

课前回顾

第1讲 MATLAB基础概述

内容提要

- 1.1 MATLAB是什么？
 - MATLAB概述
- 1.2 为什么选学MATLAB？
 - MATLAB的功能与特点
- 1.3 MATLAB的安装与激活
- 1.4 MATLAB的工作界面
- 1.5 帮助窗口
- 1.6 MATLAB的文件格式

1.1 MATLAB概述

- MATLAB即**Matrix Laboratory**: “**矩阵实验室**”
- MATLAB语言是由美国的Cleve Moler博士于**1980年**开发的, 设计者初衷是为解决“线性代数”课程的矩阵运算问题
- 1984年Cleve Moler博士等人创立**MathWorks**公司, 并推出第一个MATLAB商业版
- 目前已推出最新的**MATLAB2016a**

1.1 MATLAB概述

- MATLAB语言是**科技界应用最广泛的计算机语言之一**
- 现已不仅是一个“矩阵实验室”了, 而是国际上广泛使用的**集数学运算、符号运算、数据可视化、GUI设计、程序设计、仿真**等诸多功能于一身的**科学与工程计算软件**
- 已成为**线性代数、数值分析、数学建模、信号与系统分析、自动控制、数字信号处理、通信系统仿真**等一大批课程的**教学和科研工具**

1.1 MATLAB概述

- 因其超群的风格与性能风靡全世界, **成功地广泛应用于诸多领域**:
 - 数值分析
 - 数值和符号计算
 - 工程与科学绘图
 - 航空航天工业
 - 汽车工业
 - 数字图像处理
 - 数字信号处理
 - 生物医学工程
 - 通信系统的设计与仿真
 - 控制系统的设计与仿真
 - 财务金融分析
 - 语音处理
 - 图形用户界面设计
 - 新算法研究开发。。。

1.2 MATLAB的功能与特点

- 强大的科学（数值、矩阵、符号）计算功能
- 先进的可视化/绘图工具
- 强大的图形化程序编制功能
- 丰富的面向领域应用的工具箱和模块集
- 开放性、可扩展性强、兼容性强
- 强大的联机检索功能
- 直观灵活的语言：简单易学

1.2 MATLAB的功能与特点

- 强大的科学（数值、矩阵、符号）计算功能

- 矩阵运算

- ◆ MATLAB提供了丰富的矩阵运算处理功能，是基于矩阵运算的处理工具。

- ◆ 变量——矩阵，运算——矩阵的运算

- ◆ 例如 $C = A + B$ ， A, B, C 都是矩阵,是矩阵的加运算

- ◆ 即使一个常数 $Y=5$ ，MATLAB也看做是一个 1×1 的矩阵

1.2 MATLAB的功能与特点

- 强大的科学（数值、矩阵、符号）计算功能

- 数值运算-示例-求解线性系统

$$\begin{cases} 3x_1 + x_2 - x_3 = 3.6 \\ x_1 + 2x_2 + 4x_3 = 2.1 \\ -x_1 + 4x_2 + 5x_3 = -1.4 \end{cases}$$

对于线性系统有 $Ax=b$

```
>>A=[3 1 -1;1 2 4;-1 4 5];b=[3.6;2.1;-1.4];
```

```
>>x=A\b x = [ 1.4818 -0.4606 0.3848]'
```

1.2 MATLAB的功能与特点

- 强大的科学（数值、矩阵、符号）计算功能

- 符号运算

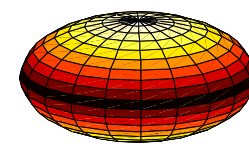
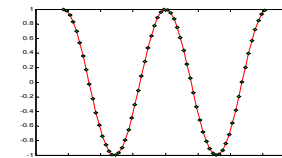
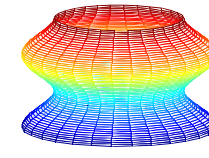
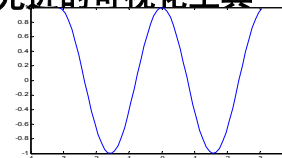
- ◆ 符号运算即用字符串进行数学分析

- ◆ 允许变量不赋值而参与运算

- ◆ 用于解代数方程、微积分、复合导数、积分、二重积分、有理函数、微分方程、泰勒级数展开、寻优等等，可求得解析符号解

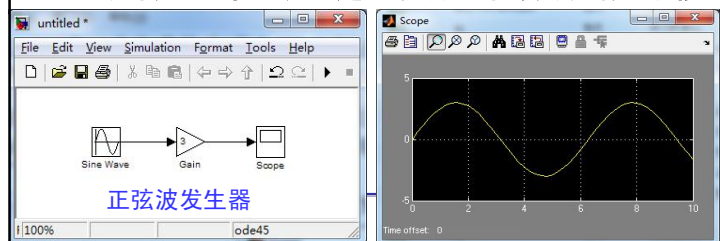
1.2 MATLAB的功能与特点

- 先进的可视化工具



1.2 MATLAB的功能与特点

- 强大的图形化程序编制功能
 - 动态系统进行建模、仿真和分析的软件包
 - 用结构图编程，而不用程序编程
 - 只需拖几个方块、连几条线，即可实现编程功能



1.2 MATLAB的功能与特点

- 开放性、可扩展性强
 - MATLAB所有函数都是开放的
 - 用户可按自己意愿随意更改
 - 正因为此功能，使得MATLAB的应用越来越广泛
- 强大的联机检索帮助系统
 - 可随时检索MATLAB函数
 - 可随时查询MATLAB函数的使用方法

1.2 MATLAB的功能与特点

- 丰富的面向领域应用的工具箱和模块集 **50+**
 - MATLAB主工具箱
 - 通信工具箱
 - 符号数学工具箱
 - 系统辨识工具箱
 - SIMULINK仿真工具箱
 - 神经网络工具箱
 - 控制系统工具箱
 - 金融工具箱
 - 信号处理工具箱
 - 优化工具箱
 - 图像处理工具箱
 - 曲线拟合工具箱

1.2 MATLAB的功能与特点

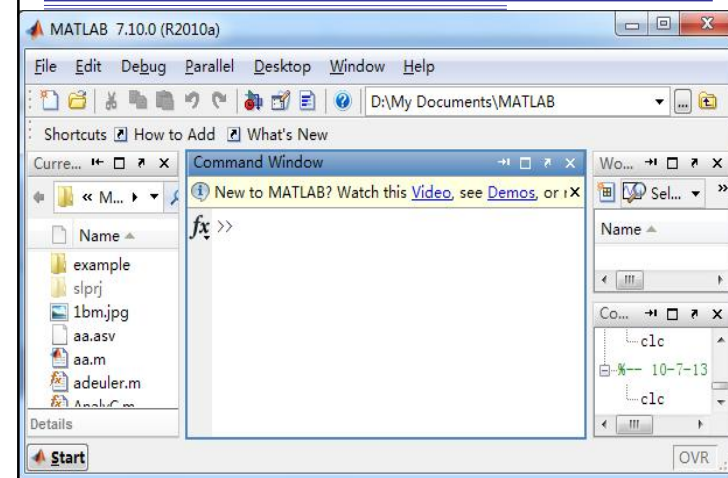
- 直观灵活的语言：简单易学
 - MATLAB是一种直译式的高级语言，符合人类逻辑，比其它程序设计语言容易-伪代码
 - MATLAB语言与其它语言的关系仿佛和C语言与汇编语言的关系一样
 - 标志着计算机语言向“智能化”方向发展，被称为**第四代编程语言**

1.3 MATLAB的安装与激活

- (5) 若输入正确的密钥，系统将弹出如下图所示的 Installation Type对话框。若选中custom按钮，将按需选择所需工具箱，如下图所示（推荐）。



1.4 MATLAB的工作界面



1.5.2 命令窗口查询帮助系统

- help系列:
 - help、help+函数（类）名、helpwin及helpdesk
 - help %显示help 主题一览表
 - help plotxyz %显示有关三维绘图指令帮助信息
- lookfor命令
 - 查找不确切名称函数时，lookfor命令可根据用户提供的关键字，搜索出一组与之相关的命令
 - lookfor fourier %寻找含有傅立叶变换的相关指令

1.6 MATLAB的文件格式

- M文件: 以.m为扩展名
- 数据文件: 以.mat为扩展名
- 图形文件: 以.fig为扩展名
- MEX 文件: 以.mex或.dll为扩展名
- 模型文件: 以.mdl为扩展名
- 仿真文件: 以.s扩展名

小结

- MATLAB是一个优秀的软件
 - 易用性与可靠性相结合
 - 通用性与专业性相结合
 - 一般目的的应用与高深的科学技术应用相结合

好好学习、学以致用

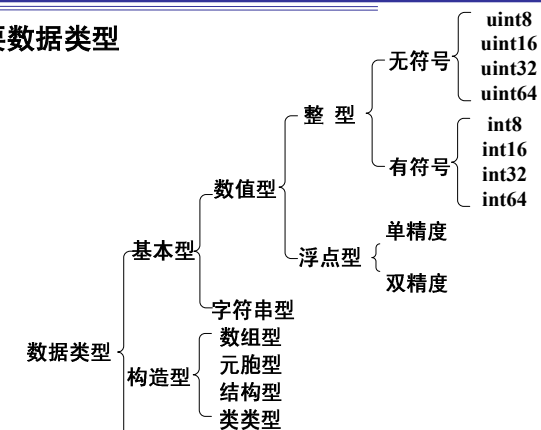
第2章 MATLAB的数据类型

MATLAB的数据类型

- Matlab的基本数据存储类型有**十几种**，不同专业的工具箱中还有特殊的数据类型，且还支持用户自定义数据类型
- Matlab数据类型的最大特点是每一种类型都是以**数组为基础**，把每种类型的数据都作为数组来处理
- Matlab的默认数据存储类型为**双精度浮点类型(double)**，可以利用转化函数存储为其它类型
- 特点：**不用事先对变量的类型进行定义或说明**，系统会根据变量被赋值的类型自动进行类型识别

MATLAB的数据类型

• 主要数据类型



常量

- 常量：MATLAB中预定义的变量，这些特殊变量称为常量
- 常用系统内部常量

常量符号	常量含义
ans	默认变量名，未赋值运算结果自动赋给ans
eps	容差变量,小于该值时,可认为0,在pc机上为 2^{-52}
realmax	最大浮点数, 2^{1023}
realmin	最小浮点数, 2^{-1022}
pi	圆周率 π 的双精度近似值
i, j	虚数单位, 定义为 $i^2=j^2=-1$
inf	表示正无穷大, 由0做除数引入此常量
NaN	不合法数值, 非数值, 产生于 $0 \times \infty$, $0/0$ 等运算
computer	计算机类型, 如PCWIN
version	MATLAB版本字符串, 如8.2.0.701 (R2013b)

常量

例：系统内部常量的调用与赋值。

```
>> clear all;    %首先清除用户自定义的变量
>> pi           %查看系统的内部常量pi
ans =
    3.1416
>> pi=ans+1.5    %用户重新赋值pi, 查看效果
pi =
    4.6416
```

- 在无指定输出变量时，系统自动把值赋给了ans
- 通过对pi进行重新赋值，默认值（pi）被临时覆盖
- 清空内存后，pi又恢复到系统默认值

变量

- 变量是任何程序设计语言的基本元素之一
- Matlab 并不要求对所使用的变量进行事先声明，也不需要指定变量类型，Matlab 会自动根据所赋予变量的值或对变量所进行的操作来确定变量的类型
- Matlab中所有变量都以数组或者矩阵形式保存
- 在赋值过程中，如果变量已经存在，Matlab 会用新值代替旧值，并以新的变量类型代替旧的变量类型

变量

- 命名规则
 - 可由英文字母、数字和下划线混合组成
 - 第一个字符必须是一个英文字母，最多可包括63个字符
 - 变量名不得包含空格和标点，但可以有下连字符
 - 不能使用函数名、系统保留字
 - 变量名和函数名对字母的大小写敏感：
 - x_1 , α , x_1 , \dots
 - X , x , Z , z 是不同变量

变量查询函数

- 变量查询函数: who 和 whos
- 作用: 都是列出在matlab工作空间中已经驻留的变量名清单
- 不同: whos在给出驻留变量同时, 还给出他们的维数及性质

• 例:

```
>> a=3; b=pi;
```

```
>> who
```

Your variables are:

a b

```
>> whos
```

Name	Size	Bytes	Class	Attributes
a	1x1	8	double	
b	1x1	8	double	

变量删除和内存重组命令

- **clear**: 删除所有自定义变量并恢复除eps外所有预定义变量
- **clear variable1**: 仅删除名为变量variable1的变量
- **clear variable1 variable2 ...**: 删除变量variable1 variable2...
- **clear a***: 删除所有以a开头的变量
- **pack**: 重组和压缩已分配的内存碎片. 当Matlab 的内存满后, 可以利用这个命令实现在不清除任何变量的情况下得到更多的空间
- **pack filename**: 用名为filename的文件作为临时文件, 重组和压缩已分配的内存

变量类型

- **局部变量 (Local)**: 局部变量是在函数中定义的变量。每个函数都有自己的局部变量, 只能被定义它的函数访问。当函数运行时, 其变量保存在自己的工作区中, 一旦函数退出运行, 内存中变量将不复存在
- **全局变量 (Global)**: 多个函数可共享的变量; 使用global函数定义

```
>> clear all; %清除原先用户自定义的变量
>> global MAXLEN; MAXLEN=45;
```
- **永久变量 (Persistent)**: 类似于static变量; 在M文件中定义和使用, 永久保存, 只有在clear时被清除; 使用persistent函数定义

```
>> persistent MAXLEN; MAXLEN=45;
```

数值型变量

- 数值型数据包括
 - 整型变量: 带符号和无符号 (1/2/4/8字节)
 - 浮点型变量: 单精度 (4字节) 和双精度 (8字节)
- Matlab进行数值计算时, 默认所有变量都为双精度的浮点数(double)
- 使用整型变量和单精度变量可以节约内存空间

整型变量

- Matlab中有4种有符号整数类型，4种无符号整数类型

数据类型	取值范围	转换函数
带符号 8 位整型	$[-2^7, 2^7 - 1]$	int8
带符号 16 位整型	$[-2^{15}, 2^{15} - 1]$	int16
带符号 32 位整型	$[-2^{31}, 2^{31} - 1]$	int32
带符号 64 位整型	$[-2^{63}, 2^{63} - 1]$	int64
无符号 8 位整型	$[0, 2^8 - 1]$	uint8
无符号 16 位整型	$[0, 2^{16} - 1]$	uint16
无符号 32 位整型	$[0, 2^{32} - 1]$	uint32
无符号 64 位整型	$[0, 2^{64} - 1]$	uint64

整型变量

- 类型转换：(默认存储类型为double，利用类型转化函数可将数据存储为整型，如有小数部分，四舍五入处理)

```

x1=int8(11);      x1=11
x2=int8(15.49);   x2=15
x3=int8(15.5);    x3=16
x4=int8(130);     x4=127
x5=uint8(255);    x5=255
x6=uint8(256);    x6=255
x7=uint8(-1);     x7=0
    
```

整型变量

- 最大整数与最小整数
 - 可以用 `intmax` 和 `intmin` 函数来查询不同整型所能表示的最大整数和最小整数

– 示例：

```

>>intmax('int8')      ans = 127
>>intmin('int8')      ans = -128
>>intmax('uint8')     ans = 0
>>intmin('uint8')     ans = 255
>>intmax intmax('int32') ans = 2147483647
>>intmin intmin('int32') ans = -2147483648
    
```

浮点型变量

- 浮点型：
 - 单精度(single): 32位(4字节)存储空间
 - 双精度(double): 64位(8字节)存储空间
- 最大浮点数与最小浮点数
 - 可以用 `realmax`和 `realmin`函数来查询不同浮点数所能表示的最大浮点数和最小浮点数

```

>>realmax('single');  ans = 3.4028e+38
>>realmin('single');  ans = 1.1755e-38
>>realmax; realmax('double ') ans = 1.7977e+308
>>realmin; realmin('double ') ans = 2.2251e-308
    
```

浮点型变量

• 浮点数转换函数

- single: 将数据转换成单精度浮点数
- double: 将数据转换成双精度浮点数
- 注: 直接输入变量值创建的是double类型, 创建single类型需要用输入类型转化函数
- 例:

```
>>x=35; y= single(x);  
>>a=class(x), b=class(y)  
a = double  
b = single
```

浮点型变量

• 浮点数的取整函数

函数	含义	示例 X=[-1.5, 1.5]
round(X)	round the elements of X towards nearest integer	[-2, 2]
fix(X)	rounds the elements of X to the nearest integers towards zero	[-1, 1]
floor(X)	rounds the elements of X to the nearest integers towards minus infinity	[-2, 1]
ceil(X)	rounds the elements of X to the nearest integers towards positive infinity	[-1, 2]

数值型变量

• 输出格式控制: format

示例	说明	输出
format (short)	短格式, 小数点后保留4位有效数字; 大于1000时: 科学计数法(5位)	314.159 314.1590 3141.59 3.1416e+03
format long	长格式, 15位数字表示	3.14159265358979
format short e	短格式e, 5位科学计数法表示	3.1416e+000
format long e	长格式e, 15位科学计数法表示	3.141592653589793e+000
format rat	分数(近似有理数表示)格式	355/113
format hex	十六进制表示	400921fb54442d18
format +	显示大矩阵用, 正数、负数、零分别用+、-、空格表示	+
format bank	银行格式(圆角分、两位小数)	3.14

复数

- Matlab表示复数时, 可用字母i或j表示虚部(等价)
- 创建复数可直接输入或利用函数complex(a,b)

```
>> 2-3i    %虚部与虚数单位之间不能留空格
```

```
ans =  
2.0000 - 3.0000i
```

```
>>a=2; b=-3; complex(a,b)
```

```
ans =  
2.0000 - 3.0000i
```

```
>>a=2-3i; b=2-3j; a-b
```

```
ans =  
0
```

z=complex(3)

z=complex(3,0)

逻辑型变量

- 有两种取值，逻辑真和逻辑假，用“1”和“0”表示

- 创建逻辑数组

- 通过输入“true”或“false”直接创建逻辑数组
- 通过对数组进行逻辑运算创建
- 通过MATLAB函数产生逻辑数组
- 示例:

```
>> x = [true, false, true]
```

```
x = 1 0 1
```

```
>> y = [7, 2, 9] > 5
```

```
y = 1 0 1
```

逻辑型变量

- 运算产生逻辑结果的函数

函数	操作
true、false	设值为真或假
logical	数值类型转化为逻辑类型
&(and)、 (or)、~(not)、xor、any、all	逻辑运算
&&、	与、或
==、~=、<、>、<=、>=	关系运算
is*、cellfun	测试运算
strcmp、strncmp、strcmpi、strncmpi	字符串比较

逻辑型变量

- 逻辑数组:主要用在条件语句和数组索引中

- 示例:

```
>> A=rand(3), A(A<0.5)=0
```

```
A =
```

```
0.9649 0.9572 0.1419
```

```
0.1576 0.4854 0.4218
```

```
0.9706 0.8003 0.9157
```

```
A =
```

```
0.9649 0.9572 0
```

```
0 0 0
```

```
0.9706 0.8003 0.9157
```

数组和矩阵

- Matlab中所有变量都以数组或者矩阵形式保存的

- 数组中的元素可以是字符等；矩阵中的只能是数

- 数组的运算：指数组对应元素间的运算，即点运算

- 矩阵的运算：

- 是一个二维数组，所以矩阵的加、减、数乘等运算与数组运算是一致

- 但矩阵的乘法、乘方和除法有特殊的数学含义，并不是数组对应元素的运算

数组和矩阵

• 创建数组或矩阵-直接构造法

- 又称逐个元素输入法，最直接也最简单
 - 必须以方括号[]作为起始和结束标志
 - 行内：用空格或逗号“,”间隔数组元素
 - 行间：用分号“;”或回车分开
 - 大小：不必预先定义，且元素可为值或表达式
- 适合元素较少的简单矩阵

数组和矩阵

• 创建数组或矩阵-直接构造法-示例

```
>> clear all;
>> a=[2 3.8 1/3 1e-3 sqrt(2)] %逐个元素输入构造数组
a = 2.0000 3.8000 0.3333 0.0010 1.4142
>> A=[1 2 3;4 5 6;7 8 9]
A = 1 2 3
    4 5 6
    7 8 9
>> B=[true,7,9; 3,2,8; 5,1,4/2]
B = 1 7 9
    3 2 8
    5 1 2
```

数组和矩阵

• 创建数组或矩阵-增量法

- 利用冒号生成向量
 - $x = i:j$ %若 $i < j$, 生成以 i 为初值, j 为终值, 1 为步长的等差数列
 - $x = i:j:k$ %若 $i < k$, 生成以 i 为初值, k 为终值, j 为步长的等差数列
 - $x = i:-j:k$ %若 $i > j$, 生成以 i 为初值, k 为终值, $-j$ 为步长的等差数列
- 例: `>> A=10:15`

```
A = 10 11 12 13 14 15
```

```
>> A=1:2:10
```

```
A = 1 3 5 7 9
```

数组和矩阵

• 创建数组或矩阵-增量法

- 利用 `linspace(a,b)` 生成等差向量
 - `x = linspace(2,8)` %生成100个数,以2开始,以8结束
 - `y=linspace(2,8,10)` %生成10个数,以2开始,以8结束
- 利用 `logspace(a,b)` 生成等比向量
 - `x = logspace(2,8)` %生成50个数,以 10^2 开始,以 10^8 结束
 - `y = logspace(2,8,10)` %生成10个数,以 10^2 开始,以 10^8 结束

数组和矩阵

- 创建大矩阵：可由方括号中的小矩阵来建立

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

```
>> C=[A, eye(size(A)); ones(3), A*2]
```

```
C = 1   2   3   1   0   0
     4   5   6   0   1   0
     7   8   9   0   0   1
     1   1   1   2   4   6
     1   1   1   8  10  12
     1   1   1  14  16  18
```

其中eye(3)返回3×3单位矩阵，ones(3)返回3×3全1矩阵

数组和矩阵

- 常用创建特殊矩阵函数

指令	含义
zeros	产生全0数组
ones	产生全1数组
eye	产生单位数组（对高维不适用）
magic	产生魔方数组（对高维不适用）
rand	产生均匀分布随机数组
randn	产生正态分布随机数组
diag	产生对角形数组（对高维不适用）
size()	返回指定矩阵的行数和列数

数组和矩阵

- 矩阵元素：

- MATLAB允许对矩阵的单个元素进行赋值和操作，而不影响其他元素的值。如果给出的行下标或列下标大于原来矩阵的行或列，则MATLAB将自动扩展原来的矩阵，并将扩展后未赋值的矩阵元素置为0。

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

```
>> A(4,3)=99 %将矩阵A的第4行第3列的元素赋值99
```

```
A = 1   2   3
     3   4   5
     7   8   9
     0   0  99
```

MATLAB运算符

- 算术运算法

操作符	功能说明
+	加，相同
-	减，相同
*	矩阵乘， 矩阵乘法规则，内维相同矩阵相乘
.*	数组乘， 对应元素相乘
^	矩阵乘方， 矩阵乘方规则（方阵）
.^	数组乘方， 对应元素相乘
\	矩阵左除， a\b表示矩阵a的逆乘以b
.\	数组左除(除)， a.\b=b./a
/	矩阵右除， a/b表示矩阵a乘以矩阵b的逆
./	数组右除(除以)： a/b
'	矩阵转置， 共轭转置(针对非复矩阵与.'相同)
.'	数组转置， 非共轭转置

MATLAB运算符

• 关系运算符

操作符	功能说明
==	等于
~=	不等于
>	大于
<	小于
>=	大于等于
<=	小于等于

MATLAB运算符

• 逻辑运算符:

逻辑运算符	逻辑运算	说明
&	And	逻辑与
	Or	逻辑或
~	Not	逻辑非
	Xor	逻辑异或
	any(x)	如果在一个向量x中, 任何元素是非零, 返回1; 矩阵x中的每一列有非零元素, 返回1
	all(x)	如果在一个向量x中, 所有元素非零, 返回1; 矩阵x中的每一列所有元素非零, 返回1

MATLAB运算符

• 各种运算符的优先级 :

优先级	运算符
最高	()(小括号)
↓	.(转置) '(共轭转置) .^ (数组乘方) ^ (矩阵乘方)
↓	+(一元加法) -(一元减法) ~(取反)
↓	.*(乘法) *(矩阵乘法) ./ (右除) /(矩阵右除) .\ (左除) \ (矩阵左除)
↓	+(加法) -(减法)
↓	:(冒号)
↓	< (小于) <= (小于或等于) > (大于) >= (大于或等于) == (等于) ~= (不等于)
↓	& (逻辑与)
最低	(逻辑或)

数组和矩阵

• 矩阵的拆分-利用冒号表达式获得子矩阵

指令格式	指令功能
$A(r,c)$	数组A中r指定行、c指定列之元素组成的子数组
$A(r,:)$	数组A中r指定行对应的所有列之元素组成的子数组
$A(:,c)$	数组A中c指定列对应的所有行之元素组成的子数组
$A(:)$	数组A中各列元素首尾相连组成的“一维长列”子数组
$A(i)$	“一维长列”子数组中的第i个元素
$A(r,c)=Sa$	数组A中r指定行、c指定列之元素组成的子数组的赋值
$A(:)=D(:)$	数组全元素赋值, 保持A的行宽、列长不变, A、D两组元素总合应相同

数组和矩阵

• 矩阵的拆分-利用空矩阵删除元素

- MATLAB中定义[]为空矩阵；给变量X赋空矩阵：X=[]
- 注意，X=[]与clear X不同：clear是将X从工作空间中清除，而空矩阵则存在于工作空间，只是维数为0
- 将某些元素从矩阵中删除，可采用将其置为空矩阵的方法就是一种有效的方法。如：

```
>> A=[1 4 7 2;5 8 3 6;9 11 15 19]    >> A(:,3:4)=[]
```

A =	1	4	7	2
	5	8	3	6
	9	11	15	19

A =	1	4
	5	8
	9	11

数组和矩阵

• 常用操作函数

指令	含义
diag	提取对角线元素，或生成对角阵
flipud	以数组“水平中线”为对称轴，交换上下对称位置上的数组元素
fliplr	以数组“垂直中线”为对称轴，交换左右对称位置上的数组元素
reshape	在总元素数不变的前提下，改变数组的“行数、列数”
rot90	矩阵逆时针旋转90度
det	方阵的行列式值
rank	矩阵的秩

数组和矩阵

• 常用操作函数

- >> B= magic(3) +1 %矩阵加法 >> C=A*3 %矩阵乘法

B =	9	2	7
	4	6	8
	5	10	3

B =	24	3	18
	9	15	21
	12	27	6

- >> C=flipud(B) %上下翻转 >> C=rot90(B) %逆旋转90度

C =	5	10	3
	4	6	8
	9	2	7

C =	7	8	3
	2	6	10
	9	4	5

数组和矩阵

• 常用数组操作函数

- >> C=reshape(B, 1,9)

C =	9	4	5	2	6	10	7	8	3
-----	---	---	---	---	---	----	---	---	---

```
>> C=reshape(c, 3,3)
```

C =	9	2	7
	4	6	8
	5	10	3

数组和矩阵

• 矩阵求值-方阵的行列式:

– $d = \det(X)$: 求方阵X所对应的行列式的值, 返回d

`>> A=magic(4) %创建一个4阶魔方矩阵`

A =

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

`>> B=det(A)`

B = -1/689889648801

数组和矩阵

• 矩阵求值-矩阵的特征值:

– MATLAB提供了eig函数实现原本复杂的矩阵特征值求解

– $d = \text{eig}(A)$: 求矩阵A的特征值, 构成向量d

– $d = \text{eig}(A,B)$: 如果矩阵A和B都为一方阵, 则返回一个广义的特征值

– $[V,D] = \text{eig}(A)$: 求矩阵A的全部特征值, 构成对角矩阵D, 并求A的特征向量构成V的列向量

• 矩阵的秩: $b = \text{rank}(A)$

• 矩阵(方阵)的迹: $b = \text{trace}(A)$

• 矩阵的(伪)逆: $b = (\text{p})\text{inv}(A)$

数组和矩阵

• 矩阵分解:

– 三角分解(可逆方阵)

• 将一个矩阵分解成一个下三角矩阵L和一个上三角矩阵U的乘积, 即 $A=LU$

– $[L,U] = \text{lu}(A)$

• U为上三角阵, L为下三角阵或其变换形式, 满足 $LU=A$

– 正交分解/QR分解(满秩矩阵)

• 将矩阵A分解成一个正交矩阵Q($QQ'=E$)与一个上三角矩阵R的乘积 $A=QR$

– $[Q,R] = \text{qr}(A)$

– $[Q,R,E] = \text{qr}(A)$

数组和矩阵

• 矩阵分解:

– 特征值分解EVD(任意矩阵)

• 任意一个n阶方阵X可以分解为 $XV=VD$, 其中D为X的特征值对角阵, V为X的特征向量矩阵

• $[V,D] = \text{eig}(X)$

• $[V,D] = \text{eig}(X,Y)$ %计算广义特征值矩阵D和广义特征向量矩阵V, 使得 $XV=YVD$

– 奇异值分解SVD(任意矩阵)

• 任意一个 $m \times n$ 维的矩阵X可以分解为 $X=USV'$, U, V均为酉矩阵(复数矩阵 $UU'=U'U=E$), S为 $m \times n$ 维的对角矩阵, 其对角线元素为X的从大到小排序的非负奇异值

• $[U,S,V] = \text{svd}(X)$

数组和矩阵

- 矩阵分解:
 - Cholesky分解
 - 如果A为n阶对称正定矩阵($z'Az > 0$), 则存在一个实的非奇异上三角阵R, 满足 $R'*R=A$, 称为Cholesky分解
 - $R = \text{chol}(A)$
若A非正定, 则产生错误信息。
 - $[R,p] = \text{chol}(A)$
不产生任何错误信息。若A正定, 则 $p=0$, R与上相同; 若A非正定, 则p为正整数, R是有序的上三角阵

字符串

- 字符串: 即一维字符数组
 - 所有的字符串都用单引号括起来
 - 字符串的每一个字符都是该字符串中的一个元素
 - 可通过使用下标访问数组元素的方式对字符串中的每个字符进行访问
 - 字符串中字符以ASCII码形式存储, 因而大小写有区别

字符串

- 创建字符串-单引号赋值
 - 创建字符串最简单的方法是用单引号赋值
- ```
>> c1='Beihang';
>> whos c1
```
- | Name | Size | Bytes | Class | Attributes |
|------|------|-------|-------|------------|
| c1   | 1x7  | 14    | char  |            |
- ```
>> c2=c1(1)  
c2 = B  
>> c3=('A'=='a')  
c3 = 0
```

字符串

- 创建字符串-字符串的转换
 - 在MATLAB中, 可以将其他类型数据转换为字符串的常用函数有char、num2str、int2str、mat2str。这些函数可以把数字、矩阵等转换成字符串
- ```
>> c1=char(97)
c1 = a
>> c2=num2str(97); class(c2)
ans = char
>> c3=int2str(97); class(c2)
ans = char
```

## 字符串

- 字符串的合并-水平合并：得到一个更长的字符串
  - 直接使用中括号:在中括号中用**逗号**实现水平合并

```
>> str1=['I ','Love ','Matlab!']
```

```
str1 = I Love Matlab!
```
  - 使用 **strcat** 函数
    - 使用**strcat**时，自动去除原字符串**结尾处的空格**

```
>> str2=strcat('I ','Love ','Matlab!')
```

```
str2 = ILoveMatlab!
```

```
>> str2=strcat('I ','Love ','Matlab!')
```

```
str2 = ILoveMatlab!
```

```
>> str2=strcat('I ',' ' Love ',' Matlab!')
```

```
str2 = I Love Matlab!
```

## 字符串

- 字符串的合并-垂直合并：得到一个**二维字符数组**
  - 直接使用中括号：在中括号中用**分号**实现垂直合并，**必须保证每个字符串的长度相等**，否则需用空格补齐

```
>> str3=['Matrix ';'Laboratory']
```

```
str3 =
Matrix
Laboratory
```

补3个空格，使长度相等
  - 使用 **strvcat** 函数
    - 使用**strcat**时，自动去除原字符串**结尾处的空格**

```
>> str4=strvcat('Matrix ','Laboratory')
```

```
str4 = Matrix
Laboratory
```

## 字符串

- 字符串长度
  - Matlab中**length()**和**size()**可测量字符串、数组或矩阵大小
  - **length()**: 测量各维中最大维的大小
  - **size()**: 以向量形式给出各维的大小
  - 关系:  $\text{length}() = \max(\text{size}())$ 

```
>> c1='I love Matlab!'
```

```
c1 = I love Matlab!
```

```
>> size(c1)
```

```
ans = 1 14
```

```
>> length(c1)
```

```
ans = 14
```

## 字符串

- 字符串的元素操作函数

| 函数                             | 描述                          |
|--------------------------------|-----------------------------|
| cellstr                        | 将字符数组转化为字符串单元数组             |
| char                           | 将字符串单元数组转化为字符数组             |
| iscellstr/isspace/<br>isletter | 判断是否为字符串单元数组/空格/字母          |
| sort                           | 元素排序                        |
| strcat                         | 串接字符串                       |
| strmatch                       | 查找匹配字符串                     |
| findstr                        | 在一个字符串中查找另一字符串              |
| strrep                         | 替换字符串                       |
| strcmp                         | 比较字符串                       |
| strcmpi                        | 忽略大小写比较字符串                  |
| strncmp                        | 比较字符串的前n个字符                 |
| upper、lower                    | 转换为大写、转换为小写                 |
| blanks、deblank                 | 产生空字符串、移除字符串尾部的空格           |
| strtok                         | 返回字符串中第一个分隔（空格，回车和Tab键）前的部分 |

## 字符串

### • 字符串的字符分类

- isletter(c1)是否为字母; isspace(c1) 是否是空格
- 通用: isstrprop('str', 'category')

| Category | Description                                                                          |
|----------|--------------------------------------------------------------------------------------|
| alpha    | True for those elements of str that are alphabetic                                   |
| alphanum | True for those elements of str that are alphanumeric                                 |
| cntrl    | True for those elements of str that are control characters (for example, char(0:20)) |
| digit    | True for those elements of str that are numeric digits                               |
| lower    | True for those elements of str that are lowercase letters                            |
| print    | True for those elements of str that are graphic characters, plus char(32)            |
| punct    | True for those elements of str that are punctuation characters                       |
| upper    | True for those elements of str that are uppercase letters                            |
| xdigit   | True for those elements of str that are valid hexadecimal digits                     |

## 字符串

### • 常见字符串转换函数

| 函 数      | 功 能               | 函 数     | 功 能               |
|----------|-------------------|---------|-------------------|
| hex2dec  | 将16进制字符串转化为10进制整数 | dec2hex | 将10进制整数转化为16进制字符串 |
| bin2dec  | 将2进制字符串转化为10进制整数  | dec2bin | 将10进制整数转化为2进制字符串  |
| base2dec | 转化B底字符串为10进制整数    | hex2num | 将16进制字符串转化为双精度数   |
| upper    | 改该字符串为大写          | lower   | 改该字符串为小写          |
| fprintf  | 把格式化的文本写到文件中或显示屏上 | sprintf | 用格式控制, 数字转换成字符串   |
| sscanf   | 用格式控制, 字符串转换成数字   | char    | ASCII码转换成字符串      |
| num2str  | 数字转换成字符串          | int2str | 整数转换成字符串          |

## 字符串

### • 字符串的元素操作函数-举例-用户名密码注册验证

```
>>ID='zhangsan'; pw='123abc';
>>isstrprop(pw, 'alphanum') %判断pw是否为字母或数组
ans = 1 1 1 1 1 1
>>l=length(pw); l>=6 %判断pw是否大于等于6位
ans = 1
>>strcmp(ID, pw); %判断ID和pw是否相同
ans = 0
>>findstr(ID, pw) %判断pw是否为ID的一部分
ans = []
>> disp('恭喜, 此用户名密码合法, 注册成功!') %提示
恭喜, 此用户名密码合法, 注册成功!
```

## 函数句柄

### • 函数句柄:提供间接调用函数方法的MATLAB数据类型

### • 创建和调用函数句柄

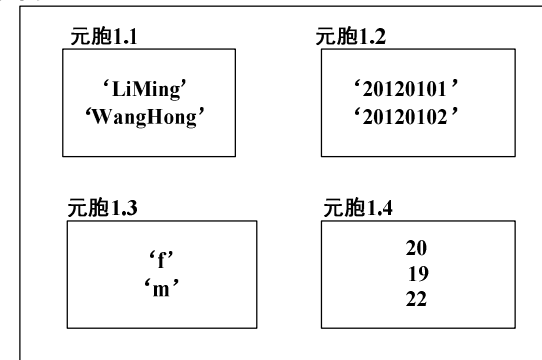
- 在已有函数名前加符号@, 即可创建函数句柄
  - 例如, 存在已有函数, 名为functionname
  - 创建一个函数句柄fun1: fun1=@functionname
- 可提供匿名函数创建一个函数句柄
  - 例如, 创建一个计算x平方的函数sqr=@(x)x.^2
- 利用句柄调用函数: 将函数句柄看作函数名, 调用格式: fun1(arg1, arg2, ..., argn)
- 示例: >> sqr=@(x)x.^2; a=sqr(5) a = 25

## 元胞数组

- 元胞数组是Matlab特有的一种数据类型
- 组成它的元素是元胞(cell); **元胞可用来存储不同类型数据的单元**; 每个元胞存储一种类型的Matlab数组, 此数组中的数据可以是任何一种Matlab数据类型或用户自定义的类型, 其大小也可以是任意的
- 元胞数组可以将**不同类型或不同尺寸**的数据存储到**同一个数组**当中
- 访问单元数组的方法与矩阵索引方法基本相同, 区别在于单元数组索引时, 需要用{}将下标置于其中

## 元胞数组

### • 示例



## 元胞数组

- 创建元胞数组:
  - 创建元胞数组与创建一般数组基本相同
  - **区别在于一般数组用[], 元胞数组用{}**
- 示例:

```
>> a={2 4 7;3 9 6;1 8 5}, 'Li Si',2+3i,1:2:10
a =
[3x3 double] 'Li Si' [2.0000 + 3.0000i] [1x5 double]

>> b{1,1}=[2 4 7;3 9 6;1 8 5]; b{1,2}='Li Si';
>> b{1,3}=[2 + 3i]; b{1,4}=1:2:10
b =
[3x3 double] 'Li Si' [2.0000 + 3.0000i] [1x5 double]
```

## 元胞数组

- 创建元胞数组:
  - 利用cell函数创建元胞数组

```
>> a=cell(3,3)
a =
[] [] []
[] [] []
[] [] []

>> a{3,2}=rand(3,5); a{1,1}='hello; world'
a =
'hello; world' [] []
[] [] []
[] [3x5 double] []
```

## 元胞数组

### • 元胞数组的操作：提取元胞数组内容

- 可利用**元素地址**访问元胞；也可用**deal函数**查看元胞内容

```
>> a={2 4 7;3 9 6;1 8 5}, 'Li Si'; 2+3i,1:2:10}
>> a{1,2} %查看元胞数组第1行第2列的元胞元素
ans = Li Si
>> [a b c d]=deal(a{:}) %用deal函数查看元胞数组内容
a = 2 4 7
 3 9 6
 1 8 5
b = 2.0000 + 3.0000i
c = Li Si
d = 1 3 5 7 9
```

## 元胞数组

### • 元胞数组的操作：元胞数组的删除

- 对单元数组向量下标赋空值即可删除单元数组的行或列

```
>> a={2 4 7;3 9 6;1 8 5}, 'Li Si'; 2+3i,1:2:10}
a = [3x3 double] 'Li Si'
 [2.0000 + 3.0000i] [1x5 double]
>> a{1,2}=[] %删除元胞数组第1行第2列的元胞元素
a = [3x3 double] []
 [2.0000 + 3.0000i] [1x5 double]
>> a(:,2)=[] %删除元胞数组的第2列
a = [3x3 double]
 [2.0000 + 3.0000i]
```

## 元胞数组

### • 元胞数组的操作：元胞数组的转换

- num2cell函数

- **C = num2cell(A)**: 把多维数组A转换成元胞数组C。C的行数、列数与A相同
- **C = num2cell(A, dim)**: 把多维数组A转换成元胞数组，指定dim维作为单独元胞

- mat2cell函数

- **c = mat2cell(x, m, n)**: 把矩阵分为子块再转换成元胞数组型。m和n是正整数向量，表示元胞数组行列的大小。m、n中所有元素之和分别是矩阵x的行数和列数
- **c = mat2cell(x, r)**: 返回一个列单元数组。正整数向量r各元素之和就是数组x的行数

## 元胞数组

### • 元胞数组的操作：元胞数组的显示

- 直接在命令窗输入单元数组名，可显示单元数组的构成单元，显示单元内容可使用**celldisp**函数

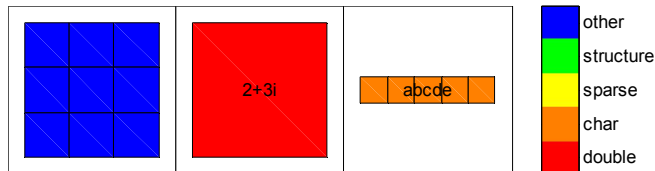
```
>> a={int8([2 4 7;3 9 6;1 8 5]), 2+3i,'abcde'};
>> a
a = [3x3 int8] [2.0000 + 3.0000i] 'abcde'
>> celldisp(a) %显示元胞数组内容
a{1} = 2 4 7
 3 9 6
 1 8 5
a{2} = 2.0000 + 3.0000i
a{3} = abcde
```

## 元胞数组

### • 元胞数组的操作：元胞数组的显示

- cellplot(A, 'legend') 绘制单元数组结构图, 参数legend控制是否显示色彩图例

```
>> a={int8([2 4 7;3 9 6;1 8 5]), 2+3i,'abcde'};
>> celldisp(a) %图形显示元胞数组内容
```



## 结构数组

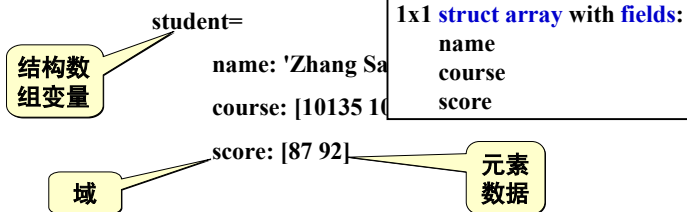
- 类似元胞数组，结构数组也能在一个数组里存放各类数据
- 构数组的基本成分是**结构**，即根据字段组合起来的不同类型的数据集合
- 结构必须在划分“域”后才能使用。数据不能直接存放于结构中，而只能存放在域中
- 结构的域可以存放任何类型和大小的数组（如任意维数值数组、字符串数组、符号对象等）。而且，不同结构的同名域中存放的内容可以不同
- 对结构的编址方法也有单下标编址和全下标编址两种

## 结构数组

### • 创建结构数组

- 利用赋值语句创建结构

```
>> student(1).name='Zhang San';
>> student(1).course=[10135 10096];
>> student(1).score=[87 92]
```

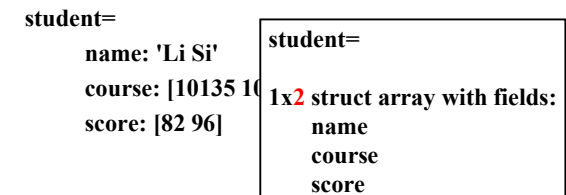


## 结构数组

### • 创建结构数组

- 利用创建赋值语句继续扩展

```
>> student(2).name='Li Si';
>> student(2).course=[10135 10096];
>> student(2).score=[82 96]
```





## 结构数组

### • 创建结构数组

- 利用结构函数`struct`创建结构

```
>> student(3)=struct('name','Wang Wu','course',[10568
10063],'score',[79 86])
```

```
>> student(4)=struct('name','Zhao Liu','course',[10568
10063],'score',[91 80])
```

student = 1x4 struct array with fields:

```
name
course
score
```

## 结构数组

### • 访问结构数组中的数据

- 利用结构数组索引可以对结构数组的字段值或字段元素值进行访问或赋值;也可利用`getfield`函数访问

```
>> str1=student(2).name
```

```
str1 = Li Si
```

```
>> getfield(student(2), 'name')
```

```
ans = Li Si
```

```
>> student.name
```

```
ans = Zhang San
```

```
ans = Li Si
```

```
ans = Wang Wu
```

```
ans = Zhao Liu
```

## 结构数组

### • 添加域

- 可以利用赋值语句添加域
- 还可利用`setfield`函数添加字段或修改域的值

```
>> student(1).sex='male'
```

student = 1x4 struct array with fields:

```
name
course
score
sex
```

## 结构数组

### • 删除域

- 可利用`rmfield`函数删除结构数组域的删除

```
>> student=rmfield(student, {'course' 'score'})
```

student = 1x4 struct array with fields:

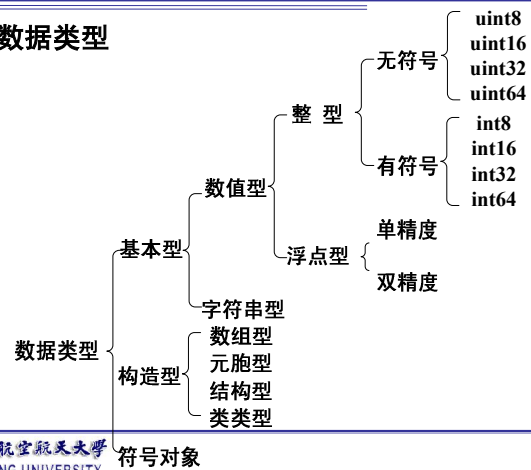
```
name
sex
```

```
>> student(1).course
```

Reference to non-existent field 'course'.

## MATLAB的数据类型

### • 主要数据类型



## 课外实验一

- **实验名称：** MATLAB的数据类型与运算
- **实验目的：** 通过实验使得学生了解MATLAB的工作环境、数据类型及其基本运算。
- **实验内容：**
  - 熟悉MATLAB的工作环境
  - 掌握MATLAB数值数组、矩阵、元胞结构和结构数组等主要数据类型
  - 掌握MATLAB主要数据类型的基本运算操作