

国家精品课程/ 国家精品资源共享课程/ 国家级精品教材

国家级十一(二)五规划教材/ 教育部自动化专业教学指导委员会牵头规划系列教材

控制系统计算机辅助设计

第2章: MATLAB语言程序设计基础

主讲: 修贤超

MATLAB语言程序设计基础

- 2.1 程序设计语言基础
- 2.2 基本数学运算
- 2.3 MATLAB语言的流程结构
- 2.4 函数的编写与调试
- 2.5 二维图形绘制
- 2.6 三维图形绘制



MATLAB程序设计语言基础

- MATLAB语言的变量与常量
- 数据结构
- MATLAB的基本语句结构
- 冒号表达式与子矩阵提取



MATLAB语言的变量与常量

- MATLAB语言的变量名规则
 - 由一个字母引导，后面可以为其他字符
 - 区分大小写 $Abc \neq ABc$
 - 有效 `MYvar12`, `MY_Var12` 和 `MyVar12_`
 - 错误的变量名 `12MyVar`, `_MyVar12`, `MyVar-`
- MATLAB的保留常量
 - `eps`, `i`, `j`, `pi`, `NaN`, `Inf`, `i=sqrt(-1)`
 - `lastwarn`, `lasterr`



MATLAB语言的数据结构

- 数值型数据结构
- 符号型数据结构
- 其他数据结构



数值型数据

- 双精度数值变量
 - IEEE标准, 64位 (8字节)
 - 11指数位, 52数值位和一个符号位
 - 数据范围 $-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$
 - double() 函数的转换
- 其他数据类型
 - 单精度数据结构 single(), 32位
 - uint8(), 常用于图像表示和处理, 8位
 - int8(),int16(),int32(),uint16(),uint32()



符号型数据结构

➤ 符号型，`sym(A)`，常用于公式推导和求解析解

➤ 变量声明 `syms var_list var_props`

➤ 显示符号变量的任何精度

`vpa(A)`, `vpa(A, n)`

默认精度——32位十进制有效数字

➤ 显示符号变量的属性 `assumptions()`

➤ 设置符号变量类型 `assume()`, `assumeAlso()`

```
>> syms x real; assume(x>=-1); assumeAlso(x<5);
```



例2-1 符号型数值与双精度数值的区别

- 例：1/3的双精度存储
 - 双精度存储——0.333.....33，15位有限数字
 - 其余的数都被截断了，没有规律
- 1/3的符号变量存储
 - 一直存储1/3
- 二者的转换

`sym(1/3), double(1/3)`



例2-2 圆周率 π

- 求出 π 的 300 位有效数字
- MATLAB代码

```
>> vpa(pi,300)
```

- 可以“任意”增加有效数字的位数
- 随着有效数字位数的增加，程序运行的时间变长，如果位数要求过多可能无法显示



例2-4 试用符号型数据结构表示数值 12345678901234567890

➤ 直接输入方法

```
>> A=sym(12345678901234567890)
```

➤ 正确的输入方法

```
>> B=sym('12345678901234567890123456789012345678901234567890')
```



其他数据结构

- 字符串型数据：用单引号括起来
- 多维数组：是矩阵的直接扩展，多个下标
- 单元数组：将不同类型数据集成到一个变量名下面，用 {} 表示
- 表格数据结构：可以定义带表头的表格
- 类与对象：可以定义重载函数，类与对象是“面向对象”编程的基本元素 —— 传递函数对象



MATLAB的基本语句结构

➤ 直接赋值语句

➤ 接赋值语句的基本结构

```
variable = expression
```

➤ 赋值语句的末尾加一个分号可以阻止显示运算结果

➤ 保留变量：`ans`

➤ 存放最近一次无赋值变量语句的运算结果



例2-5 矩阵输入方法

➤ 表示矩阵 $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$

➤ MATLAB代码

```
>> A=[1,2,3; 4 5,6; 7,8 0]
```

➤ 其他命令（动态定维）

```
>> A=[1,2,3; 4 5,6; 7,8 0];  
A=[[A; [1 2 3]], [1;2;3;4]]
```



例2-6 复数矩阵输入

➤ 试输入复数矩阵

$$B = \begin{bmatrix} 1+j9 & 2+j8 & 3+j7 \\ 4+j6 & 5+j5 & 6+j4 \\ 7+j3 & 8+j2 & 0+j1 \end{bmatrix}$$

➤ MATLAB代码

```
>> B=[1+9i,2+8i,3+7i;  
      4+6i 5+5i,6+4i; 7+3i,8+2i 1i]
```

➤ 需要避免的语句

```
>> B=[1 +9i,2+8i,3+7j;  
      4+6j 5+5i,6+4i; 7+3i,8+2j 1i]
```



函数调用的语句格式

➤ 函数调用语句

$[returned_arguments] =$
 $function_name(input_arguments)$

➤ 函数调用举例 $[U \ S \ V] = svd(X)$

➤ 函数可以通过不同的方式被调用

➤ 内核函数, *.m函数

➤ 匿名函数、inline函数 (不建议使用)

➤ 重载函数、私有函数等



冒号表达式与子矩阵提取

- 冒号表达式是定义行向量的有效方法

$$v = s_1 : s_2 : s_3$$

- 开始于值 s_1 ，步长为值 s_2 ，终止于值 s_3
 - 默认步长值为1



例2-7 冒号表达式生成

➤ 用不同的步距生成 $t \in [0, \pi]$ 间的向量

```
>> v1=0: 0.2: pi
```

```
>> v1a=linspace(0,pi,50)
```

```
>> v2=0: -0.1: pi
```

```
>> v3=0:pi
```

```
>> v4=pi:-1:0
```



子矩阵提取

➤ 子矩阵提取的基本语句格式


$$B=A(v_1, v_2)$$

➤ v_1 表示子矩阵要保留的行号构成的向量

➤ v_2 表示要保留的列号构成的向量

➤ $:$ ，表示要提取所有行或列，取决于其位置

➤ end的使用

➤ 举例： `>> B=A(1:2:end,:), C=A([1 1 1 1],:)`



例2-8 子矩阵提取

➤ 矩阵A的不同子矩阵提取

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$$

➤ MATLAB代码

```
>> A=[1,2,3; 4,5,6; 7,8,0];  
B1=A(1:2:end, :)  
B2=A([3,2,1], [2 3])  
B3=A(:,end:-1:1)
```



MATLAB语言程序设计基础

- 2.1 程序设计语言基础
- 2.2 基本数学运算
- 2.3 MATLAB语言的流程结构
- 2.4 函数的编写与调试
- 2.5 二维图形绘制
- 2.6 三维图形绘制



矩阵的基本数学运算

- 矩阵的代数运算
- 矩阵的逻辑运算
- 矩阵的比较运算
- 解析结果的化简与变换
- 基本数论运算



矩阵转置

➤ 矩阵表示

➤ 矩阵 A , n 行 m 列, 被称作 $n \times m$ 矩阵

➤ Hermite转置 $C = A'$

$$C = A^H, c_{ji} = a_{ij}^*, i = 1, \dots, n, j = 1, \dots, m$$

➤ 一般转置 $C = A.'$

$$B = A^T, b_{ji} = a_{ij}, i = 1, \dots, n, j = 1, \dots, m$$



加减法运算

➤ 数学表示

$$c_{ij} = a_{ij} \pm b_{ij}, \quad i = 1, \dots, n, j = 1, \dots, m$$

➤ 用C语言编程难于实现，同 $A*B$

➤ MATLAB语法

$$C = A + B \quad C = A - B$$

➤ 注意：任一个变量可以为标量

➤ 如果矩阵维数不匹配，系统会报错



矩阵乘法

➤ 数学表示: $C = AB$

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}, \quad i = 1, \dots, n, j = 1, \dots, m$$

➤ MATLAB表示:

$$C = A * B$$

➤ 注意: 系统自动检测矩阵维数是否匹配

➤ 可直接用于标量、复数等



矩阵的除法

➤ 矩阵左除：

➤ 求解线性方程组： $AX = B$

➤ MATLAB解法： $X = A \setminus B$

➤ 最小二乘解；若 A 为非奇异方阵，则 $X = A^{-1}B$

➤ 使得误差最小 $\min_X \|AX - B\|_2$

➤ 矩阵右除

➤ 求解线性方程组 $XA = B$

➤ MATLAB解法 $X = B/A$

➤ 等效的运算 $B/A = (A' \setminus B')'$



矩阵翻转

➤ 左右翻转

$$b_{ij} = a_{i,n+1-j}, \quad \mathbf{B} = \text{fliplr}(\mathbf{A})$$

➤ 上下翻转

$$c_{ij} = a_{m+1-i,j}, \quad \mathbf{B} = \text{flipud}(\mathbf{A})$$

➤ 旋转 90°

$$d_{ij} = a_{j,n+1-i}, \quad \mathbf{D} = \text{rot90}(\mathbf{A})$$

➤ 如何旋转 180° ?

$$\mathbf{D} = \text{rot90}(\mathbf{A}, k)$$



矩阵乘方

➤ 求矩阵 A 的 x 次幂 $F = A^x$

➤ 数学描述

➤ MATLAB命令 $F = A^x$

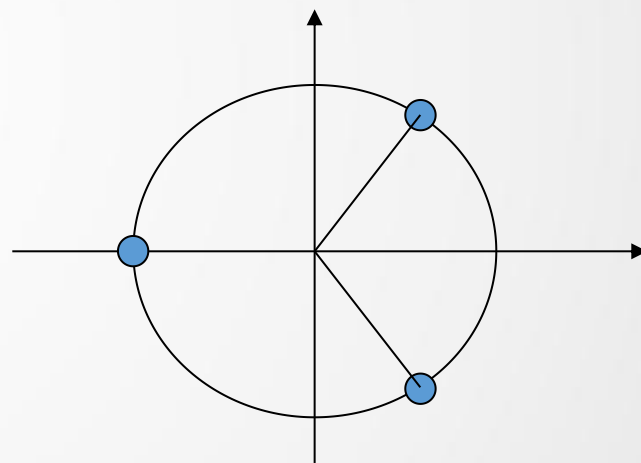
➤ A 必须为方阵

➤ x 为整数

➤ x 为非整数

➤ $\sqrt[k]{-1}$ 开方的多解：旋转

➤ 得出一个解，乘以 $k-1$ 次标量 $\gamma = e^{2\pi j/k}$





例2-9 矩阵的三次方根

- 求矩阵 A 的全部三次方根，并检验结果

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$$

- MATLAB代码

```
>> A=[1,2,3; 4,5,6; 7,8,0];  
C=A^(1/3), e=norm(A-C^3)
```

- 另两个根

```
>> j1=exp(sqrt(-1)*2*pi/3);  
A1=C*j1, A2=C*j1^2,  
norm(A-A1^3), norm(A-A2^3)
```



点运算

➤ 矩阵对应元素的直接运算 $C=A.*B$, $c_{ij} = a_{ij}b_{ij}$

➤ 例如: $B=A.^A$, $b_{ij} = a_{ij}^{a_{ij}}$

➤ 另一个例子:

$$\begin{bmatrix} 1^1 & 2^2 & 3^3 \\ 4^4 & 5^5 & 6^6 \\ 7^7 & 8^8 & 0^0 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 27 \\ 256 & 3125 & 46656 \\ 823543 & 16777216 & 1 \end{bmatrix}$$

```
>> A=[1,2,3; 4 5,6; 7,8 0]; B=A.^A
```

➤ 应用举例: 绘图 $y = f(x) = x^2$, $y_i = x_i^2$ $y=x.^2$



矩阵的逻辑运算

- 逻辑变量：
 - 当前版本有逻辑变量
 - 非 0 表示逻辑 1
- 逻辑运算（相应元素间的运算）
 - 与运算 $A \& B$
 - 或运算 $A | B$
 - 非运算 $B = \sim A$
 - 异或运算 $\text{xor}(A, B)$



矩阵的比较运算

➤ 各种允许的比较关系

➤ $>$, \geq , $<$, \leq , $==$, \sim , $\text{find}()$, $\text{all}()$, $\text{any}()$

➤ 例2-10 比较运算实例

```
>> A=[1,2,3; 4 5,6; 7,8 0];  
      i=find(A>=5)'
```

```
>> [i,j]=find(A>=5)
```

```
>> a1=all(A>=5), a2=any(A>=5)
```



解析结果的化简与变换

- 函数`simplify()` 用于数学公式的化简

$$s_1 = \text{simplify}(s)$$

- 其他常用化简函数

- `numden()`, `collect()`, `expand()`, `factor()`



例2-11 多项式处理

➤ 化简多项式

$$P(s) = (s + 3)^2(s^2 + 3s + 2)(s^3 + 12s^2 + 48s + 64)$$

➤ 用不同的函数求解

```
>> syms s;  
P=(s+3)^2*(s^2+3*s+2)*(s^3+12*s^2+48*s+64)
```

➤ 多项式的化简 `>> P1=simplify(P)`

➤ 多项式因式分解与展开

```
>> P3=factor(P), P4=prod(P3)
```



变量替换

➤ 变量替换语句

$$f_1 = \text{subs}(f, x_1, x_1^*)$$

$$f_1 = \text{subs}(f, \{x_1, x_2, \dots, x_n\}, \{x_1^*, x_2^*, \dots, x_n^*\})$$

➤ 该函数执行点运算

➤ 后者同时替换若干个变量

➤ 转换成LaTeX表示 $f_1 = \text{latex}(f)$

➤ 得出字符串，需要LaTeX环境解释



例2-12 双线性变换

- 试用 $s = (z - 1)/(z + 1)$ 对 $P(s)$ 进行双线性变换
- 多项式表达式

$$P(s) = (s + 3)^2(s^2 + 3s + 2)(s^3 + 12s^2 + 48s + 64)$$

- 变量替换的MATLAB代码

```
>> syms s z;  
P=(s+3)^2*(s^2+3*s+2)*(s^3+12*s^2+48*s+64);  
P1=simplify(subs(P,s,(z-1)/(z+1)))  
latex(P1)
```



LaTeX处理结果

➤ 结果: `8\,\{\frac { \left(2\,z+1 \right) ^{2}`
`z \left(3\,z+1 \right) \left(`
`5\,z+3 \right) ^{3}}`
`{ \left(z+1 \right) ^{7}}`

➤ LaTeX排版显示

$$8 \frac{(2z + 1)^2 z (3z + 1) (5z + 3)^3}{(z + 1)^7}$$

➤ LaTeX下载 建议: <http://www.ctex.org>



基本数据运算

➤ 常用计算函数

Function	Calling format
<code>floor()</code>	$n = \text{floor}(x)$
<code>ceil()</code>	$n = \text{ceil}(x)$
<code>round()</code>	$n = \text{round}(x)$
<code>fix()</code>	$n = \text{fix}(x)$
<code>rat()</code>	$[n, d] = \text{rat}(x)$
<code>rem()</code>	$B = \text{rem}(A, C)$
<code>gcd()</code>	$k = \text{gcd}(n, m)$
<code>lcm()</code>	$k = \text{lcm}(n, m)$
<code>factor()</code>	<code>factor(n)</code>
<code>isprime()</code>	$v_1 = \text{isprime}(v)$



例2-13 不同的取整函数

- 运用各种函数，对下面的数据进行取整运算

-0.2765, 0.5772, 1.4597, 2.1091, 1.191, -1.6187

- MATLAB代码：


```
>> A=[-0.2765,0.5772,1.4597,...  
      2.1091,1.191,-1.6187];  
v1=floor(A), v2=ceil(A),  
v3=round(A), v4=fix(A)
```



例2-14 Hilbert矩阵

- 假设 3×3 的 Hilbert 矩阵可以由 `hilb()` 定义，试对其进行有理数变换。

$$h_{i,j} = \frac{1}{i+j-1}$$

- 结果:  `>> A=hilb(3); [n,d]=rat(A)`

$$\mathbf{n} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}$$



例2-15 最大公约数与最小公倍数

- 给定两个整数1856120和1483720
 - 求其最大公约数与最小公倍数
 - 求出所得出的最小公倍数的质因数分解
- MATLAB求解语句:

```
>> m=sym(1856120); n=sym(1483720);  
    gcd(m,n), lcm(m,n),  
    factor(lcm(n,m))
```



例2-16 找出1000以内全部质数

➤ 试求出1 ~ 1000间的全部质数

```
>> A=1:1000; B=A(isprime(A))  
>> C=primes(1000)
```

➤ 得到的所有质数

2	29	67	107	157	199	257	311	367	421	467	541	599	647	709	769	829	887	967
3	31	71	109	163	211	263	313	373	431	479	547	601	653	719	773	839	907	971
5	37	73	113	167	223	269	317	379	433	487	557	607	659	727	787	853	911	977
7	41	79	127	173	227	271	331	383	439	491	563	613	661	733	797	857	919	983
11	43	83	131	179	229	277	337	389	443	499	569	617	673	739	809	859	929	991
13	47	89	137	181	233	281	347	397	449	503	571	619	677	743	811	863	937	997
17	53	97	139	191	239	283	349	401	457	509	577	631	683	751	821	877	941	
19	59	101	149	193	241	293	353	409	461	521	587	641	691	757	823	881	947	
23	61	103	151	197	251	307	359	419	463	523	593	643	701	761	827	883	953	



例2-16 全排列计算

- MATLAB的perms()函数可以计算全排列
- 假设5个人照集体照，共有多少种排列方式
 - 给出所有的排列可能 $B=A(v_1, v_2)$

```
>> P=perms(1:5), size(P)
```

- 如果5个人记作a,b,c,d,e，则

```
>> P=perms('abcde'), size(P)
```



数据结构小结

- 变量名命名规则
- 保留的常量，可以改写 `i`, `j`, `eps`, `NaN`, `Inf`
- 计算用数据结构、数据结构的区别
 - 双精度数据结构 `double`
 - 符号型数据结构 `syms`, `sym`, `vpa`
- 其他数据结构：字符串、结构体、类与对象



矩阵与向量输入小结

- 两种基本语句结构
 - 直接赋值语句
 - 返回变量如果不给出，则生成 ans
 - 分号的使用
 - 函数调用语句
- 矩阵与向量的输入方法、复数矩阵
- 冒号表达式 $v = s_1 : s_2 : s_3$
- 子矩阵提取 $B = A(v_1, v_2)$



矩阵代数运算小结

- 不同形式的代数运算
 - 两种不同形式的转置——直接转置与Hermit转置
 - 加减乘除（左除右除）、乘方开方
 - 多根开方的计算与检验
- 点运算



矩阵其他运算小结

- 矩阵的逻辑运算
 - 与 $\&$ ，或 $|$ ，非 \sim ，异或 xor
 - 逻辑运算是相应元素直接的运算，相当于点运算
- 比较运算
- 符号表达式化简与转换
- 数据计算：gcd, lcm, factor, 全排列



Q & A

感谢您的聆听和反馈

国家精品课程/ 国家精品资源共享课程/ 国家级精品教材

国家级十一(二)五规划教材/ 教育部自动化专业教学指导委员会牵头规划系列教材

控制系统计算机辅助设计

第2章: MATLAB语言程序设计基础

主讲: 修贤超

MATLAB语言程序设计基础

- 2.1 程序设计语言基础
- 2.2 基本数学运算
- 2.3 MATLAB语言的流程结构
- 2.4 函数的编写与调试
- 2.5 二维图形绘制
- 2.6 三维图形绘制

MATLAB语言的流程结构

- 利用流程结构，可以编写出复杂程序
- MATLAB支持的流程结构
 - 循环结构
 - 转移结构
 - 开关结构
 - 试探结构

MATLAB的循环结构

➤ for 循环结构

```
for i=v, loop programs body, end
```

➤ v 为一个向量，循环变量 i 每次从 v 向量中取一个数值，执行一次循环体的内容，如此下去，直至执行完 v 向量中所有的分量

➤ v 的内容可以任意排列

➤ C语言类似的结构

```
for (i=0; i<=n-1; i++) loop body
```

while循环结构

➤ while循环结构

```
while (condition)
    loop structure body,
end
```

- while结构的特点是可以设置出口
- for和while结构侧重点不同

例2-18 循环求解

➤ 用循环求解 $\sum_{i=1}^{100} i$

```
>> s1=0; for i=1:100, s1=s1+i; end,s1  
s2=0; i=1;  
while (i<=100), s2=s2+i; i=i+1; end  
s2
```

➤ 更简单的语句

```
>> sum(1:100)
```

例2-19 While循环

- 用循环求解求最小的 m ，使下式成立

$$\sum_{i=1}^m i > 10000$$

- 使用while结构

```
>> s=0; m=0;  
    while (s<=10000), m=m+1;  
        s=s+m; end, s, m
```

- 这里不宜应用for结构

例2-20 向量化编程

➤ 求和
$$S = \sum_{i=1}^{100000} \left(\frac{1}{2^i} + \frac{1}{3^i} \right)$$

➤ 使用MATLAB的循环结构和向量化运算方法

```
>> tic, s=0;
    for i=1:100000, s=s+1/2^i+1/3^i;
    end; toc
tic, i=1:100000;
s=sum(1./2.^i+1./3.^i); toc
```

条件转移结构

➤ 语句结构

```
if (condition 1)
    statement group 1
elseif (condition 2)
    statement group 2
    :
    :
else
    statement group n + 1
end
```

```
if (condition)
    statement group 1
end

if (condition)
    statement group 1
else
    statement group 2
end
```

例2-21 for循环

- 用for循环求解最大的 m ，使下式成立

$$\sum_{i=1}^m i > 10000$$

- MATLAB代码

```
>> s=0;  
for i=1:10000, s=s+i;  
    if s>10000, break;  
end, end
```

- 远比while结构繁琐

开关结构

➤ 开关结构的一般形式

```
switch switch expression
case expression 1, statements 1
case {expression 2, expression 3, ...,
      expression m}, statements 2
      :
otherwise, statements n
end
```

➤ 与C语言不同之处

试探结构

- 全新结构 `try,`
`statement group 1,`
`catch,`
`statement group 2,`
`end`
- 优点:
 - 设置错误陷阱
 - 提高算法的执行效率

MATLAB语言程序设计基础

- 2.1 程序设计语言基础
- 2.2 基本数学运算
- 2.3 MATLAB语言的流程结构
- 2.4 函数的编写与调试
- 2.5 二维图形绘制
- 2.6 三维图形绘制

函数编写与调试

- 编程：脚本程序与M-函数（主流）
- 为什么使用MATLAB函数？
- MATLAB函数的编写
 - MATLAB 语言函数的基本结构
 - 可变输入输出个数的处理
 - inline函数与匿名函数
 - 伪代码与保密处理

例2-22 M-脚本文件实现

➤ 问题

$$\sum_{i=1}^m i > 10000$$

➤ M-脚本以M-文件的形式被存取

➤ 为什么需要函数？

➤ 若最大值不为 10000，M-文件就需要被修改

➤ 则需要一种新的文件形式——M函数

➤ 输入变量：N

➤ 返回变量：s, m

MATLAB语言函数的基本结构

➤ 信息处理单元



➤ 函数的程序结构

`function [return vars] =funname(input vars)`

comments led by %

input and output variables check

main body of the function

例2-23 M-函数实现

➤ 可以编写一个M-函数

➤ M-文件如下

```
function [m,s]=findsum(k)
s=0; m=0;
while (s<=k), m=m+1; s=s+m; end
```

➤ 举例



```
>> [m1,s1]=findsum(145323)
```

➤ 优点：无需修改程序

例2-24 Hilbert长方形矩阵

- 编写一个函数生成 $n \times m$ Hilbert 矩阵

$$h_{i,j} = \frac{1}{i+j-1}$$

- 要求

- 输入变量: n, m , 输出变量 H
- 若只给出一个输入参数, 则自动生成方阵
- 在函数中给出合适的帮助信息
- 检测输入和返回变量的个数 nargin, nargout

例2-25 阶乘的递归调用

- 利用函数的递归调用

- 计算阶乘 $n! = n(n-1)!$

- 递归调用函数

- 很自然的递归调用格式

```
function k=my_fact(n)
k=n*my_fact(n-1);
```

- 存在的问题

- 出口的设置

递归调用的正常实现

➤ 设置出口，变量检测

```
function k=my_fact(n)
if nargin~=1,
    error('Error: Only one input variable accepted');
end
if abs(n-floor(n))>eps | n<0
    error('n should be a non-negative integer');
end
if n>1, k=n*my_fact(n-1);
elseif any([0 1]==n), k=1; end
```

阶乘的不同计算方法

➤ 计算11!

```
>> my_fact(11)
```

➤ 其他函数

```
factorial( $n$ )
```

```
prod(1: $n$ )
```

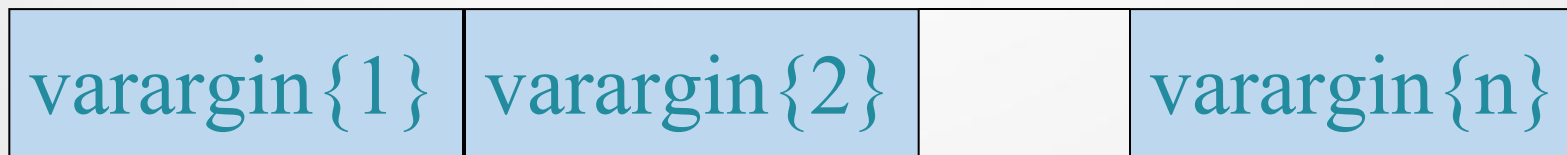
```
gamma(1+ $n$ )
```

```
gamma(1+sym( $n$ ))
```

➤ 符号运算 `factorial(sym(n))`

可变输入输出个数的处理

- 处理可变输入输出的感觉函数
 - 输入输出变量: `varargin`、`varargout`
- 变量的提取——单元数组 (`cell`)
 - `varargin{1}`, `varargin{2}`, ..., `varargin{n}`
- 存储与传递方法示意图



例2-27 任意多输入变元

➤ conv()可以计算两个多项式的积，试使用varargin实现任意多个多项式的积

➤ MATLAB编程

➤ 每次从varargin中提取一个做累乘

```
function a=convs(varargin), a=1;  
for i=1:nargin, a=conv(a,varargin{i}); end
```

```
>> P=[1 2 4 0 5]; Q=[1 2];  
F=[1 2 3]; D=convs(P,Q,F), E=conv(conv(P,Q),F)  
G=convs(P,Q,F,[1,1],[1,3],[1,1])
```

inline 函数与匿名函数

- inline 函数（功能重叠，不建议使用）
 - 可以免去编辑M-文件

fun=inline(function expression, list of variables)

- MATLAB7.0 以上版本提供的匿名函数
 - MATLAB工作空间内的变量可以直接使用

f=@(list of variables) function_contents

伪代码与代码保密处理

- 伪代码化的目的

- 提高程序的执行速度

- 保密：把ASCII的 .m文件转换成二进制代码

- 伪代码语句

```
pcode mytest          pcode *.m
```

```
pcode mytest -inplace
```

- 用户一定要在安全的位置保留.m源文件，不能轻易删除，因为.p文件是不可逆的



控制流程结构小结

- 循环结构的两种结构
 - for 结构与while结构
 - 两种结构的优缺点与适用范围
 - 语句break可以终止当前的循环
- 条件转移语句 if else elseif
- 开关结构 switch case及其与C语言的异同
- 试探结构 try catch



MATLAB函数编写小结

- 为什么需要函数结构而不是脚本结构
- 函数的入口与输入输出变元处理
 - 由function关键词引导
 - 输入、输出变元个数 nargin, nargout
 - 输入、输出变元的内容 varargin, varargout
- 递归调用结构
- 匿名函数、伪代码技术 pcode



Q & A

感谢您的聆听和反馈

国家精品课程/ 国家精品资源共享课程/ 国家级精品教材

国家级十一(二)五规划教材/ 教育部自动化专业教学指导委员会牵头规划系列教材

控制系统计算机辅助设计

第2章: MATLAB语言程序设计基础

主讲: 修贤超

MATLAB语言程序设计基础

- 2.1 程序设计语言基础
- 2.2 基本数学运算
- 2.3 MATLAB语言的流程结构
- 2.4 函数的编写与调试
- 2.5 二维图形绘制
- 2.6 三维图形绘制



二维图形绘制

- 科学可视化在科学研究中很重要，用图形的方式将数据直观地表示出来
 - 二维图形绘制基本语句
 - 多纵轴曲线绘制
 - 其他二维图形绘制语句
 - 隐函数绘制及应用
 - 图形修饰
 - 数据文件的读取与存储

二维图形绘制基本语句

➤ 已知数据绘图

➤ 两个序列 $t = t_1, t_2, \dots, t_n$ 和 y_1, y_2, \dots, y_n 构造向量: \dots, y_n

$$t = [t_1, t_2, \dots, t_n]$$

$$y = [y_1, y_2, \dots, y_n]$$

➤ 利用这些数据绘图 `plot(t, y)`

➤ 已知函数绘图

➤ 描述函数再绘图 `fplot(f, [x_m, x_M])`

例2-28 函数曲线绘制与检验

➤ 绘制函数

$$y = \sin(\tan x) - \tan(\sin x), \quad x \in [-\pi, \pi]$$

➤ MATLAB代码

```
>> x=[-pi : 0.05: pi];  
y=sin(tan(x))-tan(sin(x)); plot(x,y)
```

➤ 问题：如何检查曲线？

➤ 不同的步距，看看能否得出同样的结果

变步长向量

- 改用不同的步长，观察曲线是否相同
 - 全程换用较小步长
- 变步长：在 $\pm\pi/2$ 附近用小步长

```
>> x=[-pi:0.05:-1.8,-1.801:.001:-1.2,...  
      -1.2:0.05:1.2,1.201:0.001:1.8,...  
      1.81:0.05:pi];  
y=sin(tan(x))-tan(sin(x));  
plot(x,y)
```

例2-29 分段函数

➤ 绘制饱和函数方程
$$y = \begin{cases} 1.1\text{sign}(x), & |x| > 1.1 \\ x, & |x| \leq 1.1 \end{cases}$$

➤ MATLAB绘图语句（互斥条件）

```
>> x=[-2:0.02:2];  
    y=1.1*sign(x).*(abs(x)>1.1)+...  
      x.*(abs(x)<=1.1); plot(x,y)
```

➤ 更简单的命令——折线

```
>> plot([-2,-1.1,1.1,2],[-1.1,-1.1,1.1,1.1])
```

其他调用格式

- t 仍为向量，而 y 为矩阵，亦即 $\text{plot}(t, y)$

$$y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{bmatrix}$$

- t 和 y 为矩阵，且 t 和 y 的行和列数均相同
- 假设有对这样的向量或矩阵

$$(t_1, y_1), (t_2, y_2), \cdots, (t_m, y_m), \quad \text{plot}(t_1, y_1, t_2, y_2, \cdots, t_m, y_m)$$

更一般调用格式

➤ 改变曲线性质

```
h = plot( $t_1, y_1$ , option 1,  $t_2, y_2$ , option 2,  
        ...,  $t_m, y_m$ , option  $m$ )
```

line type	colour		marks	
'-'	'b'	'c'	'*'	'pentagram'
'--'	'g'	'k'	'.'	'o'
':'	'm'	'r'	'x'	'square'
'-.'	'w'	'y'	'v'	'diamond'
'none'			'^'	'hexagram'
			'>'	'<'

'--bv'

图形修饰与属性设置

- 每一个窗口、曲线和坐标轴都是一个对象
- 对象的属性可以通过函数 `set()` 来设置
- 可以通过函数 `get()` 来获取

```
set(handle, 'p_name 1', ...  
    p_value 1, 'p_name 2', p_value 2, ...)  
  
v = get(object, 'p_name')  
      'Color', [1 0 0])
```

- 图形对象的属性还可以通过快捷菜单（鼠标右键）直接修改

例2-30 多纵轴曲线的绘制

➤ 若两条曲线的幅值相差悬殊，调用双纵轴绘制函数`plotyy()`

➤ 试绘制曲线 $y_1 = \sin x$, $y_2 = 0.01 \cos x$

```
>> x=0:0.01:2*pi;  
    y1=sin(x); y2=0.01*cos(x);  
    plot(x,y1,x,y2,'--')  
  
>> plotyy(x,y1,x,y2)
```

➤ 三、四纵轴图形可以下载相应函数绘制

➤ `plotyyy()`、`plot4y()`，从MathWorks File Exchange下载，还可以使用`plotxx()`函数

其他二维图形绘制语句

➤ 不同的绘制函数

<code>bar(x,y)</code>	<code>comet(x,y)</code>
<code>compass(x,y)</code>	<code>errorbar(x,y,y_m,y_M)</code>
<code>feather(x,y)</code>	<code>fill(x,y,c)</code>
<code>hist(y,n)</code>	<code>loglog(x,y)</code>
<code>polarplot(x,y)</code>	<code>quiver(x,y)</code>
<code>stairs(x,y)</code>	<code>stem(x,y)</code>
<code>semilogx(x,y)</code>	<code>semilogy(x,y)</code>

例2-31 极坐标曲线

➤ 绘制极坐标函数

$$\rho = 5 \sin(4\theta/3)$$

$$\rho = 5 \sin(\theta/3)$$

➤ 绘制曲线

```
>> theta=0:0.01:6*pi;  
rho=5*sin(4*theta/3);  
polarplot(theta,rho)
```

```
>> rho=5*sin(theta/3); polarplot(theta,rho)
```

例2-32 不同的二维图形

- 以正弦数据为例，试在同一窗口的不同区域用不同的绘图方式绘制出相应的曲线
- MATLAB求解语句：

```
>> t=0:.2:2*pi; y=sin(t);  
subplot(2,2,1), stairs(t,y)  
subplot(2,2,2), stem(t,y)  
subplot(2,2,3), bar(t,y)  
subplot(2,2,4), semilogx(t,y)
```

绘图区域设置

- 下述函数用于将一个绘图窗口分割成几个不同的区域

`subplot(n, m, k)`

- n 和 m 分别是行数和列数
- k 则指示区域的序号
- 利用图形窗口菜单
 - 添加坐标系
 - 拖动鼠标调整坐标系的大小

隐函数绘制及应用

➤ 隐函数举例 $x^2 \sin(x + y^2) + y^2 e^{x+y} + 5 \cos(x^2 + y) = 0$

➤ 隐函数绘图语句 $f(x,y) = 0$

`fimplicit(implicit function expression)`

➤ 用符号表达式、匿名函数表示隐函数

➤ 默认区域是 $[-5,5]$ ，还可以使用早期版本的 `ezplot` 函数

➤ 其他语法 `fimplicit(im_function, [xm, xM])`

`fimplicit(im_function, [xm, xM, ym, yM])`

例2-33 隐函数曲线

➤ 试绘制隐函数 $x^2 \sin(x + y^2) + y^2 e^{x+y} + 5 \cos(x^2 + y) = 0$

➤ MATLAB语句

```
>> syms x y;  
h=fimplicit(x^2*sin(x+y^2)+y^2*exp(x+y)+5*cos(x^2+y));  
set(h,'Color','b')
```

➤ 扩大范围

```
>> h=fimplicit(x^2*sin(x+y^2)+y^2*exp(x+y)+...  
5*cos(x^2+y),[-10 10]);  
set(h,'Color','b')
```

数据文件的读取与存储

- 可以采用save和load命令存储和读取数据

```
save mydat A B C
```

```
save /ascii mydat.dat A B C
```

```
X=load(filename)
```

- MATLAB和Excel交互数据

```
X=xlsread(filename,range) 'B6:C67'
```

- 写文件 xlswrite()

例2-34 Excel文件读取

- 已知Excel文件 census.xls给出某省人口数
 - 第5-67行存储数据
 - B列存储年份，C列存储人口数
- 先读入MATLAB再绘图

```
>> X=xlsread('census.xls','B5:C67');  
      t=X(:,1); p=X(:,2); plot(t,p)
```

- 更简单方法——Copy & Paste

MATLAB语言程序设计基础

- 2.1 程序设计语言基础
- 2.2 基本数学运算
- 2.3 MATLAB语言的流程结构
- 2.4 函数的编写与调试
- 2.5 二维图形绘制
- 2.6 三维图形绘制

三维图形表示

- 三维图形用常规的计算机语言难以绘制，使用MATLAB可以容易地得出
- 本节主要内容
 - 三维曲线绘制
 - 三维曲面绘制
 - 三维带状图的绘制
 - 三维图形视角设置

三维曲线绘制

➤ 三维曲线绘制（空间质点的运动轨迹）

```
plot3(x, y, z)
```

```
plot3(x1, y1, z1, option 1, x2, y2, z2, option 2, ...
```

```
    xm, ym, zm, option m)
```

➤ 其他三维曲线绘制函数

➤ stem3, fill3, bar3等

➤ 绘制三维轨迹图：comet3

例2-35 空间质点的位置

➤ 试绘制参数方程

$$x(t) = t^3 \sin(3t)e^{-t}, y(t) = t^3 \cos(3t)e^{-t}, z = t^2$$

➤ 质点的空间位置（随时间变化）

➤ 其中， $t \in [0, 2\pi]$

➤ MATLAB绘图

```
>> t=0:0.01:2*pi;  
x=t.^3.*sin(3*t).*exp(-t); y=t.^3.*cos(3*t).*exp(-t);  
z=t.^2; plot3(x,y,z), grid
```

其他曲线绘制

➤ 使用 `stem3()` 函数

```
>> stem3(x,y,z); hold on;  
plot3(x,y,z), grid; hold off
```

➤ 运行轨迹 `comet3(x,y,z)`

➤ 图形窗口的工具栏

➤ 3D绘图和视角变换

➤ 读取坐标值、局部放大

三维曲面绘制

- 一般曲面绘制 $z = f(x, y)$

 - `[x, y] = meshgrid(v1, v2)`

 - `z = ...`, for instance `z = x.*y`

 - `surf(x, y, z)` or `mesh(x, y, z)`

- 其他函数

 - `surf1()`, `surfc()`

- 等高线绘制

 - `contour()`, `contours()`

例2-36 三维曲面

- 给出二元函数如下，绘制3D图形

$$z = f(x, y) = (x^2 - 2x)e^{-x^2 - y^2 - xy}$$

- MATLAB绘图

```
>> [x,y]=meshgrid(-3:0.1:3,-2:0.1:2);  
z=(x.^2-2*x).*exp(-x.^2-y.^2-x.*y);  
mesh(x,y,z)
```

- 表面图

```
>> surf(x,y,z)
```

例2-37 另一个函数

- 试绘制出二元函数

$$z = f(x, y) = \frac{1}{\sqrt{(1-x)^2 + y^2}} + \frac{1}{\sqrt{(1+x)^2 + y^2}}$$

- 绘制3D图形

```
>> [x,y]=meshgrid(-2:.1:2);  
z=1./((sqrt((1-x).^2+y.^2))+1./((sqrt((1+x).^2+y.^2)));  
surf(x,y,z), shading flat
```

条带图的绘制

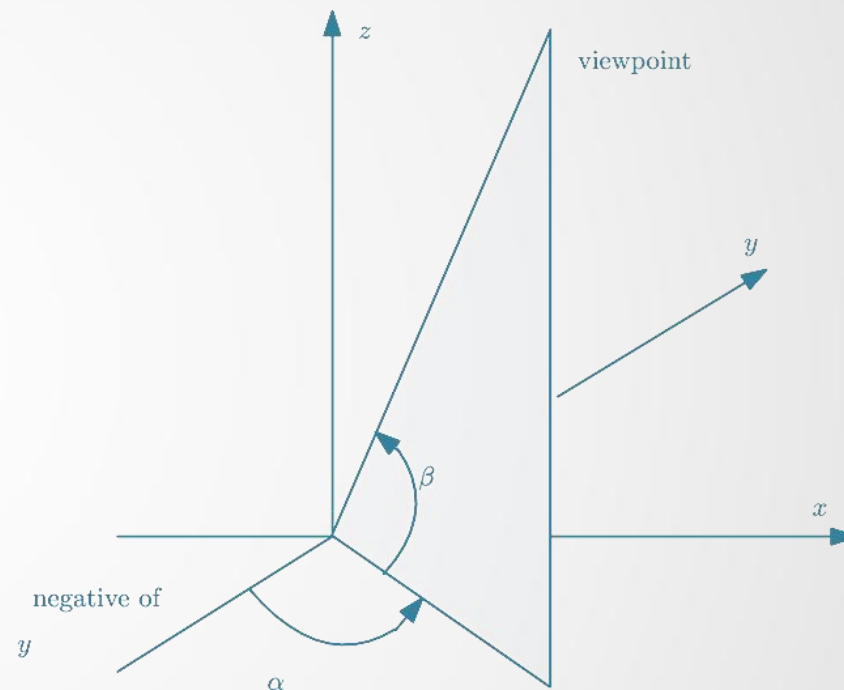
- 二阶控制系统的阶跃响应 $G(s) = \frac{1}{s^2 + 2\zeta s + 1}$
- 一些命令将在后续章节介绍
- 可以用循环结构绘制条带图，局限性

```
>> zet=0:0.1:1.4; t=[0:0.1:10]'; Y=[];  
    for zeta=zet  
        G=tf(1,[1,2*zeta,1]); y=step(G,t); Y=[Y y];  
    end  
    ribbon(t,Y,0.2)
```

```
>> surf(zet,t,Y)
```

三维图形视角设置

- 两种方法改变图形的视角
 - 直接采用工具栏
 - 命令语句 `view()`
`view(α , β)`
 - 读角度 `[α , β]=view(3)`
- α 定义为方位角， β 定义为仰角



例2-38 三视图的设置

➤ 函数 $z = f(x, y) = (x^2 - 2x)e^{-x^2 - y^2 - xy}$

➤ 默认视角的提取 `>> [a b]=view(3);`

➤ 三视图绘制的MATLAB代码

```
>> [x,y] = meshgrid(-3:0.1:3,-2:0.1:2);  
z=(x.^2-2*x).*exp(-x.^2-y.^2-x.*y);  
subplot(224), surf(x,y,z),  
subplot(221), surf(x,y,z), view(0,90);  
subplot(222), surf(x,y,z), view(90,0);  
subplot(223), surf(x,y,z), view(0,0);
```



二维曲线绘制小结

- 如果有数据，就可以考虑用图形表示
 - 画图函数plot
 - 函数plot的不同调用方法与修饰、
 - 图形对象的句柄
- 多纵轴图形的绘制方法 plotyy
- 多横轴图形



特殊二维图形绘制小结

- 不同的画图命令，如stem, stairs, comet, bar, hist, semilogx
- 图形窗口的分割 subplot
- 隐函数的绘制 fimplicit, ezplot
- 图形修饰、LaTeX支持
- 数据文件读写 save/load, xlsread/xlswrite



三维图形绘制小结

- 两种三维图形
 - 三维曲线——空间质点的轨迹 `plot3`, `comet3`
 - 三维曲面 $z = f(x,y)$, `meshgrid`, `mesh`, `surf`等
 - 介于二者之间, 条形图, `ribbon`
- 三维视角的设置 `view`
- 三视图:
 - 俯视 `view(0,90)`, 主视 `view(0,0)`, 右视 `view(0,90)`



Q & A

感谢您的聆听和反馈