

第3章 MATLAB程序设计

3.1 M文件

3.2 程序控制结构

3.3 函数文件

3.4 程序举例

3.5 程序调试

3.1 M文件

3.1.1 M文件概述

用MATLAB语言编写的程序，称为M文件。M文件可以根据调用方式的不同分为两类：命令文件(Script File)和函数文件(Function File)。

例3-1 分别建立命令文件和函数文件，将华氏温度f转换为摄氏温度c。

程序1:

首先建立命令文件并以文件名f2c.m存盘。

```
clear;           %清除工作空间中的变量  
f=input('Input Fahrenheit temperature: ');  
c=5*(f-32)/9
```

然后在MATLAB的命令窗口中输入f2c，将会执行该命令文件，执行情况为：

```
Input Fahrenheit temperature: 73
```

```
c =
```

```
22.7778
```

程序2:

首先建立函数文件f2c.m。

```
function c=f2c(f)
```

```
c=5*(f-32)/9
```

然后在MATLAB的命令窗口调用该函数文件。

```
clear;
```

```
y=input('Input Fahrenheit temperature: ');
```

```
x=f2c(y)
```

输出情况为:

```
Input Fahrenheit temperature: 70
```

```
c =
```

```
21.1111
```

```
x =
```

```
21.1111
```

3.1.2 M文件的建立与打开

M文件是一个文本文件，它可以用任何编辑程序来建立和编辑，而一般常用且最为方便的是使用MATLAB提供的文本编辑器。

1. 建立新的M文件

为建立新的M文件，启动MATLAB文本编辑器有3种方法：

(1) 菜单操作。从MATLAB主窗口的File菜单中选择New菜单项，再选择M-file命令，屏幕上将出现MATLAB 文本编辑器窗口。

(2) 命令操作。在MATLAB命令窗口输入命令edit，启动MATLAB文本编辑器后，输入M文件的内容并存盘。

(3) 命令按钮操作。单击MATLAB主窗口工具栏上的New M-File命令按钮，启动MATLAB文本编辑器后，输入M文件的内容并存盘。

2. 打开已有的M文件

打开已有的M文件，也有3种方法：

(1) 菜单操作。从MATLAB主窗口的File菜单中选择Open命令，则屏幕出现Open对话框，在Open对话框中选中所需打开的M文件。在文档窗口可以对打开的M文件进行编辑修改，编辑完成后，将M文件存盘。

(2) 命令操作。在MATLAB命令窗口输入命令：`edit 文件名`，则打开指定的M文件。

(3) 命令按钮操作。单击MATLAB主窗口工具栏上的Open File命令按钮，再从弹出的对话框中选择所需打开的M文件。

3.2 程序控制结构

3.2.1 顺序结构

1. 数据的输入

从键盘输入数据，则可以使用input函数来进行，该函数的调用格式为：

A=input(提示信息, 选项);

其中提示信息为一个字符串，用于提示用户输入什么样的数据。

如果在input函数调用时采用's'选项，则允许用户输入一个字符串。例如，想输入一个人的姓名，可采用命令：

xm=input('What's your name?','s');

2. 数据的输出

MATLAB提供的命令窗口输出函数主要有**disp**函数，其调用格式为

disp(输出项)

其中输出项既可以为字符串，也可以为矩阵。

例3-2 输入x,y的值，并将它们的值互换后输出。

程序如下：

```
x=input('Input x please.');
```

```
y=input('Input y please.');
```

```
z=x;
```

```
x=y;
```

```
y=z;
```

```
disp(x);
```

```
disp(y);
```

例3-3 求一元二次方程 $ax^2 + bx + c = 0$ 的根。

程序如下：

```
a=input('a=?');
```

```
b=input('b=?');
```

```
c=input('c=?');
```

```
d=b*b-4*a*c;
```

```
x=[(-b+sqrt(d))/(2*a),(-b-sqrt(d))/(2*a)];
```

```
disp(['x1=',num2str(x(1)),',x2=',num2str(x(2))]);
```

3. 程序的暂停

暂停程序的执行可以使用`pause`函数，其调用格式为：

`pause(延迟秒数)`

如果省略延迟时间，直接使用`pause`，则将暂停程序，直到用户按任一键后程序继续执行。

若要强化中止程序的运行可使用`Ctrl+C`命令。

3.2.2 选择结构

1. if语句

在MATLAB中，if语句有3种格式。

(1) 单分支if语句：

```
if 条件  
    语句组  
end
```

当条件成立时，则执行语句组，执行完之后继续执行if语句的后继语句，若条件不成立，则直接执行if语句的后继语句。

(2) 双分支if语句:

```
if 条件
    语句组1
else
    语句组2
end
```

当条件成立时，执行语句组1，否则执行语句组2，语句组1或语句组2执行后，再执行if语句的后继语句。

例3-4 计算分段函数的值。

程序如下：

```
x=input('请输入x的值:');
```

```
if x<=0
```

```
    y= (x+sqrt(pi))/exp(2);
```

```
else
```

```
    y=log(x+sqrt(1+x*x))/2;
```

```
end
```

```
Y
```

(3) 多分支if语句:

if 条件1

语句组1

elseif 条件2

语句组2

.....

elseif 条件m

语句组m

else

语句组n

end

语句用于实现多分支选择结构。

例3-5 输入一个字符，若为大写字母，则输出其对应的小写字母；若为小写字母，则输出其对应的大写字母；若为数字字符则输出其对应的数值，若为其他字符则原样输出。

```
c=input('请输入一个字符','s');  
if c>='A' & c<='Z'  
    disp(setstr(abs(c)+abs('a')-abs('A')));  
elseif c>='a' & c<='z'  
    disp(setstr(abs(c)- abs('a')+abs('A')));  
elseif c>='0' & c<='9'  
    disp(abs(c)-abs('0'));  
else  
    disp(c);  
end
```


2. switch语句

switch语句根据表达式的取值不同，分别执行不同的语句，其语句格式为：

switch 表达式

case 表达式1

语句组1

case 表达式2

语句组2

.....

case 表达式m

语句组m

otherwise

语句组n

end

当表达式的值等于表达式1的值时，
执行语句组1，当表达式的值等于
表达式2的值时，执行语句组2，...，
当表达式的值等于表达式m的值时，
执行语句组m，当表达式的值不等
于case所列的表达式的值时，执行
语句组n。当任意一个分支的语句
执行完后，直接执行switch语句的
下一句。

例3-6 某商场对顾客所购买的商品实行打折销售，标准如下(商品价格用`price`来表示):

`price`<200 没有折扣

`200`≤`price`<`500` 3%折扣

`500`≤`price`<`1000` 5%折扣

`1000`≤`price`<`2500` 8%折扣

`2500`≤`price`<`5000` 10%折扣

`5000`≤`price` 14%折扣

输入所售商品的价格，求其实际销售价格。

程序如下：

```
price=input('请输入商品价格');  
switch fix(price/100)  
    case {0,1}          %价格小于200  
        rate=0;  
    case {2,3,4}        %价格大于等于200但小于500  
        rate=3/100;  
    case num2cell(5:9)  %价格大于等于500但小于1000  
        rate=5/100;  
    case num2cell(10:24) %价格大于等于1000但小于2500  
        rate=8/100;  
    case num2cell(25:49) %价格大于等于2500但小于5000  
        rate=10/100;  
    otherwise           %价格大于等于5000  
        rate=14/100;  
end  
price=price*(1-rate)    %输出商品实际销售价格
```

3. try语句

语句格式为：

try

语句组1

catch

语句组2

end

try语句先试探性执行语句组1，如果语句组1在执行过程中出现错误，则将错误信息赋给保留的**lasterr**变量，并转去执行语句组2。

例3-7 矩阵乘法运算要求两矩阵的维数相容，否则会出错。先求两矩阵的乘积，若出错，则自动转去求两矩阵的点乘。

程序如下：

```
A=[1,2,3;4,5,6]; B=[7,8,9;10,11,12];
```

```
try
```

```
    C=A*B;
```

```
catch
```

```
    C=A.*B;
```

```
end
```

```
C
```

```
lasterr           %显示出错原因
```

3.2.3 循环结构

1. for语句

for语句的格式为:

```
for 循环变量=表达式1:表达式2:表达式3  
    循环体语句
```

```
end
```

其中表达式1的值为循环变量的初值，表达式2的值为步长，表达式3的值为循环变量的终值。步长为1时，表达式2可以省略。

例3-8 一个三位整数各位数字的立方和等于该数本身则称该数为水仙花数。输出全部水仙花数。

程序如下：

```
for m=100:999  
m1=fix(m/100);           %求m的百位数字  
m2=rem(fix(m/10),10); %求m的十位数字  
m3=rem(m,10);           %求m的个位数字  
if m==m1*m1*m1+m2*m2*m2+m3*m3*m3  
disp(m)  
end  
end
```


例3-9 已知，当 $n=100$ 时，求 y 的值。

程序如下：

```
y=0;
```

```
n=100;
```

```
for i=1:n
```

```
    y=y+1/(2*i-1);
```

```
end
```

```
y
```

在实际MATLAB编程中，采用循环语句会降低其执行速度，所以前面的程序通常由下面的程序来代替：

```
n=100;
```

```
i=1:2:2*n-1;
```

```
y=sum(1./i);
```

```
y
```

for语句更一般的格式为：

for 循环变量=矩阵表达式
 循环体语句

end

执行过程是依次将矩阵的各列元素赋给循环变量，然后执行循环体语句，直至各列元素处理完毕。

例3-10 写出下列程序的执行结果。

s=0;

a=[12,13,14;15,16,17;18,19,20;21,22,23];

for k=a

s=s+k;

end

disp(s');

2. while语句

while语句的一般格式为：

```
while (条件)  
    循环体语句
```

```
end
```

其执行过程为：若条件成立，则执行循环体语句，执行后再判断条件是否成立，如果不成立则跳出循环。

例3-11 从键盘输入若干个数，当输入0时结束输入，求这些数的平均值和它们之和。

程序如下：

```
sum=0;
cnt=0;
val=input('Enter a number (end in 0):');
while (val~=0)
    sum=sum+val;
    cnt=cnt+1;
    val=input('Enter a number (end in 0):');
end
if (cnt > 0)
    sum
    mean=sum/cnt
end
```

3. break语句和continue语句

与循环结构相关的语句还有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。

例3-12 求[100, 200]之间第一个能被21整除的整数。

程序如下：

```
for n=100:200
if rem(n,21)~=0
    continue
end
break
end
n
```


4. 循环的嵌套

如果一个循环结构的循环体又包括一个循环结构，就称为循环的嵌套，或称为多重循环结构。

例3-13 若一个数等于它的各个真因子之和，则称该数为完数，如 $6=1+2+3$ ，所以6是完数。求[1,500]之间的全部完数。

```
for m=1:500
s=0;
for k=1:m/2
if rem(m,k)==0
s=s+k;
end
end
if m==s
disp(m);
end
end
```

3.3 函数文件

3.3.1 函数文件的基本结构

函数文件由**function**语句引导，其基本结构为：

function 输出形参表=函数名(输入形参表)

注释说明部分

函数体语句

其中以**function**开头的一行为引导行，表示该M文件是一个函数文件。函数名的命名规则与变量名相同。输入形参为函数的输入参数，输出形参为函数的输出参数。当输出形参多于一个时，则应该用方括号括起来。

例3-14 编写函数文件求半径为 r 的圆的面积和周长。

函数文件如下：

```
function [s,p]=fcircle(r)
```

```
%CIRCLE calculate the area and perimeter of a  
circle of radii r
```

```
%r      圆半径
```

```
%s      圆面积
```

```
%p      圆周长
```

```
%2004年7月30日编
```

```
s=pi*r*r;
```

```
p=2*pi*r;
```

3.3.2 函数调用

函数调用的一般格式是：

[输出实参表]=函数名(输入实参表)

要注意的是，函数调用时各实参出现的顺序、个数，应与函数定义时形参的顺序、个数一致，否则会出错。函数调用时，先将实参传递给相应的形参，从而实现参数传递，然后再执行函数的功能。

例3-15 利用函数文件，实现直角坐标(x,y)与极坐标(ρ, θ)之间的转换。

函数文件tran.m:

```
function [rho,theta]=tran(x,y)
```

```
rho=sqrt(x*x+y*y);
```

```
theta=atan(y/x);
```

调用tran.m的命令文件main1.m:

```
x=input('Please input x=:');
```

```
y=input('Please input y=:');
```

```
[rho,the]=tran(x,y);
```

```
rho
```

```
the
```

在MATLAB中，函数可以嵌套调用，即一个函数可以调用别的函数，甚至调用它自身。一个函数调用它自身称为函数的递归调用。

例3-16 利用函数的递归调用，求 $n!$ 。

$n!$ 本身就是以递归的形式定义的：

显然，求 $n!$ 需要求 $(n-1)!$ ，这时可采用递归调用。递归调用函数文件factor.m如下：

```
function f=factor(n)  
if n<=1  
    f=1;  
else  
    f=factor(n-1)*n;    %递归调用求(n-1)!  
end
```

3.3.3 函数参数的可调性

在调用函数时，MATLAB用两个永久变量nargin和nargout分别记录调用该函数时的输入实参和输出实参的个数。只要在函数文件中包含这两个变量，就可以准确地知道该函数文件被调用时的输入输出参数个数，从而决定函数如何处理。

例3-17 nargin用法示例。

函数文件examp.m:

```
function fout=chararray(a,b,c)
```

```
if nargin==1
```

```
    fout=a;
```

```
elseif nargin==2
```

```
    fout=a+b;
```

```
elseif nargin==3
```

```
    fout=(a*b*c)/2;
```

```
end
```

命令文件mydemo.m:

```
x=[1:3];
```

```
y=[1;2;3];
```

```
examp(x)
```

```
examp(x,y')
```

```
examp(x,y,3)
```


3.3.4 全局变量与局部变量

全局变量用global命令定义，格式为：

global 变量名

例3-18 全局变量应用示例。

先建立函数文件wadd.m，该函数将输入的参数加权相加。

```
function f=wadd(x,y)  
global ALPHA BETA  
f=ALPHA*x+BETA*y;
```

在命令窗口中输入：

```
global ALPHA BETA  
ALPHA=1;  
BETA=2;  
s=wadd(1,2)
```

3.4 程序举例

例3-19 猜数游戏。首先由计算机产生[1,100]之间的随机整数，然后由用户猜测所产生的随机数。根据用户猜测的情况给出不同提示，如猜测的数大于产生的数，则显示“High”，小于则显示“Low”，等于则显示“You won”，同时退出游戏。用户最多可以猜7次。

例3-20 用筛选法求某自然数范围内的全部素数。

素数是大于1，且除了1和它本身以外，不能被任何其他整数所整除的整数。用筛选法求素数的基本思想是：要找出2~m之间的全部素数，首先在2~m中划去2的倍数(不包括2)，然后划去3的倍数(不包括3)，由于4已被划去，再找5的倍数(不包括5)，...，直到再划去不超过的数的倍数，剩下的数都是素数。

例3-21 设，求 $s=$ 。

求函数 $f(x)$ 在 $[a,b]$ 上的定积分，其几何意义就是求曲线 $y=f(x)$ 与直线 $x=a$ ， $x=b$ ， $y=0$ 所围成的曲边梯形的面积。为了求得曲边梯形面积，先将积分区间 $[a,b]$ 分成 n 等分，每个区间的宽度为 $h=(b-a)/n$ ，对应地将曲边梯形分成 n 等分，每个小部分即是一个小曲边梯形。近似求出每个小曲边梯形面积，然后将 n 个小曲边梯形的面积加起来，就得到总面积，即定积分的近似值。近似地求每个小曲边梯形的面积，常用的方法有：矩形法、梯形法以及辛普生法等。

例3-22 Fibonacci数列定义如下：

$$f_1=1$$

$$f_2=1$$

$$f_n=f_{n-1}+f_{n-2} \quad (n>2)$$

求Fibonacci数列的第20项。

例3-23 根据矩阵指数的幂级数展开式求矩阵指数。

3.5 程序调试

3.5.1 程序调试概述

一般来说，应用程序的错误有两类，一类是语法错误，另一类是运行时的错误。语法错误包括词法或文法的错误，例如函数名的拼写错、表达式书写错等。

程序运行时的错误是指程序的运行结果有错误，这类错误也称为程序逻辑错误。

3.5.2 调试器

1. Debug菜单项

该菜单项用于程序调试，需要与Breakpoints菜单项配合使用。

2. Breakpoints菜单项

该菜单项共有6个菜单命令，前两个是用于在程序中设置和清除断点的，后4个是设置停止条件的，用于临时停止M文件的执行，并给用户一个检查局部变量的机会，相当于在M文件指定的行号前加入了一个keyboard命令。

3.5.3 调试命令

除了采用调试器调试程序外，**MATLAB**还提供了一些命令用于程序调试。命令的功能和调试器菜单命令类似，具体使用方法请读者查询**MATLAB**帮助文档。