

第3章 MATLAB 程序设计

MATLAB 语言仅靠一条一条地输入语句,难以实现复杂功能,为了实现诸如循环、条件和分支等功能,就要像其他计算机语言一样进行程序设计。MATLAB 语言的程序设计,则利用了 M 文件,而 M 文件是由一系列的 MATLAB 语句组成的。

3.1 MATLAB 的 M 文件

因为 MATLAB 本身可以被认为是一高效的语言,所以用它可编写出具有特殊意义的磁盘文件来,这些磁盘文件由一系列的 MATLAB 语句组成,它既可能是由一系列窗口命令语句构成的文本文件(也称为脚本文件,简称为 MATLAB 的程序),又可以是由各种控制语句和说明语句构成的函数文件(简称为 MATLAB 的函数)。由于它们都是由 ASCII 码构成的,其扩展名均为“.m”,故统称为 M 文件。

由于 M 文件具有普通的文本格式,因而可以用任何编辑器建立和编辑。但一般最常用、而且最为方便地是使用 MATLAB 自带的编辑器,即利用 MATLAB 操作界面中的菜单命令 File→New→M-File 或 File→Open 打开的 M 文件编辑窗口对 M 文件进行建立和编辑。MATLAB 为了进一步方便用户对 M 文件的建立和编辑,在窗口中也设置了快捷工具“”和“”。

3.1.1 文本文件

文本文件(脚本文件)由一系列的 MATLAB 语句组成,它类似于 DOS 下的批处理文件,在 MATLAB 命令窗口的提示符下直接键入文本文件名,回车后便可自动执行文本文件中的一系列命令,直至给出最终结果。文本文件在工作空间中运算的变量为全局变量。

例 3-1 利用 MATLAB 的文本文件,求以下方程

$$\begin{cases} y_1 = 3x_1 + x_2 + x_3 \\ y_2 = 3x_1 - x_2 - x_3 \end{cases}$$

在 $x_1=-2, x_2=3, x_3=1$ 时的值。

解 ① 首先在 MATLAB 6.x/7.x 的操作界面中,利用菜单命令 File→New→M-File,打开 M 文件编辑器,然后在编辑器中根据例中所给方程编写以下文本文件,并以 ex3_1_1 为文件名进行保存(后缀.m 自动追加)。

```
%ex3_1_1.m
x1=-2;x2=3;x3=1;
y1=3*x1+x2+x3
y2=3*x1-x2-x3
```

其中,带%的语句为说明语句,不被 MATLAB 所执行,它可以在命令窗口中利用 help ex3_1_1 命令来显示%后的内容。

② 当以上文本文件 ex3_1_1.m 建立后,在 MATLAB 命令窗口中输入

```
>>ex3_1_1
```

回车后显示：

```
y1=
    -2
y2=
   -10
```

由于文本文件中的变量为全局变量，故以上变量 x_1 , x_2 , x_3 的值，也可在文本文件外先给定，此时的文本文件 ex3_1_2.m 为

```
%ex3_1_2.m
y1=3*x1+x2+x3
y2=3*x1-x2-x3
```

当以上文本文件 ex3_1_2.m 建立后，利用以下命令，同样可以得到以上结果。

```
>>x1=-2;x2=3;x3=1;ex3_1_2
```

对于 MATLAB 8.x/9.x，则利用其主页 (HOME) 中新建 (New) 菜单下的文本 (Script) 命令或主页 (HOME) 中新建文本 (New Script) 快捷工具 “” 打开 M 文本文件编辑窗口。

以上两种方式下，文本文件中变量的值都被保存下来，这与下面的函数文件是不同的。

3.1.2 函数文件

函数文件的功能是建立一个函数，且这个函数可以同 MATLAB 的库函数一样使用，它与文本文件不同，在一般情况下不能单独键入函数文件的文件名来运行一个函数文件，它必须由其他语句来调用，函数文件允许有多个输入参数和多个输出参数值。其基本格式如下

function [f1,f2,f3,...]=fun(x,y,z,...)

注释说明语句

函数体语句

其中， $x,y,z,\dots$ 是形式输入参数；而 $f1,f2,f3,\dots$ 是返回的形式输出参数值；**fun** 是函数名。

实际上，函数名一般就是这个函数文件的磁盘文件名，注释语句段的内容同样可用 **help** 命令显示出来。

调用一个函数文件只需直接使用与这个函数一致的格式

[y1,y2,y3,...]=fun(a,b,c,...)

其中， $a,b,c,\dots$ 是相应的实际输入参数；而 $y1,y2,y3,\dots$ 是相应的实际输出参数值。

例 3-2 利用 MATLAB 的函数文件，求以下方程

$$\begin{cases} y_1 = 3x_1 + x_2 + x_3 \\ y_2 = 3x_1 - x_2 - x_3 \end{cases}$$

在 $x_1 = -2, x_2 = 3, x_3 = 1$ 时的值。

解 ① 由于函数文件的建立与文本文件完全一样，故与例 3-1 一样首先根据例中所给方程在 MATLAB 的 M 文件编辑器下，建立以下函数文件 ex3_2.m

```
%ex3_2.m
function [b1,b2]=ex3_2(a1,a2,a3)
b1=3*a1+a2+a3;
b2=3*a1-a2-a3;
```

② 当以上函数文件 ex3_2.m 建立后，在 MATLAB 命令窗口中输入以下命令

```
>>x1=-2;x2=3;x3=1;[y1,y2]=ex3_2(x1,x2,x3)
```

结果显示:

```
y1=
    -2
y2=
   -10
```

对于 MATLAB 8.x/9.x, 则利用其主页 (HOME) 中新建 (New) 菜单下的函数 (function) 命令, 打开 M 函数文件编辑窗口, 它与 MATLAB 的 M 文本文件编辑窗口略有不同, 其区别在于已经设置其第一行由 `function` 开头, 最后一行由 `end` 结尾的标准函数文件格式 (如同 MATLAB 6.x/7.x 一样, 这里的 `end` 也可删除不要)。

函数文件中定义的变量为局部变量, 也就是说它只在函数内有效。即在该函数返回后, 这些变量会自动在 MATLAB 工作空间中清除掉, 这与文本文件是不同的, 但可通过命令

`global <变量>`

来定义一个全局变量。

函数文件与文本文件另一个区别在于其第一行由 `function` 开头, 且有函数名和输入形式参数与输出形式参数, 没有这一行的磁盘文件就是文本文件。

由上可知, MATLAB 实际上可以认为是一种解释性语言, 用户可以在 MATLAB 工作环境下一条一条地键入命令, 也可以直接键入用 MATLAB 的语言编写的 M 文件名, 或将它们结合起来使用, 这样 MATLAB 语言对此命令或 M 文件中各条命令进行翻译, 然后在 MATLAB 环境下对它进行处理, 最后返回运算结果。所以说 MATLAB 语言的一般结构为:

窗口命令 + M 文件

3.2 MATLAB 的程序结构

MATLAB 是一个功能极强的高度集成化程序设计语言, 它具备一般程序设计语言的基本语句结构, 并且它的功能更强, 由它编写出来的程序结构简单, 可读性强。同其他高级语言一样, MATLAB 也提供了条件转移语句、循环语句和一些常用的控制语句, 从而使得 MATLAB 语言的编程显得十分灵活。

3.2.1 循环语句

在实际计算中, 经常会遇到许多有规律的重复计算, 此时就要根据循环条件对某些语句重复执行。MATLAB 中包括两种循环语句: `for` 语句和 `while` 语句。

1. `for` 语句的基本格式

在 MATLAB 中, `for` 语句的基本命令格式为:

```
for 循环变量=表达式 1: 表达式 3: 表达式 2
    循环语句组
end
```

在 MATLAB 的循环语句基本格式中, 循环变量可以取作任何 MATLAB 变量, 表达式 1、表达式 2 和表达式 3 的定义和 C 语言相似, 即首先将循环变量的初值赋成表达式 1 的值, 然后判断循环变量的值, 如果此时循环变量的值介于表达式 1 和表达式 2 的值之间, 则执行循环体中的语句, 否则结束循环语句的执行。执行完一次循环体中的语句之后, 则会将循环变量自增一个表达式 3 的值, 然后再判断循环变量是否介于表达式 1 和表达式 2 之间, 如果满足仍再执行循环体直至不满足为止, 这时将结束循环语句的执行, 而继续执行后面的语句。如果表达式

3 的值为 1，则可省略表达式 3。

例 3-3 求 $\sum_{i=1}^{100} i$ 的值。

解 MATLAB 程序 ex3_3_1.m 如下

```
%ex3_3_1.m
mysum=0;
for i=1:100
    mysum=mysum+i;
end
mysum
```

根据以上方法编写 MATLAB 的程序文件 ex3_3_1.m，其运行结果显示：

```
mysum =
      5050
```

实际编程中，在 MATLAB 下采用循环语句会降低其执行速度，所以上面的程序可以由下面的命令来代替，以提高运行速度。

```
>>i=1:100; mysum=sum(i)
```

其中，sum()为内部函数，其作用是求出 i 向量的各个元素之和。

2. while 语句的基本结构

在 MATLAB 中，while 语句的基本命令格式为：

```
while （条件式）
    循环体条件组
end
```

其执行方式为，若条件式中的条件成立，则执行循环体的内容，执行后再判断表达式是否仍然成立，如果表达式不成立，则跳出循环，向下继续执行。

例如对于上面的例 3-3，如果改用 while 循环语句，则可以写出下面的程序

```
%ex3_3_2.m
mysum=0;i=1;
while (i<=100)
    mysum = mysum +i; i=i+1;
end
mysum
```

MATLAB 提供的循环语句 for 和 while 是允许多级嵌套的，而且它们之间也允许相互嵌套，这和 C 语言等高级程序设计语言是一致的。

3.2.2 控制语句

在程序设计语言中，经常会遇到提前终止循环、跳出子程序、显示执行过程等，此时就要用到以下控制程序流命令。

（1）echo 命令

一般来说当一个 M 文件运行时，文件中的命令不在屏幕上显示出来，而利用 echo 命令可以使 M 文件在运行时把其中的命令显示在工作空间中，这对于调试、演示等很有用。其命令

格式为

echo on	%显示其后所有执行的命令文件的指令;
echo off	%不显示其后所有执行的命令文件的指令;
echo	%在上述两种情况下进行切换;
echo filename on	%显示由 filename 指定的 M 文件的执行命令;
echo filename off	%不显示由 filename 指定的 M 文件的执行命令;
echo on all	%显示其后的所有 M 文件的执行命令;
echo off all	%不显示其后的所有 M 文件的执行命令。

(2) break 命令

在 MATLAB 中, break 命令经常与 for 或 while 等语句一起使用, 其作用就是终止本次循环, 跳出最内层的循环。使用 break 命令可以不必等到循环的自然结束, 而是根据条件, 遇到 break 命令后强行退出循环过程。

(3) continue 命令

在 MATLAB 中, continue 命令也经常与 for 或 while 等语句一起使用, 其作用是结束本次循环, 即跳过循环体下面尚未执行的命令, 接着进行下一次是否执行循环的判断。

(4) pause 命令

pause()命令使用户暂停运行程序, 当按任一键时再恢复执行。其中 pause(n)中的 n 为等待的秒数。

(5) return 命令

return 命令能使当前正在运行的函数正常退出, 并返回调用它的函数, 继续运行。

3.2.3 转移语句

在程序设计中, 经常要根据一定的条件来执行不同的命令。当某些条件满足时, 只执行其中的某个命令或某些命令。在 MATLAB 中, 条件转移语句包括: if-else-end 语句和 switch-case-otherwise 语句。

1. if-else-end 语句的基本格式

在 MATLAB 中, 最简单的条件结构: if-end 语句命令格式为

```
if expression
    statements
end
```

当给出的条件式 expression 成立时, 则执行该条件块结构中的语句内容 statements, 执行完之后继续向下执行, 若条件不成立, 则跳出条件块而直接向下执行。

例 3-4 求满足 $\sum_{i=1}^m i > 1000$ 的最小 m 值。

解 MATLAB 程序 ex3_4.m 如下。

```
%ex3_4.m
mysum=0;
for m=1:1000
    mysum=mysum+m;
    if (mysum>1000) break; end
end
m
```

运行结果显示：

```
m =
    45
```

MATLAB 还提供了其他两种条件结构：if-else-end 格式和 if-else if-end 格式，这两种格式的调用方法分别为

if expression	和	if expression1
statements1		statements1
else		else if expression2
statements2		statements2
end		else if expression3
		statements3
		⋮
		end

例 3-5 如果想对一个变量 x 自动赋值。当从键盘输入 y 或 Y 时(表示是)， x 自动赋为 1 值；当从键盘输入 n 或 N 时(表示否)， x 自动赋为 0 值；输入其他字符时终止程序。

解 要实现这样的功能，可由下列的 while 循环程序来执行。

```
%ex3_5.m
ikey=0;
while(ikey==0)
    s1=input('若给 x 赋值请输入[y/n]? ','s');
    if(s1=='y'|s1=='Y')
        ikey=1; x=1
    else if (s1=='n'|s1=='N') ikey=1; x=0; end
        break
    end
end
end
```

2. switch-case-otherwise 语句的基本格式

MATLAB 中 switch-case-otherwise 语句的调用格式为

```
switch switch-expression
case case-expression1
    statements1;
case case-expression2
    statements2;
case case-expression3
    statements3;
    ⋮
otherwise
    statementsn;
end
```

switch-case-otherwise 语句中，switch-expression 给出了开关条件，当有 case-expression 与之匹配时，就执行其后的语句；如果没有 case-expression 与之匹配，就执行 otherwise 后面的语句。在执行过程中，只有一个 case 命令被执行。当执行完命令后，程序就跳出分支结构，执

行 end 后面的命令。

例如对于以下 MATLAB 函数文件 myfun.m。

```
function f=myfun(n)
switch n
case 0
    f=1;
case 1
    f=2;
otherwise
    f=8;
end
```

在 MATLAB 命令窗口输入以下命令

```
>>y=myfun(1)
```

结果显示：

```
y =
    2
```

小 结

本章主要介绍了在 MATLAB 中进行程序设计的基础内容，主要包括 MATLAB 的 M 文件和程序结构。通过本章学习应重点掌握以下内容：

- (1) MATLAB 的 M 文件的建立与使用；
- (2) MATLAB 的文本文件与函数文件的区别与特点；
- (3) MATLAB 的循环语句；
- (4) MATLAB 的控制语句；
- (5) MATLAB 的转移语句。

习 题

3-1 利用 MATLAB 文本文件和函数文件两种方式求函数

$$y = \sqrt{|x|} + x^3$$

在 $x=-4$ 时的值。

3-2 利用 MATLAB 的循环语句求函数

$$y = \sqrt{|x|} + x^3$$

在 $x=-2$, $x=-3$ 和 $x=4$ 时的值。

3-3 利用 MATLAB 的控制语句判断函数

$$y = \sqrt{|x|} + x^3$$

在 $x=-2$, $x=-3$ 和 $x=4$ 时的最小值和最大值。

本章习题答案可扫右边二维码。

