数学软件 Matlab

—— 编程基础（脚本）

本讲主要内容

- M 文件
- 基本运算：算术、关系、逻辑
- 简单的输入输出
- 控制结构：顺序、选择、循环

Matlab 编程入门

❑ Matlab 编程简介

- Matlab 作为一种广泛应用于科学计算的工具软件，不仅具有强大的数值计算能力和丰富的绘图功能，同时也可以与 C、FORTRAN 等高级语言一样进行程序设计

- 利用 Matlab 的程序控制功能，将相关 Matlab 命令编成程序存储在一个文件中（M 文件），然后在命令窗口中运行该文件，Matlab 就会自动依次执行文件中的命令，直到全部命令执行完毕

- 在 Matlab 程序设计中，要充分利用 Matlab 数据结构的特点，提高编程效率

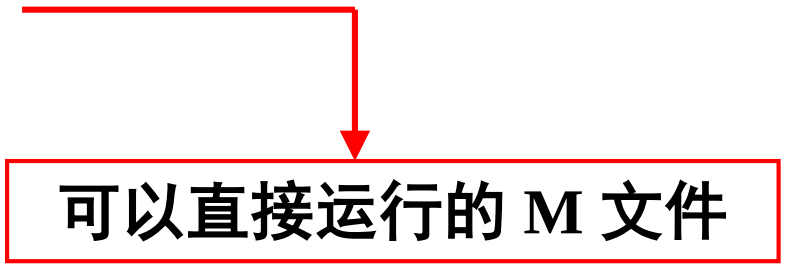
M 文件

M 文件介绍

- 用 Matlab 语言编写的程序称为 M 文件
- M 文件以 .m 为扩展名
- M 文件是由若干 Matlab 命令组合在一起构成的，它可以完成某些操作，也可以实现某种算法

两类重要的 M 文件（调用方式不同）

- Script：脚本文件/命令文件
- Function：函数文件



M 文件

❑ M 文件的创建的与编辑

M 文件是文本文件，可以用任何文本编辑器来建立和编辑，通常使用 Matlab 自带的 M 文件编辑器

- 新建一个 M 文件
  - 菜单操作 (File → New → M-File)
  - 命令操作 (edit M 文件名)
  - 命令按钮 (快捷键)
- 打开已有的 M 文件
  - 菜单操作 (File → Open)
  - 命令操作 (edit M 文件名)
  - 命令按钮 (快捷键)
  - 双击 M 文件

编程示例

例：用 mesh 绘制半径为 3 的球

- 命令行方式：

```
>> u=[0:pi/60:2*pi];
>> v=[0:pi/60:pi];
>> [U,V]=meshgrid(u,v);
>> R=3;
>> X=R*sin(V).*cos(U);
>> Y=R*sin(V).*sin(U);
>> Z=R*cos(V);
>> mesh(X,Y,Z);
>> axis equal;
```

文件的命名规则与变量相同！

- 编程方式：新建一个 M 文件 myprg1.m，内容如下：

```
u=[0:pi/60:2*pi];
v=[0:pi/60:pi];
[U,V]=meshgrid(u,v);
R=3;
X=R*sin(V).*cos(U);
Y=R*sin(V).*sin(U);
Z=R*cos(V);
mesh(X,Y,Z);
axis equal;
```

在命令窗口输入 myprg1，即可执行该 M 文件

本讲主要内容

- M 文件
- 基本运算：算术、关系、逻辑
- 简单的输入输出
- 控制结构：顺序、选择、循环

关系运算

❑ 关系运算符

<	小于	<=	小于等于
>	大于	>=	大于等于
==	等于	~=	不等于

- 比较大小，如果结论是 真 则返回 1 ，否则返回 0
- 注意 == 与 = 的区别
- 关系操作符可以比较两个同样大小的数组，或用来比较一个数组和一个标量，在后一种情况，标量和数组中的每一个元素相比较，比较结果与数组大小一样

关系运算举例

例：

```
>> 2+2==4
```

```
>> 2>3
```

```
>> A=[1 3 5; 2 0 6];
>> B=[3 1 0; 2 4 6];
>> A>=B
```

```
>> x=[5 0; 4 2];
>> x<4
```

逻辑运算

❑ 逻辑运算符

&	与 (Elementwise AND)
|	或 (Elementwise OR)
~	非
xor(x,y)	异或

```
A & B   等价于 and(A,B)
A | B   等价于 or(A,B)
 ~ A    等价于 not(A)
```

&&	与 (Short-circuit AND)
||	或 (Short-circuit OR)

逻辑运算表

运算对象		与	或	非	异或
A	B	A&B	A|B	~A	Xor(A,B)
0	0	0	0	1	0
0	1	0	1	1	1
1	0	0	1	0	1
1	1	1	1	0	0

在 Matlab 中，0 表示 “假”，非零表示 “真”

逻辑运算

- 逻辑运算函数：all、any

any(x)

如果向量 x 中存在非零元素，则返回 1，
否则返回 0

all(x)

如果向量 x 中所有元素都非零，则返回 1，
否则返回 0

若 x 为矩阵，则 any 和 all 按列运算，
返回一个 0-1 向量

一些测试函数

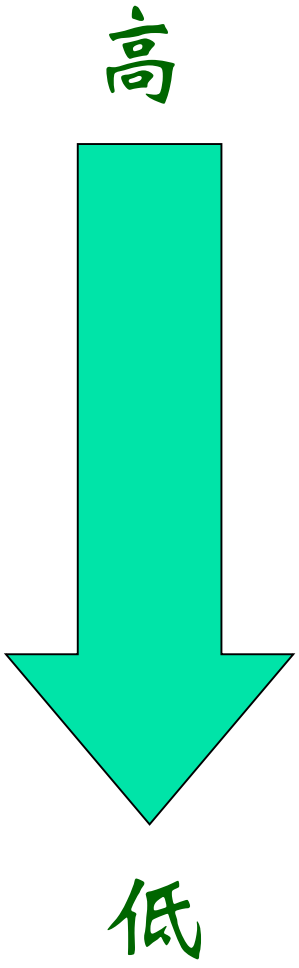
- 测试函数

isfinite(x)	若 x 为有限值，返回真值 1
isinf(x)	若 x 为无穷大，返回真值 1
isnan(x)	若 x 为不定值，返回真值 1
isreal(x)	若 x 无虚部，返回真值 1
isstr(x)	若 x 为一个字符串，返回真值 1
isempty(x)	若 x 为空，返回真值 1

```
>> isfinite(5)
>> isinf(5)
```

运算优先级

括号
幂，点幂
正号，负号，逻辑非
乘，除，点乘，点除
加，减
冒号运算
关系运算
&
|
&&
||



本讲主要内容

- M 文件
- 基本运算：算术、关系、逻辑
- 简单的输入输出
- 控制结构：顺序、选择、循环

input

- 数据的输入：input

```
A=input(提示信息)
```

  - 其中 提示信息 为字符串，
  - 该命令要求用户输入 A 的值 (可以是数或字符串)

例：

```
A=input('Please input A: ')
```

例：

```
name=input('What''s your name? ')
```

  - 输入字符串时必须带单引号
  - 单引号的输出：两个连续的单引号

disp

- 数据的输出：disp

```
disp(X)
```

  - 输出变量 X 的值，X 可以是数值矩阵或字符串
  - 一次只能输出一个变量

例：

```
>> A='Hello, Tom!';
>> disp(A)
```

```
>> B=[1 2 3; 4 5 6; 7 8 9];
>> disp(B)
```

fprintf

- 数据的格式化输出：fprintf

```
fprintf(format,variables)
```

- 按指定的格式将变量的值输出到屏幕或指定的文件
- format 用来指定数据输出时采用的格式，包含：
  普通字符串、格式字符串、转义字符

格式字符串：以 % 开头，包括：

- flags（可选）
- Width and precision fields（可选）
- Conversion character（格式说明符，必须）

格式说明符和转义字符

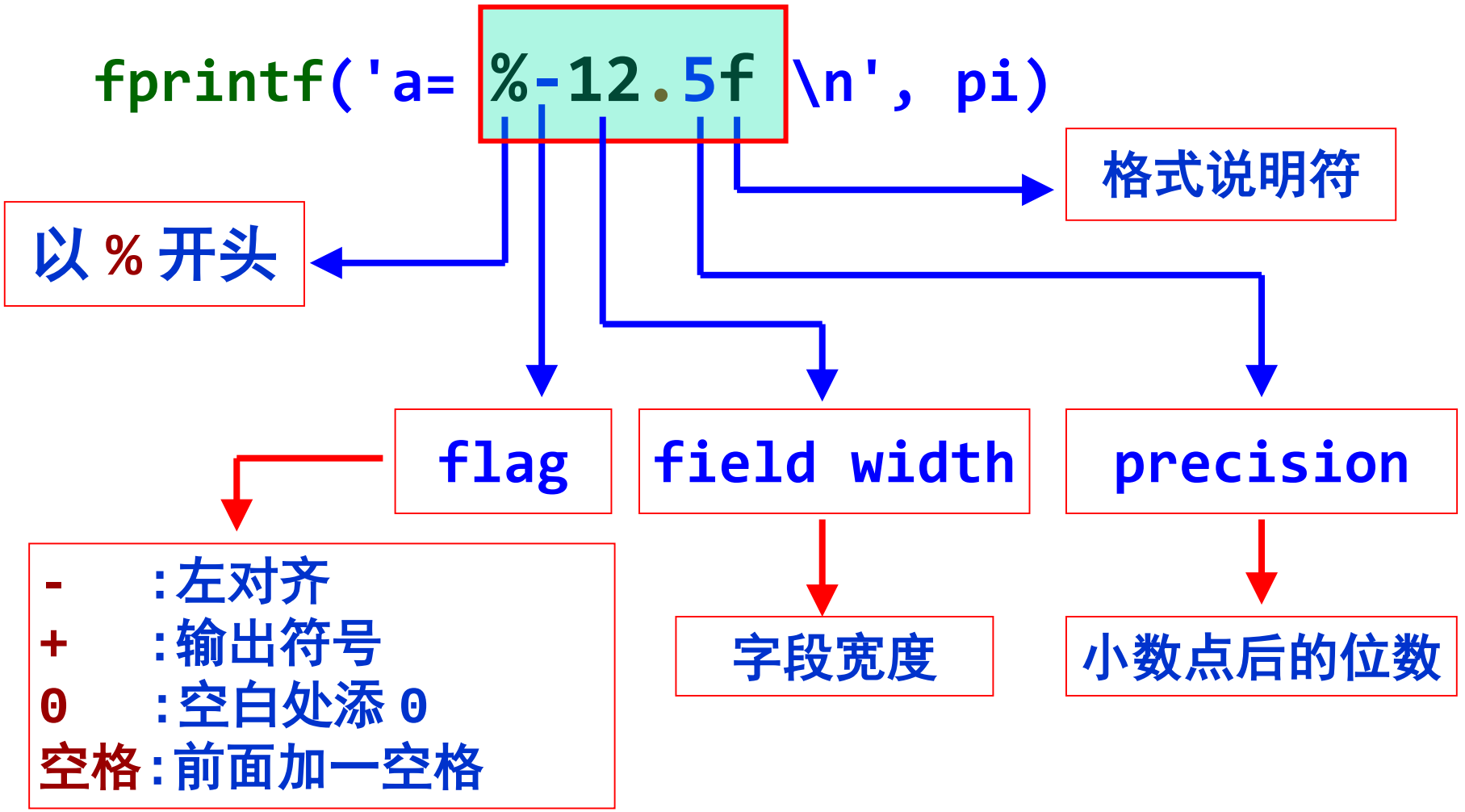
- 常见的格式说明符

c	字符型	g	浮点数（自动）
d	十进制整数	o	八进制
e	浮点数（科学计数法）	s	字符串
f	浮点数（小数形式）	x/X	十六进制

- 常见的转义字符

\b	退后一格	\t	水平制表符
\f	换页	\\	反斜杠
\n	换行	''	单引号
\r	回车	%%	百分号

fprintf


fprintf('a= %-12.5f \n', pi)
以 % 开头
格式说明符
flag
field width
precision
- :左对齐
+ :输出符号
0 :空白处添 0
空格:前面加一空格
字段宽度
小数点后的位数

fprintf

例：

```
>> a='Hello';
>> b=2.4;
>> c=100*pi;
>> fprintf('a=%s, b=%f, c=%e\n',a,b,c)
```

- format 中的格式字符串要与输出变量一一对应

例：

```
>> fprintf(' Today is Monday\n')
```

- 可以没有输出变量

本讲主要内容

- M 文件
- 基本运算：算术、关系、逻辑
- 简单的输入输出
- 控制结构：顺序、选择、循环

M文件控制流

- 程序控制结构有三种：

  顺序结构、选择结构和循环结构

  任何复杂的程序都由这三种基本结构组成

- 顺序结构
  - 按排列顺序依次执行各条语句，直到程序的最后
  - 这是最简单的一种程序结构，一般涉及数据的输入输出、数据的计算或处理等

选择结构

- **选择结构**

  根据给定的条件成立或不成立，分别执行不同的语句

- **选择结构的实现**
  - **if 语句**
  - **switch 语句**

if 条件语句

- 单分支结构

```
if 条件表达式
    语句组
end
```

- 双分支结构

```
if 条件表达式
    语句组1
else
    语句组2
end
```

if 条件语句

- 多分支结构

```
if 条件表达式1
    语句组1
elseif 条件表达式2
    语句组2
  ...  ...
elseif 条件表达式m
    语句组m
else
    语句组
end
```

switch 语句

- 根据表达式的不同取值，分别执行不同的语句

```
switch 表达式
    case 表达式1
         语句组1
    case 表达式2
         语句组2
       ... ...
    case 表达式m
         语句组m
    otherwise
         语句组
end
```

- 先计算表达式的值，然后将它依次与各个 case 指令后表达式的值进行比较，当比较结果为真时，就执行相应语句组，然后跳出 switch 结构。
- switch 后面的表达式的值可以是一个标量或字符串。
- otherwise 指令可以不出现。
- 如果所有的比较结果都为假，则执行 otherwise 后面的语句组。

循环结构

- **循环结构**

  根据给定的条件，重复执行指定的语句

- **循环结构的实现**
  - **for 语句**
  - **while 语句**

for 循环

```
for  循环变量 = 取值列表
       循环体
end
```

- 取值列表 通常是一个向量
- 将取值列表中的值依次赋给循环变量，直到全部取完，循环结束

- 不要在循环体内改变循环变量的值
- 为了提高代码的运行效率，应尽可能提高代码的向量化程度

for 循环

例： 已知 $y=1+\frac{1}{3}+\frac{1}{5}+\cdots+\frac{1}{2n-1}$，当 n=100 时，求 y 的值

```
clear;
y=0;  n=100;
for k=1:n
    y=y+1/(2*k-1);
end
```

例： 计算 s = 1 + 3 + 10 - 28 + 30 + 50 - 12 - 8

```
clear;
s=0;
x=[1,3,10,-28,30,50,-12,-8];
for k=x
    s=s+k;
end
```

while 循环

```
while 条件表达式
      循环体
end
```

- 当条件表达式的值为真（非 0）时，执行循环体语句
- 循环语句可以嵌套使用
- 通常，如果预先知道循环的次数，可采用 for 循环；
  如果预先无法确定循环次数，则可使用 while 循环。

例： 数论中的一个有趣问题：3n+1 问题：

任取一个正整数，如果是偶数，用 2 除，如果是奇数，用 3 乘再加 1，反复这个过程，直到所得到的数为 1。

问：是否存在使该过程永不中止的整数？

编程示例

```
while 1
  n=input('Please enter n (nonpositive quit): ');
  if n<=0, break; end
  nt = n;  % 将 n 的初始值记录下来
  while n>1
    if mod(n,2)==0
       n = n/2;
    else
       n = 3*n+1;
    end
    fprintf(' n=%d \n', n);
  end
  fprintf(' n=%d is not we need! \n', nt);
end
```

while编程示例

例：计算 Matlab 中 eps 值

```
k=0;   EPS=1;
while  (1+EPS) > 1
   EPS = EPS/2;
   k = k+1;
end
```

- 这个例子给出了估计 eps 的一种方法。
- 这里我们用大写 EPS，因此系统中 eps 的值不会被覆盖。当 EPS=eps 时，条件仍成立；而当 EPS<eps/2 时，条件为假，退出循环，所以最后应有 EPS<eps/2。k 用来记录循环次数。

其它流控制语句

- **break 和 continue**
  - break 语句用于终止循环的执行，即跳出最内层循环
  - continue 语句用于结束本次循环，进行下一次循环
  - break 和 continue 一般与 if 语句配合使用

- **return**
  - return 语句用于退出正在运行的脚本或函数，通常用在函数文件中

pause

- 暂停：pause

```
pause 或 pause(n)
```

- 其中 n 是暂停的时间，以秒为单位
- 若缺省，则将暂停程序，直到用户按任意键后继续

- pause off　屏蔽程序中所有 pause 的作用
- pause on　打开 pause 的作用

若想强行终止正常运行的程序，可以使用 Ctrl+c

编程示例

例： **猜数游戏**

首先由计算机随机产生一个 [1,100] 之间的一个整数，然后由用户猜测所产生的这个数。根据用户猜测的情况给出不同的提示，如果猜测的数大于产生的数，则显示 Higher，小于则显示 Lower，等于则显示 You won！，同时退出游戏。用户最多有 7 次机会。

```
x = randi(100,1);  % 由计算机随机产生一个[1,100]的整数
n = 7; % 有7次机会
flag = 1;
fprintf('欢迎参加猜数游戏！你共有 %d 次机会。\n', n);
fprintf('请猜一个 1 到 100 之间的一个整数\n');
for k = 1 : n
    guess=input('Enter your guess: ');
    if guess < x
        disp('Lower');
    elseif  guess>x
        disp('higher');
    else
        disp('Congratulation, You won!');
        flag = 0; break;
    end
    fprintf('你还有  %d 次机会！\n',n-k);
end

if flag==1
    disp('Sorry, You lost!')
end
```