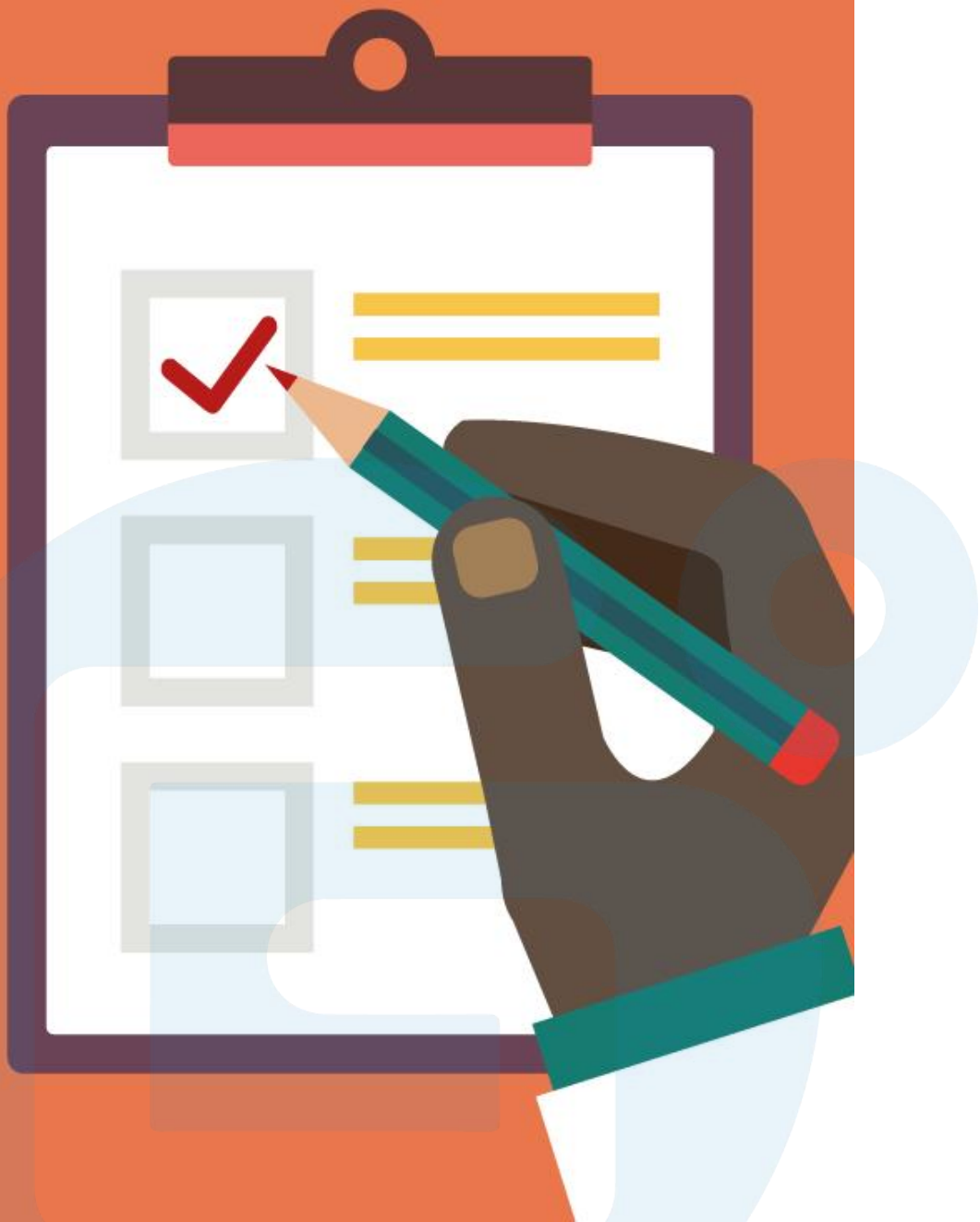


Talk is cheap, show me the code

第一课：Matlab入门

Matlab入门课程系列



OUTLINE

- Matlab简介
 - Matlab数据类型
 - 下节课预习
-



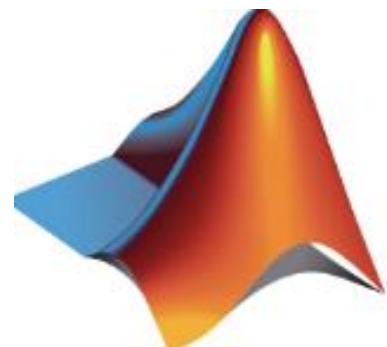
一 Matlab简介



- Matlab: 矩阵实验室(Matrix Laboratory)的简称, 1984年 MathWorks开发的商业数值计算软件
- 编程语言/开发环境/大号计算器/工业模拟软件/绘图软件.....

- **主要特点:**

- 强大的数值计算、**矩阵处理能力**, 优化后的计算速度
- 极其**丰富的toolbox**, **图形界面操作**, 可用于各行各业
- 收费, 正版Matlab由基础数学包+各个toolbox+simulink组成。



- **谁会用到Matlab**

几乎所有要和计算、数学打交道的专业, 都可以利用matlab作为生产工具。包括信号处理、数值计算、动态仿真、金融经济、控制系统、图像处理、机器学习等.....

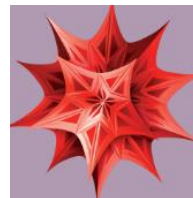
- **推荐阅读:** 链接: <https://pan.baidu.com/s/1ul-IduZHMrXlgY0ApWH4-g> 提取码: k8dd

- 除了Matlab以外，常使用一些类似的数学软件和编程语言，例如商业软件里的Mathematica和Maple，编程语言里的Fortran、python和R。

- 商业软件：

- Mathematica

- Maple



Mathematica是专为研究人员开发的。横向比较的话Mathematica是最好的物理学科研的工具，Matlab是最好的技术科学数值求解的工具，Maple在符号计算方面则有更丰富的经验，更适合教学，也更便宜。

- 编程语言：

- Fortran

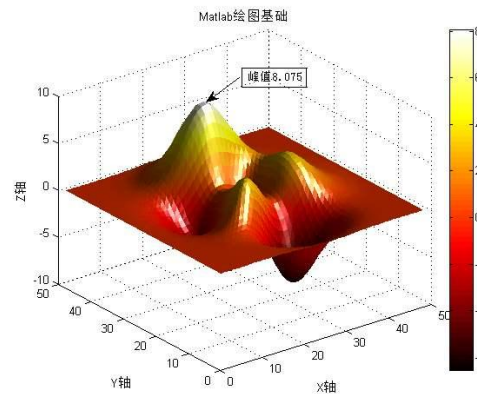
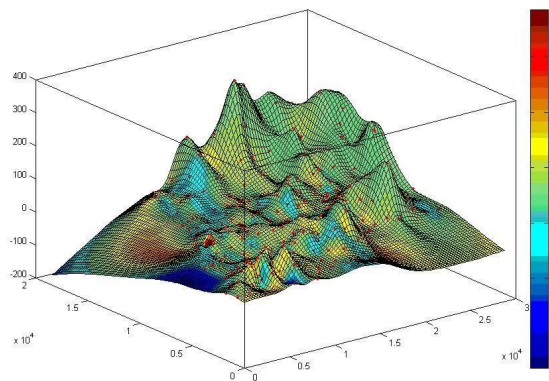
- python(numpy+scipy+matplotlib+pandas)

- R

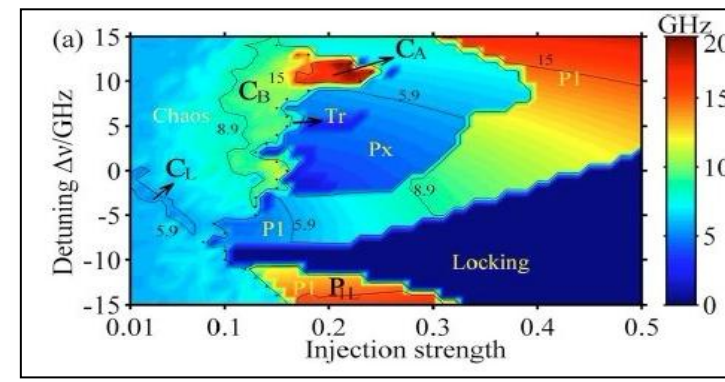
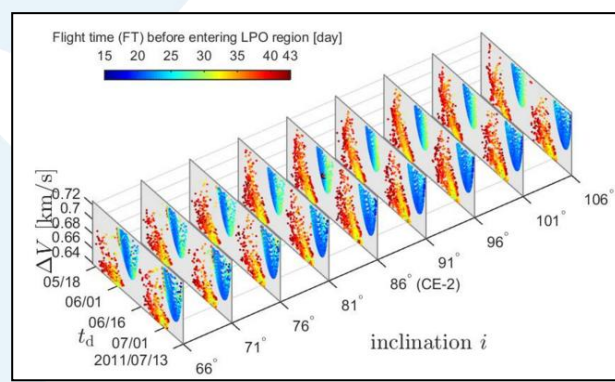
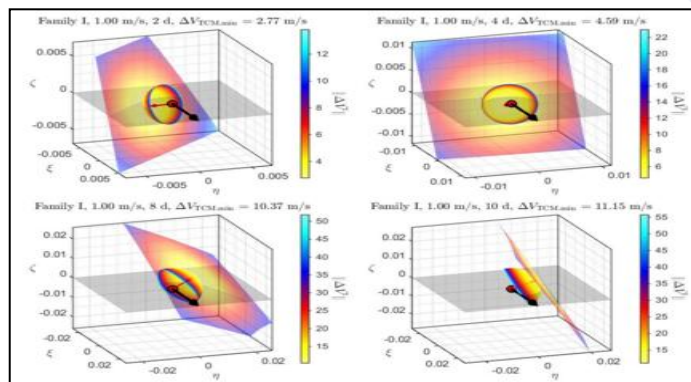
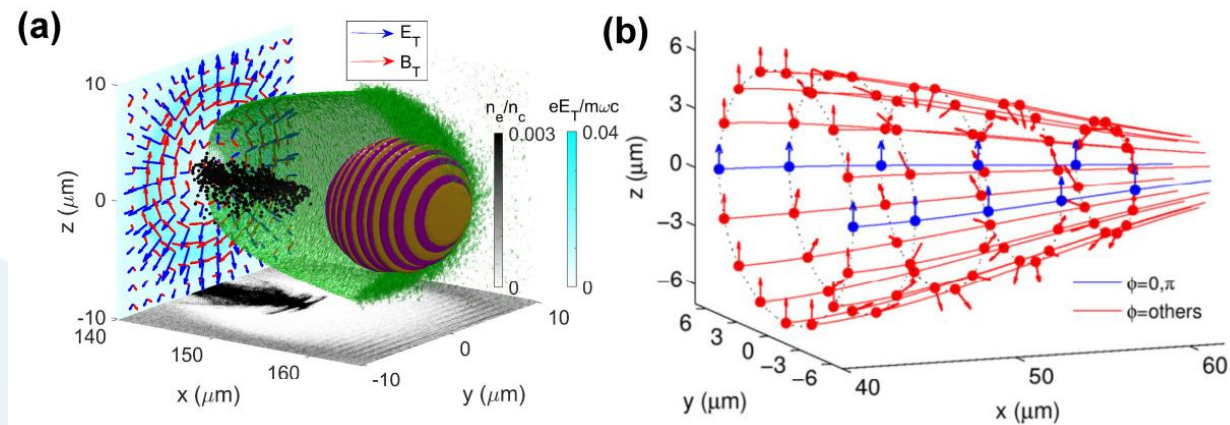
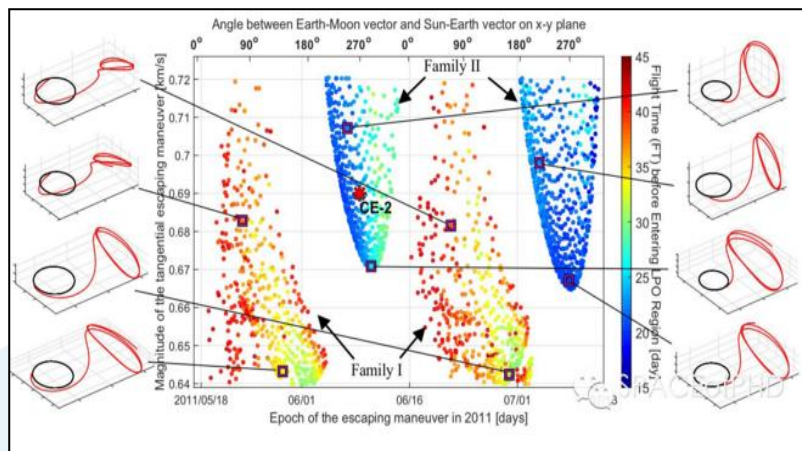
古老的数值语言fortran，新兴的python，统计学里的R。主要特点：免费开源。

为什么选择Matlab

- 计算功能简单可靠，不需要编写太多的代码。
- 绘图很方便，无需额外导出数据绘图，可以直接在matlab里生成图片。
- 功能强大的toolbox，方便实现专业功能。
- 详细而且丰富的帮助文档。
- 当然，还有一点，国内的大学和科研机构都在大规模使用matlab...



Matlab科研绘图展示



Matlab组成

- 开发环境 (IDE)

一套方便用户使用matlab函数和文件的工具集，包括图形化的用户接口

- 数学函数库

大量的算法工具函数，从基本的加减乘除，到复杂的矩阵求逆，傅里叶变换等

- 开发语言

尽管不是通用的编程语言，matlab语言却在数值计算方面做到了简单易学，功能完备

- 图形处理系统

可以直接从数据中生成图形，包括三维图、动画图等

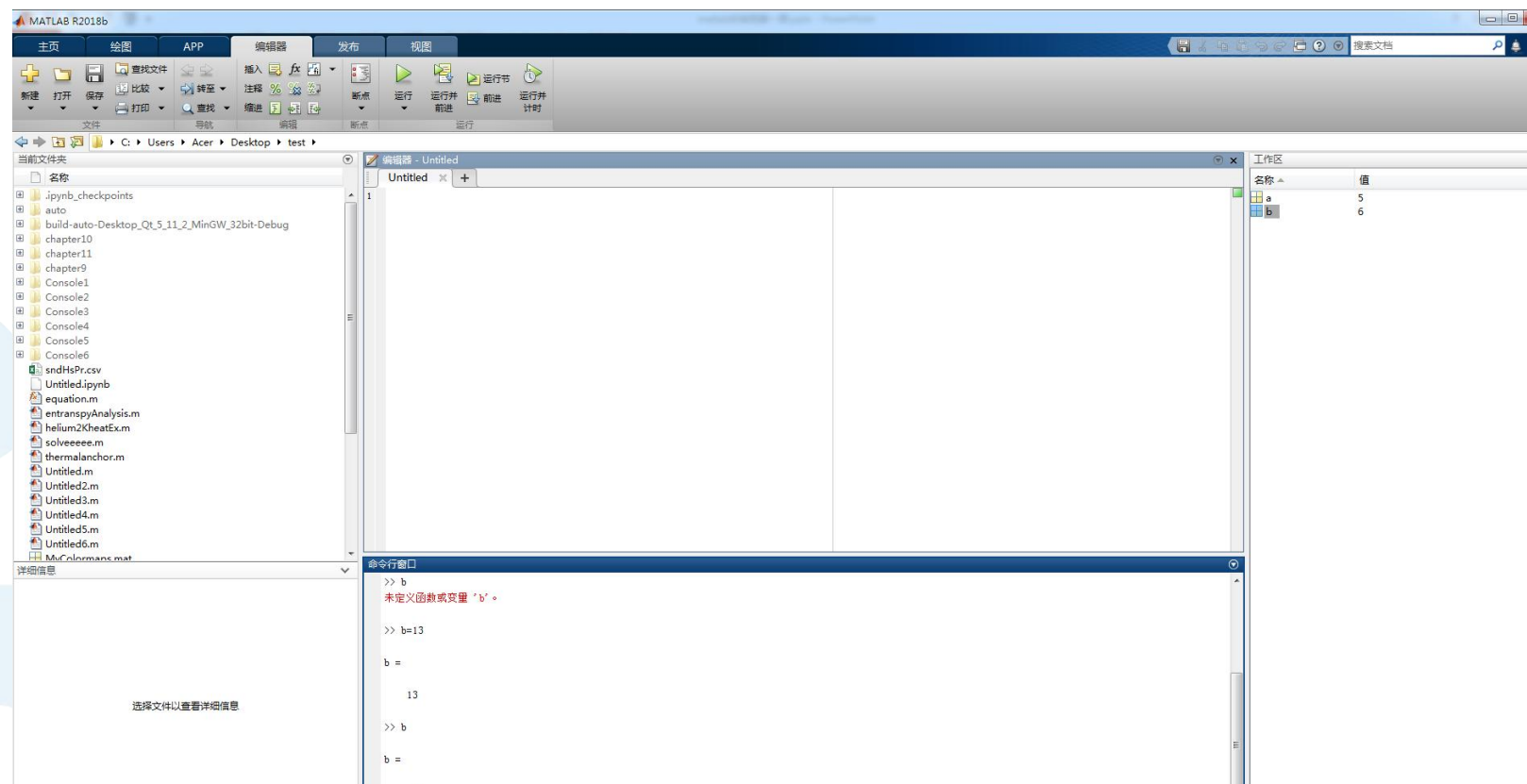
- 程序接口

可以调用c语言，fortran语言等

Matlab简介

Matlab界面

- 菜单栏/工具栏
- 命令行窗口
- 工作区



Matlab通用命令

命 令	命令说明	命 令	命令说明
cd	显示或改变当前工作文件夹	load	加载指定文件的变量
dir	显示当前文件夹或指定目录下的文件	diary	日志文件命令
clc	清除工作窗中的所有显示内容	!	调用 DOS 命令
home	将光标移至命令行窗口的最左上角	exit	退出 MATLAB
clf	清除图形窗口	quit	退出 MATLAB
type	显示文件内容	pack	收集内存碎片
clear	清理内存变量	hold	图形保持开关
echo	工作窗信息显示开关	path	显示搜索目录
disp	显示变量或文字内容	save	保存内存变量到指定文件

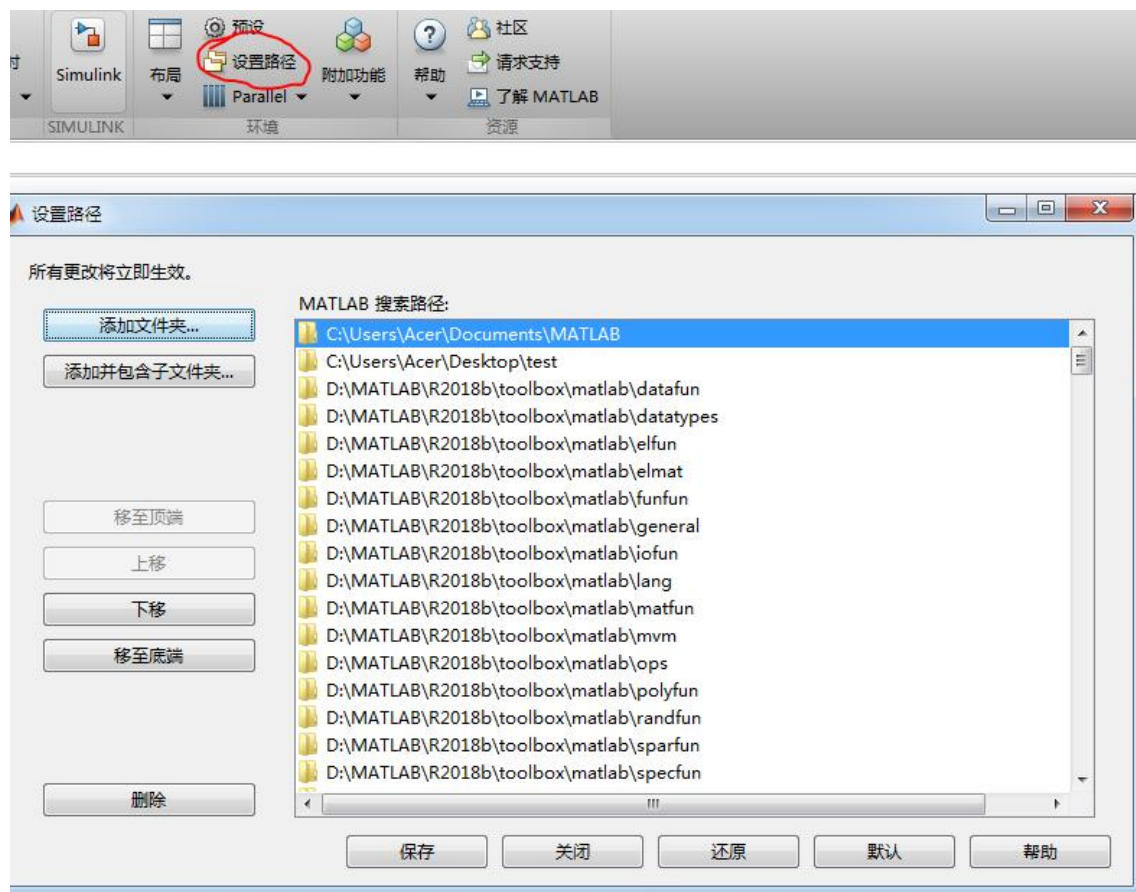
Matlab标点的使用

标 点	说 明	标 点	说 明
:	冒号，具有多种应用功能	%	百分号，注释标记
;	分号，区分行及取消运行结果显示	!	惊叹号，调用操作系统运算
,	逗号，区分列及函数参数分隔符	=	等号，赋值标记
()	括号，指定运算的优先级	'	单引号，字符串的标识符
[]	方括号，定义矩阵	.	小数点及对象域访问
{}	大括号，构造单元数组	...	续行符号

Matlab路径

matlab是一种较为智能的语言，变量在使用的时候无需事先声明，当用户输入一个字符串比如ABCDE的时候，程序会依次执行以下搜索动作：

1. 检查ABCDE是不是工作区里的变量名。
2. 检查是不是内置函数
3. 检查当前工作路径下有没有ABCDE.m
4. 检查路径里面有没有ABCDE.m
5. 都没有，就报错



Matlab帮助文档

- 文档是优秀软件的必要组成部分，关系到软件的原理、维护、记录、交流、改进等一系列工作。matlab的流行很大程度上依赖于其优秀的帮助系统。

- 文本帮助:

常用命令为help lookfor which doc get type+ fun name

- 演示帮助（在线）:

demos

- 帮助导航器（在线）:

doc helpbrowser



Matlab小练习

在命令行窗口（console）进行简单的编码练习，注意区分“；”和“shift+enter”的用法
（简单的大号计算器功能）

```
命令行窗口
>> x=rand(3,2);
>> y=rand(2,3);
>> A=sin(x)
B=cos(2*y)

A =

    0.7275    0.7916
    0.7869    0.5910
    0.1266    0.0974

B =

    0.8488   -0.3375    0.9507
    0.4591   -0.3513   -0.3620
```



二 Matlab数据类型



Matlab数据类型

- matlab里所有的变量都有各自的类型（type），最基础的包括数值类型、逻辑类型、字符串、函数句柄、结构体和单元数组6种。它们实际上都是按照数组的形式储存和操作的，另外还有面向对象的高级数据类型，类和java类。
- 数值类型是最常用的数据结构。主要包括整数、单精度浮点数、双精度浮点数、复数等。默认是按照双精度浮点数保持数值。

数据格式	示 例	说 明
int8, uint8 int16, uint16 int32, uint32 int64, uint64	int32(820)	有符号和无符号的整数类型 相同数值的整数类型占用比浮点类型更少的内存 除了 int64 和 uint64 类型外的所有整数类型，都可以进行数学运算
single	single(128.1)	单精度浮点类型 相同数值的单精度浮点类型比双精度浮点类型占用更少内存 精度与能够表示的数值范围比双精度浮点类型小
double	333.77 1.000-1.000i	双精度浮点类型，MATLAB 中默认的数据类型

- 数值类型/整数

整数类型	数值范围	转换函数
有符号 8 位整数	$-2^7 \sim 2^7 - 1$	int8
无符号 8 位整数	$0 \sim 2^8 - 1$	uint8
有符号 16 位整数	$-2^{15} \sim 2^{15} - 1$	int16
无符号 16 位整数	$0 \sim 2^{16} - 1$	uint16
有符号 32 位整数	$-2^{31} \sim 2^{31} - 1$	int32
无符号 32 位整数	$0 \sim 2^{32} - 1$	uint32
有符号 64 位整数	$-2^{63} \sim 2^{63} - 1$	int64
无符号 64 位整数	$0 \sim 2^{64} - 1$	uint64

- 取整函数

函 数	运算法则	示 例
floor(x)	向下取整	floor (1.2)=1 floor (2.5)=2 floor (- 2.5)= - 3
ceil(x)	向上取整	ceil (1.2)=2 ceil (2.5)=3 ceil (- 2.5)= - 2
round(x)	取最接近的整数 如果小数部分是 0.5，则向绝对值大的方向取整	round (1.2)=1 round (2.5)=3 round (- 2.5)= - 3
fix(x)	向 0 取整	fix(1.2)=1 fix(2.5)=2 fix(- 2.5)= - 2

- 数值类型/浮点数

浮点类型	存储位宽	各数位的意义	数值范围	转换函数
单精度	32	0~22 位表示小数部分 23~30 位表示指数部分 31 位表示符号 (0 正 1 负)	- 3.40282e+038 ~ - 1.17549e-038 1.17549e-038 ~ 3.40282e+038	single
双精度	64	0~51 位表示小数部分 52~62 位表示指数部分 63 位表示符号 (0 正 1 负)	- 1.79769e+308 ~ - 2.22507e-308 2.22507e - 308 ~ 1.79769e+308	double

- 单精度浮点数不能与整数进行数学运算，另外，所有计算机处理的数据，实际上都不是连续的
- eps函数可以用来显示某个数据距离它最近的一个数据的gap。
- 利用format long命令，或者直接在设置里调整显示方式，使matlab输出的数据精度更高。

- 数值类型/复数

可以直接利用关键字i和j作为虚部标志，也可以利用complex命令生成复数

函 数	说 明	函 数	说 明
real(z)	返回复数 z 的实部	imag(z)	返回复数 z 的虚部
abs(z)	返回复数 z 的模	angle(z)	返回复数 z 的辐角
conj(z)	返回复数 z 的共轭复数	complex(a,b)	以 a 为实部、 b 为虚部创建复数

- 无限和非数

Inf和Nan用来表示无穷量和非数

```
>> 0/0  
ans =  
     NaN  
>> 1/0  
ans =  
     Inf
```

- 逻辑类型

逻辑类型又叫布尔类型，只有0和1两个量。常用逻辑运算符和关系操作符

关系操作符	说 明
<	小于
<=	小于或等于
>	大于
>=	大于或等于
==	等于
~=	不等于

逻辑运算符	说 明
&	与
	或
~	非

- 字符和字符串类型

文本被转换为ASCII码储存在一维数组（向量）内，但是展示出来的形式是字符串。

```
>> str='hello world'
str =
    'hello world'
>> str1=abs(str)
str1 =
    104    101    108    108    111    32    119    111    114    108    100
>> str2=str1+1
str2 =
    105    102    109    109    112    33    120    112    115    109    101
>> str3=char(str2)
str3 =
    'ifmmp!xpsme'
```

```
>> str1(1:5:1)
ans =
    104
>> str(1:5:1)
ans =
    'h'
>> str(1:2)
ans =
    'he'
>> str(1:6)
ans =
    'hello '
```

- 函数句柄

- Matlab里对函数的调用分为直接调用和间接调用。
- 直接调用时，被调函数称为子函数，但是子函数只能被与其M文件同名的主函数或在M文件中的其它函数调用，且一个文件只能有一个主函数。
- 间接调用使用函数句柄调用。创建函数句柄利用操作符@进行声明。

Function_Handle=@function_filename

- 此外还可以利用@操作符声明匿名函数。多个匿名函数组合可以形成复杂的表达式。

```
% 函数句柄的创建 方式1 : 直接加@  
% 语法: @函数名fun1 = @sin;  
  
% 方式2 : str2func函数  
% 语法: str2fun('函数名')  
fun2 = str2func('cos');  
  
% 方式3 : 匿名函数  
% 语法: @(参数列表)单行表达式  
fun3 = @(x, y)x.^2 + y.^2;  
  
% 函数句柄的调用fun1(pi);
```

```
>> a=@(x,y)x*x+y;  
b=@(x,y)(x+5)*y;  
c=@(x,y)a(x,y)*b(x,y);  
x=2;y=1;  
c(x,y)  
ans =  
    35
```

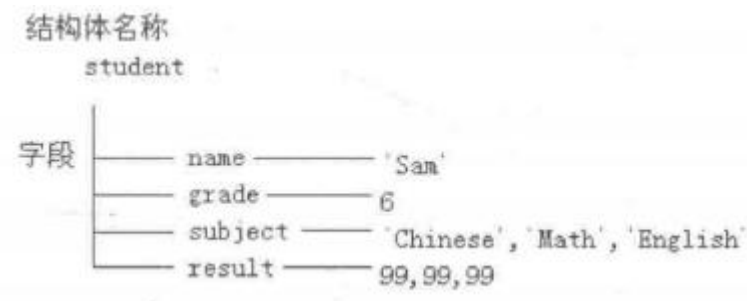

- 函数句柄

函数名称	函数功能
functions(funhandle)	返回一个结构体，存储了函数的名称、函数类型（simple 或 overloaded），以及函数 M 文件的位置
func2str(funhandle)	将函数句柄转换为函数名称字符串

函数名称	函数功能
str2func(str)	将字符串代表的函数转换为函数句柄
save filename.mat funhandle	将函数句柄保存在*.mat 文件中
load filename.mat funhandle	把*.mat 文件中存储的函数句柄加载到工作区
isa(var, 'function_handle')	检测变量 var 是否是函数句柄
isequal(funhandlea, funhandleb)	检测两个函数句柄是否对应于同一个函数

- 结构体

结构体是数据容器，把多个相关联但不同类型的数据封装在一个对象中。



- 有两种方法创建结构体:

1. 直接赋值
2. 使用struct函数

```
>> student.name='bob';
student.age=19;
student.ID=400888888;
student.Body=['60.5','kg','178.3','cm']
student =
    包含以下字段的 struct:

    name: 'bob'
    age: 19
    ID: 400888888
    Body: '60.5kg178.3cm'
```

```
>> Information=struct('name','John','age','22')
Information =
    包含以下字段的 struct:

    name: 'John'
    age: '22'

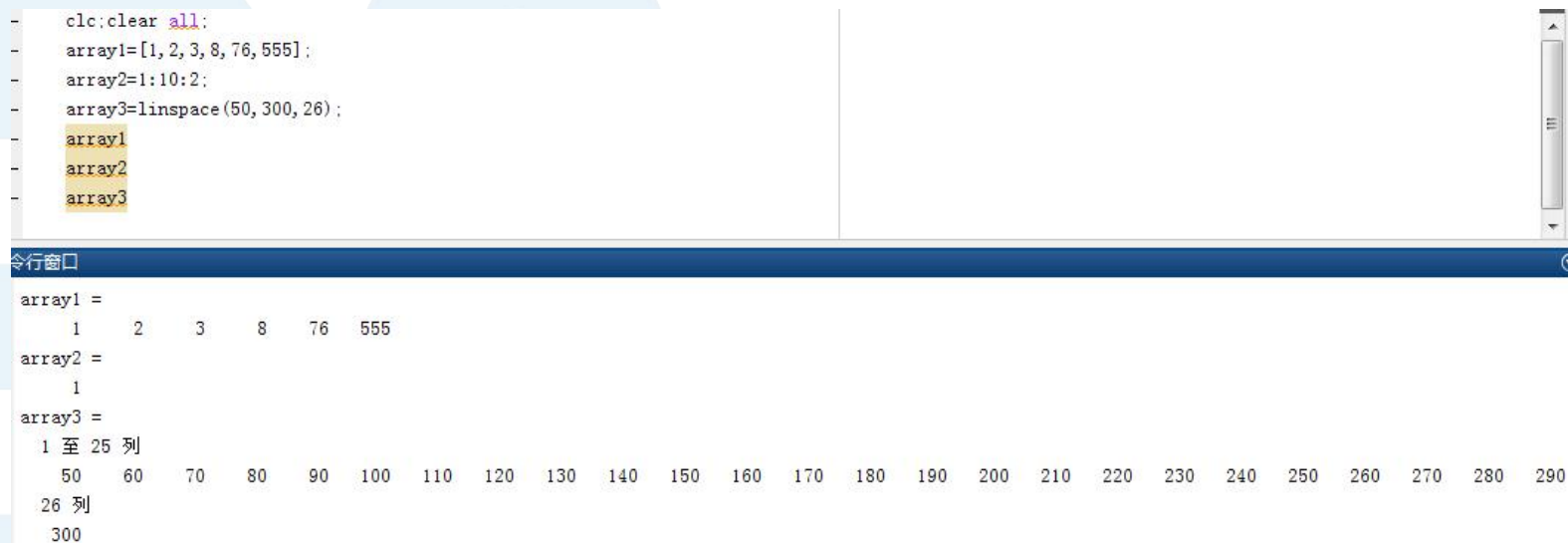
>> Information=struct('name','John','age',22)
Information =
    包含以下字段的 struct:

    name: 'John'
    age: 22

>> Information.age
ans =
    22
```

- 数组

- 数组是高维度的矩阵，且类型不限，可以是数值，也可以是别的类型。矩阵是数组的一个子类，更多的是数学上的定义。矩阵在matlab里一般特指二维数值数组，它的存储和运算采用线性代数法则，与数组不同，需要特别注意。
- 采用中括号来声明一个数组。



The screenshot shows the MATLAB Command Window with the following code and output:

```
clc;clear all;  
array1=[1, 2, 3, 8, 76, 555];  
array2=1:10:2;  
array3=linspace(50, 300, 26);  
array1  
array2  
array3
```

The output displays the contents of the arrays:

```
array1 =  
     1     2     3     8    76   555  
  
array2 =  
     1  
  
array3 =  
  1 至 25 列  
    50    60    70    80    90   100   110   120   130   140   150   160   170   180   190   200   210   220   230   240   250   260   270   280   290  
 26 列  
    300
```

- 单元数组

- cell, 单元数组是广义上的多维数组，组成单元数组的每个cell（单元/元胞）可以是数值，也可以是矩阵，还可以是另一个cell。
- 创建方法有3种：直接赋值法、利用函数cell()、利用{}直接创建cell数组的所有单元。

两种赋值方法：内容索引法（content indexing）和单元索引法（cell indexing）。

（1）内容索引法：赋值语句的左边用大括号 {} 将标识单元的下标括起来，右边为单元的内容。

```
>> C{1,1}='this is a cell_array';
>> C{1,2}=[23,-4,45;-34,567,2];
>> C{2,1}=[];
>> C{2,2}=324-454i;
>> class(C)

ans =

'cell'
```

```
>> C

C =

2x2 cell 数组

{'this is a cell_array'} {'23,-4,45;-34,567,2' }
{0x0 double } {[3.2400e+02 - 4.5400e+02i]}
```

（2）单元索引法：赋值语句的左边用小括号（）将标识单元的下标括起来，右边用大括号将存储于单元中的数据括起来。

```
>> C(1,1)='this is a cell_array';
>> C(1,2)=[23,-4,45;-34,567,2];
>> C(2,1)=[];
>> C(2,2)=324-454i;
>> whos
```

Name	Size	Bytes	Class	Attributes
C	2x2	552	cell	
U	62101x10	4968080	double	
V	198344x10	15867520	double	
ans	1x4	8	char	
cell	1x5	704	cell	
seed	1x1	8	double	
session	475504x1	443973576	cell	
stu	1x2	918	struct	

```
>> C

C =

2x2 cell 数组

{'this is a cell_array'} {2x3 double }
{0x0 double } {[3.2400e+02 - 4.5400e+02i]}
```

- 单元数组

直接赋值法、利用函数cell()、利用{}直接创建cell数组的所有单元。

利用函数cell() 可以创建一个元胞数组，还可以规定其大小。

```
%此即为创建了一个 m×n 的元胞数组。  
%但这只是预定义一个元胞数组，后面需要利用赋值法对元胞的每个单元进行初始化赋值。方法同直接赋值法。  
cell_array=cell(m,n)
```

这是一种快捷的元胞数组创建方法，利用 {} 可以直接快速地创建所需元胞数组。在大括号中一次列出所需创建元胞数组的内容，它们之间用逗号 (,) 隔开，行与行之间用分号 (;) 隔开，这样就可以创建一个已经初始化好的元胞数组。

利用 {} 创建元胞数组的格式为：

```
cellname={cell 1,1},[cell 1,2],.....,[cell 1,n] ; ..... ; [cell m,1],[cell m,2],.....,[cell m,n]; %此即为创建了一个 m×n 的元胞数组，并将单元进行了初始化。
```

```
>> cell={ 'This ','is ','cellarray' }, [1253,45], [ ];214,15-56i,eye(4))  
  
cell =  
  
2×3 cell 数组  
  
{ 'This is cellarray' } {1×2 double } {0×0 double}  
{ [ 214] } {[15.0000 -56.0000i]} {4×4 double}
```

- 单元数组的寻址

在编程时，一个元胞数组的每一个单元都是一个指针，指向其他的数据结构，而这些数据结构可以是不同的数据类型。在MATLAB中，元胞数组创建时，没有显示内容单元的内容，显示的只是它相应的数据类型。而MATLAB可以通过单元的名称来寻访其单元包含的内容。

在MATLAB中，元胞数组的寻访有以下两种形式。

(1) 用()将下标括起来寻访其单元的结构类型。

(2) 用{}将下标括起来寻访其单元的内容。

```
>> A=cell(1,1)

A =

1×1 cell 数组

{'This is cellarray'}
```

```
>> a=cell{1,1}

a =

'This is cellarray'
```

```
B =

1×1 cell 数组

{1×2 double}
```

```
>> b=cell{1,2}

b =

1253 45
```

(3) 此外，可以用celldisp()函数对元胞数组的所有内容进行寻访。

```
>> celldisp(cell)
```

```
cell{1,1} =

This is cellarray
```

```
cell{2,1} =

214
```

```
cell{1,2} =

1253 45
```

```
cell{2,2} =

15.0000 -56.0000i
```

```
cell{1,3} =
```

```
[]
```

```
cell{2,3} =
```

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

- 单元数组还可以执行合并、删除、修改、重塑等操作，与矩阵操作类似

Map类型

map 是 map 类中的对象，由一个 `containers` 包来定义，可以通过构造函数来实现：

```
>> map = containers.Map({'Monday', 'Tuesday', 'Wednesday'}, {1,2,3})  
  
map =  
  
Map with properties:  
  
    Count: 3  
    KeyType: char  
    ValueType: double
```

上面是一个直接的创建方法，map对象的创建过程可以分为：

1. 创建一个空map对象
2. 使用 `keys` 和 `values` 方法对其进行内容填充

使用下面的代码创建一个空map：

```
>> emptyMap = containers.Map()  
  
emptyMap =  
  
Map with properties:  
  
    Count: 0  
    KeyType: char  
    ValueType: any
```


Map类型

读取 map 对象

使用 `keys` 和 `maps` 函数可以查看 map 对象的内容：

```
>> keys(map)
ans =
    'Monday'    'Tuesday'    'Wednesday'
>> values(map)
ans =
    [1]    [2]    [3]
```

使用 `()` 实现 通过键值访问数值功能，这个是核心功能：

```
>> map('Monday')
ans =
    1
```

删除

使用 `remove` 函数 删除 键-值 对：

```
>> remove(map, 'Wednesday')
ans =
Map with properties:
    Count: 2
    KeyType: char
    ValueType: double
```

修改

使用 `()` 操作直接修改即可：

```
>> map('Monday') = 11
map =
Map with properties:
    Count: 2
    KeyType: char
    ValueType: double
>> map('Monday')
ans =
    11
```

添加

和修改一样，使用 `()` 操作：

```
>> map('Friday') = 5
map =
Map with properties:
    Count: 3
    KeyType: char
    ValueType: double
```



三 下节课预习



1. 数组和矩阵的基本知识。数组运算、矩阵操作、矩阵元素计算、矩阵运算等。
2. matlab编程基础。.M文件编辑器、变量的跟踪与传递、程序结构（循环、条件等）、脚本与函数基础、调试程序。
3. 基本的计算性能优化策略。



感谢参与 下堂课见

