.1.2
Integrator	连续函数积分	7.1.1
Out1	输出端口模块	7.1.2
Parallel RLC Branch	RLC 并联支路	7.3
Powergui	营造 SimPowerSystems 仿真环境	7.3
Random Number	随机数模块	7.2
Random Source	随机信号模块	7.1.2
Scope	示波模块	7.1.1,7.1.2
Series RLC Branch	RLC 串联支路	7.3
Simulink	SIMULINK 基本库	7.1.1
Sine Wave	正弦波输出	7.1.2
Transfer Fcn	传递函数模块	7.1.2
Voltage Measurement	电压测量器	7.3