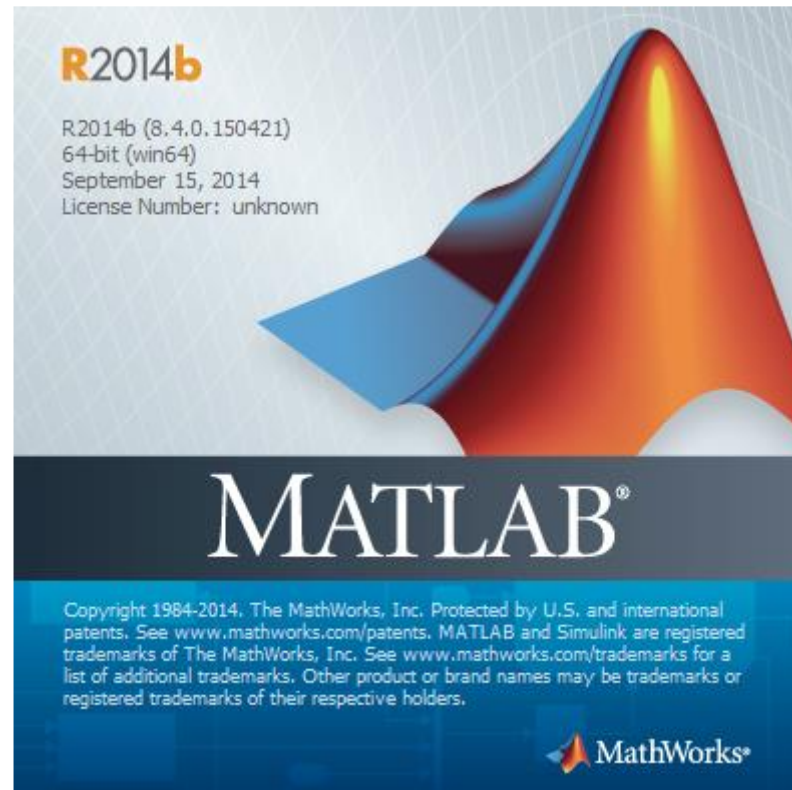


第2章-20200302更新

MATLAB软件与应用



前言

授课：雨课堂**GAXZUT**+腾讯会议000000000

改为腾讯课堂？

备用：学校课程平台(下载课件自学)

课后交流：QQ群**1046970565**

第二章 矩阵运算

- 常量，变量，数据类型
- 矩阵的类型与定义方式
- 矩阵的基本操作与运算
- 字符串、元胞与结构

字符串

字符串

- 字符可构成字符串，或字符数组/字符矩阵
- 字符串可被视为行向量
- 字符串中每个字符（含空格）作为一个元素

字符串相当于一种特殊数组/矩阵
沿用了矩阵的一般操作方法和习惯

字符串

- 用英文半角单引号 ' ' 来界定一个字符串
- 多个字符串变量可用方括号 [] 连接组成新的字符串变量，实质上属于矩阵的拼接

【例】命名字符串变量

```
str1 = 'I like MATLAB,';    % 建立字符串变量 str1
str2 = ' JavaScript, and Perl!';    % 建立字符串变量str2
str3 = [str1 str2]          % 直接连接str1及str2，以建立str3
```

str3 =

I like MATLAB, JavaScript, and Perl!

[str1 , str2] 和 [str1; str2] 是否正确？会得到什么结果？

字符串

- 字符串内容中的单引号，由两个连续的单引号来表示
- **length**命令也可计算字符串变量的长度（字符数量）

【例】

```
str1 = 'Let's go. '; % 字符串为let's go.
```

```
length(str1) % 计算字符串长度9
```

ans =

9

软件会通过颜色变化
对字符串书写是否
正确给出提示。



字符串未完成

正确会变紫色

引号错误

字符串

- **double** 查看字符串变量对应的 ASCII 内码
也可以使用int8 int16等
- **char** 将 ASCII 码转换为字符串形式
- **num2str**将数字转换为字符串形式

【例】 字符串有关的转换

```
x = 'I've got a date!';
```

```
y = double(x) % x每个字符元素的 ASCII 码
```

```
y2 = num2str(y) % 数字转化为字符串
```

```
z = char(y) % 将ASCII 码恢复成字符串
```

注意y与y2的区别

字符串

■ **class** 或 **ischar**

判断变量的类型或它是否为字符串变量

【例】 判断一个变量是否为字符串变量

```
chinese = '今日事，今日毕';
```

```
out1 = class(chinese)    % out1 的值是 char
```

```
x = chinese+1;
```

```
out2 = ischar(x)  % out2 是 0 代表 不是字符串变量
```

```
out3 = class(x)  % out3是double代表 x 不是字符串
```

多行字符串的存储

■ 方法1: 直接使用中括号[] 创建二维字符矩阵

必须确保每个字符串(即每一行)的长度一样, 这是矩阵的一般要求, 否则需要在较短字符串结尾补齐空格(自己做, 非自动)

【例】: 二维字符数组存储多行字符串变量

```
s = ['aa  '; 'bb  '; 'cccc'] %注意空格
```

s =

aa

bb

cccc

这里分别有
两个空格

多行字符串的存储

- **方法2:** 用char 指令存储多字符串（仍产生二维字符矩阵，每行长度不同时会自动补齐空格）

`char('aa', 'bb', 'cccc')` % 注意空格及引号

可使用 `deblank` 来清除字符串尾部的空格。

【例】：使用`deblank`命令清除字符串尾部空格

```
d= char('aa', 'bb', 'cccc');
```

```
d1 = d(1,:);           % (1,:)代表第一行的全部元素
```

```
d2 = deblank(d1);      % 使用 deblank 指令清除尾部的空格字符
```

```
length(d1)             % 显示变量 d1 的长度=4
```

```
length(d2)             % 显示变量 d2 的长度=2
```

字符串常用操作

- **strcmp** : 比较字符串内容, 相同为1, 不同为0

【例】

```
str1 = 'today';
```

```
str2 = 'tomorrow';
```

```
str3 = 'today';
```

```
out1 = strcmp(str1, str2)    % 比较字符串 str1 和 str2
```

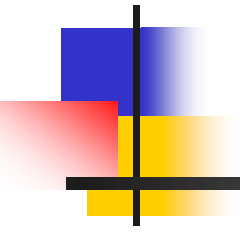
结果 out1 =

0

```
out2 = strcmp(str1, str3)    % 比较字符串 str1 和 str3
```

结果 out2 =

1



元胞

元胞数组 (cell)

- 特殊数据类型，一个矩阵中存放不同类型的数据
- 每个单元相当于一个“盒子”
- “盒子”可容纳不同类型的MATLAB数据

元胞数组的一个元素也称为一个单元或一个元胞(cell)。

【例】 2x2元胞矩阵，各元素数据类型可不一致，包含字符串，浮点型，甚至包含不同形状的矩阵。

'date'	[1,2,3]
'ABCDEF'	3.14

元胞矩阵（数组）的创建

通过赋值来创建元胞矩阵，2类方法

- 元胞索引(cell indexing)，大括号在=右边

$a(1, 2) = \{\dots \dots \dots\}$

或用大括号把所有元素括起来一起赋值

$a = \{'James Bond', [1\ 2; 3\ 4; 5\ 6]; pi, ones(5)\}$

- 元胞内容索引(content indexing)，大括号在=左边

$a\{1, 2\} = [\dots] \text{ or } '\dots'$

注意括号的区别

元胞矩阵的创建

cell indexing方式创建元胞矩阵

元素下标用 ()
与一般矩阵相同

$a(1,1) = \{[1\ 4\ 3; 0\ 5\ 8; 7\ 2\ 9]\};$

$a(1,2) = \{'Anne\ Smith'\};$

$a(2,1) = \{3+7i\};$

$a(2,2) = \{-\pi:\pi/10:\pi\};$

a

赋值内容
用 {} 包含

结果是 $a=$

$[3 \times 3\ \text{double}]$ $'Anne\ Smith'$

$[3.0000 + 7.0000i]$ $[1 \times 21\ \text{double}]$

$a = \{[1\ 4\ 3; 0\ 5\ 8; 7\ 2\ 9], 'Anne\ Smith'; 3+7i, -\pi:\pi/10:\pi\}$

%此行代码也能得到相同的a

元胞数组的创建

content indexing方式创建元胞矩阵

元素下标用 { }
与一般矩阵不同

b{1,1} = 'James Bond' ;

b{1,2} = [1 2;3 4;5 6];

b{2,1} = pi;

b{2,2} = zeros(5);

b

赋值内容
与一般矩阵
形式相同

结果是 **b =**

'James Bond' [3x2 double]

[3.1416] [5x5 double]

元胞矩阵的创建

通过拼接来创建元胞矩阵

【例】连接元胞矩阵a、b，生成元胞矩阵c。

c=[a b] 这时c是

3x3 double	'Anne Smith'	'James Bond'	3x2 double
3+7i	1x21 double	3.1416	5x5 double

c=[a; b] 这时c是

3x3 double	'Anne Smith'
3+7i	1x21 double
'James Bond'	3x2 double
3.1416	5x5 double

元胞矩阵的创建

cell函数创建元胞矩阵

```
b = cell(2, 3)
```

```
b =
```

```
    []    []    []  
    []    []    []
```

```
b(1,3) = {1:3}; %元胞索引
```

```
b{2,3} = 1:5; %元胞内容索引
```

```
b
```

```
b=    []    []    [1 x 3 double]  
      []    []    [1 x 5 double]
```

显示元胞矩阵的内容

```
a = {[1 4 3; 0 5 8; 7 2 9], 'Anne Smith'; 3+7i, -pi:pi/10:pi} ;
```

a %某些具体内容没有显示出来

a =

[3x3 double] 'Anne Smith'

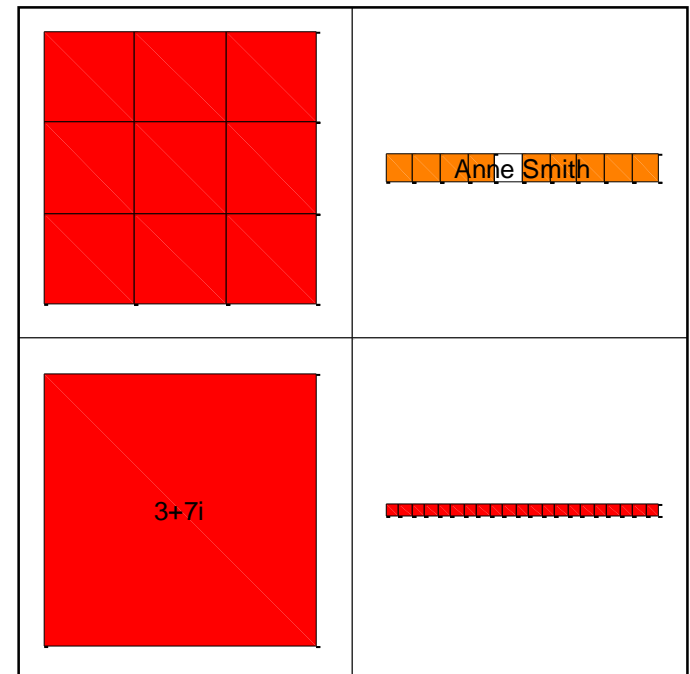
[3.0000 + 7.0000i] [1x21 double]

celldisp(a) %显示详细内容

cellplot(a) %图形方式显示元胞结构

a{2, 2} %通过下标显示某元胞具体内容

a{:} %显示全部元胞内容(比a详细)



元胞矩阵的读取

- 直接取用元胞矩阵的某个单元（元胞）
- 取用元胞矩阵某单元内的数据
- 一次读取或删除多个单元

元胞矩阵的读取

【例】

读取元胞矩阵b的第1行、第2列单元的内容

```
b = {'James Bond',[1 2;3 4;5 6];pi,zeros(5)}
```

```
f1=b{1, 2} %用内容索引
```

b =

'James Bond' [3x2 double]


[3.1416] [5x5 double]

f1 =

1 2

3 4

5 6



1	2
3	4
5	6

f1=b(1, 2) 得到什么结果?

元胞矩阵的读取

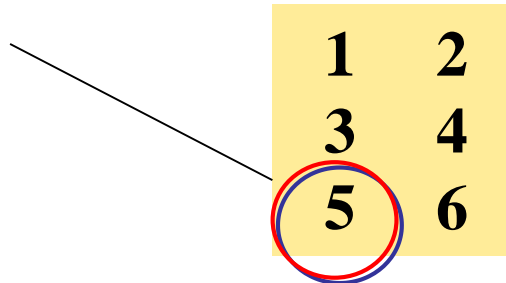
【例】

读取b{1,2}的第3行、第1列的数据。

$$f2 = b\{1,2\}(3,1)$$

f2 =

5



1	2
3	4
5	6

读取单元内的部分数据：

元胞内容索引 + 一般矩阵索引

元胞矩阵的删除

【例】 读取元胞矩阵a第1行的所有元胞。

```
a={ [1 4 3; 0 5 8; 7 2 9], 'Anne Smith'; 3+7i, -pi:pi/10:pi} ;
```

```
f3=a(1,:)
```

```
f3=
```

```
[3x3 double] 'Anne Smith'
```

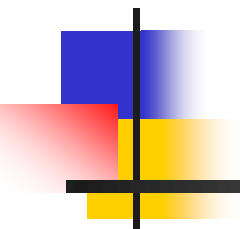
【例】 删除元胞矩阵a第1行

```
a(1,:) = []
```

```
a =
```

```
[3.0000 + 7.0000i] [1x21 double]
```

如果是 `a(1,:)={[]}` , 结果是什么?



结构

结构与结构矩阵（数组）

■ 结构(structure)

- 由字段（或称为域，fields）组成
- 每个字段可以是任一种Matlab数据类型
- 与C语言的结构类型相似

■ 结构数组(structure array)

- 多个结构可形成结构数组(structure array)
- 结构数组的元素就是一个结构

结构矩阵（数组）的创建

■ 方法一：直接赋值

【例】创建关于学生名单的结构矩阵，
包含姓名(name)、学号(id)、成绩(scores)。

```
student.name = '张三';      % 加入 name 字段
student.id = '010012';      % 加入 id 字段
student.scores = [88, 75, 90]; % 加入 scores 字段
student                      % 显示结构变量的数据

student =
    name: '张三'
    id: '010012'
   scores: [88,75,90]
```

此时，Matlab视 student 为一个1x1结构矩阵

结构矩阵的创建

接着，加入第二个学生的信息

```
student(2).name = '李四';    % 加入 name 字段  
student(2).id = '010013';    % 加入 id 字段  
student(2).scores=[68, 85, 92]; % 加入 scores 字段  
student    % 此时细节就不显示了
```

```
student =
```

```
1x2 struct array with fields:
```

```
name
```

```
id
```

```
scores
```

结构矩阵的创建

此时，`student`变量已成为一个1x2的结构矩阵

只是简单输入变量名`student`的话，将不再显示详细的数据信息。

- 只能定位到具体元素才能显示结构的详细数据

`student(1)`

或

`student(1).name`

结构矩阵的创建

■ 方法二：利用struct函数

格式

struct (field1, value1, field2, value2, ...)

field1、field2、...是结构字段名，

value1、value2、...是字段所包含的数据。

结构矩阵的创建

【例】 使用struct创建结构矩阵

结构矩阵包含
姓名(字符)和
分数(浮点矩阵)

```
student=struct('name','张三','scores',[70 60]);  
student(2)=struct('name','李四','scores',[80 70]);  
student(1), student(2) %显示student(1),student(2)
```

```
ans =
```

```
name: '张三'
```

```
scores: [70 60]
```

```
ans =
```

```
name: '李四'
```

```
scores: [80 70]
```

结构矩阵的创建

【例】 用struct创建结构数组变量（一次建立多个元素）

```
student = struct('name', {'张三', '李四'}, ...  
                'scores', {[70 60], [80 70]});  
student(1), student(2)
```

```
ans =
```

```
name: '张三'
```

```
scores: [70 60]
```

```
ans =
```

```
name: '李四'
```

```
scores: [80 70]
```


结构矩阵的嵌套

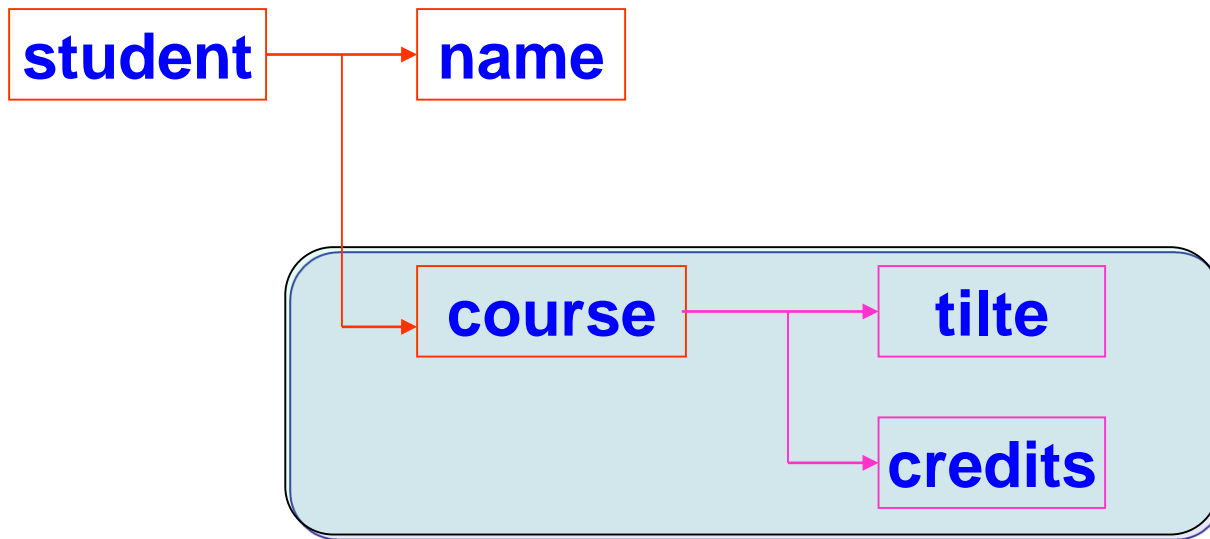
- 结构矩阵中结构的字段还可以是结构。

```
student = struct('name', {'张三', '李四'});  
student(1).course(1).title = 'Fluid Mechanics';  
student(1).course(1).credits = 2;  
student(1).course(2).title = 'Structural Mechanics';  
student(1).course(2).credits = 3;  
student(2).course(1).title = 'Fluid Mechanics';  
student(2).course(1).credits = 2;  
student(2).course(2).title = 'CFD';  
student(2).course(2).credits = 1;  
student(1).course %course是内嵌的另一结构，含两个字段
```

```
ans = 1x2 struct array with fields:  
    title  
    credits
```

结构矩阵的嵌套

以上student结构矩阵的构架层次：



course是内嵌的另一结构矩阵，含两个字段

结构矩阵的内容的寻访和修改

【例】

```
student_name=student(1).name
```

```
student_name=
```

张三

```
student(1).name='Newton'; %定位到字段修改其内容
```

```
student(1).name
```

```
ans=
```

Newton

结构矩阵的排列

- **cat函数: 排列结构矩阵某字段的值**

格式: **cat(dim, structurefield)**

- **dim=1, 竖排; dim=2, 横排**

例

cat(1, student.scores)

ans =

张三

李四

cat(2, student.scores)

ans =

张三 李四

关于元胞和结构的常用函数

函 数	功 能
<code>struct</code>	创建结构数组
<code>struct2cell</code>	结构数组转化为元胞数组
<code>cell2struct</code>	元胞数组转化为结构数组
<code>rmfield</code>	删除结构中的某字段
<code>setfield</code>	设置指定字段的值
<code>isstruct</code>	判断变量是否为结构类型
<code>isfield</code>	判断字段是否存在
<code>getfield</code>	得到指定字段的值
<code>deal</code>	将参数值（如元胞数组）的内容输出为独立变量
<code>fieldnames</code>	得到结构的字段名称

第二章结束

谢谢

第2章要点提示

1. 数据的 `save` `load`
2. 数组/矩阵/向量的多种定义方式(注意 `,` `;` `:` 的灵活运用)
3. 数组/矩阵/向量的多种常用操作, 如 `合并` `置空`等
4. 数组元素的灵活访问和引用
5. 冒号: 在数组中的灵活运用
6. 数组的基本运算, 特别注意`*` 与`.*`的对比
7. 单下标和全下标的相互转换和内在规则
8. 数组/矩阵/向量的其他常用操作 (`max` `min` `mean` 等)
9. 字符串数组的基本定义, 注意字符串中的单引号'如何表示
10. 元胞数组、结构数组的概念和基本定义方式