第三章 M文件初步

➢前面的章节介绍了如何在MATLAB操作桌面完成运算，适合于指令条数不太多的情况。如果用户需要用到多条指令或者需要经常这些指令，则采用编写程序的方式完成起来会更方便。

- MATLAB为用户专门提供了M文件，可以让用户自行将指令写成程序，存储为文件后运行完成相应的工作。

M文件简介

- M文件分为脚本文件(Script File，标识为)和函数文件(Function File，标识为)两种形式，它们的扩展名都是.m。脚本文件的效果等同于将指令逐条输入指令窗执行，因此，脚本文件在执行时，用户可以查看或者调用工作空间中的变量。函数文件则需要通过输入宗量和输出宗量来传递信息，如果没有特别设置，函数文件犹如一个暗箱，函数文件中的中间变量在工作空间是看不见的，并随着函数文件执行结束而擦除。MATLAB强大的科学技术资源来自于MATLAB内部储存了丰富的函数文件，日益丰富的函数文件资源也是MATLAB版本升级的基础。

M文件简介

- 本章将系统介绍M文件的种类、函数文件的构造，以及MATLAB程序的跟踪调试。

本章主要内容包括：

- M文件的分类及其编写入门
- MATLAB中不同流程控制语句及其使用技巧
- 不同类型函数文件的创建及其调试

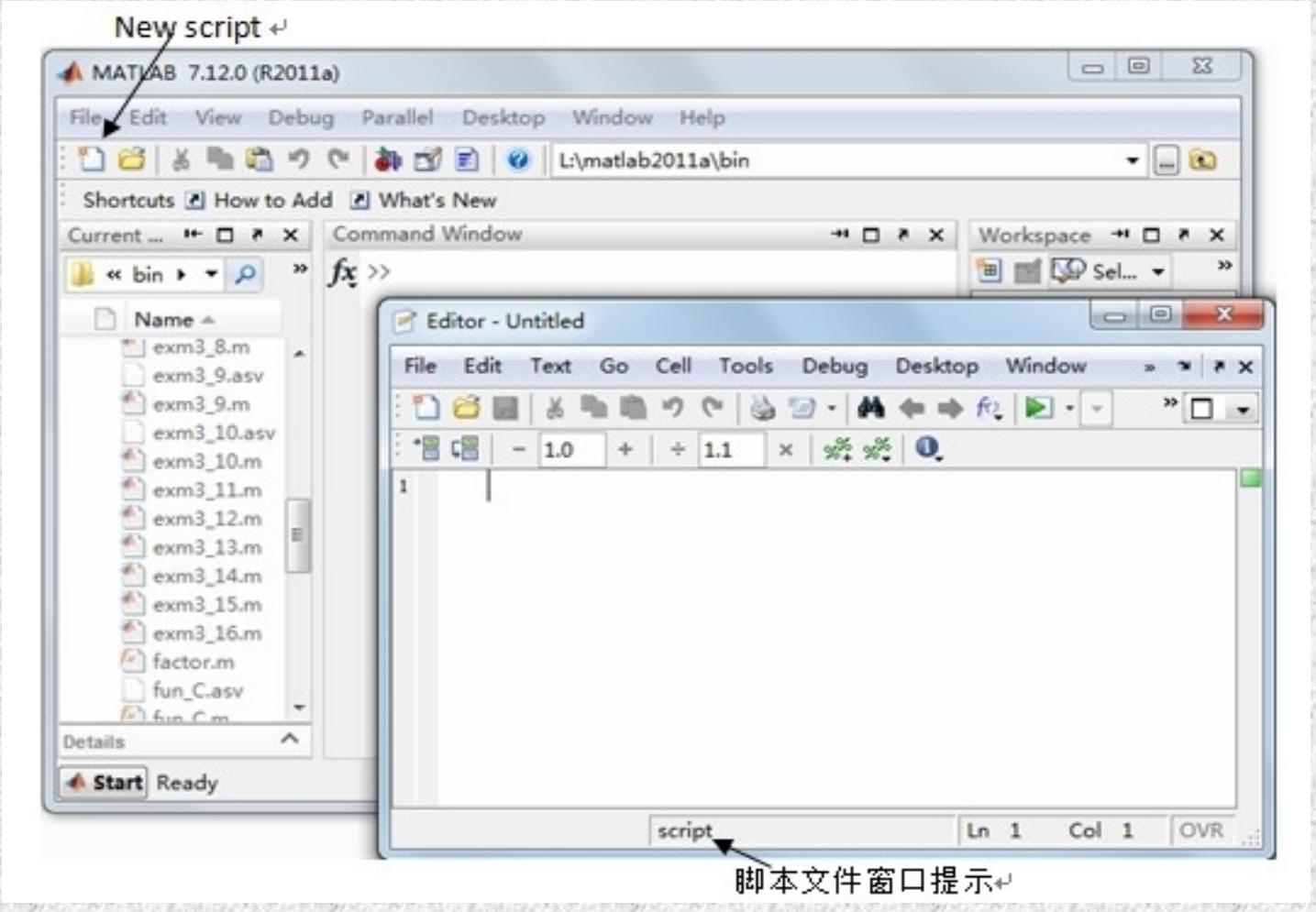
3.1 M文件入门

3.1.1 M文件的建立

- M文件是个文本文件，可以用任何编辑程序来建立和编辑，一般最常用且最方便的是使用MATLAB提供的文件编辑器(MATLAB Editor/Debugger)。
- 缺省情况下，M文件编辑器不随MATLAB的启动而开启，只有编写M文件才能启动。为建立新的M文件，启动MATLAB文件编辑器有3种方法

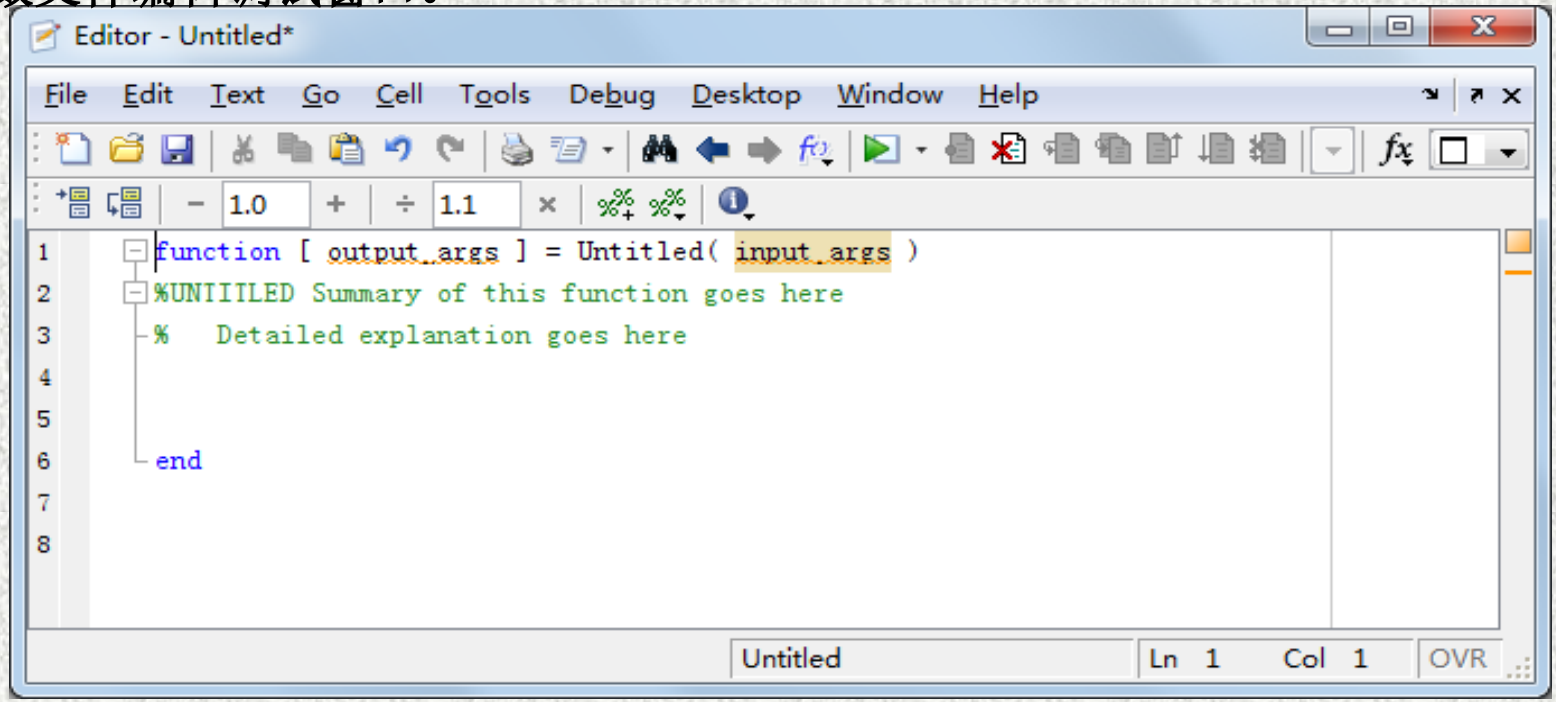
（1）命令按钮操作

鼠标左键单击MATLAB主窗口工具栏上的New Script命令按钮，则会出现标准的MATLAB函数文件编辑窗口



（2）菜单操作

从MATLAB主窗口中选择File→New→Script，出现MATLAB脚本文件编辑窗口；如果希望新建函数文件，则选择File→New→Function，则出现标准的MATLAB函数文件编辑调试窗口。



（3）命令操作：在MATLAB命令窗口执行指令edit，MATLAB将启动新建图3.1的脚本文件编辑窗口。

3.1.2 M文件编写初步

本节将通过编写脚本文件和函数文件执行相同的运算。通过这两个文件，用户可以初步了解脚本文件和函数文件。对于其中涉及的语言结构，文章后面将有详细介绍。

【例3-1】编写一个脚本文件，计算 $\sum_{n=1}^{20} n$

- 编写脚本文件的步骤：

（1）启动M文件编辑调试器，建立新的脚本文件。

（2）编写程序并保存

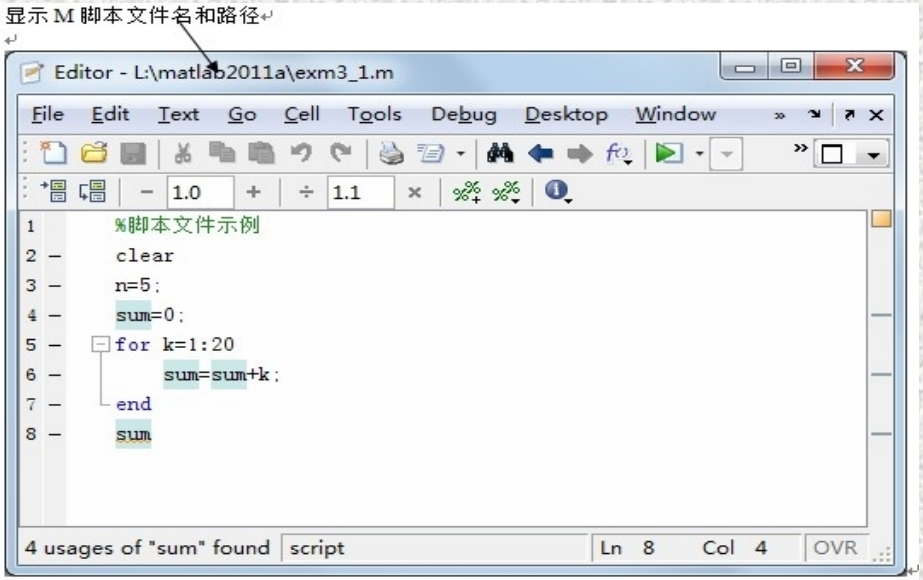
在脚本文件空白处输入指令：

```
%脚本文件示例
    clear
    n=20;
    sum=0;
    for k=1:n
        sum=sum+k;
    end
    sum
```

【例3-1】编写一个脚本文件，计算 $\sum_{n=1}^{20} n$

指令输入完毕后保存该文件：

- 单击M文件编辑调试器工具条上的保存图标 ,便出现标准的文件保存对话框。在文件保存对话框中选定目录，键入程序的文件名（如exm3_1）,单击保存即可；
- 或者从M文件编辑调试器窗口中选择File→Save或者Save As也可以保存。保存完毕后原来的未命名(Untitled)M文件立即显示输入的文件名和路径。



【例3-1】编写一个脚本文件，计算 $\sum_{n=1}^{20} n$

（3）运行脚本文件

设置脚本文件exm3_1.m所在目录为当前目录，或者让该目录处于MATLAB的检索路径上，在指令窗中输入文件名exm3_1并运行。

```
>> exm3_1
sum =
   210
```

【说明】：

◆ 运行M文件的操作方法有很多，最常用的方法有：

（1）在指令窗中运行M文件，M文件不带扩展名。

（2）在当前目录窗中，用鼠标右键单击待运行文件，再从引出的现场菜单中选择【Run】即可。

◆ 在M文件编辑器中，注释部分可以采用汉字，并总可以获得准确显示。

◆ 当使用M文件便器调试器保存文件时，不必写出文件的扩展名。

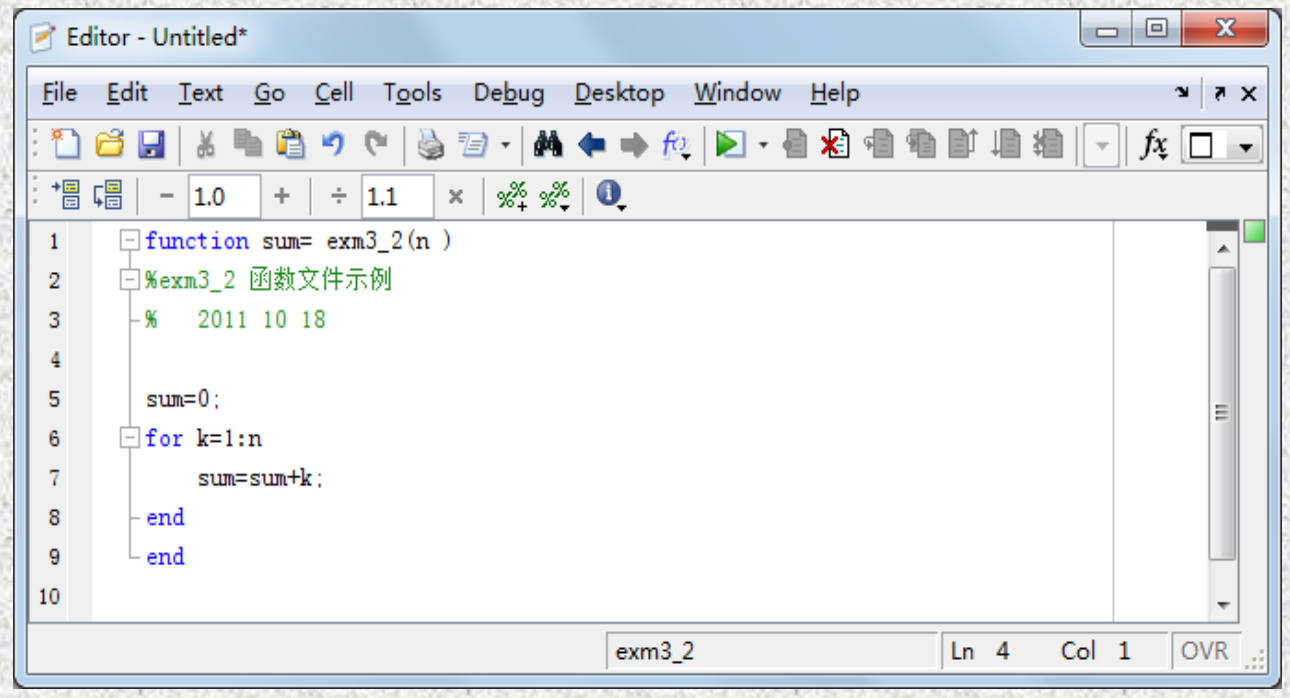
【例3-2】运用M函数文件编写例3-1中的求和计算，要求n为任意自然数。

编写函数文件的步骤：

（1）启动M函数文件编辑调试器，建立新的函数文件。

（2）编写程序并保存

将输入宗量(input args)、输出宗量(output args)和函数名等处做相应修改保存该函数文件，文件名为exm3_2。



【例3-2】运用M函数文件编写例3-1中的求和计算，要求n为任意自然数。

（2）运行函数文件

在MATLAB指令窗中输入指令并运行：

```
>> sum=exm3_2(20)
sum =
   210
```

所得结果与上题相同。

说明：

- 在指令窗中执行函数文件时必须事前在工作空间中产生输入宗量的值，或者直接将输入宗量的值键入函数名后面的小括号中。用户如果像执行脚本文件一样执行函数文件，即仅键入exm3_2，则MATLAB将提示出错。
- 函数文件在保存时，MATLAB将函数名默认为文件名，建议用户不要随意修改。一旦函数文件的文件名和函数名不一致时，MATLAB将以函数文件的文件名为准来执行。
- 用户可以通过在脚本文件中使用交互式输入指令input()，达到和函数文件相似的执行效果。建议用户将脚本文件exm3_1中的语句"n=20"修改为"n=input('n=?');"，重新运行后观察其执行情况。

3.2 MATLAB流程控制结构

- MATLAB为用户提供了4种流程控制结构：条件结构、循环结构、开关结构和试探结构。
- 用户可以根据某些判断结果来控制程序流的执行次序。
- 与其他程序语言相比，除了试探结构为MATLAB所特有外，其他结构和用法都十分相似，因此本节只结合MATLAB特点对这几种流程控制结构做简要说明。

3.2.1 if条件结构

If条件结构是实现分支结构程序的最常用的一种语句，能够实现单分支、双分支和多分支结构。

- 单分支结构的形式：

```
if 逻辑表达式
 指令语句组
    end
```

- 双分支结构的形式：

```
if  逻辑表达式
      指令语句组1
    else
      指令语句组2
    end
```

3.2.1 if条件结构

- ✓ 多分支结构的形式：

```
if  逻辑表达式1
        指令语句组1
    else if  逻辑表达式2
            指令语句组2
 …      …
    else if   逻辑表达式n
            指令语句组n
    else
            指令语句组n+1
    end
```

3.2.1 if条件结构

【说明】：

- 在单分支结构中，当逻辑表达式为“逻辑真”（非0）时，则执行相应的指令语句组，否则，则跳过该指令组。对于双分支结构，当逻辑表达式为“逻辑真”时，则执行指令语句组1，否则，执行指令语句组2；多分支结构中，MATLAB将依次判断逻辑表达式是否为“逻辑真”，当前面所有的逻辑表达式都为“逻辑假”（0）时，MATLAB执行指令语句组n+1，并结束该结构。

- 逻辑表达式的形式大致有两种：一种是以标量或者数组形式；另一种是由多个逻辑表达式进行关系运算的形式。当形式为数组时，只有当数组中的所有元素都为“逻辑真”时，MATLAB才会执行对应条件的语句。后者的关系运算主要为“与（&）”“或（|）”运算。

【例3-3】求分段函数 $y=\begin{cases} 2\sqrt{x}, & 0\le x\le 1 \\ 1+x+\ln x & x>1 \end{cases}$ 的值。

编写文件名为exm3_3的脚本文件：

```
clear
x=input('请输入x=?');
if x>=0 & x<=1
    y=2*sqrt(x)
elseif x>1
    y=1+x+log(x)
else
   disp('No defination')
end
```

【例3-3】求分段函数 $y=\begin{cases} 2\sqrt{x}, & 0\le x\le 1 \\ 1+x+\ln x & x>1 \end{cases}$ 的值。

在指令窗中执行文件exm3_3.m，运行结果为：

请输入x=?

用户将需要计算的数值送给MATLAB,这里以x=6为例：

请输入x=?6

y =

8.7918

3.2.1 switch-case开关结构

MATLAB从5.0版本开始，提供了switch-case开关结构，调用格式为：

```
 switch  开关表达式
case    表达式1
          指令语句组1
case    表达式2
          指令语句组2
…           …
case    表达式n
          指令语句组n
otherwise
          指令语句组n+1
end
```

【说明】：

<1>开关结构在运行时，MATLAB将开关表达式的值依次和各个case后面的表达式进行比较，如果是“逻辑真”，将执行相应的指令语句，如果是“逻辑假”，则取下一个case后面的表达式比较。如果所有case后面表达式都与开关表达式的值不相等，则执行otherwise后面的指令语句组。

<2>开关表达式形式有两种：一种是标量；另外一种为字符串。对于字符串形式，MATLAB在比较时将调用函数strcmp(),得出字符串比较的逻辑输出值，MATLAB根据该逻辑值的真假来判断是否执行该case后面的语句。

【例3-4】switch-case开关结构实例：

- 通过键盘输入百分制成绩，输出成绩的等级，其中90~100分等级为A，80~89分等级为B，70~79分等级为C，60~69分等级为D，60分以下等级为E。
- 编写文件名为exm3_4的脚本文件：

【例3-4】switch-case开关结构实例：

```
n=input('请输入百分制成绩n=?');
if n<0 | n>100
   disp ('输入有误，请重新输入百分制成绩')
else
    t=fix(n/10);%fix()为截断取整函数
  switch t
    case {9,10}
        disp('A')
    case 8
        disp ('B')
   case 7
        disp('C')
    case 6
        disp('D')
    otherwise
      disp ('E')
  end
  end
```

【例3-4】switch-case开关结构实例：

- 在指令窗中执行文件exm3_4.m，并以n=86分为例，运行结果为：

请输入百分制成绩n=?86

B

【例3-5】求分段函数的值

$$y=\begin{cases}\dfrac{\sqrt{1+x^2}}{\ln 3}, x\le 1, \text{且} x\ne -3\\ \dfrac{x+\lg 7}{e}, x>2\\ \dfrac{x}{\pi}, \text{其他}\end{cases}$$

编写文件名为exm3_5的脚本文件：

方法一：利用switch-case开关结构:

```
clear
x=input('请输入x的值：');
switch 1
    case x<=1& x~=-3
        y=sqrt(1+x^2)/log(3)
    case x>2
        y=(x+log10(7))/exp(1)
    otherwise
        y=x/pi
end
```

注意：**程序中switch后面的1为逻辑1，表示条件永真。**该题为三分支结构问题，可以用 if条件结构来编写。

方法二：利用 if条件结构

```
clear
x=input('请输入x的值：');
if x<=1&x~=-3
        y=sqrt(1+x^2)/log(3)
elseif x>2
        y=(x+log10(7))/exp(1)
else
        y=x/pi
end
```

3.2.3 try试探结构

MATLAB从5.2版本开始添加了这一新的结构，其一般格式如下：

```
try
  指令语句组 1
catch
 指令语句组 2
end
```

【说明】：试探结构首先试探性地执行指令语句组1，如果在此语句组执行过程中出现错误，则将错误信息赋给保留的lasterr变量，并放弃这组语句，转而执行语句组2中的语句。若语句组2执行过程中又出现错误，MATLAB将终止该结构。

3.2.3 try试探结构

【例3-6】对3阶魔方阵的行进行引用，当行下标超出魔方阵的最大行数时改为对最后一行的引用，并显示出错警告信息。

编写文件名为exm3_6的脚本文件：

```
N=input('提取魔方阵的第N行元素，其中N=?');
A=magic(3); %执行函数magic()产生3×3魔方矩阵
try
   A_N=A(N,:) %取A的第N行
catch
A_end=A(end,:) %取A的最后一行
end
```

3.2.3 try试探结构

在指令窗中执行文件exm3_6.m，并以N=6为例，运行结果为：

```
取魔法阵的第N行，其中N=?6
A_end =
     4     9     2
ans =
Attempted to access A(6,:); index out of bounds because
     size(A)=[3,3].
```

3.2.4 for循环结构

for语句是一种基本的实现循环结构的语句，能够以确定的次数执行某一段程序。

for语句的格式如下：

```
for 循环变量=表达式
    循环体指令语句组
end
```

【说明】

“表达式”可以是MATLAB指令产生的数组，也可以是任意给定的一个数组。

循环变量从“表达式”中的第一个数值（或第一列数组）一直循环到“表达式”的最后一个数值（或最后一列数组）。

【例3-7】运用for...end循环结构计算 $\sum_{n=0}^{m}\frac{1}{2^n+1}$ 的值，（其中m=100）

运用for...end循环结构计算，编写文件名为exm3_7的脚本文件：

```
clear
y=0;
for n=0:100
    y=y+1/(2^n+1);
end
y
```

在指令窗中执行文件exm3_7.m，运行结果为：

```
y =
   1.2645
```

【例3-8】运用循环结构计算积分 $s=\int_0^3 \frac{\ln x}{2x^2+1}dx$

```
a=0+eps;%利用极小数代替0，避免计算时出错
b=3;n=1000;h=(b-a)/n;
x=a:h:b;y=0;
f=log(x)./(2*x.^2+1);
for i=1:n
    s(i)=(f(i)+f(i+1))*h/2; %计算细分后梯形的面积
    y=y+s(i); %迭代法
end
y
```

在指令窗中执行该文件，运行结果为：

```
y =
   -0.7730
```

【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。运用for...end循环结构输出全部水仙花数。

```
clear
for m=[100:999]
  m1=fix(m/100);          %求m的百位数字,函数fix()功能为截断取整
  m2=rem(fix(m/10),10);   %求m的十位数字,rem()为求余函数
  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
```

运行结果为：

```
   153
   370
   371
   407
```

3.2.5 while循环结构

while结构是MATLAB语言实现循环结构的另一种基本方式，以不定的次数重复执行某一段程序。

while循环结构的一般形式如下：

```
while  逻辑表达式
       指令语句组
end
```

- **执行时，只要逻辑表达式为“逻辑真”（非0），就执行指令语句组，执行后再返回到while引导的逻辑表达式处，继续判断；如果逻辑表达式为“逻辑假”（0），则跳出循环。**
- **通常，逻辑表达式的值为一个标量，但数组也同样有效。如果逻辑表达式的值为数组，要求数组的所有元素都为“逻辑真”（非0），MATLAB才会执行循环体中的指令语句组。如果逻辑表达式为“空”，MATLAB则认为表达式的值为逻辑假，不执行循环体语句指令组。**

【例3-9】运用while...end循环结构实例：从键盘输入若干个数，当输入为0时结束输入。求这些数的平均值和它们的和。

```
clear
val=input('Enter a number (end in 0):');
sum=0;cnt=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
```

用户可以在指令窗中执行该文件，将数字逐个输入并获得运行结果。

与循环语句有关的语句还有break和continue语句，它们一般和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，程序将跳出该循环；continue语句控制跳过循环体内的某些语句，当在循环体内执行到该语句时，程序将跳过本次循环，继续下一次循环。

【例3-10】求[200,300]间第一个能被18整除的整数。

编写文件名为exm3_10的脚本文件：

```
clear
for n=200:300
    if rem(n,18)~=0
        continue
    end
    break
end
n
```

在指令窗中执行文件exm3_10.m，运行结果为：

```
n =
  216
```

3.2.6 控制程序流的其他常用指令

MATLAB中有几个指令能够使用户所编辑的程序具有交互性，使用这些指令有助于用户对程序做调试。

1.input()和keyboard

(1)input()指令

input()指令常用格式为：

v=input('message')

v=input('message','s')

- input()指令将MATLAB的“控制权”暂时交给用户。当用户通过键盘从指令窗中输入数据并执行后，所输入的内容被保存在变量v中，同时“控制权”被交还给MATLAB。
- message为显示在指令窗中的字符串，用户在指令窗中通过该提示给变量v赋值。第一种格式中，用户可以输入数值、字符串、元胞等各种形式的数据形式，变量v保持原来数据形式不变；第二种格式中无论用户输入什么类型的数据形式，变量v总是被看做是字符串。

3.2.6 控制程序流的其他常用指令

(2)keyboard指令

当程序遇到该指令时，MATLAB将“控制权”交给用户，用户从键盘上输入各种MATLAB指令。当用户执行指令return时，“控制权”交还给MATLAB。

2.disp()指令和echo指令

disp()指令的常用格式为：

disp(a)：显示变量a的结果，无须在指令窗中显示字符a。

echo指令控制函数文件程序是否需要在屏幕上显示：当MATLAB执行程序中遇到echo on指令时，会显示当前执行的程序内容；echo off为解除echo on指令。

3.pause指令

pause指令的格式为：

pause：暂停执行程序，等待用户按任意键继续执行程序。

pause(n)：暂停n秒后，继续执行程序。

3.2.6 控制程序流的其他常用指令

4.警示指令

当程序执行中出现错误时，利用警示指令可以了解出错情况。常用的警示指令有：

lasterr：显示MATLAB自动判断的最新出错原因，终止运行程序。

lastwarn：显示MATLAB自动判断的最新出错原因，继续运行程序。

warning('message')：显示警告信息message，继续运行程序。

error('message')：显示警告信息message，终止运行程序。

errortrap：出错后，是否继续运行程序的切换开关。

3.2.7 加快MATLAB程序运行速度的技巧

由于MATLAB语言为解释性语言，因此，较其他语言来说，MATLAB语言运行速度不太理想，因此，提高MATLAB程序运行效率就显得尤为重要。本文通过一个例题给用户一个具体的体会。

【例3-11】提高MATLAB代码运行效率实例：采用不同编程方法计算1-100000的平方根，比较运行时间。

3.2.7 加快MATLAB程序运行速度的技巧

编写文件名为exm3_11的脚本文件：

```
clear
nm=100000;
% 循环变量为标量时，显示运行时间
tic %打开定时器
for n=1:nm %循环变量为标量
    a(n)=sqrt(n);
end
t1=toc %显示定时器的时间
% 循环变量为向量时，显示运行时间
clear a in
tic;
in=[1:nm];
for n=in %循环变量为向量
    a(n)=sqrt(n);
end
t2=toc
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/335001304142012011