

# 第 3 章 MATLAB 程序流程控制

Lecturer: 白煌

杭州师范大学  
信息科学与技术学院

2022.10.14



# 本章要点

- M 文件的概念与基本操作
- MATLAB 程序控制结构
- MATLAB 函数文件
- MATLAB 程序调试与优化



# 目录

① 3.1 M 文件

② 3.2 程序控制结构



## 3.1.1 M 文件的分类

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：

- 命令文件或脚本文件（Script File）
- 函数文件（Function File）



### 3.1.1 M 文件的分类

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  $c$ 。

$$c = \frac{5}{9}(f - 32)$$



## 3.1.1 M 文件的分类

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  $c$ 。

$$c = \frac{5}{9}(f - 32)$$

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 的命令行窗口中输入 `f2c`, 将会执行该命令文件并得到执行结果。



## 3.1.1 M 文件的分类

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 的命令行窗口调用该函数文件。

```
>> clear
>> y=input('Input Fahrenheit temperature: ');
>> x=f2c(y)
```



## 3.1.2 M 文件的建立与打开

### 1. 建立新的 M 文件:





## 3.1.2 M 文件的建立与打开

### 1. 建立新的 M 文件:

- 在 MATLAB 主窗口选择“主页”选项卡，在“文件”命令组中单击“新建脚本”命令按钮，屏幕上将出现 MATLAB 编辑器窗口。



## 3.1.2 M 文件的建立与打开

### 1. 建立新的 M 文件:

- 在 MATLAB 主窗口选择“主页”选项卡，在“文件”命令组中单击“新建脚本”命令按钮，屏幕上将出现 MATLAB 编辑器窗口。
- 在 MATLAB 命令行窗口输入命令：

>> edit 文件名

启动 MATLAB 编辑器后，输入 M 文件的内容并存盘。



## 3.1.2 M 文件的建立与打开

### 1. 建立新的 M 文件:

- 在 MATLAB 主窗口选择“主页”选项卡，在“文件”命令组中单击“新建脚本”命令按钮，屏幕上将出现 MATLAB 编辑器窗口。
- 在 MATLAB 命令行窗口输入命令：  
`>> edit 文件名`  
启动 MATLAB 编辑器后，输入 M 文件的内容并存盘。
- 在命令历史窗口选中一些命令（按住 Ctrl 可同时选择多条命令），然后从右键快捷菜单中选择“创建脚本”命令，将会启动 MATLAB 编辑器，并在编辑区中加入所选中的命令。编辑完成后，在编辑器窗口选择“编辑器”选项卡，在“文件”命令组中单击“保存”命令按钮存盘。



## 3.1.2 M 文件的建立与打开

2. 打开已有的 M 文件:



## 3.1.2 M 文件的建立与打开

### 2. 打开已有的 M 文件:

- 在 MATLAB 主窗口选择“主页”选项卡，在“文件”命令组中单击“打开”命令按钮，再从弹出的下拉菜单中选择“打开”命令，从“打开”对话框中选择所需打开的M文件，也可以从弹出的下拉菜单中选择最近使用的文件。



## 3.1.2 M 文件的建立与打开

### 2. 打开已有的 M 文件:

- 在 MATLAB 主窗口选择“主页”选项卡，在“文件”命令组中单击“打开”命令按钮，再从弹出的下拉菜单中选择“打开”命令，从“打开”对话框中选择所需打开的M文件，也可以从弹出的下拉菜单中选择最近使用的文件。
- 在 MATLAB 命令行窗口输入命令：  
`>> edit 文件名`  
则打开指定的 M 文件。



## 3.1.2 M 文件的建立与打开

### 2. 打开已有的 M 文件:

- 在 MATLAB 主窗口选择“主页”选项卡，在“文件”命令组中单击“打开”命令按钮，再从弹出的下拉菜单中选择“打开”命令，从“打开”对话框中选择所需打开的 M 文件，也可以从弹出的下拉菜单中选择最近使用的文件。
- 在 MATLAB 命令行窗口输入命令：  
`>> edit 文件名`  
则打开指定的 M 文件。
- 在当前文件夹窗口双击要打开的 M 文件，则打开该 M 文件。



## 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。





## 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');`



## 3.2.1 顺序结构

### 2. 数据的输出:

MATLAB 提供的命令行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。



## 3.2.1 顺序结构

例 3-2: 输入  $x$ 、 $y$  的值，并将它们的值互换后输出。



## 3.2.1 顺序结构

例 3-2: 输入  $x$ 、 $y$  的值，并将它们的值互换后输出。

```
x=input('Input x please. ');  
y=input('Input y please. ');  
z=x;  
x=y;  
y=z;  
disp(x);  
disp(y);
```



## 3.2.1 顺序结构

例 3-3: 求一元二次方程  $ax^2 + bx + c = 0$  的根。



## 3.2.1 顺序结构

例 3-3: 求一元二次方程  $ax^2 + bx + c = 0$  的根。

```
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))]);
```



## 3.2.1 顺序结构

### 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。





## 3.2.2 选择结构

### (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。



## 3.2.2 选择结构

例 3-4: 计算分段函数的值。



## 3.2.2 选择结构

例 3-4: 计算分段函数的值。

```
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
```



## 3.2.2 选择结构

例 3-4: 计算分段函数的值。

```
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
```

也可以用单分支 if 语句来实现。



## 3.2.2 选择结构

### (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



## 3.2.2 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。



## 3.2.2 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(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);  
end
```



## 3.2.2 选择结构

### 2. switch语句

switch 表达式

case 值 1

语句组 1

case 值 2

语句组 2

.....

case 值 m

语句组 m

otherwise

语句组 n

end





## 3.2.2 选择结构

当表达式的值等于值 1 时，执行语句组 1，当表达式的值等于值 2 时，执行语句组 2，.....，当表达式的值等于值  $m$  时，执行语句组  $m$ ，当表达式的值不等于 case 所列的值时，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。



## 3.2.2 选择结构

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):

- $\text{price} < 200$             没有折扣
- $200 \leq \text{price} < 500$     3% 折扣
- $500 \leq \text{price} < 1000$     5% 折扣
- $1000 \leq \text{price} < 2500$     8% 折扣
- $2500 \leq \text{price} < 5000$     10% 折扣
- $5000 \leq \text{price}$             14% 折扣

输入所售商品的价格, 求其实际销售价格。



## 3.2.2 选择结构

### 3. try 语句

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。



## 3.2.2 选择结构

例 3-7：矩阵乘法运算要求两矩阵的尺寸相容，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。



## 3.2.2 选择结构

例 3-7: 矩阵乘法运算要求两矩阵的尺寸相容, 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。

```
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 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。



### 3.2.3 循环结构

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。



### 3.2.3 循环结构

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。

```
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
```





### 3.2.3 循环结构

例 3-9: 已知  $y = 1 + \frac{1}{3} + \frac{1}{5} + \cdots + \frac{1}{2n-1}$ , 当  $n = 100$  时, 求  $y$  的值。



### 3.2.3 循环结构

例 3-9: 已知  $y = 1 + \frac{1}{3} + \frac{1}{5} + \cdots + \frac{1}{2n-1}$ , 当  $n = 100$  时, 求  $y$  的值。

```
y=0;  
n=100;  
for i=1:n  
    y=y+1/(2*i-1);  
end  
y
```



## 3.2.3 循环结构

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。



## 3.2.3 循环结构

例 3-10: 写出下列程序的执行结果。



## 3.2.3 循环结构

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');
```



## 3.2.3 循环结构

### 2. while 语句

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。



## 3.2.3 循环结构

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的和。



## 3.2.3 循环结构

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的和。

```
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
```





## 3.2.3 循环结构

### 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。



## 3.2.3 循环结构

例 3-12: 求  $[100, 200]$  之间第一个能被 21 整除的整数。



## 3.2.3 循环结构

例 3-12: 求 [100, 200] 之间第一个能被 21 整除的整数。

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



## 3.2.3 循环结构

### 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。



### 3.2.3 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。



### 3.2.3 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
```

