

第5讲 函数文件

(第4章 MATLAB 程序流程控制)

目的:

掌握 matlab 常用函数的分类与定义以及调用方法

一、匿名函数的建立与调用(退出 matlab 就会被释放)

二、自定义的永久保存的函数(以函数文件的形式存放在硬盘目录中, 定义中可以包含下方的匿名函数和符号函数)

三、符号函数(退出 matlab 就会被释放)(第九章内容, 以后再讲)

(一)、匿名函数的建立与调用(重要, 常用)

所谓匿名函数就是不需要保存为*.m 文件的一种函数结构, 匿名函数定义相对*.m 文件来说简单易用, 优点是简单易用, 缺点也是简单易用, 导致不能象*.m 文件一样实现复杂的函数功能。可以在脚本文件、命令行窗口、*.m 函数中使用匿名函数。

匿名函数的定义格式是: 函数句柄变量=@(匿名函数输入参数)匿名函数表达式。

其中匿名函数句柄变量是指指定的匿名函数的函数名, 在 matlab 中将它的类型称为函数句柄, 可以理解为操作函数的手柄。

匿名函数输入参数就是自变量。

匿名函数表达式就是用自变量构造的表达式。

在匿名函数定义中, 如果输入参数是矩阵, 记得在写表达式时使用点运算.* ./ .^等。

例 1, 可以命令行窗口建立如下匿名函数

```
>> fx=@(x)x^2-3*x+1;
```

这里定义了一个匿名函数, 句柄变量名字是 fx。

定义好匿名函数后就可以直接使用它, 例如在命令行窗口输入:

```
>>fx(2)
```

```
ans =  
-1
```

此语句使用句柄变量 fx 将参数 2 传递给表达式, 系统会计算表达式值并给出结果

例 2 建立多元匿名函数

```
>> fxy=@(x,y)x^2+sin(y);
```

```
>> fxy(2,pi/2) %使用句柄变量 fxy 将参数 2, pi/2 传递给表达式
```

```
ans =  
5
```

例 3, 可以使用一个函数句柄变量传递自变量参数给多个匿名函数表达式

```
>> fxyxy=@(x,y)[x^2+sin(y),x+y]
```

%此处定义了两个二元函数, 分别是 $x^2+\sin(y)$ 和 $x+y$

```
>> fxyxy(2, pi/2)
```

```
ans =  
5.0000    3.5708
```

上面语句使用句柄 fxyxy 同时给两个函数传递了参数 2, pi/2, 系统分别计算了它们的函数值, 计算结果是由函数值所构成的行向量。

如果将定义 $\text{fxyxy}=@(x,y)[x^2+\sin(y),x+y]$ 中的逗号改为分号，结果将显示为列

```
>> fxyxy=@(x,y)[x^2+sin(y);x+y]
>> fxyxy(2, pi/2)
ans =
    5.0000
    3.5708
```

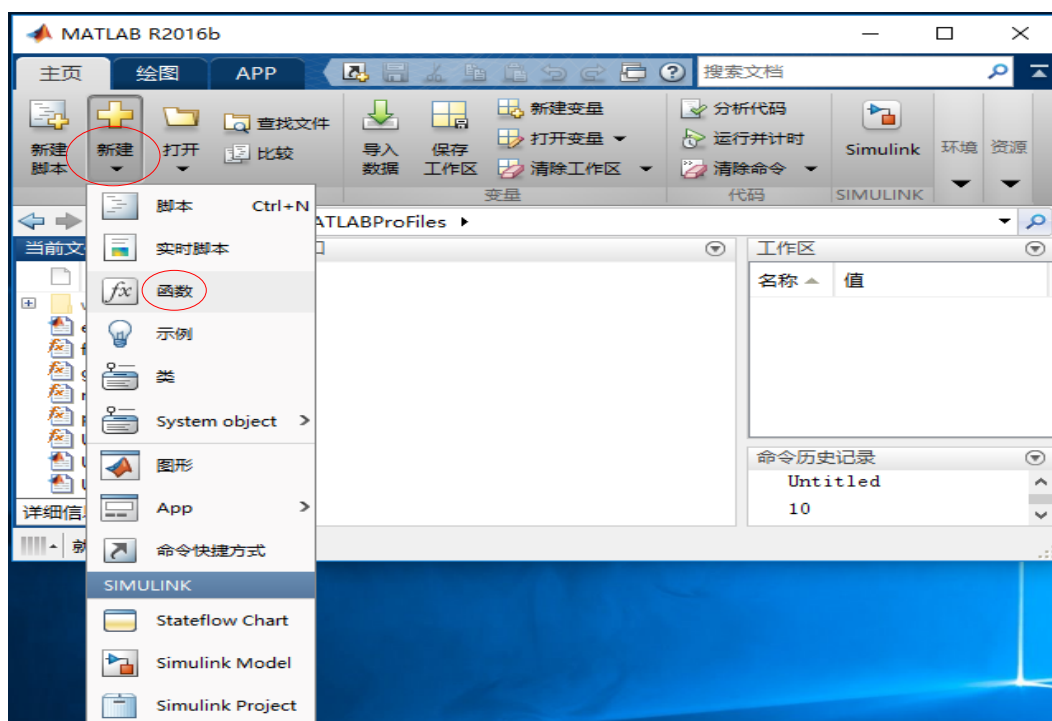
在匿名函数定义中，如果输入参数是矩阵，记得在写表达式时使用点运算 $.* ./ .^$ 等。

二、自定义的永久保存的函数

步骤：

1、函数的建立与保存

(1) 建立函数文件。在主窗口左上角单击“新建”按钮，在下拉菜单中选中“函数”。



(2) 弹出编辑器窗口，其中有函数的引导行等内容。然后，修改输入自己的函数内容。

M 函数的格式是：

`function [输出变量列表]=函数名(输入变量列表)`

中间这些是执行语句...

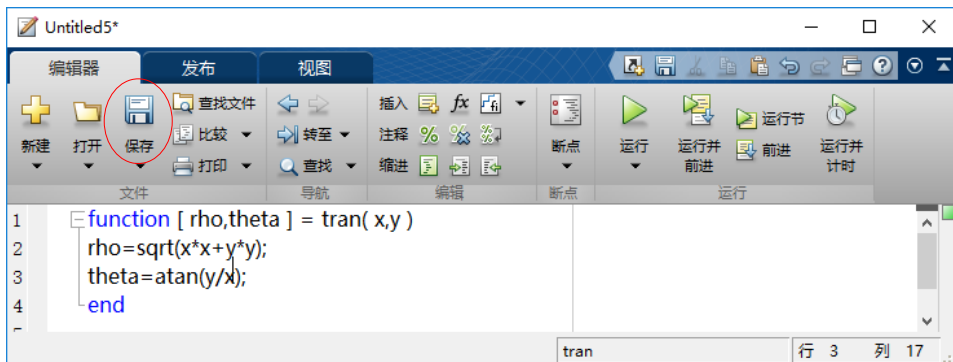
`end` %结尾 `end` 可以不写，但为了结构完整，建议大家写上

例如，建立一个名字叫 `tran` 的函数，实现直角坐标 (x,y) 与极坐标 (θ,ρ) 的转换。转换

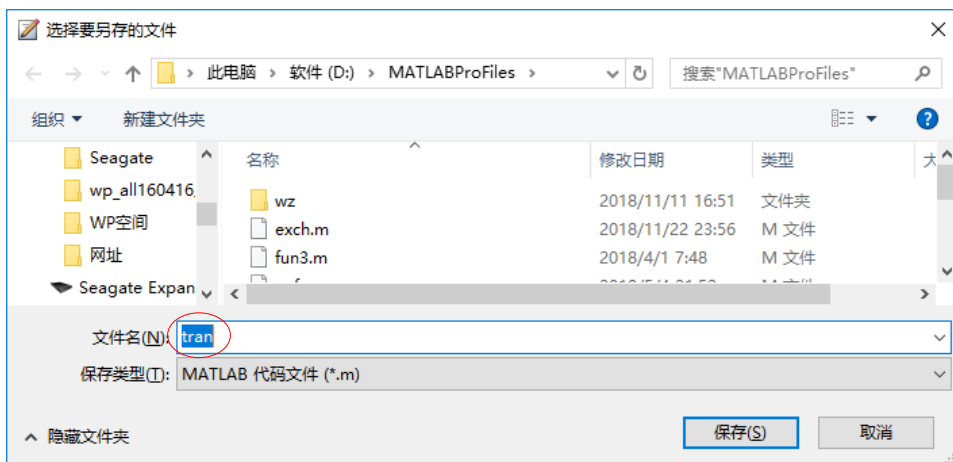
公式为 $\rho = \sqrt{x^2 + y^2}, \theta = \arctan\left(\frac{y}{x}\right)$ 。显然，自变量为 x, y 输出变量为 ρ, θ 。

```
function [rho,theta]=tran(x,y)
    rho=sqrt(x^2+y^2);
    theta=atan(y/x);
end
```

(3) 输入结束，单击“保存”按钮。注意：后续每次修改后，也要点击“保存”，否则本次修改无效。程序有修改时，“保存”图标鲜亮，保存后，“保存”图标暗淡。

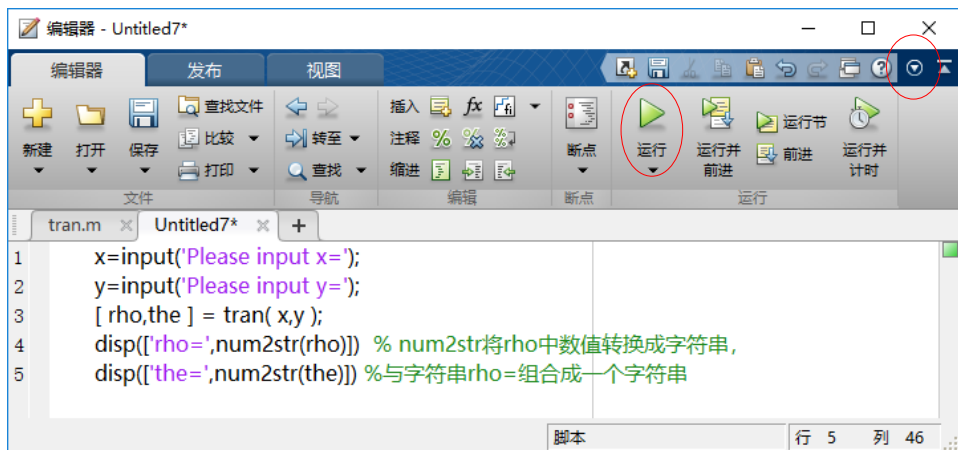


(4) 函数文件名一定要取成建立的函数的名字，此处是 tran。单击“保存”。

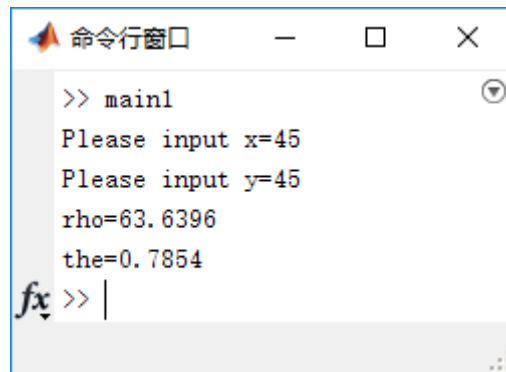


2、函数的调用 (调用前需确保函数名和 M 函数文件名相同)

- (1) 调用方法一：可以在命令行窗口输入比如 `>>[a,b]=tran(1,1)`
- (2) 调用方法二：另外建立脚本文件，在脚本文件中调用。



(6) 确定脚本文件名，输入 main1。转到命令行窗口，从键盘输入数据响应后，输出结果。



练习 1 求给定复数的指数、对数、正弦和余弦

定义一个函数文件，求给定复数的指数、对数、正弦和余弦，并在脚本文件中调用该函数文件。

给出函数文件：

```
function [ex, lo, si, co]=lab5_1(x)
ex=exp(x);lo=log(x);si=sin(x);co=cos(x);
end
```

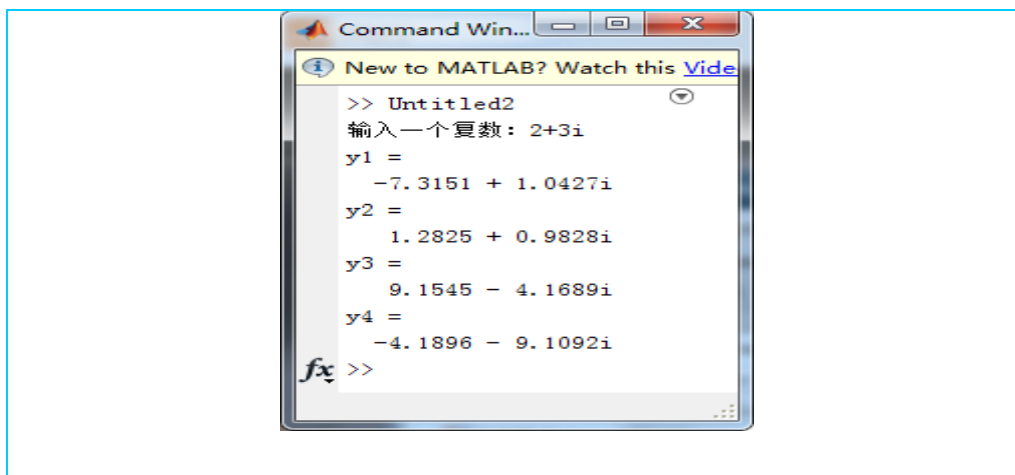
(1) 在命令行窗口调用：

```
>>[a, b, c, d]=lab5_1(2+3i)
```

(2) 建立脚本文件在脚本中调用：

```
x=input(' 输入一个复数： ');
[y1, y2, y3, y4]=lab5_1(x)
```

给出运行结果：



【提示】

不要在函数文件中显示结果，由调用的程序决定是否显示结果。

注意！ 有输入参数的函数文件如果直接点击菜单栏上的运行按钮运行将出错，这是因为输入参数（函数的自变量）还没有赋值，程序无法运行，将给出错误信息。初学者容易犯这个错误，误以为和脚本文件一样，建立完文件马上按“运行”按钮，显示出来的是错误信息，便束手无策。

函数文件一般是被调用才运行的，调用时，必须给函数的输入参数赋值。只有无参的函数（或输入参数可取默认值）例外，可像脚本文件一样直接运行。

练习 2. 求所有两位绝对素数

11 13 17 31 37 71 73 79 97

一个自然数是素数，且它的数字位置经过任意对换后仍为素数，则称是绝对素数。例如 13 是绝对素数。试求所有两位绝对素数。

要求：定义一个判断素数的函数文件，不能用系统提供的判断素数的函数。再定义一个脚本文件，在其中调用判断素数的函数文件。

给出判断素数的函数文件：

%设计函数1用于判断单个数是否为素数：若x为素数，则b=1；否则b=0。

```
function b=fprime1(x)
if x<2
    b=0; return; %return语句：停止执行后面语句，退出函数
end
b=1;
for k=2:sqrt(x)
    if rem(x,k)==0
        b=0; return;
    end
end
end
```

%设计函数2用于判断一组数里是否有素数

```
function y=fprime2(p)
%p为一组整数（向量），返回的y是p中的素数或空矩阵。
m=fix(sqrt(max(p))); % m是p中最大值的根值并取整
for i=2:m
    n=find(rem(p,i)==0&p~=i);
    %使用find函数找到p中能被i整除且不为i的那些元素的序号n（这些元素不是素数）
    p(n)=[];%从p中删除第n个元素
end
y=p;
end
```

函数调用：

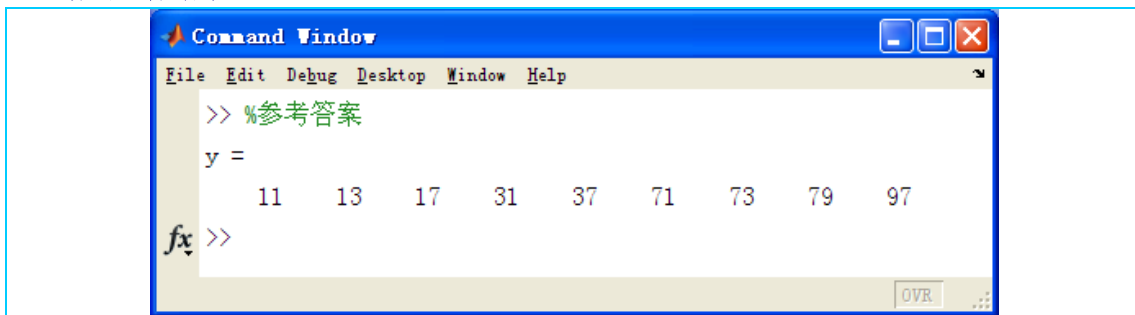
%可以在脚本文件中调用函数1

```
clc;clear;
y=[];%在内存中预建立一个空矩阵用于存放素数
for k=11:99
    s=10*rem(k,10)+fix(k/10); %交换k的个位和十位位置得到数s
    if fprime1(k)&&fprime1(s) %判断k和s是否都是素数
        y=[y,k];
    end
end
y
```

%也可以在命令行窗口调用函数2

```
>>y1=fprime2(10:99); %y1是10：99中的素数
>>s=10*rem(y1,10)+fix(y1/10); %交换y1中数字的个位和十位
>>y2=fprime2(s) %y2是s中的素数
```

给出运行结果：



【提示】

一个素数是除了 1 和它本身外没有其它因子的正整数。

编程时，可逐个取 $2 \sim \sqrt{m}$ 除 m ，若某个能整除（用 rem 函数判断），则 m 不是素数，函数返回逻辑假 false 或 0，否则， m 是素数，返回逻辑真 true 或 1（或非 0 值）。

练习 3：变量和函数值都为矩阵的函数

设 $f(x) = \frac{1}{(x-2)^2 + 0.1} + \frac{1}{(x-3)^4 + 0.01}$ ，编写一个 MATLAB 函数文件 fx.m

要求：编写的函数要能使调用 f_x 时， x 可用矩阵代入，得出的 $f_x(x)$ 为同阶矩阵。
即，如果自变量录入的是一个矩阵，那么要对矩阵中的每个元素求函数值。

给出函数文件：

```
function y=fx(x)
y=1./((x-2).^2+0.1)+1./((x-3).^4+0.01);
end
```

命令行输入：

```
fx([1 2 3;4 5 6;7 8 9])
```

给出运行结果：



【提示】

使用 input 语句，从键盘输入矩阵时，要加方括号。注意要用点运算。

练习 4：函数调用

设 $f(n) = n + 10\ln(n^2 + 5)$ ，已知 $y = \frac{f(40)}{f(30) + f(20)}$ ，求 y 的值？

给出函数及运行结果：

```
function s=f(n)
s=n+10*log(n^2+5);
end
>> y=f(40)/(f(30)+f(20))
y =
    0.6390
```

了解两个内置的特殊变量：nargin 和 nargout，其中 nargin 表示输入变量个数，nargout 表示输出变量个数。通过下面例子来说明他们的作用。

练习 5：编写 m 函数 cxy，形参为 x, y, r ，函数的作用是计算 $x+ry$ 和 $x-ry$ 的值，其中 r 为参

数，默认取 1.

```
function [z1,z2]=cxy(x,y,r)
    if nargin==2
        r=1;
    elseif nargin<2 %如果输入的参数不足，用 return 结束函数执行
        disp('输入的实参数目不足');
        return;
    end %if 语句结尾
    if nargin<=1 %如果只有一个输出，则把两个结果合成向量显示
        z1=[x+r*y,x-r*y];
    else
        z1=x+r*y;
        z2=x-r*y;
    end % if 语句结尾
end %函数结尾
```

可以在命令行窗口对函数进行测试

```
>> [a,b]=cxy(2,5,2) %3 个输入，2 个输出
>> c=cxy(2,5,2) %3 个输入，1 个输出
>> [a,b]=cxy(2,5) %2 个输入，2 个输出
>> c=cxy(2,5) %2 个输入，1 个输出
>> c=cxy(2) %1 个输入，1 个输出
```

练习 6（函数的递归调用）：编写 m 函数计算迭代型数列值 $x_{n+1} = \frac{1}{x_n + 1}$

```
function y=fdg(x,n)
```

%用 if 语句检查是不是要算 x1,如果是，直接给 x1 的值，如果不是，进入迭代。

```
if n==1
    y=x;
else
    y=1/(fdg(x,n-1)+1);
end
end
```

命令行测试：

```
>>fdg(2,1)
ans=
    2
>> fdg(2,2)
ans=
    0.3333
```



```
>> fdg(2,3)
ans=
    0.7500
```

递归调用的过程（以 $\text{fdg}(2,3)$ 为例）如下：

运行 $\text{fdg}(2,3)$ 时，首先检查 $n==1$ ？否，于是 $y=1/(\text{fdg}(2,2)+1)$ ，由于 $\text{fdg}(2,2)$ 没有现值，系统会将 y 挂起，再重新开辟一个内存单元存储运行 $\text{fdg}(2,2)$ ，对 $\text{fdg}(2,2)$ ，检查 $n==1$ ？否，于是在新存储单元里赋值 $y=1/(\text{fdg}(2,1)+1)$ ，同理由于 $\text{fdg}(2,1)$ 没有现值，系统再重新开辟一个内存单元存储运行 $\text{fdg}(2,1)$ ，对 $\text{fdg}(2,1)$ ，检查 $n==1$ ？是，于是 $y=2$ （即 $\text{fdg}(2,1)=2$ ），将之回代入上一个存储单元，得到上一个存储单元的 y 值（ $y=0.3333$ ，即 $\text{fdg}(2,2)=0.3333$ ），再一次回代入最初的存储单元，得到最后的 y 值（ $y=0.7500$ ，即 $\text{fdg}(2,3)=0.7500$ ）。