数学软件 Matlab

—— Matlab 编程（函数）

内容提要

- 函数文件的定义
- 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，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(输入形参列表)
 % 注释说明部分 (可选)
函数体语句
```

- 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

- 函数必须是一个单独的 M文件
- 函数文件的文件名必须与函数名一致

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;
fprintf('c=%g \n', c)
```

- 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;
```

- 函数可以有多个输入参数和多个输出参数
- 也可以没有输入参数或输出参数

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
```

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=%d \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
```

```
>> y1=ex4nargin(3)
>> y2=ex4nargin(3,4)
```

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

```
>> x=ex4nargout(5,3)
>> [x,y]=ex4nargin(5,3)
```

内容提要

- 函数文件的定义
- 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

局部变量与全局变量

- **局部变量与全局变量**
  - 局部变量：只能在其被定义的 M 文件中使用
  - 全局变量：可以在多个 M 文件中使用
  - Matlab 中，变量默认是局部变量
- **局部变量**
  - Matlab 中，变量默认是局部变量，即一个M文件中定义的变量不能被其它 M 文件引用
  - 当函数调用完毕后，该函数文件中所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

程序示例

```
% main.m
clear;
a=1; b=3;
z=mysquaresum(a,b);
fprintf(' z=%d \n', z);
disp(mysum)  % ERROR
```

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

局部变量与全局变量

- 全局变量
  - 全局变量的定义或声明

```
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！
  - 在需要使用全局变量的所有M文件中，都要进行声明
  - 定义全局变量是 M文件间传递信息的一种手段

程序示例

```
% main.m
clear;
global a b;  % 声明 a b 是全局变量
a=1; b=3;
z=mysquaresum();
fprintf(' z=%d \n', z);
disp(mysum);  % ERROR，mysum 不是全局变量
```

```
% mysquaresum.m
function mysum = mysquaresum()
global a b;    % 这里也必须声明 a b 是全局变量
global mysum;
mysum=a^2+b^2; % OK
```

全局变量

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，全局变量将给程序调试和维护带来不便，故**一般不提倡使用全局变量**。

程序示例

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

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

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

内容提要

- 函数文件的定义
- 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

子函数

- 子函数
  - 一个函数文件中可以包含一个或多个函数，其中第一个称为主函数，其它函数称为子函数
  - 除全局变量外，所有函数（主函数、子函数）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M文件中的其它子函数调用
- 外部 M 文件只能调用主函数

子函数举例

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

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

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

```
>> x=rand(6,1)
>> [y1,y2]=ex4subfun(x)
```

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；它具有内联函数的所有优点，并且效率比内联函数高。

- 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)
```

匿名函数

- 匿名函数支持变量替换

```
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)
```

上机作业

1、兔子繁殖问题：(Fibonacci number)

假设每对大兔每月生出一对小兔，且新生的小兔满二个月后长成大兔就能生育，那么从刚出生的一对小兔算起，12个月总共有多少对兔子？三年年底呢？试编写一个函数计算该题，其中输入为月数，输出为兔子对数。

（函数名取为 m71）（只需交函数文件）

2、编写一个函数，函数名为 m72，要求：

- **一个输出参数，三个输入参数**
- **当输入一个参数时，输出一个出错信息，并返回**
- **当输入两个或三个参数时，计算它们的阶乘的和**

上机作业

3、编写一个函数文件： m73.m，实现两个向量的加运算或减运算，要求：

(1) 一个输出参数，三个输入参数;

(2) 当输入参数是两个时，计算它们的和；

(3) 当输入参数是三个时，计算前两个参数的差。

（ 在长度较短的向量前面添 0，使得两个向量长度相等）