



数学软件 Matlab

—— 自定义函数的几种方式

自定义函数的方式

□ Matlab 中自定义函数的常见方式

- 脚本文件 / 函数文件 + 函数文件
- 函数文件 + 子函数
- 内联函数和匿名函数
- 符号表达式 + `subs`

函数文件

- 方式一：命令文件 / 函数文件 + 函数文件

```
% 脚本文件: myfile1.m
```

```
clear
```

```
for t=1:10;
```

```
    y=mylfg(t);
```

```
        fprintf('%4d^(1/3) = %6.4f\n',t,y);
```

```
end
```

调用函数时要注意实参与形参的匹配!

```
% 函数文件: mylfg.m
```

```
function y = mylfg(x)
```

```
y=x^(1/3);
```

函数名与文件名必须一致!

函数必须单独写一个文件!
不能与命令文件写在同一个文件中!

子函数

- 方式二：函数文件 + 子函数

```
% 函数文件: funtry2.m
function []=funtry2()
for t=1:10
    y=1fg2(t);
    fprintf('%4d^(1/3) = %6.4f\n',t,y);
end

function y=1fg2(x) % 子函数
y=x^(1/3);
```

函数文件中可以定义一个或多个子函数，此时我们称该函数为主函数，子函数只能被主函数或同一个函数文件中的其它子函数调用！

函数可以使用 `end` 表示结束，但这不是必须的。

内联函数

- 方式三：内联函数和匿名函数

- 内联函数

```
f=inline('函数表达式', '变量1', '变量2', ...)
```

- 调用方式

```
y=f(实参列表)
```

实参列表应与定义时的变量顺序保持一致

根据实际情况，定义函数时可能需要使用数组运算

例：

```
f=inline('x^2+y', 'x', 'y')  
y=f(2,3)
```

匿名函数

- 方式三：内联函数和匿名函数

- 匿名函数 + 函数句柄

$f = @(变量列表) 表达式$

- 调用方式

$y=f(实参列表)$

例：

```
f = @(x,y) x^2 + y;  
y = f(2,3)
```

实参列表应与定义时的变量列表保持一致

根据实际情况，可能需要使用数组运算

匿名函数

- 这种函数定义方式是将 f 作为一个内部函数调用。其特点是：调用方式最接近于我们平时对函数的定义，使程序更具可读性。同时由于它是基于 Matlab 的数值计算内核的，所以它的运算速度较快，程序更有效率。

- 这种定义方式的缺点：
 - 只能定义一些简单的函数；
 - 不支持符号运算。

符号表达式

- 方式四：符号表达式 + subs

- 首先定义一个符号表达式/函数，用 subs 命令调用

```
syms x  
f=1/(1+x^2);
```

```
subs(f,x,代替 x 的数值或符号)
```

- 优点：可以进行符号计算
- 缺点：由于使用符号运算内核，运算速度会大大降低。

上机作业

1、已知 Chebyshev 多项式定义如下：

$$T_0(x) = 1, T_1(x) = x,$$

$$T_{n+1}(x) = 2x T_n(x) - T_{n-1}(x), n = 1, 2, \dots$$

利用符号计算，编程计算 $T_{10}(x)$ （注： $T_n(x)$ 为 n 阶多项式）

程序取名 m81.m

2、利用多项式运算，采用递归方法，实现上题

程序取名 m82.m