

# MATLAB统计分析与应用

## —— MATLAB程序设计

主讲人：谢中华

ssstudy.com

科学软件学习网

# 主要内容

➤ MATLAB语言的流程结构

➤ M代码的编写与调试

# 第一节 MATLAB语言的流程结构

MATLAB作为一种程序设计语言，它提供了选择语句结构和循环语句结构，其中选择语句结构包括：if/elseif条件转移语句结构，switch开关语句结构，try...catch试探语句结构；循环语句结构包括：for循环和while循环语句结构。此外，MATLAB还提供了continue、break、return和pause等流程控制函数

# 一、选择结构

## 1. if/elseif条件转移语句结构

格式一：if 条件  
          语句组  
          end

格式二：  if 条件  
          语句组1  
          else  
          语句组2  
          end

# 1. if/elseif条件转移语句结构

格式三： 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m+1

end

**【例3.1-1】** 输入3个实数，判断以这3个数为边长能否构成三角形，若构成三角形，利用海伦公式求其面积。

% 交互式输入一个包含三个元素的向量

```
A = input('请输入三角形的三条边：');
```

```
if A(1) + A(2) > A(3) & A(1) + A(3) > A(2) & A(2) + A(3) >  
A(1)
```

```
    p = (A(1) + A(2) + A(3)) / 2;
```

% 用海伦公式求三角形面积

```
s = sqrt(p*(p - A(1))*(p - A(2))*(p - A(3)));
```

```
disp(['该三角形面积为：' num2str(s)]);
```

```
else
```

```
    disp('不能构成一个三角形。')
```

```
end
```

运行：请输入三角形的三条边：[4 5 6]

9.9216

## 2. switch/case开关语句结构

switch语句根据变量或表达式的取值不同，分别执行不同的语句。其格式为：

```
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m+1
```

```
end
```

### 【例3.1-2】根据变量 num 的值来决定显示的内容。

```
num=input('请输入一个数 : '); % 交互式输入一个数
switch num % 根据num的不同取值显示不同的信息
    case -1
        disp('I am a teacher.');
```

case 0

```
        disp('I am a student.');
```

case 1

```
        disp('You are a teacher.');
```

otherwise

```
        disp('You are a student.');
```

end



### 3. try...catch试探语句结构

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3.1-3】在figure窗口中添加工作表。

```
fig = figure;  
TablePosition = [0.0172,0.0250,0.8945,0.7937];  
try  
    SpreadSheet = actxcontrol('OWC11.Spreadsheet.11',...  
        TablePosition,fig);  
catch  
    datatable = uitable(fig,'units','normalized',...  
        'position',[0.0168 0.0299 0.9642 0.9415],...  
        'data',cell(10,5));  
end  
lasterr %显示出错原因
```

## 二、循环结构

### 1. for 循环

格式： for 循环变量 = 向量  
循环体语句

end

注：循环中，循环变量依次从向量中取值。

## 2. while循环

格式：  
while 条件  
    循环体语句  
end

## 3. 循环套嵌

【例 3.1-4】令  $y = f(n) = \sum_{i=1}^n i^2$ ，求使得  $y \leq 2000$

的最大的正整数  $n$  和相应的  $y$  值。

% 程序1：

```
y = 0;
for i = 1:100
    y = y + i^2;
    if y > 2000
        break; %跳出循环
    end
end
n = i - 1
y = y - i^2

n = 17
y = 1785
```

% 程序2

```
y = 0;
i = 0;
while y <= 2000
    i = i + 1;
    y = y + i^2;
end
n = i - 1
y = y - i^2
```

### 三、continue、break、return和pause函数

#### 1. continue函数

continue函数只能用在for或while循环结构的循环体语句中，它的功能是跳过当步循环直接执行下一次循环，通常与if语句结合使用。

#### 2. break函数

break函数也只能用在for或while循环结构的循环体语句中，它的功能是跳出break函数所在层循环，通常与if语句结合使用。

### 3. return函数

return函数的用法比较灵活，通常用在某个函数体里面，根据需要，可以用在函数体的任何地方，其功能是跳出正在调用的函数，通常与if语句结合使用。

## 4. pause函数

pause函数用来实现暂停功能

pause      %暂停程序的执行，等待用户按  
            任意键继续

pause(n)    %暂停程序的执行，n秒后继续，  
            n为非负实数

pause on    %开启暂停功能，使后续pause和  
            pause(n)指令可以执行

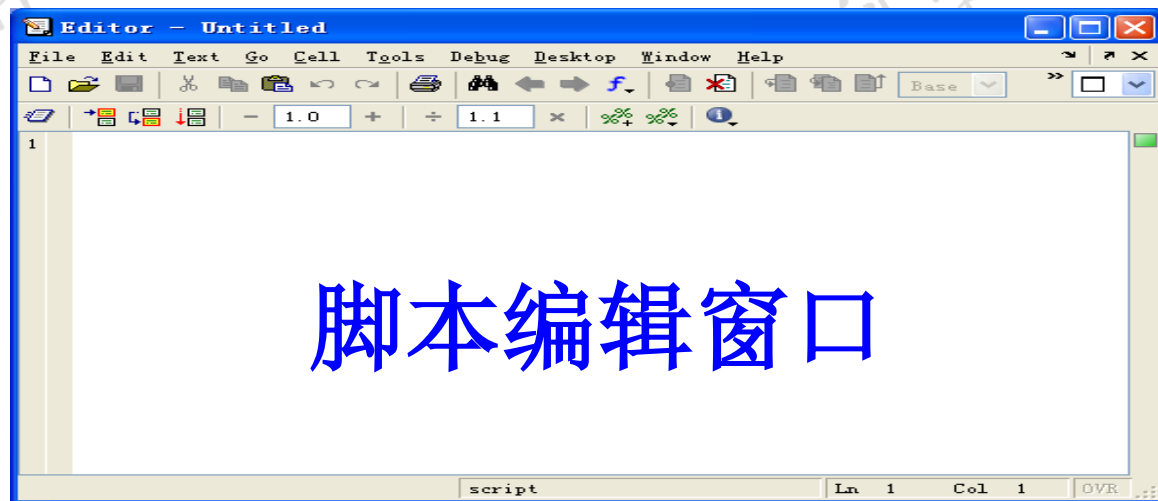
pause off   %关闭暂停功能，不执行后续  
            pause和pause(n)指令。



## 第二节 M代码的编写与调试

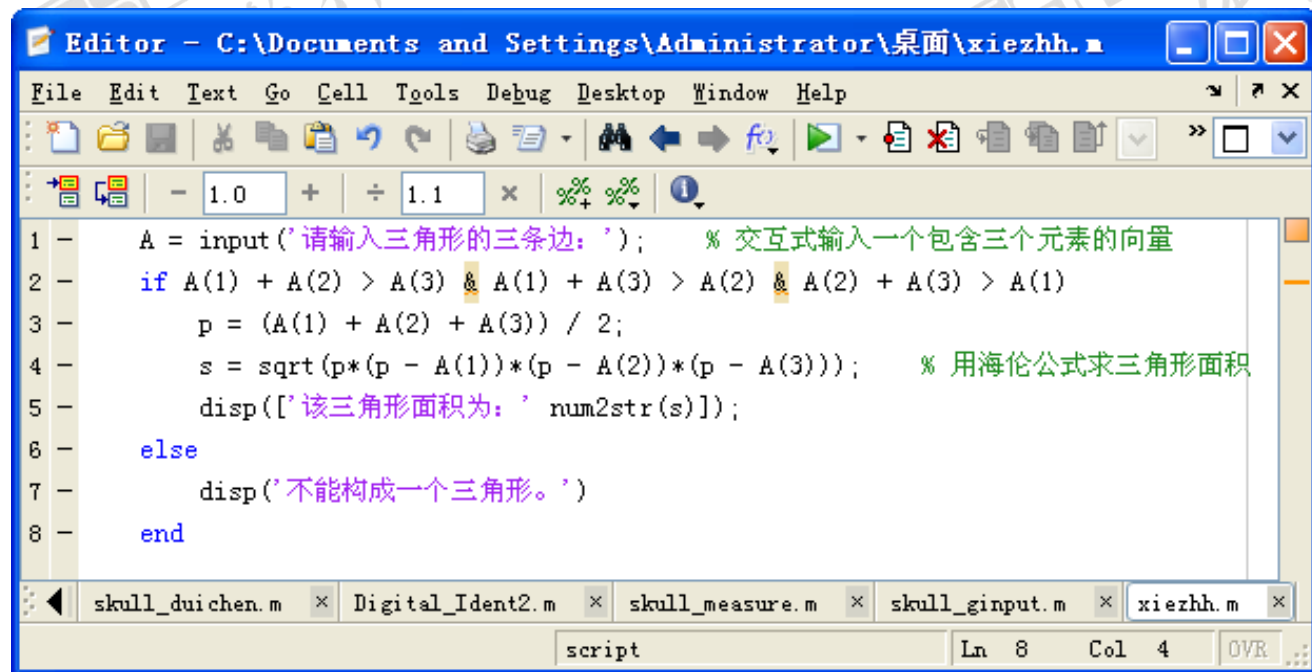
### 一、编辑m文件

单击快捷按钮  , 或依次单击菜单  
File→New→M-File ,打开如下脚本编辑窗口



## 二、脚本文件

所谓的**脚本文件**，就是将一些MATLAB命令简单的堆砌在一起保存成的M文件。



### 三、函数文件

**函数文件**就是按照一定格式编写的，可由用户指定输入和输出进行调用的M文件。

```
function [out1, out2, ...] = funname(in1, in2, ...)
```

注释说明部分（%号引导的行）

函数体

```
end
```

其中out1, out2, ...为**输出参数列表**，in1, in2, ...为**输入参数列表**，funname为**函数名**。

**注意：**函数输出参数列表中提到的变量要

在函数体中予以赋值，函数名与变量名的命名规则相同，另外函数名最好与文件名相同，并且自编函数不要与内部函数重名，否则极易引起错误。

$$f(x) = \begin{cases} x^2 \sin(x), & x < 0 \\ x \cos(x), & x \geq 0 \end{cases}$$

### 【例3.2-1】 利用函数文件求解例3.1-4。

```
function [n,y]=sumlm(m)
```

```
% 求n和y, 使得 $1^2+2^2+\dots+n^2 \leq m$ 
```

```
y = 0;
```

```
i = 0;
```

```
while y <= m
```

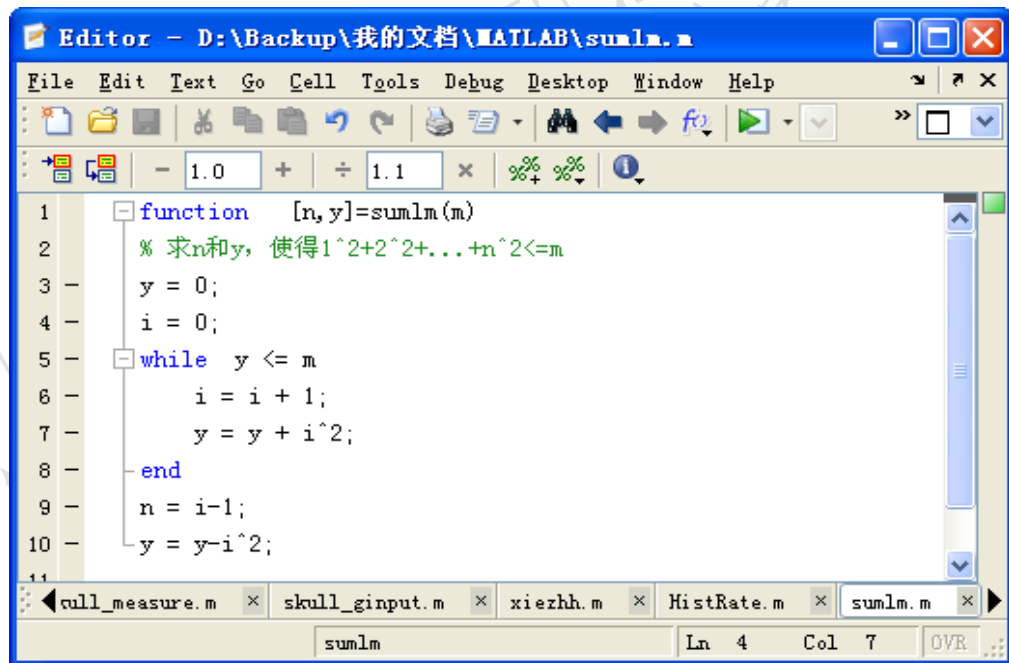
```
    i = i + 1;
```

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

```
end
```

```
n = i-1;
```

```
y = y-i^2;
```



The image shows a screenshot of the MATLAB Editor window. The title bar reads "Editor - D:\Backup\我的文档\MATLAB\sumlm.m". The menu bar includes File, Edit, Text, Go, Cell, Tools, Debug, Desktop, Window, and Help. The toolbar contains various icons for file operations, editing, and execution. The code editor displays the following MATLAB code:

```
1 function [n,y]=sumlm(m)
2 % 求n和y, 使得 $1^2+2^2+\dots+n^2 \leq m$ 
3 y = 0;
4 i = 0;
5 while y <= m
6     i = i + 1;
7     y = y + i^2;
8 end
9 n = i-1;
10 y = y-i^2;
```

The status bar at the bottom shows the current file is "sumlm", with the cursor at Line 4, Column 7.

## 四、匿名函数（在命令窗口或脚本文件中编辑）

### 1. 定义格式

匿名函数由@符引导建立，其定义格式为：

$f = @(arg1,arg2,...) \text{expr}$

### 2. 调用格式

匿名函数的调用格式为：

$y=f(arg1,arg2,...)$

## 【例3.2-2】建立匿名函数，求解 $\sin(\frac{\pi}{8})\cos(\frac{\pi}{9})$ 。

匿名函数程序如下：

```
f2 = @(x,y)sin(x).*cos(y);
```

```
y = f2(pi/8,pi/9)
```

$y =$

0.3596

# 五、子函数与嵌套函数

## 1. 子函数

通常在一个MATLAB主函数的内部会调用一些其他的MATLAB函数，我们把被调用的函数称为该主函数的子函数，子函数可以是MATLAB自带的内部函数，也可以是自编的外部函数；可以是以function打头的函数，也可以是匿名函数。



## 2. 嵌套函数

把一个或多个子函数放到同一个主函数的函数体内部而构成的函数称为嵌套函数。像循环的嵌套一样，嵌套函数可以是一层嵌套，也可以是多层嵌套，其一般形式如下。



### 单层嵌套

%—**嵌**—

```
function x = A(p1, p2)
```

```
...
```

```
function y = B(p3)
```

```
...
```

```
end
```

```
...
```

```
end
```

## ➤ 单层嵌套

%一嵌多

```
function x = A(p1, p2)
```

```
...
```

```
function y = B(p3)
```

```
...
```

```
end
```

```
function z = C(p4)
```

```
...
```

```
end
```

```
...
```

```
end
```

## ➤ 多层嵌套

% 多层嵌套

```
function x = A(p1, p2)
```

```
...
```

```
function y = B(p3)
```

```
...
```

```
function z = C(p4)
```

```
...
```

```
end
```

```
...
```

```
end
```

```
...
```

```
end
```

### 【例3.2-3】通过嵌套函数的方式编写函数 $y = \sqrt{(x+1)^2 + e^x} - 1$

```
function y = mainfun(x)
% 通过嵌套函数的方式编写函数
y = subfun1(x) + subfun2(x);
% 子函数1
function y1 = subfun1(x1)
    y1 = (x1 + 1)^2;
end
% 子函数2
function y2 = subfun2(x2)
    y2 = exp(x2);
end
y = subfun3(y);
end
%%-----
% 子函数3
%%-----
function y = subfun3(x)
y = sqrt(x) - 1;
end
```

## 六、函数的递归调用

【例3.2-4】生成斐波那契数列的第  $n$  项。

```
function y = fibonacci(n)
```

```
% 生成斐波那契数列的第n项
```

```
if (n < 0) | (round(n) ~= n)
```

```
    warning('输入参数应为非负整数');
```

```
    y = [];
```

```
    return;
```

```
elseif n < 2
```

```
    y = n;
```

```
else
```

```
    y = fibonacci(n-2)+fibonacci(n-1);
```

```
end
```

## 七、M代码的调试 ( debug )

1. 语法错误和运行结果错误
2. 设置断点进行调试

## 八. 设置全局变量

- 为说明全局变量的使用，在ComputeXY函数中定义全局变量a, b, c，程序如下：

```
function [x,y]=ComputeXY(t)
```

```
global a b c %注意变量之间用空格而不是逗号或其它标点隔开
```

```
x=cos(a*t);
```

```
y=abs(x)+c;
```

➤ 在命令窗口调用时应采用以下语句

```
global a b c
```

```
a=1.4;b=2;c=0.75;
```

```
[u,v]=ComputeXY(0:pi/4:pi)
```

则可获得正确结果



*Thank You*