

附录 A MATLAB 基础

MATLAB 是由美国 Mathworks 公司推出的一个科技应用软件, 其名字来源于矩阵 (matrix) 和实验室 (laboratory) 两词的前 3 个字符。它是一种广泛应用于工程计算及数值分析领域的新型高级语言, 可以把科学计算、结果可视化和编程都集中在一个使用非常方便的环境中。MATLAB 功能强大、简单易学、编程效率高, 深受广大科技工作者的欢迎。在国际学术界, MATLAB 已经被确认为是准确、可靠的科学计算标准软件。

MATLAB 特点如下:

(1) MATLAB 是一个交互式软件系统, 输入一条命令, 立即就可以得出该命令的结果。

(2) 数值计算功能。以矩阵作为基本单位, 但无需预先指定维数 (动态定维); 按照 IEEE 的数值计算标准进行计算; 提供十分丰富的数值计算函数, 方便计算, 提高效率; 命令与数学中的符号、公式非常接近, 可读性强, 容易掌握。

(3) 强大的符号运算功能。

(4) 绘图功能。提供了丰富的绘图命令, 能实现一系列可视化操作。

(5) 编程功能。具有程序结果控制、函数调用、数据结构、输入输出、面向对象等程序语言特征, 而且简单易学、编程效率高。

(6) 丰富的工具箱。工具箱实际上是用 MATLAB 的基本语句编成的各种子程序集, 用于解决某一方面的专门问题或实现某一类的新算法。工具箱可分为功能型和领域型。功能型工具箱主要用来扩充 MATLAB 的符号计算功能、图形建模仿真功能、文字处理功能以及与硬件实时交互等功能, 能用于多种学科。领域型工具箱专业性很强, 如控制系统工具箱 (Control System Toolbox)、信号处理工具箱 (Signal Processing Toolbox)、符号数学工具箱 (Symbolic Math Toolbox)、统计工具箱 (Statistics Toolbox)、优化工具箱 (Optimization Toolbox)、财政金融工具箱 (Financial Toolbox) 等。

A.1 MATLAB 语言的数据结构

强大方便的数值运算功能是 MATLAB 语言的最显著特色之一。从计算精度要求出发, MATLAB 最常用的数质量为双精度浮点数, 占 8 个字节 (64 位) 遵从 IEEE 计数法, 有 11 个指数位、53 位尾数及一个符号位, 值域的近似范围为 $-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$, 其 MATLAB 表示为 double。MATLAB 最基本的数据结构为复数双精度浮点矩阵。考虑到一些特殊的应用, 如图像处理, MATLAB 语言还引入了无符号的 8 位整型数据类型, 其 MATLAB 表示为 uint8, 其值域为 0 ~ 255, 这样可以大大节省 MATLAB 的存储空间, 提高处理速度。此外, 在 MATLAB 中还可以使用其他的数据类型, 如 int8, int16, int32, uint16 和 uint32 等, 每个类型后面的数字表示其位数, 其含义不难理解。

除了数值运算外, MATLAB 及其符号运算工具箱还可以进行公式推导和解析求解, 这时需要使用符号型变量。符号型变量可以由 syms 命令来定义。

为方便编程, MATLAB 还允许其他更高级的数据类型, 如字符串、多维数组、结构数组、细胞数组、类和对象等。

A.1.1 常量和变量

1. 变量

MATLAB 中的变量可用来存放数据, 也可用来存放向量或矩阵, 并进行各种运算。

变量的命名规则是: (1) 变量名区分大小写; (2) 变量名以字母开始, 可以由字母、数字、下划线组成, 但不能使用标点符号; (3) 变量名长度不超过 63 位, 最多只能含有 63 个字符, 后面的字符无效。

2. 常量

在 MATLAB 语言中还为特定常数保留了一些名称, 虽然这些常量都可以重新赋值, 但建议在编程时应尽量避免对这些量重新赋值。

(1) eps: 机器的浮点运算误差限。PC 级上 eps 的默认值为 2.2204×10^{-16} , 若某个量的绝对值小于 eps, 则从数值运算的角度可以认为这个量为 0。

(2) i 和 j: 若 i 或 j 量不被改写, 则它们表示纯虚数单位 i 。但在 MATLAB 程序编写过程中经常事先改写这两个变量的值, 如在循环过程中常用这两个变量来表示循环变量。如果想恢复该变量, 则可以用下面的形式设置: $i = \text{sqrt}(-1)$, 即对 -1 求平方根。

(3) **inf**: 无穷大量 $+\infty$ 的 MATLAB 表示。同样地, $-\infty$ 可以表示为 **-inf**。在 MATLAB 程序执行时, 即使遇到了以 0 为除数的运算, 也不会终止程序的运行, 而只给出一个“除 0”警告, 并将结果赋成 **inf**, 这样的定义方式符合 IEEE 的标准。

(4) **NaN**: 不定式 (not a number), 通常由 0/0 运算、**inf/inf** 及其他可能的运算得出。

(5) **pi**: 圆周率 π 的双精度浮点表示。

3. 符号变量

在 MATLAB 中进行符号运算时, 需要先用 **syms** 命令创建符号变量和表达式, 如:

```
>>syms x
```

其中的 **>>** 为 MATLAB 的提示符, 由软件自动给出, 在提示符下可以输入各种各样的 MATLAB 命令。

syms 不仅可以声明一个变量, 还可以指定这个变量的类型, 比如:

声明变量 **x, y** 为实数类型, 可用命令: **>>syms x y real**

声明变量 **x, y** 为整数类型, 可用命令: **>>syms x y integer**

4. 变量的查询与清除

在命令窗口中, 只要输入 **who**, 就可以看到工作空间中所有曾经设定并至今有效的变量。如果输入 **whos**, 不但会显示所有的变量, 而且会将该变量的名称、性质等都显示出来, 即显示变量的详细资料。输入 **clear**, 就清除工作空间中的所有变量。如果输入 **clear** 变量名, 只清除工作空间中指定变量名的变量。

A.1.2 赋值语句

MATLAB 的赋值语句有下面两种结构:

(1) 直接赋值语句

其基本结构为:

赋值变量=赋值表达式

这一过程把等号右边的表示式直接赋给左边的赋值变量, 并返回到 MATLAB 的工作空间。如果赋值表达式后面没有分号, 则将在 MATLAB 命令窗口中显示表达式的运算结果。若不想显示运算结果, 则应在赋值语句末尾加一个分号。如果省略了赋值变量和等号, 则表示式运算的结果将赋给保留变量 **ans**。所以说, 保留变量 **ans** 将永远存放最近一次无赋值变量语句的运算结果。

(2) 函数调用语句

其基本结构为:

[返回变量列表]=函数名(输入变量列表)

其中, 函数名的要求和变量名的要求是一致的, 一般函数名应该对应在 MATLAB 路径下的一个文件, 例如, 函数名 **myfun** 一般对应于 **myfun.m** 文件。当然, 还有一些函数名需对应于 MATLAB 的内置 (built-in) 函数。

返回变量列表和输入变量列表均可以由若干个变量名组成, 它们之间应该用逗号, 返回值变量还允许用空格分隔, 如 **[V, D]=eigs(A, 1)**, 该函数求给定矩阵 **A** 的模最大的特征值 **D** 及对应的特征向量 **V**, 所得的结果由 **V** 和 **D** 两个变量返回。如果不想显示函数调用的最终结果, 在函数调用语句后仍应该加个分号, 如 **[V, D]=eigs(A, 1);**。

A.1.3 矩阵的 MATLAB 表示

1. 矩阵的输入

复数矩阵为 MATLAB 的基本变量单元。例如, 矩阵 $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ 可以由下面的语

句直接输入到 MATLAB 的工作空间中:

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

在该语句中, 空格和逗号都可以用来分割同一行的元素, 而分号 (或换行) 用来表示换行。给出了上面的命令, 就可以在 MATLAB 的工作空间中建立一个 **A** 变量了。同时, 该语句将得出如下显示的结果:

```
A=
```

```

1   2   3
4   5   6
7   8   9

```

在 MATLAB 编程中有一个约定：如果在一个赋值语句后面没有分号，则等号右边的变量将在 MATLAB 命令窗口中显示出来。如果不想显示中间结果，则应该在语句末尾加一个分号，如：

```
>>A=[1, 2, 3; 4, 5, 6; 7 8 9]; %不显示结果，但进行赋值
```

这里的%表示后面的语句为注释语句。

在 MATLAB 中也可以容易地输入向量和标量。

学会了矩阵的基本表示方法之后，就可以容易地理解下面矩阵（由三个子矩阵构成）赋值表达式的方式和结果了。

```
>>A=[A; [1, 3, 5]],[1; 2; 3; 4]] %在矩阵下面先补一行，再补一列
```

这样，新的 A 矩阵就变成 4×4 矩阵。可见，利用 MATLAB 环境可以随意修改矩阵的维数。

MATLAB 语言定义了独特的冒号表达式来给行向量赋值，其基本格式

```
a=s1: s2: s3
```

其中，s1 为起始值，s2 为步长间距，s3 为终止值。如果 s2 的值为负值，则要求 s1 的值大于 s3 的值，否则结果为一个空向量 a。如果省略了 s2 的值，则步长间距取默认值 1。

可以通过下面的语句定义一个行向量：

```
>>b=0: 0.1: 1.05
```

该语句可以建立起向量 $b=[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]$ 。

复数矩阵的输入同样也是很简单的，在 MATLAB 环境中定义了两个记号 i 和 j，可以用来直接输入复数矩阵。例如，如果想在 MATLAB 中输入如下复数矩阵

$$C = \begin{bmatrix} 1+9i & 2+8i & 3+7i \\ 4+6i & 5+5i & 6+4i \\ 7+3i & 8+2i & i \end{bmatrix},$$

则可以通过下面的 MATLAB 语句直接进行赋值：

```
>>c=[1+9i, 2+8i, 3+7i; 4+6i, 5+5i, 6+4i; 7+3i, 8+2i, i]
```

2.数据类型的转换

可以用 $B=\text{sym}(A)$ 将已知的双精度变量 A 转换成符号型变量。从符号型到双精度型的变量转换可以用 $C=\text{double}(B)$ 实现。另外，还有其他类型转换函数，如 num2str, int2str 等。

A.1.4 多维数组的定义

除了标准的二维矩阵外，MATLAB 还定义了三维或多维数组。假设有若干个维数相同的矩阵 A_1, A_2, \dots, A_m ，那么把这若干个矩阵一页一页地叠起来，就可以构成一个 m 维数组。三维数组在 RGB 式彩色图像描述中十分有用，因为，这样的三维数组可以将图像的红色、绿色和蓝色分量分别用像素矩阵表示，然后把这 3 个矩阵整合成一个三维数组。

假设可以定义如下 A_1 、 A_2 、 A_3 矩阵

```
>>A1=[1, 2, 3; 4, 5, 6; 7, 8, 9]; A2=A1'; A3=A1-A2;
```

则通过下面的方法就可以定义出一个三维数组 A4：

```
>>A4(:, :, 1)=A1; A4(:, :, 2)=A2; A4(:, :, 3)=A3
```

这样可以得出如下的三维数组表示：

$$A_4(:, :, 1) = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad A_4(:, :, 2) = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}, \quad A_4(:, :, 3) = \begin{bmatrix} 0 & -2 & -4 \\ 2 & 0 & -2 \\ 4 & 2 & 0 \end{bmatrix}.$$

MATLAB 提供了一个 cat 函数来构造多维数组，该函数的调用格式为：

```
A=cat(n, A1, A2, ..., Am)
```

其中，n=1 和 2 时分别构造 $[A1; A2; \dots; Am]$ 和 $[A1, A2, \dots, Am]$ ，结果是二维数组，而 n=3

可以构造出三维数组。例如，例如上面的命令可以由下面的简单函数调用语句取代，

>>A5=cat(3, A1, A2, A3)

这样得出的赋值效果和 A4 完全一致。

A.2 MATLAB 的矩阵运算

矩阵是 MATLAB 数据存储的基本单元，而矩阵的运算是 MATLAB 语言的核心，在 MATLAB 语言系统中几乎一切运算均是以对矩阵的操作为基础的。矩阵的运算是按一定的运算规则进行的，其规则是由运算符决定的。

A.2.1 操作符与运算符

1.操作符

在编辑程序或命令中，当标点或其他符号表示特定的操作功能时就称其为操作符。表 A.1 列出了操作符。

表 A.1 操作符

操作符	使用说明
:	冒号。(1) m: n 产生一个数组[m, m+1, ..., n]; (2) m: k: n 产生一个数组[m, m+k, ..., N] (这里 $N \leq n$); (3) A(:, j)取矩阵的第 j 列; (4) A(k, :)取矩阵 A 的第 k 行; (5) A(:)把 A 矩阵展开成一个长的列向量。
;	分号。(1) 在矩阵定义中表示行与行之间的分隔符; (2) 在命令语句的结尾表示不显示这行语句的执行结果。
...	续行号。一个命令语句非常长，一行写不完，可以分几行写，此时在行的末尾加上续行号，表示是一个命令语句。
%	百分号。在编程时引导注释行，而系统解释执行程序时，%后面的内容不作处理。

2.运算符

算法运算符是构成运算的最基本的操作命令，可以在 MATLAB 的命令窗口中直接运行。运算符可分为三类：算术运算符、关系运算符与逻辑运算符。不同的运算符及功能说明见表 A.2，表 A.3，表 A.4。

表 A.2 算术运算符

运算符	算术运算符
+	加法运算。两个数相加或两个同型矩阵相加。如果是一个矩阵和一个数相加，则这个数自动扩展为与矩阵同维数的一个矩阵。
-	减法运算。两个数相减或两个同型矩阵相减。
*	乘法运算。两个数相乘或两个可乘矩阵相乘。
/	除法运算。两个数 a/b 表示 $a \div b$ ，两个矩阵 A/B 表示 A 右乘 B 的广义逆矩阵。
\	除法运算。两个数 a\b 表示 $b \div a$ ，两个矩阵 A\B 表示 B 左乘 A 的广义逆矩阵。
^	乘幂运算。数的幂次或一个方阵的幂次。
.*	点乘运算。两个同型矩阵对应元素相乘。
./	点除运算。两个同型矩阵对应元素相除。
.\	点除运算。两个同型矩阵对应元素左除。
.^	点乘幂运算。A.^B 表示矩阵 A 中每个元素取 B 中对应元素的幂次。

表 A.3 关系运算符

运算符	功能说明	运算符	功能说明
>	判断大于关系	>=	判断大于等于关系
<	判断小于关系	<=	判断小于等于关系
==	判断等于关系	~=	判断不等于关系

表 A.4 逻辑运算符

运算符	功能说明	运算符	功能说明
&	与运算	~	非运算

	或运算	xor(a, b)	异或运算
--	-----	-----------	------

关系运算符主要用于比较数、字符串、矩阵之间的大小或不等关系，其返回值是 0 或 1。

逻辑运算符主要用于逻辑表达式和进行逻辑运算，参与运算的逻辑量以 0 代表“假”，以任意非 0 数代表“真”。逻辑表达式和逻辑函数的值以 0 表示“假”，以 1 表示“真”。

A.2.1 特殊矩阵和矩阵的操作

1.特殊矩阵

对于一些比较特殊的矩阵，由于其具有特殊的结构，MATLAB 提供了一些函数用于生成这些矩阵，见表 A.5。

表 A.5 生成特殊矩阵的命令函数

命令函数	功能说明
[]	生成空矩阵，当对一项操作无结果时，返回空矩阵，空矩阵的大小为零。
zeros(m, n)	生成一个 m 行、n 列的零矩阵。
ones(m, n)	生成一个 m 行、n 列的元素全为 1 的矩阵。
eye(m, n)	生成一个 m 行、n 列的单位矩阵（注意数学上单位矩阵都是方阵）。
rand(m, n)	生成一个 m 行、n 列的[0,1]区间上均匀分布的随机数矩阵。
randn(m, n)	生成一个 m 行、n 列的标准正态分布的随机数矩阵。

2.矩阵中元素或块的操作

对矩阵中元素或块的常用操作见表 A.6。

表 A.6 矩阵中元素或块的常用操作

表达式或命令函数	功能说明
A(:)	依次提出矩阵 A 的每一列，将 A 拉伸为一个列向量。
A([i1:i2], [j1:j2])	提出矩阵 A 的第 i1~i2 行，第 j1~j2 列，构成新矩阵。
A([i1,i2,i3,i4], :)	提出矩阵 A 的指定的第 i1、i2、i3、i4 行，构成新矩阵。
A(:, [j1,j2,j3,j4])	提出矩阵 A 的指定的第 j1、j2、j3、j4 列，构成新矩阵。
A([i2:-1:i1], :)	以逆序提出矩阵 A 的第 i1~i2 行，构成新矩阵。
A(:, [j2:-1:j1])	以逆序提出矩阵 A 的第 j1~j2 列，构成新矩阵。
A([i1:i2], :)=[]	删除 A 的第 i1~i2 行，构成新矩阵。
A(:, [j1:j2])=[]	删除 A 的第 j1~j2 列，构成新矩阵。
[A B]或[A; B]	将矩阵 A 和 B 拼接成新矩阵。
diag(A, k)	抽取矩阵 A 的第 k 条对角线元素向量（主对角线编号为 0，上方的编号依次加 1，下方的编号依次减 1）； 若 A 为向量，生成一个以 A 为第 k 对角线元素的方阵。
tril(A, k)	抽取矩阵 A 的第 k 条对角线下方的部分。
triu(A, K)	抽取矩阵 A 的第 k 条对角线上方的部分。
flipud(A)	矩阵 A 进行上下翻转。
fliplr(A)	矩阵 A 进行左右翻转。
A'	矩阵 A 的转置。
A'	矩阵 A 的共轭转置。
rot90(A)	矩阵 A 逆时针旋转 90°。

3.矩阵的基本函数运算

矩阵的函数运算是矩阵运算中最实用的部分，常用的主要有以下几个，见表 A.7。

表 A.7 矩阵的函数运算命令

命令	功能	命令	功能
det(A)	求矩阵 A 的行列式	rref(A)	求矩阵 A 的行最简形
inv(A)	求方阵 A 的逆阵	rank(A)	求矩阵 A 的秩

size(A)	求矩阵 A 的行数和列数	trace(A)	求矩阵 A 的迹
eig(A)	求 A 的特征值及特征向量	[Q,R]=qr(A)	求正交矩阵 Q 和上三角阵 R 满足 A=QR
orth(A)	将非奇异矩阵 A 正交规范化		

4.矩阵的数据处理

MATLAB 具有强大的数据处理功能，比如数据的排序、求最大值、求和、求均值等。常用的数据处理命令见表 A.8。

表 A.8 常用数据处理的命令

命令	功能	命令	功能
max(A)	求向量或矩阵列的最大值	min(A)	求向量或矩阵列的最小值
mean(A)	求向量或矩阵列的平均值	median(A)	求向量或矩阵列的中间值
sum(A)	求向量或矩阵列的元素和	prod(A)	求向量或矩阵列的元素乘积
var(A)	求向量或矩阵列的方差	std(A)	求向量或矩阵列的标准差
cov(A)	矩阵列向量之间的协方差矩阵	corrcoef(A)	矩阵列向量之间的相关系数矩阵
length(A)	求向量所含元素个数	find(A)	求向量或矩阵中非零元素的地址

A.3 M 文件与编程

A.3.1 M 文件

M 文件是由 MATLAB 语句（命令或函数）构成的 ASCII 码文本文件，文件名必须以“.m”为扩展名。M 文件通过 M 文件编辑/调试器生成。在命令窗口调用 M 文件，可实现一次执行多条 MATLAB 语句的功能。M 文件有命令文件和函数文件两种形式。

1.命令文件

命令文件是 MATLAB 命令或函数的组合，没有输入输出参数，执行命令文件只需在命令窗口中键入文件名按回车或在 M 文件编辑/调试器窗口激活状态下按“F5”键。

当用户要运行的指令较多时，可以直接从键盘上逐行输入指令，但这样做显得很麻烦，而命令文件则可以较好地解决这一问题。用于可以将一组相关命令编辑在同一个 ASCII 码文件中，运行时只需输入文件名，MATLAB 就会自动按顺序执行文件中的命令。这类似于批处理文件。命令文件中的语句可以访问 MATLAB 工作空间（Workspace）中的所有数据。在运行的过程中所产生的变量均是全局变量。这些变量一旦生成，就一直保存在工作空间中，除非用户将它们清除（使用 clear 命令）。

2.函数文件

函数文件是另一种形式的 M 文件，可以有输入参数和返回输出参数，函数在自己的工作空间中操作局部变量，它的第一句可执行语句是以 function 引导的定义语句。在函数文件中的变量除非用 global 声明为全局变量外，都是局部变量，它们在函数执行过程中驻留在内存中，在函数执行结束时自动消失。函数文件不单单具有命令文件的功能，更重要的是它提供了与其他 MATLAB 函数和程序的接口，因此功能更加强大。

MATLAB 函数文件的格式为：

function [返回参数 1, 返回参数 2, ...]=函数名（输入参数 1, 输入参数 2, ...）

函数体

注 A.1 function 是区分命令文件与函数文件的重要标志；函数体包含所有函数程序代码，是函数的主体部分；函数文件保存的文件名应与用户定义的函数名一致。在命令窗口中以文件名调用函数。

例 A.1 定义函数 $f(x, y) = xy$ ，并计算 $f(2, 6)$ 。

在编辑器中写出如下程序：

```
function f=myfunA1(x,y);
```

```
f=x.*y;
```

保存为 myfunA1.m（这是文件名，与函数名一致），然后在命令窗口中执行

```
>>myfunA1(2,6)
```

就可以求得 $f(2,3)=12$ 。

3. 匿名函数

函数文件一般存储在一个单独的文件中，调用语句放在其他命令文件中，使用起来有时不太方便。匿名函数不需要存储在一个文件中，匿名函数可以接受输入并返回输出，就像标准函数一样。但是，匿名函数可能只包含一个可执行语句，可以和调用它的语句写在同一个命令文件中。

例 A.2（续例 A.1）定义匿名函数 $f(x,y)=xy$ ，并计算 $f(2,6)$ 。

编写如下的 MATLAB 程序：

```
funA1=@(x,y)x.*y;  
z=funA1(2,6)
```

保存在文件 gA2.m 中，在命令窗口运行 gA2 就可以求得 $f(2,6)=12$ 。

A.3.2 流程控制结构

作为一种程序设计语言，MATLAB 提供了循环语句结构、条件控制语句结构和开关语句结构等。

1. 循环语句结构

循环语句有两种结构：for...end 结构和 while...end 结构。

（1）for...end 结构，其调用格式如下：

```
for 循环变量=初值:步长:终值  
    循环体  
end
```

其执行过程为：将初值赋给循环变量，执行循环体；执行完一次循环之后，循环变量自增一个步长的值，然后再判断循环变量的值是否介于初值和终值之间，如果满足，仍然执行循环体，直至不满足为止。

（2）while...end 结构，其调用格式如下：

```
while 表达式  
    循环体  
end
```

其执行过程为：若表达式的值为真，则执行循环体语句，执行后再判断表达式的值是否为真，直到表达式的值为假时跳出循环。

while 语句一般用于事先不能确定循环次数的情况。

2. 条件控制语句结构

（1）if...end 结构，其调用格式如下：

```
if 条件式  
    语句体  
end
```

其执行过程为：当条件式为真时，执行语句体，否则不执行。

（2）if...else...end 结构，其调用格式如下：

```
if 条件式  
    语句体 1  
else  
    语句体 2
```

```
end
```

其执行过程为：当表达式的值为真时，执行语句体 1，否则执行语句体 2。

（3）if...elseif...else...end 结构，其调用格式为：

```
if 条件式 1, 语句体 1  
elseif 条件式 2, 语句体 2  
elseif 条件式 3, 语句体 3  
.....  
elseif 条件式 n, 语句体 n  
else, 语句体 n+1
```

end

其执行过程为：当条件式 i ($1 \leq i \leq n$) 为真时，执行对应的语句体 i ，否则执行语句体 $n+1$ 。

例 A.3 用 MATLAB 生成如下的矩阵

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \end{bmatrix}.$$

编写的如下的 MATLAB 程序：

```
clc, clear
ncols=6; nrows=4; A(ncols,ncols)=1; %初始化，赋什么值都可以
for c=1:ncols
    for r=1:nrows
        if r==c, A(r,c)=2;
        elseif abs(r-c)==1, A(r,c)=-1;
        else, A(r,c)=0;
        end
    end
end
A %显示生成的 A 矩阵
```

3. 开关语句结构

MATLAB 提供的 switch 开关语句结构，其调用格式如下：

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

其执行过程为：开关表达式的值与每一个 case 后面表达式的值比较，若与第 i ($1 \leq i \leq n$) 个 case 后面的表达式 i 的值相等，就执行语句体 i ；若都不相同，则执行 otherwise 后面的语句体 $n+1$ 。

例 A.4 输入成绩 n ； $n < 60$ 时，输出不及格； $60 \leq n < 70$ 时，输出一般， $70 \leq n < 80$ 时，输出良好； $n \geq 80$ 时，输出优秀。

```
clc, clear
n=input('请输入成绩: \n');
k=floor(n/10);
switch k
case {8,9,10}, disp('优秀')
case 7, disp('良好')
case 6, disp('一般')
otherwise, disp('不及格')
end
```

4. 其他流程控制语句

其他流程控制语句包括 continue 语句、break 语句和 return 语句。

(1) continue 语句用于 for 循环和 while 循环中，其作用就是终止一次循环的执行，跳过循环体中所有剩余的未被执行的语句，去执行下一次循环。

(2) break 语句也常用于 for 循环和 while 循环中，其作用就是终止当前循环的执行，跳出循环体，去执行循环体外的下一行语句。

(3) **return** 语句用于终止当前的命令序列，并返回到调用的函数或键盘。

A.4 MATLAB 绘图

MATLAB 除了强大的数值分析功能外，其受到工程技术人员广泛接受与使用的另一个重要原因是它提供了方便的绘图功能。用户只需指定绘图方式，并提供充足的绘图数据，即可以得出所需的图形。MATLAB 还对绘出的图形提供了各种修饰方法，使绘出的图形更美观。

MATLAB 提出了句柄图形学（**handle graphics**）的概念，为面向对象的图形处理提供了十分有用的工具。在图形绘制时，其中每个图形元素（如其坐标轴或图形上的曲线、文字等）都是一个独立的对象。用户可以对其中任何一个图形元素进行单独修改，而不影响图形的其他部分，具有这样特点的绘图称为向量化的绘图。这种向量化的绘图要求给每个图形元素分配一个句柄（**handle**），以后再对该图形元素做进一步操作时，则只需对该句柄进行操作即可。

A.4.1 二维绘图函数

1. 绘图基本流程

MATLAB 提供了丰富的二维和三维绘图函数和绘图工具，绘制图形一般需要经过 6 个步骤。

(1) 准备数据

绘制二维曲线，先需要准备横坐标（自变量）数据，再通过函数关系定义纵坐标（函数值）数据。

(2) 创建图形窗口，指定绘图位置

MATLAB 默认的图形窗口为 **Figure1**，可以使用函数 **figure** 创建图形窗口，作为当前绘图窗口。需要在同一图形窗口绘制多幅图形时，可以使用函数 **subplot** 选择子图位置，还可以调用函数 **set** 设置窗口属性。

(3) 绘制图形

调用绘图函数绘制曲线，并设置曲线的样式和标记属性，如线型、颜色、数据点标记符号等。

(4) 设置坐标轴和图形注释

设置坐标轴包括坐标轴的范围、刻度、网格线和坐标轴边框等，图形注释包括图形标题、坐标轴名称、图例、文字说明等。

(5) 修饰三维图形

设置三维图形的着色、光照效果、材质、视角和三度（横、纵、高）比例等。

(6) 保存或导出图形。

将绘制的图形保存为 **.fig** 文件或其他类型的图形文件。

2. 基本绘图函数

plot 函数是绘制二维图形的最基本函数，它是针对向量或矩阵的列来绘制曲线的，可以绘制线段和曲线。函数 **plot** 的最典型调用方式是三元组形式：

plot(x, y, 'color-style-marker')

其中 **x**, **y** 为同维数的向量（或矩阵），**x** 作为点的横坐标，**y** 作为点的纵坐标，**plot** 命令用直线连接相邻两数据点绘制图形。**color**、**style** 和 **marker** 分别是颜色、线型和数据点标记，它们之间没有先后顺序之分。

颜色、线型和数据点符号如表 A.9 所示。

表 A.9 颜色、线型、数据点符号

颜色符号	颜色	线型符号	线型	数据点符号	标记
b （默认）	蓝色	-（默认）	实线	+	十字
r	红色	:	短虚线	*	星号
y	黄色	--	长虚线	o	圆圈
g	绿色	-.	点划线	x	叉号
c	蓝绿色			s	正方形
m	紫红色			d	菱形
k	黑色			p	五角星

w	白色			h	六角形
---	----	--	--	---	-----

画二维曲线图时，当知道曲线的函数表达式时，可以使用 3 种方式画图：（1）用描点画图命令 `plot`，（2）用函数画图命令 `fplot`，（3）用“Easy-to-use”函数画图命令 `ezplot`，该命令既可以执行符号函数画图，也可以执行匿名函数画图。

3.图形窗口的设置

图形窗口是以图形图像的方式显示绘图函数的运行结果。

（1）创建新窗口

图形窗口（Figure Window）是 MATLAB 绘制的所有图形的输出专用窗口。当 MATLAB 没有打开图形窗口而执行了一条绘图命令时，系统会自动创建一个图形窗口，以后再使用绘图命令时，将刷新当前图形窗口，使用 `figure` 函数可以创建多个图形窗口。调用格式为：

`figure` 或 `figure(n)`

其中 `n` 为窗口编号。

（2）绘制叠加图形

有时希望将后续图形和前面的图形叠加进行比较，MATLAB 中采用函数 `hold` 启动（或关闭）图形保持功能，将新产生的图形叠加到已有的图形上。常用的 3 种调用格式如下：

`hold on`：启动图形保持功能，允许在当前图形上绘制其他图形。

`hold off`：关闭图形保持功能，之后再执行绘图命令，自动清除当前窗口中原有的内容，然后绘制新图形。

`hold`：在以上两个命令之间切换。

（3）绘制子图

MATLAB 中使用函数 `subplot` 可以将同一图形窗口分成多个不同子窗口，调用格式为：

`subplot(m,n,k)`：将当前图形窗口分割成 `m×n` 个子窗口，并选择第 `k` 个子窗口作为当前绘图窗口。子窗口的序号按行优先编号，左上方为第一幅，先从左至右再从上至下排列，子窗口彼此独立，有自己的坐标轴。

4.图形修饰

（1）设置坐标轴和网格线

绘制图形时，系统自动给出图形的坐标轴，利用函数 `axis` 可以设置坐标轴的刻度和范围。函数，利用函数 `grid` 可以设置网格线。调用格式和功能如表 A.10 所示。

表 A.10 设置坐标轴和网格线

类型	调用格式	功能说明
坐标轴显示方式	<code>axis('auto')</code> 或 <code>axis auto</code>	将坐标轴设置返回默认状态
	<code>axis('square')</code> 或 <code>axis square</code>	将坐标轴设置为正方形（系统默认为矩形）
	<code>axis('equal')</code> 或 <code>axis equal</code>	将两个坐标轴刻度设定为相等
	<code>axis('off')</code> 或 <code>axis off</code>	不显示坐标轴
	<code>axis('on')</code> 或 <code>axis on</code>	显示坐标轴的所有设置
坐标轴范围	<code>axis([xmin xmax ymin ymax])</code>	设定坐标轴的范围为 $xmin \leq x \leq xmax$ ， $ymin \leq y \leq ymax$
网格线	<code>grid on</code>	显示网格线
	<code>grid off</code>	不显示网格线
	<code>grid minor</code>	设置网格线间的间距

（2）命令方式添加标注

图形绘制完后，MATLAB 提供了一些特殊的图形函数，用于修饰绘制好的图形，如图形标题、坐标轴标记、图例和文字注释等。调用格式和功能如表 A.11 所示。

表 A.11 图形标注函数

标注类型	调用格式	功能说明
标题标	<code>title('s')</code>	在当前坐标轴的顶部加上字符串 <code>s</code> ，作为该图

注		形的标题。
	<code>title('s', Name, Value)</code>	参数 Name 、 Value 定义标注文本的属性和属性值，包括字体大小、字体、字体粗细等。
坐标轴标注	<code>xlabel('s')</code>	用字符串 <i>s</i> 标记 <i>x</i> 轴。
	<code>xlabel('s', Name, Value)</code>	参数 Name 、 Value 定义标注文本的属性和属性值。
	<code>ylabel('s')</code>	用字符串 <i>s</i> 标记 <i>y</i> 轴。
	<code>ylabel('s', Name, Value)</code>	参数 Name 、 Value 定义标注文本的属性和属性值。
图例标注	<code>legend('s1', 's2', ..., 'sn')</code>	用指定的字符串 <i>s1</i> , <i>s2</i> , ..., <i>sn</i> 在当前图形中添加图例。
	<code>legend(labels, 'Location', Value)</code>	在指定位置处添加图例 <i>labels</i> 。
	<code>legend(labels, Name, Value)</code>	参数 Name 、 Value 定义标注文本的属性和属性值。
文本标注	<code>text(x, y, 's')</code>	在二维图形指定位置(<i>x</i> , <i>y</i>)处添加文本注释。
	<code>text(x, y, z, 's')</code>	在三维图形指定位置(<i>x</i> , <i>y</i> , <i>z</i>)处添加文本注释。
	<code>text(..., Name, Value)</code>	参数 Name 、 Value 定义标注文本的属性和属性值。
	<code>gtext('s1', ..., 'sn')</code>	利用鼠标确定位置添加由 <i>s1</i> , ..., <i>sn</i> 组成的一组文本注释。
线条、箭头、图框标注	<code>annotation('line', x, y)</code>	添加从点(<i>x</i> (1), <i>y</i> (1))到(<i>x</i> (2), <i>y</i> (2))的线条， <i>x</i> , <i>y</i> 表示整个图形的比例，取值为[0, 1]。
	<code>annotation('arrow', x, y)</code>	添加从点(<i>x</i> (1), <i>y</i> (1))到(<i>x</i> (2), <i>y</i> (2))的箭头。
	<code>annotation('doublearrow', x, y)</code>	添加从点(<i>x</i> (1), <i>y</i> (1))到(<i>x</i> (2), <i>y</i> (2))的双箭头。
	<code>annotation('textarrow', x, y)</code>	添加从点(<i>x</i> (1), <i>y</i> (1))到(<i>x</i> (2), <i>y</i> (2))的带文本框的箭头。
	<code>annotation('textbox', [x, y, w, h])</code>	添加左下角坐标为(<i>x</i> , <i>y</i>)，宽为 <i>w</i> ，高为 <i>h</i> 的文本框， <i>x</i> , <i>y</i> , <i>w</i> , <i>h</i> 表示整个图形的比例，取值为[0, 1]。
	<code>annotation('ellipse', [x, y, w, h])</code>	添加左下角坐标为(<i>x</i> , <i>y</i>)，宽为 <i>w</i> ，高为 <i>h</i> 的椭圆。
	<code>annotation('rectangle', [x, y, w, h])</code>	添加左下角坐标为(<i>x</i> , <i>y</i>)，宽为 <i>w</i> ，高为 <i>h</i> 的矩形框。
	<code>annotation(..., Name, Value)</code>	参数 Name 、 Value 定义标注的属性和属性值。

例 A.5 在同一图形窗口绘制 $y = \sin x$ 和 $y = \cos x$ 的图形，用 16 号的黑体添加标题和图例。

编写 MATLAB 程序如下：

```
clc, clear, close all
x=-2*pi:0.1:2*pi; y1=sin(x); y2=cos(x);
plot(x,y1,'b-*'), hold on, plot(x,y2,'g--o')
title('y=sinx 和 y=cosx','Fontname','黑体','FontSize',12)
legend({'sinx','cosx'},'Location','NorthEast')
所绘制的图形见图 A.1。
```

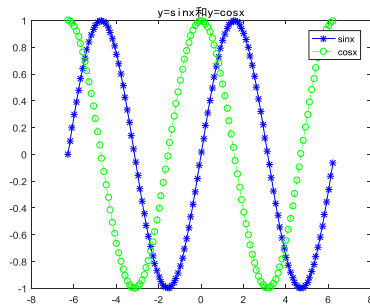


图 A.1 $y = \sin x$ 和 $y = \cos x$ 的图形

例 A.6 在同一图形窗口绘制 $y = \sin x$ 和 $y = \cos x$ 的图形，用箭头对图形进行标注。

编写 MATLAB 程序如下：

```
clc, clear, close all
fplot(@(x)sin(x),[-2*pi,2*pi],'b-*'), hold on
fplot(@(x)cos(x),[-2*pi,2*pi],'k--o')
annotation('textarrow',[0.25,0.33],[0.49,0.49],'String','y=sinx')
annotation('textarrow',[0.21,0.24],[0.40,0.40],'String','y=cosx')
```

所绘制的图形见图 A.2。

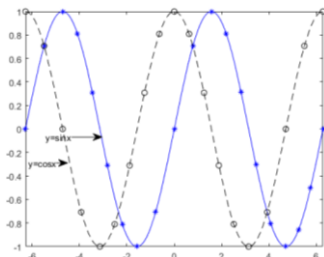


图 A.2 $y = \sin x$ 和 $y = \cos x$ 的箭头标注图形

(3) 特殊符号的标注

图形文本标注中可以使用希腊字符、数学符号或者上标和下标字体等特殊字符，常用的特殊符号如表 A.12 所示。

表 A.12 特殊符号表

函数字符	代表字符	函数字符	代表字符
<code>\alpha</code>	α	<code>\omega</code>	ω
<code>\beta</code>	β	<code>\Gamma</code>	Γ
<code>\gamma</code>	γ	<code>\Delta</code>	Δ
<code>\delta</code>	δ	<code>\Pi</code>	Π
<code>\epsilon</code>	ϵ	<code>\Sigma</code>	Σ
<code>\zeta</code>	ζ	<code>\Omega</code>	Ω
<code>\eta</code>	η	<code>\infty</code>	∞
<code>\theta</code>	θ	<code>\partial</code>	∂
<code>\lambda</code>	λ	<code>\geq</code>	\geq
<code>\mu</code>	μ	<code>\leq</code>	\leq
<code>\nu</code>	ν	<code>\leftrightarrow</code>	\leftrightarrow
<code>\xi</code>	ξ	<code>\leftarrow</code>	\leftarrow
<code>\pi</code>	π	<code>\rightarrow</code>	\rightarrow
<code>\rho</code>	ρ	<code>\uparrow</code>	\uparrow
<code>\sigma</code>	σ	<code>\downarrow</code>	\downarrow
<code>\tau</code>	τ		

(4) 双 y 轴标注

MATLAB 提供的函数 `yyaxis` 可以提供双 y 轴标注。

例 A.7 在同一个图形窗口，分别绘制 $y = \sin(3x)$ 和 $y = e^{0.5x} \sin(3x)$ 的图形，并进行双 y 轴标注。

编写的 MATLAB 程序如下：

```
clc, clear, close all
x=linspace(0,10); %0 到 10 之间等间距产生 100 个数
y=sin(3*x); yyaxis left, plot(x,y)
z=sin(3*x).*exp(0.5*x);
yyaxis right, plot(x,z,'k'), ylim([-150 150])
所绘制的图形见图 A.3。
```

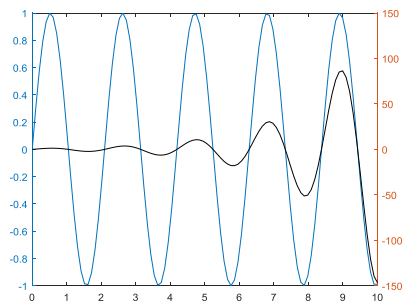


图 A.3 双 y 标注图形

5. 获取图形数据

绘制好图形后，有时需要获取某点的数据，MATLAB 提供了 `ginput` 函数，能够方便地通过鼠标获取二维平面图中任一点的坐标值。调用格式和功能如表 A.13 所示。

表 A.13 获取图形数据点函数 `ginput`

调用格式	功能说明
<code>[x,y]=ginput(n)</code>	通过鼠标选取 n 个点，并将其坐标值保存在 <code>[x,y]</code> 中，按 “Enter” 键结束选取
<code>[x,y]=ginput</code>	通过鼠标选择点，取点数不受限制，并将其坐标值保存在 <code>[x,y]</code> 中，按 “Enter” 键结束取点。

6. 图形文件的输出

图形窗口中绘制的图形可以保存为多种类型的图形文件。在图形窗口中选择窗口菜单栏中的 “file” 菜单，再从下拉菜单中选择 “Save as” 子菜单。在弹出的对话框中选择文件保存的目录、文件名和文件类型格式，单击 “保存” 按钮即可。

A.4.2 三维绘图命令

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

`plot3(x,y,z)` 通过描点连线画出曲线，这里 x,y,z 都是 n 维向量，分别表示该曲线上点集的横坐标、纵坐标、竖坐标。

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

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

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

例 A.8 画出三维曲线 $x = t \cos t$ ， $y = t \sin t$ ， $z = t$ 的图形。

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

例 A.9 二次曲面

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

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

$x=@(t,z)2*z.*\cos(t); y=@(t,z)\sqrt{2}*z.*\sin(t);$
 $z=@(t,z)z; \text{ezsurf}(x,y,z,[0,2*\pi,-6,6])$

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

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

使用 ezsurf 画图的 MATLAB 程序如下:

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

使用 surf 画图的 MATLAB 程序如下:

$t=0:0.1:2*\pi; s=-\pi/2:0.1:\pi/2;$
 $[t,s]=\text{meshgrid}(t,s); \% \text{生成 2 维网格数据}$
 $x=2*\cos(t).*\cos(s); y=\sqrt{3}*\sin(t).*\cos(s);$
 $z=\sqrt{2}*\sin(s); \text{surf}(x,y,z)$

(3) (显函数) 双曲抛物面或马鞍面 $z = xy$ 。

$\text{ezmesh}(@ (x,y)x.*y,[-6,6,-5,5])$

标注字符使用 LaTeX 格式时, 程序如下:

$\text{ezmesh}(@ (x,y)x.*y,[-6,6,-5,5])$
 $\text{xlabel}('$x$', 'Interpreter', 'Latex')$
 $\text{ylabel}('$y$', 'Interpreter', 'Latex')$
 $\text{zlabel}('$z$', 'Interpreter', 'Latex', 'Rotation', 0)$
 $\text{title}('$xy$', 'Interpreter', 'Latex')$

例 A.10 莫比乌斯带是一种拓扑学结构, 它只有一个面和一个边界, 是 1858 年由德国数学家、天文学家莫比乌斯和约翰·李斯丁独立发现的。其参数方程为

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

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

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

例 A.11 已知平面区域 $0 \leq x \leq 1400$, $0 \leq y \leq 1200$ 步长间隔为 100 的网格节点高程数据见表 A.14(单位: m)。

表 A.14 高程数据表

y/x	0	100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
1200	1350	1370	1390	1400	1410	960	940	880	800	690	570	430	290	210	150
1100	1370	1390	1410	1430	1440	1140	1110	1050	950	820	690	540	380	300	210
1000	1380	1410	1430	1450	1470	1320	1280	1200	1080	940	780	620	460	370	350
900	1420	1430	1450	1480	1500	1550	1510	1430	1300	1200	980	850	750	550	500
800	1430	1450	1460	1500	1550	1600	1550	1600	1600	1550	1500	1500	1500	1550	1550
700	950	1190	1370	1500	1200	1100	1550	1600	1550	1380	1070	900	1050	1150	1200

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

(1) 画出该区域的等高线.

(2) 画出该区域的三维表面图.

画等高线及三维表面图的 MATLAB 程序如下:

```
clc, clear
```

```
a=load('data14.txt'); %把表 A.14 中的高程数据保存在纯文本文件 data14.txt 中
x0=0:100:1400; y0=1200:-100:0;
```

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

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

所画出的图形见图 A.4.

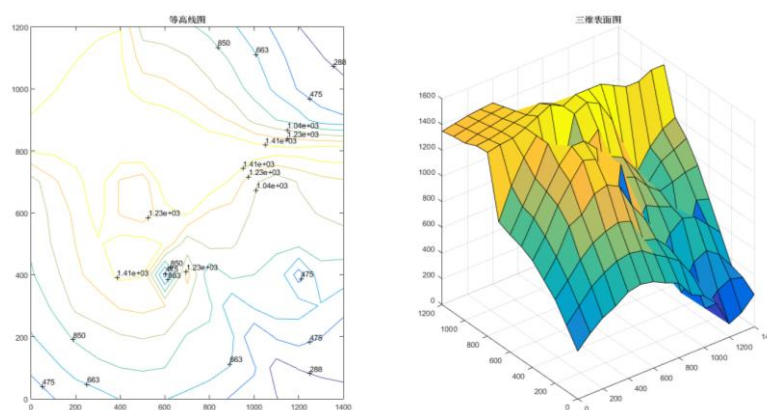


图 A.4 等高线图及三维表面图

例 A.12 已知

$$C(N, M) = h \left[\frac{\rho}{1-\rho} + \frac{\frac{\lambda}{\theta} \left[\frac{\lambda}{\theta} \left(1 - \left(\frac{\lambda}{\theta + \lambda} \right)^N \right) - N \left(\frac{\lambda}{\theta + \lambda} \right)^N \right] \left(1 - \left(\frac{\theta}{\theta + \lambda} \right)^M \right)}{\left(\frac{\theta}{\theta + \lambda} \right)^M \frac{\lambda}{\lambda + \theta} + \left(1 - \left(\frac{\theta}{\theta + \lambda} \right)^M \right) \frac{\lambda}{\theta} \left[1 - \left(\frac{\lambda}{\theta + \lambda} \right)^N \right]} \right] + \frac{c_0 \lambda (1 - \rho)}{\left(\frac{\theta}{\theta + \lambda} \right)^M + \left(1 - \left(\frac{\theta}{\theta + \lambda} \right)^M \right) \frac{\lambda + \theta}{\lambda} \left[\frac{\lambda}{\theta} \left(1 - \left(\frac{\lambda}{\theta + \lambda} \right)^{N-1} \right) + \left(\frac{\lambda}{\theta + \lambda} \right)^N \right]},$$

其中, $\lambda=0.4$, $\mu=2.0$, $\theta=0.1$, $h=40$, $c_0=800$, $N=1,2,\dots,11$, $M=1,2,\dots,22$, 求 $C(N, M)$ 的最小值, 并画出 $C(N, M)$ 的图形.

计算及画图的 MATLAB 程序如下:

```
clc, clear, close all
```

```
h=40; L=0.4; m=2; t=0.1; r=L/m; c0=800;
```

```
zN=@(N)(L/(t+L)).^N; zM=@(M)(t/(t+L)).^M;
```

```
c=@(N,M)h*(r/(1-r)+L/t*(L/t*(1-zN(N))-N.*zN(N)).*(1-zM(M))./...
```

```
(zM(M)*L/(L+t)+(1-zM(M))*L/t.*(1-zN(N)))+...
```

```
c0*L*(1-r)./(zM(M)+(1-zM(M))*(L+t)/L.*(L/t*(1-zN(N-1))+zN(N)));
```

```
N=1:11; M=1:22; [N,M]=meshgrid(N,M);
```

```
s=c(N,M), ms=min(min(s)) %求最小值
```

```
ezmesh(c,[1,11,1,22]), title('')
```

```
xlabel('$N$', 'Interpreter', 'Latex'), ylabel('$M$', 'Interpreter', 'Latex')
```

`zlabel('$C(N,M)$','Interpreter','Latex')`

求得的最小值为 145.7182，所画的图形见图 A.6。

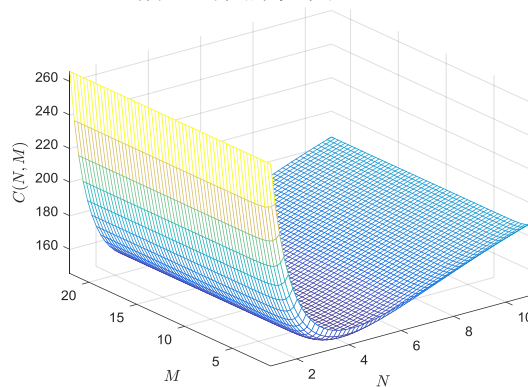


图 A.5 $C(N, M)$ 的图形

A.4.3 四维空间绘图命令

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

1. `cplxgrid` 构建一个极坐标的复数数据网格

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

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

2. `cplxmap` 对复变函数做图

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

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

3. `cplxroot` 画复数的 n 次根函数曲面

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

表 A.15 画复变函数 $f(z) = z^3$ 的图形。记

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

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

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

```
clc, clear, close all
z=cplxgrid(30); %生成网格数据
subplot(121), cplxmap(z,z.^3), colorbar %直接用工具箱的画图命令
title('工具箱画图效果')
%以下使用实函数 surf 命令画图，和上面做一下对比
r=linspace(0,1,50); t=linspace(0,2*pi,50);
[t,r]=meshgrid(t,r); %生成极坐标的网络数据
[x,y]=pol2cart(t,r); %极坐标转化为直角坐标
u=x.^3-3*x.*y.^2; v=3*x.^2.*y-y.^3; %计算复函数的实部和虚部
subplot(122), surf(x,y,u,v), colorbar %使用实函数画图
title('surf 命令画图效果')
所画的图形见图 A.6。
```

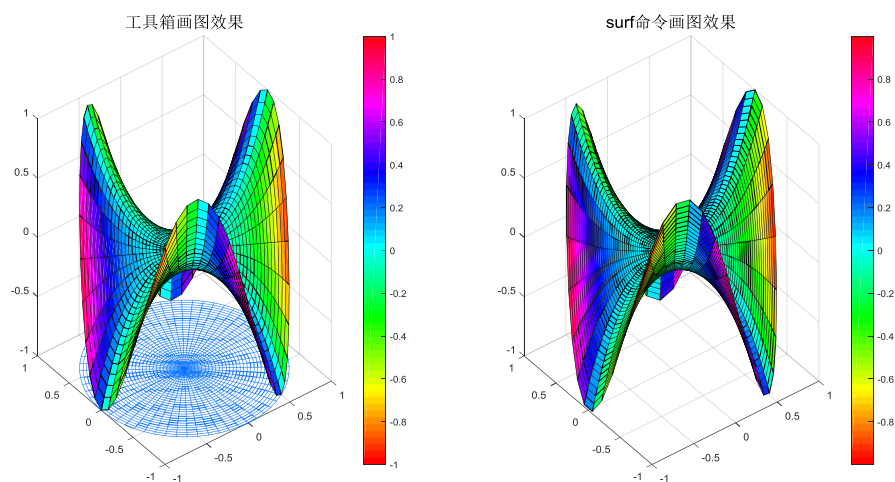



图 A.6 $f(z) = z^3$ 的图形

A.5 代数方程的求解

A.5.1 线性方程组的求解

1. 求线性方程组的一个解

MATLAB 中求解线性方程组 $Ax = b$ 的命令为 $x = \text{pinv}(A) * b$ ，无论数学上线性方程组 $Ax = b$ 有唯一解，多解或无解时，MATLAB 上述命令总是只给出一个解。

(1) $Ax = b$ 有唯一解时，MATLAB 也给出唯一解；

(2) $Ax = b$ 无解时，MATLAB 给出最小二乘解，所谓的最小二乘解是指 $\|Ax - b\|^2 = [Ax - b, Ax - b]$ 最小，其中 $[\cdot, \cdot]$ 表示向量的内积；

(3) $Ax = b$ 有无穷多解时，MATLAB 给出最小范数解。

注 A.2 当 A 列满秩时，可以使用命令 $x = A \backslash b$ 代替命令 $x = \text{pinv}(A) * b$ 。

例 A.13 求解下列非齐次线性方程组：

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

解 首先计算系数矩阵 A 的秩和增广矩阵 $[A, b]$ 的秩，两者不相等，线性方程组无解。就可以利用 MATLAB 求最小二乘解。

```
clc, clear
```

```
a=[4,2,-1;3,-1,2;11,3,0]; %系数矩阵
```

```
b=[2;10;8]; %常数项列
```

```
r1=rank(a) %计算系数矩阵的秩
```

```
r2=rank([a,b]) %计算增广矩阵的秩
```

```
x1=pinv(a)*b %求最小二乘解的数值解
```

```
x2=pinv(sym(a))*sym(b) %求最小二乘解的符号解
```

求得最小二乘解的数值解为 $x_1 = 1.2130$ ， $x_2 = -1.4478$ ， $x_3 = 1.9565$ 。

例 A.14 求解下列非齐次线性方程组：

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

解 系数矩阵 A 的秩 $R(A)$ 与增广矩阵的秩 $R([A, b])$ 都等于 2，线性方程组有无穷多解。就可以利用 MATLAB 求最小范数解。

```
clc, clear
```

```
a=[2 3 1;1,-2,4;3,8,-2;4,-1,9]; %系数矩阵
```

```
b=[4;-5;13;-6]; %常数项列
```

```

r1=rank(a) %求系数矩阵的秩
r2=rank([a,b]) %求增广矩阵的秩
x1=pinv(a)*b %求最小范数数值解
x2=pinv(sym(a))*sym(b) %求最小范数符号解
求得的最小范数符号解为  $x_1 = 1/3$ ,  $x_2 = 4/3$ ,  $x_3 = -2/3$ 。

```

例 A.15 解下列矩阵方程

$$X \begin{bmatrix} 2 & 1 & -1 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 3 \\ 4 & 3 & 2 \end{bmatrix}.$$

解 记矩阵方程为 $XA = B$, 因 $\det A = 3 \neq 0$, 故 A 可逆, 用 A^{-1} 右乘方程的两边得

$$X = BA^{-1} = \begin{bmatrix} -2 & 2 & 1 \\ -\frac{8}{3} & 5 & -\frac{2}{3} \end{bmatrix}.$$

用 MATLAB 计算的程序如下:

```

clc, clear
a=[2,1,-1;2,1,0;1,-1,1];
b=[1,-1,3;4,3,2];
c=det(a) %计算矩阵 a 的行列式
x1=b/a %求数值解, 或者使用 x=b*inv(a)
x2=sym(b)*inv(sym(a)) %求符号解
为了说明符号矩阵的定义, 我们使用下面将要介绍的 solve 命令求解上述线性方程
组, 当然实际上没有必要使用 solve 命令求解线性方程组的。

```

```

clc, clear
a=[2,1,-1;2,1,0;1,-1,1];
b=[1,-1,3;4,3,2];
x=sym('x%d%d',[2,3]) %定义 2×3 的符号矩阵
s=solve(x*a==b) %求符号方程组的解, 返回值 s 为结构数组
ss=[s.x11,s.x12,s.x13;s.x21,s.x22,s.x23] %显示方程组的解

```

2. 求线性方程组的通解

MATLAB 的函数 `rref` 可以求得矩阵的行最简形, 其调用格式为:

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

其中, 返回值 R 是矩阵 A 的行最简形, jb 是 A 的列向量组的最大无关组的标号。

利用增广矩阵的行最简形, 就可以写出线性方程组的通解。

例 A.16 (续例 A. 14) 求下列非齐次线性方程组

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

的通解。

利用如下的 MATLAB 程序:

```

clc, clear
a=[2 3 1;1,-2,4;3,8,-2;4,-1,9]; %系数矩阵
b=[4;-5;13;-6]; %常数项列
c=rref([a,b]) %求增广阵的行最简形
求得增广阵的行最简行为

```

$$\begin{bmatrix} 1 & 0 & 2 & -1 \\ 0 & 1 & -1 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

得到同解方程组：

$$\begin{cases} x + 2z = -1, \\ y - z = 1, \end{cases} \Rightarrow \begin{cases} x = -2z - 1, \\ y = z + 2, \end{cases}$$

即得通解为

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = c \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix} \quad (c \in R).$$

A.5.2 非线性方程的求解

1. 简单非线性方程的符号解

MATLAB 求符号方程的函数为 `solve`，其调用格式为

`s=solve(eqn,var)` %其中 `eqn` 为符号方程，`var` 指定符号变量，`s` 为求得的符号解。

`[y1,...,yN]=solve(eqns,vars,Name,Value)` %求符号方程组的解。

例 A.17 解方程组

$$\begin{cases} x + 2y - z = 27, \\ x + z = 3, \\ x^2 - 3y^2 = 28. \end{cases}$$

`clc, clear, syms x y z`

`[sz,sy,sx]=solve([x+2*y-z==27,x+z==3,x^2-3*y^2==28],[z,y,x])`

`xx=double(sx), yy=double(sy), zz=double(sz)` %把符号数转换为浮点数

求得方程组的两组符号解，这里就不具体给出了。

2. 非线性方程（组）的数值解

（1）求非线性方程的零点

`roots` 求一元多项式的所有零点。

例 A.18 求下列方程的所有解

$$x^6 - 3x^2 - 1 = 0.$$

`clc, clear`

`p=[1,0,0,0,-3,0,-1];`

`r=roots(p)`

求得方程的 6 个解分别为 $\pm 1.3709, \pm 1.2378i, \pm 0.5893i$ 。

`fzero` 求给定初值附近的一个数值解，其调用格式为

`x=fzero(fun,x0)` %`fun` 为定义方程的函数或匿名函数；`x0` 为一个值或区间，作为初始条件。

始条件。

例 A.19（续例 A.18）求方程

$$x^6 - 3x^2 - 1 = 0$$

在初值 0.2 附近的一个根。

`clc, clear`

`yx=@(x)x.^6-3*x.^2-1;`

`x=fzero(yx,0.2)`

求得的根为 1.3709。

（2）非线性方程组的解

MATLAB 函数 `fsolve` 求非线性方程组的解，其调用格式为：

`x=fsolve(fun,x0,options)`

其中，`fun` 是定义非线性方程组的函数或匿名函数，`x0` 为初值，`options` 为优化参数设置，返回值 `x` 是求得的初值 `x0` 附近的一个解。

例 A.20（续例 A.17）求方程组

$$\begin{cases} x+2y-z=27, \\ x+z=3, \\ x^2-3y^2=28. \end{cases}$$

在初值 $[x, y, z] = [0.2, 2, 5]$ 附近的一组解。

```
clc, clear
f=@(t)[t(1)+2*t(2)-t(3)-27; t(1)+t(3)-3
      t(1)^2-3*t(2)^2-28]; %定义方程组的匿名函数
s=fsolve(f,[0.2,2,5])
求得的一个解为  $x=10.0601$ ,  $y=4.9399$ ,  $z=-7.0601$ 。
```

例 A.21 求方程组

$$\begin{cases} 291(x_1 - x_2) = 1, \\ 347X = x_1 + x_2, \\ 134X + x_3 = 0, \\ 495X = x_4 \end{cases}$$

的解，其中 $X = x_4(x_1 + x_2) - 2x_3^2$ 。

```
clc, clear
X=@(x)x(4)*(x(1)+x(2))-2*x(3)^2;
fx=@(x)[291*(x(1)-x(2))-1
        347*X(x)-x(1)-x(2)
        134*X(x)+x(3)
        495*X(x)-x(4)]
x0=fsolve(fx,rand(1,4)) %初值取为一个随机的行向量
```

A.5.3 矛盾方程组的最小二乘解

1. 线性方程组的最小二乘解

上面已经给出求线性方程组 $A\mathbf{x} = \mathbf{b}$ 的最小二乘解的命令为 $\mathbf{x} = \text{pinv}(A) * \mathbf{b}$ 。

例 A.22 为了测量刀具的磨损速度，我们做这样的实验：经过一定时间（如每隔一小时），测量一次刀具的厚度，得到一组实验数据 $(t_i, y_i) (i=1, 2, \dots, 8)$ 如表 A.16 所示。试根据实验数据建立 y 与 t 之间的经验公式 $y = at + b$ 。

表 A.16 实验观测数据

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

解 拟合参数 a, b 的准则是最小二乘准则，即求 a, b ，使得

$$\delta(a, b) = \sum_{i=1}^8 (at_i + b - y_i)^2$$

达到最小值。实际上是求解如下关于 a, b 作为未知数的线性方程组

$$\begin{cases} t_1 a + b = y_1, \\ t_2 a + b = y_2, \\ \dots\dots\dots \\ t_8 a + b = y_8. \end{cases}$$

上述方程组是两个未知数、8 个方程的超定线性方程组，MATLAB 求解时，刚好就给出最小二乘解。

MATLAB 的求解程序为：

```
clc, clear
t=[0:7]';
y=[27.0, 26.8, 26.5, 26.3, 26.1, 25.7, 25.3, 24.8]';
A=[t,ones(8,1)]; %构造线性方程组的系数矩阵
x=A\y %这里 A 是列满秩，使用 pinv(A)*y 的简化格式
```

求得 $a = -0.3036$, $b = 27.1250$ 。

2.非线性方程组的最小二乘解

对于非线性方程组

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

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

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

的最小值。MATLAB 求非线性方程组最小二乘解的命令为

`lsqnonlin(fun,x0),`

其中 `fun` 是定义向量函数

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

的匿名函数的返回值, `x0` 为 \mathbf{x} 的初始值。

例 A. 23 已知 4 个观测站的位置坐标 $(x_i, y_i)(i=1,2,3,4)$, 每个观测站都探测到距未知信号的距离 $d_i(i=1,2,3,4)$, 已知数据见表 A.17, 试定位未知信号的位置坐标 (x, y) 。

表 A.17 观测站的位置坐标及探测到的距离

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

解 未知信号的位置坐标 (x, y) 满足非线性方程组:

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

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

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

的最小点问题。

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

计算的 MATLAB 程序如下:

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

A. 6 常微分方程的求解

A.6.1 常微分方程的符号解

在 MATLAB 中, 符号运算工具箱提供了功能强大的求常微分方程符号解命令 `dsolve`。

1. 老版本 MATLAB 中 `dsolve` 函数的使用

在老版本 MATLAB 中, 用符号 `D` 表示对变量的求导, 如 `Dy` 表示对变量 `y` 求一阶导数, 当需要求变量的 `n` 阶导数时, 用 `Dn` 表示, `D4y` 表示对变量 `y` 求 4 阶导数。

例 A.24 求解二阶线性微分方程 $y'' - 2y' + y = e^x$, $y(0) = 1$, $y'(0) = -1$ 。

```
clc,clear
y=dsolve('D2y-2*Dy+y=exp(x)','y(0)=1,Dy(0)=-1','x')
pretty(y) %书写习惯的显示方式
```

得到二阶微分方程解析解 $y = e^x + \frac{x^2 e^x}{2} - 2xe^x$ 。

注 A.3: `dsolve` 命令中默认的函数自变量为 `t`, 如果函数的自变量为 `x`, 必须指明。

2. 新版本 MATLAB 中 `dsolve` 函数的使用

新版本 MATLAB 中使用函数 `diff` 求符号导数。

新版本中 `dsolve` 的调用格式如下:

`S=dsolve(eqn, cond, Name, Value)` %eqn 为符号微分方程, cond 为初值或边值条件。

`[y1, ..., yN]=dsolve(eqns, conds, Name, Value)` %eqns 为符号微分方程组, conds 为初值或边值条件。

例 A.25(续例 A.24)求解二阶线性微分方程 $y'' - 2y' + y = e^x$, $y(0) = 1$, $y'(0) = -1$ 。

```
clc, clear, syms y(x) %定义符号函数 y, 自变量为 x
dy=diff(y); %定义 y 的一阶导数, 目的是下面赋初值
y=dsolve(diff(y,2)-2*diff(y)+y==exp(x),y(0)==1,dy(0)==-1)
```

例 A.26 求下列 Logistic 方程

$$\begin{cases} \frac{dx}{dt} = r(1 - \frac{x}{x_m})x, \\ x(t_0) = x_0. \end{cases}$$

的符号解。

```
clc, clear, syms x(t) r xm x0 t0
x=dsolve(diff(x)==r*(1-x/xm)*x,x(t0)==x0)
x=simplify(x), pretty(x) %化简, 并用分数线居中的显示格式显示
求得
```

$$x(t) = \frac{x_0 x_m e^{rt}}{x_0 e^{rt} - x_0 e^{rt_0} + x_m e^{rt_0}}.$$

3. 求线性微分方程组的符号解

例 A.27 试求解下列 Cauchy 问题

$$\begin{cases} \frac{dx}{dt} = Ax, \\ x(0) = [1, 1, 1]^T. \end{cases}$$

的解, 其中 $x(t) = [x_1(t), x_2(t), x_3(t)]^T$, $A = \begin{bmatrix} 3 & -1 & 1 \\ 2 & 0 & -1 \\ 1 & -1 & 2 \end{bmatrix}$.

```
clc, clear, syms x1(t) x2(t) x3(t)
x=[x1;x2;x3]; A=[3 -1 1;2 0 -1;1 -1 2];
x0=ones(3,1); %初值条件
```

```
x=dsolve(diff(x)==A*x,x(0)==x0)
s=[x.x1; x.x2; x.x3] %以向量的形式显示解
pretty(s)
```

$$\text{求得的解为 } x(t) = \begin{bmatrix} \frac{4}{3}e^{3t} - \frac{1}{2}e^{2t} + \frac{1}{6} \\ \frac{2}{3}e^{3t} - \frac{1}{2}e^{2t} + \frac{5}{6} \\ \frac{2}{3}e^{3t} + \frac{1}{3} \end{bmatrix}.$$

A.6.2 常微分方程的数值解

大多数常微分方程（组）是无法求得符号解的，只能求数值解。**MATLAB** 有多个求数值解的命令，这里我们只介绍 **ode45**，**ode45** 用龙格库塔方法求一阶微分方程或微分方程组的数值解。

对一阶微分方程或方程组的初值问题

$$\begin{cases} y' = f(t, y), \\ y(t_0) = y_0, \end{cases}$$

其中 y 和 f 可以为向量. 函数**ode45**有如下两种调用格式:

[t,y]=ode45(fun,tspan,y0) 或 **s=ode45(fun,tspan,y0)**

其中 **fun** 是用 **M** 函数或匿名函数定义的 $f(t, y)$ ，**tspan**=[t_0 ,**tfinal**](这里 t_0 必须是初值条件中自变量的取值，**tfinal** 可以比 t_0 小)是求解区间，**y0** 是初值。返回值 **t** 是 **MATLAB** 自动离散化的区间[t_0 ,**tfinal**]上的点，**y** 的列是对应于 **t** 的函数值；如果只有一个返回值 **s**，则 **s** 是一个结构数组。

利用结构数组 **s** 和 **MATLAB** 函数 **deval**，我们可以计算任何感兴趣点 **x** 的函数值，调用格式为

y=deval(s,x)

其中 **x** 为标量或向量，返回值 **y** 的行是对应于 **x** 的数值解。

1.求一阶微分方程（组）的数值解

例 A.28 求微分方程

$$\begin{cases} y' = -2y + 2x^2 + 2x, \\ y(0) = 1. \end{cases}$$

在 $0 \leq x \leq 0.5$ 时的数值解。

```
clc, clear
yx=@(x,y)-2*y+2*x^2+2*x; %定义微分方程右端项的匿名函数
[x,y]=ode45(yx,[0,0.5],1) %第一种返回格式
sol=ode45(yx,[0,0.5],1) %第二种返回格式
y2=deval(sol,x) %计算自变量 x 对应的函数值，这里 y2 为行向量
check=[y,y2'] %比较两种返回值的結果是一样的，但一个是列向量，一个是行向量
```

例 A.29 Lorenz 模型的混沌效应

Lorenz 模型是由美国气象学家 **Lorenz** 在研究大气运动时，通过对对流模型简化，只保留 3 个变量提出的一个完全确定性的三阶自治常微分方程组（不显含时间变量），其方程为

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = \rho x - y - xz, \\ \dot{z} = xy - \beta z. \end{cases}$$

其中三个参数 σ 为 Prandtl 数, ρ 为 Rayleigh 数, β 为方向比。Lorenz 模型如今已经成为混沌领域的经典模型,第一个混沌吸引子—Lorenz 吸引子也是在这个系统中被发现的。系统中三个参数的选择对系统会不会进入混沌状态起着重要的作用。图 A.7 给出了 Lorenz 模型在 $\sigma=10$, $\rho=28$, $\beta=8/3$ 时系统的三维演化轨迹。由图 A.7 可见,经过长时间运行后,系统只在三维空间的一个有限区域内运动,即在三维相空间里的测度为零。图 A.7 显示出我们经常听到的“蝴蝶效应”。

计算及画图的 MATLAB 程序如下:

```
clc, clear
sigma=10; rho=28; beta=8/3;
g=@(t,f)[sigma*(f(2)-f(1)); rho*f(1)-f(2)-f(1)*f(3); f(1)*f(2)-beta*f(3)]; %定义微分方程组的右端项
xyz0=rand(3,1); %初始值
[t,xyz]=ode45(g,[0,100],xyz0); %求数值解
plot3(xyz(:,1),xyz(:,2),xyz(:,3)) %画图 9-2
xlabel('\it x(t)'), ylabel('\it y(t)'), zlabel('\it z(t)','rotation',0)
box on %加盒子线, 以突出立体感
```

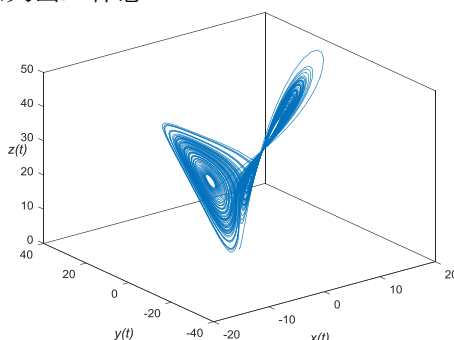


图 A.7 典型的 Lorenz 相轨线

例 A.30 求下列常微分方程组的数值解。

$$\begin{cases} \dot{x}_i = v_i, & i=1,2,3, \\ \dot{v}_i = \sum_{j=1}^3 a_{ij} \text{sign}(h(v_i) - h(v_j)) |h(v_i) - h(v_j)|^{0.3}, & i=1,2,3, \\ [x_1(0), x_2(0), x_3(0), v_1(0), v_2(0), v_3(0)] = [-1, 0, 1, 1, 0, -1]. \end{cases}$$

其中 $h(s) = s^3 + 2$, $A = (a_{ij})_{3 \times 3} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \\ 2 & 3 & 0 \end{bmatrix}$.

定义微分方程组右端项的 MATLAB 函数 (文件名为 Afun.m) 如下:

```
function dx=Afun(t,x)
a=[0 1 2;1 0 3;2 3 0]; alpha=0.3;
h=@(s)s.^3+2;
v=zeros(3,1);
for i=1:3
    for j=1:3
        v(i)=a(i,j)*sign(h(x(j+3))-h(x(i+3)))*abs(h(x(j+3))-h(x(i+3)))^alpha;
    end
end
end
```



```
dx=[x(4);x(5);x(6);v(1);v(2);v(3)];
求微分方程组数值解的 MATLAB 程序如下:
clc, clear
x0=[-1 0 1 1 0 -1]';
[t,x]=ode45(@Afun,[0,2],x0)
subplot(121),plot(t,x(:,[1:3]))
subplot(122),plot(t,x(:,[4:6]))
```

2.求高阶微分方程（组）的数值解

MATLAB 工具箱无法直接求解高阶方程（组）的数值解，对于高阶常微分方程(组)，首先要做变量变换化为一阶方程组，然后再用 MATLAB 的相关命令求数值解。

例 A.31(续例 A.25) 求解二阶线性微分方程 $y''-2y'+y=e^x$, $y(0)=1$, $y'(0)=-1$, 在区间 $[-1,0]$ 上的数值解，并与符号解进行比较。

求数值解时，首先做变量替换；设 $y_1=y$, $y_2=y'$ ，则把二阶微分方程化成如下的两个函数的一阶方程组：

$$\begin{cases} y_1' = y_2, & y_1(0) = 1, \\ y_2' = 2y_2 - y_1 + e^x, & y_2(0) = -1. \end{cases}$$

计算和画图的 MATLAB 程序如下：

```
clc, clear, close all
syms y(x) %定义符号函数 y，自变量为 x
dy=diff(y); %定义 y 的一阶导数，目的是下面赋初值
y=dsolve(diff(y,2)-2*diff(y)+y==exp(x),y(0)==1,dy(0)==-1)
h=ezplot(y,[-1,0]) %画符号解的曲线
title(''), set(h,'Color','r','LineWidth',2)
hold on
dy=@(x,y)[y(2);2*y(2)-y(1)+exp(x)];
s=ode45(dy,[0,-1],[1;-1]); %注意求解区间[0,-1]左端点必须是初始时刻
fplot(@(x)deval(s,x,1),[-1,0],'*') %画第一个分量，即 y 在[-1,0]区间上的数值解
legend('符号解','[-1,0]区间上数值解')
```

A.7 MATLAB 的数据处理

A.7.1 细胞数组和结构数据

MATLAB 很多函数的返回值是细胞数组或结构数组，这里我们简单介绍一下细胞数组和结构数组。

1.细胞数组

细胞数组是 MATLAB 中的一类特殊的数组。在 MATLAB 中，由于有细胞数组这个数据类型，才能把不同类型、不同维数的数据组成为一个数组。

细胞数组的创建有两种方法：通过赋值语句或 cell 函数创建。

(1) 使用赋值语句创建细胞数组：细胞数组使用花括号“{}”来创建，使用逗号“,”或空格分隔同一行的单元，使用分号“;”来分行。

(2) 使用 cell 函数创建空细胞数组。

MATLAB 提供的与细胞数组有关函数如表 A.18 所示。

表 A.18 细胞数组的有关函数

函 数	说 明
cell2struct(cellArray,field,dim)	将细胞数组转换成结构数组
iscell(c)	判断指定数组是否是细胞数组
struct2cell(s)	将 $m \times n$ 的结构数组 s(带有 p 个域)转换成 $p \times m \times n$ 的细胞数组
mat2cell(A,m,n)	将矩阵拆分成细胞数组矩阵

cell2mat(c)	将细胞数组合并成矩阵
num2cell(A)	将数值数组转换成细胞数组
celldisp(c)	显示细胞数组内容
cellplot(c)	显示细胞数组结构图

2. 结构数组

有时需要将不同的数据类型组合成一个整体，以便于引用。这些组合在一个整体中的数据是相互联系的。例如，一个学生的学号、姓名、性别、年龄、成绩、家庭地址等项都是和该学生有联系的。

MATLAB 与其它高级语言一样具有结构类型的数据。结构类型是包含一组彼此相关、数据结构相同但类型不同的数据类型。结构类型的变量可以是任意一种 MATLAB 数据类型的变量，也可以是一维的、二维的或者多维的。但是，在访问结构类型数据的元素时，需要使用下标配合域名的形式。

MATLAB 提供两种方法建立结构数组，用户可以直接给结构数组成员变量赋值建立结构数组，也可以利用函数 `struct` 建立结构数组。

MATLAB 提供的与结构数组有关函数如表 A.19 所示。

表 A.19 结构数组的有关函数

函数名	作用
<code>struct</code>	生成结构数组
<code>fieldnames(s)</code>	获取指定结构数组所有域名
<code>getfield(s,'field')</code>	获取指定域的值
<code>isfield(s,'field')</code>	判断是否是指定结构数组中的域
<code>orderfields(s)</code>	对结构数组域名按首字符重新排序
<code>setfield(s,'field',value)</code>	设置结构数组指定域的值
<code>rmfield(s,'field')</code>	删除指定结构数组中的域
<code>isstruct</code>	检查数组是否为结构类型

MATLAB 命令

`f=dir('*.*')`

可以显示当前目录下所有后缀名为 `m` 的文件信息，返回值 `f` 是一个结构数组，包括 5 个域：`name`，`date`，`bytes`，`isdir`，`datenum`；通过结构数组的元素个数就可以知道当前目录下 `m` 文件的个数，通过 `name` 域可以知道当前目录下所有 `m` 文件的名称。

`dir` 命令可以读出所有类型文件的信息。

例 A.32 读入当前目录下所有的后缀名为 `bmp` 的图片文件，并把数据保存在细胞数组中。

```
clc, clear
f=dir('*.*'); %读入当前目录下所有 bmp 图像文件的信息，保存在结构数组 f 中
n=length(f) %计算 bmp 文件的个数
for i=1:n
    a{i}=imread(f(i).name); %把第 i 个 bmp 图像数据保存在细胞数组中的第 i 个元素中
end
```

A.7.2 一般文件操作

MATLAB 中的关于文件方面的函数和 C 语言相似，见表 A.20。

表 A.20 MATLAB 的文件操作命令

函数分类	函数名	作用
打开和关闭文	<code>fopen</code>	打开文件

件	<code>fclose</code>	关闭文件
读写二进制文件	<code>fread</code>	读二进制文件
	<code>fwrite</code>	写二进制文件
格式 I/O	<code>fscanf</code>	从文件中读格式数据
	<code>fprintf</code>	写格式数据
	<code>fgetl</code>	从文件中读行,不返回行结束符
	<code>fgets</code>	从文件中读行,返回行结束符
读写字符串	<code>sprintf</code>	把格式数据写入字符串
	<code>sscanf</code>	格式读入字符串
文件定位	<code>feof</code>	检验是否为文件结尾
	<code>fseek</code>	设置文件定位器
	<code>ftell</code>	获取文件定位器
	<code>frewind</code>	返回到文件的开头

1.文件的打开和关闭

对文件读写之前应该“打开”该文件，在使用结束之后应“关闭”该文件。

函数 `fopen` 用于打开文件，其调用格式为：

`fid=fopen(filename,permission)`

`fid` 是文件标识符（file identifier），`fopen` 指令执行成功后就会返回一个正的 `fid` 值，如果 `fopen` 指令执行失败，`fid` 就返回-1。

`filename` 是文件名。

`permission` 是文件允许操作的类型，可设为以下几个值：

‘r’ 只读

‘w’ 只写

‘a’ 追加（append）

‘r+’ 可读可写

与 `fopen` 对应的指令为 `fclose`，它用于关闭文件，其指令格式为：

`status=fclose(fid)`

如果成功关闭文件，`status` 返回的值就是 0。

注 A.4 一定要养成好的编程习惯，文件操作完之后，要关闭文件，即释放文件句柄，如果不关闭句柄，还占用内存空间，如果打开的文件数量太多，内存会溢出的。

2.二进制文件的读写操作

读二进制文件的函数 `fread`，其调用格式为

`[A,count]=fread(fileID,sizeA,precision)`

其中，`A` 是用于存放读取数据的矩阵，`count` 是返回所读取的数据元素个数；`fileID` 是文件句柄；`sizeA` 为可选项，若不选用则读取整个文件内容，若选用则它的值可以是下列值：`N`（读取 `N` 个元素到一个列向量）、`inf`（读取整个文件）、`[m,n]`（读取数据到 `m×n` 矩阵中，数据按列存放）；`precision` 控制所读数据的精度。

写二进制文件的函数 `fwrite`，其调用格式为

`fwrite(fileID, A, precision)`

其中 `fileID` 为文件句柄，`A` 是要写入文件的数组，`precision` 控制所写数据的精度。

3.加载各种类型数据文件命令 `importdata`

MATLAB 函数 `importdata` 可以把各种类型文件的数据加载到 MATLAB 工作空间中，其调用格式为：

`A=importdata(filename)` %将数据加载到数组 `A` 中。

`A=importdata('-pastespecial')` %从系统剪贴板而不是文件加载数据。

`A=importdata(___,delimiterIn)` %将 `delimiterIn` 解释为 ASCII 文件或剪贴板数据中的列分隔符。

`A=importdata(___,delimiterIn,headerlinesIn)` %从 ASCII 文件或剪贴板加载数据，并

读取从第 `headerlinesIn` 行开始的数值数据。

(1) 图像文件

例 A.33 导出并显示示例图像 `ngc6543a.jpg`。

```
A=importdata('ngc6543a.jpg'); %加载 MATLAB 内置图像文件
image(A)
```

显示的图像如图 A.8 所示。

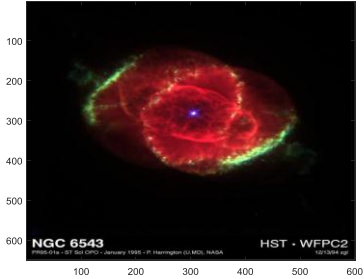


图 A.8 `ngc6543a.jpg` 对应的图像

(2) 文本文件

例 A.34 使用文本编辑器创建一个带有列标题的称为 `myfile01.txt` 的空格分隔 ASCII 文件。

```
Day1 Day2 Day3 Day4 Day5 Day6 Day7
95.01 76.21 61.54 40.57 5.79 20.28 1.53
23.11 45.65 79.19 93.55 35.29 19.87 74.68
60.68 1.85 92.18 91.69 81.32 60.38 44.51
48.60 82.14 73.82 41.03 0.99 27.22 93.18
89.13 44.47 17.63 89.36 13.89 19.88 46.60
```

使用 MATLAB 导入该文件。

```
clc, clear
filename='myfile01.txt';
delimiterIn=' '; headerlinesIn=1;
A=importdata(filename,delimiterIn,headerlinesIn) %返回值 A 为结构数组
a=A.data %提出其中的数值矩阵
```

(3) EXCEL 文件

例 A.35 EXCEL 文件 `myfile02.xlsx` 中存放如图 A.9 的数据。

	A	B	C	D	E	F	G
1	日期	开盘	最高	最低	收盘	交易量	交易额
2	2007/6/4	33.76	33.99	31	32.44	282444	921965312
3	2007/6/5	31.9	33	29.2	32.79	329276	1032631552
4	2007/6/6	31.9	32.86	31	32.27	236677	756290880
5	2007/6/7	32.41	34	32.16	32.73	255289	845447232
6	2007/6/8	32.7	32.7	31.18	31.6	272817	862057728

图 A.9 EXCEL 文件中的数据

使用 MATLAB 导入该文件。

```
clc, clear, format long g %长小数的显示格式
a=importdata('myfile02.xlsx'); %返回值 a 为结构数组
b=a.data.Sheet1 %提出其中的数值矩阵
format %恢复到短小数的显示格式
```

(4) 要导入文件的名称和扩展名

要导入文件的名称和扩展名，指定为字符向量或字符串。如果 `importdata` 可识别文件扩展名，则会调用用于导入关联文件格式的 MATLAB 辅助函数(如用于 `mat` 文件的 `load`，或用于 EXCEL 文件的 `xlsread`)。否则，`importdata` 会将文件解释为分割的 ASCII 文件。

对于 ASCII 文件和 EXCEL 文件，`importdata` 会查找矩阵格式(即像矩阵一样)的数值数据，文本标题可显示在数值数据的上方或左侧。

A.7.3 `mat` 文件

1.生成 mat 文件

利用 MATLAB 的 save 函数可以生成 mat 文件，其调用格式为：

save(filename) %将当前工作区中的所有变量保存在 MATLAB 格式的二进制文件（mat 文件）filename 中。

save(filename,variables) %仅保存 variables 指定的变量或字段。

save(filename,variables,fmt) %以 fmt 指定的文件格式保存。

save(filename,variables, '-append') %将新变量添加到一个现有文件中。

save filename 是命令形式的语法。命令形式无需键入括号和将输入括在单引号内。

例如，要保存名为 test.mat 的文件，下列语句是等效的：

save test.mat X %命令形式

save('test.mat','X') %函数形式

在 MATLAB 命令窗口运行

>>save matfile1

就把当前工作空间中的所有变量都保存到 matfile1.mat 文件中。

例 A.36 创建两个变量 p 和 q，并将其保存到名为 pqfile.mat 的文件中。

p=rand(1,10); q=ones(10);

save('pqfile.mat','p','q')

也可使用命令语法保存变量 p 和 q。

save pqfile.mat p q

2.读取 mat 文件数据

首先是要打开的 mat 文件所在的目录设置为当前工作目录，然后执行如下命令：

load(filename)或 load filename

就将 filename 中的所有数据加载到当前工作环境中。

例 A.37 加载例 A.36 生成文件 pqfile.mat 中的数据。

加载全部数据，使用命令

load('pqfile.mat')

或

load pqfile.mat

加载变量 p，使用命令

load('pqfile.mat','p')

或

load pqfile.mat p

A.7.4 文本文件

1.读取纯数值型数据的文本文件

读取数值型数据的文本文件的命令是 load 或 textread。

例 A.38 读取整行整列数值型数据的文本文件。

文本文件 Adata30.txt 存放如下格式的数据，把数据加载到 MATLAB 工作空间。

1 2 3 4

5 6 7 8

MATLAB 调用格式如下：

a=load('Adata30.txt')

或者

a=textread('Adata30.txt')

例 A.39 读取非整行整列数值型数据的文本文件。

文本文件 Adata31.txt 存放如下格式的数据，把数据加载到 MATLAB 工作空间。

1 2 3 4

5 6 7

MATLAB 调用格式如下：

a=textread('Adata31.txt')

注 A.5 得到的 a 矩阵的值为

```
1 2 3 4
5 6 7 0
```

2. 读取混合型数据的文本文件

读文本文件中的混合型数据，除了使用前面介绍的 `importdata` 命令外，还可以使用命令 `textscan`，`textscan` 可以从文本文件或字符串读取格式化数据。`textscan` 的调用格式为：

C=textscan(fileID, formatSpec): 将已打开的文本文件中的数据读取到细胞数组 **C**。该文本文件由文件标识符 `fileID` 指示。使用 `fopen` 可打开文件并获取 `fileID` 值。完成文件读取后，请调用 `fclose(fileID)` 来关闭文件。

C=textscan(fileID, formatSpec, N): 按 `formatSpec` 读取文件数据 **N** 次，其中 **N** 是一个正整数。要在 **N** 个周期后从文件读取其他数据，请使用原 `fileID` 再次调用 `textscan` 进行扫描。如果通过调用具有相同文件标识符 (`fileID`) 的 `textscan` 恢复文件的文本扫描，则 `textscan` 将在上次终止读取的点处自动恢复读取。

C=textscan(chr, formatSpec): 将字符向量 `chr` 中的文本读取到细胞数组 **C** 中。从字符向量读取文本时，对 `textscan` 的每一次重复调用都会从开头位置重新开始扫描。要上次位置重新开始扫描，需要指定 `position` 输出参数。`textscan` 尝试将字符向量 `chr` 中的数据与 `formatSpec` 中指定的格式匹配。

例 A.40 混合数据的读入。

文本文件 `Adata32.txt` 存放如下数据，读入其中的数据。

```
A 1 2 3 4
B 5 6 7 8
```

使用 `importdata` 命令的 MATLAB 程序如下：

```
a=importdata('Adata32.txt')
```

```
b=a.data %读取其中的数值矩阵
```

使用 `textscan` 命令的 MATLAB 程序如下：

```
fid=fopen('Adata32.txt')
```

```
a=textscan(fid,'%s%d%d%d%d','CollectOutput',1) %把相邻的同类型数据合并
```

```
b=a{2} %提出数值型数据矩阵
```

```
fclose(fid)
```

例 A.41 混合数据的读入。

文本文件 `Adata33.txt` 存放如下数据，读入其中的数据。

日期	开盘	最高	最低	收盘	交易量	交易额
2007/06/04	33.76	33.99	31.00	32.44	282444.00	921965312.00
2007/06/05	31.90	33.00	29.20	32.79	329276.00	1032631552.00
2007/06/06	31.90	32.86	31.00	32.27	236677.00	756290880.00
2007/06/07	32.41	34.00	32.16	32.73	255289.00	845447232.00
2007/06/08	32.70	32.70	31.18	31.60	272817.00	862057728.00

MATLAB 程序如下：

```
clc, clear, format long g %长小数的显示格式
```

```
fid = fopen('Adata33.txt');
```

```
fgetl(fid) %读第 1 行的表头
```

```
A=textscan(fid, '%s %f %f %f %f %f %f', 'CollectOutput', true) %A 为 1×2 的细胞数组
```

```
B=A{2} %提取数值矩阵
```

```
fclose(fid); format %恢复到短小数的显示格式
```

或者编写如下 MATLAB 程序：

```
clc, clear, format long g
```

```
a=importdata('Adata33.txt')
```

```
b=a.data %提出数值矩阵
```

```
format
```

例 A.42 混合数据的读入。

文本文件 **Adata34.txt** 存放如下数据，读入其中的数据。

Sally 09/12/2005 12.34 45 Yes

Larry 10/12/2005 34.56 54 Yes

Tommy 11/12/2005 67.89 23 No

MATLAB 程序如下：

```
clc, clear
fid=fopen('Adata34.txt');
A=textscan(fid, '%s %s %f %d %s', 'CollectOutput', 1) %合并细胞数组相邻同类型数据
B1=A{1,2}, B2=A{1,3} %提取需要的数值数据
fclose(fid);
```

例 A.43 输入非长方形文本文件中的数值数据。

文本文件 **Adata35.txt** 中存放如下数据：

```
begin
v1=12.67
v2=3.14
v3=6.778
end
begin
v1=21.78
v2=5.24
v3=9.838
end
```

MATLAB 读入数值数据的程序如下：

```
clc, clear
fid = fopen('Adata35.txt');
c=textscan(fid, '%*s v1=%f v2=%f v3=%f %*s', 'Delimiter', '\n', 'CollectOutput', 1)
c=cell2mat(c) %把细胞数组转换成数值矩阵
fclose(fid);
```

注 A.6 这里的 “*” 表示跳过一个字符串的数据域。

3.时间序列数据

例 A.44（续例 A.41） 读入 **Adata33.txt** 中的数据。

```
clc, clear, format long g
a=readtable('Adata33.txt','ReadRowName',false)
b=table2cell(a) %把表格转成细胞数组
c=cell2mat(b(:,2:end))) %把细胞数组转换成矩阵
format
```

例 A.45 时间序列数据构造及转换。

```
randn('seed',sum(100*clock)); %初始化随机数发生器
a=randn(6,1);
b=[today:today+5]' %从今天到后面 5 天
fts=fints(b,a) %生长 fints 格式数据
fts(3)=NaN; %将第 3 个数据变为缺失值 NaN
newdata=fillts(fts,'linear') %用线性插值填补时间序列中的缺失数据
data=fts2mat(newdata) %把时间序列数据转换成矩阵
```

例 A.46 对于 MATLAB 当前工作路径下的所有纯文本文件，进行相关的数据操作。

```

tf=dir('*.txt') %提出纯文本文件的信息，返回值是结构数据
n=length(tf); %计算纯文本文件的个数
fts=ascii2fts(tf(1).name); %读第一个文件中的时间序列数据
fts=extfield(fts,{'series2','series3'}); %提出第 2 和第 3 字段
for i=2:n
    tp1=ascii2fts(tf(i).name); %读入新的时间序列数据
    tp2=extfield(tp1,{'series2','series3'}); %提出第 2 和第 3 字段
    str1=['series',num2str(2*i)]; str2=['series',num2str(2*i+1)];
    tp3=fints(tp2.dates,fts2mat(tp2),{str1,str2}); %把时间序列字段改名，同名字段的
两个时间序列数据是无法合并的
    fts=merge(fts,tp3); %合并两个时间序列的数据
end

```

注 A.7 上面程序是使用我们自己的特定数据，这里只是说明编程思路。

4.字符串的文本数据

例 A.47 统计下列五行字符串中字符 a、c、g、t 出现的频数。

```

1.aggcacggaaaaacgggaataacggaggaggacttggcacggcattacacggagg
2.cggaggacaaacgggatggcggtattggaggtggcggactgttcgggga
3.gggacggatacggattctggccacggacggaaaggaggacacggcggacataca
4.atggataacggaaacaaaccagacaaacttcggtagaatacagaagctta
5.cggctggcggacaacggactggcggttccaaaaacggaggaggcggacggaggc

```

把上述五行复制到一个纯文本数据文件 Adata39.txt 中，编写如下程序：

```

clc, clear
fid=fopen('data39.txt','r');
i=1;
while (~feof(fid))
    data=fgetl(fid);
    a=length(find(data==97));
    b=length(find(data==99));
    c=length(find(data==103));
    d=length(find(data==116));
    e=length(find(data>=97&data<=122));
    f(i,:)= [a b c d e a+b+c+d];
    i=i+1;
end
f, he=sum(f)
fclose(fid);

```

例 A.48 某计算机机房的一台计算机经常出故障，研究者每隔 15 分钟观察一次计算机的运行状态，收集了 24 小时的数据（共作 97 次观察）。用 1 表示正常状态，用 0 表示不正常状态，所得的数据序列如下：

```

111001001111111001111011111100111111110001101101
1110110110101111011101111110011011111100111

```

求在 96 次状态转移中，“ $0 \rightarrow 0$ ， $0 \rightarrow 1$ ， $1 \rightarrow 0$ ， $1 \rightarrow 1$ ”各有几次？

把上述数据序列保存到纯文本文件 data1830.txt 中，存放在 MATLAB 的当前工作目录下。编写程序如下：

```

clc,clear
fid=fopen('data1830.txt','r');
a=[];
while (~feof(fid))
    a=[a fgetl(fid)]; %把所有读入的字符组成一个大字符串
end
for i=0:1

```



```

for j=0:1
    s=[int2str(i),int2str(j)]; %构造查找的子字符串
    f(i+1,j+1)=length(findstr(s,a));
end
end
f %显示统计矩阵
求得 96 次状态转移的情况是:

```

$0 \rightarrow 0$, 8 次; $0 \rightarrow 1$, 18 次;
 $1 \rightarrow 0$, 18 次; $1 \rightarrow 1$, 52 次.

5. 数据写入文本文件

例 A.49 使用 `dlmwrite` 命令把矩阵 `b` 保存到纯文本文件 `Adata36.txt`。

```

b=randi([1,6],5) %生成 5 阶随机整数矩阵
dlmwrite('Adata36.txt',b)

```

例 A.50 生成服从标准正态分布随机数的 300×200 矩阵, 然后用 `fprintf` 命令保存到纯文本文件 `Adata36.txt`。

```

clc, clear
fid=fopen('Adata37.txt','w');
a=normrnd(0,1,300,200);
fprintf(fid,'%f\n',a);
fclose(fid);

```

注 A.8 对于高维矩阵, 用 `dlmwrite` 构造的文本文件, Lingo 软件不识别; 为了 Lingo 软件识别, 文本文件必须用 `fprintf` 构造, 而且数据之间的分割符为“\n”。另外, MATLAB 数据是逐列存贮的, Lingo 数据是逐行存储的, 这里把 `a` 矩阵的转置矩阵 `a'` 写入文本文件供 Lingo 调用。

A.7.5 EXCEL 文件

1. 读入数据

MATLAB 读入 EXCEL 文件的命令是 `xlsread`, 使用格式为

```

num=xlsread(filename,sheet,Range)
[num,txt]=xlsread(filename,sheet,Range)

```

其中第 1 个返回值是数值矩阵, 第 2 个返回值是字符串的细胞数组, `sheet` 是表单序号, `Range` 是数据域的范围。

例 A.51 把 EXCEL 文件 `Adata37.xlsx` 的表单 `Sheet1` 的域“`A2:D5`”中的数据赋给 `a`, 表单 `Sheet2` 中的全部数据赋给 `b`。

```

a=xlsread('Adata38.xlsx',1,'A2:D5')
b=xlsread('Adata38.xlsx',2)

```

注 A.9 把 EXCEL 文件的所有表单数据全部读入也可以使用命令 `importdata`。

```

c=importdata('Adata38.xls')

```

返回值 `mydata` 是一结构数组。

例 A.52 已知 11 个地点的位置坐标如表 A.21 所示, 画出这 11 个点的位置并进行标注。

表 A.21 地点名称及坐标数据

位置名称	x 坐标	y 坐标	位置名称	x 坐标	y 坐标
基地 R	865	141	无名高地	690	131
基地 S	941	187	山谷 1	254	495
基地 T	711	841	山谷 11	736	443
101 高地	782	726	山谷 01	128	789

12 高地	769	385	山谷 001	349	816
116 高地	453	956			

把表 A.21 中的数据保存在 EXCEL 文件 Adata43.xlsx 中，编写 MATLAB 程序如下：

```
clc, clear
[a,b]=xlsread('Adata43.xlsx')
c=[a(:,[1,2]);a([1:end-1],[4,5])]; %提取需要的数据
b={b{:,1},b{[1:end-1],4}} %提取非空字符串，构造新的字符串细胞数组
plot(c([1:3],1),c([1:3],2),'P') %画前 3 个点
hold on, plot(c([4:end],1),c([4:end],2),'*') %画其余点
text(c(:,1)+5,c(:,2),b) %对所有的点进行标注
```

画出的 11 个地点的示意图见图 A.10。

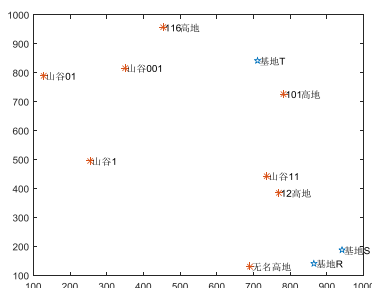


图 A.10 位置示意图

注 A.10 MATLAB 图形标注中使用的字符串一般要求是细胞数组字符串。

2.数据写入 EXCEL 文件

MATLAB 把数据写入 EXCEL 文件的命令是 `xlswrite`，使用格式为
`xlswrite(filename,A,sheet, Range)`

其中 `filename` 是要写入数据的文件名，`A` 是要写入的矩阵，`sheet` 是表单名或表单序号，`Range` 是数据域的地址或数据域的左上角地址。

例 A.53 把一个 5×10 矩阵 `a` 写到 EXCEL 文件 Adata40.xls 表单 Sheet1 中，把一个 3×3 矩阵 `b` 写入表单 Sheet2 的 B2 开始的域中。

```
a=rand(5,10);
xlswrite('Adata40.xls',a) %默认写入第 1 个表单，A1 开始的数据域中
b=rand(3);
xlswrite('data40.xls',b,2,'B2') % 2 是表单序号，B2 是数据域左上角开始地址
```

A.7.6 图像文件

例 A.54 把一个比较大的 bmp 图像文件 Adata46.bmp，转化成比较小的 jpg 文件，命名成 Adata46.jpg，并显示。

```
a=imread('Adata46.bmp'); %非工具箱图像文件，我们自己的一个图像文件
subplot(121), imshow(a), title('原 bmp 图像') %显示原 bmp 图像
imwrite(a,'Adata46.jpg');
subplot(122), imshow('data1832.jpg'), title('新 jpg 图像') %对比显示新的 jpg 图像
```

例 A.55 生成 10 幅彩色 jpg 文件，依次命名成 `jpg1.jpg`，...，`jpg10.jpg`。

```
clc, clear
for i=1:10
    str=['jpg',int2str(i),'.jpg'];
    a(:,1)=rand(500); a(:,2)=rand(500)+100; a(:,3)=rand(500)+200;
```

```

    imwrite(a,str);
end

```

例 A.56 碎纸片的拼接复原（2013 年全国大学生数学建模竞赛 B 题附件 1）。

```

clc, clear, file=dir('*.bmp'); %读 bmp 文件的信息
tind=1; a=imread(file(1).name); %读入第一个图像
a=double(a); jj=[2:19];
L1=a(:,1); L2=a(:,end); %拼接的大图形左边界和右边界初始化
for i=1:18
    tcha1=[]; tcha2=[];
    for j=jj
        a2=imread(file(j).name); a2=double(a2);
        e1=a2(:,1); e2=a2(:,end);
        cha1=sum(abs(L1-e1)); cha2=sum(abs(L2-e1));
        tcha1=[tcha1,cha1]; tcha2=[tcha2,cha2]; %计算左右边界的差异
    end
    m1=min(tcha1); m2=min(tcha2);
    if sum(abs(L1-255))<1
        ind=find(tcha2==m2); tind=[tind,jj(ind)]; %右拼接
        tt=imread(file(jj(ind)).name); tt=double(tt);
        L2=tt(:,end); jj(ind)=[];
    elseif sum(abs(L2-255))<1
        ind=find(tcha1==m1); tind=[jj(ind),tind]; %左拼接
        tt=imread(file(jj(ind)).name); tt=double(tt);
        L1=tt(:,1); jj(ind)=[];
    else
        if m1<m2
            ind=find(tcha1==m1); tind=[jj(ind),tind]; %左拼接
            tt=imread(file(jj(ind)).name); tt=double(tt);
            L1=tt(:,1); jj(ind)=[];
        else
            ind=find(tcha2==m2); tind=[tind,jj(ind)]; %右拼接
            tt=imread(file(jj(ind)).name); tt=double(tt);
            L2=tt(:,end); jj(ind)=[];
        end
    end
end
end
ta=[];
for k=1:19
    ta=[ta,imread(file(tind(k)).name)];
end
tind=tind-1, imshow(ta) %显示拼图顺序及拼接结果

```

习题 A

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

A.2 分别用 mesh 和 ezmesh 画出二元函数 $f(x, y) = (x^2 - 2x)e^{-x^2 - y^2 - xy}$ 的图形。

A.3 附件 1: 区域高程数据.xlsx 给出了某区域 43.65×58.2 (km) 的高程数据, 画出该区域的三维网格图和等高线图, 在 A (30, 0) 和 B (43, 30) (单位: km) 点处建立了两个基地, 在等高线图上标注出这两个点。

A.4 已知

$$\begin{cases} \int_{10}^{20} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 0.6006, \\ \int_0^9 \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 0.2661, \end{cases}$$

求 μ, σ 的值。

A.5 计算下列行列式的值

$$(1) D_{10} = \begin{vmatrix} a & & & & b \\ & \ddots & & & \\ & & a & b & \\ & & c & d & \\ & \ddots & & & \\ c & & & & d \end{vmatrix}, \quad (2) D_{10} = \begin{vmatrix} a & b & & & \\ c & a & b & & \\ & \ddots & \ddots & \ddots & \\ & & c & a & b \\ & & & c & a \end{vmatrix}.$$

A.6 化二次型

$$f = 2x_1^2 + x_2^2 - 4x_1x_2 - 4x_2x_3$$

为标准型，并画出 $f=1$ 的曲面。

A.7 求解下列非齐次线性方程组：

$$\begin{cases} 2x + y - z + w = 1, \\ 3x - 2y + z - 3w = 4, \\ x + 4y - 3z + 5w = -2. \end{cases}$$

A.8 求解下列非齐次线性方程组：

$$\begin{cases} 4x_1 - x_2 = 1, \\ 2x_1 + 4x_2 - x_3 = 2, \\ 2x_2 + 4x_3 - x_4 = 3, \\ \ddots \\ 2x_{98} + 4x_{99} - x_{100} = 99, \\ 2x_{99} + 4x_{100} = 100. \end{cases}$$

A.9 在研究某单分子化学反应速度时，得到数据 $(t_i, y_i) (i=1, 2, \dots, 8)$ 如表所示。中 t 表示从实验开始算起的时间， y 表示时刻 t 反应物的量。试根据上述数据定出经验公式 $y = ke^{mt}$ ，其中 k, m 是待定常数。

表 A.22 观测数据

t_i	3	6	9	12	15	18	21	24
y_i	57.6	41.9	31.0	22.7	16.6	12.2	8.9	6.5

A.10 设矩阵 $A = \begin{bmatrix} 1 & -2 & -4 \\ -2 & x & -2 \\ -4 & -2 & 1 \end{bmatrix}$ 与 $B = \begin{bmatrix} 5 & & \\ & -4 & \\ & & y \end{bmatrix}$ 相似，求 x, y 。

A.11 已知 $f(x) = (|x+1| - |x-1|)/2 + \sin x$ ， $g(x) = (|x+3| - |x-3|)/2 + \cos x$ ，求下列超定（矛盾）方程组的最小二乘解。

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

A.12 求微分方程组的数值解。

$$\begin{cases} x' = -x^3 - y, & x(0) = 1, \\ y' = x - y^3, & y(0) = 0.5, \end{cases} \quad 0 \leq t \leq 30.$$

要求画出 $x(t), y(t)$ 的解曲线图形，在相平面上画出轨线。

A.13 求微分方程组（竖直加热板的自然对流）的数值解。

$$\begin{cases} \frac{d^3 f}{d\eta^3} + 3f \frac{d^2 f}{d\eta^2} - 2\left(\frac{df}{d\eta}\right)^2 + T = 0, \\ \frac{d^2 T}{d\eta^2} + 2.1f \frac{dT}{d\eta} = 0. \end{cases}$$

已知当 $\eta = 0$ 时， $f = 0$ ， $\frac{df}{d\eta} = 0$ ， $\frac{d^2 f}{d\eta^2} = 0.68$ ， $T = 1$ ， $\frac{dT}{d\eta} = -0.5$ 。要求在区间 $[0, 10]$ 上，画出 $f(\eta), T(\eta)$ 的解曲线。

A.14 报童以每份 0.3 元的价格买进报纸，以 0.5 元的价格出售。当天销售不出去的报纸将以 0.2 元的价格退还报社。根据长期统计，假设已经得到了 159 天报纸需求量的情况见表 A.23。

表 A.23 159 天报纸需求量情况

需求量 r	100	120	140	160	180	200	220	240	260	280
天数	3	9	13	22	32	35	20	15	8	2

现在需用数学模型解决以下问题：

（1）若将报纸需求量看作离散型分布，试根据给出统计数据，求出报纸需求量的分布律，并建立数学模型，确定报童每天买进报纸的数量，使报童的平均总收入最大？

（2）若将报纸需求量看作连续型分布，试根据给出的统计数据，进行分布假设检验，确定该报纸需求量的分布，并建立数学模型，确定报童每天买进报纸的数量，使报童的平均总收入最大？