



Matlab程序设计

第5讲



Matlab命令的执行方式

■ 交互式命令执行方式（命令窗口）

逐条输入，逐条执行，操作简单、直观，但速度慢，命令修改不便。

■ M文件的程序执行方式

将命令编成程序存储在一个文件中（M文件），依次运行文件中的命令，可以重复进行。

■ Matlab程序设计有传统高级语言的特征，又有自己独特的特点，可以利用数据结构的优点，使程序结构简单，编程效率高。



5.1 M文件的分类

- 用Matlab语言编写的程序，称为M文件。

是由若干Matlab命令组合在一起构成的，它可以完成某些操作，也可以实现某种算法。

- M文件根据调用方式的不同分为两类：

命令文件 (Script File)

函数文件 (Function File)

- 它们的扩展名都是.m



命令文件和函数文件的区别

- 命令文件没有输入参数，也不返回输出参数；函数文件可以带输入参数，也可以返回输出参数。
- 命令文件对工作空间中的变量进行操作，文件中所有命令的执行结果也返回工作空间中；函数文件中定义的变量为局部变量，当函数文件执行完毕时，这些变量也被清除。
- 命令文件可以直接运行；函数文件不能直接运行，要以函数调用的方式来调用它。



例5.1 建立文件将变量a、b的值互换。

命令文件：

```
clear;
```

```
a = 1:10;
```

```
b = [11,12,13,14;15,16,17,18];
```

```
c = a; a = b; b = c;
```

```
a
```

```
b
```

将文件保存为**exch**，并在命令窗口执行。

执行结果：

```
a =
```

```
11 12 13 14
```

```
15 16 17 18
```

```
b =
```

```
1 2 3 4 5 6 7 8 9 10
```



函数文件

fexch.m

function [a,b] = fexch(a,b)

c = a; a = b; b = c;

然后在命令窗口调用该函数文件：

clear;

x = 1:10;

y = [11,12,13,14;15,16,17,18];

[a,b] = fexch(x,y)

输出结果为：

x =

11 12 13 14

15 16 17 18

y =

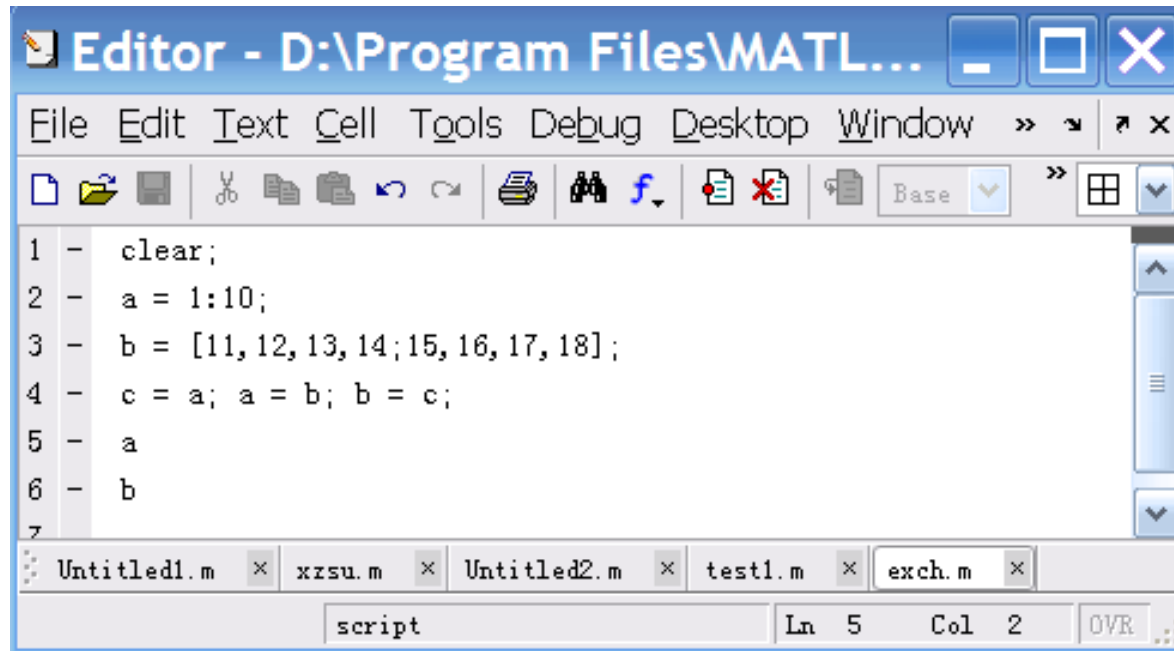
1 2 3 4 5 6 7 8 9 10

函数参数a, b, c未保留在工作空间中, x, y保留在工作空间中。

M文件的建立与打开

M文件是一个文本文件，可以用任何编辑程序来建立和编辑，一般最常用的是使用Matlab提供的文本编辑器。

该编辑器是一个集编辑和调试于一体的工作环境。





5.2 程序控制结构

- 顺序结构
- 选择结构
- 循环结构

任何复杂的程序都可以由这3种基本结构构成。



5.2.1 顺序结构

- 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。（最简单的一种程序）

1、数据的输入

从键盘输入数据，则可以使用**input**函数来进行，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，用于提示用户输入数据。

例如：从键盘输入A矩阵，可以采用下面的命令来完成

A = input（‘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：**xm = input(‘What’s your name?’,’s’);**

5.2.1 顺序结构

2、数据的输出

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。例如：

A = 'Hello, Tom';

disp(A)

输出为： Hello, Tom

又如：**A = [1,2,3;4,5,6;7,8,9];**

disp(A)

输出为：

1 2 3

4 5 6

7 8 9

%disp函数输出格式更紧凑

5.2.1 顺序结构

例5.2 求一元二次方程 $ax^2 + bx + c = 0$ 的根。

由于Matlab能进行复数运算，所以不需要判断方程的判别式，可直接根据求根公式求根。

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))]);
```

程序输出为：

a=?4

b=?78

c=?54

x1=-0.7188,x2=-18.7812



5.2.1 顺序结构

3、程序的暂停

程序执行过程中暂停，可用pause函数，其调用格式为：

pause (延迟描述)

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。

若要强行中止程序的运行可按Ctrl+C键。



5.2.2 选择结构

选择结构是根据给定的条件成立或不成立，分别执行不同的语句。
Matlab用于实现选择结构的语句有if语句，switch语句和try语句。

1. if语句

在Matlab中，if语句有3种格式。

(1)单分支if语句

语句格式：

```
if 条件
    语句组
end
```

例如：当x是整数矩阵时，输出x的值

```
if fix(x)==x
    disp(x);
end
```



5.2.2 选择结构

(2) 双分支if语句

语句格式:

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，然后再执行if语句的后续语句。



5.2.2 选择结构

例5.3 计算分段函数：

$$y = \begin{cases} \cos(x+1) + \sqrt{x^2 + 1}, & x = 10 \\ x\sqrt{x} + \sqrt{x}, & x \neq 10 \end{cases}$$

程序如下：

```
x = input('请输入x的值: ');  
if x == 10  
    y = cos(x+1)+sqrt(x*x+1);  
else  
    y = x*sqrt(x+sqrt(x));  
end  
y
```

也可以用单分支if语句来实现：

```
x = input('请输入x的值: ');  
y = cos(x+1)+sqrt(x*x+1);  
if x~=10  
    y = x*sqrt(x+sqrt(x));  
end  
y
```



5.2.2 选择结构

(3) 多分支if语句

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

例5.4 大小写字母的置换

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。

程序如下：

```
c = input('请输入一个字符:', 's');
```

```
if c >= 'A' & c <= 'Z'
```

```
    disp(char(abs(c) + (abs('a') - abs('A'))));
```

```
elseif c >= 'a' & c <= 'z'
```

```
    disp(char(abs(c) - (abs('a') - abs('A'))));
```

```
elseif c >= '0' & c <= '9'
```

```
    disp(abs(c) - abs('0'));
```

```
else
```

```
    disp(c);
```

abs函数可以得到ASCII码



5.2.2 选择结构

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：

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

switch子句后面的表达式应为一个标量或一个字符串；
case子句后面的表达式不仅可以为一个标量或一个字符串，还可以为一个单元矩阵。



例5.5

某商场对顾客所购买的商品实行打折销售，标准如下：

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。

请同学们试着编程实现。



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}
        rate = 3/100;        %价格大于等于200但小于500
    case num2cell(5:9)
        rate = 5/100;        %价格大于等于500但小于1000
    case num2cell(10:24)
        rate = 8/100;        %价格大于等于1000但小于2500
    case num2cell(25:49)
        rate = 10/100;       %价格大于等于2500但小于5000
    otherwise
        rate = 14/100;       %价格大于等于5000
end
price = price*(1-rate)       %输出商品实际销售价格
```

num2cell函数是将数值
矩阵转化为单元矩阵。



5.2.2 选择结构

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语句先试探性执行语句组1，如果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5.6

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
```

```
C =
```

```
    7    16    27
```

```
   40    55    72
```

```
ans =
```

```
Error using ==> mtimes
```

```
Inner matrix dimensions must agree.
```



5.2.3 循环结构

循环是指按照给定的条件，重复执行指定的语句，Matlab提供了两种实现循环结构的语句：**for语句**和**while语句**。

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5.7

已知 $y = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{n^2}$ ，当 $n=100$ 时，求 y 的值。

程序如下：

```
y = 0; n = 100;
```

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

```
    y = y+1/i^2;
```

```
end
```

```
y
```

输出结果为：

```
y =
```

```
    1.6350
```

利用Matlab的特点，常用向量运算来代替循环操作，程序可以如下：

```
n = 100;
```

```
i = 1:n;
```

```
f = 1./i.^2;
```

```
y = sum(f)
```




5.2.3 循环结构

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5.8

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的和。

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

输出结果为：

```
Enter a number(end in 0):67
Enter a number(end in 0):89
Enter a number(end in 0):93
Enter a number(end in 0):70
Enter a number(end in 0):0
sum =
    319
mean =
    79.7500
```



5.2.3 循环结构

3、break语句和continur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5.9 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
```

程序输出结果为：

```
n =
    105
```