



Matlab程序设计

第6讲



5.3 函数文件

函数文件是另一种形式的M文件，每一个函数文件都**定义一个函数**。
Matlab提供的标准函数大部分是由函数文件定义的。

5.3.1 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。
当输出形参多于一个时，应该用**方括号**括起来。



说明：

1. 关于函数文件名

函数文件名通常由函数名再加上扩展名.m组成。

当函数文件名与函数名不同时，Matlab将忽略函数名而确认文件名
因此调用时使用函数文件名。

2. 关于注释说明部分

注释说明包括3部分：

① 紧随引导行之后以%开头的第一注释行。

这一行一般包括大写的函数文件名和函数功能简要描述，供lookfor
关键词查询和help在线帮助时使用。

② 第一注释行及之后连续的注释行。

通常包括函数输入/输出参数的含义及调用格式说明等信息，构成全
部在线帮助文本。

说明：

③ 与在线帮助文本相隔一空行的注释行。

包括函数文件编写和修改的信息，如作者和版本等。

3、关于return语句

如果在函数文件中插入了return语句，则执行到该语句就结束函数的执行，流程转至调用该函数的位置。通常也不使用return语句。

例5.10 编写函数文件，求半径为r的圆的面积和周长。

函数文件如下：

```
function [s,p] = fcircle(r)
```

```
% FCIRCLE calculate the area and perimeter of a circle of radii r
```

```
% r    圆半径
```

```
% s    圆面积
```

```
% p    圆周长
```

```
%2006年2月30日编
```

```
s = pi*r*r;
```

```
p = 2*pi*r;
```



说明:

将以上函数文件以文件名fcircle.m保存，然后在命令窗口调用。

```
[s,p] = fcircle(10)
```

输出结果是:

```
s =
```

```
314.1593
```

```
p =
```

```
62.8319
```

采用help命令或lookfor命令可以显示出注释说明部分的内容。

```
help fcircle
```

屏幕显示

```
FCIRCLE calculate the area and perimeter of a circle of radii r
```

```
r    圆半径
```

```
s    圆面积
```

```
p    圆周长
```



5.3.2 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)

注意：函数调用时，各实参出现的顺序、个数，应与函数定义时相同。

例5.11 利用函数文件，实现直角坐标(x,y)与极坐标(ρ , θ)之间的转换。

函数文件：tran.m:

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

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
```

函数的嵌套调用

在Matlab中，函数可以**嵌套调用**，即一个函数可以调用别的函数。

一个函数调用自身称为函数的**递归调用**。

例5.12 利用函数的递归调用，求 **$n!$** 。

$n!$ 本身就是以递归的形式定义的：

$$n! = \begin{cases} 1, & n \leq 1 \\ n(n-1)!, & n > 1 \end{cases}$$

显然，求 **$n!$** 需要求 **$(n-1)!$** ，这时可采用递归调用。

函数如下：

```
function f = factor(n)
```

```
if n<=1
```

```
    f = 1;
```

```
else
```

```
    f = factor(n-1)*n; %递归调用求(n-1)!
```

```
end
```



函数的嵌套调用

在命令文件中调用该函数文件，求 $s = 1! + 2! + 3! + 4! + 5!$ 。

```
s = 0;  
for i = 1:5  
    s = s + factor(i);  
end
```

s

在命令窗口运行命令文件，结果如下：

```
s =  
    153
```


5.3.3 函数参数的可调性

Matlab在函数调用上有一个与一般高级语言不同之处：

函数所传递参数数目的可调性，即参数的数量可以改变。

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。

例5.13 nargin用法示例

函数文件examp.m:

```
function fout = chararray(a,b,c)
if nargin == 1
    fout = a;end
if nargin == 2
    fout = a+b;end
if nargin == 3
    fout = (a*b*c)/2;
end
```

命令文件:

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

5.3.4 全局变量与局部变量

Matlab中，函数文件中的变量是**局部变量**。

如在若干函数中，**都把**某一变量定义为**全局变量**，那么这些函数将共用这个变量。

全局变量的作用域是整个Matlab的工作空间，所有函数都可以对它进行存取和修改。

全局变量用**global**命令定义，格式为：

global 变量名

例5.13 全局变量应用示例。

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)

BETA = 2;

global ALPHA BETA

s = wadd(1,2)

f = ALPHA*x + BETA*y;

输出为：

在命令窗口中输入：

s =

global ALPHA BETA

5

ALPHA = 1;



5.4 程序调试

程序调试是程序设计的重要环节，也是程序设计人员必须掌握的重要技能。Matlab提供了相应的程序调试功能，即可以**通过文本编辑器**对程序进行调试，又可以**在命令窗口结合具体的命令**进行。

5.4.1 程序调试概述

一般说来，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，给出相应的错误信息，并标出错误在程序中的行号。例如：

输入下列程序：

```
A = 87;
```

```
B = 9.3;
```

```
C = A+*B;
```

系统将给出错误信息：

```
??? Error: File: Untitled1.m Line: 3 Column: 7
```

```
Unexpected MATLAB operator.
```

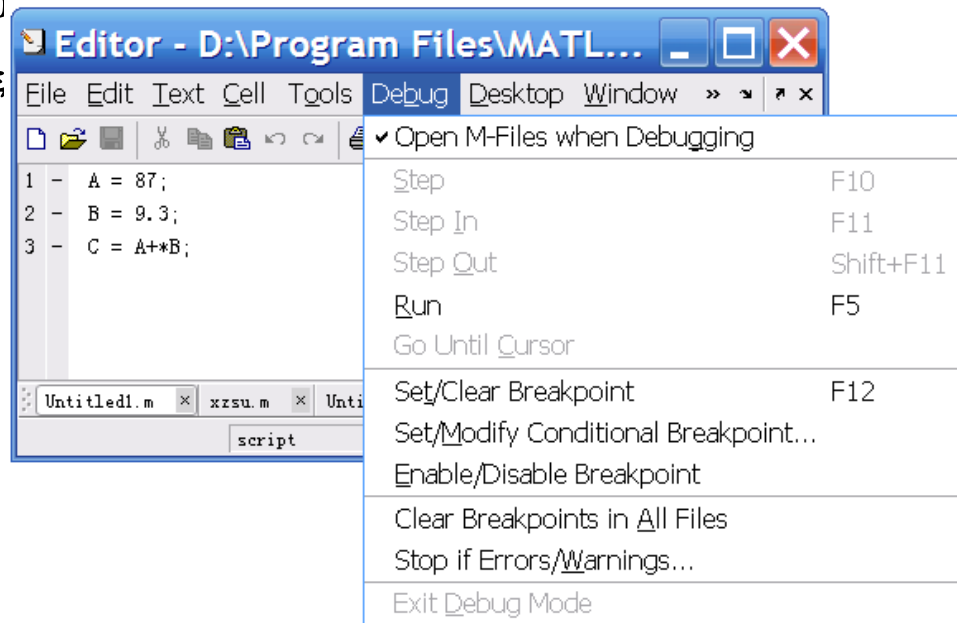
通过分析Matlab给出的错误信息，不难排查程序中的语法错误。

5.4.1 程序调试概述

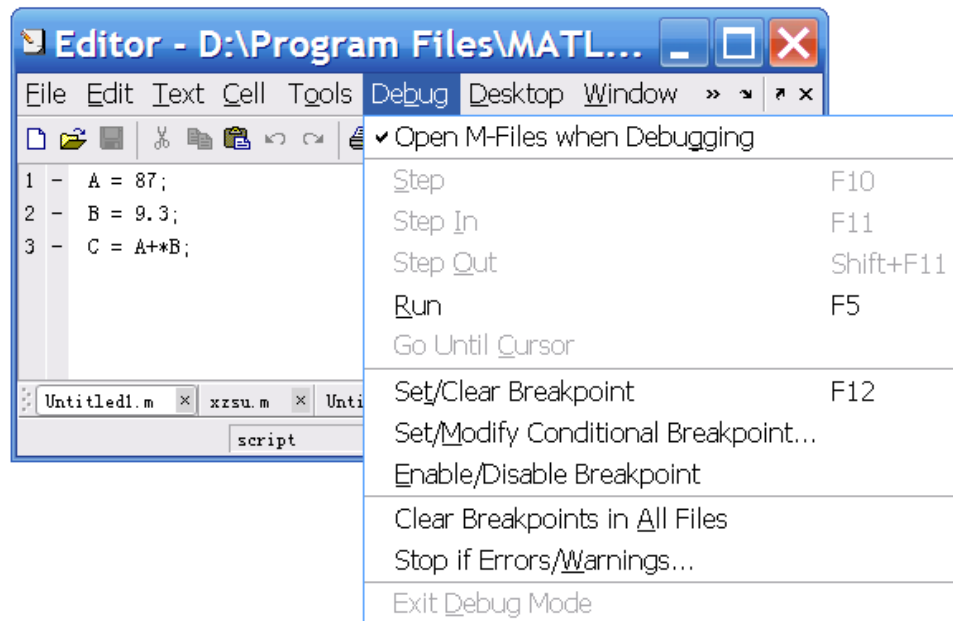
Matlab对**程序逻辑错误**是无能为力的，不会给出任何提示信息。可以通过调试手段来发现。

采取的方法如下：

- ① 将程序的一些**主要中间结果输出到命令窗口**，从而确定错误的区段。
- ② 使用Matlab的**调试菜单(debug)**，通过该菜单中的选项来实现程序调试。
- ③ 或使用**命令方式**来实现程序



5.4.2 Matlab调试菜单



1、控制单步运行

step: 单步运行，不进入函数；

step in: 单步运行，进入函数；

step out: 停止单步运行；

save and run: 存储文件并开始运行。

2. 断点操作

stop if error/warnings: 在程序执行出现错误或警告时，停止程序运行，进入调试状态。



5.5 Matlab矩阵分析与处理

5.5.1 特殊矩阵

常见的特殊矩阵有零矩阵、幺矩阵、单位矩阵等，这类特殊矩阵在应用中具有通用性。

1、通用的特殊矩阵

常用的产生通用特殊矩阵的函数有：

zeros：产生全0矩阵（零矩阵）。

ones：产生全1矩阵（幺矩阵）。

eye：产生单位矩阵。

rand：产生0~1间均匀分布的随机矩阵。

randn：产生均值为0，方差为1的标准正态分布随机矩阵。

产生(0,1)区间均匀分布随机矩阵使用rand函数

产生均值为0，方差为1的标准正态分布随机矩阵使用randn函数



例5.16 建立随机矩阵：

- ① 在区间[20,50]内均匀分布的5阶随机矩阵。
- ② 均值为0.6，方差为0.1的5阶正态分布随机矩阵。

命令如下：

$x = 20 + (50 - 20) * \text{rand}(5)$

x =

48.5039	42.8629	38.4630	32.1712	21.7367
26.9342	33.6940	43.7581	48.0641	30.5860
38.2053	20.5551	47.6544	47.5071	44.3950
34.5795	44.6422	42.1462	32.3081	20.2958
46.7390	33.3411	25.2880	46.8095	24.1667

$y = 0.6 + \text{sqrt}(0.1) * \text{randn}(5)$

5.5 矩阵结构变换

1、对角阵与三角阵

只有对角线上有非零元素的矩阵称为对角矩阵，在研究矩阵时，有时候需要将矩阵的对角线上的元素提取出来形成一个列向量，有时也需要用一个向量构造一个对角阵。

(1) 提取矩阵的对角线元素函数：diag

例如：

```
A = [1,2,3;4,5,6];
```

```
D = diag(A)
```

```
D =
```

```
1
```

```
5
```

diag函数还有一种形式：diag(A,k)提取第k条对角线的元素。

例如：

```
D1 = diag(A,1)
```

```
D =
```

```
2
```

```
6
```


5.5 矩阵结构变换

(2)构造对角矩阵

如果V是一个m个元素的向量，`diag(V)`将产生一个 $m \times m$ 对角矩阵，其主对角线元素即为向量V的元素。

例如：

```
diag([1,2,-1,4])
```

ans =

1	0	0	0
0	2	0	0
0	0	-1	0
0	0	0	4

例如：

```
diag(1:3,-1)
```

ans =

0	0	0	0
1	0	0	0
0	2	0	0
0	0	3	0



例5.17

建立一个 5×5 矩阵A，然后将A的第一行元素乘以1，第二行乘以2，...第五行乘以5。

解：

用一个对角矩阵左乘一个矩阵时，相当于用对角阵的第一个元素乘以该矩阵的第一行，依次类推。

命令如下：

```
A = ones(5);
```

```
D = diag(1:5);
```

```
D * A
```

```
ans =
```

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5

5.6 矩阵求逆与线性方程组求解

5.6.1 矩阵的逆

对于一个方阵A，如果存在一个与其同阶的方阵B，使得：

$$A \cdot B = B \cdot A = I \quad (I \text{ 为单位矩阵})$$

则称B为A的逆矩阵，当然，A也是B的逆矩阵。求方阵A的逆矩阵可调用函数**inv(A)**。

例5.18 求方阵A的逆矩阵，且验证。

```
A = [1,-1,1;5,-4,3;2,1,1];
```

```
B = inv(A);
```

```
A*B
```

```
ans =
```

```
1.0000    0    0
-0.0000    1.0000    0
-0.0000    0    1.0000
```

5.6.2 用矩阵求逆方法求解线性方程组

将包含n个未知数，由n个方程构成的线性方程组表示为：

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

其矩阵表示形式为：

$$Ax = b$$

其中：

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

因此： $x = A^{-1}b$



例5.19 用求逆矩阵A的方法解线性方程组

$$\begin{cases} x + 2y + 3z = 5 \\ x + 4y + 9z = -2 \\ x + 8y + 27z = 6 \end{cases}$$

命令如下：

```
A = [1,2,3;1,4,9;1,8,27];
```

```
b = [5,-2,6]';
```

```
x = inv(A)*b  %x = A\b
```

```
x =
```

```
23.0000
```

```
-14.5000
```

```
3.6667
```

也可以运用左除运算符求解。



5.7 矩阵行列式值

把一个方程看做一个行列式，并按行列式的规则求值，称为行列式的值。在Matlab中，使用函数`det(A)`得到。

例如：

`A = rand(5)`

A =

0.9501	0.7621	0.6154	0.4057	0.0579
0.2311	0.4565	0.7919	0.9355	0.3529
0.6068	0.0185	0.9218	0.9169	0.8132
0.4860	0.8214	0.7382	0.4103	0.0099
0.8913	0.4447	0.1763	0.8936	0.1389

`B = det(A)`

B =

-0.0071