



武汉大学

Wuhan University

第二章 MATLAB语言基础



§2.1 MATLAB基本操作

MATLAB语句形式 变量 = 表达式 ;

通过等于符号将表达式的值赋予变量。当键入回车键时，该语句被执行。语句执行之后，窗口自动显示出语句执行的结果。如果希望结果不被显示，则只要在语句之后加上一个分号（;）即可。此时尽管不显示结果，但它依然被赋值并在MATLAB工作空间中分配了内存。

e.g. $x=3+5;$

$x=3+5$



武汉大学

Wuhan University

工作空间管理命令

1、MATLAB的工作空间包含了一组可以在命令窗口中调整（调用）的参数。

who：显示当前工作空间中所有变量的一个简单列表；

whos：则列出变量的大小、数据格式等详细信息；

clear 变量名：清除指定的变量；

clear：清除工作空间中所有的变量；

clear all：清除工作空间中所有的变量、函数和MEX文件；



2、保存和载入workspace

(1) save filename variables

- 将变量列表variables所列出的变量保存到磁盘文件filename中
- Variables所表示的变量列表中，不能用逗号，各个不同的变量之间只能用空格来分隔。
- 未列出variables时，表示将当前工作空间中所有变量都保持到磁盘文件中。
- 缺省的磁盘文件扩展名为“.mat”，可以使用“-”定义不同的存储格式（ASCII、V4等），缺省保存路径为.....\matlab\works。

e.g. save data.mat



(2) load filename variables

- 将以前用save命令保存的变量variables从磁盘文件中调入MATLAB工作空间。
- 用load 命令调入的变量，其名称为用save命令保存时的名称，取值也一样。
- Variables所表示的变量列表中，不能用逗号，各个同的变量之间只能用空格来分隔。
- 未列出variables时，表示将磁盘文件中的所有变量都调入工作空间。

e.g. load data.mat



武汉大学

Wuhan University

3、退出工作空间

- quit 或 exit



武汉大学

Wuhan University

窗口命令

clf : 清除当前图形窗口 (Figure) 中所有非隐藏图形对象 ;

close : 关闭当前的图形窗口 ;

close all : 关闭所有的图形窗口 ;

clc : 清除命令窗口中的内容 , 光标回到窗口的左上角 ;

home : 光标回到窗口的左上角 ;



武汉大学

Wuhan University

显示格式设置命令

- 方向键和控制键可以编辑修改已输入的命令

more off : 不允许分页 more on : 允许分页

more(n) : 指定每页输出的行数

e.g. more(2);

x=3-5

echo on : 显示正在执行的M文件语句

format type : 控制输出变量的数据格式



武汉大学

Wuhan University

文件管理命令

- 文件管理的命令，包括列文件名、显示或删除文件、显示或改变当前目录等。（ what、dir、type、delete、cd、which ）

what：显示当前目录下所有与matlab相关的文件及它们的路径。

dir：显示当前目录下所有的文件

which：显示某个文件的路径

e.g. which data.mat



武汉大学

Wuhan University

cd path : 由当前目录进入path目录

cd .. : 返回上一级目录

cd : 显示当前目录

type filename : 在命令窗口中显示文件内容

e.g. type data.mat

delete filename : 删除文件filename



武汉大学

Wuhan University

帮助命令

help : 在命令窗口中显示MALTAB的相关帮助信息

e.g. help format

doc : 在help浏览器中显示某个函数的联机帮助文档

e.g. doc format



武汉大学

Wuhan University

§2.2 MATLAB数值计算

2.2.1 基本计算

复数

- 格式一：经典教科书的直角坐标表示法

```
>> Z1=3+4i
```

- 格式二：采用运算符构成的直角坐标表示法和极坐标表示法

```
>> Z2=1+2*i      %运算符构成的直角坐标表示法
```

```
>> Z3=2*exp(i*pi/6) %运算符构成的极坐标表示法  
( 推荐 , 宜于数值矩阵描述 )
```

```
>> Z=Z1*Z2/Z3
```



- `real(z)`
给出复数 z 的实部, $a = r \cos \theta$
- `imag(z)`
给出复数 z 的虚部, $b = r \sin \theta$
- `abs(z)`
给出复数 z 的模, $r = \sqrt{a^2 + b^2}$
- `angle(z)`
给出复数 z 的相位 θ

e.g. $z = 5 \cdot \exp(i \cdot 4 \cdot \pi / 4);$

`real(z)`

`imag(z)`

`abs(z)`

`angle(z)`



变量

- 变量的命名：变量的名字必须**以字母开头**（不能超过19个字符），之后可以是任意字母、数字或下划线；变量名称区分字母的大小写；变量中不能包含有标点符号。

e.g. X, x, data2006_9_6

- 变量命名不要使用MATLAB中已有的函数名或命令名（如clear），
- 不要对以下的特殊变量赋值
 - ans：用于结果的缺省变量名
 - i、j：虚数单位
 - pi：圆周率
 - nargin：函数的输入变量个数
 - eps：计算机的最小数



武汉大学

Wuhan University

nargout : 函数的输出变量个数

inf : 无穷大

realmin : 最小正实数

realmax : 最大正实数

nan : 不定量

flops : 浮点运算数



武汉大学

Wuhan University

■ 常用的数学运算符

$+$, $-$, $*$ (乘) , $/$ (左除) , \backslash (右除) , $^$ (幂)

乘号不可省

- 在运算式中，MATLAB通常不需要考虑空格；多条命令可以放在一行中，它们之间需要用分号隔开；逗号告诉MATLAB显示结果，而分号则禁止结果显示。

e.g. $x = 3 + 5 * \exp(7^3 / 1.57) - 9$



常用数学函数

- $\text{abs}(x)$ 取绝对值或取模
- $\sin(x)$, $\cos(x)$, $\tan(x)$, $\text{asin}(x)$, $\text{acos}(x)$, $\text{atan}(x)$ 三角函数
- $\text{sqrt}(x)$ 平方函数
- $\text{exp}(x)$ 指数函数
- $\text{sign}(x)$ 符号函数，正数返回1，0返回0，负数返回-1
- $\log(x)$ e底对数函数
- $\log_{10}(x)$ 10底对数函数
- $\text{conj}(x)$ 共扼复数函数， $\text{conj}(a+b*i)=a-b*i$



2.2.2 数组计算

数组的创建

■ 逐个元素输入法

```
>>x=[3 4+3i abs(-1) pi/3]
```

数组生成不但可以使用纯数字（含复数），也可以使用变量（或者说采用一个表达式）。数组的元素直接排列在方括号内，行与行之间用分号隔开，每行内的元素使用空格或逗号隔开。大的数组可以用分行输入，回车键代表分号。

```
>> a=1; b=2; c=3;
```

```
>> x=[5 b c; a*b a+c c/b]
```



■ 语句生成法

(1) **冒号法**：用线性等间距生成向量数组
(start:step:end)。其中start为起始值，step为步长，end为终止值。当步长为1时可省略step参数；另外step也可以取负数。

```
>> a=1:2:10
```

(2) **线性赋值语句**：a=linspace(n1,n2,n)

在线性空间上，行矢量的值从n1到n2，数据个数为n，缺省n为100。

```
>> a=linspace(1,10,10)
```



(3) **对数赋值语句** : $a = \text{logspace}(n1, n2, n)$

在对数空间上，行矢量的值从 10^{n1} 到 10^{n2} ，数据个数为 n ，缺省 n 为50。这个指令为建立对数频域轴坐标提供了方便。

`>>a=logspace(1,3,3)`

(4) **一些常用的特殊数组**

单位数组 : $\text{eye}(m)$; $\text{eye}(m, n)$ - m, n 较小的对角后补0

零数组 : $\text{zeros}(m, n)$; $\text{zeros}(m)$

一数组 : $\text{ones}(m, n)$; $\text{ones}(m)$

对角数组 : 对角元素向量 $V = [a1, a2, \dots, an]$ $A = \text{diag}(V)$

随机数组 : $\text{rand}(m, n)$ 产生一个 $m \times n$ 的均匀分布的随机数组



■ 生成高阶数组

(1) 利用ones/zeros函数生成数组

ones(x,y,z) 所生成数组初值全为1,zeros函数生成数组初值全为0

```
>> ones(1,4,2)
```

特殊用法ones (x:y) 生成y-x+1维数组 , 每一维度的数值为x~y相应的数

```
>> ones(1:3)
```

(2) 利用rand函数生成数组

rand (x,y,z) 所生成数组初值全为0~1的随机值
用法与效果同ones



数组的大小

- `[m,n]=size(a,x)` : 返回矩阵的行列数m与n, 当x=1, 则只返回行数m, 当x=2, 则只返回列数n。
- `length(a)=max(size(a))` : 返回行数或列数的最大值。
- `rank(a)` : 求矩阵的秩

```
>> a=[1 2 3;3 4 5];
```

```
>> [m,n]=size(a)
```

```
>> length(a)
```

```
>> max(size(a))
```

```
>> rank(a)
```



武汉大学

Wuhan University

数组的寻访和赋值

■ 绝对地址寻访

(1) 单一元素寻访

$x(n)$ %寻访第 n 个元素，元素地址按维数方向统一排列

```
>> x=rand(2,4)
```

```
x =
```

```
0.5155    0.4329    0.5798    0.5298
```

```
0.3340    0.2259    0.7604    0.6405
```

```
>> x(3)
```

```
ans =
```

```
0.4329
```



(2) 多元素寻访

x([n1,n2,n3]) %逐一地址寻访多个元素，使用[]和，

```
>> x([1,3,6])
```

```
ans =
```

```
0.5155 0.4329 0.7604
```

x(n1:n2) %寻访连续地址元素

```
>> x(2:4)
```

```
ans =
```

```
0.3340 0.4329 0.2259
```

x(n:end) %寻访第n-1个以后的元素

```
>> x(6:end)
```

```
ans =
```

```
0.7604 0.5298 0.6405
```




武汉大学

Wuhan University

(3) 重复元素寻访

x ([n1,n2,n3,n2,n4]) %可寻访重复元素

>> x([1,3,5,3,7])

ans =

0.5155 0.4329 0.5798 0.4329 0.5298



武汉大学

Wuhan University

- **相对地址寻访**（以二维数组为例）

- （1）指定行列地址寻访单一元素

- $x(n1, n2)$

- （2）寻访某一行上所有元素

- $x(n1, :)$

- （3）寻访某一列上所有元素

- $x(:, n2)$



武汉大学

Wuhan University

■ 数组的赋值

对数组中元素的赋值可在寻访到后直接赋值；对整个数组可用 $x(:)$ 全赋值。

```
>> x=rand(2,4)
```

```
x =
```

```
0.9501    0.6068    0.8913    0.4565  
0.2311    0.4860    0.7621    0.0185
```

```
>> x(:)=2:9
```

```
x =
```

```
2    4    6    8  
3    5    7    9
```



武汉大学

Wuhan University

数组和矩阵的运算

- 对数组中元素的运算遵从代数计算规则；对矩阵遵从矩阵运算规则。对规则相同的运算（如加，减），二者的运算指令形式相同；对于运算规则不同的运算（如乘、除），对元素的运算以.标明。

指令	含 义	指令	含 义
A'	非共轭转置	A'	共轭转置
$A=S$	把标量 S 赋给 A 的每个元素	$A=S$	把标量 S 赋给 A 的每个元素
$S+A, A-S$	标量 S 分别和 A 的每个元素之和，之差	$S+A, A-S$	标量 S 分别和 A 的每个元素之和，之差
$S*A$	标量 S 分别和 A 的元素之积	$S*A$	标量 S 分别和 A 的每个元素之积
$S/A, A./S$	A 的元素分别除以 S	$S*inv(A)$	A 阵的逆矩阵
$A.^n$	A 的每个元素自乘 n 次	A^n	A 阵为方阵时，自乘 n 次
$A.^p$	对 A 各元素分别求非整数幂	A^p	方阵 A 的非整数乘方
$p.^A$	以 p 为底，分别以 A 的元素为指数求幂	P^A	A 阵为方阵时，标量的矩阵乘方
$A+B$	对应元素相加	$A+B$	矩阵相加
$A-B$	减	$A-B$	矩阵相减
$A.*B$	乘	$A*B$	维数相同矩阵相乘
$A./B$	A 的元素除以 B 的对应元素	A/B	A 右除 B
$B./A$	A 的元素除以 B 的对应元素	B/A	A 左除 B
$Sqrt(A)$	对 A 的各个元素求平方根	$Sqrtm(A)$	A 的矩阵平方根函数
$F(A)$	求 A 各个元素的函数值	$Funm(A, 'FN')$	一般矩阵函数



武汉大学

Wuhan University

数组计算

```
>> x=rand(2,4)
```

```
x =
```

```
0.1389    0.1987    0.2722    0.0153  
0.2028    0.6038    0.1988    0.7468
```

```
>> y=ones(2,4)
```

```
y =
```

```
1     1     1     1  
1     1     1     1
```

```
>> z=x.*y    %元素相乘
```

```
z =
```

```
0.1389    0.1987    0.2722    0.0153  
0.2028    0.6038    0.1988    0.7468
```



武汉大学

Wuhan University

%%%%%%%%矩阵计算

```
>> y=ones(4,2)
```

y =

1	1
1	1
1	1
1	1

```
>> z=x*y      %矩阵相乘
```

z =

0.6251	0.6251
1.7522	1.7522



2.2.3 数值计算函数

MATLAB中数值计算函数众多，这也是MATLAB在计算领域得到广泛应用的重要原因。

- Arrays and Matrices(数组与复数矩阵)类
- Linear Algebra (线性代数) 类
- Elementary Math (初等数学) 类
- Nonlinear Numerical Methods (非线性数值计算) 类
- Sparse Matrices (稀疏矩阵) 类
- Data Analysis and Fourier Transforms
- Coordinate System Conversion
- Specialized Math
- Constants
- Polynomial
-



武汉大学

Wuhan University

以Elementary Math为例，又包含小类；

- Trigonometric类：sin等三角函数
- Exponential类：幂运算、对数运算函数
- Complex类：对复数求实部、虚部、幅角等的函数
- Rounding and Remainder类：对非整数取整、除法后取余等的函数
- Discrete Math类：求阶乘、逻辑真值、质数生成与判断的函数



武汉大学

Wuhan University

```
>> p=primes(8) %质数函数primes , 生成比8小的质数 ,  
p =  
    2    3    5    7  
  
>> p=sin(p) %正弦函数sin ,  
p =  
    0.9093    0.1411   -0.9589    0.6570  
  
>> p=abs(p) %绝对值函数abs ,  
p =  
    0.9093    0.1411    0.9589    0.6570  
  
>> p=log(p) 对数函数log ,  
p =  
   -0.0951   -1.9581   -0.0419   -0.4201
```



武汉大学

Wuhan University

```
>> p=exp(p)
```

指数函数exp ,

```
p =
```

```
0.9093 0.1411 0.9589 0.6570
```

```
>> p=fix(p)
```

取整函数fix , 小数部分忽略。

```
p =
```

```
0 0 0 0
```



§2.3 MATLAB矩阵&函数分析

2.3.1 MATLAB矩阵分析

常用分析函数

- 矢量范数和矩阵范数

$$\|x\|_p = \left(\sum_i x_i^p \right)^{1/p} \quad \|A\|_p = \max_x \frac{\|Ax\|_p}{\|x\|_p}$$

矢量x和矩阵A的p范数。

范数用来研究线性方程组的解。如线性方程组的简单迭代法等。 $p = 2$ 为欧拉范数。



武汉大学

Wuhan University

`norm(A,p)` %对 $1 \leq p < \infty$ ，得到其范数，P缺省值为2
`norm(A,inf)` %得到 $\max(\text{abs}(A))$;

```
>> X=[2 0 -1];  
>> N=[norm(X,1) norm(X) norm(X,inf)]
```

```
N =  
    3.0000    2.2361    2.0000
```

```
>> A=fix(10*rand(3,2))
```

```
A =  
    4    4  
    0    6  
    8    7
```

```
>> n=[norm(A,1) norm(A) norm(A,inf)]
```

```
n =  
   17.0000   12.7774   15.0000
```



武汉大学

Wuhan University

■ 矩阵求逆

$V = \text{inv}(A)$

■ 矩阵的行列式

$X = \text{det}(A)$

```
>> A=pascal(4) %按照pascal三角生成矩阵
```

```
A =
```

```
1 1 1 1
1 2 3 4
1 3 6 10
1 4 10 20
```

```
>> V=inv(A)
```

```
V =
```

```
4.0000 -6.0000 4.0000 -1.0000
-6.0000 14.0000 -11.0000 3.0000
4.0000 -11.0000 10.0000 -3.0000
-1.0000 3.0000 -3.0000 1.0000
```

```
>> X=det(A)
```

```
X =
```

```
1
```



武汉大学

Wuhan University

- 伪逆矩阵

$C = \text{pinv}(A)$

对不存在逆矩阵的非方阵，MATLAB将 $\text{inv}(A' * A) * A'$ 定义为A的伪逆，其在一定程度上可代表矩阵的逆。

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

```
A =
```

```
1    2    3
```

```
4    5    6
```

```
>> C=pinv(A)
```

```
C =
```

```
-0.9444    0.4444
```

```
-0.1111    0.1111
```

```
0.7222   -0.2222
```



武汉大学

Wuhan University

■ 求解线性方程组

$$6x_1 + 3x_2 + 4x_3 = 3$$

$$-2x_1 + 5x_2 + 7x_3 = -4$$

$$8x_1 - 4x_2 - 3x_3 = -7$$

将方程组写成矩阵的形式， $AX=B$ ，有

```
>> A=[6 3 4;-2 5 7;8 -4 -3];
```

```
>> B=[3;-4;-7];
```

```
>> X=B/A
```

错误使用 /
矩阵维度必须一致。

```
>> X=A\B
```

X =

0.6000

7.0000

-5.4000



武汉大学

Wuhan University

■ 矩阵的LU分解

将**方阵**分解成上 (U)、下 (L) 三角矩阵

$[L,U]=lu(A)$ %标准U三角矩阵和置换L矩阵，满足 $L*U=A$

```
>> A=[1 2 3;4 5 6;7 8 9];
```

```
>> [L,U]=lu(A)
```

L =

0.1429	1.0000	0
0.5714	0.5000	1.0000
1.0000	0	0

U =

7.0000	8.0000	9.0000
0	0.8571	1.7143
0	0	0.0000



武汉大学

Wuhan University

$[L,U,P]=lu(A)$ %标准U三角矩阵、标准L三角矩阵和置换矩阵P，满足 $L*U=P*A$

```
>> A=[1 2 3;4 5 6;7 8 9];
```

```
>> [L,U,P]=lu(A)
```

L =

```
1.0000    0    0
0.1429    1.0000    0
0.5714    0.5000    1.0000
```

U =

```
7.0000    8.0000    9.0000
    0    0.8571    1.7143
    0    0    0.0000
```

P =

```
0    0    1
1    0    0
0    1    0
```



■ 矩阵的正交分解

将矩阵分解成一个正交矩阵与另一个矩阵的乘积

$[Q,R]=qr[A]$ %正交矩阵Q和上三角矩阵R,

$A=Q*R, Q' * Q=I$

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

```
>> [Q,R]=qr(A)
```

```
Q =
```

```
-0.1690    0.8971    0.4082  
-0.5071    0.2760   -0.8165  
-0.8452   -0.3450    0.4082
```

```
R =
```

```
-5.9161   -7.4374  
    0    0.8281  
    0     0
```



■ 矩阵的特征值分析

$$[V,D]=\text{eig}(A)$$

对角阵D的对角元为A的特征值，V的列向量是相应的特征向量， $A*V=V*D$ 。

```
>> A=[-1,-6,0;-16,2,6;-10,20,-5];
```

```
>> [V,D]=eig(A)
```

V =

-0.2344	0.4230	-0.2251
0.6277	0.2385	-0.5136
0.7423	0.8742	0.8280

D =

15.0695	0	0
0	-4.3827	0
0	0	-14.6868



武汉大学

Wuhan University

其他矩阵分析函数

- 矩阵左右翻转函数 `fliplr`
- 矩阵上下翻转函数 `flipud`
- 矩阵阶数重组函数 `reshape`
- 矩阵整体反时针旋转函数 `rot90`
- 对角矩阵和矩阵的对角化函数 `diag`
- 取矩阵的左下三角部分函数 `tril`
- 取矩阵的右上三角部分函数 `triu`
- 矩阵一列化符号 :



2.3.2 MATLAB函数分析

- 函数极值分析

单变函数求极小值

$[x,y]=\text{fminbnd}(\text{'fun'},x1,x2,\text{options})$

多变量函数求极小值

$[x,y]=\text{fminsearch}(\text{'fun'},x0,\text{options})$

- 单变函数零点分析

$[x,\text{fval}]=\text{fzero}(\text{'fun'},x0,\text{options})$



■ 函数的数值积分

思想：采用不同的方法将积分化为求和

低阶数值积分函数 $q = \text{quad}('fun', a, b, tol)$

按照自适应Simpson积分方法给出 $[a, b]$ 区间的数值积分，
积分绝对误差为 tol

高阶数值积分函数 $q = \text{quadl}('fun', a, b, tol)$

按照自适应Lobatto积分方法给出 $[a, b]$ 区间的数值积分，
积分绝对误差为 tol

梯形面积积分函数 $T = \text{trapz}(X, Y)$

按照梯形积分方法，根据 X 计算 Y 的积分



双重积分函数 $q = \text{dblquad}('fun', xa, xb, ya, yb, tol)$

在精度 tol 内求解函数 fun 在 $[xa, xb]$ 和 $[ya, yb]$ 内的双重积分。

■ 数值/函数的 n 阶微分

$y = \text{diff}(x, n)$

除数值微分外， diff 还可结合 syms 函数完成函数表达式的微分。

```
>> syms x
```

```
>> diff(x^4, 2)
```

```
ans =
```

```
12*x^2
```



武汉大学

Wuhan University

■ 微分方程求解

常微分方程的初值问题

非刚性问题 ode45 , ode23 , ode113

刚性问题 ode15s ode23s ode23t ode23tb

$[T,Y]=ode45('odefun',tspan,y0)$

采用ode45的方法在tspan给定的范围内，按照初始条件y0求解方程odefun

常微分方程的边界问题

偏微方程的求解



2.3.2 MATLAB数据插值与拟合

■ 数据插值函数

一元插值函数

`y=interp1(X,Y,xi,method)`

采用method指定方法对点集(X,Y)插值计算xi上的函数值y。

`pp=interp1(X,Y, method,'pp')`

采用method指定方法对点集(X,Y)进行插值，但返回结果为分段多项式。

Method : 'nearest' 最近邻插值 'linear' 分段线性插值

'spline' 样条插值 'pchip' 三次Hermite多项式插值
'cubic' 三次Hermite多项式插值



武汉大学

Wuhan University

二元插值函数

$z = \text{interp2}(X, Y, Z, x_i, y_i, \text{method})$

采用method指定方法对点集(X,Y,Z)插值计算 x_i, y_i 上的函数值z。

Method : 'nearest' 最近邻插值 'linear' 分段线性插值
'cubic' 三次Hermite多项式插值

■ 其他插值方法：

拉格朗日插值 艾特肯插值

牛顿插值 埃尔米特插值



武汉大学

Wuhan University

■ 函数逼近与曲线拟合

对区间 $[a,b]$ 上的连续函数 $f(x)$ ，如采用一简单函数去近似复杂的 $f(x)$ ，则称为函数逼近。

如 $f(x)$ 的表达式未知，采用一条曲线来描述 $f(x)$ 的数据点集，则称为曲线拟合。

傅立叶逼近

$$[A_0, A, B] = \text{FZZ}(\text{func}, T, n)$$

其中 func 为已知函数， T 为已知函数的周期， n 为展开级数的项数， A_0 为展开后的常数项， A 为展开后的余弦项系数， B 为展开后的正弦项系数。



武汉大学

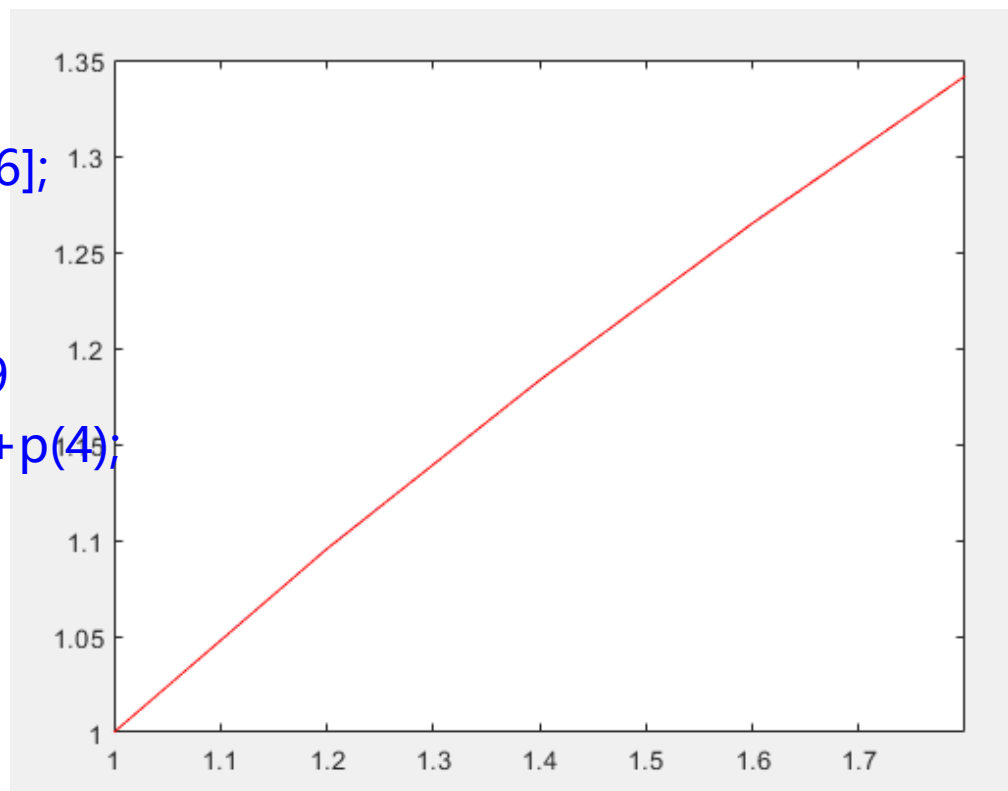
Wuhan University

最小二乘法拟合

$P = \text{polyfit}(X, Y, N)$

(X,Y)为待拟合的数据点集，N为多项式次数，P为拟合后的多项式系数。

```
>> x=1:0.2:1.8;  
>> y=[1 1.0954 1.1832 1.2649 1.3416];  
>> p=polyfit(x,y,3)  
p =  
    0.0271   -0.1914    0.7994    0.3649  
>> y_f=p(1)*x.^3+p(2)*x.^2+p(3)*x+p(4);  
>> plot(x,y,'b',x,y_f,'r')
```





§2.4 MATLAB符号运算

MATLAB符号计算是对未赋值的符号对象(可以是常数、变量、表达式)进行运算和处理。MATLAB具有符号数学工具箱(Symbolic Math Toolbox)，将符号运算结合到MATLAB的数值运算环境。

2.4.1 创建符号常量

符号常量是不含变量的符号表达式。

用sym命令来创建符号常量。

```
sym( '常量' )           %创建符号常量
```

```
>> a=sym('sin2')
```

```
a =
```

```
sin2
```



武汉大学

Wuhan University

用syms创建多个符号变量

```
syms( 'arg1' , 'arg2' , ...,参数)
```

%把字符变量定义为符号变量

```
syms arg1 arg2 ...,参数
```

%把字符变量定义为符号变量的简洁形式

```
>>syms a b c x;
```

%创建多个符号变量

```
f2=a*x^2+b*x+c
```

%创建符号表达式

```
f2 =
```

```
a*x^2+b*x+c
```

```
>>syms('a','b','c','x');
```

```
f3=a*x^2+b*x+c;
```

%创建符号表达式



武汉大学

Wuhan University

用sym和syms命令也可以创建符号矩阵。

```
>>A=sym('A',3)
```

A =

[A1_1, A1_2, A1_3]

[A2_1, A2_2, A2_3]

[A3_1, A3_2, A3_3]

```
>>syms a b c d
```

```
A=[a b;c d]
```

A =

[a, b]

[c, d]



2.4.2 符号表达式的运算

■ 符号运算适用于大部分代数运算

```
>> f=str2sym('2*x^2+3*x+4')
f =
2*x^2+3*x+4
>> g=str2sym('5*x+6')
g =
5*x+6
>> f+g           %符号表达式相加
ans =
2*x^2+8*x+10
>> f*g           %符号表达式相乘
ans =
(2*x^2+3*x+4)*(5*x+6)
```

```
>> syms a11 a12 a21 a22;
>> A=[a11 a12;a21 a22] %创建符号矩阵
A =
[ a11, a12]
[ a21, a22]
>> det(A)         %计算行列式
ans =
a11*a22-a12*a21
>> A.'            %计算非共轭转置
ans =
[ a11, a21]
[ a12, a22]
>> eig(A)         %计算特征值
```




武汉大学

Wuhan University

■ 求反函数和复合函数

`finverse(f,v)` %对指定自变量v的函数f(v)求反函数

`compose(f,g)` %计算复合函数f(g(x))

■ 符号极限

Symbolic Math Toolbox提供了直接求表达式极限的函数**limit**

limit 函数的用法表

表达式	函数格式	说明
$\lim_{x \rightarrow 0} f(x)$	<code>limit(f)</code>	对 x 求趋近于 0 的极限
$\lim_{x \rightarrow a} f(x)$	<code>limit(f,x,a)</code>	对 x 求趋近于 a 的极限，当左右极限不相等时极限不存在。
$\lim_{x \rightarrow a^-} f(x)$	<code>limit(f,x,a,'left')</code>	对 x 求左趋近于 a 的极限
$\lim_{x \rightarrow a^+} f(x)$	<code>limit(f,x,a,'right')</code>	对 x 求右趋近于 a 的极限



■ 符号微分

`diff(f)/diff(f,t)` %求f对自由变量/变量t的一阶微分

diff(f,n)/diff(f,t,n) %求f对自由变量/变量t的n阶微分

■ 符号积分

```
int(f, 't' )           %求符号变量t的不定积分
```

```
int(f, 't',a,b) %求符号变量t的积分
```

int(f, 't' , 'm' , 'n') %求符号变量t的符号范围积分

说明：与符号微分相比，符号积分复杂得多，有时可能不存在，即使存在也可能限于条件，MATLAB无法顺利得出。当MATLAB不能找到积分时，它将给出警告提示并返回该函数的原表达式。



■ 符号级数

`symsum(s,x,a,b)` %计算表达式s的级数和

```
>> syms x k
```

```
>> s1=symsum(1/k^2,1,10) %计算级数的前10项和
```

```
s1 =
```

```
1968329/1270080
```

```
>> s2=symsum(x^k,k,0,inf) %计算对k为自变量的级数和
```

```
s2 =
```

```
piecewise(1 <= x, Inf, abs(x) < 1, -1/(x - 1))
```

`taylor (F,x,n)` %求泰勒级数展开

```
>> syms x
```

```
>> s1= taylor(exp(x),x,'OrderMode','Relative','Order',8) %展开前8项
```

```
s1 =
```

```
x^7/5040 + x^6/720 + x^5/120 + x^4/24 + x^3/6 + x^2/2 + x + 1
```



■ 符号代数方程

`solve('eq' , ' v')` %求方程关于指定变量的解

`solve('eq1' , ' eq2' , ' v1' , ' v2' ,...)`
%求方程组关于指定变量的解

■ 符号常微分方程

`dsolve('eq' , ' con' , ' v')` %求解微分方程

`dsolve('eq1,eq2...' , ' con1,con2...' , ' v1,v2...')`
%求解微分方程组



Wuhan University

■ Symbolic Math Toolbox中的算术运算方式

- 数值型：浮点运算。
- 有理数型：精确符号运算。
- VPA型：任意精度运算。

```
>> a1 = 2/3 %数值型
```

a1 =

0.6667

```
>> a2 = sym(2/3)      %有理数型
```

 $a_2 =$

2/3

```
>> a3 =vpa('2/3',32)    %VPA型
```

 $a_3 =$

.66666666666666666666666666666667



武汉大学

Wuhan University

- **符号运算与数值运算的区别：**
 - ✓ 符号运算不需要进行数值运算，不会出现截断误差，因此符号运算是非常准确的。
 - ✓ 符号运算可以得出完全的封闭解或任意精度的数值解。
 - ✓ 符号运算的时间较长，而数值型运算速度快。



武汉大学

Wuhan University

§2.5 MATLAB图形绘制

MATLAB提供了丰富的绘图功能

- help graph2d可得到所有画二维图形的命令
- help graph3d可得到所有画三维图形的命令



2.5.1 二维绘图基本指令

基本的二维绘图命令plot

`plot(x1,y1,option1,x2,y2,option2,...)`

- `x1,y1`给出的数据分别为`x,y`轴坐标值，`option1`为选项参数，以逐点连折线的方式绘制1个二维图形；同时类似地绘制第二个二维图形，……等。
- 这是`plot`命令的完全格式，在实际应用中可以根据需要进行简化。比如：`plot(x,y);`
`plot(x,y,option)`



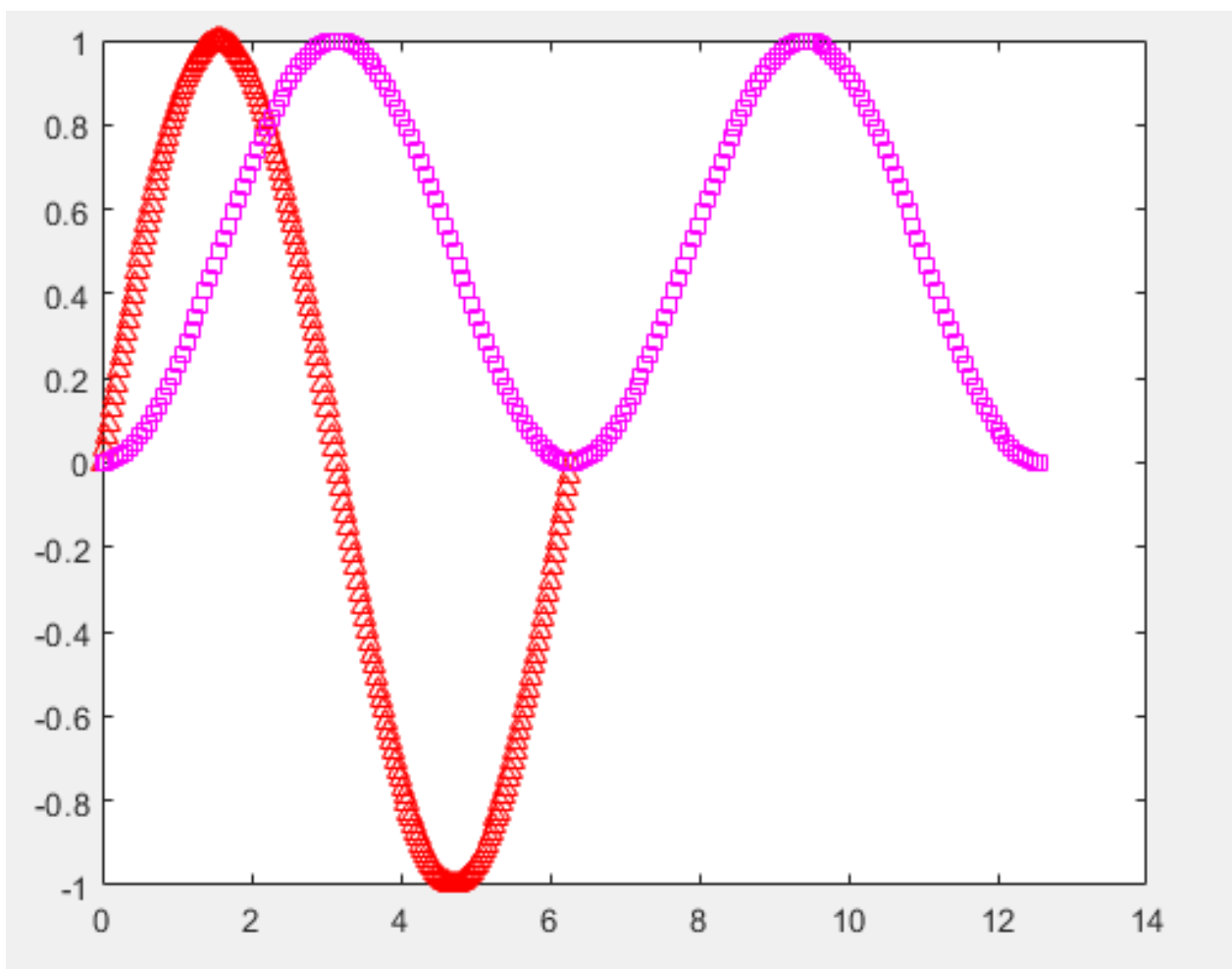
- 选项参数option定义了图形曲线的颜色、线型及标示符号，它由一对单引号括起来。详细参数表见p75表4.1。

```
>> t=0:pi/100:2*pi; %绘制正弦曲线和正弦平方曲线
>> y1=sin(t);
>> y2=sin(t).*sin(t);
>> plot(t);
>> plot(t,y1);          %绘制正弦曲线
>> plot(t,y1,t,y2);     %同一窗口内绘制正弦曲线和正弦
    平方曲线
>> plot(t,y1,'r^',2*t,y2,'mS'); %红色上向三角绘制y1,紫
    色正方形绘制y2
```



武汉大学

Wuhan University





多子图绘制命令subplot

subplot(m,n,k) %作出 (m*n) 幅子图中的第 k 幅图形

subplot('position' , [left bottom width height]) %在指定位置作出子图

>> t=0:pi/100:2*pi; %绘制正弦曲线和正弦平方曲线

>> y1=sin(t);

>> y2=sin(t).*sin(t);

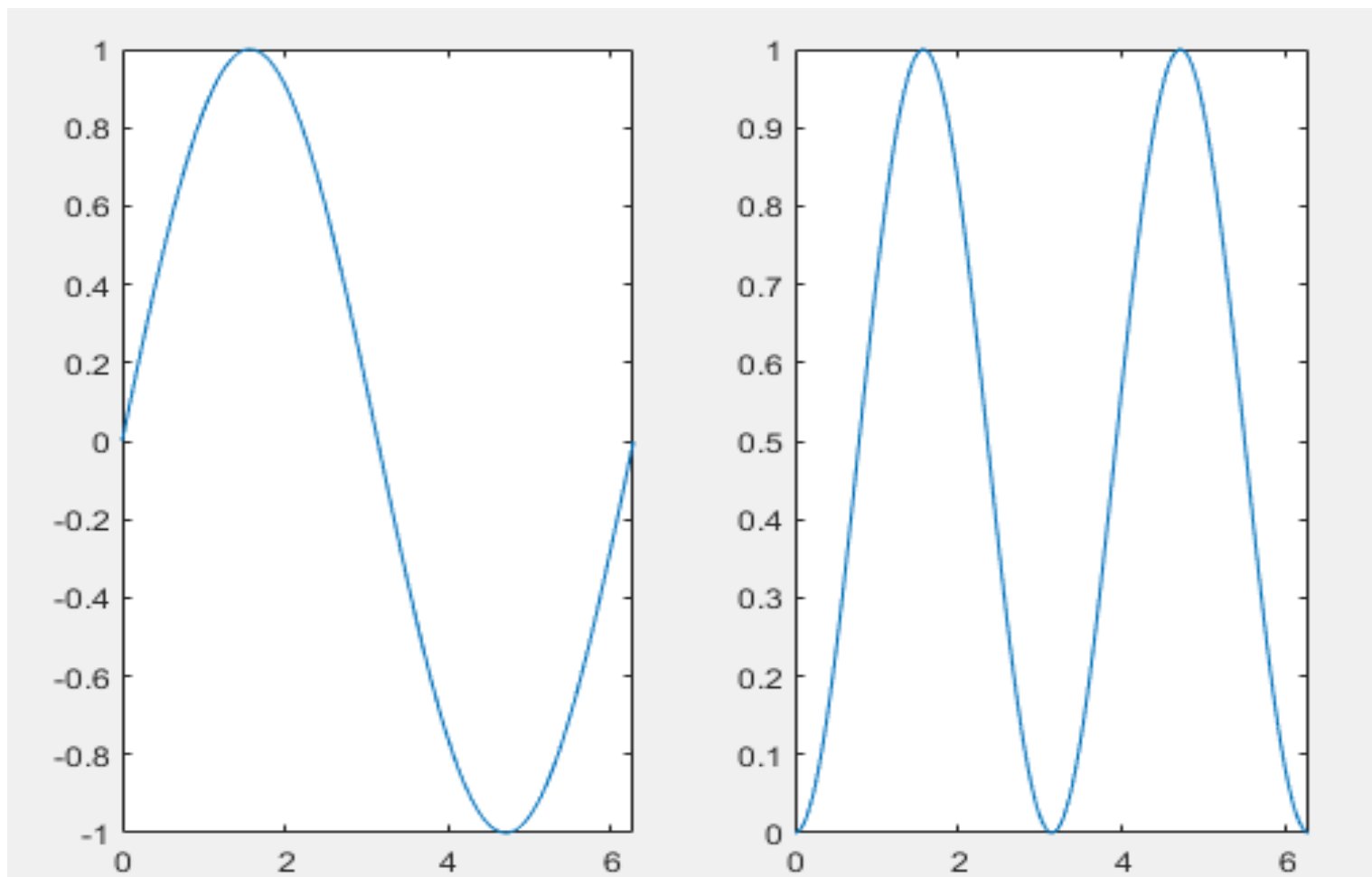
>> subplot(1,2,1),plot(t,y1); % (1*2) , 一行两列, 横排

>> subplot(1,2,2),plot(t,y2);



武汉大学

Wuhan University

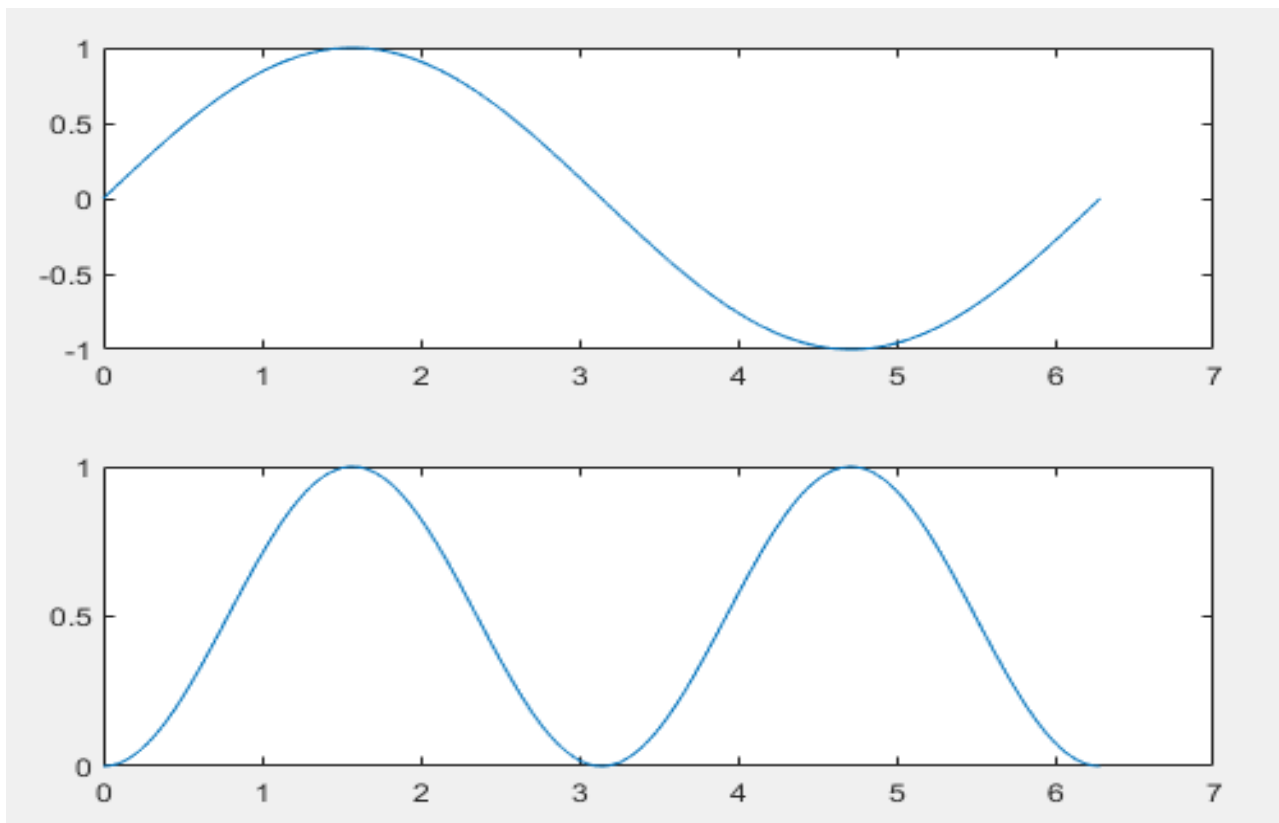




武汉大学

Wuhan University

```
>> subplot(2,1,1),plot(t,y1); % ( 2*1 ) , 两行一列 , 横排  
>> subplot(2,1,2),plot(t,y2);
```





选择图像

- `figure (1) ; figure (2) ; ... ; figure(n)`
打开不同的图形窗口，以便绘制不同的图形。
 - >> `figure(1)`
 - >> `plot(t,y1)`
 - >> `figure(2)`
 - >> `plot(t,y2)`
- **grid on**：在所画出的图形坐标中加入栅格(加栅格标尺)
- **grid off**：除去图形坐标中的栅格
- **hold on**：把当前图形保持在屏幕上不变，同时允许在这个坐标内绘制另外一个图形。（图形叠加）
- **hold off**：使新图覆盖旧的图形



武汉大学

Wuhan University

设定轴的范围

- `axis([xmin xmax ymin ymax])`
- `axis('equal')`：将x坐标轴和y坐标轴的单位刻度大小调整为一样。



文字标示

- **title('字符串')**

在所画图形的最上端显示说明该图形标题的字符串。

- **xlabel('字符串'), ylabel('字符串')**

设置x, y坐标轴的名称。

输入特殊的文字需要用反斜杠(\)开头。

- **legend('字符串1','字符串2',..., '字符串n')**

在屏幕上开启一个小视窗, 然后依据绘图命令的先后次序, 用对应的字符串区分图形上的线。

- **text(x,y,'字符串')**

在图形的指定坐标位置(x,y)处, 标示单引号括起来的字符串。

- **gtext('字符串')**

利用鼠标在图形的某一位置标示字符串。



武汉大学

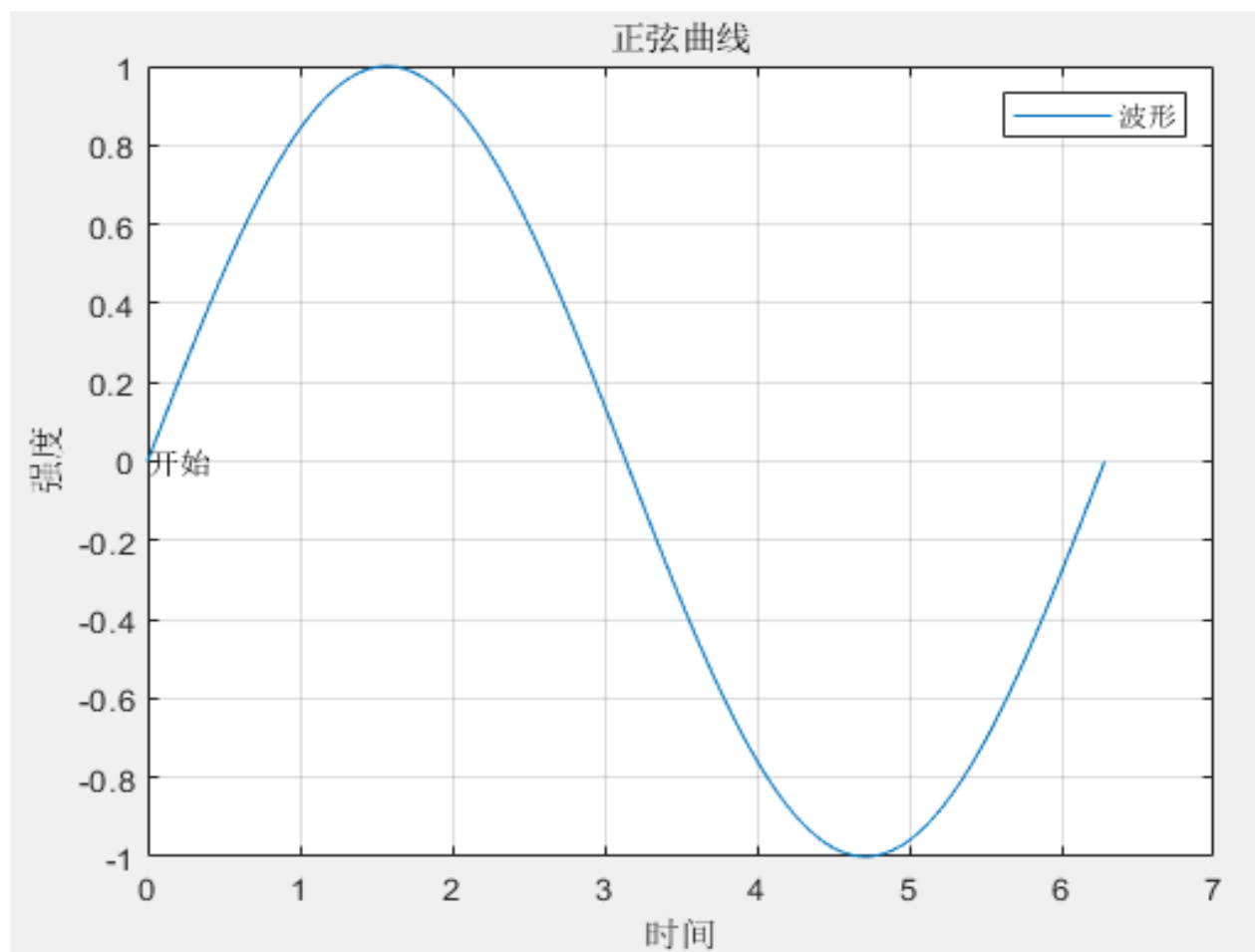
Wuhan University

```
>> t=0:pi/100:2*pi;  
>> y=sin(t);  
>> plot(t,y);  
>> title('正弦曲线');  
>> xlabel('时间')  
>> ylabel('强度')  
>> legend('波形')  
>> text(0,0,'开始')  
>> grid on
```



武汉大学

Wuhan University





武汉大学

Wuhan University

semilogx : 绘制以x轴为对数坐标（以10为底），y轴为线性坐标的半对数坐标图形。

semilogy : 绘制以y轴为对数坐标（以10为底），x轴为线性坐标的半对数坐标图形。



武汉大学

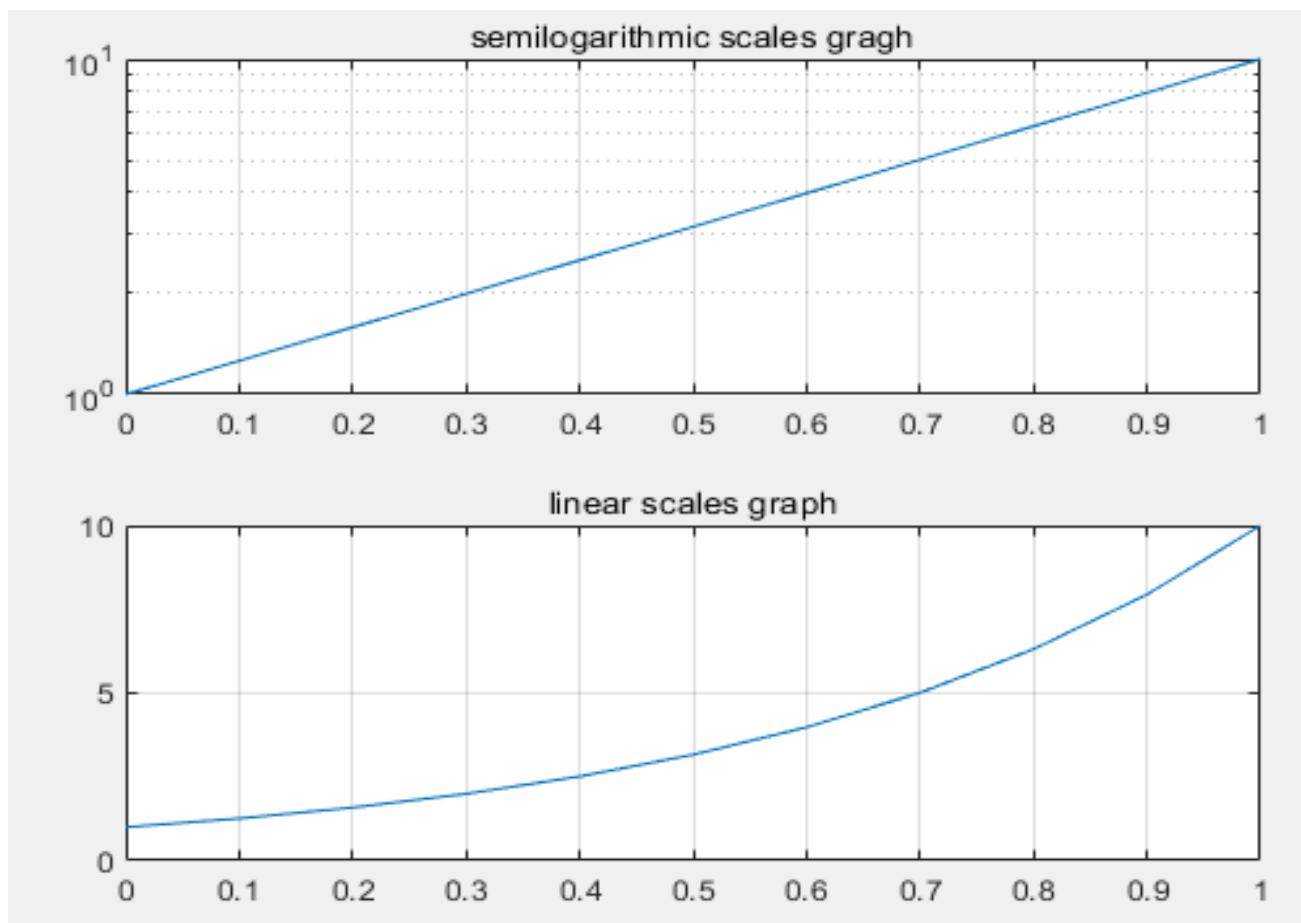
Wuhan University

```
clear
close
clc
x=0:0.1:1;
y=10.^x;
subplot(2,1,1)
semilogy(x,y)           %以y轴为对数坐标
title('semilogarithmic scales graph')
grid on
subplot(2,1,2)
plot(x,y)
title('linear scales graph') %线性坐标
grid on
```



武汉大学

Wuhan University





武汉大学

Wuhan University

应用型绘图指令：可用于数值统计分析或离散数据处理

- `hist(y,x); stairs(x,y); stem(x,y)`
- 补充说明：对于图形的属性编辑同样可以通过在图形窗口上直接进行。但图形窗口关闭之后编辑结果不会保存。



2.5.2 三维绘图

基本的三维绘图命令

- `plot3(x,y,z)`
- `plot3(x1,y1,z1,x2,y2,z2,...)`

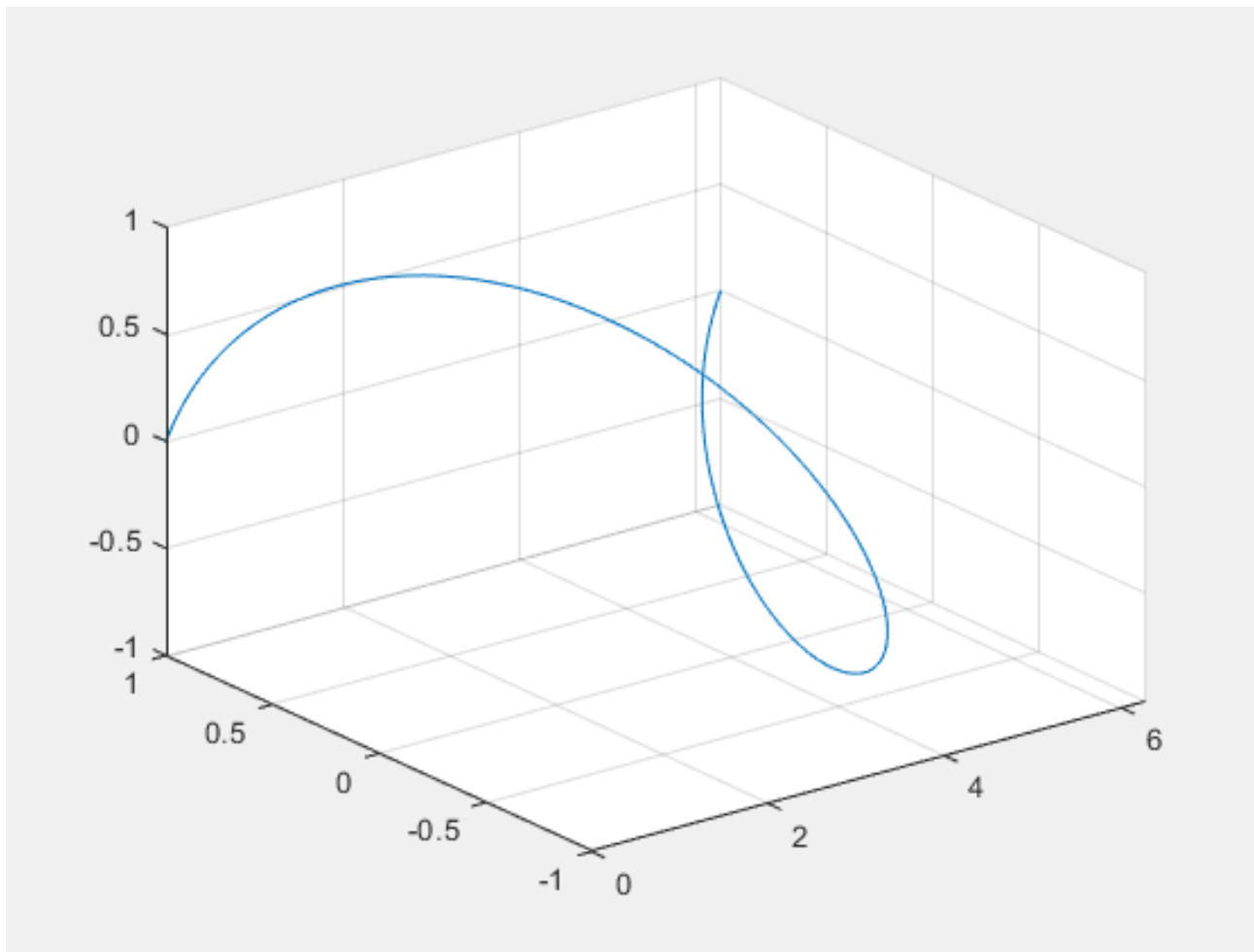
`plot3`的应用方法和`plot`相似，只是在使用时多增加了一“维数 - Z”。

```
>> x=0:pi/100:2*pi;  
>> y=cos(x);  
>> z=sin(x);  
>> plot3(x,y,z);  
>> grid on
```



武汉大学

Wuhan University





三维网线图和曲面图

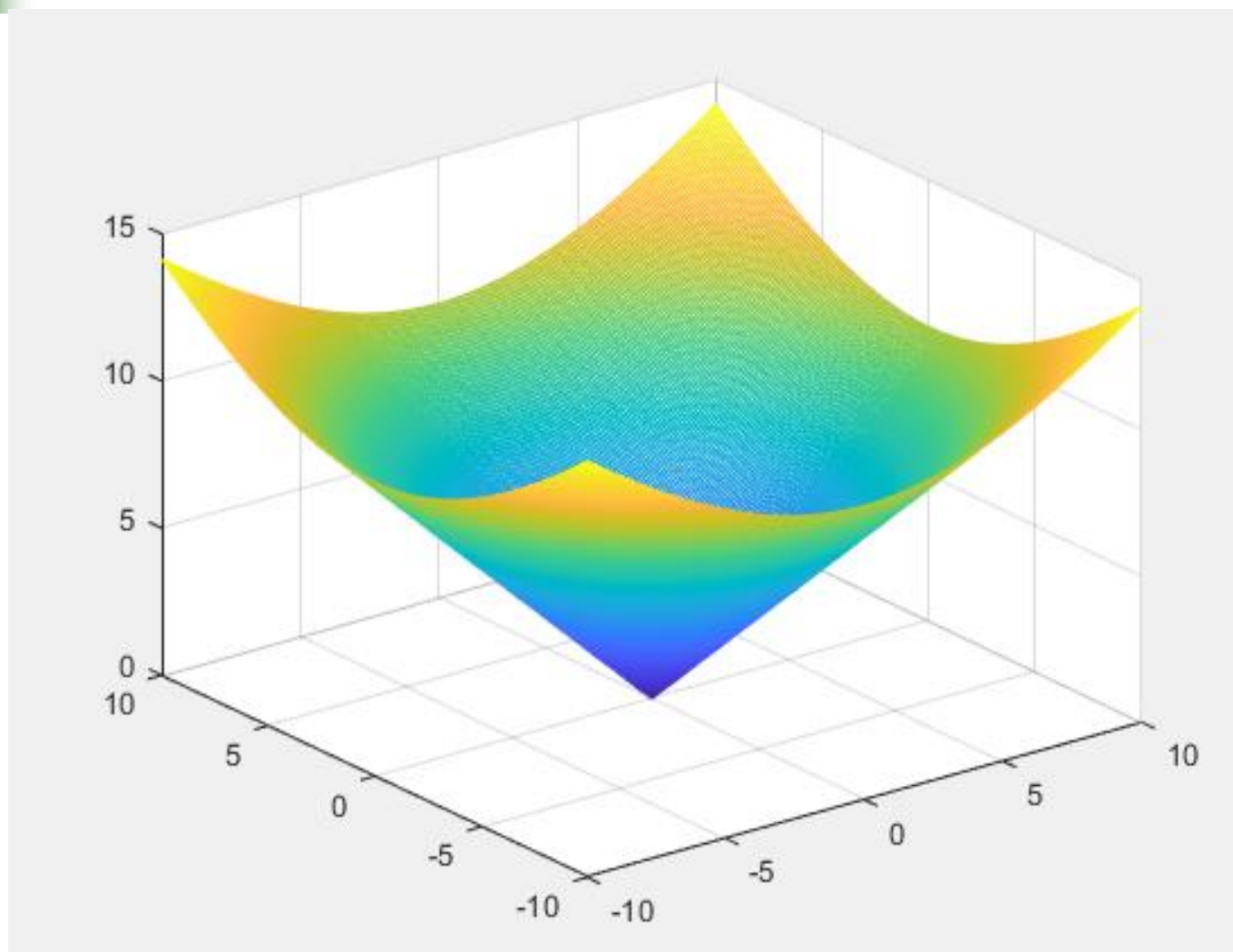
除数据外，还要对三维输出图形进行色彩、明暗、光照和视点等的处理，以使所输出的三维图形更加理想。网线图和曲面图的基本指令语法结构如下：

- `meshgrid (x,y)` %生成x-y坐标上的“格点” 矩阵
 - `mesh (x,y,z)` %作出网线图
 - `surf (x,y,z)` %作出曲面图
- ```
>> x=-10:0.1:10;
>> y=x;
>> [x,y]=meshgrid(x,y); %作出“格点” 矩阵
>> z1=sqrt(x.^2+y.^2);
>> mesh(x,y,z1)
```



武汉大学

Wuhan University

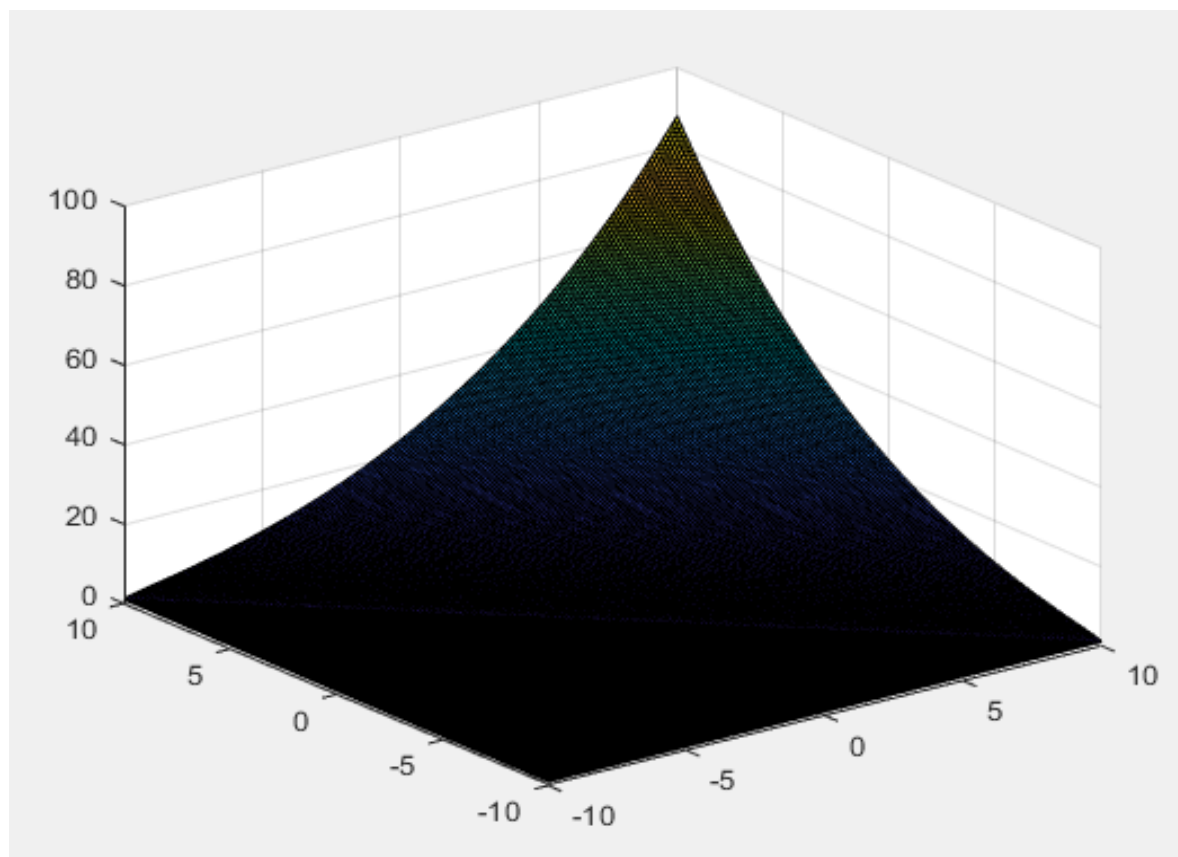




武汉大学

Wuhan University

```
>> z2=abs(exp(sqrt(x+y)));
>> surf(x,y,z2)
```





## §2.6 MATLAB代码设计入门

### 2.6.1 MATLAB的程序类型

MATLAB的程序类型有两种，一种是M脚本文件；另外一种M函数（function）文件。

#### 1、脚本M文件

- 以.m格式进行存取，包含一连串的MATLAB指令和必要的注解。需要在工作空间中创建并获取变量，也就是说处理的数据为命令窗口中的数据，没有输入参数，也不会返回参数。
- 程序运行时只需在工作空间中键入其名称即可。



## 2、函数M文件

与在命令窗口中输入命令一样，函数接受输入参数，然后执行并输出结果。用help命令可以显示它的注释说明。具有标准的基本结构。

### (1) 函数定义行（关键字function）

```
function[out1,out2,...]=filename(in1,in2,...)
```

输入和输出（返回）的参数个数分别由nargin和nargout两个MATLAB保留的变量来给出。

### (2) 第一行帮助行，即H1行

以（%）开头，作为lookfor指令搜索的行

### (3) 函数体说明及有关注解

以（%）开头，用以说明函数的作用及有关内容

如果不希望显示某段信息，可在它的前面加空行



武汉大学

Wuhan University

#### (4) 函数体语句

函数体内使用的除返回和输入变量这些在function语句中直接引用的变量以外的所有变量都是局部变量，即在该函数返回之后，这些变量会自动在MATLAB的工作空间中清除掉。如果希望这些中间变量成为在整个程序中都起作用的变量，则可以将它们设置为全局变量。



武汉大学

Wuhan University

## 脚本M文件和函数M文件的区别

- 形式上：M函数文件第一行以包含“function”为引导的“函数声明行”，M脚本文件则没有。
- 内涵上：M脚本文件是MATLAB指令的集合，无输入变量，无返回值，不可直接调用；而M函数文件有返回值，可相互调用。



武汉大学

Wuhan University

%ex2\_1.m M脚本文件

clear;

V=40;R=500;Ra=25000;Rb=100000;Rc=125000;Rd=40000;Re=37500;

R1=(Rb\*Rc)/(Ra+Rb+Rc);

R2=(Rc\*Ra)/(Ra+Rb+Rc);

R3=(Ra\*Rb)/(Ra+Rb+Rc);

Req=R+R1+1/(1/(R2+Re)+1/(R3+Rd));

I=V/Req; W=Req\*I^2 %%此处无 ; 才能显示W输出运行结果

>> path(path, D:\matlab\work') %设置文件路径

>> ex2\_1

>> x=2\*W

>> y=2\*ex2\_1

尝试将 SCRIPT ex2\_1 作为函数执行:

E:\MATLAB电子技术与应用\2019\2019\adjustment ppt\ex2\_1.m  
%无返回值, 不可直接调用

>> y=I %因为在工作空间内, 可调用其中变量





武汉大学

Wuhan University

## %ex2\_2.m M函数文件

```
function [I,W]=ex2_2(V) %V为输入变量 , i,W为输出变量
R=500;Ra=25000;Rb=100000;Rc=125000;Rd=40000;Re=37500;
R1=(Rb*Rc)/(Ra+Rb+Rc);
R2=(Rc*Ra)/(Ra+Rb+Rc);
R3=(Ra*Rb)/(Ra+Rb+Rc);
Req=R+R1+1/(1/(R2+Re)+1/(R3+Rd));
I=V/Req;
W=Req*I^2;
%运行结果
>> [x,y]=ex2_2(40) %可直接调用
x =
 5.2980e-004
y =
 0.0212
```



武汉大学

Wuhan University

## 2.6.2 MATLAB程序的基本设计原则

### 1、MATLAB程序的基本组成结构

%说明

清除命令：清除workspace中的变量和图形（clear,close）

定义变量：包括全局变量的声明及参数值的设定

逐行执行命令：

... ..

控制循环：包含for,if then,switch,while等语句

逐行执行命令

... ..

end

绘图命令：将运算结果绘制出来



## 2、MATLAB程序编写基本注意事项

- %后面的内容是程序的注解，要善于运用注解使程序更具可读性。
- 养成在主程序开头用clear指令清除变量的习惯，以消除工作空间中其他变量对程序运行的影响。但注意在子程序中不要用clear。
- 参数值要集中放在程序的开始部分，以便维护。要充分利用MATLAB工具箱提供的指令来执行所要进行的运算，在语句行之后输入分号使其及中间结果不在屏幕上显示，以提高执行速度。
- input指令可以用来输入一些临时的数据；而对于大量参数，则通过建立一个存储参数的子程序，在主程序中用子程序的名称来调用。



武汉大学

Wuhan University

- 程序尽量模块化，也就是采用主程序调用子程序的方法，将所有子程序合并在一起执行全部的操作。
- 充分利用Debugger来进行程序的调试（设置断点、单步执行、连续执行），并利用其他工具箱或图形用户界面（GUI）的设计技巧，将设计结果集成到一起。
- 设置好MATLAB的工作路径，以便程序运行。



武汉大学

Wuhan University

### 3、局部变量与全局变量

- 局部变量

默认的变量属性，只能在当前MATLAB文件中有效，与其他M文件中的局部变量和基本工作空间中的变量无关。

- 全局变量

采用global可将变量声明成全局变量。声明成全局变量后，可与工作空间和其他M文件共享同一值。

- 永久变量

采用persistent可将变量声明成永久变量。只用于M文件中，在每次调用时保持值不变。



武汉大学

Wuhan University

```
>> clear
>> global y %if without it, warning
>> y=3
y =
 3
>> ex2_3
z =
 8
```

Ex2\_3.m

```
x=2;
global y
z=x^y
```



## 2.6.3关系与逻辑运算符

- MATLAB的运算符有三种类型：算术运算符、关系运算符、逻辑运算符。它们的处理顺序依次为算术运算符、关系运算符、逻辑运算符。这里我们着重介绍后两种运算符。

- **关系运算符**

假设有： $A=[1\ 2\ -1\ -5]; B=[0\ 2\ 3\ 1];$

< 小于  $A < B$   $\text{ans}=[0\ 0\ 1\ 1]$   $A < 1$   $\text{ans}=[0\ 0\ 1\ 1]$

> 大于  $A > B$   $\text{ans}=[1\ 0\ 0\ 0]$   $A > 1$   $\text{ans}=[0\ 1\ 0\ 0]$

<= 小于等于  $A \leq B$   $\text{ans}=[0\ 1\ 1\ 1]$

>= 大于等于  $A \geq B$   $\text{ans}=[1\ 1\ 0\ 0]$

== 等于  $A == B$   $\text{ans}=[0\ 1\ 0\ 0]$  ;  $A = 1$   $\text{ans}=[1\ 0\ 0\ 0]$

~= 不等于  $A \sim B$   $\text{ans}=[1\ 0\ 1\ 1]$  ;  $A \sim 1$   $\text{ans}=[0\ 1\ 1\ 1]$



武汉大学

Wuhan University

## ■ 逻辑运算符

在处理逻辑运算时，运算元只有两个值即0和1，所以如果指定的数为0，MATLAB认为其为0，而任何数不等于0，则认为1。

设有： $A=[5 \ -4 \ 0 \ -0.5]; B=[0 \ 1 \ 0 \ 9]$

& 与  $A \& B=[0 \ 1 \ 0 \ 1] \quad A \& 1=[1 \ 1 \ 0 \ 1]$

| 或  $A|B=[1 \ 1 \ 0 \ 1] \quad A|1=[1 \ 1 \ 1 \ 1]$

~ 非  $\sim A=[0 \ 0 \ 1 \ 0] \quad \sim 1=0$





## 2.6.4 MATLAB控制流语句

### ■ for循环语句

基本格式

for 循环变量 = 起始值 : 步长 : 终止值

循环体

end

```
x=0;
```

```
for i=1:5
```

```
 x=i+x;
```

```
end;
```

步长缺省值为1，可以在正实数或负实数范围内任意指定。对于正数，循环变量的值大于终止值时，循环结束；对于负数，循环变量的值小于终止值时，循环结束。循环结构可以嵌套使用。



## ■ while循环语句

while 表达式  
    循环体  
End

```
x(1)=0;
x(2)=1;
i=2;
while x(i)<888
 x(i+1)=x(i-1)+x(i);
 i=i+1;
end
```

若表达式为真，执行循环体内容，执行后再判断表达式是否为真，若不为真，则跳出循环体，向下继续执行。

While循环和for循环的区别在于，while循环结构的循环体被执行的次数不是确定的，而for结构中循环体的执行次数是确定的。



## ■ if , else , elseif语句

( 1 ) if      逻辑表达式  
                 执行语句

end

( 2 ) if   逻辑表达式  
            执行语句1

else

    执行语句2

end

( 3 ) if   逻辑表达式1  
            执行语句1

elseif   逻辑表达式2  
          执行语句2

end



武汉大学

Wuhan University

if-else的执行方式为：如果逻辑表达式的值为真，则执行语句1，然后跳过语句2，向下执行；如果为假，则执行语句2，然后向下执行。

if-elseif的执行方式为：如果逻辑表达式1的值为真，则执行语句1；如果为假，则判断逻辑表达式2，如果为真，则执行语句2，否则向下执行。



## switch语句

switch 表达式 ( %可以是标量或字符串 )

case 值1  
语句1

case 值2  
语句2

....  
otherwise  
语句3

end

执行方式：表达式的值和哪种情况(case)的值相同，就执行哪种情况中的语句，如果不同，则执行otherwise中的语句。格式中也可以不包括otherwise，这时如果表达式的值与列出的各种情况都不相同，则继续向下执行。



武汉大学

Wuhan University

## ■ try-catch语句

try

语句1

catch

语句2

end

try-catch语句中的语句1总是先被执行，如果语句1正确，则跳出try-catch结构；否则执行语句2。因此，只有在语句1错误的情况下语句2才能被执行。



武汉大学

Wuhan University

## 其他控制流语句

- keyboard      键盘输入MATLAB指令
- break      中断
- continue      跳过循环中其后的命令
- pause      暂停，按任意键后继续
- error( 'message' )      显示出错信息并退出程序
- lasterr      显示出错信息并退出程序
- lastwarn      显示最新警告，程序继续运行
- warning( 'message' )      显示警告信息，程序继续运行



## 2.6.5 数据&文件的输入/输出语句

### 交互输入/输出命令

■ `u=input('提示内容\n', 's')`

在屏幕上显示提示内容，等待键盘输入，并将输入值赋给变量u，参数s表示‘提示内容’以字符串的形式幅值，\n表示换行，\\表示反斜线。

```
>> x=input('Please input the value of x')
```

```
Please input the value of x [1 0 0;0 1 0;0 0 1]
```

```
x =
```

```
1 0 0
0 1 0
0 0 1
```

注：若未输入对应值而直接回车，则返回空矩阵[]





武汉大学

Wuhan University

- `k=menu( 'title' , ' option1' , ' option2' ,....., ' optionn' )`

菜单输入命令，产生一个供用户输入的选择菜单。显示以title为标题的菜单，选项为option1 ~ optionn，并将所选选项编号赋给k。

```
>> k=menu('student','San Zhang','Si Li','Wu Wang')
```

```
k =
```

```
2
```



武汉大学

Wuhan University

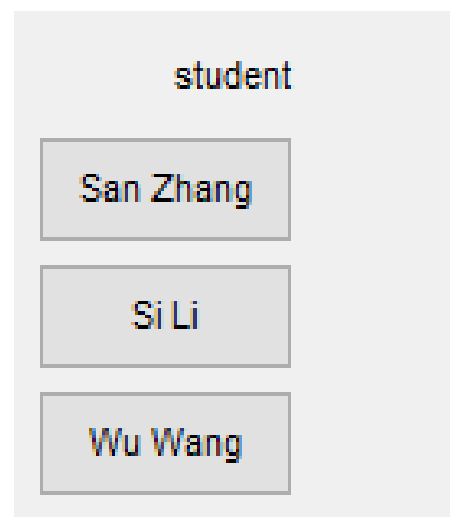
- `k=menu( 'title' , ' option1' , ' option2' ,....., ' optionn' )`

菜单输入命令，产生一个供用户输入的选择菜单。显示以title为标题的菜单，选项为option1 ~ optionn，并将所选选项编号赋给k。

```
>> k=menu('student','San Zhang','Si Li','Wu Wang')
```

```
k =
```

```
2
```





武汉大学

Wuhan University

- sprintf(format,val)

按指定格式输出变量，类似C中printf。

```
>> sprintf('%d' ,round(pi)) %% round - 四舍五入
```

```
ans =
```

```
3
```

```
>> x=sprintf('%s','WHU')
```

```
x =
```

```
WHU
```

```
>> sprintf('%1.3f',pi)
```

```
ans =
```

```
3.142
```

```
>> sprintf('%1.3g',pi)
```

```
ans =
```

```
3.14
```



## 文件输入/输出命令与函数

- save 文件名 变量名 选项

将变量存入指定文件中。

选项：-ascii 以8位ASCII格式保存数据

-ascii -double 以16位ASCII格式保存数据

-ascii -tabs 以8位ASCII格式保存，用tab作分隔符

-ascii -double -tabs .....

- load 文件名

从指定文件中调入变量。



武汉大学

Wuhan University

- `f_id=fopen(文件名, '允许模式', 格式)`  
以'允许模式'打开文件。

允许模式：

'r' 打开文件进行读操作（默认形式）

'w' 删除已存在文件中的内容或生成一个新文件，打开进行写操作

.....

格式：

n – native    l – ieee-le    b – ieee-be

.....



武汉大学

Wuhan University

- status=fclose(file/all)

关闭某个或所有文件，关闭成功返回0给status，关闭失败返回-1给status。

其他输入/输出函数

|         |         |          |
|---------|---------|----------|
| fread   | fwrite  | fscanf   |
| fprintf | fgets   | fgetl    |
| ferror  | feof    | imread   |
| imwrite | imfinfo | auread   |
| auwrite | wavread | wavwrite |



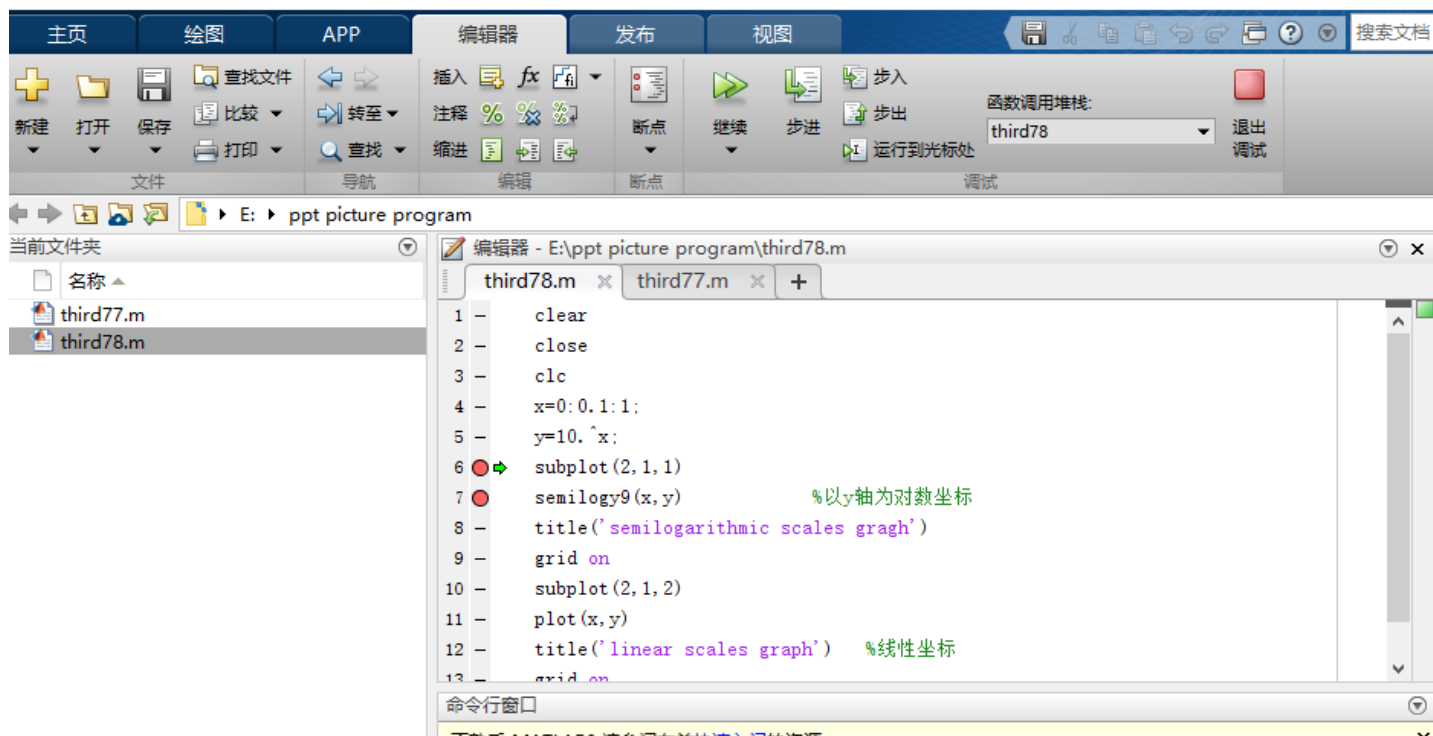
武汉大学

Wuhan University

## 2.6.6 M文件的调试

- M文件编辑器Editor的菜单中的Debug选项为M文件的调试菜单。可点击行首的小横线“-”使之成为红色圆点“•” 或将光标放置在行中任意位置后按〔F12〕设置断点。

MATLAB R2019a - academic use





武汉大学

Wuhan University

- ex2\_4.m 投资组合



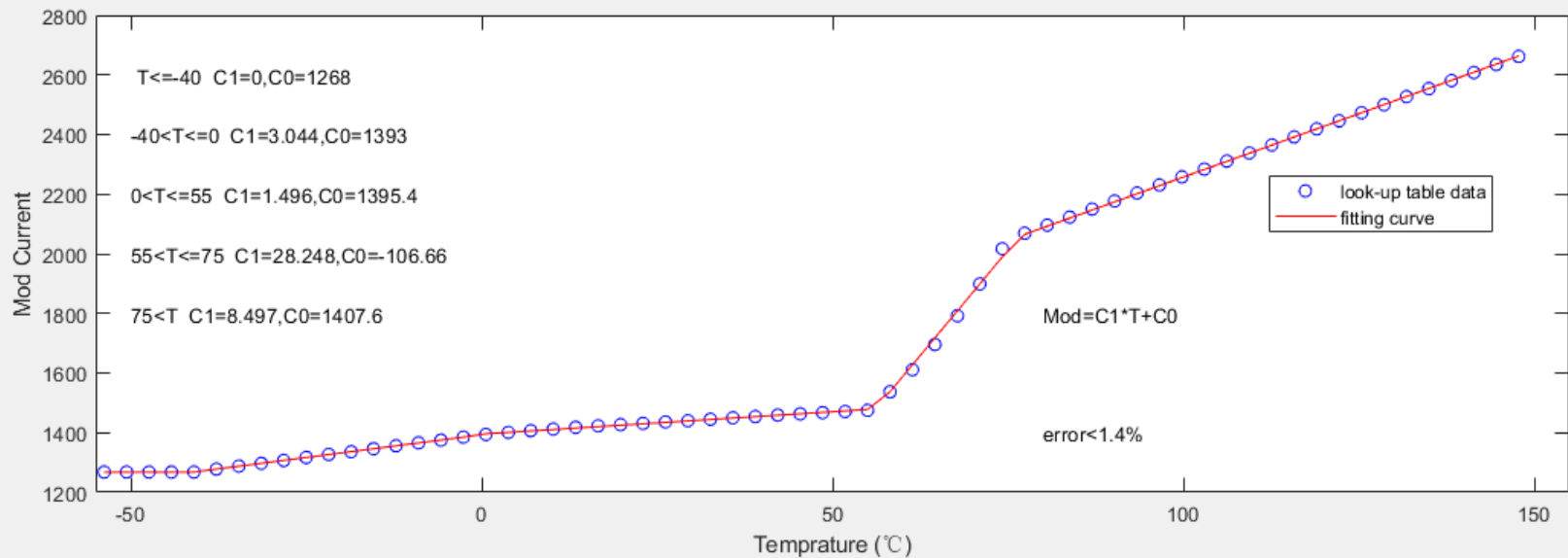
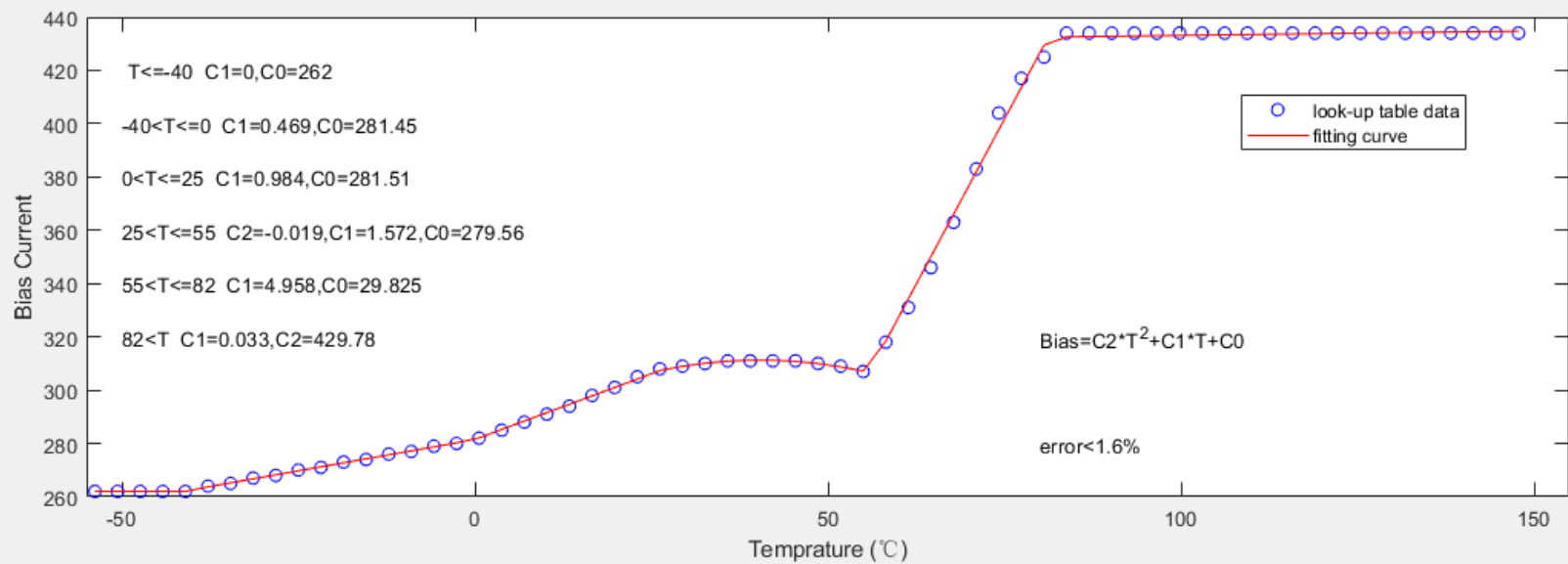


武汉大学

Wuhan University

- ex2\_5.m 激光发射管的驱动偏置电流、调制电流的公式拟合。

采用最小二乘法的方式对偏置电流Bias和调制电流Mod进行分段线性拟合。





武汉大学

Wuhan University

## §2.6 MATLAB接口简介

- 为更好的利用外部已有丰富的C语言和Fortran语言编写的程序资源，也为使MATLAB提供的丰富的函数资源能够被调用，MATLAB提供了对外接口技术。

### 2.6.1 MATLAB接口技术简介

- 参考：

刘志俭，《MATLAB应用程序用户接口指南》，科学出版社

江泽林，刘维，《实战MATLAB之文件与数据接口技术》，北京航空航天大学出版社



## MEX文件--外部程序调用接口

- MEX文件是一种“可在MATLAB中调用的C（ or Fortran ）语言衍生程序”。其源码文件是由C or Fortran语言编写的，经MATLAB编译器处理而生成的动态链接程序.dll（二进制文件）。其调用方式于MATLAB内建函数完全一样，在命令提示符下键入文件名即可。
- 调用C语言或Fortran语言编写的程序。
- 在MATLAB中运行不很有效的计算瓶颈（一般出现在包含不可避免的循环时），可用C或Fortran重新编写后构成MEX文件解决。
- 直接面向硬件编写的C或Fortran程序可以通过MEX文件被MATLAB调用。



武汉大学

Wuhan University

- **MAT文件应用程序**—数据输入输出接口
- MAT文件是MATLAB数据存储的默认文件格式，以.mat为后缀。MATLAB提供了带mat前缀的API函数库，使我们能容易的对MAT文件进行操作。通过编写能够调用这些函数库的C或Fortran应用程序，就能够实现MATLAB与应用程序的数据交换和共享。



## ■ MATLAB引擎

- MATLAB引擎函数库是MATLAB提供的一系列程序的集合，它允许用于在C或Fortran语言应用程序总对MATLAB进行调用，将MATLAB作为一个计算引擎，让其在后台运行，完成复杂的矩阵计算，从而讲话前台用于程序设计的任务。
- MATLAB在其他语言编写的应用程序中被当作数学库程序调用，充分利用MATLAB指令简单、计算可靠的优点。
- MATLAB在专用系统总被当作计算引擎使用时，前台是C等语言编写的GUI图形用户接口，后台由MATLAB执行各种复杂的计算分析，以缩短用户的开发时间。



武汉大学

Wuhan University

## 2.6.2 MATLAB与EDA工具接口

### 现状

- 传统的方式，MATLAB在电路设计中的作用只限于完成算法级设计或DSP的设计，与真正意义上的IC设计相去甚远，但随着IC产业的迅速崛起，MATLAB也逐步增加了与常用EDA工具软件的接口。
- Matlab7.0的一个亮点就是在toolbox中新增加了一个link for Modelsim的工具箱。借助这个工具箱，可以实现MATLAB和Simulink与ModelSim HDL模拟器的链接，因而允许设计人员在MATLAB中制作测试平台，然后把HDL模型返回Simulink。



- 三家EDA供应商开发了与MATLAB和Simulink的新链接。新兴的Catalytic公司推出了一款工具，可以方便地将浮点MATLAB模型转化成定点模型；CoWare公司增强了它的信号处理工作系统(SPW)与MATLAB之间的链接；AccelChip公司将MATLAB模型合成为RTL Verilog，并扩展了它的IP库。此外，Cadence公司也发布了其PSpice模拟仿真器与MATLAB的介面。
- DSP Builder：Altera DSP Builder将The MathWorks MATLAB和Simulink系统级设计工具算法开发、仿真和验证功能与VHDL综合、仿真和Altera开发工具整合在一起，实现了算法设计到电路设计的集成。





武汉大学

Wuhan University

## MATLAB与Modelsim

- Modelsim是目前与MATLAB结合得最好也是联合使用得最多的EDA工具，下面将简介MATLAB与Modelsim联合使用流程。
- Toolbox中的Link for ModelSim工具箱提供了一个使用MATLAB环境来验证Modelsim VHDL模型的接口。通过编写VHDL代码和相关的MATLAB函数来共享数据就可以使用这个接口。总体流程如下：



武汉大学

Wuhan University

- 1、编写可供MATLAB使用的VHDL代码。结构体命名要符合MATLAB函数命名的范；定义的VHDL数据类型要符合Link for ModelSim要求；
  - 2、编译和调试此VHDL代码
  - 3、编写MATLAB函数测试基准，注意Modelsim与MATLAB间数据格式的转换。
  - 4、将此MATLAB函数放置到MATLAB寻找路径中
- 具体流程和实验方法见

<http://www.mathworks.com/access/helpdesk/help/toolbox/modelsim/index.html>



武汉大学

Wuhan University

- 习题一：编写M文件，计算以下问题：

$$\square\square\square\square \times \square = \square\square\square\square$$

以上9个□代表1~9这9个数字，不得遗漏或重复

- 要求：给出解算思路和M文件代码
- 注意算法的效率



## ■ 习题二：丁字路口红绿灯问题建模

丁字路口，每条车道同时只允许一辆车通过，如图所示每分钟内各车道车辆数为

| 车道  | 1  | 2  | 3  | 4  |
|-----|----|----|----|----|
| 车辆数 | 20 | 12 | 18 | 10 |

- 要求：需对哪几条车道建红绿灯（无黄灯）？  
建立MATLAB模型，求解最优化红绿灯的时间应该多长。
- 提高：若为十字路口，按现行交通规则提出交通灯设置方案。



武汉大学

Wuhan University

