

# 第二讲

数值数组及运算  
字符串、元胞和构架数组

# 讲授内容

- 一、二维数组的创建、寻访
- 数组运算和矩阵运算的区别
- 实现数组的基本函数
- 多项式的表达、创建和操作
- 常用标准生成函数和数组构造技法
- 高维数组的创建、寻访和操作
- 非数NaN、空数组概念和应用
- 关系和逻辑操作
- 字符串数组、元胞数组和构架数组

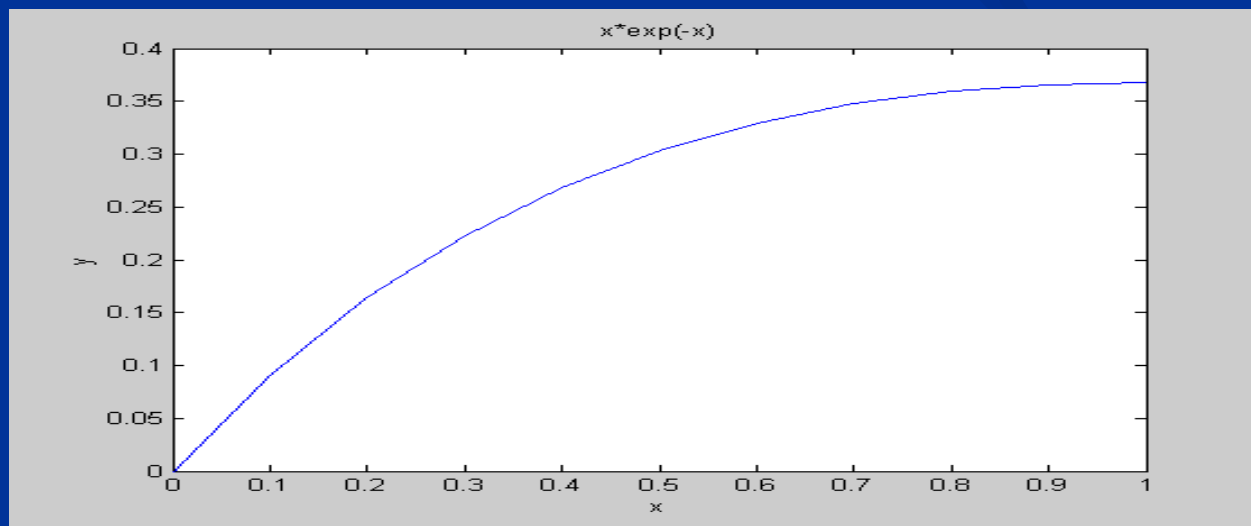
# 一、引导

- 例子：绘制函数  $y = xe^{-x}$  在  $0 \leq x \leq 1$  的曲线

`x=0:0.1:1`      %定义自变量的采样点取值数组

`y=x.*exp(-x)`    %利用数组运算计算各自变量采  
                         %样点上的函数值

`plot(x,y),xlabel('x'),ylabel('y'),title('y=x*exp(-x)')`  
%绘图



## 二、一维数组的创建和寻访

# 一维数组的创建

- 逐个元素输入法

`X=[2 pi/2 sqrt(3) 3+5i 6]`

- 冒号生成法

`X=a:inc:b`

- 定数线性采样法

`x=linspace(a,b,n)`

- 定数对数采样法

`X=logspace(a,b,n)`

在信号处理和控制科学中，常用来产生频率响应的  
频率自变量采样点

# 一维数组子数组的访问

$X(3)$  访问数组 $x$ 的第三个元素

$X([1\ 2\ 5])$  访问数组 $x$ 的第一、二、五元素

$X(1:3)$  访问数组 $x$ 第1到3元素

$X(3:end)$  访问数组 $x$ 第3到最后一个元素

$X(3:-1:1)$  访问数组 $x$ 第3、2、1元素

$X(\text{find}(x>0.5))$  访问大于0.5元素的子数组

$X([1\ 2\ 3\ 3\ 2\ 1])$  对元素可以重复访问

# 一维数组的赋值

$$X(3)=0$$

$$X([1\ 4])=[2\ 2]$$

说明：要保证被重新赋值的子数组的长度与送入的数组长度相同

# 三、二维数组的创建

## 1.直接输入法

- 整个数组必须以方括号“[]”为首尾
- 数组的行与行之间必须用分号“;”或者[Enter]隔离;
- 数组元素必须由“,”或空格分隔;

$M_r=[1,2,3;4,5,6], M_i=[11\ 12\ 13;14\ 15\ 16]$

$CN=M_r+i*M_i$



# 三、二维数组的创建

## 2.利用M文件创建和保存数组

对于今后经常需要调整的数组，尤其是比较大而复杂的数组，为他专门建立一个M文件是值得的。

- 打开文件编辑器，输入数组AM
- 保存此文件，命名MyMatrix.m
- 在Matlab指令窗中，运行MyMarix.m文件，数组AM就会自动生成到Matlab内存中

## 四、二维数组的标识

- “全下标”标识  $A(3,5)$

- “单下标”标识  $A(4)$

先对二维元素进行“一维编号”

第 $r$ 行 $c$ 列对应的单下标:  $l=(c-1)*m+r$

sub2ind 据全下标换算出单下标

ind2sub 据单下标换算出全下标

- “逻辑1”标识

通过与 $A$ 同样大小的逻辑数组 $L$ 中“逻辑1”所在的位置, 指示 $A$ 中元素的位置

■ 例：找出数组  $A = \begin{bmatrix} -4 & -2 & 0 & 2 & 4 \\ -3 & -1 & 1 & 3 & 5 \end{bmatrix}$  中

所有绝对值大于3的元素

```
A=zeros(2,5);
```

```
A(:)=-4:5;
```

```
L=abs(A)>3
```

```
islogical(L)
```

```
X=A(L)
```

## 五、二维数组的子数组的寻访赋值

|             |                        |
|-------------|------------------------|
| $A(r,c)$    | 由A的“r指定行”和“C制定列”上的元素组成 |
| $A(r,:)$    | 由A的“r指定行”和全部列上的元素组成    |
| $A(:,r)$    |                        |
| $A(:)$      | 单下标全元素，形成一维长列          |
| $A(s)$      | 单下标寻访                  |
| $A(L)$      | 逻辑1寻访，生成一维列数组          |
| $A(r,c)=sa$ | 对A(r,c)进行赋值            |
| $A(:)=D(:)$ | 全元素赋值方式                |
| $A(s)=sa$   |                        |

# 六、执行数组运算的常用函数

1.定义:  $f(X)=[f(x_{ij})]_{m \times n}$

2.执行数组运算的常用函数

- 三角函数和双曲函数
- 指数函数
- 复数函数
- 园整函数和求余函数
- 坐标变换函数
- 其他特殊函数

# 七、矩阵和数组的基本运算

从外观形状和数据结构上看，二维数组和数学中的矩阵没有区别，但是矩阵作为映射算子的体现，矩阵有着明确而严格的数学规则，下表对数组运算和矩阵运算作了比较。

| 数组运算             |                 | 矩阵运算            |           |
|------------------|-----------------|-----------------|-----------|
| 指令               | 含义              | 指令              | 含义        |
| $A.'$            | 非共轭转置           | $A'$            | 共轭转置      |
| $s+B$            | 标量s分别与B之和       |                 |           |
| $s.*A$           | S分别与A的元素之积      | $S*A$           |           |
| $A.^n$           | A的每个元素自乘n次      | $A^n$           | A为方阵，自乘n次 |
| $A+B$            | 对应元素相加          | $A+B$           | 矩阵相加      |
| $A.*B$           | 对应元素相乘          | $A*B$           | 内维相同矩阵的乘积 |
| $A ./ B$         | A 的元素被 B 的对应元素除 | $A / B$         | A 右除 B    |
| $B.\backslash A$ | 一定与上相同          | $B\backslash A$ | A 左除 B    |

## ■ 例：两种不同转置的比较

■ `A=zeros(2,3);`

■ `A(:)=1:6;`                      %全数组赋值法

■ `A=A*(1+i);` %运用标量和数组乘产生复数矩阵

■ `A_A=A.'`                      %数组转置，即非共轭转置

■ `A_M=A'`                      %矩阵转置，即共轭转置



## 八、多项式的表达方式及其操作

# 1.多项式的表达和创建

- Matlab约定向量中的元素是按多项式降幂排列的

$$f(x) = a_1 x^n + a_2 x^{n-1} + \dots + a_n x + a_{n+1}$$

可用行向量  $p = [a_1 \ a_2 \ \dots \ a_n \ a_{n+1}]$  表示

- 多项式行向量创建方法

(1) 多项式系数向量直接输入法

(2) 利用指令  $P = \text{poly}(AR)$ , 产生多项式系数向量



■ 例：由给定根向量求多项式系数向量

$R = [-0.5, -0.3 + 0.4i, -0.3 - 0.4i];$  %根向量

$P = \text{poly}(R)$  %R的特征多项式

$PR = \text{real}(P)$  %求PR的实部

$PPR = \text{poly2str}(PR, 'x')$

$P = 1.0000 \quad 1.1000 \quad 0.5500 \quad 0.1250$

$PR = 1.0000 \quad 1.1000 \quad 0.5500 \quad 0.1250$

$PPR = x^3 + 1.1 x^2 + 0.55 x + 0.125$

## 2.多项式的运算函数

## 2.1 roots —— 求多项式的根

```
a=[1 2 3;4 5 6;7 8 0];
```

```
p=poly(a)
```

```
p =
```

```
1.00 -6.00 -72.00 -27.00
```

```
r=roots(p)
```

```
r = 12.12
```

```
-5.73 ——显然 r是矩阵a的特征值
```

```
-0.39
```

当然我们可用poly令其返回多项式形式

```
p2=poly(r)
```

```
p2 =
```

```
1.00   -6.00  -72.00  -27.00
```

- matlab规定多项式系数向量用行向量表示，一组根用列向量表示。

## 2.2.conv, convs 多项式乘运算

例: $a(x)=x^2+2x+3$ ;  $b(x)=4x^2+5x+6$ ;

$$c = (x^2+2x+3)(4x^2+5x+6)$$

$a=[1 \ 2 \ 3]$ ;  $b=[4 \ 5 \ 6]$ ;

$c=\text{conv}(a,b)=\text{conv}([1 \ 2 \ 3],[4 \ 5 \ 6])$

$c = 4.00 \ 13.00 \ 28.00 \ 27.00 \ 18.00$

$p=\text{poly2str}(c,'x')$

$p = 4 x^4 + 13 x^3 + 28 x^2 + 27 x + 18$



## 2.3 deconv多项式除运算

$a=[1\ 2\ 3];$

$c = [4.00\ 13.00\ 28.00\ 27.00\ 18.00]$

$d=\text{deconv}(c,a)$

$d = 4.00\quad 5.00\quad 6.00$

$[d,r]=\text{deconv}(c,a)$

└───→ 余数

└───→  $c$ 除 $a$ 后的整数

## 2.4 多项式微分

matlab提供了polyder函数多项式的微分。

命令格式:

polyder(p): 求p的微分

polyder(a,b): 求多项式a,b乘积的微分

[p,q]=polyder(a,b): 求多项式a,b商的微分

例: a=[1 2 3 4 5]; poly2str(a,'x')

ans =  $x^4 + 2x^3 + 3x^2 + 4x + 5$

b=polyder(a)

b = 4      6      6      4

poly2str(b,'x')

ans =  $4x^3 + 6x^2 + 6x + 4$

## 九、标准数组生成函数和 数组操作函数

# 标准数组生成函数

- `diag`          产生对角形数组
- `eye`            产生单位数组
- `magic`          产生魔方数组
- `ones`           产生全1数组
- `rand`           产生均匀分布的随机数组
- `randn`          产生正态分布的数组
- `zeros`          产生全0的数组

```
» eye(2,3)
```

```
ans=
```

```
1 0 0
```

```
0 1 0
```

```
» zeros(2,3)
```

```
ans=
```

```
0 0 0
```

```
0 0 0
```

```
» ones(2,3)
```

```
ans=
```

```
1 1 1
```

```
1 1 1
```

```
» V=[5 7 2]; A=diag(V)
```

```
A=
```

```
5 0 0
```

```
0 7 0
```

```
0 0 2
```

```
» eye(2)
```

```
ans=
```

```
1 0
```

```
0 1
```

```
» zeros(2)
```

```
ans=
```

```
0 0
```

```
0 0
```

```
» ones(2)
```

```
ans=
```

```
1 1
```

```
1 1
```

如果已知 $A$ 为方阵，则 $V=\text{diag}(A)$ 可以提取 $A$ 的对角元素构成向量 $V$ 。

# 数组操作函数

| 指令      | 含义                            |
|---------|-------------------------------|
| cat     | 把“大小”相同的若干数组，沿“指定维”方向，串联成高维数组 |
| diag    | 提取对角元素，或生成对角阵                 |
| flipdim | 以指定维的对称轴进行翻转                  |
| fliplr  | 左右交换                          |
| flipud  | 上下交换                          |
| kron    | Kronecker乘法                   |
| repmat  | 按指定的次序，对矩阵各维的次序进行重组           |
| reshape | 改变各维的大小                       |
| rot90   | 逆时针旋转二维数组                     |
| tril    | 提取数组下三角部分，生成下三角阵              |
| triu    | 生成上三角阵                        |

# 十、数组构造技法综合

- 数组的扩展
  - (1) 数组的赋值扩展法
  - (2) 多次寻访扩展法
  - (3) 合成扩展法
- 提取子数组，合成新数组
- 单下标寻访和reshape指令
- 对行或列通加一个数的方法
- 逻辑函数的运用

# 十一、高维数组

## 1.高维数组的创建

- 直接通过“全下标”元素赋值
- 由若干同样大小的低维数组组合成高维数组
- 由函数ones,zeros,rand,randn直接创建
- 借助cat, repmat, reshape等函数构造



例：全下标赋值

`A(2,2,2)=1`

`B(2,5,:)=1:3`

低维数组组合

`A=ones(2,3)`

`A(:, :, 2)=ones(2,3)*2;`

`A(:, :, 3)=ones(2,3)*3`

由函数`ones`,`zeros`,`rand`,`randn`直接创建

`rand(2,4,3)`

借助`cat`,`repmat`,`reshape`函数

`cat(3,ones(2,3),ones(2,3)*2,ones(2,3)*3)`

`repmat(ones(2,3),[1,1,3])`

`reshape(1:12,2,2,3)`

# 十一、高维数组

## 2、高维数组的标识

(1) 数组的维数和ndims指令

(2) 数组的大小和size指令

```
clear; A=reshape(1:24,2,3,4)
```

```
dim_A=ndims(A)
```

```
size_A=size(A)
```

```
L_A=length(A);
```

# 十二、非数和空数组

## 1. 非数NaN

### ■ 非数的定义

$0/0, \infty/\infty, 0 \times \infty$  等运算都会产生非数，用NaN或nan记述

### ■ 非数的性质

NaN参与运算所得的结果也是NaN，具有传递性  
非数没有大小的概念，不能比较两个非数的大小

### ■ 非数的功用

真实记述 $0/0, \infty/\infty, 0 \times \infty$ 等运算的后果

避免可能因 $0/0, \infty/\infty, 0 \times \infty$ 等运算造成执行的中断  
在数据可视化中用来剪裁图形

## 2.空数组

- 定义

某维长度为0或若干维长度为0的数组

- 功用

在没有空数组参与的运算时，计算结果中的空可以合理解释所得的结果没有含义；  
运用空数组对其他非空数组赋值，可以收缩数组的大小，而不改变那数组的维数。

- `A=reshape(-4:5,2,5)`

- `L2=A>10;`

- `find(L2);`

- `A(:,[2,4])=[]`

- 用于子数组的删除和数组大小的收缩

# 十三、关系操作和逻辑操作

## ■ 关系操作

| 关系符号 | 意义    |
|------|-------|
| <    | 小于    |
| <=   | 小于或等于 |
| >    | 大于    |
| >=   | 大于或等于 |
| ==   | 等于    |
| ~=   | 不等于   |

■ 例：关系操作运算：求近似极限、修补图形窗口

```
t=-2*pi:pi/10:2*pi;      %存在0根
```

```
y=sin(t)./t              %在t=0处，产生NaN
```

```
tt=t+(t==0)*eps;  %逻辑数组参与运算，0被一个  
机器零小数代替
```

```
yy=sin(tt)./tt;
```

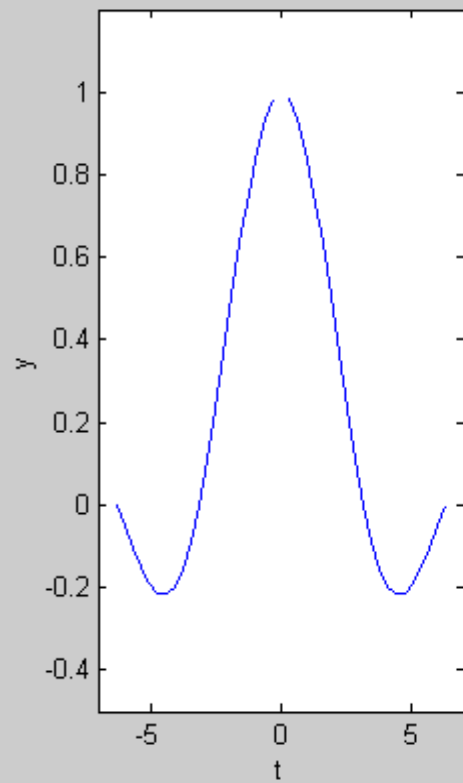
```
subplot(1,2,1),plot(t,y),axis([-7,7,-0.5,1.2]);
```

```
xlabel('t'),ylabel('y'),title('残缺图形');
```

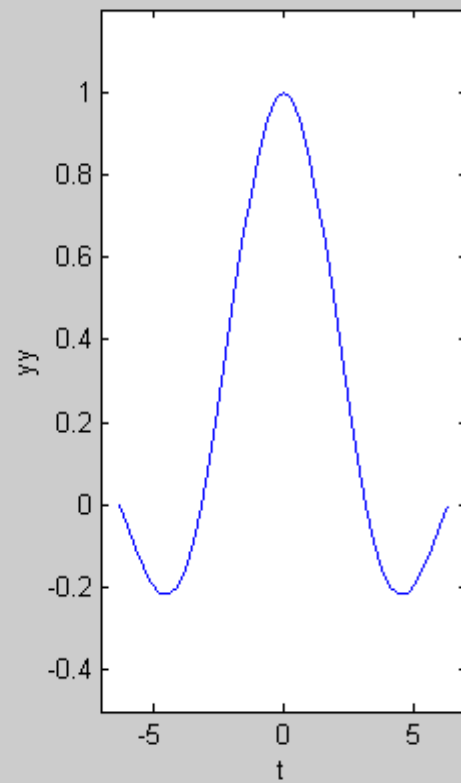
```
subplot(1,2,2),plot(tt,yy),axis([-7,7,-0.5,1.2]);
```

```
xlabel('tt'),ylabel('yy'),title('正确图形')
```

残缺图形



正确图形





# 逻辑操作

## ■ 数组逻辑操作

&    逻辑与    and

|    逻辑或    or

~    逻辑非    not

xor 相异元素返回1，相同元素返回0

## ■ 二进制位逻辑操作

| 指令     | 含义              | 举例： $A=28 \Rightarrow 11100$<br>$B=11 \Rightarrow 1011$ |
|--------|-----------------|---|
| bitand | 数位与、和           | $A \& B \Rightarrow 1000 \Rightarrow 8$                 |
| bitor  | 数位或             | $A   B \Rightarrow 1111 \Rightarrow 15$                 |
| bitcmp | 数位否、非           | $\sim B \Rightarrow 0100 \Rightarrow 4$                 |
| bitxor | 相异数位返回1，相同数位返回0 | $\text{xor}(A, B) \Rightarrow 0111 \Rightarrow 9$       |

## ■ 先决逻辑操作

| 指令 | 含义                         | 举例                            |
|----|----------------------------|-------------------------------|
| && | 当左被运算量为“真”时，与右被算量执行“与”逻辑运算 | $(b \neq 0) \&\& (a / b > 5)$ |
|    | 两个运量量有一个为真的时候就为真           | $(b == 0)    (a / b > 0)$     |

# 十四、字符串数组

- 1、字符串的约定
  - ✓ 字符串用单引号输入或赋值;
  - ✓ 字符串的每个字符都是都是字符数组的一个元素;
  - ✓ 字符串和字符数组基本上等价。

```
>>s='symbolic'
```

```
s =
```

```
symbolic
```

```
>>size(s)
```

```
ans =
```

```
1    8
```

```
>>s(3)
```

```
ans =
```

```
m
```

- 字符串的转换

`double` 字符串转换为数值代码

`num2str` 数字转换为字符串

`int2str` 整数转换为字符串

`mat2str` 矩阵转换为字符串

`str2num` 转换字符串为数字

```
>>double(s)
```

```
ans =
```

```
49 50 51 50 51 52
```

- 字符串操作

strcat strcmp strvcap strncmp

findstr upper lower blanks

deblank ... ..

- 执行字符串

```
>>t='1/(a*b)';a=2;b=3;c=eval(t)
```

```
c =
```

```
0.1667
```

# 十五、元胞数组

## 1.元胞数组的创建和显示

### ■ 元胞标识寻访和内容编址寻访的不同

对二维数组:  $A(2,3)$  表示数组A的第2行第3列上的元素

对元胞数组:  $A(2,3)$ 元胞元素

$A\{2,3\}$ 元胞所允许存放的内容

■ 例：演示（2×2）元胞数组的创建

```
c_str=char('this','is cell');
```

```
r=reshape(1:9,3,3);
```

```
cn=[1+2i];
```

```
s_sym=sym('sin(-3*t)*exp(t)');
```

(1) 外标识元胞元素赋值法

```
A(1,1)={c_str};A(1,2)={r};A(2,1)={cn};
```

```
A(2,2)={s_sym};
```

(2) 编制元胞元素内涵的直接赋值法

```
B{1,1}=c_str;B{1,2}=r;B{2,1}=cn;
```

```
B{2,2}=s_sym;
```

(3) celldisp(B) 显示元胞数组内容

(4) cellplot(A,'legend')



## 2 元胞数组的扩充、收缩和重组

### ■ 扩充

(1) 利用cell指令创建元胞数组

```
C=cell(2);
```

```
C(:,1)={char('another','string');10:-1:1}
```

(2)元胞数组的“列”扩充和“行扩充”

```
AC=[A C];A_C=[A;C]
```

### ■ 收缩和重组

```
A_C(3,:)=[] %删除第三行
```

```
R_A_C=reshape(A_C,2,3)
```

### ■ 3. 元胞数组内容的调取示例

(1) 调取一个元胞

```
f1=R_A_C(1,3) %得到的元胞
```

(2) 调取一个元胞的内容

```
f2=R_A_C{1,3} %得到的是内容
```

(3) 调取元胞内的数组

```
f3=R_A_C{1,1}(:,[1 2 5 6])
```

(4) 同时调取多个元胞内容

```
[f4,f5,f6]=deal(R_A_C{[1,3,4]})
```

## 4 元胞数组转换函数

- `C=num2cell(A,dimN)`
- `C=mat2cell(B,M,N)`
- `D=cell2mat(C)`

# 十六、构架数组

- 构架数组和元胞数组一样，在一个数组中存放各类数据，从一定意义上讲，构架数组的能力比元胞数组更强。
- 构架数组的基本组分是构架。构架必须在划分“域”后才能使用。

# 构架数组的创建和显示

- 直接创建法及显示

- (1) 创建

- `green_house.name='No1';`
    - `green_house.volume='2000';`
    - `green_house.parameter.temperature=...`  
`[31.2 30.4 31.6];`
    - `green_house.parameter.humidity=...`  
`[62.4 59.5 57.7];`

- (2) 显示

- `green_house`

- `green_house.parameter`

- `green_house.parameter.temperature`

(3) 直接对域赋值法“构架数组”

```
green_house(2,3).name='6';
```

(4) 显示构架数组的结构和内容

```
green_house
```

```
green_house(2,3)
```

■ 利用构造函数struct创建  
方法（1）：

■ `cell(2,3);`

■ `green_house_1=struct('name',a,'volume',a,...`  
■ `'parameter',a(1,2))`

方法（2）：

`green_house_2=struct('name',a,'volume',[],`  
`'parameter',[]);`

方法（3）：

`green_house_3(2,3)=struct('name',[],'volume',[],`  
`'parameter',[]);`

方法（4）：

`a1={'6'};a2={'3200'};`  
`green_house_4(2,3)=struct('name',a1,'volume',a2,`  
`parameter',[]);`

## 十七、四种主要数据类型基本构成比较

| 数据类型  | 基本组分 | 组分内涵                              | 基本组分占用字节数 |
|-------|------|-----------------------------------|-----------|
| 数值数组  | 元素   | 双精度实数标量<br>或双精度复数标量               | 8<br>16   |
| 字符串数组 | 元素   | 字符                                | 2         |
| 元胞数组  | 元胞   | 可以存放任何类型、任何大小的数据                  | 不定        |
| 构架数组  | 构架   | 只有挂接在构架上的“域”才能存放数据，数据可以是任何类型、任何大小 | 不定        |



## 小 结

本节介绍了matlab语言的数值数组、字符串、元胞和构架数组的创建和寻访，通过学习应该掌握：

- 如何创建数组、修改数组
- 标准函数的生成
- 字符串的创建
- 元胞数组的创建和调取
- 构架数组的创建、调取和设置