第1章

MATLAB 仿真基础

电工技术领域涉及电路系统的模型建立、系统分析、系统设计的基本理论和相关技术，其特点是概念抽象、数学含量大、计算繁杂、曲线绘制复杂，学生难以理解和掌握。20 世纪 80 年代初期出现的以矩阵运算为特点的 MATLAB 语言，具有可视化编程能力的图形用户界面、Simulink 仿真功能和电气系统工具箱中丰富的库函数等，为电工技术领域的设计与仿真提供了强有力的工具，给系统的分析与设计带来了极大的方便，成为电路分析领域的一个很好的辅助工具，现今 MATLAB 已经成为国际、国内科学界最流行的辅助分析工具之一。

本章主要介绍有关 MATLAB 语言的一些基本知识，如 MATLAB 语言的特点、数值计算、基本的程序设计功能以及 Simulink 仿真集成环境等。

1.1 MATLAB 概述

1.1.1 初识 MATLAB

MATLAB 是“Matrix Laboratory”的缩写，意为“矩阵实验室”，是当今国际很流行的科学计算软件。在欧美大学里，诸如应用代数、数理统计、电工电子、自动控制、数字信号处理、模拟与数字通信、动态系统仿真等课程的教科书，都有 MATLAB 内容。MATLAB 是攻读学位的大学生、硕士生、博士生必须掌握的基本工具。在设计研究单位和工业部门，MATLAB 被认作是进行高效研究和开发的首选软件工具。在许多诸如控制论、时间序列分析、系统仿真、图像信号处理等方面产生的大量矩阵及其相应的计算问题，自己去编写大量的繁复的计算程序，不仅会消耗大量的时间和精力，减缓工作进程，而且往往质量不高。Mathwork 软件公司推出的 MATLAB 软件给人们提供了一个方便的数值计算平台。

MATLAB 是一个交互式的系统，它的基本运算单元是不需指定维数的矩阵，按照 IEEE 的数值计算标准（能正确处理无穷数 Inf（Infinity）、无定义数 NaN（Not-A-Number）及其运算）进行计算。系统提供了大量的矩阵及其他运算函数，可以方便地进行一些很复杂的计算，而且运算效率极高。MATLAB 命令和数学中的符号、公式非常接近，可读性强，容易掌握，还可利用它所提供的编程语言进行编程完成特定的工作。除基本部分外，MATLAB 还根据各专

门领域中的特殊需要提供了许多可选的工具箱，如应用于电工领域的电气系统(Power System)工具箱、自动控制领域的 Control System 工具箱和神经网络领域的 Neural Network 工具箱等。尤为重要的是，MATLAB 提供了可视化动态仿真环境——Simulink，可实现动态系统的直观建模、仿真与分析，并支持连续、离散及两者混合的线性和非线性系统，因此使一个复杂系统的输入和仿真变得相当简单。

现在，MATLAB 已经不仅仅是一个“矩阵实验室”了，它已经发展为适合多学科，功能强大的大型软件，成为高级课程的基本教学工具。如 MATLAB 可以实现以下功能：

① 微积分：微分，积分，求极限，泰勒展开，级数求和；

② 代数：求逆，特征值，行列式，代数方程解的简化，数学表达式的指定精度求值；

③ 数值分析：插值与拟合，数值微分与积分，函数逼近，代数方程和微分方程的数值解和符号解；

④ 统计计算：均值，方差，概率，参数估计，假设检验，相关性和回归分析，统计绘图，随机数产生器等；

⑤ 优化问题的求解：线性规划，非线性规划等问题的求解；

⑥ 动态系统模拟仿真等。

MATLAB 可以提供的工具箱如表 1-1 所示。

表 1-1 MATLAB 常见工具箱

Signal Process	信号处理	System Identification	系统辨识
Optimization	优化	Neural Network	神经网络
Control System	自动控制	Spline	样条
Symbolic Math	符号代数	Image Process	图像处理
Nonlinear Control	非线性控制	Statistics	统计

1.1.2 MATLAB 的特点

一种语言之所以能如此迅速地普及，显示出如此旺盛的生命力，是由于它有着不同于其他语言的特点，正如同 FORTRAN 和 C 等高级语言使人们摆脱了需要直接对计算机硬件资源进行操作一样，被称作第四代计算机语言的 MATLAB，利用其丰富的函数资源，使编程人员从繁琐的程序代码中解放出来。MATLAB 最突出的特点就是简洁。MATLAB 用更直观的、符合人们思维习惯的代码，代替了 C 和 FORTRAN 语言的冗长代码。MATLAB 给用户带来的是最直观、最简洁的程序开发环境。以下简单介绍 MATLAB 的主要特点。

1. 简洁灵活的语言风格

语言十分简单，使用灵活方便，书写形式自由，具有“草稿纸”功能。如果不是太多的任务，那么在命令窗口中可以按照自己的思路写入命令，并且可以即时看到结果。

2. 方便的数值运算

在 MATLAB 环境中，有超过 500 种数学、统计、科学及工程方面的函数可使用，函数的标识自然，使得问题和解答像数学式子一般简单明了，让使用者可全力发挥在解题方面，而

非浪费在计算机操作上。

3. 先进的资料视觉化功能

MATLAB 的物件导向图形架构让使用者可执行视觉数据，并制作高品质的图形，完成科学性或工程性图文并茂的文章。

4. 功能强大的工具箱

工具箱可分两类，即功能性工具箱和学科性工具箱。前者主要用来扩充其符号计算功能、图示建模仿真、文字处理及与硬件实时交互的功能。而学科性工具箱是专业性较强的，如优化、统计、控制、小波、图像处理和通信工具箱等。

5. 高阶但简单的程式环境

做为一种直译式的程式语言，MATLAB 允许使用者在短时间内写完程序，所花的时间约为用 FORTRAN 或 C 的几分之一，而且不需要编译(compile)及链接 (link) 即能执行，同时包含了更多及更容易使用的内建功能。

6. 开放及可延伸的架构

MATLAB 允许使用者接触它大多数的数学原使码，检视运算法，更改现存函数，甚至加入自己的函数，使 MATLAB 成为使用者所需要的环境。

7. 完善的帮助系统

MATLAB 有非常完善的帮助系统。总地来讲，有以下几个方面：

(1) 临场帮助

这些帮助内容，大多嵌附在 M 文件中，即时性强，反应速度快。它对求助内容的回答最及时准确，用 help 命令查询即可，方便可靠。

(2) 综合型在线帮助文档 helpdesk

该文库以 HTML 超文本形式独立存在。整个文库按 MATLAB 的功能和核心内容编排，系统性强，且可以借助“超链接”方便地进行交叉查阅。

(3) 演示软件 demo

这是一个内容广泛的演示程序。MATLAB 一向重视演示软件的设计，软件本身就带有完善的演示程序。

1.2　MATLAB 数值与运算

MATLAB 语言最早是专门为进行矩阵计算所设计的一门语言，数值数组(Numeric Array)和数组运算(Array Operations)始终是 MATLAB 的核心内容。自 MATLAB 5. x 版起，由于其“面向对象”的特征，这种数值数组(以下简称为数组)成为了 MATALB 最重要的一种内建数据类型(Built-in Data Type)，而数值运算就是定义在这种数据结构上的方法(Method)。MATLAB 语言作为一门主要用于计算的语言，与 C 或 FORTRAN 等高级语言有所不同。在本节中，主要介绍 MATLAB 语言数值运算的功能，这是 MATLAB 的最基本和最重要的部分。

1.2.1　变量与赋值

变量代表一个或若干个内存单元，为了对变量所对应的存储单元进行访问，需要给变量

命名。在 MATLAB 中，变量名是以字母开头，后接字母、数字或下划线的字符序列，最多 63 个字符。例如，matl12、mat_vl、mysim56_均为合法的变量名，而 56mysim、_mat 为非法的变量名。另外，在 MATLAB 中，变量名区分字母的大小写，如 mat、MAT 和 maT 表示三个不同的变量。

值得注意的是，MATLAB 提供的标准函数名以及命令名必须用小写字母。例如，求矩阵 A 的逆矩阵用 inv(A)，不能写成 Inv(A)或 INV(A)，否则会出错。

MATLAB 赋值语句有两种格式：

(1) 变量＝表达式；

(2) 表达式。

其中表达式是用运算符将有关运算量连接起来的式子，其结果是一个矩阵。

在第一种语句形式下，MATLAB 将右边表达式的值赋给左边的变量，而在第二种语句形式下，将表达式的值赋给 MATLAB 的预定义变量 ans。

在 MATLAB 后面可以加上注释，用于解释或说明语句的含义，对语句结果不产生任何影响。注释以％开头，后面是注释的内容。

在赋值运算符右边使用变量，必须事先给变量赋值，因此，下面的表达式会产生错误：

```
>>x=2
x=
    2
>>t=x+a
???Undefined function or variable 'a'.
```

下面的表达式则不会产生错误：

```
>>x=2
x=
    2
>>a=3.5
a=
    3.5000
>>t=x+a
t=
    5.5000
```

在很多时候，并不需要 MATLAB 输出结果，只需要在表达式后面加上分号(;)即可。在下面的命令中，开始输入 x ＝3，MATLAB 及时地报告结果，第二次输入“x＝3;”，MATLAB 就没有再花费空间输出结果，而是直接跳到命令提示符，等待下次输入。

```
>>x=3
x=
    3
>>x=3;
```

我们还可以在一行中包含多个表达式。例如，下面的表达式是合法的：

```
>>x=2; y=4; z=x*y
```

```
z=
    8
```

注意那两个分号，它们告诉 MATLAB 不需要输出 x 和 y 的值。

当做许多计算时，结果可能会产生大量变量，可以通过在命令窗口中输入 who 来刷新内存，告诉 MATLAB 显示到目前为止所有变量名称。如果输入 whos，我们会得到更多信息，告诉我们当前内存中的变量、类型、每个变量所分配的内存空间以及它们是否是复数。要清除全部变量只需输入 clear 然后回车即可，要清除特定变量，则在 clear 后面带上变量名列表。

MATLAB 的数据显示格式（见表 1-2）由 format 命令来控制，它只影响结果在屏幕上的显示，不影响其计算与存储。MATLAB 总是以双精度执行所有的运算。

表 1-2　数据的显示格式

格　　式		对 应 结 果	
命　　令	含　　义	4/3	1.2345e－6
format short	短格式	1.3333	0.0000
format long	长格式	1.33333333333333	0.00000123450000
format short e	短格式 e 方式	1.3333e＋000	1.2345e－006
format long e	长格式 e 方式	1.33333333333333e＋000	1.234500000000000e－006
format short g	短格式 g 方式	1.3333	1.2345e－006
format long g	长格式 g 方式	1.33333333333333	1.234500000000000e－006
format hex	16 进制格式	3ff5555555555555	3eb4b6231abfd271
format ＋	＋格式	＋	＋
format rat	分数格式	4/3	1/810045
format bank	银行格式	1.33	0.00

1.2.2　MATLAB 常用数学函数

MATLAB 提供丰富的标准初等数学函数，包括 abs（绝对值）、sqrt（平方根）、exp（指数函数）和 sin（正弦函数）等。对负数取平方根或对数不会出错；MATLAB 将自动产生复数的结果。MATLAB 也提供很多高等数学函数，比如 Bessel（贝塞尔）函数和 Gamma（伽玛）函数。这些函数绝大部分支持复变量。要得到一个初等数学函数的列表，可以输入：

```
>>help elfun
```

要得到一个高等数学函数或者矩阵函数的列表，可以输入：

```
>>help specfun
>>help elmat
```

表 1-3～表 1-8 给出了常用的数学函数。

表 1-3　三角函数和双曲函数

名称	含　义	名称	含　义	名称	含　义
sin	正弦	csc	余割	atanh	反双曲正切
cos	余弦	asec	反正割	acoth	反双曲余切
tan	正切	acsc	反余割	sech	双曲正割
cot	余切	sinh	双曲正弦	csch	双曲余割
asin	反正弦	cosh	双曲余弦	asech	反双曲正割
acos	反余弦	tanh	双曲正切	acsch	反双曲余割
atan	反正切	coth	双曲余切	atan2	四象限反正切
acot	反余切	asinh	反双曲正弦		
sec	正割	acosh	反双曲余弦		

表 1-4　指数函数

名称	含　义	名称	含　义	名称	含　义
exp	E 为底的指数	log10	10 为底的对数	pow2	2 的幂
log	自然对数	log2	2 为底的对数	sqrt	平方根

表 1-5　复数函数

名称	含　义	名称	含　义	名称	含　义
abs	绝对值	conj	复数共轭	real	复数实部
angle	相角	imag	复数虚部		

表 1-6　圆整函数和求余函数

名称	含　义	名称	含　义
ceil	向＋∞圆整	rem	求余数
fix	向 0 圆整	round	向靠近整数圆整
floor	向－∞圆整	sign	符号函数
mod	模除求余		

表 1-7　矩阵变换函数

名称	含　义	名称	含　义
fiplr	矩阵左右翻转	diag	产生或提取对角阵
fipud	矩阵上下翻转	tril	产生下三角
fipdim	矩阵特定维翻转	triu	产生上三角
Rot90	矩阵反时针 90°翻转		

表 1-8　其他函数

名　称	含　义	名　称	含　义
min	最小值	max	最大值
mean	平均值	median	中位数
std	标准差	diff	相邻元素的差
sort	排序	length	个数
norm	欧氏(Euclidean)长度	sum	总和
prod	总乘积	dot	内积
cumsum	累计元素总和	cumprod	累计元素总乘积
cross	外积		

1.2.3　矩阵运算

1.2.3.1　矩阵的创建与操作

1. 矩阵的创建

(1) 直接输入法

任何矩阵(向量)都可以直接按行方式输入每个元素：同一行中的元素用逗号(,)或者用空格符来分隔,且空格个数不限;不同的行用分号(;)分隔。所有元素处于一对方括号([ ])内。

【例 1-1】 按行方式输入元素。

```
>>a=[1,2,3]                                %输入完这一行,按回车键可得下面的结果
a=
     1     2     3
```

(2) 由小矩阵建立大矩阵

大矩阵可由方括号中的小矩阵建立起来。

【例 1-2】 利用小矩阵建立大矩阵。

```
>>A=[1  2  3; 4  5  6 ; 7  8  9]           %可利用它建立一个大矩阵
>>B=[A  A^2; A+32  2*A]
A=
     1     2     3
     4     5     6
     7     8     9
B=
     1     2     3    30    36    42
     4     5     6    66    81    96
     7     8     9   102   126   150
    33    34    35     2     4     6
    36    37    38     8    10    12
    39    40    41    14    16    18
```

注意：在同一行的各个小矩阵要有相同的行数，在同一列上的小矩阵要有相同的列数。

(3) 利用冒号建立一个矩阵

【例 1-3】 产生一维矩阵。

```
>>x=[1:5]                                    %产生一个 1 到 5 单位增量的一维矩阵
x=
    1    2    3    4    5
```

(4) 通过函数建立

MATLAB 提供了一批产生矩阵的函数，如表 1-9 所示。

表 1-9 MATLAB 提供的产生矩阵的函数

zeros	产生一个零矩阵	diag	产生一个对角矩阵
ones	生成全 1 矩阵	tril	取一个矩阵的下三角
eye	生成单位矩阵	triu	取一个矩阵的上三角
magic	生成魔术方阵	pascal	生成 PASCAL 矩阵

例如：

```
ones(3)
ans=
    1    1    1
    1    1    1
    1    1    1
eye(3)
ans=
    1    0    0
    0    1    0
    0    0    1
```

除了以上产生标准矩阵的函数外，MATLAB 还提供了产生随机(向量)矩阵的函数 rand 和 randn，及产生均匀级数的函数 linspace、产生对数级数的函数 logspace 和产生网格的函数 meshgrid 等. 详细使用请查阅随机文档。

2. 矩阵行列删除

【例 1-4】 建一矩阵，删除其某行某列。

```
>>a=[1 2 3;5 10 15;7 8 19;0 20 0]
a=
    1    2    3
    5   10   15
    7    8   19
    0   20    0
>>a(2,:)=[]                                  %删除指定行，冒号表示所有行或列
a=
    1    2    3
```

```
    7    8   19
    0   20    0
>>a(:,3)=[]                                    %删除指定列
a=
    1    2
    7    8
    0   20
```

3. 矩阵连接

【例 1-5】 建两矩阵,对其连接。

```
>>a=[1  2;3  4];
>>b=[a a+5;a-5 zeros(size(a))]                 %用[ ]将小矩阵连接成大矩阵
b=
    1    2    6    7
    3    4    8    9
   -4   -3    0    0
   -2   -1    0    0
>>c=[a;5 10]
c=
    1    2
    3    4
    5   10
```

4. 矩阵操作

表 1-10 给出了矩阵操作的基本命令。

表 1-10　矩阵操作基本命令

命　令	含　义	命　令	含　义
diag	对角矩阵和矩阵的对角化	fliplr	矩阵左右翻转
rot90	矩阵旋转 90°	flipud	矩阵上下翻转
reshape	矩阵元素重新排列	cat	矩阵连接
tril	矩阵的下三角阵	repmat	复制并平铺矩阵
triu	矩阵的上三角阵		

例如,函数 fliplr (Flip matrix in the left/right direction)的功能是对矩阵进行左右旋转。如:

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

则 fliplr(x)为

```
    3    2    1
    6    5    4
```

1.2.3.2 矩阵的下标

(1) 全下标

行下标+列下标 ,如 a(i,j),几何概念清楚,引述简单。

(2) 单下标(linear index):矩阵元素在内存中是按“列”存储的。

(3) “一维编号”

把矩阵的所有列按先左后右的次序首尾相接排成“一维长列”后依次编号。

(4) 两种下标的转换关系

A 为(m×n)的二维数组(矩阵),其中某元素的“全下标”为(r,c),则相应的“单下标”为 k=(c-1)×m+r 。

1. 元素定位

单个的数组元素的位置可在括号中表达,如:

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

其中 a(3,3)=9 a(1,3)=3, a(3,1)=7,可用元素表达式进行运算和赋值产生新元素,如执行 a(3,3)=a(1,3)+a(3,1),则有

```
a=[1    2    3
   4    5    6
   7    8   10]
```

下标可以是一个一维数组。对于矩阵来说,利用下标可以调动某些元素构成新的子数组。设 b 是一个 10×10 阶数组,则 b(1:5,3)指 b 中的第 1 行到第 5 行处于第 3 列的元素组成 5×1 阶子数组。

b(1:5, 7:10)指前 5 行处于后四列中的元素构成 5×4 阶的子数组。

b(:, [3,5,10])=c(:, 1:3) 表示将 c 数组的前三列赋值给 b 数组的第三、第五和第十列。

a(:,n:-1:1) 即为由原来 a 数组中取 n 至 1 负增长的列元素组成一个新的数组,其行数为 a 数组的行数,列数为 n 。

2. 改变数组尺寸

【例 1-6】 将一个 2×3 阶的数组改变为 6×1 阶。

```
>>a=[1  2  3 ; 4  5  6] '
>>b=a(:)
a=
    1    4
    2    5
    3    6
b=
    1
    2
```

```
3
4
5
6
```

1.2.3.3　矩阵的算数运算

1. 加减运算

运算符为“＋”、“－”，两个运算对象必须是同阶矩阵，否则将给出错误信息。

【例 1-7】 加法运算。

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

运行结果给出错误信息：

```
???Error using==>±
Matrix dimensions must agree.
```

但标量，即 1×1 阶矩阵可以和其他不同维数的矩阵进行加减运算，如：a＋5

```
ans=
    6    7    8
    9   10   11
   12   13   14
```

2. 乘除运算

矩阵在进行乘除运算时与通常的运算符号相同(＊，/，\)。

(1) 矩阵乘法

条件：两矩阵中前一矩阵的列数与后一矩阵的行数相同。

【例 1-8】 乘法运算。

```
>>x=[1  2; 3  4];
>>y=[5  6; 7  8];
>>x*y
ans=
    19    22
    43    50
```

(2) 矩阵除法

条件：a 矩阵是非奇异方阵，则 a\b(左除)和 b/a(右除)都可以实现。a\b 等效于 a 矩阵的逆左乘 b 矩阵，即 a\b＝inv(a)＊b，b/a 等效于 a 矩阵的逆右乘 b 矩阵，即 b/a＝b＊inv(a)。

通常 x＝a\b 是 a＊x＝b 的解，x＝b/a 是 x＊a＝b 的解。一般 a\ b≠b/a，右除与左除的关系为(b/a)'＝(a'\b')。

3. 矩阵的乘方

条件：在 a^p 中 a，p 不可都是矩阵，必须一个是标量，一个是方阵。a^p 意思是 a 的 p

次方。

若 a 是一个方阵，p 是一个标量，且 p 是大于 1 的整数，则 a 的 p 次幂即为 a 自乘 p 次。

当 p 是不为整数的标量时，a^p＝V * D.^p/V 其中 D 为矩阵 a 的特征值矩阵，V 为对应的特征矢量阵，可用 eig 函数求出 D 和 V，[V,D]＝eig(a)。

当 p 是方阵而 a 是标量时，a^p＝V * a.^D/V，其中[V,D]＝eig(p)。

4. 矩阵的行列式

命令格式：det(A)。

【例 1-9】 求矩阵的行列式。

```
>>det([1,2;3,4])
ans=
    -2
```

5. 矩阵的逆矩阵

命令格式：inv(A)。

【例 1-10】 求矩阵的逆矩阵。

```
>>inv([1,2;3,4])
ans=
    -2.0000    1.0000
     1.5000   -0.5000
```

6. 矩阵的迹

命令格式：trace(A)

【例 1-11】 求矩阵的迹。

```
>>trace([1,2;-1,3])
ans=
    4
```

7. 矩阵的秩

命令格式：rank(A)

【例 1-12】 求矩阵的秩。

```
>>A=[2 1 -1;2 1 2;1 -1 1];
>>r=rank(A)
r=3
```

表 1-11 给出了矩阵算术运算表达式和对应的 M 函数。

1.2.3.4 矩阵的逻辑运算

1. 关系运算

条件：对于两个矩阵的关系运算，两边的矩阵必须具有同样尺寸。

关系运算符：

＜(小于)、＜＝(小于等于)、＞(大于)、＞＝(大于等于)、＝＝(等于)、～＝(不等于)。

表 1-11　算术运算和对应的 M 函数

功　能	算术表达式	M 函 数	功　能	算术表达式	M 函 数
加法	a+b	plus(a,b)	矩阵左除	a\b	mldivide(a,b)
减法	a−b	minus(a,b)	数组左除	a.\b	ldivide(a,b)
矩阵乘法	a * b	mtimes(a,b)	矩阵求幂	a^b	mpower(a,b)
数组乘法	a. * b	times(a,b)	数组求幂	a.^b	power(a,b)
矩阵右除	a/b	mrdivide(a,b)	矩阵共轭转置	a′	ctranspose(a)
数组右除	a./b	rdivide(a,b)	非共轭转置	a.′	transpose(a)

【例 1-13】 标量。

```
>>2+2~=4
ans=
     0
```

2. 逻辑运算

逻辑运算符：

&(与，AND)，|(或，OR)，～(非，NOT)

条件：对于两个矩阵的逻辑运算，两边的矩阵必须具有同样尺寸。

“～”是一元算符，当 a 为零时，返回信息为 1；为非零时，返回信息为 0。# p|(～p) 返回值为 1，# p&(～p) 返回值为 0。

【例 1-14】 建立两矩阵，并对其进行逻辑运算。

```
>>a=[1  2  3; 4  5  6];
>>b=[-1  0  0; 0  0.5  0];
>>a&b
ans=
    1    0    0
    0    1    0
>>a|b
ans=
     1    1    1
     1    1    1
>>~b
ans=
    0    1    1
    1    0    1
```

1.3　MATLAB 程序设计

本节将重点介绍 MATLAB 的程序设计。程序设计是一门语言的灵魂，是人类智慧和能力的具体体现，程序设计是分析问题与解决问题的过程。MATLAB 作为一种高级语言，

和其他高级语言一样可以进行控制流的程序设计，从而完成一定的功能，实现一定的算法。假如读者想灵活运用 MATLAB 去解决实际问题，想充分调用 MATLAB 科学技术资源，就必须掌握好程序设计的方法与技巧。

1.3.1 M 文件

前面介绍在 MATLAB 中所做的运算，适合于所要计算的算式不太长或是想以交谈式方式做运算，如果要计算的算式很长，有数十行或是须要一再执行的算式，则那样的方式就行不通了。MATLAB 提供了所谓的 M-file 的方式，可让使用者自行将指令及算式写成程序然后储存成一个特别的文档，其扩展名是 m，譬如 pic. m，其中的 pic 就是文件名称。

1.3.1.1 M 文件的建立和编辑

建立 M 文件的一般步骤如下：

(1) 打开文件编辑器

最简单的方法是在操作桌面的工具栏上选择新建文件夹或打开已有文件夹，也可以在命令窗口输入命令 edit 建立新文件或输入 edit filename，打开名为 filename 的 M 文件；

(2) 编写程序内容

编写新的文件或修改已有文件；

(3) 保存文件

文件运行前必须完成保存操作，与一般的文件编辑保存操作相同；

(4) 运行文件

在命令窗口输入文件名即可运行。如要在编辑器中直接完成运行，可在编辑器的 Debug 菜单下单击 save and run 选项，或按 Run 快捷键，最快捷的方法是直接按 F5 键执行运行。

【例 1-15】 简单小程序。

① 单击 MATLAB 指令窗工具栏上的 New File 图标，就可打开如图 1-1 所示的 MATLAB 文件编辑调试器。用户即可在空白窗口中编写程序。

② 输入图 1-1 所示程序（pic. m）

图 1-1 程序文件

③ 单击编辑调试器工具栏图标，在弹出的 Windows 标准风格的“保存为”对话框中，选择保存文件夹，输入新编文件名(如 pic)，单击“保存”按钮，就完成了文件保存。

④使 pic. m 所在目录成为当前目录(系统默认路径)，或让该目录处在 MATLAB 的搜

索路径上。然后在指令窗口运行图 1-1 中的指令，便可得到如图 1-2 所示的图形。

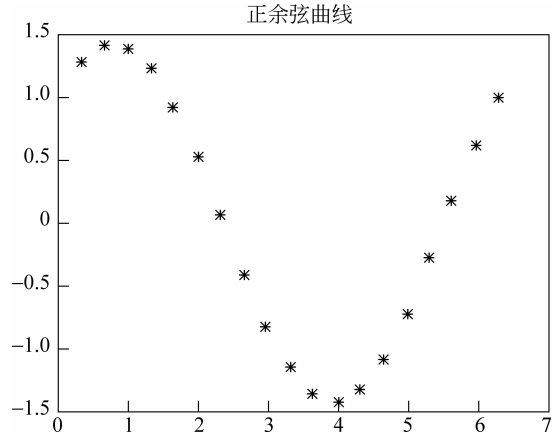

图 1-2　程序运行结果

1.3.1.2　M 文件的分类

M 文件有两种形式，即命令文件(Script File)和函数文件(Function File)。命令文件通常用于执行一系列简单的 MATLAB 命令，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。和命令文件不同，函数文件可以接受参数，也可以返回参数，在一般情况下用户不能单独输入函数文件名来运行函数文件，而必须由其他语句来调用。MATLAB 的大多数应用程序都是以函数文件的形式给出的，如求矩阵特征多项式的函数 ploy( )。

1. 命令文件

命令文件是 M 文件中最简单的一种，是可用于自动重复执行的一组 MATLAB 命令和函数组合，不需输出输入参数，用 M 文件可以调用工作空间已有的变量或创建新的变量。运行过程中产生的变量都是全局变量。

建立一个命令文件等价于从命令窗口中顺序输入文件里的命令，程序不需要预先定义，只要依次将命令编辑在命令文件中，再将程序保存成为扩展名为.m 的 M 文件即可。

运行命令文件时，只需在命令窗口输入文件名即可。

2. 函数文件

一个函数 M 文件与脚本文件类似之处在于它们都是一个有.m 扩展名的文本文件。如同脚本 M 文件一样，函数 M 文件不进入命令窗口，而是由文本编辑器所创建的外部文本文件。一个函数的 M 文件与脚本文件在通信方面是不同的。函数与 MATLAB 工作空间之间的通信，只通过传递给它的变量和通过它所创建的输出变量。在函数内中间变量不出现在 MATLAB 工作空间，或与 MATLAB 工作空间不交互。一个函数的 M 文件的第一行把 M 文件定义为一个函数，并指定它的名字。它与文件名相同，但没有.m 扩展名；同时也定义了它的输入和输出变量。

M 文件函数必须遵循以下特定的规则：

(1) 函数名和文件名必须相同，例如函数 funexam 存储在名为 funexam.m 文件中。

(2) MATLAB 头一次执行一个 M 文件函数时,它打开相应的文本文件并将命令编辑成存储器的内部表示,以加速执行以后所有的调用。如果函数包含了对其他 M 文件函数的引用,它们也同样被编译到存储器。普通的脚本 M 文件不被编译,即使它们是从函数 M 文件内调用;打开脚本 M 文件,调用一次就逐行进行注释。

(3) 在函数 M 文件中,到第一个非注释行为止的注释行是帮助文本。当需要帮助时,返回该文本。

(4) 第一行帮助行,名为 H1 行,是由 lookfor 命令搜索的行。

(5) 函数可以有零个或更多个输入参量。函数可以有零个或更多个输出参量。

(6) 函数可以按少于函数 M 文件中所规定的输入和输出变量进行调用,但不能用多于函数 M 文件中所规定的输入和输出变量数目。如果输入和输出变量数目多于函数 M 文件中 function 语句一开始所规定的数目,则调用时自动返回一个错误。

(7) 当函数有一个以上输出变量时,输出变量包含在括号内。例如,[V,D] = eig(A)。不要把这个句法与等号右边的[V,D] 相混淆。右边的[V,D] 是由数组 V 和 D 所组成的。

(8) 当调用一个函数时,所用的输入和输出的参量的数目,在函数内是规定好的。函数工作空间变量 nargin 包含输入参量个数;函数工作空间变量 nargout 包含输出参量个数。事实上,这些变量常用来设置默认输入变量,并决定用户所希望的输出变量。

(9) 当一个函数说明一个或多个输出变量,但没有要求输出时,就简单地不给输出变量赋任何值。

(10) 函数有它们自己的专用工作空间,它与 MATLAB 的工作空间分开。函数内变量与 MATLAB 工作空间之间唯一的联系是函数的输入和输出变量。如果函数任一输入变量值发生变化,其变化仅在函数内出现,不影响 MATLAB 工作空间的变量。那么函数内所创建的变量只驻留在函数的工作空间,而且只在函数执行期间临时存在,以后就消失。因此,从一个调用到下一个调用,在函数工作空间变量存储信息是不可能的。

(11) 如果一个预定的变量(例如 pi)在 MATLAB 工作空间重新定义,它不会延伸到函数的工作空间。逆向有同样的属性,即函数内的重新定义变量不会延伸到 MATLAB 的工作空间中。

(12) 当调用一个函数时,输入变量不会复制到函数的工作空间,但使它们的值在函数内可读。然而,改变输入变量内的任何值,数组就复制到函数工作空间。进而,按默认,如果输出变量与输入变量相同,例如,函数 x=fun(x, y, z) 中的 x ,那么就将它复制到函数的工作空间。因此,为了节约存储和增加速度,最好是从大数组中抽取元素,然后对它们作修正,而不是将整个数组复制到函数的工作空间。

(13) 如果变量说明是全局的,则函数可以与其他函数、MATLAB 工作空间和递归调用本身共享变量。为了在函数内或 MATLAB 工作空间中访问全局变量,在每一个所希望的工作空间,变量必须说明是全局的。全局变量使用的例子可以在 MATLAB 函数 tic 和 toc 中看到,它们合在一起工作如一个跑表。

(14) 从函数 M 文件内可以调用脚本文件。在这种情况下,脚本文件查看函数工作空间,不查看 MATLAB 工作空间。从函数 M 文件内调用的脚本文件不必用调用函数编译到内存。函数每调用一次,它们就被打开和解释。因此,从函数 M 文件内调用脚本文件减慢了函数的执行。

(15) 函数可以递归调用,即 M 文件函数能调用它们本身。

(16) 当函数 M 文件到达 M 文件终点,或者碰到返回命令 return,就结束执行和返回。return 命令提供了一种结束一个函数的简单方法,而不必到达文件的终点。

(17) MATLAB 函数 error 在命令窗口显示一个字符串,放弃函数执行,把控制权返回给键盘。这个函数对提示函数使用不当很有用。

总之,函数 M 文件提供了一个简单的扩展 MATLAB 功能的方法。事实上,MATLAB 本身的许多标准函数就是 M 文件函数。

1.3.2　控制流

计算机编程语言和可编程计算器提供许多功能,它允许你根据决策结构控制命令执行流程。控制流极其重要,因为它使过去的计算影响将来的运算。MATLAB 提供三种决策或控制流结构:for 循环,while 循环和 if-else-end 结构。由于这些结构经常包含大量的 MATLAB 命令,故经常出现在 M 文件中,而不是直接加在 MATLAB 提示符下。

1. for 循环

for 循环允许一组命令以固定的和预定的次数重复。for 循环的一般形式是:

```
for  x=array
    {commands}
end
```

在 for 和 end 语句之间的{commands}按数组中的每一列执行一次。在每一次迭代中,x 被指定为数组的下一列,即在第 n 次循环中,x=array(:,n)。

2. while 循环

与 for 循环以固定次数求一组命令的值相反,While 循环以不定的次数求一组语句的值。While 循环的一般形式是:

```
while expression
    {commands}
end
```

只要在表达式里的所有元素为真,就执行 while 和 end 语句之间的{commands}。通常,表达式的求值给出一个标量值,但数组值也同样有效。在数组情况下,所得到数组的所有元素必须都为真。

3. if-else-end 结构

很多情况下,命令的序列必须根据关系的检验有条件地执行。在编程语言里,这种逻辑由某种 If-Else-End 结构来提供。最简单的 If-Else-End 结构如下:

```
if  expression
    {commands}
end
```

如果在表达式中的所有元素为真(非零),那么就执行 if 和 end 语言之间的{commands}。在表达式包含有几个逻辑子表达式时,即使前一个子表达式决定了表达式的最后逻辑状态,仍要计算所有的子表达式。

除上述三种结构外，下面再介绍几种语句。

4. switch-case-end 语句

通过对某个变量值的比较做各种不同的执行选择。

形式：

```
switch 表达式(数字或字符串)
  case  表达式 1
    语句体 1
  case  表达式 2
    语句体 2
  ……
  otherwise
    语句体 n
end
```

5. try-catch 语句

try-catch 语句用于检测错误并改变流程，形式如下：

```
try
  语句体 1
catch
  语句体 2
end
```

先执行 try 下面的语句体 1，如没有错误，就跳出该结构；如出错误，则执行 catch 语句下面的语句体 2。可用 lasterr 函数查询错误信息，查询结果为空字符串时表示语句体 1 成功执行。

6. breake 和 continue 语句

与循环结构相关的语句有 break 语句和 continue 语句。它们一般与 if 语句配合使用。

Break 语句用于终止循环的执行。当在循环体内执行到该语句时，程序将跳出循环，继续执行循环语句的下一语句。

Continue 语句控制跳过循环体中的某些语句。当在循环体内执行到该语句时，程序将跳过循环体中所有剩下的语句，继续下一次循环。

1.3.3 变量和函数种类

1.3.3.1 函数变量及其作用域

在 MATLAB 语言中，变量可以分为输入变量、输出变量和函数内使用的变量。

输入变量相当于函数的入口数据，也是一个函数操作的主要对象，从某种意义上来说，函数的功能在于对输入变量进行一定的操作从而实现一定的功能。函数的输入变量为局部变量，函数对输入变量的一切操作和修改如果不依靠输出变量的话，将不会影响工作区间中该变量的值。

1. 局部变量

局部变量是在函数内部使用的变量，其影响范围只能在本函数内。每个函数在运行时，

都占有独立的函数工作空间，此工作空间和 MATLAB 的工作空间是相互独立的，局部变量仅存在于函数的工作空间内。当函数执行完毕之后，该变量即自行消失。

2. 全局变量

在 MATLAB 语言中，函数内部定义的变量都是局部变量，它们不被加载到工作区间中。当用户需要使用全局变量时要使用 global 函数来进行定义，而且在任何使用该全局变量的函数中都应加以定义，即使是在命令窗口也不例外。

3. 永久变量

除了通过全局变量共享数据外，函数式 M 文件还可以通过声明一个变量 persistent 来对函数中重复使用和递归调用的变量的访问进行限制，使用格式形如 persistent (X Y Z)。永久变量与全局变量类似，但是它的范围被限制在声明这些变量的函数内部，不允许在其他的函数中对它们进行改变。只要 M 文件还在 MATLAB 7 的内存中，永久变量就存在。

1.3.3.2　函数的分类

1. 主函数

M 文件中的第一个函数叫做主函数，前边章节中所引用的函数事实上都是主函数，主函数之后可以是任意数量的子函数，它们可以作为主程序的子程序。一般来说，在命令窗口或是其他的 M 文件中只能调用主函数，调用的时候就是直接调用其函数名。

比如，函数 average 的 M 文件 average. m 如下：

```
function y=average(x)
%AVERAGE Mean of vector elements
y=sum(x)/length(x);                    %Actual computation
```

2. 匿名函数

匿名函数提供了一种创建简单程序的方法，使用它用户可以不必每次都编写 M 文件。用户可以在 MATLAB 7 的命令窗口或是其他任意 M 文件和脚本文件中使用匿名函数。

匿名函数的格式如下：

```
fhandle=@(arglist) expr
```

3. 嵌套式函数

在 MATLAB 7 中，可以在一个函数的内部定义一个或多个其他的函数，这些在内部定义的函数被称作嵌套式函数。应当注意的是，在嵌套式函数的内部也可以定义嵌套式函数。

定义嵌套式函数时，只需在另一个 M 文件的内部定义该函数即可，同其他 M 文件一样，嵌套式函数包含有 M 文件的所有基本部分。

4. 子函数

与其他的高级语言一样，在 MATLAB 7 语言中也可以很方便地定义子函数，用来扩充函数的功能。在函数文件中题头定义的函数为主函数，而在函数体内定义的其他函数都被视为子函数。子函数只能为主函数或同一主函数下的其他子函数所使用。

5. 局部函数

MATLAB 7 中把放置在目录 private 下的函数称为局部函数，这些函数只有在 private 目录的父目录中的函数才可以调用，其他目录下的函数不能调用。

局部函数与子函数所不同的是，局部函数可以被其父目录下的所有函数所调用，而子函数则只能被其所在的 M 文件的主函数所调用。所以，局部函数在可用的范围上大于子函数；在函数编辑的结构上，局部函数与一般的函数文件的编辑相同，而子函数只能在主函数文件中编辑。

1.4 Simulink 基础

Simulink 中的“Simu”一词表示可用于计算机仿真，而“link”一词表示它能进行系统连接，即把一系列模块连接起来，构成复杂的系统模型。作为 MATLAB 的一个重要组成部分，Simulink 由于它所具有的上述的两大功能和特色，以及所提供的可视化仿真环境、快捷简便的操作方法，而使其成为目前最受欢迎的仿真软件。

本节主要介绍 Simulink 的基本功能和基本操作方法，并通过举例介绍如何利用 Simulink 进行系统建模和仿真。

1.4.1 Simulink 概述

在工程实际中，控制系统的结构往往很复杂，如果不借助专用的系统建模软件，则很难准确地把一个控制系统的复杂模型输入计算机，对其进行进一步的分析与仿真。

1990 年，Math Works 软件公司为 MATLAB 提供了新的控制系统模型图输入与仿真工具，并命名为 SIMULAB，该工具很快就在控制工程界获得了广泛的认可，使得仿真软件进入了模型化图形组态阶段。但因其名字与当时比较著名的软件 SIMULA 类似，所以 1992 年正式将该软件更名为 Simulink。

Simulink 的出现，给控制系统分析与设计带来了福音。为了更好地理解和掌握 Simulink，本节首先介绍计算机仿真技术和仿真建模方法的基本概念，以便对建模和仿真有个初步和整体的认识。

1.4.1.1 系统与模型

1. 系统

系统只指具有某些特定功能、相互联系、相互作用的元素的集合。这里的系统是指广义上的系统，泛指自然界的一切现象与过程。例如工程系统（如控制系统、通信系统等）和非工程系统（如股市系统、交通系统、生物系统等）。

2. 系统模型

系统模型是对实际系统的一种抽象，是对系统本质（或是系统的某种特性）的一种描述。模型具有与系统相似的特性。好的模型能够反映实际系统的主要特征和运动规律。

模型可以分为实体模型和数学模型。

实体模型又称物理效应模型，是根据系统之间的相似性而建立起来的物理模型，如建筑模型等。

数学模型包括原始系统数学模型和仿真系统数学模型。原始系统数学模型是对系统的原始数学描述。仿真系统数学模型是一种适合于在计算机上演算的模型，主要是指根据计算机的运算特点、仿真方式、计算方法、精度要求将原始系统数学模型转换为计算机程序。