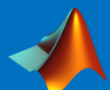
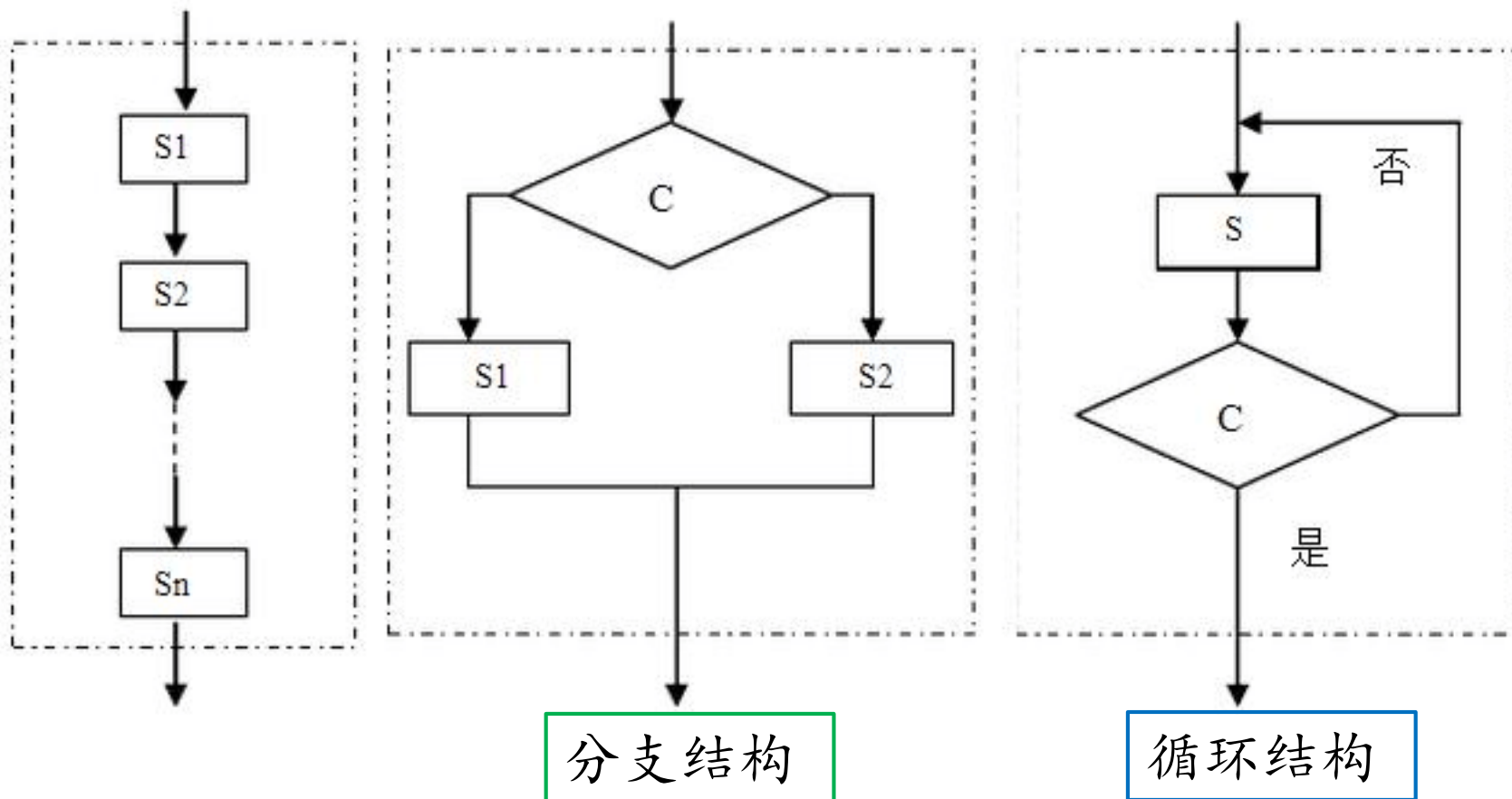


第1章 MATLAB程序设计语言与 初等数学运算

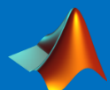
- 1.8 逻辑和关系运算
- 1.9 程序流程控制语句



计算机程序结构

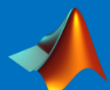


➤ 逻辑、关系运算和流程控制语句是实现复杂程序结构的基础；



关系运算

- 关系操作符： $==($ 等于 $)$ ， $\neq($ 不等于 $)$ ， $>($ 大于 $)$ ， $<($ 小于 $)$ ， $\geq($ 大于等于 $)$ ， $\leq($ 小于等于 $)$
- 关系运算的结果是二值逻辑量，它只能取1(真)或0(假)
- 两个同维矩阵间的关系运算规定为它们对应元素间的关系运算，运算结果仍是一个布尔矩阵
- “数”跟“矩阵”进行关系运算，规定为数与矩阵的每个元素进行关系运算，运算结果是一个与矩阵维数相同的布尔矩阵



例题

已知 $a1 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, $a2=5$ 。求 $a1 \geq a2$ 的运算结果

在命令窗口输入:


```
>> a1=[1 2 3;4 5 6;7 8 9]; a2=5; a1>=a2
```

运行后可观察到结果:

ans=

0	0	0
0	1	1
1	1	1

```
>> 1-(4/3-1)*3==0
```

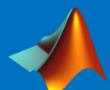


```
ans = 0
```



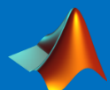
逻辑运算

- 两个逻辑量之间可以进行“与”、“或”和“非”三种基本逻辑运算及由它们组合而成的其它运算
- MATLAB中的逻辑操作符包括： $\&$ (逻辑与)， $|$ (逻辑或)， \sim (逻辑非)， xor （异或）， $\&\&$ （先决与）和 $||$ （先决否）
- 在逻辑运算中，非零元素的逻辑量为1，表示真，零元素的逻辑量为0，表示假；逻辑运算的结果仍然是逻辑量0（假）或1（真）
- 维数相同的矩阵进行逻辑运算时，定义为它们对应元素的逻辑运算结果
- 数与矩阵间进行逻辑运算，规则与关系运算相同，是数与矩阵各个元素间的逻辑运算



逻辑运算真值表

逻辑量及其运算	真值（逻辑量）			
A	1	1	0	0
B	1	0	1	0
$\sim A$	0	0	1	1
$\sim B$	0	1	0	1
$A \& B$	1	0	0	0
$A B$	1	1	1	0
$\text{xor}(A, B)$	0	1	1	0
$A \& \& B$	1	0	0	0
$A B$	1	1	1	0



例题

求数值矩阵a1的逻辑“非”，a1和1的“异或”

$a1=[1\ 0\ -5\ 0;3\ -2\ 0\ 6;0\ 0\ 5\ 7;9\ 2\ 1\ 9];$

解：键入

`>>a2=~a1`

`a2 =`

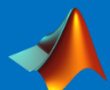
0	1	0	1
0	0	1	0
1	1	0	0
0	0	0	0

`>>a3=xor(a1,1)`

`a3=`

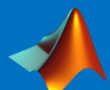
0	1	0	1
0	0	1	0
1	1	0	0
0	0	0	0

非0数值在逻辑运算中被视为真（1）



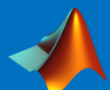
运算符的优先级

- MATLAB表达式中可能包含多种运算符：**数学运算符、关系运算符和逻辑运算符**
- 各运算符执行的先后是根据**优先级**别执行的
- 不管运算符的位置如何，具有高优先级的运算符先执行
- 具有相同优先级别的运算符则按先左后右的次序执行



运算符的优先级

优先级	运算符					
高	1	()				
	2	!	'	.^	^	一元运算符
	3	代数正+	代数负-	~		
	4	.*	.\	./	*	\
	5	+	-			
	6	:				
	7	<	>	==	>=	<=
	8	&				
	9					二元运算符
	10	&&				
低	11					



例题

以下命令的执行结果是什么？

`>> 3+5>7+8` 0

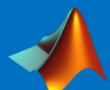
`>> 3+(5>7)+8` 11

`>> 3+(5>7)+~8` 3

`>> 3+5>7+~8*2` 1

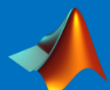
`>> 3+(5>7)+~8*2:5` 3 4 5

`>> 5>7&~8` 0



MATLAB程序流程控制

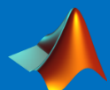
- MATLAB的程序流程控制语句包括：
 - 分支结构： **if**, **switch**
 - 循环结构： **for**, **while**
 - 错误控制： **try...catch**
 - 流程终止： **break**, **continue**, **return**, **error**



if选择语句

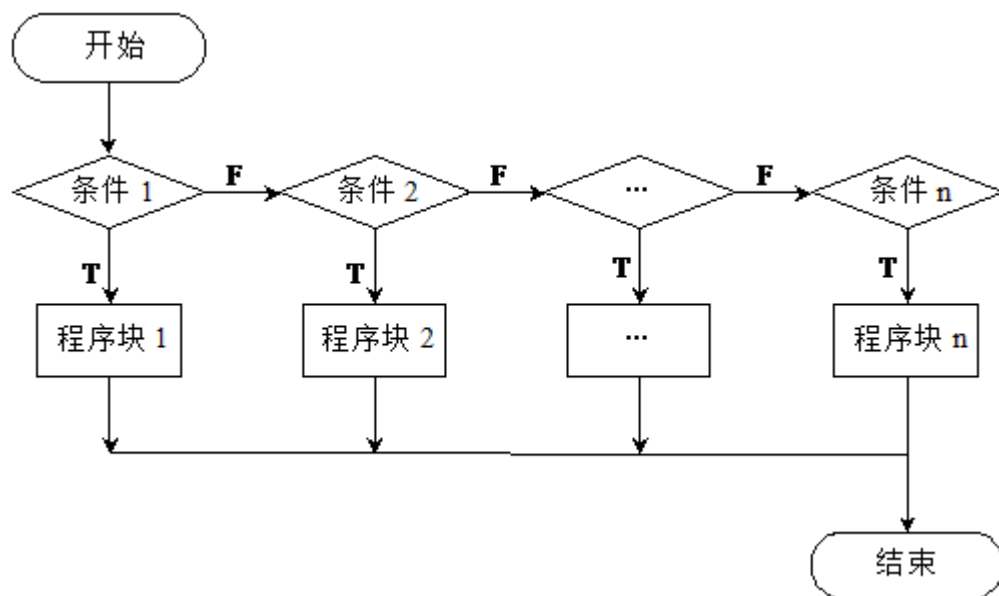
if语句的一般格式:

```
if condition1 % condition1均为关系、逻辑表达式或其组合;  
    statements1    %如果condition1的值为True, 则执行该语  
                  句组, 然后执行end以后语句; 如果为假则执行elseif语句;  
elseif condition2  
    statements2    %如果condition2的值为True, 则执行该语  
                  句组, 然后执行end以后的语句  
else  
    statements3    %如果condition1和condition2的都为False  
                  , 则执行该语句组  
end
```



if语句的说明

if选择结构中，只有if和end是必须存在的；elseif和else均可省略，elseif子句数量可选。



if condition
statements
end

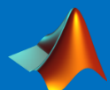
if condition
statements
else
statements
end

if condition
statements
elseif
statements
end



error函数

- 当程序出现无效的或不恰当的条件时，可使用error函数终止某个程序或函数的运行
- 程序执行到error函数时，会在命令窗口显示函数所包含的信息，之后终止程序或函数的执行，并将程序控制权返回到命令窗口命令行
- error函数可用于程序或函数任何位置，并不限于if，for，switch和while结构



例题

编写一个程序，当输入数字大于0时计算其平方根，当小于0时终止运算并返回错误信息(Negative input not allowed)

脚本文件
形式

```
x = input('x=')
if x>=0
    y=sqrt(x)
else
    error('Negative input not allowed')
end
```

函数形式

```
function y=caly(x)
if x>=0
    y=sqrt(x)
else
    error('Negative input not allowed')
end
```

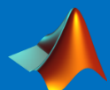


例题

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

以下程序实现根据输入的x值，计算对应函数y的值的功能，试补充完整以下程序。

```
function y=Cal(x)
if       x<0      
    y=abs(x)
elseif       x==0      
    y=1
elseif       x>0      
    y=sqrt(x)
end                    
```



switch多重分支结构

switch...case...otherwise语句的能力与if...else...end语句类似，但对多重选择的情况switch语句使代码更加易读。

switch的一般格式：

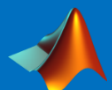
```
switch test_expr    %测试表达式test_expr可以是标量或字符串
case value
    statements      %当test_expr值是value时，执行该语句组
case {value1,value2,...}
    statements      %当test_expr值是value1或value2或.....时，执
                    行该语句组
otherwise %可以省略
    statements
end
```



例题

编程判断并显示当输入自变量 x 的具体数值时，函数 $y=x^4-19x^3+x+32$ 的值是奇数、偶数（0视为偶数，不分正负），还是其它

```
function Chaldemo(x)
y=x^4-19*x^3+x+32;
switch mod(y,2)
    case 1
        disp('It is odd')
    case 0
        disp('It is even')
    otherwise
        disp('It is others')
end
```

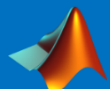


for循环结构

for循环结构用于循环次数已知的情況，其一般格式如下：

```
for 循环变量 = 表达式1(初值):表达式2(步长):表达式3(终值)
    statements （语句组）
end
```

for...end循环的执行过程是：先计算初值和终值，并把初值赋给循环变量；再判断循环变量的值是否超过了终值。若超过，则退出循环，执行end后面的语句，否则执行循环体的语句组，之后将循环变量加上一个步长，然后重复执行循环体内容，直至循环变量超过终值而退出循环为止



例题

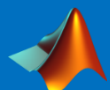
编程计算1至1000的加和值

```
clear
j=0;
for i=1:1000
    j=j+i;
end
j
i
```



```
j =
    500500
i =
    1000
```

可见循环变量的最终值为表达式的终值



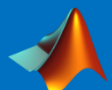
for循环中循环变量的值

以下程序的运行结果？

```
for i=1:3  
    i=i*2  
end
```



```
i = 2  
i = 4  
i = 6
```



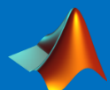
for循环与程序的效率

编程计算 $\sin(3x)$ 在 $x=0, \pi/360, 2\pi/360, 3\pi/360$
....., 2π 处的值?

```
clear, clc
tic
dx = pi/360;
nx = 1 + 2*pi/dx;
for i = 1:nx
    x(i) = (i-1)*dx;
    y(i) = sin(3*x(i));
end
t1=toc
```

```
clear
tic
x = 0:pi/360:2*pi;
y = sin(3*x);
t2=toc
```

采用向量化运算方式通常可以提高程序效率!



嵌套循环

for循环可以嵌套使用

```
for 循环变量 = 表达式1(初值):表达式2(步长):表达式3(终值)
    statements （语句组）
    for 循环变量2 = 表达式1:表达式2:表达式3
        statements （语句组）
    end
end
```

利用for循环编程计算以下表达式的值

$$\sum_{m=1}^5 \sum_{n=1}^5 m / n^2$$

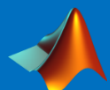
```
a=0
for m=1:5
    for n=1:5
        a=a+m/n^2
    end
end
```



while循环结构

```
while condition (表达式)  
    statements (执行语句组)  
end
```

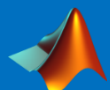
- ✓ while循环用于循环次数未知的情况；
- ✓ 当执行while...end循环时会首先测试condition（条件表达式）。如果condition为False(零)，则直接跳出循环，执行end后面的语句。如果condition为True(非零)，则执行语句组statements，然后退回到while语句再测试条件；
- ✓ 根据表达式的值，while循环体中的语句可能不会执行或执行无限次（死循环）。



例题

利用while循环，求满足 $\sum_{n=1}^m n^2 > 10^4$ 的最小整数m

```
function WhileDemo  
a=0;n=1;  
while a<=1e4  
    a=a+n^2;  
    n=n+1;  
end  
m=n-1, a
```



例题

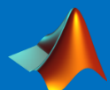
以下程序运行后，屏幕显示结果分别为

5 15

```
for x=1:5  
    y=x^2-2*x;  
end  
disp([x,y])
```

6 15

```
x=1;  
while x<=5  
    y=x^2-2*x;  
    x=x+1;  
end  
disp([x,y])
```



控制语句-break

break语句通常置于for循环或while循环内，根据条件执行break语句，以直接退出最内层的for循环或while循环。

利用for循环，求

满足 $\sum_{n=1}^m n^2 > 10^4$ 的

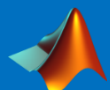
最小整数m

```
function ForBreakDemo
a=0;
for i=1:1000
    a=a+i^2;
    if a>10000
        break
    end
end
m=i
```



return语句

- 在某个函数内部执行return语句时，可立即退出该函数，并返回到调用它的函数，继续运行；
- 当程序已完成return所在函数的任务并可直接返回时，return是非常有用的；
- return通常放置于函数内的一个控制结构（如if语句）内；
- error可视为异常终止函数，而return则是成功后正常终止。



例题

使用直径 $D=1\text{ mm}$ 的铜球在一种油中进行沉降实验，测得铜球的沉降速度 $u_t=1.5\text{ cm/s}$ 。已知铜球的密度为 $\rho_s=8.9\text{ g/cm}^3$ ，油的密度 $\rho_f=0.85\text{ g/cm}^3$ ，求该油的粘度 μ_f 为多少？

模型：定义颗粒直径为基准的雷诺数，在不同的雷诺数范围内，沉降速度可以按如下式计算：

$$u_t = d_p^2 (\rho_p - \rho_f) g / 18 \mu_f \quad \text{Re}_p \leq 2$$

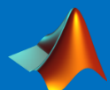
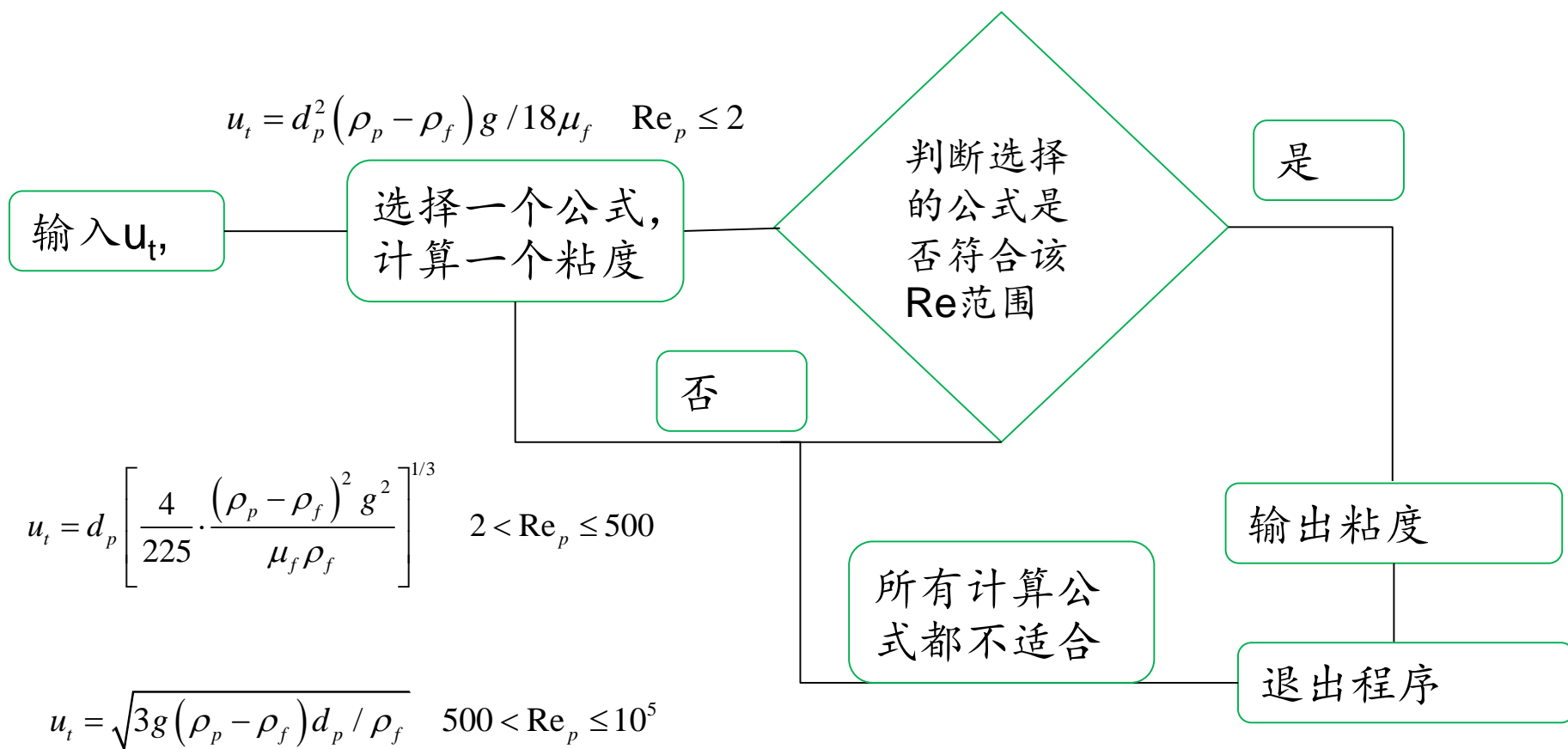
$$u_t = d_p \left[\frac{4}{225} \cdot \frac{(\rho_p - \rho_f)^2 g^2}{\mu_f \rho_f} \right]^{1/3} \quad 2 < \text{Re}_p \leq 500$$

$$u_t = \sqrt{3g(\rho_p - \rho_f)d_p / \rho_f} \quad 500 < \text{Re}_p \leq 10^5$$

其中 g 为重力加速度 9.8 m/s^2 。

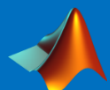


例题



例题

```
function SettleForVis
%Detemine viscosity by sedimentation experiment
dp=1e-3;ut=0.5;rs=8.9e3;rf=850;g=9.8;%using SI units
miu= dp^2*(rs-rf)*g/(18*ut);
Re=dp*ut*rf/miu;
if Re<=2
    fprintf('The Re is %.2f\n',Re)
    disp(['The viscosity of the fluid is ',num2str(miu),' PaS'])
    return
else
    miu=(dp/ut)^3*4/225*(rs-rf)^2*g^2/rf;
    Re=dp*ut*rf/miu;
    if Re>2&Re<500
        fprintf('The Re is %.2f\n',Re)
        disp(['The viscosity of the fluid is ',num2str(miu),' PaS'])
        return
    else
        disp('The viscosity could not be determined')
    end
end
end
```



例题

在一串联连续搅拌釜式反应器中进行等温液相反应， $A \rightarrow R$ ，
反应速率：

$$-r_A = kC_A^2, k = 0.2$$

假定10个反应釜具有相同的体积 $V=2.0$ L。当第一个反应釜入口浓度为 1.0 mol/L，进料速率 $v=0.5$ L/min。编写一个函数计算并在屏幕输出各个反应釜出口浓度。

对每个反应釜写出物料平衡方程： $vC_{i-1} - vC_i = VkC_i^2$

可以求解出每个釜中反应物浓度为：

$$C_i = \frac{-1 + \sqrt{1 + 4k\tau C_{i-1}}}{2k\tau}$$

其中 $\tau=V/v$ 表示反应物在釜中的平均停留时间。



例题

```
function CSTR
```

```
V=2.0;C0=1.0;v=0.5;k=0.2;
```

```
tao=V/v;
```

```
C(1)=(-1+sqrt(1+4*k*tao*C0))/(2*k*tao);
```

```
disp(['The outlet concentration of 1 tank is ',num2str(C(1))])
```

```
for i=2:10
```

```
    C(i)=(-1+sqrt(1+4*k*tao*C(i-1)))/(2*k*tao);
```

```
    disp(['The outlet concentration of ', num2str(i), ' tank is...'  
,num2str(C(i))])
```

```
end
```

可以将for循环去除，采用以下语句代替：

```
C2=(-1+sqrt(1+4*k*tao*C1))/(2*k*tao);
```

```
C3=(-1+sqrt(1+4*k*tao*C2))/(2*k*tao);
```

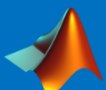
```
C4=(-1+sqrt(1+4*k*tao*C3))/(2*k*tao);
```

```
C5=(-1+sqrt(1+4*k*tao*C4))/(2*k*tao);
```

```
.....
```

```
C10=(-1+sqrt(1+4*k*tao*C9))/(2*k*tao);
```

for循环结构适用于循环次数已知时的重复运算。



例题

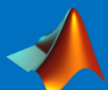
在串联连续搅拌釜式反应器中进行等温液相反应， $A \rightarrow R$ ，
反应速率： $-r_A = kC_A^2, k = 0.2$

假定所有反应釜具有相同的体积 $V=2.0$ L。当第一个反应釜入口浓度为 1.0 mol/L，进料速率 $v=0.5$ L/min。编写一个函数计算并需要多少个釜才能使A的转化率超过80%。

对每个反应釜写出物料平衡方程： $vC_{i-1} - vC_i = VkC_i^2$

可以求解出每个釜中反应物浓度为：
$$C_i = \frac{-1 + \sqrt{1 + 4k\tau C_{i-1}}}{2k\tau}$$

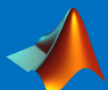
其中 $\tau=V/v$ 表示反应物在釜中的平均停留时间。



例题

与上例不同的是：本例中反应釜的个数未知，因此需要重复计算的次数未知，此时可以采用while循环结构完成。

```
function CSTRDesign
V=2.0;C0=1.0;v=0.5;k=0.2;
tao=V/v;
i=1;%设置一个循环变量
C(i)=(-1+sqrt(1+4*k*tao*C0))/(2*k*tao);
while (C0-C(i))/C0<0.8
    i=i+1; %每次循环，使循环变量递增
    C(i)=(-1+sqrt(1+4*k*tao*C(i-1)))/(2*k*tao);
end
disp(['The number of tank needed to reach 80% conversion is',num2str(i)])
disp('The outlet concentration of each tank is:')
disp(C)
```



例题

例题18 编写一个MATLAB函数使其可以选择不同的状态方程计算4536 mol氮气在200°C和0.4248 m³体积时的压力。

函数中可以使用的状态方程包括：

理想气体状态方程： $P = RT / V$

RK方程：

$$P = \frac{RT}{V-b} - \frac{a}{T^{0.5}V(V+b)} \quad a = 0.42748R^2T_c^{2.5} / P_c \quad b = 0.08664RT_c / P_c$$

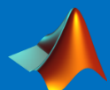
PR方程：

$$P = \frac{RT}{V-b} - \frac{a(T)}{V(V+b)+b(V-b)} \quad a(T) = 0.45724 \frac{R^2T_c^2}{P_c} \alpha(T)$$

$$m = 0.37464 + 1.5422\omega - 0.26992\omega^2$$

$$(\alpha(T))^{0.5} = 1 + m(1 - (T/T_c)^{0.5}) \quad b = \frac{0.0778RT_c}{P_c}$$

其中，P、T分别表示体系的压力（Pa），和温度（K）；V为摩尔体积（m³/mol）、R为理想气体常数，8.314 J/(mol·K)；T_c和P_c分别为物质的临界温度和压力，对于氮气其值分别为126.2 K和 3.39×10⁶ Pa；ω为偏心因子，其值为0.037。

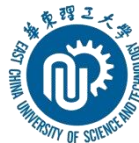


例题

```
function Cha01Demo20
%This function calculate P by using different EOS
V=0.4248; T=200+273.15; n=4536; %the known conditions
Vm=V/n;%the molar volume
R=8.314;
Tc=126.2; Pc=3.39e6; omiga=0.037
disp('Please choose which EOS should be used')
disp('1 - ideal gas')
disp('2 - RK')
disp('3 - PR')
EOS=input('EOS=');
```



例题

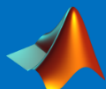


```
switch EOS
case 1
    P=R*T/Vm;
    disp(['The pressure is ',num2str(P/101325),' atm'])
case 2
    a=0.42748*R^2*Tc^2.5/Pc;
    b=0.08664*R*Tc/Pc;
    P=R*T/(Vm-b)-a/(T^0.5*Vm*(Vm+b));
    disp(['The pressure is ',num2str(P/101325),' atm'])
case 3
    m=0.37464+1.5422*omiga-0.26992*omiga^2;
    alpha=(1+m*(1-(T/Tc)^0.5))^2;
    a=0.45724*R^2*Tc^2/Pc*alpha;
    b=0.0778*R*Tc/Pc;
    P=R*T/(Vm-b)-a/(Vm*(Vm+b)+b*(Vm-b));
    disp(['The pressure is ',num2str(P/101325),' atm'])
otherwise
    disp('Wrong input!')
end
```

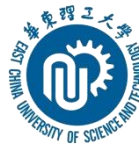


本讲小结

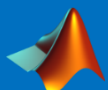
- 1) 关系运算符：六种
- 2) 逻辑运算符：六种
- 3) 各种运算的优先级：括号>算术运算符>冒号运算符>关系运算符>逻辑运算符
- 4) 程序流程控制语句：if, switch, while, for,



作业

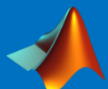


公共邮箱下载文档：work04.pdf，直接打印
、完成后上交



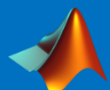
课堂练习

1. $[1\ 3; 2\ 4] * [1; 1]$ 的计算结果为_____。
2. $3 - -3$ 的计算结果为_____； $3 - + -3$ 的计算结果为_____。
3. $\text{find}(0:90:360) == 0$ 的计算结果为_____。
4. 变量 $a=2$ 表示 a 的数据类型为_____，而 $a='2'$ 表示 a 的数据类型为_____； $a\{1,1\}=[1, 2; 3, 4]$ 表示 a 为_____； $a(1).\text{Name}$ 表示 a 为_____。
5. 采用匿名函数计算 $a = 2, b = 3, f(x) = a \sin(b x) + b \cos(a x)$ 在 $x=1$ 时的值，其MATLAB命令为：_____, _____。



课堂练习

6. 在函数中定义全局变量A, B, C的命令为_____。
7. 在MATLAB中运行以下命令, 则得到屏幕显示为: _____。
- ```
>>A=0.0:0.5:3;
>>fprintf('\t%.2f' , A)
```
8. 已知time, con, sel都是1行10列的向量, 以下plot命令使用错误的是:
- A) plot(time,con,':kd')
  - B) plot(time,con,'b:k')
  - C) plot(time,con,'p.-')
  - D) plot(time,con,'b-',time,sel,'b-')
  - E) plot(time,[con,sel],'-\*')



# 课堂练习

9.按以下要求编写一个函数计算：

$$A = \frac{y}{x} + \sin(45^\circ) \times x$$

的值，其中 $x > 0$ 时， $y = \sqrt[3]{x}$ ； $x < 0$ 时， $y = 2/x$ ； $x = 0$ 时，返回错误信息（ $x$  can't be zero）。

要求：1)主函数名称为excer1， $x$ 作为输入变量， $A$ 作为输出变量；2)主函数中包括一个子函数myfun用于计算 $y$ 的值。

