

MATLAB 程序设计基础

MATLAB 程序设计既有传统高级语言的特征,又有自己独特的优点。在 MATLAB 程序设计中,充分利用 MATLAB 数据结构的特点,可以使程序结构简单、编程效率高。本章介绍有关 MATLAB 程序控制结构以及程序设计的基本方法。

3.1 脚本文件和函数文件

MATLAB 在执行命令时,是一种交互的命令执行方式,在命令窗口逐条输入指令,执行时 MATLAB 逐条解释执行。这种方式虽然简单、直观,但速度慢,执行过程不保留,当某些操作需要反复进行时,更使人感到不便。另一种 M 文件的程序执行方式,是将有关命令变成程序存储在一个文件中(M 文件),当需要运行时,直接调用运行,运行时 MATLAB 就自动依次执行该文件中的命令。



脚本文件和函数文件

M 文件可分为两大类: M 脚本文件和 M 函数文件,这两种 MATLAB 程序代码所编写的文件通常都是以 m 为扩展名,因此都统称为 M 文件。M 函数文件是 MATLAB 的主流。MATLAB 本身的一系列工具箱的内部函数就是 MATLAB 的开发者设计的一些 M 函数,提供给人们使用。

MATLAB 的脚本文件(script File)比较简单,当需要在命令窗口运行大量的命令时,直接从命令窗口输入比较麻烦,可以打开 M 文件编辑器,将这组命令存放在脚本文件中,运行时只要输入脚本文件名,MATLAB 就会自动执行该文件。打开 M 文件编辑器的方式因 MATLAB 的版本不同稍有区别,在 MATLAB R2011a 版本中,执行 File→new→script 命令,即可打开脚本文件的编辑器,执行 File→new→function 命令,即可打开函数文件的编辑器。函数文件的标志就是以 function 开头,如图 3-1 所示。

脚本文件可在命令窗口直接执行,也称为命令文件,直接输入文件名即可执行。例如,文件名为 f1.m,则在命令窗口直接输入 f1 即可执行这个脚本文件,得到运行结果,如图 3-2 所示,其变量定义在工作空间中。

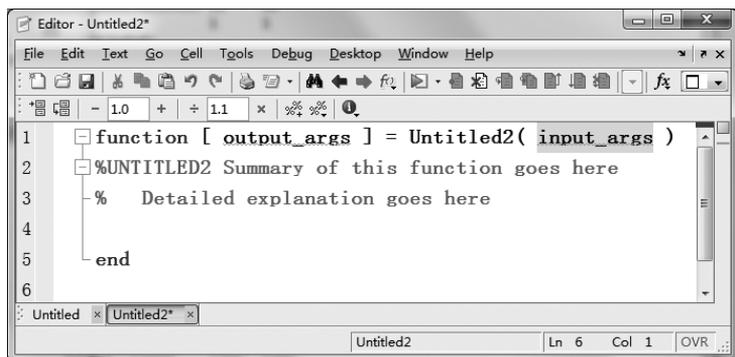
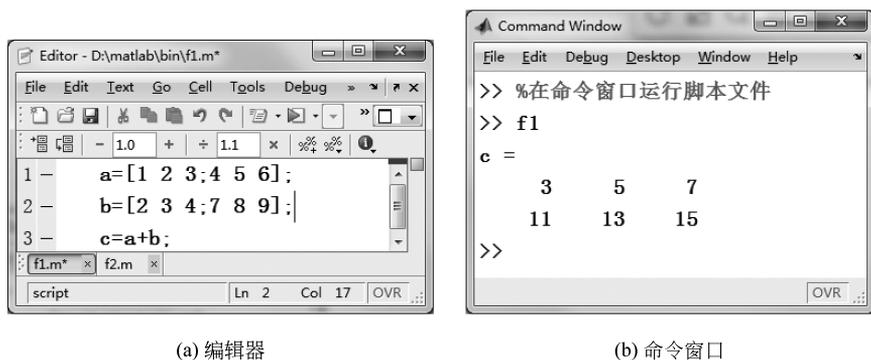


图 3-1 定义函数文件



(a) 编辑器

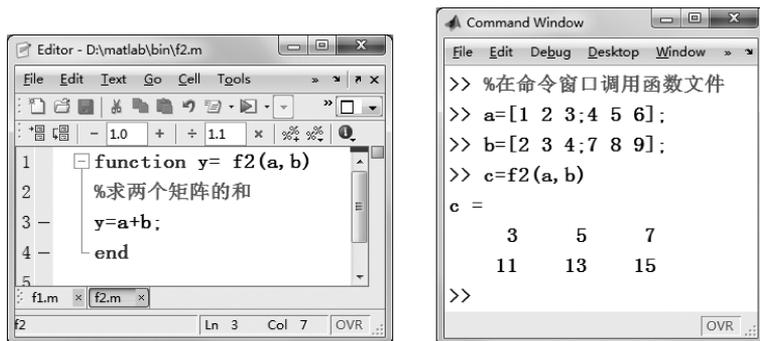
(b) 命令窗口

图 3-2 脚本文件的运行方式

函数文件是定义一个函数(function),不能在编辑器窗口直接执行,而必须以函数调用的方式来执行它。函数文件在保存时需要文件名和函数名一致。当函数文件名与函数名不相同, MATLAB 将忽略函数名,调用时使用函数文件名。例如,函数名为 f_2 ,那么保存的文件名也应该为 f_2 ,在命令窗口调用函数时,使用如下的一般调用形式:

函数名 (实际参数)

例如 $f_2(a,b)$,函数文件的定义和调用如图 3-3 所示。



(a) 函数文件的定义

(b) 函数文件的调用

图 3-3 函数文件的定义和调用

3.2 顺序结构程序设计

顺序结构是指按照程序中语句的排列顺序依次执行,直到程序的最后一个语句。这也是最简单的一种程序结构。一般涉及数据的输入、数据的计算或处理、数据的输出等内容。



输入和输出

1. 数据的输入

从键盘输入数据,可使用 input 函数来实现,其格式为

```
A=input(提示信息,选项);
```

其中,提示信息为字符串,用于提示用户输入什么样的数据。例如:

```
A=input('请输入 A 矩阵:');
```

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 函数和 fprintf 函数。

1) disp 函数将数据输出到 MATLAB 的命令窗口

disp 函数的调用格式为

```
disp(输出项)
```

其中,输出项既可以是字符串,也可以是矩阵。

用 disp 函数显示矩阵时将不显示矩阵的名字,而且其格式更紧密,且不留任何没有意义的空行。

2) 用 fprintf 函数格式化输出数据到文件中

fprintf 函数显示带有相关文本的一个或多个值,允许程序员控制显示数据的方式。它在命令行窗口打印一个数据的一般格式如下:

```
fprintf(文件句柄 fid,格式 format,数据 data)
```

其中,fid 表示由 fopen 函数打开的文件句柄,如果 fid 省略,则直接输出在屏幕上;format 用于表示一个描述打印数据方式的字符串;data 代表要打印的一个或多个标量或数组。format 包括两方面的内容:一方面是打印文本内容;另一方面是打印内容中的数据格式。例如:

```
>>fprintf('the value of pi is% 6.2f\n',pi)
the value of pi is  3.14
```

打印的结果为 the value of pi is 3.14,后面带有一个换行符。转义序列%6.2f代表在本函数中的第一个数据项将占有6个字符宽度,小数点后有2位小数。

fprintf函数有一个重大的局限性,只能显示复数的实部。当计算结果是复数时,这个局限性将会产生错误。在这种情况下,最好用disp显示数据。

format命令中格式符的含义如表3-1所示。

表 3-1 format 命令中格式符的含义

格式符	功 能
%d	把值作为整数来处理
%e	用科学记数法来显示数据
%f	用于格式化浮点数,并显示这个数
%g	用科学记数格式,或浮点数格式,根据长度最短的显示
\n	换行符

3. 程序的暂停

MATLAB 中程序暂停函数的调用格式为

pause(延时秒数)

若省去延时秒数,直到用户按任意键程序继续执行,按 Ctrl+C 键强行中止程序的执行。

例如,要求输入两个同型矩阵,然后求两个矩阵的和,再将结果显示在屏幕上。定义脚本文件如图3-4(a)所示,文件名为f3.m。运行时,在命令窗口输入文件名f3,出现提示信息“输入矩阵a”,此时输入[1 2 3;4 5 6]并回车,再出现提示信息“输入矩阵b”,此时输入[5 6 7;2 3 4],则显示运行结果,如图3-4(b)所示。

```

1 - a=input('输入矩阵a');
2 - b=input('输入矩阵b');
3 - c=a+b;
4 - disp('a与b的和: ')
5 - disp(c)

```

(a) 脚本文件

```

>> f3
输入矩阵a[1 2 3;4 5 6]
输入矩阵b[5 6 7;2 3 4]
a与b的和:
     6     8    10
     6     8    10
>>

```

(b) 显示结果

图 3-4 input 函数和 disp 函数举例

3.3 选择结构程序设计

选择结构又称为分支结构,是根据给定的条件是否成立来决定程序的执行流程。MATLAB 中可以用 if 语句、switch 语句和 try 语句实现选择结构。

1. if 语句

MATLAB 中,if 语句有 3 种格式。

1) 单分支 if 语句

格式:

```
if 条件
    语句组
end
```

如果条件成立就执行语句组,条件通常是关系表达式或逻辑表达式,当条件结果为标量时,非 0 表示条件成立,0 表示条件不成立;当条件结果为矩阵时,如果矩阵为非空,且不包含 0 元素,条件成立,否则不成立。

例如,[1,2;0,3]表示条件时,条件不成立;[1,2;3,4]表示条件时,条件成立。

2) 双分支 if 语句

格式:

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

如果条件成立则执行语句组 1,否则执行语句组 2。

3) 多分支 if 语句

格式:

```
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
```

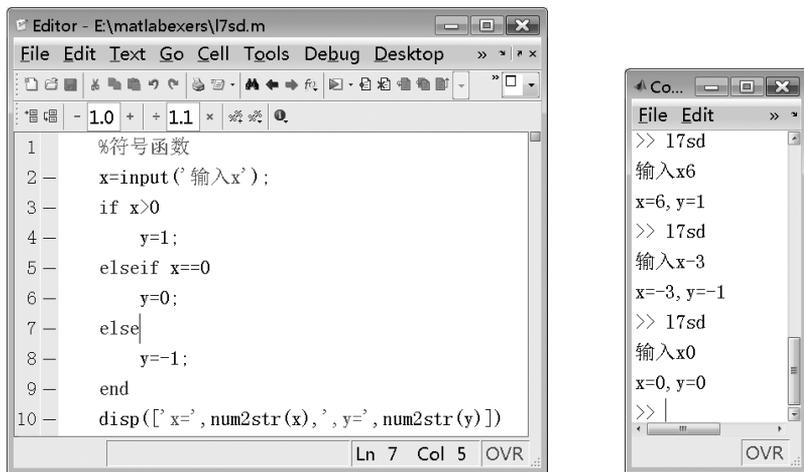


if 语句

【例 3.1】 编写程序完成以下分段函数,要求输入 x 的值,输出相应的 y 值。

$$y = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

程序和运行过程见图 3-5,这里脚本文件保存为 17sd.m。在命令窗口输入 17sd,出现输入提示符“输入 x”,输入一个值后,即可看到运算结果。程序中的 num2str 函数的功能是将数字转换为字符串。



(a) 分段函数脚本文件的定义

(b) 分段函数脚本文件的调用

图 3-5 分段函数脚本文件的定义和调用

【例 3.2】 编写函数,完成符号函数的功能。

首先要定义函数,计算符号函数时,需要已知 x ,因此函数要有一个形式参数 x ,再给函数一个名字,这里用 ff3。这样,函数首部即可确定为 `function y=ff3(x)`,当 x 为形参时,它的值是调用函数时传递过来的,所以在函数中无须也不能再输入 x ,即当 x 为已知时,求完函数值也不在函数内打印,而是由函数名带回到调用它的位置,详见图 3-6。

【例 3.3】 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数的平方;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(lower(c))
elseif c>='a' && c<='z'
    disp(upper(c))
elseif c>='0' && c<='9'
    disp(str2double(c)^2)
else
    disp(c)
end
```

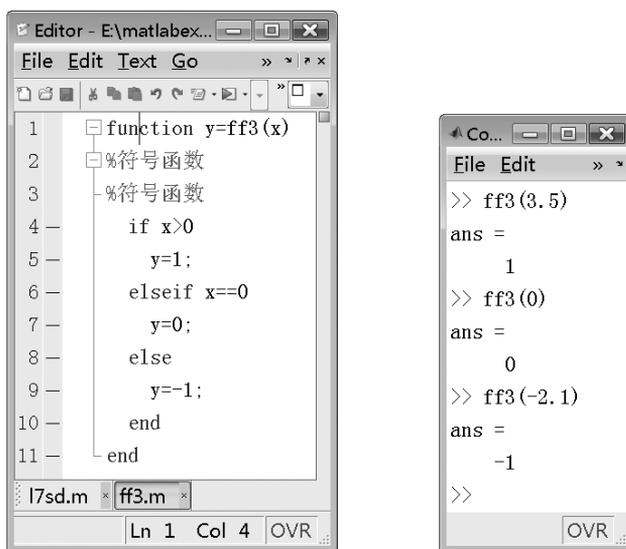


图 3-6 分段函数的函数文件的定义和调用

2. switch 语句

switch 语句是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

```

【例 3.4】 输入某个学生的成绩 g (假设 $0 \leq g \leq 100$)。如果 $g \geq 90$, 输出 A; $80 \leq g < 90$, 输出 B; $70 \leq g < 80$, 输出 C; $60 \leq g < 70$, 输出 D; $g < 60$, 输出 E。

建立脚本文件,取名 swi1.m,程序代码如下:

```

x=input('输入整数成绩');
switch(fix(x/10))
case {9,10}
disp('A')
case 8

```



switch 和 try 语句

```

disp('B')
case 7
disp('C')
case 6
disp('D')
otherwise
disp('E')
end

```

在命令窗口输入 `swil`, 输入 88, 得到 B。

```

>>swil
输入整数成绩 88
B

```

3. try 语句

try 语句是一种试探性执行语句, 其语句格式为

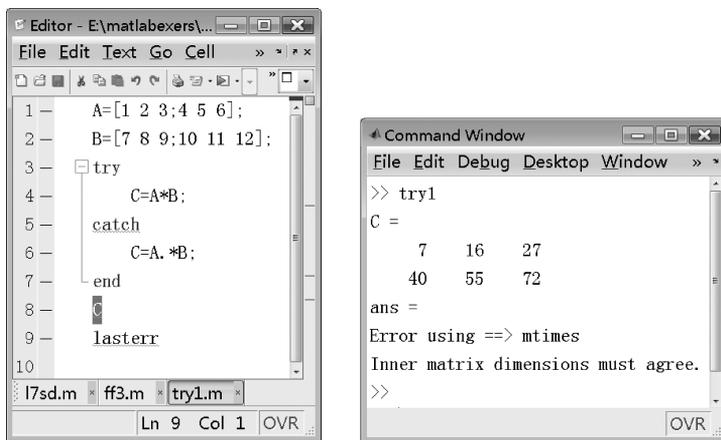
```

try
    语句组 1
catch
    语句组 2
end

```

try 语句先试探执行语句组 1, 如果语句组 1 在执行过程中出现错误, 则将错误信息保留在 `lasterr` 变量中, 并转去执行语句组 2。

图 3-7 的例子为使用 try 语句试探两个矩阵相乘, 如果不能进行矩阵乘法, 就进行点乘。执行 `C=A * B` 时出错, 因为第一个矩阵的列数和第二个矩阵的行数不相等, 不能进行矩阵乘法, 转去执行 `C=A .* B`, 即两个矩阵的对应元素相乘。结果 C 矩阵就是矩阵点乘的结果, 最后输出错误信息。如果文件最后没有 `lasterr` 一行, 则不会输出错误信息。



(a) try 语句的定义

(b) try 语句的执行

图 3-7 try 语句举例

3.4 循环结构程序设计

循环是指按照给定的条件,重复执行指定的语句,这是十分重要的一种程序结构。MATLAB 提供了两种实现循环的结构语句: for 语句和 while 语句。

1. for 语句

1) 简单格式

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



for 语句

【例 3.5】 一个 3 位整数,其各位数字的立方和等于该数本身则称为水仙花数,求出 100~999 的全部水仙花数。

```
for m=100:999;
    m1=fix(m/100);           %求 m 的百位数字,fix 向 0 方向取整
    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
```

2) for 更一般的表达式

```
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。实际上,“初值:增量:终值”是一个仅为一行的矩阵(行向量),因而列向量是单个数据。

【例 3.6】 矩阵作为循环变量。a 矩阵中存放的是 5 个学生 4 门课的成绩,求每个人的总成绩。

```
s=0;
a=[65, 76, 56, 78; 98, 83, 74, 85; 76, 67, 78, 79; 98, 58, 42, 73; 67, 89, 76, 87];
for k=a
    s=s+k;
end
disp(s');
```

结果为

```
275 340 300 271 319
```

即将 4 列加在一起,得到 5 个学生的 4 门课的成绩总和,输出的是 s 的转置,所以是一行结果。

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: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 3.7】 用 while 循环求 1~100 整数的和。

```
sum=0;
    i=1;
while i<=100
    sum=sum+i;
    i=i+1;
end
sum
```

运行结果

```
sum=
    5050
```

3. 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。可以按照嵌套层数,分别叫作二重循环、三重循环等。处于内部的循环叫作内循环,处于外部的循环叫作外循环。在设计多重循环时,要特别注意内、外循环之间的关系,以及各语句放置的位置。

4. break 语句和 continue 语句

break 语句用来跳出循环体,结束整个循环。continue 语句用来结束本次循环,接着进行下一次是否执行循环的判断。该语句一般与 if 语句配合使用。

在多重循环中,break 语句只能使程序跳出包含它的最内层的那个循环。

【例 3.8】 求 [100,200]区间第一个能被 33 整除的整数。

```
for n=100:200
    if rem(n,33)~=0
        continue
```



while 和 break 语句

```

        end
    n
    break
end

```

运行结果为

```

n=
132

```

【思考】 如果不用 continue 语句,只用 break 语句程序该如何修改?

实验与习题 3

3.1 输入一个 3 行 2 列的矩阵,求这个矩阵和它的转置的乘积。

3.2 输入 x ,计算函数值 y , x 可以是标量,也可以是向量。

$$y = 1/2e^{x/3} + x^2 \sin(x)$$

3.3 输入 k 和 t ,计算乘积 $z = (t-1) * (t-2) * \dots * (t-(k-1)) * (t-(k+1)) * \dots * (t-7)$ 。

3.4 输入一个正整数 n ,求 n 以内的奇数和。

3.5 产生一个 4×4 的整数矩阵,每个元素的值介于 7~15,然后求主对角元素的和。

3.6 根据以下程序段,下列()选项是错的。

```

a=[1,2,3,4;5,6,7,8];
s=0
for n=a
    s=s+n;
end
s

```

A. $s(1)=10$ B. $s(1,1)=10$ C. $s(2,1)=6$ D. $s(2)=26$

3.7 以下程序段输出的结果为()。

```

a=[1 2 3];
b=[2 4 6];
try
    c=a.*b;
catch
    c=a*b;
end
c

```

A. 没有结果 B. 28 C. [3 6 9] D. [2 8 18]

3.8 执行以下程序后, x 的值为()。

```
a=[1 2 3;4 0 6; 2 1 5];
x=2;
if a
    x=1;
else
    x=0;
end
x
```

A. 2 B. 1 C. 0 D. a

3.9 执行以下程序,命令窗口中输入 9,则输出结果为()。

```
a=input('input a');
b=sqrt(a);
disp(['a=', num2str(a), ' b=', num2str(b)])
```

A. a=9 b=3 B. 3 C. 9 D. [a=9 b=3]

3.10 执行以下两句,在命令窗口输入 abc,则输出为()。

```
a=input('input a','s');
disp(['a=',a])
```

A. abc B. a C. abc=a D. a=abc

3.11 利用秦九韶算法计算多项式 $y = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1}$ 的值。

例如,计算 $3x^2 + 2x + 1$,当 $x = 2$ 时值为 17。

$$y = (\dots((p_1x + p_2)x + p_3)x + \dots + p_n)x + p_{n+1}$$

%秦九韶算法求多项式的值

```
n=input('输入多项式次数 n');
```

```
str=['按降幂顺序输入', num2str(n+1), '个系数'];
```

```
%num2str 函数把数值转换为字符串。
```

```
p= (1) (str);                      %输入
```

```
x=input('输入 x');
```

```
y=p(1);
```

```
for i=2:n+1
```

```
    y= (2) ;
```

```
end
```

```
(3) (['多项式值为', num2str(y)])                      %输出
```

运行过程如下:

输入多项式次数 n 2

n=

2

按降幂顺序输入 3 个系数 [3 2 1]



秦九韶算法

输入 $x=2$

多项式值为 17

3.12 使用 MATLAB 函数求多项式 $3x^2+2x+1$, 当 $x=2$ 时的值。

```
>>a=[3 2 1];
```

```
>>polyval(a,2)
```

说明: `polyval(a,2)` 就是求系数为 a 的多项式当 $x=2$ 时的值。



polyval