

# 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 3.3 函数文件
- 2 3.4 特殊形式的函数
- 3 3.5 程序调试与优化
- 4 3.6 程序举例



## 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。



## 3.3.1 函数文件的基本结构

例 3-14: 编写函数文件求半径为  $r$  的圆的面积和周长。



## 3.3.1 函数文件的基本结构

例 3-14: 编写函数文件求半径为  $r$  的圆的面积和周长。

```
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
```

```
% 署名日期
```

```
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。



## 3.3.2 函数调用

例 3-15: 利用函数文件, 实现直角坐标  $(x,y)$  与极坐标  $(\rho,\theta)$  之间的转换。





## 3.3.2 函数调用

例 3-15: 利用函数文件, 实现直角坐标  $(x,y)$  与极坐标  $(\rho,\theta)$  之间的转换。

- ① 建立函数文件 tran.m:

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

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

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



## 3.3.2 函数调用

例 3-15: 利用函数文件, 实现直角坐标  $(x,y)$  与极坐标  $(\rho,\theta)$  之间的转换。

- ❶ 建立函数文件 tran.m:

```
function [rho,theta] = tran(x,y)
rho=sqrt(x*x+y*y);
theta=atan(y/x);
```

- ❷ 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
```



## 3.3.2 函数调用

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。



## 3.3.2 函数调用

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  $n!$ 。



## 3.3.2 函数调用

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  $n!$ 。

```
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 用两个预定义变量 `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。



### 3.3.3 函数参数的可调性

例 3-17: nargin 用法示例。



### 3.3.3 函数参数的可调性

例 3-17: nargin 用法示例。

① 函数文件 chararray.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
```





### 3.3.3 函数参数的可调性

例 3-17: nargin 用法示例。

① 函数文件 chararray.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
```

② 命令文件 mydemo.m:

```
x=[1:3];
y=[1;2;3];
chararray(x)
chararray(x,y')
chararray(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



## 3.3.4 全局变量与局部变量

例 3-18: 全局变量应用示例。



## 3.3.4 全局变量与局部变量

例 3-18: 全局变量应用示例。

- ① 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;
```



## 3.3.4 全局变量与局部变量

例 3-18: 全局变量应用示例。

- ① 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;
```

- ② 在命令行窗口中输入:

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



## 3.4 特殊形式的函数



## 3.4 特殊形式的函数

- ① 子函数：将多个函数定义放在同一个 M 文件中，这就存在子函数的定义问题。



## 3.4 特殊形式的函数

- ① 子函数：将多个函数定义放在同一个 M 文件中，这就存在子函数的定义问题。
- ② 内联函数：以字符串形式存在的函数表达式可以通过 `inline` 函数转化成内联函数。





## 3.4 特殊形式的函数

- ❶ 子函数：将多个函数定义放在同一个 M 文件中，这就存在子函数的定义问题。
- ❷ 内联函数：以字符串形式存在的函数表达式可以通过 `inline` 函数转化成内联函数。
- ❸ 匿名函数：匿名函数的基本格式如下

函数句柄变量=@(匿名函数输入参数)匿名函数表达式

其中，函数句柄变量相当于函数的别名，利用它可以间接调用函数；“@”是创建函数句柄的运算符；“@”后面定义了一个匿名函数，包括函数输入参数和函数表达式；函数有多个输入参数时，参数间用逗号分隔。



## 3.5.1 程序调试方法

- 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。



## 3.5.1 程序调试方法

- 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。



## 3.5.1 程序调试方法

- 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 程序性能分析与优化

- 程序性能分析
  - profile
  - tic-toc
- 程序优化
  - 采用向量化运算
  - 预分配内存空间
  - 减小运算强度



## 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，不能被其他任何整数所整除的整数。用筛选法求素数的基本思想是：要找出  $2 \sim m$  之间的全部素数，首先在  $2 \sim m$  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  $\sqrt{m}$  的数的倍数，剩下的数都是素数。



## 例 3-21

求定积分。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、梯形法以及辛普森法等。





## 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。

$$e^X = 1 + X + \frac{X^2}{2!} + \frac{X^3}{3!} + \cdots + \frac{X^n}{n!} + \cdots$$



# 补充例题



## 补充例题

- ① 使用 if-else-end 语句判断当给定变量  $x$  的值时，相应的函数值

$$y = \text{sign}(x)$$



## 补充例题

- ① 使用 if-else-end 语句判断当给定变量  $x$  的值时，相应的函数值

$$y = \text{sign}(x)$$

- ② 使用 switch-case-end 语句求任意底数的对数函数值

$$y = \log_n x$$



## 补充例题

- ① 使用 if-else-end 语句判断当给定变量  $x$  的值时，相应的函数值

$$y = \text{sign}(x)$$

- ② 使用 switch-case-end 语句求任意底数的对数函数值

$$y = \log_n x$$

- ③ 分别使用 for 循环和 while 循环语句求解

$$\sum_{i=1}^{100} i$$



## 补充例题

- ① 使用 if-else-end 语句判断当给定变量  $x$  的值时，相应的函数值

$$y = \text{sign}(x)$$

- ② 使用 switch-case-end 语句求任意底数的对数函数值

$$y = \log_n x$$

- ③ 分别使用 for 循环和 while 循环语句求解

$$\sum_{i=1}^{100} i$$

- ④ 使用 for 循环嵌套求解

$$x = \sin\left(\frac{n \cdot k \cdot \pi}{360}\right), \quad n \in [1 : 10], \quad k \in [1 : 4]$$



## 补充例题

- ① 使用 if-else-end 语句判断当给定变量  $x$  的值时，相应的函数值

$$y = \text{sign}(x)$$

- ② 使用 switch-case-end 语句求任意底数的对数函数值

$$y = \log_n x$$

- ③ 分别使用 for 循环和 while 循环语句求解

$$\sum_{i=1}^{100} i$$

- ④ 使用 for 循环嵌套求解

$$x = \sin\left(\frac{n \cdot k \cdot \pi}{360}\right), \quad n \in [1 : 10], \quad k \in [1 : 4]$$

- ⑤ 使用 while 循环求 Fibonacci 数列中第一个大于 9999 的元素

