数学软件 Matlab

—— Matlab 编程（函数）

本讲主要内容

- 什么是函数文件
- 如何编写函数文件
- 递归函数
- 函数参数的可调性
- 局部变量与全局变量
- 子函数
- 函数句柄、内联函数、匿名函数

Matlab 编程

- M 文件根据调用方式的不同可以分为两类：
  - **Script**：脚本文件/命令文件
    - 直接输入文件名即可运行
  - **Function**：函数文件
    - 供其它M文件调用，通常带输入参数和输出参数

函数文件

❑ 函数文件一般格式

```
function  输出形参列表=函数名(形参列表)
% 注释说明部分(可选)
函数体语句(必须)
```

- 第一行为引导行，表示该 M文件是函数文件
- 函数名的命名规则与变量名相同 (必须以字母开头)
- 当输出形参多于一个时，用方括号括起来

- 函数文件名必须与函数名一致
- 函数必须是一个单独的 M文件

编程示例

例： 将华氏温度转化为摄氏温度：

$$c = \frac{5}{9}(f - 32)$$

- 脚本文件 (f2cs.m)：

```
clear;
f=input('Please input f:');
c=5*(f-32)/9;
```

- 函数文件(f2cf.m)：

```
function c = f2cf(f)
c=5*(f-32)/9;
```

函数文件举例

例：交换两个变量的值 myswap.m

```
function [a,b] = myswap(x,y)
%  swap x and y
a = y;
b = x;
```

- 有两个输入参数和两个输出参数

函数文件举例

例：打印杨辉三角形 printyh.m

```
function printyh(n)
% 打印杨辉三角形，本函数没有输出参数
yh = 1;  disp(yh);
if n==1, return; end
yh = [1,1];  disp(yh);
for k = 3 : n
  yh_old = yh;  k2 = ceil(k/2);
  for i = 2 : k2
     yh(i) = yh_old(i-1) + yh_old(i);
  end
  yh(k2+1:k) = yh(k-k2:-1:1); disp(yh);
end
```

- 可以没有输入参数或输出参数

函数调用

❑ 函数调用的一般格式

输出实参列表=函数名(输入实参列表)

- 函数调用时，实参的顺序应与函数定义时形参的顺序一致
- 实参与形参之间的结合是通过值传递实现的
- 函数可以嵌套调用，即一个函数可以被其它函数调用，甚至可以被它自身调用，此时称为递归调用
- 函数所传递的参数具有可调性，Matlab 用两个永久变量 `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

递归函数举例

例：利用函数的递归调用计算 $n!$

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

```
% 函数文件 myfactor.m
function y=myfactor(n)
if (n<=1)
   y=1;
else
   y=n*myfactor(n-1);
end
```

递归函数举例

例：计算 1! + 2! + … + 10!

```
% main.m
%
clear;
s = 0;
n = 10;
for i = 1 : n
    s = s + myfactor(i);
end
fprintf(' s=%g \n',s)
```

参数的可调性举例

例：nargin 和 nargout 的使用

```
% ex4nargin.m
function y = ex4nargin(a,b)
if (nargin==1)
   y = a;
elseif (nargin==2)
   y = a*b;
end
```

```
% ex4nargout.m
function [p,q]=ex4nargout(a,b)
if (nargout==1)
   p = a + b;
elseif (nargout==2)
   p = a + b;
   q = a - b;
end
```

局部变量与全局变量

- 函数文件中的变量都是局部的，即一个函数文件中定义的变量不能被另一个函数文件或其它 M 文件使用
  - 当函数调用完毕后，该函数文件中定义的所有局部变量都将被释放，即全部被清除
  - 函数通过输入和输出参数与其它 M 文件进行数据传递
- 如果在若干个 M文件中，都把某个变量定义为全局变量，则这些函数将公共使用这一变量。所有函数都可以对它进行存取和修改操作
  - 定义全局变量是 M文件间传递信息的一种手段

全局变量的定义

- 全局变量的定义

```
global 变量名列表
```

- 变量名列表中的各个变量用空格隔开，不能用逗号！
- 在使用全局变量的所有M文件中，都要对其所使用的全局变量进行定义

全局变量给函数间的数据传递带来了方便，但却破坏了函数对变量的封装，降低了程序的可读性，因而在结构化程序设计中，全局变量是不受欢迎的。特别是当程序较大，子程序较多时，全局变量将个程序调试和维护带来不便，故不提倡使用全局变量。

程序示例

```
% ex4global.m
clear;
global a b
a = 1; b = 3;
y = mysquaresum(a,b);
fprintf(' a=%g, b=%g \n',a,b);
z = myproduct(a,b);
fprintf(' a=%g, b=%g\n',a,b);
```

```
% mysquaresum.m
function square_sum = mysquaresum(x,y)
square_sum = x^2 + y^2;  a = x+y;
```

```
% myproduct.m
function product = myproduct(x,y)
global a
product = x*y; a = x+y;
```

子函数

- 一个函数文件中可以含有一个或多个函数，其中第一个称为主函数，其它函数称为子函数
- 子函数由 `function` 语句引导
- 除了用 `global` 定义的全局变量外，所有函数中的变量都是局部变量，函数之间通过输入、输出参数进行数据传递

- 主函数必须位于最前面，子函数出现的次序任意
- 子函数只能被主函数和位于同一个函数文件中的其它子函数调用

调用一个函数时，Matlab 会首先检查该函数是否为一个子函数

子函数举例

```
% ex4subfun.m
function [avg, med] = ex4subfun(x) % 主函数
n = length(x);
avg = mean(x, n);
med = median(x, n);

function a = mean(x, n) % 子函数，计算平均值
a = sum(x)/n;

function m = median(x, n) % 子函数，计算中值
x = sort(x);
if rem(n, 2) == 1
   m = x((n+1)/2);
else
   m = (x(n/2)+x(n/2+1))/2;
end
```

函数句柄

函数句柄，可以理解成一个函数的代号或别名，调用函数句柄就等价于调用该函数。

❑ 函数句柄的定义

```
fhandle = @函数名
```

- @ 的作用就是将一个函数的函数句柄赋值给左边的变量

例：

```
f = @sin;
y = f(pi/3)
```

内联函数

MATLAB中的内联函数借鉴了C语言中的内联函数，使用内联函数可以减少调用的时间和空间开销。

- 内联函数的定义

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

- 由于内联函数是储存于内存中而不是在M文件中，省去了文件访问的时间，加快了程序的运行效率。
- 但内联函数只能定义一些简单的函数表达式。
- 若调用函数时涉及数组运算，则定义函数时也要用数组运算！

例：

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

匿名函数

匿名函数（anonymous function）是 MATLAB 7.0 版提出的一种全新的函数描述形式，和内联函数类似，可以让用户编写简单的函数而不需要创建M文件，因此，匿名函数具有inline函数的所有优点，并且效率比inline函数高。

- 匿名函数的定义

```
fhandle = @(输入参数列表)运算表达式
```

例：

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

```
p = 3; q = 5;
f = @(x,y) x^p + y^q;
```

匿名函数

- 匿名函数支持参数替换

```
p=3; q=5;
f=@(x,y) x^p + y^q;
y=f(2,3)
p=2; q=4;
f=@(x,y) x^p + y^q; % 当参数发生改变时，函数必须重新定义
y=f(2,3)
```

- 若调用函数时涉及数组运算，则定义函数时也需要使用数组运算

```
f=@(x) x.^2 + 1;
x=1:5;
y=f(x)
```